Domain-Sum Feature Transformation For Multi-Target Domain Adaptation

Takumi Kobayashi[1,2]
takumi.kobayashi@aist.go.jp
Lincon S. Souza[1]
lincon.souza@aist.go.jp
Kazuhiro Fukui[2]
kfukui@cs.tsukuba.ac.jp

[1] National Institute of Advanced Industrial Science and Technology (AIST)
Tsukuba, Japan

[2] University of Tsukuba
Tsukuba, Japan

Abstract

Domain adaptation effectively transfers a learner from a source domain to a target domain. Recent deep methods are based on detailed comparison between a pair of source and target domains, which makes it less applicable to multiple domains. In this paper, we address the domain adaptation on the basis of *subspace* which provides more robust metric. We analyze the subspace methods in domain adaptation to theoretically derive a subspace-based feature transformation in an efficient form of simple summation. It intrinsically contributes to closing a gap between source and target subspaces in an end-to-end deep framework. Besides, due to the robust representation of subspace and the simple transformation, the proposed method naturally deals with multiple domains both for source and target in contrast to previous approaches. *Multi-target* domain adaptation especially provides efficient inference to process multiple target domains by only a single model. In the experiments on visual domain adaptation tasks, the proposed method exhibits favorable performance in a scenario of the multi-target domains.

1 Introduction

It is costly to annotate a huge number of samples required to train deep neural networks (DNNs), which limits the annotation only to samples of a *source* domain, while test samples can be drawn from a different *target* domain. Domain adaptation [26, 33] fills a gap between the source and target domain so as to effectively transfer the learner.

Deep domain adaptation [26] based on end-to-end trainable DNNs has attracted keen attention in recent years. It leverages DNN of favorable transferability [1, 29] to transfer classifiers trained on a source domain into a target domain by means of regularization to align the source distribution with the target one. The regularization losses are formulated, for example, by adversarial techniques [5, 12, 13, 15, 23] for minimizing the discrepancy between source and target domains. Statistical moment matching also contributes to regularization through minimizing the discrepancy of mean (MMD) [14], content moment [31] and higher-order moment [17, 20]. While they globally adapt the source domain to the target, some works have recently addressed local sub-domain adaptation [25, 28, 32]. Class-related

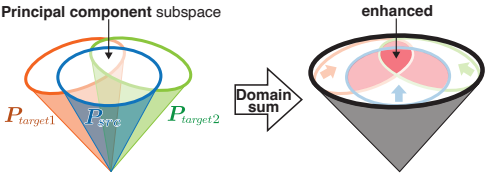


Figure 1: Multiple domain subspaces $\{\boldsymbol{P}_{src}, \boldsymbol{P}_{target1}, \boldsymbol{P}_{target2}\}$ are uniformly aggregated to produce our transformation, which enhances the principal component subspace consistent across domains. The principal component subspace is enlarged through end-to-end learning.

local distributions are useful for minutely aligning feature representation across the source and target domains; diversity over class categories also enhances the adaptation [2].

While those approaches exploit detailed characteristics of feature distributions, we focus on a *subspace* representation, a more general structure of features than the distributions; distribution and its finer structures related to class discrimination are *contained in* a subspace. Thus, the subspace is suitable for robustly describing a domain due to the following two reasons. (1) Subspace is an unsupervised representation and less dependent on the details of distributions, such as moments and local structures; for example, a subspace resorts only to eigenvectors of a matrix $\sum_i \boldsymbol{x}_i \boldsymbol{x}_i^\top$, while moment-based statistics are additionally related to the eigenvalues. (2) Subspace is estimated robustly against sampling-related issues particularly found in mini-batches; the subspace computed from *few* samples, even drawn from a domain in a *biased* way, is contained in the (inherent) subspace of that domain, while the detailed characteristics of distribution are poorly estimated due to such sampling issues.

The subspace representation attracted attention in *shallow* adaptation approaches [7, 8, 10, 21, 22]. In [8], the *Grassmann manifold* is introduced to describe the domain shift from the subspace viewpoint mathematically; a domain is shifted along a geodesic path on the manifold from the source-domain subspace to the target subspace. It leads to geodesic flow kernel (GFK) [7] by theoretically formulating the feature transformation in a closed form. Those approaches provide transformation of input features by means of subspaces. They, however, demand complex computation processes less suitable for end-to-end learning. Multiple subspaces per domain are dealt with in [22], though resorting to greedy subspace matching for a shallow domain adaptation in a scenario of single source and single target.

In this paper, by analyzing the subspace method, we propose an effective feature transformation method to reduce discrepancy between source and target subspaces in a deep framework. The proposed method is theoretically derived from the geodesic flow kernel (GFK) [7, 8] and is formulated in an efficient form simple enough to be differentiable. Our feature transformation is capable of bridging a gap between source and target subspaces intrinsically through end-to-end feature learning in a deep domain adaptation framework. The method uniformly aggregates domain-specific subspaces, regardless of source or target, as shown in Fig. 1. Therefore, we can naturally address the adaptation for *multiple target domains* which is a more generalized scenario than the standard setting of single-target domain. The multi-target domain adaptation (DA) renders a *single* versatile model applicable to classifications on multiple target domains without pre-identifying the target domain of an input sample nor preparing multiple models tailored for respective domains in a memory-consuming way. Thus, by the multi-target adaptation, we can enjoy a simple and efficient inference. While multi-target DA is addressed in a shallow framework [30], some

deep approaches are applied to this task by stacking multiple networks [6] and learning multiple teacher models [16] in less efficient ways. In contrast, our method is formulated in a simple yet effective way through summation of subspaces, thus being applicable even to multi-source and multi-target adaptation. Besides, by resorting to the subspace, the proposed method is complementary to the other deep domain adaptation approaches based on regularization of distributions, which are contained in subspaces. It is applicable in conjunction with the regularization losses for further boosting performance.

2 Method

We first review the geodesic flow kernel (GFK) [7] which describes domain shifts from a Grassmannian viewpoint, and then formulate the proposed feature transformation method.

2.1 Subspace-based domain adaptation

Training samples on a *source* domain $\mathcal{S}$ are equipped with class label $y_i \in \{1,\cdots,C\}$ and feature representation $\boldsymbol{x}_i^{\mathcal{S}} \in \mathbb{R}^d$ to build a labeled set of $\{\boldsymbol{x}_i^{\mathcal{S}}, y_i\}_{i=1}^{n_{\mathcal{S}}}$. Our objective is to construct a classifier discriminating C classes on a *target* domain $\mathcal{T}$ which provides only sample features $\{\boldsymbol{x}_j^{\mathcal{T}}\}_{j=1}^{n_{\mathcal{T}}}$ without any annotation. The target domain contrasts with the source one such as in terms of image styles (Fig. 3), thereby making it hard to directly transfer a classifier optimized on the source domain. The domains are characterized by *subspaces* such that

$$\boldsymbol{x}_i^{\mathcal{S}} \approx \boldsymbol{U}_{\mathcal{S}}\boldsymbol{U}_{\mathcal{S}}^{\top}\boldsymbol{x}_i^{\mathcal{S}} = \boldsymbol{P}_{\mathcal{S}}\boldsymbol{x}_i^{\mathcal{S}}, \quad \boldsymbol{x}_j^{\mathcal{T}} \approx \boldsymbol{U}_{\mathcal{T}}\boldsymbol{U}_{\mathcal{T}}^{\top}\boldsymbol{x}_j^{\mathcal{T}} = \boldsymbol{P}_{\mathcal{T}}\boldsymbol{x}_j^{\mathcal{T}}, \quad \forall i,j, \tag{1}$$

where $\boldsymbol{U}_{\mathcal{S}} \in \mathbb{R}^{d\times r}$ and $\boldsymbol{U}_{\mathcal{T}} \in \mathbb{R}^{d\times r}$ are orthonormal bases of rank r for the source- and target-domain subspaces, respectively; they are practically computed by applying singular-value decomposition (SVD) to $\boldsymbol{X}_{\mathcal{S}} = [\boldsymbol{x}_1^{\mathcal{S}},\cdots,\boldsymbol{x}_{n_{\mathcal{S}}}^{\mathcal{S}}]$ and $\boldsymbol{X}_{\mathcal{T}} = [\boldsymbol{x}_1^{\mathcal{T}},\cdots,\boldsymbol{x}_{n_{\mathcal{T}}}^{\mathcal{T}}]$. From a geometric viewpoint, those subspaces are represented by two points on a Grassmann manifold, which are smoothly connected via a geodesic path on the manifold [8]. By using the complementary subspace basis $\bar{\boldsymbol{U}}_{\mathcal{S}} \in \mathbb{R}^{d\times d-r}$ s.t. $\boldsymbol{U}_{\mathcal{S}}^{\top}\bar{\boldsymbol{U}}_{\mathcal{S}} = \boldsymbol{0}$, the geodesic flow [7] is formulated as

$$\boldsymbol{U}_{(t)} = \boldsymbol{U}_{\mathcal{S}}\boldsymbol{S}\boldsymbol{\Gamma}_{(t)} - \bar{\boldsymbol{U}}_{\mathcal{S}}\bar{\boldsymbol{S}}\boldsymbol{\Sigma}_{(t)}, \; t \in [0,1], \tag{2}$$

where $\boldsymbol{S} \in \mathbb{R}^{r\times r}$ and $\bar{\boldsymbol{S}} \in \mathbb{R}^{d-r\times r}$ are orthonormal matrices given by the generalized SVD of

$$\boldsymbol{U}_{\mathcal{S}}^{\top}\boldsymbol{U}_{\mathcal{T}} = \boldsymbol{S}\boldsymbol{\Gamma}\boldsymbol{T}^{\top}, \quad \bar{\boldsymbol{U}}_{\mathcal{S}}^{\top}\boldsymbol{U}_{\mathcal{T}} = -\bar{\boldsymbol{S}}\boldsymbol{\Sigma}\boldsymbol{T}^{\top}, \tag{3}$$

which uses the orthonormal matrix $\boldsymbol{T} \in \mathbb{R}^{r\times r}$. The diagonal matrix $\boldsymbol{\Gamma}$ and $\boldsymbol{\Sigma}$ are composed of $\{\cos\theta_k\}_{k=1}^r$ and $\{\sin\theta_k\}_{k=1}^r$ based on the canonical angles $\{\theta_k\}_{k=1}^r$ between the source $\boldsymbol{U}_{\mathcal{S}}$ and the target subspace $\boldsymbol{U}_{\mathcal{T}}$. The parametric form (2) is thus given by $\boldsymbol{\Gamma}_{(t)} = \mathrm{diag}(\{\cos(t\theta_k)\}_{k=1}^r)$ and $\boldsymbol{\Sigma}_{(t)} = \mathrm{diag}(\{\sin(t\theta_k)\}_{k=1}^r)$ to satisfy $\boldsymbol{U}_{(t=0)} = \boldsymbol{U}_{\mathcal{S}}$ and $\boldsymbol{U}_{(t=1)} = \boldsymbol{U}_{\mathcal{T}}$.

The geodesic flow (2) provides spectral features between source and target domains via projection $\boldsymbol{U}_{(t)}\boldsymbol{U}_{(t)}^{\top}\boldsymbol{x}$. While uniform sampling on the geodesic path produces a fixed-dimensional representation [8], geodesic flow kernel [7] is given by the integral on the path;

$$k(\boldsymbol{x}_i,\boldsymbol{x}_j) = \boldsymbol{x}_i^{\top}\boldsymbol{G}\boldsymbol{x}_j, \; \boldsymbol{G} = \int_{t=0}^{1}\boldsymbol{U}_{(t)}\boldsymbol{U}_{(t)}^{\top} = [\boldsymbol{U}_{\mathcal{S}}\boldsymbol{S}, \bar{\boldsymbol{U}}_{\mathcal{S}}\bar{\boldsymbol{S}}]\begin{bmatrix}\boldsymbol{\Lambda}_1 & \boldsymbol{\Lambda}_2 \\ \boldsymbol{\Lambda}_2 & \boldsymbol{\Lambda}_3\end{bmatrix}[\boldsymbol{U}_{\mathcal{S}}\boldsymbol{S}, \bar{\boldsymbol{U}}_{\mathcal{S}}\bar{\boldsymbol{S}}]^{\top}, \tag{4}$$

where the diagonal matrices $\boldsymbol{\Lambda}_1 \sim \boldsymbol{\Lambda}_3$ are

$$\boldsymbol{\Lambda}_1 = \mathsf{diag}\left(\left\{1+\frac{\sin(2\theta_k)}{2\theta_k}\right\}_{k=1}^{r}\right), \boldsymbol{\Lambda}_2 = \mathsf{diag}\left(\left\{\frac{\cos(2\theta_k)-1}{2\theta_k}\right\}_{k=1}^{r}\right), \boldsymbol{\Lambda}_3 = \mathsf{diag}\left(\left\{1-\frac{\sin(2\theta_k)}{2\theta_k}\right\}_{k=1}^{r}\right). \tag{5}$$

The matrix $\boldsymbol{G}^{\frac{1}{2}}$, square root of $\boldsymbol{G}$, renders an explicit feature transform of GFK as $\hat{\boldsymbol{x}} = \boldsymbol{G}^{\frac{1}{2}}\boldsymbol{x}$.

2.2 Domain-sum transformation based on subspace projection

The above approach leverages a geodesic path (2) to produce effective feature transformation in a Grassmannian manner. It, however, demands rather complicated computational processes, especially in generalized SVD (3), to construct the geodesic flow, which hinders us from incorporating the GFK into end-to-end learning. Thus, we propose a simpler formulation exploiting the subspace approach with a theoretical connection to GFK. It also gives interpretation and another formulation to GFK from a viewpoint of feature transformation.

We *roughly discretize* the integral (4) to a sum of two boundary points (source and target);

$$\boldsymbol{H} = \sum_{t\in\{0,1\}} \boldsymbol{U}_{(t)}\boldsymbol{U}_{(t)}^\top = \boldsymbol{U}_{(t=0)}\boldsymbol{U}_{(t=0)}^\top + \boldsymbol{U}_{(t=1)}\boldsymbol{U}_{(t=1)}^\top = \boldsymbol{U}_S\boldsymbol{U}_S^\top + \boldsymbol{U}_T\boldsymbol{U}_T^\top. \tag{6}$$

In spite of the rough approximation, the *domain-sum (DS)* matrix $\boldsymbol{H}$ is closely related to the GFK matrix $\boldsymbol{G}$ as follows; our propositions are proved in the supplementary material.

Proposition 1 *GFK matrix $\boldsymbol{G}$ (4) and DS matrix $\boldsymbol{H}$ (6) are similarly eigen-decomposed as*

$$\boldsymbol{G} = [\boldsymbol{U}_+, \boldsymbol{U}_-]\begin{bmatrix}\boldsymbol{\Psi}_+ & \\ & \boldsymbol{\Psi}_-\end{bmatrix}[\boldsymbol{U}_+, \boldsymbol{U}_-]^\top, \quad \boldsymbol{H} = [\boldsymbol{U}_+, \boldsymbol{U}_-]\begin{bmatrix}\boldsymbol{\Phi}_+ & \\ & \boldsymbol{\Phi}_-\end{bmatrix}[\boldsymbol{U}_+, \boldsymbol{U}_-]^\top, \tag{7}$$

where

$$\boldsymbol{U}_\pm = \mathsf{colnorm}(\boldsymbol{U}_S\boldsymbol{S} \pm \boldsymbol{U}_T\boldsymbol{T}), \boldsymbol{\Psi}_\pm = \mathsf{diag}(\{1 \pm \mathrm{sinc}\,\theta_k\}_{k=1}^r), \boldsymbol{\Phi}_\pm = \mathsf{diag}(\{1 \pm \cos\theta_k\}_{k=1}^r), \tag{8}$$

using column-wise normalization operator colnorm *to ensure orthonormality and* $\mathrm{sinc}(\theta) = \frac{\sin(\theta)}{\theta}$. *Difference between $\boldsymbol{G}$ and $\boldsymbol{H}$ is only in the functions of* sinc *and* cos *applied to the canonical angles $\{\theta_k\}_{k=1}^r$ to construct eigenvalues $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$.*

As shown in Fig. 2a, the eigenvectors $\boldsymbol{U}_+$ associated with the eigenvalues $\boldsymbol{\Phi}_+$ (and $\boldsymbol{\Psi}_+$) are sum of canonical vectors $\boldsymbol{U}_S\boldsymbol{S}$ and $\boldsymbol{U}_T\boldsymbol{T}$, indicating *principal component subspaces* consistent across the source and target domains, while $\boldsymbol{U}_-$ reflects the *difference subspaces* to discriminate two subspaces $\boldsymbol{U}_S$ and $\boldsymbol{U}_T$ [3, 4]. Thus, the projection via $\boldsymbol{U}_+$ would extract more effective features shared by two domains than $\boldsymbol{U}_-$. These analyses clarify that even the simple form $\boldsymbol{H}$ in (6) provides similar transformation to $\boldsymbol{G}$ in GFK (4). Namely, they both enhance the projection into the principal component subspace by the larger weights (eigenvalues) of $1+\mathrm{sinc}\,\theta$ in $\boldsymbol{G}$ and $1+\cos\theta$ in $\boldsymbol{H}$ while suppressing projection onto the difference subspaces via the smaller weights of $1-\mathrm{sinc}\,\theta$ and $1-\cos\theta$. These eigenvalues are compared in Fig. 2b depicting that sinc function further emphasizes/suppresses the principal/difference subspaces than cos. We can improve $\boldsymbol{H}$ based on cos by enhancing the weights on $\boldsymbol{U}_{+/-}$ as

$$\hat{\boldsymbol{H}} = \boldsymbol{H}^2 = (\boldsymbol{U}_S\boldsymbol{U}_S^\top + \boldsymbol{U}_T\boldsymbol{U}_T^\top)^2 = [\boldsymbol{U}_+, \boldsymbol{U}_-]\begin{bmatrix}\boldsymbol{\Phi}_+^2 & \\ & \boldsymbol{\Phi}_-^2\end{bmatrix}[\boldsymbol{U}_+, \boldsymbol{U}_-]^\top, \tag{9}$$

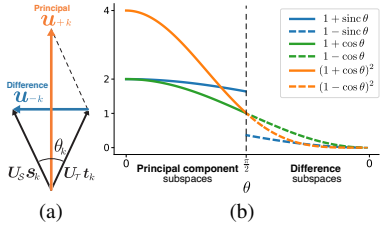


Figure 2: Eigen-decomposition of DS matrix $\boldsymbol{H}$ (6) and GFK matrix $\boldsymbol{G}$ (4). In (a), $\boldsymbol{u}_{\pm k}, \boldsymbol{s}_k$ and $\boldsymbol{t}_k$ are the k-th column vector of the matrices $\boldsymbol{U}_{\pm}$, $\boldsymbol{S}$ and $\boldsymbol{T}$ in (3).

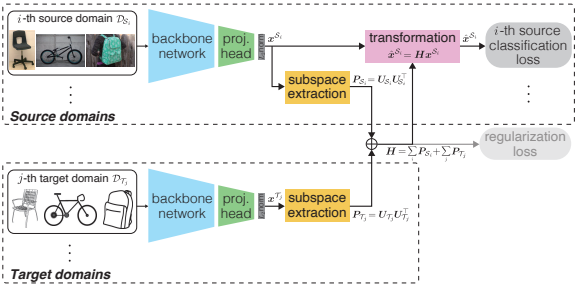


Figure 3: Proposed deep multi-domain adaptation.

of which eigenvalues $\boldsymbol{\Phi}_{\pm}^2 = \mathrm{diag}[\{(1 \pm \cos\theta_k)^2\}_{k=1}^r]$ are depicted in Fig. 2b compared to the other weighting. Thus, we propose the feature transformation using $\hat{\boldsymbol{H}}$;

- $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^\top \hat{\boldsymbol{H}} \boldsymbol{x}_j = (\boldsymbol{H}\boldsymbol{x}_i)^\top \boldsymbol{H}\boldsymbol{x}_j$ leads to our **domain-sum feature transformation** (DSFT) of

$$\hat{\boldsymbol{x}} = \boldsymbol{H}\boldsymbol{x} = (\boldsymbol{U}_S \boldsymbol{U}_S^\top + \boldsymbol{U}_T \boldsymbol{U}_T^\top)\boldsymbol{x} \tag{10}$$

  without applying SVD (3) nor square root of matrix in contrast to $\boldsymbol{G}^{\frac{1}{2}}$ in GFK.

- Weights $\boldsymbol{\Phi}_{\pm}^2$ enhance the principal component subspace $\boldsymbol{U}_+$ and suppress the difference subspace $\boldsymbol{U}_-$ more effectively than $\boldsymbol{\Phi}_{\pm}$ due to $(1+\cos\theta)^2 \geq 1+\cos\theta$ and $(1-\cos\theta)^2 < 1-\cos\theta$.

- The domain-sum matrix $\boldsymbol{H}$ uniformly aggregates domain subspaces in disregard of domain type, source or target. So, it is naturally extendable to *multiple* domains by

$$\boldsymbol{H} = \sum_{m=1}^{M} \boldsymbol{U}_m \boldsymbol{U}_m^\top, \tag{11}$$

  where M is the number of domains *both for source and target*, and $\boldsymbol{U}_m$ indicates the m-th domain subspace basis.

2.3 Deep multi-domain adaptation

The simple differentiable formulation (11) inspires us to embed the feature transformation into an end-to-end framework of deep domain adaptation. We can naturally cope with *multiple* domains *both* for source and target, as shown in Fig. 3.

The proposed transformation is applied on top of the backbone feature extractor (Fig. 3). The features produced by the backbone are normalized into unit L_2-norm for better feature representation learning [9, 11]. During training, we apply computationally stable SVD [27] to mini-batch samples for extracting domain subspaces $\{\boldsymbol{U}_m\}_{m=1}^M$, which are aggregated into the domain-sum matrix $\boldsymbol{H}$ in (11); the details about computing subspaces is shown in supplementary material. Input feature vector $\boldsymbol{x}^S$ of source domain is transformed by $\hat{\boldsymbol{x}}^S = \boldsymbol{H}\boldsymbol{x}^S$ which is then fed into a classifier module to finally produce the classification loss of softmax cross-entropy using class labels of the source samples. Domain subspaces are close to each

other through the end-to-end learning as shown below, and thus at inference a target-domain sample can be simply passed to the classifier without transformation.

The domain-sum transformation (10) endows the feature representation with high similarity across source and target domains as described in Sec. 2.2. Besides, the following proposition ensures that it contributes to closing a domain gap through end-to-end optimizing feature representation $\boldsymbol{x}$.

Proposition 2 *Suppose backbone DNN can flexibly produce a feature vector $\boldsymbol{x}$ which is normalized as $\|\boldsymbol{x}\|_2 = 1$; for a well separable classifier $\boldsymbol{W} = \{\boldsymbol{w}_c\}_{c=1}^C$, it can produce $\{\boldsymbol{x}_i\}_{i=1}^n$ such that $\boldsymbol{w}_{y_i}^\top(\boldsymbol{P}_\mathcal{S}+\boldsymbol{P}_\mathcal{T})\boldsymbol{x}_i \geq \boldsymbol{w}_c^\top(\boldsymbol{P}_\mathcal{S}+\boldsymbol{P}_\mathcal{T})\boldsymbol{x}_i\ \forall c$, where $\boldsymbol{P}_\mathcal{S}$ is a r-rank subspace projection matrix for $\boldsymbol{x}_i$, i.e., $\boldsymbol{x}_i = \boldsymbol{P}_\mathcal{S}\boldsymbol{x}_i$, and $\boldsymbol{P}_\mathcal{T}$ is an arbitrary subspace of rank r. We partition a feature space into $\mathbb{S}(\boldsymbol{\theta};\boldsymbol{P}_\mathcal{T}) = \{\boldsymbol{x} \in \mathrm{span}(\boldsymbol{P}_\mathcal{S}) | \|\boldsymbol{x}\|_2 = 1, \angle(\boldsymbol{P}_\mathcal{S}, \boldsymbol{P}_\mathcal{T}) = \boldsymbol{\theta}\}$ where $\angle$ is an operator to measure canonical angles. We define a softmax loss ℓ_{CE} optimized w.r.t $\boldsymbol{x} \in \mathbb{S}(\boldsymbol{\theta};\boldsymbol{P}_\mathcal{T})$ and $\boldsymbol{W}$, given $\boldsymbol{\theta}$ and $\boldsymbol{P}_\mathcal{T}$ as*

$$\ell_{CE}(\boldsymbol{\theta};\boldsymbol{P}_\mathcal{T}) = \min_{\{\boldsymbol{x}_i \in \mathbb{S}(\boldsymbol{\theta};\boldsymbol{P}_\mathcal{T})\}_{i=1}^n, \boldsymbol{W}} -\sum_{i=1}^n \log \frac{\exp(\boldsymbol{w}_{y_i}^\top(\boldsymbol{P}_\mathcal{S}+\boldsymbol{P}_\mathcal{T})\boldsymbol{x}_i)}{\exp(\sum_c \boldsymbol{w}_c^\top(\boldsymbol{P}_\mathcal{S}+\boldsymbol{P}_\mathcal{T})\boldsymbol{x}_i)}. \tag{12}$$

Then, we have the following relationship between the loss and the canonical angle $\boldsymbol{\theta}$;

$$\boldsymbol{\theta}^* \leq \boldsymbol{\theta} \ \Rightarrow\ \ell_{CE}(\boldsymbol{\theta}^*;\boldsymbol{P}_\mathcal{T}) \leq \ell_{CE}(\boldsymbol{\theta};\boldsymbol{P}_\mathcal{T}). \tag{13}$$

This proposition shows that, if we learn the model *with respect to* $\boldsymbol{\theta}$ of subspace angles, the softmax loss ℓ_{CE} is reduced by closing the angular gap. In particular, at the global minimum $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \ell_{CE}(\boldsymbol{\theta};\boldsymbol{P}_\mathcal{T})$, we have $\boldsymbol{\theta}^* = \boldsymbol{0}$; if $\exists k, \theta_k^* > 0$, we can further reduce the loss by $\hat{\theta}_k^* < \theta_k^*$, which contradicts the global optimality of $\boldsymbol{\theta}^*$ and induces $\boldsymbol{\theta}^* = \boldsymbol{0}$. Therefore, the proposition indicates that the transformation via $\boldsymbol{H} = \boldsymbol{P}_\mathcal{S} + \boldsymbol{P}_\mathcal{T}$ intrinsically works for *matching* subspaces across domains via simply minimizing a softmax loss in the end-to-end learning. It is applicable to multiple source and target domains. Thus, the domain-sum transformation is well compatible with our deep multiple domain adaptation in Fig. 3 .

We can further improve the proposed method in the following two points.

Regularization. As shown in Proposition 2, subspaces $\{\boldsymbol{P}_m = \boldsymbol{U}_m\boldsymbol{U}_m^\top\}_{m=1}^M$ get close to each other as end-to-end training proceeds. To further enhance the subspace matching, we propose the following regularization loss;

$$\ell_{reg} = -\frac{\|\boldsymbol{H}\|_F^2}{rM(M-1)} = \frac{1}{M-1} - \frac{1}{rM(M-1)} \sum_{i \neq j} \sum_{k=1}^r \cos^2 \theta_k^{(i,j)}. \tag{14}$$

By minimizing the regularization loss ℓ_{reg} together with the classification loss ℓ_{CE} via $\ell_{CE} + \eta\,\ell_{reg}$, the subspaces are well aligned to minimize the canonical angles $\{\boldsymbol{\theta}^{(i,j)}\}_{i,j}^M$; we set the balancing weight to $\eta = 1$.

Classifier subspace. Classification also considers *matching* between classifier vectors $\boldsymbol{W}$ and a feature vector $\boldsymbol{x}$. Thus, we can facilitate learning by increasing similarities not only among domains but also between domains and *classifier* $\boldsymbol{W}$. For that purpose, we can add the classifier subspace $\boldsymbol{U}_W\boldsymbol{U}_W^\top$ into the domain-sum matrix $\boldsymbol{H}$ (11) where $\boldsymbol{U}_W = \mathtt{svd}(\boldsymbol{W})$ is a classifier subspace basis; thereby, the domain-sum matrix is slightly modified to

$$\boldsymbol{H} = \boldsymbol{U}_W\boldsymbol{U}_W^\top + \sum_{m=1}^M \boldsymbol{U}_m\boldsymbol{U}_m^\top. \tag{15}$$

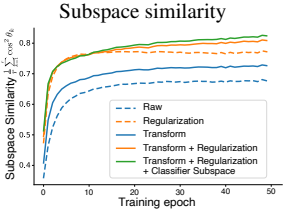
Table 1: Performance results (accuracy, %) of deep domain adaptation on Office-31 dataset in a scenario of single source and single target; we report averaged accuracy across various one-to-one adaptation. Detailed performances are shown in the supplementary material.

(a) Comparison

Method	Trans. matrix	Avg.
Raw	$\boldsymbol{I}$	81.94
Auto-Corr. (16)	$\boldsymbol{A}^{\frac{1}{2}}$	21.27
CORAL [21]	$\boldsymbol{C}_{\mathcal{T}}^{\frac{1}{2}}\boldsymbol{C}_{\mathcal{S}}^{-\frac{1}{2}}$	74.20
Principal component	$\boldsymbol{U}_{+}\boldsymbol{U}_{+}^{\top}$	80.81
GFK (7)	$\boldsymbol{G}^{\frac{1}{2}}$	84.58
Sum-of-subspaces (6)	$\boldsymbol{H}^{\frac{1}{2}}$	84.73
Ours (9)	$\boldsymbol{H}$	85.24

(b) Ablation study for extension methods (Sec. 2.3).

Trans. (10)	Reg. (14)	Cls. Sub.	Avg.
-	-	-	81.94
-	✓	-	84.45
✓	-	-	85.24
✓	✓	-	85.56
✓	✓($\eta = 2$)	-	84.59
✓	✓	✓	86.47



2.4 Discussion

In [10], GFK is simplified in a completely different way from ours by replacing the GFK matrix $\boldsymbol{G}$ (4) with an auto-correlation matrix written as

$$\boldsymbol{A} = \boldsymbol{X}_{\mathcal{S}}\boldsymbol{X}_{\mathcal{S}}^{\top} + \boldsymbol{X}_{\mathcal{T}}\boldsymbol{X}_{\mathcal{T}}^{\top} = \boldsymbol{U}_{\mathcal{S}}\boldsymbol{\Lambda}_{\mathcal{S}}^{2}\boldsymbol{U}_{\mathcal{S}}^{\top} + \boldsymbol{U}_{\mathcal{T}}\boldsymbol{\Lambda}_{\mathcal{T}}^{2}\boldsymbol{U}_{\mathcal{T}}^{\top}, \tag{16}$$

where $\boldsymbol{\Lambda}_{\mathcal{S}/\mathcal{T}}$ is singular value of $\boldsymbol{X}_{\mathcal{S}/\mathcal{T}}$. In contrast to the domain-sum matrix $\boldsymbol{H}$ (6), the matrix $\boldsymbol{A}$ contains statistics of the source and target distribution via $\boldsymbol{\Lambda}_{\mathcal{S}}$ and $\boldsymbol{\Lambda}_{\mathcal{T}}$, thereby highly biasing the feature transform to some principal directions. Besides, the dependency on statistics degrades robustness of subspaces discussed in Sec. 1.

The second-order statistics is also employed in the method of CORAL [20, 21]. For shallow adaptation, CORAL [21] provides transformation of source features as $\boldsymbol{C}_{\mathcal{T}}^{\frac{1}{2}}\boldsymbol{C}_{\mathcal{S}}^{-\frac{1}{2}}\boldsymbol{x}^{\mathcal{S}}$ using covariance matrices $\boldsymbol{C}_{\mathcal{S}/\mathcal{T}}$. As in the above approach (16), it is based on the second-order statistics, which are estimated less robustly than subspaces. CORAL is extended to deep adaptation by reformulating the concept of covariance matching into a regularization loss $\|\boldsymbol{C}_{\mathcal{S}} - \boldsymbol{C}_{\mathcal{T}}\|_F^2$ without feature transformation. In contrast, our transformation works cooperatively with the regularization (14) in a deep framework.

It is noteworthy that Proposition 1 reformulates GFK matrix $\boldsymbol{G}$ (4) into a simpler form based on eigen-decomposition of $\boldsymbol{H}$ in (7). The eigen-decomposition also provides us with a projection $\boldsymbol{U}_{+}\boldsymbol{U}_{+}^{\top}$ onto the principal component subspace, which is the special case of $\boldsymbol{H}$ by modifying $\cos\theta \to 1$ on eigenvalues in (8). It completely ignores the difference subspaces $\boldsymbol{U}_{-}$, while our $\boldsymbol{H}$ exploits all of them with proper weighting based on canonical angles $\{\theta_k\}_{k=1}^{r}$; those transformations are empirically compared in the comparison experiment of Table 1a.

3 Experimental results

We apply the proposed domain-sum feature transformation to visual domain adaptation tasks in which images are drawn from multiple modalities (Fig. 3). The detailed experimental settings and results are shown in the supplementary material.

Dataset. Methods are tested on Office-31 [19], Office-home [24], Adaptiope [18] and DomainNet [17] datasets, which poses image classification across multiple domains.

Table 2: Performance comparison (accuracy, %) in multi-target deep domain adaptation; e.g., "A→D,W" indicates transfer from source of Amazon to targets of DSLR and Webcam.

		Office-31 [19]				Office-Home [24]										
		A→ D,W	D→ A,W	W→ A,D	Avg.	A→ C,P,R	C→ A,P,R	P→ A,C,R	R→ A,C,P	Avg.	A,C→ P,R	P,R→ A,C	A,P→ C,R	C,R→ A,P	A,R→ C,P	C,P→ A,R	Avg.
Domain Concat.	raw	80.96	82.00	82.87	81.94	56.44	58.25	58.37	65.62	59.67	62.59	57.14	64.22	67.12	59.28	61.64	62.00
	DANN	81.59	80.03	82.67	81.43	47.04	50.39	54.99	63.60	54.00	60.08	59.58	64.63	65.22	57.54	58.60	60.94
	BNM	86.62	78.34	80.70	81.89	39.70	48.45	52.46	63.38	51.00	58.11	59.39	61.63	64.83	53.90	61.16	59.84
	SCDA	86.09	76.38	81.37	81.28	44.72	49.67	52.58	62.80	52.44	59.92	57.85	63.46	65.53	57.60	60.61	60.83
	DSAN	88.17	80.94	81.75	83.62	45.88	49.19	53.12	62.84	52.76	60.08	58.47	64.00	65.00	57.76	60.75	61.01
	Ours-c	88.38	84.89	85.67	86.31	67.50	67.04	63.22	68.62	66.59	77.72	63.55	70.80	75.94	71.12	73.47	72.10
Multi.	DANN	85.55	80.22	83.95	83.24	47.30	51.87	53.80	63.83	54.20	50.84	57.27	53.21	67.14	62.73	53.92	57.52
	BNM	86.06	75.49	82.00	81.18	41.59	48.60	51.73	63.25	51.29	50.51	58.28	53.67	69.59	62.41	56.03	58.41
	SCDA	86.81	76.18	82.05	81.68	46.29	51.02	53.19	64.47	53.74	51.19	57.83	54.87	69.65	62.70	57.01	58.87
	DSAN	85.83	76.97	81.73	81.51	37.43	46.03	50.66	62.58	49.17	39.16	56.00	51.74	68.86	61.26	53.20	55.04
	Ours	88.37	85.03	86.28	86.56	67.83	67.65	64.00	69.73	67.30	76.94	64.94	71.11	75.97	71.08	73.40	72.24
Joint method	+DANN	89.14	85.25	86.17	86.86	66.82	68.90	65.49	71.61	68.21	77.39	66.59	73.20	77.61	71.34	72.04	73.03
	+BNM	92.81	86.33	87.38	88.84	69.17	70.93	66.74	71.18	69.51	78.91	66.90	72.37	77.48	72.51	75.33	73.92
	+SCDA	89.95	85.19	86.01	87.05	67.90	67.93	64.32	69.84	67.50	77.13	65.28	71.25	76.28	71.25	73.72	72.49
	+DSAN	90.95	85.41	86.55	87.64	68.23	68.66	65.01	70.01	67.98	78.09	65.28	71.15	76.40	71.45	73.61	72.66

		Adaptiope [18]				DomainNet [17]						
		P→ R,S	R→ P,S	S→ P,R	Avg.	C,I,P→ Q,R,S	Q,R,S→ C,I,P	Avg.	C,I→ P,Q,R,S	P,Q→ C,I,R,S	R,S→ C,I,P,Q	Avg.
Domain Concat.	raw	54.30	55.55	43.28	51.05	8.43	11.97	10.20	6.40	9.22	10.76	8.79
	DANN [6]	58.75	56.13	45.02	53.30	7.47	11.99	9.73	5.35	7.81	10.57	7.91
	BNM [2]	54.50	55.07	45.72	51.76	8.14	12.15	10.15	6.34	9.17	10.70	8.74
	SCDA [13]	52.31	54.75	42.04	49.70	8.30	11.83	10.07	6.32	9.13	10.69	8.71
	DSAN [32]	53.65	55.13	43.74	50.84	7.04	11.21	9.13	4.64	8.45	10.50	7.86
	Ours-c	57.77	61.39	53.14	57.43	28.92	33.13	31.02	26.22	27.80	30.00	28.00
Multi.	Ours	60.09	64.59	57.68	60.79	42.23	44.30	43.26	39.10	38.99	36.67	38.25
Joint method	+DANN	64.40	71.18	62.12	65.90	41.02	43.74	42.38	38.21	38.73	36.09	37.67
	+BNM	62.24	65.89	59.63	62.59	42.18	44.14	43.16	39.52	39.18	36.76	38.49
	+SCDA	60.54	64.61	57.21	60.78	42.24	44.23	43.24	39.47	39.22	36.73	38.47
	+DSAN	61.38	65.23	58.42	61.68	42.23	44.25	43.24	39.10	39.01	36.72	38.28

Evaluation. We evaluate methods in an unsupervised adaptation framework. A classifier and backbone CNN are trained on source-domain samples equipped with annotation labels and then the optimized classifier is transferred to classify samples on target domains; during training, we are incapable of accessing labels of target-domain samples. Classification accuracies (%) are measured on the test-domain samples.

3.1 Performance analysis

We analyze the proposed method on Office-31 dataset in a simple scenario of single-source single-target domain adaptation for the ease of comparison. We apply the method in the framework of end-to-end learning as shown in Fig. 3. A backbone ResNet-50 is combined with projection head to produce 256-dimensional features which are followed by L_2 normalization; the ResNet-50 is pre-trained on ImageNet and is fine-tuned in the end-to-end learning for domain adaptation. We use batch sizes of 32 with the subspace rank of $r = 32$.

• *Transformation method.* Our method of $\hat{\boldsymbol{x}} = \boldsymbol{H}\boldsymbol{x}$ (10) is compared to the other types of transformation methods discussed in Sec. 2.4. While the methods of CORAL [21] and

auto-correlation [10] are straightforwardly applicable, we leverage Proposition 1 to embed GFK [7] in this deep architecture. The performance comparison in Table 1a shows that subspace-based transformation is superior to those containing second-order statistics of feature distribution, and the proposed method outperforms the others including GFK. In this deep adaptation, CORAL [21] significantly degrades performance as eigenvalues embedded in the transformation matrix would be less stably estimated in a mini-batch. Thus, DeepCORAL [20] get rid of the CORAL transformation and reformulate it as regularization of matching covariances to improve performance; in this case, the DeepCORAL produces 81.85% (Avg). On the other hand, subspaces are robustly estimated on a mini-batch to produce favorable performance. The proposed method requires less computation cost than the others which additionally apply eigen-decomposition (7) to compute $\boldsymbol{U}_+, \boldsymbol{G}^{\frac{1}{2}}$ and $\boldsymbol{H}^{\frac{1}{2}}$.

• *Extension method*. We then analyze the extension methods described in Sec. 2.3. Table 1b reports performance results in an ablation manner. The raw approach excludes transformation and any extension methods, exploiting only source samples for training. In the other approaches, target-domain samples participate in end-to-end learning through the subspace $\boldsymbol{P}_\mathcal{T} = \boldsymbol{U}_\mathcal{T}\boldsymbol{U}_\mathcal{T}^\top$ which is embedded in the DS matrix $\boldsymbol{H}$ (6) for transformation (10) and/or regularization (14). Table 1b demonstrates that both the transformation and the regularization contribute to matching subspaces for improving performance. The subspace similarities based on canonical angles for those approaches are also shown in Table 1b. Interestingly, the raw approach align two subspaces to some extent even by processing source samples only. Through the end-to-end training, feature representation is biased toward the object categories which are shared among source and target domains, thereby increasing the similarity of those subspaces. In accordance with Proposition 2, our domain-sum feature transformation improves the subspace matching; it produces higher similarity than the raw approach. While the regularization (14) enhances subspace matching in both the raw approach and our transformation approach, it produces better classification performance in our framework; 84.45% (raw+reg.) vs 85.56% (ours+reg.). Domain-sum transformation by $\boldsymbol{H}$ (6) naturally lets the target subspace $\boldsymbol{P}_\mathcal{T}$ join in a classification loss and thus facilitates to learning discriminative feature representation. Combination of the transformation and the regularization further improves performance as well as provides favorable matching of subspaces. The performance, however, is degraded by the larger regularization weight $\eta = 2$ than the standard setting of $\eta = 1$. Considering that our feature transformation intrinsically induces subspace matching as shown in Proposition 2, it is enough to set small weight of $\eta = 1$ to slightly inject the regularization effect into training. By adding the classifier subspace into the transformation matrix $\boldsymbol{H}$, performance is further improved. It also augments subspace matching as shown in Table 1b. Matching feature subspaces with a classifier subspace facilitates learning.

3.2 Multi-target deep domain adaption

The method is then compared to the other approaches on tasks of *multi-target* deep domain adaptation. It is noteworthy that the multi-target adaptation provides efficient inference in which the single model of classifier/feature representation learned on the source domain(s) is applicable across multiple domains without identifying domain of an input sample nor switching the model according to the target domain. We apply four types of regularization methods based on adversarial technique (DANN) [5], diversity of classifiers (BNM) [2], adversarial alignment of predictions (SCDA) [13] and local domain adaptation (DSAN) [32]. Our method is equipped with the extension techniques (Sec. 2.3, Table 1b). For fair com-

parison, we apply the same training protocol to the same backbone and projection head producing features (Fig. 3) which are subject to regularization and/or transformation for enhancing domain adaptation. Table 2 shows performance comparison where each column shows performance averaged over the multiple target domains.

As those methods other than ours are not designed to cope with multiple target domains, we concatenate multiple source/target domains in terms of mini-batch to mimic single-source single-target adaptation; the approach is denoted by "*Domain Concat*" in Table 2. For comparison, the brute-force approach to consider all pairs of source and target domains is applied, as denoted by "*Multi*" in Table 2, but there is less clear performance difference between those two approaches. While the proposed method ("Ours-c" in Table 2) shows superior performance to the others even in the concatenation scheme, our method works more effectively in the *multi*-domain approach. It exploits characteristics of multiple domains by means of respective domain subspaces and effectively aggregates them in the DS matrix (11).

Our method based on subspaces is compatible to the distribution-related regularization losses of the other adaptation methods; as discussed in Sec. 1, the subspace matching facilitates to align distributions which are contained in the subspaces. In our framework (Fig. 3), the regularization loss can be applied to raw feature representation $\boldsymbol{x}$ in a manner of domain concatenation for computational efficiency. The performance is further improved by the joint methods with ours as shown in Table 2.

4 Conclusion

We have proposed a simple yet effective feature transformation based on subspaces for reducing the discrepancy among diverse domains. The method is theoretically derived from GFK [7] and is formulated in an efficient form of domain-sum matrix $\boldsymbol{H}$. The feature transformation contributes to closing a gap between source and target subspaces in the framework of end-to-end learning. The proposed method is so simple as to naturally deal with multiple domains via aggregating domain subspaces in disregard of domain types, source or target, therefore addressing a general domain adaptation for multiple target domains. The experimental results on visual domain adaptation tasks demonstrate that the proposed method favorably improves performance in a deep multi-target framework.

References

[1] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, and Atsuto Maki. Factors of transferability for a generic convnet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9):1790–1802, 2016.

[2] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *CVPR*, pages 3941–3950, 2020.

[3] Kazuhiro Fukui and Atsuto Maki. Difference subspace and its generalization for subspace-based methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(11):2164–2177, 2015.

[4] Kazuhiro Fukui, Naoya Sogi, Takumi Kobayashi, Jing-Hao Xue, and Atsuto Maki. Discriminant feature extraction by generalized difference subspace. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1618–1635, 2023.

[5] 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):2096–2130, 2016.

[6] Behnam Gholami, Pritish Sahu, Ognjen Rudovic, Konstantinos Bousmalis, and Vladimir Pavlovic. Multiple subspace alignment improves domain adaptation. *IEEE Transactions on Image Processing*, 29:3993–4002, 2020.

[7] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pages 2066–2073, 2012.

[8] Raghuraman Gopalan, Ruonan Li, and Rama Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, pages 999–1006, 2011.

[9] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, pages 21271–21284, 2020.

[10] Xifeng Guo, Wei Chen, and Jianping Yin. A simple approach for unsupervised domain adaptation. In *ICPR*, pages 1566–1570, 2016.

[11] Elad Hoffer, Itay Hubara, and Daniel Soudry. Fix your classifier: The marginal value of training the last weight layer. In *ICLR*, pages 5822–5830, 2018.

[12] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, pages 1989–1998, 2018.

[13] Shuang Li, Mixue Xie, Fangrui Lv, Chi Harold Liu, Jian Liang, Chen Qin, and Wei Li. Semantic concentration for domain adaptation. In *ICCV*, pages 9102–9111, 2021.

[14] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105, 2015.

[15] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, pages 1647–1657, 2018.

[16] Le Thanh Nguyen-Meidine, Madhu Kiran, Jose Dolz, Eric Granger, Atif Bela, and Louis-Antoine Blais-Morin. Unsupervised multi-target domain adaptation through knowledge distillation. *arXiv,* 2007.07077, 2020.

[17] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, pages 1406–1415, 2019.

[18] Tobias Ringwald and Rainer Stiefelhagen. Adaptiope: A modern benchmark for unsupervised domain adaptation. In *WACV*, pages 101–110, 2021.

[19] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *ECCV*, pages 213–226, 2010.

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

[21] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, pages 2058–2065, 2016.

[22] Kowshik Thopalli, Rushil Anirudh, Jayaraman J. Thiagarajan, and Pavan Turaga. Multiple subspace alignment improves domain adaptation. In *ICASSP*, pages 3552–3556, 2019.

[23] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, pages 2962–2971, 2017.

[24] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, pages 5018–5027, 2017.

[25] Jindong Wang, Yiqiang Chen, Han Yu, Meiyu Huang, and Qiang Yang. Easy transfer learning by exploiting intra-domain structures. In *ICME*, pages 1210–1215, 2019.

[26] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018.

[27] Wei Wang, Zheng Dang, Yinlin Hu, Pascal Fua, and Mathieu Salzmann. Robust differentiable svd. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9): 5472–5487, 2021.

[28] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chena. Learning semantic representations for unsupervised domain adaptation. In *ICML*, pages 5423–5432, 2018.

[29] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NeurIPS*, pages 3320–3328, 2014.

[30] Huanhuan Yu, Menglei Hu, and Songcan Chen. Multi-target unsupervised domain adaptation without exactly shared categories. *arXiv,* 1809.00852, 2018.

[31] Werner Zellinger, Edwin Lughofer, Susanne Saminger-Platz, Thomas Grubinger, and Thomas Natschlager. Central moment discrepancy (cmd) for domain-invariant representation learning. In *ICLR*, 2017.

[32] 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, 2021.

[33] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2021.