Unsupervised Domain Adaptation via Discriminative Manifold Propagation

You-Wei Luo, Chuan-Xian Ren, Dao-Qing Dai, Hong Yan *Fellow, IEEE*

Abstract—Unsupervised domain adaptation is effective in leveraging rich information from a labeled source domain to an unlabeled target domain. Though deep learning and adversarial strategy made a significant breakthrough in the adaptability of features, there are two issues to be further studied. First, hard-assigned pseudo labels on the target domain are arbitrary and error-prone, and direct application of them may destroy the intrinsic data structure. Second, batch-wise training of deep learning limits the characterization of the global structure. In this paper, a Riemannian manifold learning framework is proposed to achieve transferability and discriminability simultaneously. For the first issue, this framework establishes a probabilistic discriminant criterion on the target domain via soft labels. Based on pre-built prototypes, this criterion is extended to a global approximation scheme for the second issue. Manifold metric alignment is adopted to be compatible with the embedding space. The theoretical error bounds of different alignment metrics are derived for constructive guidance. The proposed method can be used to tackle a series of variants of domain adaptation problems, including both vanilla and partial settings. Extensive experiments have been conducted to investigate the method and a comparative study shows the superiority of the discriminative manifold learning framework.

Index Terms—Unsupervised Domain Adaptation, Riemannian Manifold, Discriminant Embedding, Manifold Alignment.

1 Introduction

In machine learning, the amount of labeled data plays a crucial role during the learning process. Convolutional Neural Networks (CNNs) can achieve a significant advance in various tasks via a large number of well-labeled samples. Unfortunately, such data are often prohibitively expensive to obtain in many real-world scenarios. Applying a learned model in a new environment, i.e., the cross-domain scheme, may cause a significant degradation of recognition performance [1], [2].

Unsupervised Domain Adaptation (UDA) is designed to deal with the shortage of labels by leveraging rich labels and strong supervision from the source domain to the target domain where there is no access to the annotations [3]. Datasets composed specifically of exploratory factors and variants, such as background, style, illumination, camera views or resolution, often lead to shifting distributions (the domain shift) [4]. Classical UDA assumes the label spaces of two domains are equivalent, i.e., the vanilla UDA setting in Figure 1(a). According to the transfer theory established by Ben-David et al. [5], the primary task for cross-domain adaptation is to learn the discriminative representations while narrowing the discrepancy between domains. As this strong condition in the vanilla setting is an obstacle to some special real-world applications, partial UDA is explored as a subproblem of UDA, i.e., Figure 1(b). Partial UDA relaxes the equivalence assumption on the label space by taking the label space of the target domain as a subset of the source domain [6], [7]. Under the partial setting, the source domain is still a large-scale annotated dataset (e.g., ImageNet [8]) while the target domain can be any smaller dataset with fewer categories (e.g., Caltech-256 [9] and PASCAL VOC [10]). Unfortunately, as generalizations of vanilla methods, partial UDA methods usually also suffer from the negative transfer problem.

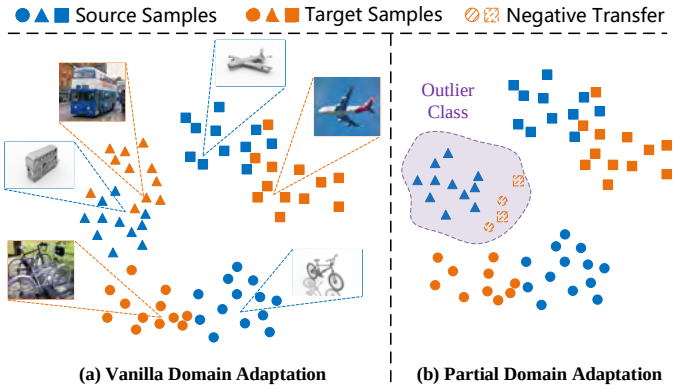


Fig. 1. Illustration of the vanilla and partial domain adaptation problems. (a) Vanilla domain adaptation assumes that the label spaces of the source and target domains are equivalent, and the *domain shift* problem is the major difficulty. (b) Partial domain adaptation assumes that the label space of the target domain is a subset of the source domain. Performance can be further degraded by the *negative transfer* problem, which means that target samples are aligned to outlier classes (e.g., the "triangle"). Best viewed in color.

Early domain adaptation approaches with hand-crafted features focussed on learning domain-invariant information. Statistical methods attempt to align the domains in embedding space based on moment statistics, such as the mean and covariance [4], [11], and then minimize the distribution discrepancy in that space. Manifold and subspace learning methods also achieve significant successes in transfer learning tasks [12], [13]. In [14], the authors build a geodesic flow on the manifold to model the domain shift and form an optimal distance measurement between samples from different domains. Moon et al. [13] explore the global information in an online UDA by proposing a mean target space

Y.W. Luo, C.X. Ren, and D.Q. Dai are with the Intelligent Data Center, School of Mathematics, Sun Yat-Sen University, Guangzhou, 510275, China. H. Yan is with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong. Luo and Ren contributed equally to this work.
This work is supported in part by the National Natural Science Foundation of China under Grants 61976229, 61906046, 61572536, 11631015, U1611265, in part by the Science and Technology Program of Guangzhou under Grant 201804010248 and City University of Hong Kong (Project 9610460).

which considers the coherency among all target-data batches. Dictionary learning approaches merge the common features of different domains into a shared dictionary and find sparse or low-rank codings for input samples [15].

CNNs can learn abstract representations with nonlinear transformation [16], which suppress the negative effects caused by variable explanatory factors in domain shift [2]. Early work [2], [17] attempted to transfer the source classifier, with sufficient supervision, to the target domain by minimizing domain discrepancy, supposing that a well distributed alignment leads to an effective application of a trained classifier on the target domain. Adversarial confusion methods [18], [19], which are inspired by Generative Adversarial Nets (GANs) [20], produce generated features that are domain-indistinguishable and form a well-aligned marginal distribution. Adversarial-based weighting methods use the discriminator of GANs to evaluate the probability of negative transfer in a partial UDA from the instance-level [6], [21], class-level [7], or both [22]. However, the conditional distributions of those methods are not guaranteed [23], [24]. Several methods achieve improvements in accuracy by employing uncertainty information, e.g., pseudo labels and soft labels, to enhance the discrimination in the target domain [23], [25], [26]. Other methods revisit the trade-off between transferability and discrimination, and then build a more discriminative [27] or transferable [28] model.

In this paper, we propose a novel framework called Discriminative Manifold Propagation (**DMP**) to deal with both vanilla and partial UDA problems. DMP primarily considers two issues in existing UDA methods. First, direct utilization of uncertain information is error-prone and should be treated cautiously [23], as hard-assigned labels may change the intrinsic data structure [29]. Second, batch-wise training in deep learning limits the capture of global information, thus models may be misled by extreme local distributions. As the ability to transfer and discriminate are both valuable [27], the method proposed here develops a unified rule for these two properties. The main idea is to describe the domains by a sequence of latent manifolds. In contrast to earlier works, which build discriminative models directly on the source domain [30] or both domains [31], we establish a more relaxed criterion and enhance target discriminability transductively. We extend this criterion to a global approximation scheme, which overcomes problems caused by batch-wise training. Inspired by prior work on manifold learning [14], [32], we employ the manifold metrics to measure the domain discrepancy.

This paper extends previous work [33] by: 1) extending the manifold learning framework by including the affine Grassmann distance and the Log-Euclidean metric; 2) deriving a new error bound for the affine Grassmann distance; 3) studying another variant of UDA (the partial UDA problem) and extending a theoretical error bound to this case; 4) extending a unified algorithm to simultaneously deal with the domain shift and negative transfer problems in different UDA settings; 5) conducting experiments, including parameter selection and ablation analysis, to validate the effectiveness of DMP in both the vanilla and partial settings. Our contributions are summarized as follows.

- To explore the discriminative structure of the target domain and reduce uncertainty information, a probabilistic manifold embedding criterion is proposed. This criterion constructs an intra-class separable structure on the source domain. Target discriminability is achieved by a probabilistic and truncated intra-class compactness constraint and the inter-class separability is transduced from the source domain. A global structure learning scheme is extended based on the pre-built prototypes.
- A manifold alignment framework that is consistent with the manifold assumption on the embedding space is proposed. It establishes a series of abstract descriptors (i.e. the basis) for the original data based on different manifolds, and aligns the domains by minimizing the discrepancy between the abstract descriptors. The theoretical error bounds are derived to facilitate the selection of components.
- Both the vanilla and partial UDA problems can be tackled effectively by the proposed method. It extends the discriminant criterion and manifold alignment to a weighting scheme, which alleviates negative transfer from the outlier classes. A theoretical error bound is derived under the partial setting, which gives a theoretical interpretation of the proposed weighting strategy.

2 Related Work

In this section, we review the two subproblems of UDA. The primary goal of UDA is to learn a classifier that generalizes well on the target domain. Since there are no annotations on the target domain, the key is to learn the abstract representations, which transduce discriminative information about the source to the target domains.

2.1 Vanilla UDA Methods

The label space of the source domain $\mathcal{X}^s$ and target domain $\mathcal{X}^t$ is denoted as $\mathcal{C}^s$ and $\mathcal{C}^t$, respectively. Under the vanilla UDA setting, the label spaces completely overlap, i.e., $\mathcal{C}^s = \mathcal{C}^t$. A simple yet effective solution is to mitigate the domain shift or bias.

Approaches with hand-crafted features usually focus on the learning of domain-invariant or discriminative features [4], [11]. Based on the manifold assumption, Gopalan et al. [14] take the source and target domains as points on the Grassmann manifold, and propose to generate multiple subspaces from those points. Then the distance is measured by the geodesic flow between those points. Shekhar et al. [15] construct a shared dictionary for both source and target domains, while minimizing the reconstructed error based on the learned dictionary. Ren et al. [12] explore an optimal experimental design based on the covariance of structured feature translators to tackle the nonlinear and heterogeneity problems. Das et al. [26] propose to match the vertexes and edges of the domains, and refine the pseudo labels by keeping the unlabelled samples away from the decision boundaries.

Deep learning methods enhance transferability by exploring the abstract representations that disentangle the exploratory factors of variants hidden in the data [16]. Distribution alignment methods directly minimize the discrepancy between domains based on moment statistics directly. For the first-order statistic, maximum mean discrepancy (MMD) is a popular and effective metric. Deep Adaptation Network (DAN) [2] exploits the multiple kernel variant of MMD to maximize test power and reduce the probability of Type II error jointly. Ren et al. [34] improved conditional MMD (CMMD) under the auto-encoder framework. For second-order moment statistics, Deep CORAL is a simple, yet effective, method that measures the distance between domains by their corresponding covariances [17]. Chen et al. [30] add instances- and centers-based loss functions to the Deep CORAL model

to enhance the discriminability of the source domain. Inspired by GANs, lots of adversarial approaches with different purposes were developed [18], [35]. Joint Adaptation Network (JAN) [36] proposes a joint MMD distance and adopts adversarial training to make the domains more distinguishable. Based on the domain shared generator, Unsupervised Domain Adaptation with Similarity Learning (SimNet) [25] develops a classifier composed of the prototypes from different classes, then the target samples are classified by the most similar prototype.

Domain-specific and Task-specific methods aim to tackle the problem of compact representations in high-level layers. Chang et al. [37] assign different domains using distinct batch-normalization layers and shared feature extraction layers in the first stage, and then train the target classifier by gradually modifying pseudo labels. Ding et al. [29] propose an end-to-end low-rank coding method via domain-specific dictionaries. Maximum Classifier Discrepancy (MCD) [38] minimizes the maximum discrepancy of classifiers adversarially to form a tight classification boundary. Recent research suggests that discriminability plays a crucial role in the formation of class distributions [23], [24], [27]. Conditional Domain Adversarial Network (CDAN) [23] encodes target predictions into deep features, then models the joint distributions of features and labels. Batch Spectral Penalization (BSP) [27] builds a spectral penalty to enhance transferability while keeping the main discriminability.

2.2 Partial UDA Methods

The most typical characterization of the partial UDA setting is $\mathcal{C}^t \subset \mathcal{C}^s$, thus those two domains partially overlap. The outlier classes refer to the categories that are not shared by the two domains, i.e., $\mathcal{C}^s/\mathcal{C}^t$. If target samples are aligned to the outlier categories, they are likely to be misclassified.

Weight-based methods assign lower weights to the less transferable samples or classes (i.e., the outlier samples or classes), which leads to a declining influence of negative transfer. Partial Adversarial Domain Adaptation (PADA) [7] is a class-level weighting method that takes the target predictions as outlier class measurements. As the probability values of the outlier classes are supposed to be significantly smaller than the shared classes, the source classes are weighted by the mean value of target predictions to reduce the misaligned samples. Importance Weighted Adversarial Nets (IWAN) [6] introduces an auxiliary domain classifier to estimate the probability that the source sample belongs to the outlier classes. The sample with larger probability will be assigned a smaller weight during domain confusion. Example Transfer Network (ETN) [21] extends the idea of IWAN by introducing an auxiliary label predictor with leaky-softmax activation. Selective Adversarial Networks (SAN) [22] proposes class-specific domain discriminators to circumvent negative transfer. Adversarial learning is guided by both instance-level and class-level weights from the target predictions.

Some recent methods revisit the UDA problem from the perspective of feature transferability [27], [28]. They investigate the discriminability or transferability based on the singular values or norms of learned features. Adaptive Feature Norm (AFN) [28] builds a unified computation for the vanilla and partial UDA problems by progressively matching the feature norms of two domains to a large value. It suggests that the larger norm regions are more suitable for safe transfers.

3 Multilayer Riemannian Manifold Embedding and Alignment

In this section, we propose the DMP framework. The motivation is presented in Section 3.1. An overview of DMP and the network architecture are shown in Section 3.2. In Section 3.3, we propose the global discriminant criterion for manifold embedding. The manifold metric alignment is presented in Section 3.4.

3.1 Motivations

The Riemannian manifold $\mathcal{M}$ usually consists of objects such as a linear subspace, an affine/convex hull, and a symmetric positive definite (SPD) matrix [32]. From the perspective of discriminative embedding, graph-based criteria [39] are widely adopted in manifold learning and domain adaptation. These methods establish an instances-based connection graph or similarity graph to construct a separable space [15], [25]. One of the most common assumptions in UDA based on a statistical distribution, the alignment based on covariance matrices, that lie on the Riemannian manifold, equips the domain with the manifold and statistical properties. Motivated by previous attempts [32], [40], our work aims to embed a graph-based discriminant criterion to the target domain, and align the source and target domain based on the manifold assumption.

Given a feature matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$ and its mean vector $\bar{\mathbf{x}} \in \mathbb{R}^d$, where d denotes the dimension of features and n the sample size. Denote by S the input space (e.g., Euclidean space and Hilbert space), manifold learning aims to learn a nonlinear mapping

$$f:\ S \rightarrow \mathcal{M},$$

where $\mathcal{M}$ is the low-dimensional embedding manifold. Based on the SPD representation setting, the image of a given covariance matrix $\mathbf{C}(\mathbf{X}) = \frac{1}{n-1}(\mathbf{X} - \bar{\mathbf{x}}\mathbf{1}_n^T)(\mathbf{X} - \bar{\mathbf{x}}\mathbf{1}_n^T)^T \in \mathbb{R}^{d\times d}$ is a lower-order SPD matrix $\mathbf{C}' = f(\mathbf{C}) \in \mathbb{R}^{d'\times d'}$, where $\mathbf{1}_n$ is an n-dimensional vector with all elements equal to 1 and $(\cdot)^T$ is the transpose operation. $\mathbf{C}'$ can be decomposed as the inner product of a lower-order matrix $\mathbf{X}'$, i.e., $\mathbf{C}' = \mathbf{X}'\mathbf{X}'^T$. Learning of the projector f can be deduced to find a nonlinear transformation

$$g:\ \mathbf{X} \mapsto g(\mathbf{X}),$$

where $g(\mathbf{X})$ is the approximation of $\mathbf{X}'$. That is, the image of mapping f can be approximated by the inner product of $g(\mathbf{X})$, i.e., $f(\mathbf{C}) \approx g(\mathbf{X})g(\mathbf{X})^T$.

For domain adaptation, the original source and target domains can be taken as two Euclidean spaces, where the discriminant information is relatively inadequate. Thus the latent manifolds, i.e., $\mathcal{M}^s$ of the source and $\mathcal{M}^t$ target, should be compact, representative and discriminative. Compared with the images, the manifold embedding features are supposed to contain more task-specific information (i.e., class information) and be domain indistinguishable. In DMP, we extract the task-specific information from the images by equipping the discriminant criterion with the predictive information, while removing most of the domain information (e.g., styles and views) by aligning the domains based on the manifold metrics.

3.2 Low-Dimensional Manifold Layers

We now focus on the nonlinear transformation g for the input features $\mathbf{X}$. In general, CNNs are used to obtain the projection g. Figure 2 shows the detailed network architecture of the proposed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.3014218, IEEE Transactions on Pattern Analysis and Machine Intelligence

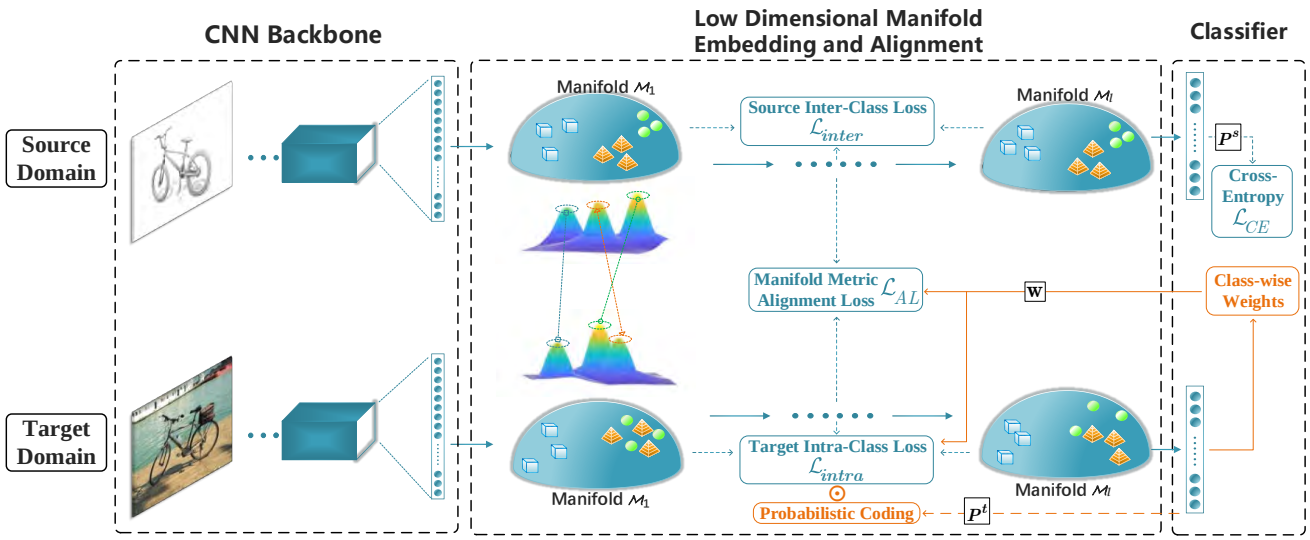


Fig. 2. Overview of the proposed multilayer Riemannian manifolds embedding and alignment network. A CNN-based feature extractor is adopted to learn the common representations of both domains. The discriminative information are transferred via the Riemannian manifold layers, where fully connected layers are equipped with the proposed proposed soft discriminant criterion and manifold metric domain alignment. Best viewed in color.

method. Let $\mathbf{\Theta}$ be the parameters of the networks. To explore the latent Riemannian representations of the raw features, the output features of the CNN backbone are sent into progressive low-dimensional manifold layers, as shown in Figure 2. Since there are natural geometric differences between the raw and embedding spaces, a multilayer scheme is adopted to reduce the dimension of features progressively.

The Riemannian manifold layers $\{\mathcal{M}_l | l = 1, 2, \ldots, L\}$ are achieved via fully connected layers. The CNNs and Riemannian manifold layers are shared by both domains so that the common projections can be exploited to map the two domains to a shared low-dimensional space. Therefore, any manifold layer $\mathcal{M}_i$ should have the following properties.

- Discriminative Structure: The intra-class samples of the target domain are compact, while the inter-class samples are separable.
- Consistent Structure: The source and target domains are aligned with the manifold metrics. Then the domain discrepancy can be represented as the distance between two submanifolds on $\mathcal{M}_i$, and minimized based on the defined manifold metrics (e.g., Grassmann distance, affine Grassmann distance, Log-Euclidean metric or manifold principal angle similarity).

We model the properties by loss terms $\mathcal{L}_{DS}$ and $\mathcal{L}_{AL}$, which will be mathematically formulated later. To satisfy the first property, two similarity-based criteria are explored, i.e., the source discriminant inter-class loss and the target discriminant intra-class loss in Figure 2. Specifically, the target intra-class loss is used to enlarge the similarities between the target samples and their corresponding source class-wise centers, while the source inter-class loss is used to find a balanced geometric structure of the source class-wise mean vectors. The discriminative structure loss is denoted by

$$\mathcal{L}_{DS} = \sum_l (\mathcal{L}^l_{inter} + \mathcal{L}^l_{intra}),$$

where $\mathcal{L}^l_{inter}$ and $\mathcal{L}^l_{inter}$ are the similarity-based loss functions of the l-th Riemannian manifold layer $\mathcal{M}_l$.

The second property is satisfied by the manifold metric alignment loss in Figure 2. The overall alignment loss can be written as

$$\mathcal{L}_{AL} = \sum_l \mathcal{L}^l_{align},$$

where $\mathcal{L}^l_{align}$ is the alignment loss of $\mathcal{M}_l$.

Finally, the proposed objective function is

$$\min_{\mathbf{\Theta}} \mathcal{L} = \mathcal{L}_{CE} + \lambda_1 \mathcal{L}_{DS} + \lambda_2 \mathcal{L}_{AL}, \quad (1)$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss of the classifier on the source domain and $\{\lambda_1, \lambda_2\}$ are penalty parameters. The introduction and derivation of these loss terms are presented in the following sections.

3.3 Discriminative Structure Learning

In this section, we describe how to embed the discriminative structure into the manifold layers. The main idea is shown in Figure 3. Since there exists a distribution discrepancy between different domains (e.g., Figure 3(b)), models that build a discriminant criterion only on the source domain are less discriminative on the target domain. The discriminant learning process will be error-prone if only the target uncertain information is used. If discriminability is required for both domains, the ability of model to generalize may be decreased. To relax the constraint, we propose to focus on the inter-class separability of the source domain and the soft intra-class compactness of the target domain. This uses prototypes (e.g., center vectors), which have been shown to be effective and robust to the domain shift problem [25], [31], as an intermediary during discriminant learning. As illustrated in Figure 3(d), the proposed method achieves the discriminative property on the target domain transductively.

Without loss of generality, we only formulate the loss terms in the l-th Riemannian manifold layer $\mathcal{M}_l$. Let $\mathbf{H}^s_l = [\mathbf{h}^s_{l,1}, \mathbf{h}^s_{l,2}, \ldots, \mathbf{h}^s_{l,n^s}] \in \mathbb{R}^{d_l \times n^s}$ and $\mathbf{H}^t_l \in \mathbb{R}^{d_l \times n^t}$ be the feature matrices of $\mathcal{M}_l$. Since class centers of the source domain are used in both loss terms, the source mean vector $\bar{\mathbf{h}}^s_l \in \mathbb{R}^{d_l}$ and source class-wise mean matrix $\bar{\mathbf{H}}^s_l \in \mathbb{R}^{d_l \times c}$ are computed, where $c = |\mathcal{C}^s|$ is the number of source categories. Let

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.3014218, IEEE Transactions on Pattern Analysis and Machine Intelligence

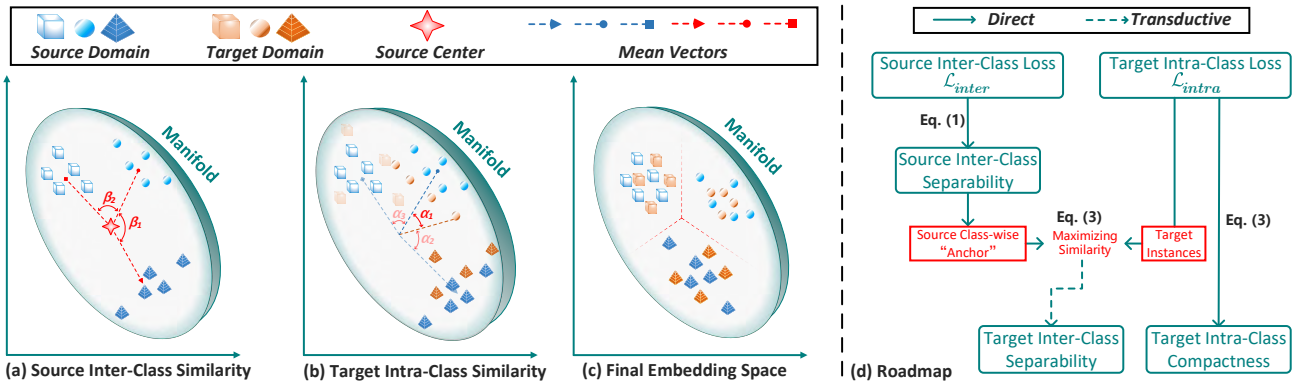


Fig. 3. Illustration of the discriminative structure learning framework. (a) The source inter-class similarity forms a separable space for the source samples. (b) The target intra-class similarity constructs a compact space for the samples from the same category. (c) The final embedding space, where the target domain is discriminative. (d) The roadmap for achieving the discriminative property on the target domain. The target intra-class compactness is implemented by $\mathcal{L}_{intra}$ directly and the target inter-class separability is reached by $\mathcal{L}_{inter}$ transductively. Best viewed in color.

$\mathbf{P}^t = [\mathbf{p}_1^t, \mathbf{p}_2^t, \ldots, \mathbf{p}_{n^t}^t] \in \mathbb{R}^{c \times n^t}$ and $\mathbf{P}^s \in \mathbb{R}^{c \times n^s}$ be the softmax predictions of the target and source samples from the classifier layer, respectively, and $y_i \in \{1, 2, \ldots, c\}$ be the ground-truth label of the i-th source sample. The lower case letter with subscript ij (e.g., a_{ij}) represents the (i, j) entry of its corresponding matrix (e.g., $\mathbf{A}$).

3.3.1 Source Inter-Class Similarity

To learn a separable geometric structure of the class distribution, the similarity measurement is adopted here, which is also shown in Figure 3(a). Rather than computing the similarities between the class-wise centers and the source center directly, we consider the class-wise centers as follows

$$\hat{\mathbf{H}}_l^s \triangleq \bar{\mathbf{H}}_l^s - \bar{\mathbf{h}}_l^s \mathbf{1}_c^T. \tag{2}$$

We call $\hat{\mathbf{H}}_l^s \in \mathbb{R}^{d_l \times c}$ the centralized class means hereafter. If the columns of $\hat{\mathbf{H}}_l^s$ are normalized with the ℓ_2 norm, the between-class cosine matrix is derived as

$$\mathbf{S}_l^b = \hat{\mathbf{H}}_l^{s^T} \hat{\mathbf{H}}_l^s,$$

where $s_{l,ij}^b = \hat{\mathbf{h}}_{l,i}^{s^T} \hat{\mathbf{h}}_{l,j}^s$ indicates the similarity between i-th class and j-th class. Then the separable structure is reached by maximizing the dissimilarities between the centralized class mean vectors. Equivalently, it can be achieved by minimizing the following inter-class loss:

$$\mathcal{L}_{inter}^l(\mathbf{H}_l^s) = \frac{2}{c(c-1)} \sum_{i<j} s_{l,ij}^b. \tag{3}$$

Let us take Figure 3(a) as an example. There is a 2-dimensional space with 3 classes. Let {1,2,3} be the labels of "Ball", "Pyramid" and "Cube", respectively. Under this situation, $s_{l,12}^b$ and $s_{l,13}^b$ are depicted as $\cos(\beta_1)$ and $\cos(\beta_2)$, respectively. According to the goal of Eq. (3) and ignoring the constraints, the optimal solution occurs at $\beta_1 = \beta_2 = \frac{2}{3}\pi$, and the minimal $\mathcal{L}_{inter}^l$ equals to $-\frac{1}{2}$ (which can also be seen as the lower bound of constrained scenarios).

3.3.2 Target Intra-Class Similarity

Since there are no labels on the target domain, discriminant learning is facilitated by the soft labels $\mathbf{P}^t$ (i.e., the output of the softmax layer). Since $\mathbf{P}^t$ can be regarded as the confidence or probability of classification, the predictions are used to weight the importance, or confidence, of the supervised information provided by the soft labels. Similarly, assuming the columns of $\bar{\mathbf{H}}_l^s$ and $\mathbf{H}_l^t$ have unit length, the similarities under all classification cases can be written as

$$\mathbf{S}_l^w = \bar{\mathbf{H}}_l^{s^T} \mathbf{H}_l^t. \tag{4}$$

Note that the centers of the source classes are used instead of those of the target. The main reasons are that the inter-class structure learned from the source domain can be transduced to the target domain and that the source class centers computed from ground-truth labels are more reliable. Because there is so much uncertainty when pseudo labels are used straightforwardly on the target domain, we establish a probabilistic discriminative criterion to make use of most of the information provided by the soft labels. Intuitively, $\mathbf{P}^t$ is a naturally choice for the probabilistical weighting model. Then the probabilistic intra-class loss is formulated as

$$\mathcal{L}_{intra}^l(\mathbf{H}_l^t, \mathbf{P}^t) = -\frac{1}{n^t c} \sum_{i=1}^{c} \sum_{j=1}^{n^t} p_{ij}^t s_{l,ij}^w. \tag{5}$$

However, there is much noise in $\mathbf{P}^t$, and its values are very small. Empirically, $\mathbf{p}_i^t$ tends to be the a one-hot vectors if the softmax classifier is convergent. As truncation is an efficient way for denoising, we develop a Top-k preserving scheme for the truncated intra-class loss. Let $V_j = \{(i, j) | i = v_{1j}, v_{2j}, \ldots, v_{kj}\}$ be the index set of the k-largest elements in $\mathbf{p}_j^t$, $j = 1, 2, \ldots, n^t$. Then an indicator matrix is defined as

$$\chi_{ij} = \begin{cases} 1, & (i, j) \in V_j \\ 0, & (i, j) \notin V_j \end{cases}.$$

Finally, the intra-class loss is modified by the truncated matrix χ and written as

$$\mathcal{L}_{intra}^l(\mathbf{H}_l^t, \mathbf{P}^t) = -\frac{1}{n^t k} \sum_{i=1}^{c} \sum_{j=1}^{n^t} \chi_{ij} p_{ij}^t s_{l,ij}^w. \tag{6}$$

In conclusion, the two proposed loss terms build a probabilistic discriminant criterion on the target domain. The ground-truth labels on the source domain provide a reliable separable structure directly, where the intra-class structure is not required. The target samples are attached to the corresponding source class-wise centers via soft labels. As shown in Figure 3(c) and 3(d), the intra-class relationship on the source domain does not change much while the discriminative property of the target domain is

satisfied. The motivation behind using the cosine function is that the Euclidean distance may fail to learn a separate inter-class structure when some class centers nearly overlap, e.g., the case that $\beta_2 \approx 0$ in Figure 3(a). However, the cosine-based metric learning measures the similarity on the unit hypersphere. It can achieve the discriminative feature structure from geometric aspect, i.e., large inter-class angles and small intra-class angles.

3.3.3 Global Structure Learning

For the batch-wise training manner in deep learning models, training batch sizes of the source and target domains are set as b_s, i.e., $n^s = n^t = b_s$. The complete relation graph between the instances is time and memory consuming to obtain in deep networks. Moreover, the mean statistics $\bar{\mathbf{h}}_l^s$ and $\bar{\mathbf{H}}_l^s$ computed from the batch data are unable to reflect the complete categorical information, because the class number in the batch is often smaller than c. The direct application of classical graph embedding may be misled by some extreme local distributions, which will lead to a suboptimal solution.

Supposing that the geometry of the manifold does not change drastically after several updates, we build two *anchors* in the whole data space to acquire the global information. We propose to fix the *anchors* in each batch iteration and update them after every epoch or several iterations. Specifically, the *anchors*, i.e., $\bar{\mathbf{h}}_l^s$ in Eq. (2) and $\bar{\mathbf{H}}_l^s$ in Eq. (4), are dynamically updated. Note that the *anchors* are treated as constants in optimization. The variables, i.e., $\bar{\mathbf{H}}_l^s$ in Eq. (2) and $\mathbf{H}_l^t$ in Eq. (4), are obtained from batch data. If there are no samples of the i-th class in the current batch, then the corresponding class center $\bar{\mathbf{h}}_{l,i}^s$ is an all-zero vector. The inter-class loss is strongly supervised by source labels at the beginning, while the intra-class loss, facilitated by soft labels, is used after a certain number of iterations/epochs.

3.4 Manifold Metric Alignment

A manifold metric alignment method is proposed to satisfy domain consistency. The second-order moment statistic is an important tool to represent a manifold $\mathcal{M}$. Therefore, the alignment based on covariance not only meets the requirement of the manifold assumption, but also possesses useful statistical properties, such as the distribution assumption.

Let $\mathbf{C}_l^s$ and $\mathbf{C}_l^t$ be the covariance matrices of the source and target domains computed from batch-wise features, respectively. Assume $\mathcal{M}_l^s$ and $\mathcal{M}_l^t$ are two submanifolds of $\mathcal{M}_l$, which are represented by their corresponding covariance matrices. Before the alignment process, these two submanifolds may partially overlap, and our goal is to minimize the discrepancy under the metric, $\mathcal{M}_l$. In general, the manifold metric alignment loss is expressed as

$$\mathcal{L}_{align}^l(\mathbf{H}_l^s, \mathbf{H}_l^t) \triangleq dist(\mathcal{M}_l^s, \mathcal{M}_l^t) = d_{\mathcal{M}}(\mathbf{C}_l^s, \mathbf{C}_l^t),$$

where $d_{\mathcal{M}}(\cdot,\cdot)$ is the manifold metric to be defined.

3.4.1 Grassmann Manifold

The Grassmann manifold [41] is a well-known type of Riemannian manifold. It is a projected subspace $\mathbb{R}^{d_l'}$ deduced from the originally high-dimensional space $\mathbb{R}^{d_l}$, $d_l' < d_l$. Thus, two submanifolds $\mathcal{M}_l^s$ and $\mathcal{M}_l^t$ lying on the Grassmann manifold $\mathcal{M}_l$ are represented as two individual points. The distance between these two points is measured by the discrepancy between the their projection orthogonal bases $\mathbf{U}_l^s$ and $\mathbf{U}_l^t$ which can be obtained from the Singular Value Decomposition (SVD) of the covariance matrices $\mathbf{C}_l^s$ and $\mathbf{C}_l^t$, respectively, i.e.,

$$d_{\mathcal{M}}^G(\mathbf{C}_l^s, \mathbf{C}_l^t) = \frac{1}{d_l^2} \| \mathbf{U}_l^s \mathbf{U}_l^{s^T} - \mathbf{U}_l^t \mathbf{U}_l^{t^T} \|_F^2, \tag{7}$$

where $\| \cdot \|_F$ is the Frobenius norm.

As the dimension d_l' is required in the Grassmann distance, we establish a theoretical error bound for the selection of d_l'. Denoting the covariance of a given distribution D by $\mathbf{C}$, and the covariance drawn i.i.d. from D with sample size n by $\tilde{\mathbf{C}}$. Then, Zwald et al. [42] give the following theorem.

Theorem 1. **[42, Theorem 4]** *Supposing that* $\sup_{\mathbf{x}\in\mathcal{X}} \|\mathbf{x}\| \leq M$, *where* $\mathcal{X}$ *is the measurable space where variable* $\mathbf{x}$ *take its value. Let* $\mathbf{U}_{\mathbf{C}}^{d'}$ *and* $\mathbf{U}_{\tilde{\mathbf{C}}}^{d'}$ *be the orthogonal projectors of the subspaces spanned by the first* d' *eigenvectors of* $\mathbf{C}$ *and* $\tilde{\mathbf{C}}$, *respectively. Let* $\lambda_1 > \lambda_2 > \cdots > \lambda_{d'} > \lambda_{d'+1} \geq 0$ *be the first* $d'+1$ *eigenvalues of* $\mathbf{C}$, *then for any* $n \geq \left(\frac{4M}{\lambda_{d'}-\lambda_{d'+1}}\left(1+\sqrt{\frac{\ln(1/\delta)}{2}}\right)\right)^2$ *with probability at least* $1-\delta$ *we have:*

$$\|\mathbf{U}_{\mathbf{C}}^{d'} - \mathbf{U}_{\tilde{\mathbf{C}}}^{d'}\| \leq \frac{4M}{\sqrt{n}\,(\lambda_{d'}-\lambda_{d'+1})}\left(1+\sqrt{\frac{\ln(1/\delta)}{2}}\right). \tag{8}$$

This theorem shows the relation between the error and d'. Denote the right side of Eq. (8) as $\frac{E(\delta)}{\lambda_{d'}-\lambda_{d'+1}}$. To extend the inequality to the Grassmann distance, we derive following lemma.

Lemma 2. *Based on the condition in Theorem 1, we have*

$$\|\mathbf{U}_{\mathbf{C}}^{d'}\mathbf{U}_{\mathbf{C}}^{d'^T} - \mathbf{U}_{\tilde{\mathbf{C}}}^{d'}\mathbf{U}_{\tilde{\mathbf{C}}}^{d'^T}\|_F \leq 2\sqrt{2}E(\delta)\frac{\sqrt{d'}}{\lambda_{d'}-\lambda_{d'+1}}$$

with probability at least $1-\delta$.

Based on Lemma 2, the following theorem gives a error bound of $d_{\mathcal{M}}(\mathbf{C}^s, \mathbf{C}^t)$ w.r.t. its n sample approximation $d_{\tilde{\mathcal{M}}}(\tilde{\mathbf{C}}^s, \tilde{\mathbf{C}}^t)$.

Theorem 3. *Assuming the condition in Theorem 1 is specified by domains. Specifically,* λ_i^s *and* λ_i^t *denote the* i*-th largest eigenvalue of the domain-specific covariance matrices* $\mathbf{C}_s$ *and* $\mathbf{C}_t$, *respectively. Denote the error index by:*

$$e^G(d') = \frac{\sqrt{d'}}{\lambda_{d'}^s - \lambda_{d'+1}^s} + \frac{\sqrt{d'}}{\lambda_{d'}^t - \lambda_{d'+1}^t}.$$

Then the following error bound holds with probability at least $1-\delta$*:*

$$|d_{\mathcal{M}}^G(\mathbf{C}^s, \mathbf{C}^t) - d_{\tilde{\mathcal{M}}}^G(\tilde{\mathbf{C}}^s, \tilde{\mathbf{C}}^t)| \leq 2\sqrt{2}E(\delta)e^G(d').$$

Theorem 3 suggests that the upper bound of the error is proportional to $e(d')$. It means that we should search for the maximal gap between the continuous eigenvalues with consideration of the inflation factor $\sqrt{d'}$. As, in a batch learning setting, the batch size b_s is usually smaller than d, thus d' only needs to be searched in $\{1, 2, \ldots, b_s - 1\}$. The proofs are provided in Section S.1 and Section S.2 of the supplementary material, respectively.

3.4.2 Affine Grassmann Manifold

The affine Grassmann manifold is a smooth manifold that consists of all d'-dimensional affine subspaces in $\mathbb{R}^d$, thus it is also called the Grassmannian of the affine subspaces [43]. For any matrix, its representation on affine Grassmann manifold is the affine combination of orthonormal d'-frames $\mathbf{U}$ with displacement $\boldsymbol{\mu}$, where

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.3014218, IEEE Transactions on Pattern Analysis and Machine Intelligence

$\boldsymbol{\mu}$ is the mean of the matrix. In [32], Huang et al. extended the similarity function on the affine Grassmann manifold to a distance metric. The distance between the corresponding representations of $\mathbf{C}_l^s$ and $\mathbf{C}_l^t$ on the affine Grassmann manifold is defined as

$$d_{\mathcal{M}}^{AG}(\mathbf{C}_l^s,\mathbf{C}_l^t) = \frac{1}{d_l^2}\Big(\|\mathbf{U}_l^s\mathbf{U}_l^{s^T} - \mathbf{U}_l^t\mathbf{U}_l^{t^T}\|_F + \|(\mathbf{I}-\mathbf{U}_l^s\mathbf{U}_l^{s^T})\boldsymbol{\mu}_l^s - (\mathbf{I}-\mathbf{U}_l^t\mathbf{U}_l^{t^T})\boldsymbol{\mu}_l^t\|_2\Big), \tag{9}$$

where $\mathbf{I}$ is the identity matrix, $\boldsymbol{\mu}_l^s$ and $\boldsymbol{\mu}_l^t$ are the mean vectors of covariances $\mathbf{C}_l^s$ and $\mathbf{C}_l^t$, respectively. Analogously, we also derive an error bound for the empirical estimation of the affine Grassmann metric.

Theorem 4. *Assuming the condition in Theorem 1 is specified by domains. Denote λ_i^s and λ_i^t as the i-th largest eigenvalue of the domain-specific covariance matrices $\mathbf{C}_s$ and $\mathbf{C}_t$, respectively. Let*

$$e^{AG}(d') = \frac{\sqrt{d'}\|\tilde{\boldsymbol{\mu}}^s\|_2}{\lambda_{d'}^s - \lambda_{d'+1}^s} + \frac{\sqrt{d'}\|\tilde{\boldsymbol{\mu}}^t\|_2}{\lambda_{d'}^t - \lambda_{d'+1}^t}$$

be the error index. Then the following error bound holds with probability at least $1-\delta$:

$$|d_{\mathcal{M}}^{AG}(\mathbf{C}^s,\mathbf{C}^t) - d_{\tilde{\mathcal{M}}}^{AG}(\tilde{\mathbf{C}}^s,\tilde{\mathbf{C}}^t)| \le 2\sqrt{2}E(\delta)\left(\frac{\sqrt{2d}}{4} + e^{AG}(d')\right).$$

Compared with the error index of the Grassmann distance e^G, the error terms of different domains in e^{AG} are weighted by the ℓ_2-norms of their mean vectors. This conclusion is consistent with the definition of an affine Grassmann manifold as the mean vectors are the coefficients of the affine transformation in Eq. (9). The proof is provided in Section S.3 of the supplementary material.

3.4.3 SPD manifold

Since the space of SPD matrices can be taken as a special type of manifold called the SPD manifold, theoretical research has been conducted to explore the non-Euclidean geometry of SPD manifolds [44], [45]. The affine invariant framework [44], called the Affine Invariant Riemannian Metric, analyzed the SPD manifold based on the inner product. Arsigny et al. [45] proposed a novel framework called Log-Euclidean which overcame the computational drawback of the affine invariant framework. Let $\mathbf{C}_l^{s/t} = \mathbf{U}_l^{s/t}\mathbf{D}_l^{s/t}\mathbf{U}_l^{s/t^T}$ be the eigendecomposition. Note that the Log-Euclidean metric requires all orthonormal basis and assumes the matrices are positive definite. Thus, we regularize the covariance matrices as $\mathbf{C}+\varepsilon\mathbf{I}$. The Log-Euclidean metric between $\mathbf{C}_l^s$ and $\mathbf{C}_l^t$ is defined as the distance between their matrix logarithms, i.e.,

$$d_{\mathcal{M}}^{LE}(\mathbf{C}_l^s,\mathbf{C}_l^t) = \frac{1}{d_l^2}\|\log(\mathbf{C}_l^s) - \log(\mathbf{C}_l^t)\|_F, \tag{10}$$

where $\log(\mathbf{C}_l^{s/t}) = \mathbf{U}_l^{s/t}\log(\mathbf{D}_l^{s/t})\mathbf{U}_l^{s/t^T}$.

4 Extension for Partial UDA

To build a unified algorithm for the vanilla and partial UDA problems, we introduce the weight-based extension for the DMP method in this section.

4.1 The Weighting Strategy

The essential problem for this extension is how can the negative information transduced from the outlier classes be mitigated. Under the partial UDA setting, the number of shared classes (i.e., target classes) is not larger than that of the source classes, i.e., $|\mathcal{C}^t| \le |\mathcal{C}^s|$. Since some shared classes are similar to the outlier classes in appearance (e.g., the bus in shared classes and car in outlier classes), the vanilla methods that align the entire feature spaces of two domains will probably lead to negative transfers. Also, the discriminant learning supervised by the soft labels should ignore the outlier classes, since intra-class compactness of the outlier classes is unnecessary and error-prone. To tackle these problems, we propose to learn the discriminative embedding and alignment with the class-wise weights $\mathbf{w} = (w_1, w_2, \ldots, w_c)^T$. It aims to assign the potential outlier classes with smaller weights during the learning process, then the model will be less sensitive to the outlier samples.

First, the discriminant criterion is modified to loosen the intra-class compactness constraints in the outlier classes. This modification preserves the intrinsic structure of the target domain and enhances discriminant learning in shared classes. The weighted intra-class loss is derived as

$$\mathcal{L}_{intra}^l(\mathbf{H}_l^t,\mathbf{P}^t) = -\frac{1}{n^t k}\sum_{i=1}^{c} w_i \sum_{j=1}^{n^t}\chi_{ij}p_{ij}^t s_{l,ij}^w. \tag{11}$$

The inter-class loss remains the same regardless of the UDA settings, since separabilities between any two classes are equally important to the final classification. The weighted discriminant criterion tends to align target instances to the class with high contribution w_i for the partial setting and treats the classes equally for the vanilla setting.

For the domain alignment, the target domain is supposed to be partially aligned to the source domain under the partial setting. Denoting the source manifold based on the shared classes by $\mathcal{M}^{s'}$, the optimal domain discrepancy is measured by $dist(\mathcal{M}^{s'},\mathcal{M}^t)$. Unfortunately, the manifold $\mathcal{M}^{s'}$ cannot be obtained, because the shared classes are unknown. To reduce the impact of outlier samples, the source feature matrix $\mathbf{H}^s$ is weighted by $\mathbf{w}$ as

$$\mathbf{h}_i^{s'} = w_{y_i}\mathbf{h}_i^s, \quad i = 1,2,\ldots,n^s.$$

Then, the weighted covariance $\mathbf{C}^{s'}$ can be computed from the weighted feature matrix $\mathbf{H}^{s'}$. The weighted alignment loss of the l-th manifold layer is described as

$$\mathcal{L}_{align}^l(\mathbf{H}_l^s,\mathbf{H}_l^t) = dist(\mathcal{M}_l^{s'},\mathcal{M}_l^t) \approx d_{\mathcal{M}}(\mathbf{C}_l^{s'},\mathbf{C}_l^t). \tag{12}$$

The weighted covariance $\mathbf{C}^{s'}$ will well characterize the manifold $\mathcal{M}^{s'}$ if the weight $\mathbf{w}$ is reliable, and the domain alignment will match the source and target domains partially to reduce the negative transfer.

As the target prediction matrix $\mathbf{P}^t$ reveals the contributions from different categories, it can be used to evaluate the probability that a certain class belongs to the shared classes. For the partial setting, the class-wise contribution vector $\boldsymbol{\pi} \in \mathbb{R}^c$ is computed as the mean value of the target prediction $\mathbf{P}^t$, i.e., $\boldsymbol{\pi} = (\sum_{i=1}^{n^t}\mathbf{p}_i^t)/n^t$. A smaller π_i indicates that the i-th class is more likely to be the outlier classes. However, the weights are actually useless in the vanilla setting, so the general weight vector $\hat{\mathbf{w}} = (w_1, w_2, \ldots, w_c)^T$ is defined as

$$\hat{w}_i = \begin{cases} \pi_i & , \text{ partial setting} \\ 1/c & , \text{ vanilla setting} \end{cases}, \quad i = 1,2,\ldots,c. \tag{13}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.3014218, IEEE Transactions on Pattern Analysis and Machine Intelligence

Algorithm 1 DMP Method

Input: Source data $\mathcal{X}^s$, Source labels $\mathbf{y}$, Target data $\mathcal{X}^t$, Batch-size b_s, Maximum iteration T_{max}, Update iteration T_{up}, Learning rate λ, Parameter λ_1 and λ_2;

Output: Learned network $\mathbf{\Theta}$, Target prediction $\hat{\mathbf{y}}^t$;

Initialize the network parameter $\mathbf{\Theta}$;
% Training Stage
for $t = 1, 2, \ldots, T_{max}$ **do**
  if $(t \bmod T_{up}) = 0$ **then**
    Forward propagate entire $\mathcal{X}^s$ without gradients; compute the source centers $\bar{\mathbf{H}}_l^s$, $\bar{\mathbf{h}}_l^s$ $(l = l_m, \ldots, L)$ and class weight vector $\hat{\mathbf{w}}$ from Eq. (13);
  end if
  Random select and forward propagate b_s samples from $\mathcal{X}^t$ and $\mathcal{X}^s$ (with label $\mathbf{y}$), respectively;
  Compute the objective function from Eq. (1).
  for $l = 1, 2 \ldots, L$ **do**
    Compute the gradient $\nabla\mathbf{\Theta}_l$ with parameters λ_1 and λ_2 from Eq. (17);
    Update the network parameter: $\mathbf{\Theta}_l \leftarrow \mathbf{\Theta}_l - \lambda\nabla\mathbf{\Theta}_l$;
  end for
end for
% Testing Stage
for $\mathcal{X}_j^t$ **in** $\mathcal{X}^t$ **do**
  Forward propagate $\mathcal{X}_j^t$ to obtain probability prediction $\mathbf{p}_j^t$;
  Compute the predicted label as $\hat{y}_j^t = \arg\max_i p_{ij}^t$
end for
Return the network parameter $\mathbf{\Theta} = \{\mathbf{\Theta}_1, \mathbf{\Theta}_2, \ldots, \mathbf{\Theta}_L\}$ and the target prediction $\hat{\mathbf{y}}^t = [\hat{y}_1^t, \hat{y}_2^t, \ldots]^T$;

The main advantages of our weighting strategy can be summarized as follows. 1) it assigns weights based on a constant mass since $\sum_i^c \hat{w}_i = 1$, which gives a consistent interpretation of the weights under different settings; 2) it considers the weighted model from both instance-level (i.e., the weighted manifold alignment) and class-level (i.e., the weighted discriminant criterion). As the weight vector is computed from the mean value of predictions, it sometimes suffers from the class imbalance problem which is the disadvantage of this strategy.

In fact, the ideal w_i should be $\frac{1}{c'}$ for the shared classes and 0 for the outlier classes. Denote the covariances computed from the estimated and ideal weights by $\mathbf{C}_{\hat{\mathbf{w}}}^{s'}$ and $\mathbf{C}_{\mathbf{w}}^{s'}$, respectively. The following theorem extends Theorem 3 to a weighting scheme.

Theorem 5. *Assuming the condition in Theorem 1 is specified by domains and $n^s = b_s \leq d$. Denote the error index under the partial setting by:*

$$e_P^G(d', \hat{\mathbf{w}}) = \alpha\|\mathbf{w}^2 - \hat{\mathbf{w}}^2\|_2 + 2\sqrt{2}E(\delta)e^G(d').$$

Then the following error bound:

$$|d_{\mathcal{M}}^G(\mathbf{C}_{\mathbf{w}}^{s'}, \mathbf{C}^t) - d_{\tilde{\mathcal{M}}}^G(\tilde{\mathbf{C}}_{\hat{\mathbf{w}}}^{s'}, \tilde{\mathbf{C}}^t)| \leq e_P^G(d', \hat{\mathbf{w}}),$$

holds with probability at least $1 - \delta$, where $\mathbf{w}^2$ and $\hat{\mathbf{w}}^2$ mean the element-wise square, and $\alpha(\hat{\mathbf{w}})$ is a constant that depends on $\hat{\mathbf{w}}$.

This theorem shows that the error under the partial setting is bounded not only by the gap between the continuous eigenvalues in $e^G(d')$, but also by the precision of the weight vector $\hat{\mathbf{w}}$. The more accurate that $\hat{\mathbf{w}}$ is helps the model effectively mitigate the misalignment problem. The proof is provided in Section S.4 of the supplementary material.

4.2 Optimization and Algorithm

The networks are optimized by back-propagation in a mini-batch training manner. For convenience, we assume that the manifold network starts from the l_m-th layer and ends at the penultimate layer. Denote the network parameter of the l-th layer as $\mathbf{\Theta}_l$ $(l = 1, 2, \ldots, l_m, \ldots, L)$. Note that $\mathbf{\Theta}$ not only includes the parameters of the manifold layers, but also contains the CNNs. The gradient of the objective function Eq. (1) referred by $\mathbf{\Theta}_l$ is divided into two parts according to the chain rule. The first part is the derivative of the objective function with respect to its input network features, and the second is the derivative of the network features with respect to the network parameter $\mathbf{\Theta}_l$. As the second part follows the standard network optimization paradigm, we only focus on the first part here.

As for the discriminant and alignment loss terms, the derivations begin with the objective function of the l'-th layer. Reformulating the inter-class loss $\mathcal{L}_{inter}^{l'}$ and intra-class loss $\mathcal{L}_{intra}^{l'}$ as

$$\mathcal{L}_{inter}^{l'}(\mathbf{H}_{l'}^s) = \frac{2}{c(c-1)}\sum_{i<j}\sum_{o=1}^{d_{l'}}\hat{h}_{l',oi}^s\hat{h}_{l',oj}^s,$$

$$\mathcal{L}_{intra}^{l'}(\mathbf{H}_{l'}^t, \mathbf{P}^t) = -\frac{1}{b_s k}\sum_{i=1}^{c}w_i\sum_{j=1}^{b_s}\chi_{ij}p_{ij}^t\sum_{o=1}^{d_{l'}}\bar{h}_{l',oi}^s h_{l',oj}^t.$$

Their derivatives are calculated as

$$\begin{aligned}\frac{\partial\mathcal{L}_{inter}^{l'}}{\partial\mathbf{\Theta}_l} &= \sum_{i<j}\sum_{o=1}^{d_{l'}}\left(\frac{\partial\mathcal{L}_{inter}^{l'}}{\partial h_{l',oi}^s}\frac{\partial h_{l',oi}^s}{\partial\mathbf{\Theta}_l} + \frac{\partial\mathcal{L}_{inter}^{l'}}{\partial h_{l',oj}^s}\frac{\partial h_{l',oj}^s}{\partial\mathbf{\Theta}_l}\right)\\ &= \frac{2}{c(c-1)}\sum_{i<j}\sum_{o=1}^{d_{l'}}\left(\frac{\hat{h}_{l',oj}^s}{n_{y_i}^s}\frac{\partial h_{l',oi}^s}{\partial\mathbf{\Theta}_l} + \frac{\hat{h}_{l',oi}^s}{n_{y_j}^s}\frac{\partial h_{l',oj}^s}{\partial\mathbf{\Theta}_l}\right),\\ \frac{\partial\mathcal{L}_{intra}^{l'}}{\partial\mathbf{\Theta}_l} &= \sum_{j=1}^{b_s}\sum_{o=1}^{d_{l'}}\frac{\partial\mathcal{L}_{intra}^{l'}}{\partial h_{l',oj}^t}\frac{\partial h_{l',oj}^t}{\partial\mathbf{\Theta}_l} + \sum_{i=1}^{c}\sum_{j=1}^{b_s}\frac{\partial\mathcal{L}_{intra}^{l'}}{\partial p_{ij}^t}\frac{\partial p_{ij}^t}{\partial\mathbf{\Theta}_l}\\ &= -\frac{1}{b_s k}\left(\sum_{i=1}^{c}\sum_{j=1}^{b_s}\sum_{o=1}^{d_{l'}}w_i\chi_{ij}p_{ij}^t\bar{h}_{l',oi}^s\frac{\partial h_{l',oj}^t}{\partial\mathbf{\Theta}_l}\right.\\ &\qquad\left.+\sum_{i=1}^{c}\sum_{j=1}^{b_s}w_i\chi_{ij}s_{l',ij}^w\frac{\partial p_{ij}^t}{\partial\mathbf{\Theta}_l}\right),\end{aligned}$$

where $n_{y_i}^s$ is the number of the y_i-th class's source samples in the current batch, and

$$\frac{\partial h_{l',oi}^s}{\partial\mathbf{\Theta}_l} = \frac{\partial h_{l',oj}^s}{\partial\mathbf{\Theta}_l} = \frac{\partial h_{l',oj}^t}{\partial\mathbf{\Theta}_l} = \mathbf{0}, \quad \text{if } l' < l.$$

The derivative of discriminative structure loss $\mathcal{L}_{DS}$ is obtained by combing the layer-wise derivatives:

$$\frac{\partial\mathcal{L}_{DS}}{\partial\mathbf{\Theta}_l} = \sum_{l'=l_m}^{L-1}\left(\frac{\partial\mathcal{L}_{inter}^{l'}}{\partial\mathbf{\Theta}_l} + \frac{\partial\mathcal{L}_{intra}^{l'}}{\partial\mathbf{\Theta}_l}\right). \tag{14}$$

In terms of manifold alignment loss, we take the case of the Grassmann manifold as an example. The manifold alignment loss $\mathcal{L}_{align}^{l'}$ requires the projection orthogonal bases of covariance matrices $\mathbf{C}_{l'}^{s'}$ and $\mathbf{C}_{l'}^t$, which are equivalent to the left-singular vectors of $\mathbf{H}_{l'}^{s'}$ and $\mathbf{H}_{l'}^t$, respectively. Specifically, the d_l' projection vectors required in the Grassmann manifold are computed from the truncated SVD:

$$\mathbf{H}_{l'}^{s'} = \mathbf{U}_{l'}^{s'}\mathbf{\Sigma}_{l'}^{s'}\mathbf{V}_{l'}^{s'^T}, \quad \mathbf{H}_{l'}^t = \mathbf{U}_{l'}^t\mathbf{\Sigma}_{l'}^t\mathbf{V}_{l'}^{t^T},$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.3014218, IEEE Transactions on Pattern Analysis and Machine Intelligence

where $\mathbf{U}_{l'}^{s'}, \mathbf{U}_{l'}^{t} \in \mathbb{R}^{d_{l'} \times d'_{l'}}$ and $\mathbf{\Sigma}_{l'}^{s'}, \mathbf{\Sigma}_{l'}^{t} \in \mathbb{R}^{d'_{l'} \times d'_{l'}}$. Following the matrix chain rule and the Jacobian of SVD [46], the derivative of $\mathcal{L}_{align}^{l'}$ is written as

$$\begin{aligned}
\frac{\partial \mathcal{L}_{align}^{l'}}{\partial \mathbf{\Theta}_l} &= \sum_{i=1}^{b_s} \sum_{o=1}^{d_{l'}} \left(\frac{\partial \mathcal{L}_{align}^{l'}}{\partial h_{l',oi}^{s'}} \frac{\partial h_{l',oi}^{s'}}{\mathbf{\Theta}_l} + \frac{\partial \mathcal{L}_{align}^{l'}}{\partial h_{l',oi}^{t}} \frac{\partial h_{l',oi}^{t}}{\mathbf{\Theta}_l} \right) \\
&= \sum_{i=1}^{b_s} \sum_{o=1}^{d_{l'}} \left[\mathrm{tr}\left(\mathbf{A}_{l',oi}^{s'} \right) \frac{\partial h_{l',oi}^{s'}}{\mathbf{\Theta}_l} + \mathrm{tr}\left(\mathbf{A}_{l',oi}^{t} \right) \frac{\partial h_{l',oi}^{t}}{\mathbf{\Theta}_l} \right], \\
\mathbf{A}_{l',oi}^{s'} &= 4 \left(\mathbf{\Omega}_{oi}^{\mathbf{U}_{l'}^{s'}} - \mathbf{U}_{l'}^{s'^T} \mathbf{U}_{l'}^{t} \mathbf{U}_{l'}^{t^T} \mathbf{U}_{l'}^{s'} \mathbf{\Omega}_{oi}^{\mathbf{U}_{l'}^{s'}} \right), \\
\mathbf{A}_{l',oi}^{t} &= 4 \left(\mathbf{\Omega}_{oi}^{\mathbf{U}_{l'}^{t}} - \mathbf{U}_{l'}^{t^T} \mathbf{U}_{l'}^{s'} \mathbf{U}_{l'}^{s'^T} \mathbf{U}_{l'}^{t} \mathbf{\Omega}_{oi}^{\mathbf{U}_{l'}^{t}} \right),
\end{aligned}$$

where $\mathbf{\Omega}_{oi}^{\mathbf{U}_{l'}^{s'}}$ and $\mathbf{\Omega}_{oi}^{\mathbf{U}_{l'}^{t}}$ are antisymmetric matrices which can be computed by solving a set of linear systems [46]. Detailed derivations are provided in Section S.5 of the supplementary material. Similarly, the derivative of alignment loss is deduced by combining the layer-wise derivatives:

$$\frac{\partial \mathcal{L}_{AL}}{\partial \mathbf{\Theta}_l} = \sum_{l'=l_m}^{L-1} \frac{\partial \mathcal{L}_{align}^{l'}}{\partial \mathbf{\Theta}_l}. \tag{15}$$

Recall that the cross-entropy loss $\mathcal{L}_{CE}$ is formulated as $\mathcal{L}_{CE} = \frac{1}{b_s} \sum_{i=1}^{b_s} -\log p_{y_i i}^{s}$. The derivative of the $\mathcal{L}_{CE}$ with respect to $\mathbf{\Theta}_l$ is

$$\frac{\partial \mathcal{L}_{CE}}{\partial \mathbf{\Theta}_l} = \frac{1}{b_s} \sum_{i=1}^{b_s} \frac{\partial \mathcal{L}_{CE}}{\partial p_{y_i i}^{s}} \frac{\partial p_{y_i i}^{s}}{\partial \mathbf{\Theta}_l} = \sum_{i=1}^{b_s} -\frac{1}{b_s p_{y_i i}^{s}} \frac{\partial p_{y_i i}^{s}}{\partial \mathbf{\Theta}_l}. \tag{16}$$

Finally, the gradient of $\mathbf{\Theta}_l$ is deduced from the Eq. (14)-(16):

$$\nabla \mathbf{\Theta}_l = \frac{\partial \mathcal{L}}{\partial \mathbf{\Theta}_l} = \frac{\partial \mathcal{L}_{CE}}{\partial \mathbf{\Theta}_l} + \lambda_1 \frac{\partial \mathcal{L}_{DS}}{\partial \mathbf{\Theta}_l} + \lambda_2 \frac{\partial \mathcal{L}_{AL}}{\partial \mathbf{\Theta}_l}. \tag{17}$$

The pseudo code for the optimization of the DMP framework is summarized in Algorithm 1.

5 Experiments and Comparative Analysis

In this section, the proposed method is evaluated on four standard UDA datasets. The experimental setting is detailed in Section 5.1. In Section 5.2, we analyze the hyper-parameters of the DMP method. Based on the results of hyper-parameter anlysis, the comparison experiments for the vanilla and partial UDA problems are conducted in Sections 5.3 and 5.4, respectively, where the results of the ablation study are also provided. Further analysis on the proposed method is presented in Section 5.5.

5.1 Datasets and Experimental Setting

Four popular domain adaptation datasets were selected, and the standard protocols were adopted.

Office 31 [47] is an object recognition dataset. It contains 3 domains with a total of 4110 images, i.e., *Amazon* (**A**), *Webcam* (**W**) and *Dslr* (**D**). All domains consist of 12 common classes, e,g., pen, monitor and speaker, and the sample sizes of the domains differ. Following the protocols in [6], [7], the target domain consists of 10 classes extracted from the Office 31 dataset under the partial setting.

Office-Home [48] dataset contains 4 domains, i.e., *Art* (**Ar**), *Clipart* (**Cl**), *Product* (**Pr**) and *Real-World* (**Rw**), and 12 domain adaptation tasks. 15500 images are assigned into 65 categories for all domains. There are around 70 images for each category. For the partial setting, the first 25 classes (in alphabetical order) are taken as the target domains. Similarly, the sample sizes of different domains are unbalanced.

ImageCLEF[1] dataset is obtained from the ImageCLEF Domain Adaptation challenge held in 2014. The domains *Caltech* (**C**), *ImageNet* (**I**), *Pascal* (**P**), *Bing* (**B**) were collected from four previous proposed datasets, i.e., Caltech-256 [9], ImageNet ILSVRC2012 [49], PASCAL VOC2012 [10] and Bing [50], respectively. Each domain consists of 600 images with 12 classes and all 4 domains are the same size. Following previous protocols [2], [18], [23], [36], we conduct adaptation tasks between *Caltech*, *ImageNet* and *Pascal*. For the partial setting, the target domain consists of the first 6 classes (in alphabetical order) of the vanilla data.

VisDA-2017 [51] is a large-scale visual domain adaptation challenge dataset. It aims to transfer the knowledge learned from sufficient synthetic data to real-life visual scenes. The source domain **S** (**synthetic**) collected 152397 images generated from 3D models. The target domain **R** (**real-image**) extracted 55388 cropped images from the Microsoft COCO databse [52]. All collected images were assigned into 12 common classes. Following the challenge track, we carried out a **S** $\rightarrow$ **R** task for the vanilla setting, and a **S** $\rightarrow$ **R6** (the first 6 classes in alphabetical order) task for the partial setting.

In the following experiments, ResNet models [53] pre-trained on ImageNet were used for feature extraction. Specifically, ResNet-50 and Adam Optimizer ($lr = 0.0002$, $\beta_1 = 0.9$, $\beta_2 = 0.999$) with batch size of 50 were used on Office 31, Office-Home and ImageCLEF datasets. ResNet-101 and the modified mini-batch SGD (momentum = 0.9, weight decay = $5e$-4) with batch sizes of 32 were employed on VisDA-2017 challenge. The initial learning rate was set as 0.003 and then adjusted by the annealing learning schedule described in [18]. A two layer Riemmanian manifold learning scheme was carried out in all experiments, with the first layer ($d_1 = 1024$) activated by Leaky ReLU ($\alpha = 0.2$) and the second ($d_2 = 512$) by Tanh. The learning rates for the feature extraction layers and Riemannian manifold layers learned from scratch were set as $0.1lr$ and lr. We implemented DMP on a GPU PyTorch platform with an NVIDIA GTX TITAN Xp.

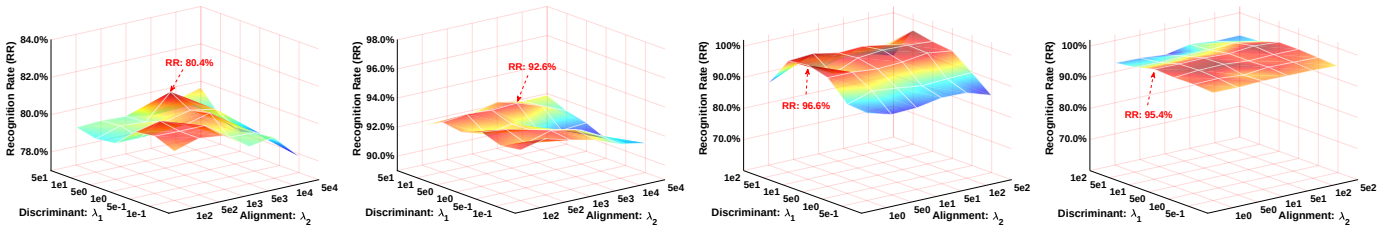
5.2 Hyper-parameter Analysis

The penalty parameters λ_1, λ_2 and truncated parameter k were investigated first. We searched the penalty parameters with the Top-1 preserving scheme. Then the truncated parameter k was analyzed based on the optimal parameters λ_1, λ_2 in the last stage.

For the vanilla setting, the experiments were conducted on ImageCLEF, and λ_1 and λ_2 were tested for each value in groups $\{1e\text{-}1, 5e\text{-}1, 1e0, 5e0, 1e1, 5e1\}$ and $\{1e2, 5e2, 1e3, 5e3, 1e4, 5e4\}$, respectively. For the partial setting, the experiments were carried out on partial Office-31, and the optimal values of λ_1 and λ_2 were searched from $\{5e\text{-}1, 1e0, 5e0, 1e1, 5e1, 1e2\}$ and $\{1e0, 5e0, 1e1, 5e1, 1e2, 5e2\}$, respectively. All settings were repeated 50 times and heatmaps of mean results are presented in Figure 4. The highest accuracies, i.e., the *peaks* in the figures, are marked with red arrows. Regions nearby the *peaks* are flat, which means the proposed method are stable in those regions. In the vanilla tasks, setting $(\lambda_1, \lambda_2) = (1e1, 5e3)$ achieved the

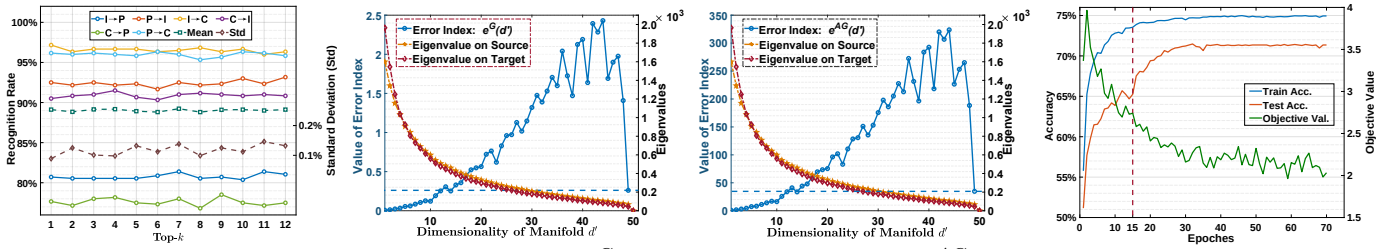
1. https://www.imageclef.org/2014/adaptation

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.3014218, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) Task $\mathbf{I} \rightarrow \mathbf{P}$ (b) Task $\mathbf{P} \rightarrow \mathbf{I}$ (c) Task $\mathbf{A} \rightarrow \mathbf{W}$ (d) Task $\mathbf{W} \rightarrow \mathbf{A}$

Fig. 4. 3D heatmaps of the recognition rates with different penalty hyper-parameters under different settings. (a)-(b): *vanilla* setting on ImageCLEF. (c)-(d): *partial* setting on Office-31. Best viewed in color.



(a) Sensitivity to parameter k (b) Visualization of $e^G(d')$ (c) Visualization of $e^{AG}(d')$ (d) Convergence curves

Fig. 5. (a): Recognition rate curves of different truncated parameters k on ImageCLEF dataset. The cyan and brown dash curves indicate the mean and standard deviation, respectively. (b)-(c): Error and eigenvalue curves w.r.t. the dimensionality d'. The $(b_s - 1)$-th error index is highlighted by the horizontal dash line. (d): Recognition rate curves and the objective curve on the Office-31 dataset (**A**→**W**). Best viewed in color.

1st and 2nd best recognition rates in tasks $\mathbf{I} \rightarrow \mathbf{P}$ and $\mathbf{P} \rightarrow \mathbf{I}$, respectively. As the domain alignment should be conservative in the partial setting, the optimal alignment parameter λ_2 is smaller than that in the vanilla setting. Figure 4(c)-(d) shows that the optimal parameter setting, i.e., (λ_1, λ_2) = (1e1,1e0), achieved high accuracies in partial tasks $\mathbf{A} \rightarrow \mathbf{W}$ and $\mathbf{W} \rightarrow \mathbf{A}$.

We fixed parameters (λ_1, λ_2)=(1e1,5e3), and then investigated the sensitivity of the proposed method with reference to the truncated parameter k. The recognition curves of all six transfer tasks on ImageCLEF are shown in Figure 5(a). Our method is robust to the selection of k, as all curves are flat and stable. The mean accuracy, shown as the cyan dashed line, suggests that the accuracies of different truncated settings are nearly the same. However, it can be observed from the brown dashed line that truncation makes the method more robust and the standard deviation of the Top-1 scheme is the smallest.

To explore the minimal errors of Grassmann and affine Grassmann distances, a numerical simulation was conducted on the ImageCLEF dataset. As the eigenvalues always decrease rapidly at the beginning and then become flat, the error bounds of dimensionality d' located in the flattened eigenvalue region are too high to assess. As shown in Figure 5(b)-(c), the trend of eigenvalues is consistent with the description. Though the dramatic decrease in the initial stage results in a lower error, the information in that area is unconvincing and insufficient to support the measurement of manifolds. Since there is a natural gap between the $(b_s - 1)$-th and the b_s-th dominant eigenvalues, $(b_s - 1)$-th error index is smaller than most of the other errors. We highlight the $(b_s - 1)$-th error index by the blue dashed line, and observe only errors of $d' = \{1, 2, \ldots, 12, 14\}$ that are lower than the $(b_s - 1)$-th error. The errors of the Grassmann distance and affine Grassmann distance have the same trend, this is because the covariances of different domains are aligned via manifold metrics. Empirically, the dimensions of the Grassmann and affine Grassmann manifolds were set as $(b_s - 1)$ hereafter.

The ablation study of manifold selection was conducted on the ImageCLEF dataset, and the results are shown in Table 1. The Grassmann, affine Grassmann and Log-Euclidean metrics are abbreviated as G, AG and LE, respectively. The accuracies of the different manifolds are almost the same under the vanilla setting, but the Grassmann manifold is better than the others under the partial setting. This is because the affine Grassmann distance and Log-Euclidean metric refer to the mean vectors and eigenvalues and are more sensitive to the weight vector $\mathbf{w}$. Empirically, the Grassmann manifold was selected for the domain alignment hereafter.

TABLE 1
Ablation study of the manifold selection on ImageCLEF.

Manifolds		I→P	P→I	I→C	C→I	C→P	P→C	Mean
Vanilla	G	80.7	92.5	97.2	90.5	77.7	96.2	89.1
	AG	81.0	92.0	96.3	91.0	76.5	96.5	88.9
	LE	81.4	92.5	96.3	91.3	78.0	95.3	89.1
Partial	G	82.4	94.5	96.7	94.3	78.7	96.4	90.5
	AG	79.8	90.9	95.7	89.7	75.9	95.4	87.9
	LE	81.0	92.0	96.3	91.0	76.5	96.5	88.9

Figure 5(d) shows the convergence curves on Office-31 **A**→**W** adaptation task. At the beginning, the objective values decrease quickly and the recognition rates tend to be stable in epochs 10-15. The intra-class structure constraint was imposed after 15 epochs, which led to a continuous improvement in the recognition rate and alleviated the over-fitting problem on the source domain.

5.3 Comparative Experiments under Vanilla Setting

In this section, we compare the proposed method with other state-of-the-art vanilla UDA approaches: DAN [2], Domain Adversarial Neural Network (DANN) [18], JAN [36], SWD [54], CDAN [23], BSP based on DANN (BSP+DANN) or CDAN (BSP+CDAN) [27], Hard AFN (HAFN) and Stepwise AFN (SAFN) [28].

The parameters λ_1 and λ_2 were set as 1e1 and 5e3, respectively. The Top-1 scheme was adopted for the target intra-class loss in Eq. (6). For the ablation study, the models without discriminative

TABLE 2
Class-wise recognition rates (%) on VisDA-2017 (ResNet-101) and recognition rates (%) on Office-Home, ImageCLEF and Office-31 (ResNet-50) under the Vanilla Setting. The superscripts denote standard deviations hereafter.

VisDA-2017	Plane	bcycl	bus	car	horse	knife	mcyle	person	plant	sktbrd	train	truck	Mean
ResNet-101 [53]	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
DAN [2]	87.1	63.0	76.5	42.0	90.3	42.9	85.9	53.1	49.7	36.3	85.8	20.7	61.1
DANN [18]	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
CDAN [23]	85.2	66.9	83.0	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38.0	73.7
BSP+DANN [27]	92.2	72.5	83.8	47.5	87.0	54.0	86.8	72.4	80.6	66.9	84.5	37.1	72.1
BSP+CDAN [27]	92.4	61.0	81.0	57.5	89.0	80.6	90.1	77.0	84.2	**77.9**	82.1	38.4	75.9
HAFN [28]	92.7	55.4	82.4	70.9	93.2	71.2	90.8	78.2	89.1	50.2	**88.9**	24.5	73.9
SAFN [28]	93.6	61.3	84.1	70.6	**94.1**	79.0	91.8	79.6	89.9	55.6	89.0	24.4	76.1
DMP (No AL)	92.8	15.3	**86.7**	**86.3**	93.8	70.7	**95.2**	68.9	**95.8**	40.4	85.1	5.6	69.7
DMP (No DS)	90.2	66.5	70.2	65.8	79.8	81.8	84.7	70.1	82.0	46.5	88.1	27.7	71.1
DMP	92.1	75.0	78.9	75.5	91.2	**81.9**	89.0	77.2	93.3	77.4	84.8	35.1	**79.3**

Office-Home	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Mean
ResNet-50 [53]	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 [2]	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 [18]	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 [36]	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 [23]	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 [23]	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
BSP+DANN [27]	51.4	68.3	75.9	56.0	67.8	68.8	57.0	49.6	75.8	70.4	57.1	80.6	64.9
BSP+CDAN [27]	52.0	68.6	76.1	58.0	70.3	70.2	58.6	50.2	77.6	**72.2**	**59.3**	**81.9**	66.3
HAFN [28]	50.2	70.1	76.6	61.1	68.0	70.7	59.5	48.4	77.3	69.4	53.0	80.2	65.4
SAFN [28]	52.0	71.7	76.3	64.2	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
DMP (No AL)	51.9	72.8	77.1	63.0	**72.0**	71.3	60.5	49.5	78.4	71.5	54.4	82.8	67.1
DMP (No DS)	51.2	72.4	**77.7**	63.0	71.4	71.4	58.6	44.6	**79.1**	71.1	53.4	81.5	66.3
DMP	**52.3**	**73.0**	77.3	**64.3**	**72.0**	**71.8**	**63.6**	**52.7**	78.5	72.0	57.7	81.6	**68.1**

Methods	ImageCLEF							Office-31						
	I→P	P→I	I→C	C→I	C→P	P→C	Mean	A→W	D→W	W→D	A→D	D→A	W→A	Mean
ResNet-50 [53]	$74.8^{0.3}$	$83.9^{0.1}$	$91.5^{0.3}$	$78.0^{0.2}$	$65.5^{0.3}$	$91.2^{0.3}$	80.7	$68.4^{0.2}$	$96.7^{0.1}$	$99.3^{0.1}$	$68.9^{0.2}$	$62.5^{0.3}$	$60.7^{0.3}$	76.1
DAN [2]	$74.5^{0.4}$	$82.2^{0.2}$	$92.8^{0.2}$	$86.3^{0.4}$	$69.2^{0.4}$	$89.8^{0.4}$	82.5	$80.5^{0.4}$	$97.1^{0.2}$	$99.6^{0.1}$	$78.6^{0.2}$	$63.6^{0.3}$	$62.8^{0.2}$	80.4
DANN [18]	$75.0^{0.3}$	$86.0^{0.3}$	$96.2^{0.4}$	$87.0^{0.5}$	$74.3^{0.5}$	$91.5^{0.6}$	85.0	$82.0^{0.4}$	$96.9^{0.2}$	$99.1^{0.1}$	$79.7^{0.4}$	$68.2^{0.4}$	$67.4^{0.5}$	82.2
JAN [36]	$76.8^{0.4}$	$88.0^{0.2}$	$94.7^{0.2}$	$89.5^{0.3}$	$74.2^{0.3}$	$91.7^{0.3}$	85.8	$85.4^{0.3}$	$97.4^{0.2}$	$99.8^{0.2}$	$84.7^{0.3}$	$68.6^{0.3}$	$70.0^{0.4}$	84.3
CDAN [23]	$76.7^{0.3}$	$90.6^{0.3}$	$97.0^{0.4}$	$90.5^{0.4}$	$74.5^{0.3}$	$93.5^{0.4}$	87.1	$93.1^{0.2}$	$98.2^{0.2}$	$\mathbf{100.0}^{0.0}$	$89.8^{0.3}$	$70.1^{0.4}$	$68.0^{0.4}$	86.6
CDAN+E [23]	$77.7^{0.3}$	$90.7^{0.2}$	$\mathbf{97.7}^{0.3}$	$\mathbf{91.3}^{0.3}$	$74.2^{0.2}$	$94.3^{0.3}$	87.7	$\mathbf{94.1}^{0.1}$	$98.6^{0.1}$	$\mathbf{100.0}^{0.0}$	$\mathbf{92.9}^{0.2}$	$71.0^{0.3}$	$69.3^{0.3}$	**87.7**
HAFN [28]	$76.9^{0.4}$	$89.0^{0.4}$	$94.4^{0.1}$	$89.6^{0.6}$	$74.9^{0.2}$	$92.9^{0.1}$	86.3	$83.4^{0.7}$	$98.3^{0.1}$	$99.7^{0.1}$	$84.4^{0.7}$	$69.4^{0.5}$	$68.5^{0.3}$	83.9
SAFN [28]	$78.0^{0.4}$	$91.7^{0.5}$	$96.2^{0.1}$	$91.1^{0.3}$	$77.0^{0.5}$	$94.7^{0.3}$	88.1	$88.8^{0.4}$	$98.4^{0.0}$	$99.8^{0.0}$	$87.7^{1.3}$	$69.8^{0.4}$	$69.7^{0.2}$	85.7
DMP (No AL)	$78.0^{0.1}$	$91.1^{0.1}$	$95.6^{0.2}$	$88.7^{0.3}$	$74.8^{0.1}$	$94.8^{0.2}$	87.3	$87.5^{0.4}$	$98.9^{0.1}$	$\mathbf{100.0}^{0.0}$	$86.8^{0.3}$	$67.6^{0.1}$	$64.1^{0.3}$	84.1
DMP (No DS)	$78.9^{0.1}$	$90.5^{0.2}$	$94.0^{0.1}$	$87.8^{0.1}$	$76.7^{0.2}$	$93.0^{0.1}$	86.8	$82.7^{0.2}$	$97.7^{0.1}$	$\mathbf{100.0}^{0.0}$	$82.0^{0.2}$	$66.2^{0.2}$	$65.5^{0.2}$	82.3
DMP	$\mathbf{80.7}^{0.1}$	$\mathbf{92.5}^{0.1}$	$97.2^{0.1}$	$90.5^{0.1}$	$\mathbf{77.7}^{0.2}$	$\mathbf{96.2}^{0.2}$	**89.1**	$93.0^{0.3}$	$\mathbf{99.0}^{0.1}$	$\mathbf{100.0}^{0.0}$	$91.0^{0.4}$	$\mathbf{71.4}^{0.2}$	$\mathbf{70.2}^{0.2}$	87.4

structure loss and manifold metric alignment loss are abbreviated as DMP (No DS) and DMP (No AL), respectively.

The experimental results on the Visda-2017 dataset are shown in the top of Table 2. The recognition rates for each class are reported. It was observed that DMP outperforms the other methods by a large margin in mean accuracy, which is directly derived from the average of class-wise accuracies. It indicates the overall performance of adaptation methods and demonstrates their ability to handle the class imbalance problem. To achieve a higher classification result, methods need to deal with their own "hard" classes (e.g., *person* for DANN [18]) and common barriers (e.g., *track*). For example, DMP (No AL) achieves top accuracies in most class-wise results, but the mean accuracy is only 69.7%. The class-wise results of DMP are more balanced than other methods because the inter-class structure learning in Eq. (3) treats all classes equally. As shown in the middle of Table 2, the proposed method improves the mean accuracy to 68.1% and achieves the highest accuracy in most of the adaptation tasks on the Office-Home dataset. The second best method is SAFN [27] which improves it mean accuracy to 67.3% and obtains highest results in tasks **Rw→Ar**, **Rw→Cl** and **Rw→Pr**.

The ablation results also validate the effectiveness of the Riemannian manifold learning framework when both loss terms are used. As the discriminative structure loss provides a separable structure and manifold metric alignment loss bridges the distribution discrepancy between the source and target domains based on the Grassmann distance, both loss terms are important. In Table 2, the accuracy of DMP is at least 1% higher than the other variants. The overall results demonstrate the importance and effectiveness of discriminant information, as BSP [27] and DMP are significantly more accurate than the other methods.

The results on the ImageCLEF dataset are shown in the bottom of Table 2. As the discrepancy between the source and target domains on the ImageCLEF dataset is relatively smaller than that in the other datasets, the baseline model ResNet-50 [53] achieves 80.7% accuracy on average. In this case, a more discriminative model is essential for improvement of recognition. DMP encodes the discriminant criterion and alignment constraint simultaneously, thus it outperforms other methods by at least 1.4%. CDAN+E [23] exploits the entropy information and improves accuracy to 87.7%. The bottom of Table 2 suggests that the accuracies of DMP surpass most of the competitors except for CDAN+E [23] on the Office-31 dataset, as CDAN+E is stronger than CDAN by encoding the entropy from the target predictions to the classifier. Note DMP is only 0.3% lower than CDAN+E on average and achieves highest accuracies on **D→A** and **W→A**.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.3014218, IEEE Transactions on Pattern Analysis and Machine Intelligence

TABLE 3
Recognition rates (%) on ImageCLEF, Office-31, Office-Home and VisDA-2017 under the Partial Setting (ResNet-50).

Methods	Office-31							ImageCLEF						
	A→W	D→W	W→D	A→D	D→A	W→A	Mean	I→P	P→I	I→C	C→I	C→P	P→C	Mean
ResNet-50 [53]	$75.6^{1.1}$	$96.3^{0.9}$	$98.1^{0.7}$	$83.4^{1.1}$	$83.9^{1.0}$	$85.0^{0.9}$	87.1	$78.3^{0.2}$	$86.9^{0.2}$	$91.0^{0.2}$	$84.3^{0.4}$	$72.5^{0.4}$	$91.5^{0.3}$	84.1
DANN [18]	$73.6^{0.2}$	$96.3^{0.3}$	$98.7^{0.2}$	$81.5^{0.2}$	$82.8^{0.2}$	$86.1^{0.2}$	86.5	$78.1^{0.2}$	$86.3^{0.2}$	$91.3^{0.4}$	$84.0^{0.3}$	$72.1^{0.3}$	$90.3^{0.2}$	83.7
PADA [7]	$86.5^{0.3}$	$99.3^{0.5}$	$\mathbf{100.0}^{0.0}$	$82.2^{0.4}$	$92.7^{0.3}$	$\mathbf{95.4}^{0.3}$	92.7	$81.7^{0.2}$	$92.1^{0.2}$	$94.6^{0.2}$	$89.8^{0.2}$	$77.7^{0.3}$	$94.1^{0.1}$	88.3
SAN [22]	$93.9^{0.5}$	$99.3^{0.5}$	$99.4^{0.1}$	$94.3^{0.3}$	$94.2^{0.4}$	$88.7^{0.4}$	95.0	$81.6^{0.2}$	$91.1^{0.2}$	$95.9^{0.3}$	$90.4^{0.6}$	$78.5^{0.2}$	$\mathbf{97.1}^{0.3}$	89.1
HAFN [28]	$87.5^{0.3}$	$96.7^{0.4}$	$99.2^{0.3}$	$87.3^{0.5}$	$89.2^{0.2}$	$90.7^{0.2}$	91.7	$79.1^{0.2}$	$87.7^{0.1}$	$93.7^{0.1}$	$90.3^{0.2}$	$77.8^{0.1}$	$94.7^{0.3}$	87.2
SAFN [28]	$87.5^{0.7}$	$96.6^{0.2}$	$99.4^{0.7}$	$89.8^{1.5}$	$92.6^{0.2}$	$92.7^{0.1}$	93.1	$79.5^{0.2}$	$90.7^{0.2}$	$93.0^{0.1}$	$90.3^{0.1}$	$77.8^{0.2}$	$94.0^{0.2}$	87.5
DMP	$94.5^{0.5}$	$99.9^{0.1}$	$\mathbf{100.0}^{0.0}$	$95.0^{1.0}$	$94.7^{0.3}$	$\mathbf{95.4}^{0.3}$	96.6	$81.5^{0.2}$	$94.3^{0.1}$	$96.2^{0.1}$	$93.0^{0.3}$	$78.2^{0.2}$	$96.5^{0.1}$	90.0
DMP+ent	$\mathbf{96.6}^{0.9}$	$\mathbf{100.0}^{0.0}$	$\mathbf{100.0}^{0.0}$	$\mathbf{96.4}^{0.9}$	$\mathbf{95.1}^{0.2}$	$\mathbf{95.4}^{0.1}$	**97.2**	$\mathbf{82.4}^{0.2}$	$\mathbf{94.5}^{0.3}$	$\mathbf{96.7}^{0.2}$	$\mathbf{94.3}^{0.1}$	$\mathbf{78.7}^{0.8}$	$96.4^{0.2}$	**90.5**

Methods	Office-Home													VisDA-2017
	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Mean	S→R6
ResNet-50 [53]	46.3	67.5	75.9	59.1	59.9	62.7	58.2	41.8	74.9	67.4	48.2	74.2	61.4	45.3
DANN [18]	43.8	67.9	77.5	63.7	59.0	67.6	56.8	37.1	76.4	69.2	44.3	77.5	61.7	51.0
IWAN [6]	53.9	54.5	78.1	61.3	48.0	63.3	54.2	52.0	81.3	76.5	56.8	82.9	63.6	-
PADA [7]	52.0	67.0	78.7	52.2	53.8	59.0	52.6	43.2	78.8	73.7	56.6	77.1	62.1	53.5
SAN [22]	44.4	68.7	74.6	67.5	65.0	77.8	59.8	44.7	80.1	72.2	50.2	78.7	65.3	-
ETN [21]	**59.2**	77.0	79.5	62.9	65.7	75.0	68.3	55.4	84.4	75.7	57.7	**84.5**	70.5	-
HAFN [28]	53.4	72.7	80.8	64.2	65.3	71.1	66.1	51.6	78.3	72.5	55.3	79.0	67.5	65.1
SAFN [28]	58.9	76.3	81.4	**70.4**	**73.0**	77.8	**72.4**	55.3	80.4	75.8	60.4	79.9	71.8	67.7
DMP	54.0	71.9	81.3	63.2	61.6	70.0	62.3	49.5	77.2	73.4	54.1	79.4	66.5	67.6
DMP+ent	59.0	**81.2**	**86.3**	68.1	72.8	**78.8**	71.2	**57.6**	**84.9**	**77.3**	**61.5**	82.9	**73.5**	**72.7**

5.4 Comparative Experiment Under Partial Setting

Comparative experiments were conducted on four datasets under the partial setting. Several advanced partial UDA methods were selected for comparison: DANN [18], IWAN [6], PADA [7], SAN [22], ETN [21], HAFN and SAFN [28].

The parameters λ_1 and λ_2 were set as $1e1$ and $1e0$, respectively. The Top-1 preserving DMP was adopted. Since part of the source information is useless and even negative for adaptation, target entropy loss, which has been widely applied in UDA problems [2], [28], was employed here. It helps the target domain preserve its intrinsic structure by pushing the decision hyper-plane to the low density region. We abbreviate the target entropy version of DMP as DMP+ent.

The results on the Office 31 and ImageCLEF datasets under the partial setting are presented in the top of Table 3. The mean accuracy of DMP+ent is higher than others. The improvements are significant in tasks **A→W**, **A→D**, **P→C** and **C→I**, and DMP+ent is at least 1.4% higher than the other methods. The original DMP method is only 0.5% to 0.6% lower than the target entropy variant DMP+ent, which demonstrates DMP is effective in mining and preserving the target discriminative structure.

The results on the Office-Home and VisDA-2017 datasets are shown in the bottom of Table 3. The results of DMP+ent are significantly higher than those of DMP. The main reasons for the significant improvement are the complex data structure and large domain discrepancy in the Office-Home and VisDA-2017 datasets. The performance of the original DMP is still better than most of the other methods except for AFN and ETN. The SAFN method surpasses the ETN method with 71.8% in accuracy, and the accuracy of DMP+ent is at least 1.7% higher than other methods.

5.5 Method Analysis

Features Visualization. To visualize the quality of the adaptation performance, we randomly selected 1200 images from the source and target domains of VisDA-2017 [51]. These 1200 images were collected from 12 classes with 100 images per class. ResNet-101 [53] and CDAN+E [23] were used to compare with DMP.

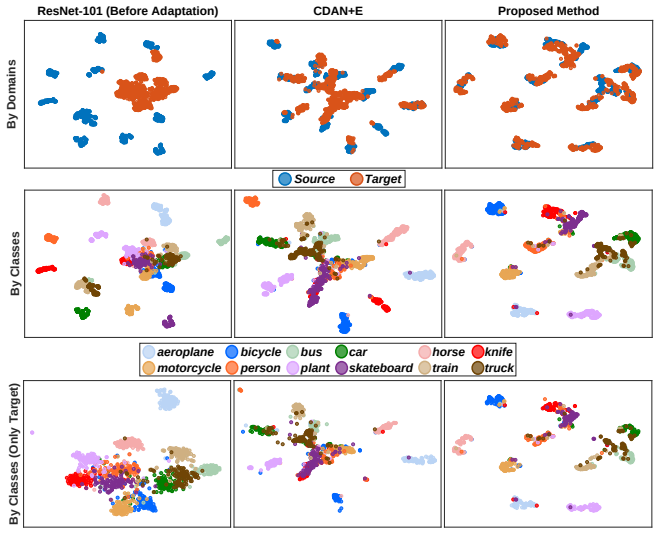
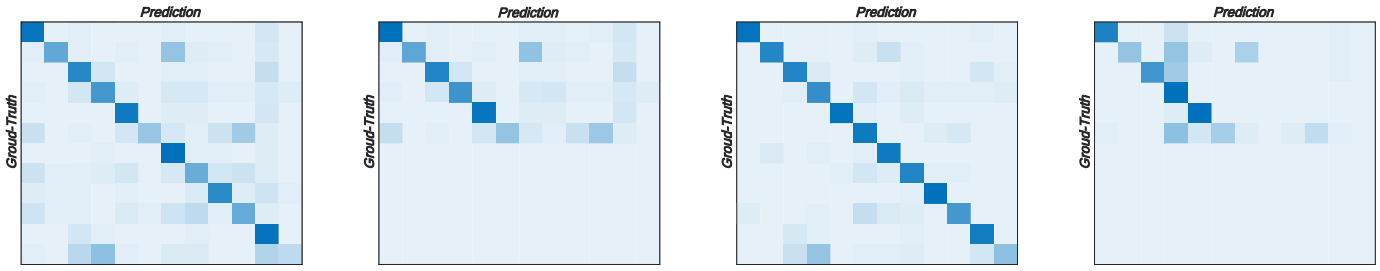


Fig. 6. Visualization of learned features using 2-D t-SNE [55] on the VisDA-2017 dataset. The first and second rows show the feature representations colored by domains and classes, respectively. The third row shows the target samples colored by classes. Best viewed in color.

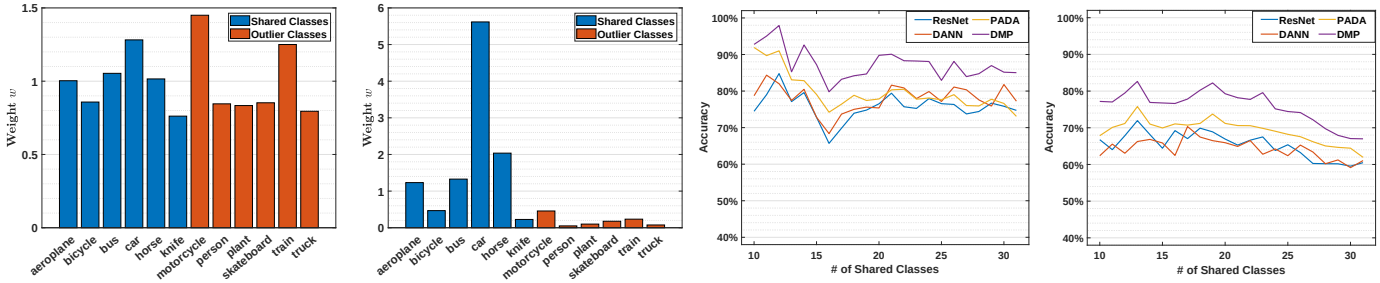
Figure 6 shows the 2-D representation spaces obtained from the t-SNE [55] algorithm. As shown in the first column, there is no alignment constraint between the source and target distributions in the ResNet-101 model. Though the categories on the source domain are separable, the target samples are really indistinguishable. CDAN+E shortens the distance between the source and target domains by using adversarial alignment. Some of the classes have been dragged away from the center, e.g., *plant, car, horse, aeroplane* and *bicycle*. However, the center is still an unresolved region, where *skateboard* and *knife* totally overlap. In the third column, our method further optimizes the structure of the embedding space. The categories are aligned better than ResNet-101 and CDAN+E, leading to a more compact target space. Since DMP achieves the discriminant criterion transductively, the intra-class samples are more compact than in the other methods, and all classes are more separable in both the source and target domains.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2020.3014218, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) ResNet under Vanilla Setting (b) ResNet under Partial Setting (c) DMP under Vanilla Setting (d) DMP under Partial Setting

Fig. 7. Confusion matrices of the target domain on the VisDA-2017 dataset; darker colors represent larger values. (a)-(b): ResNet backbone network, which is regarded as the before adaptation case; (c)-(d): DMP network, which shows the performance after adaptation. Best viewed in color.



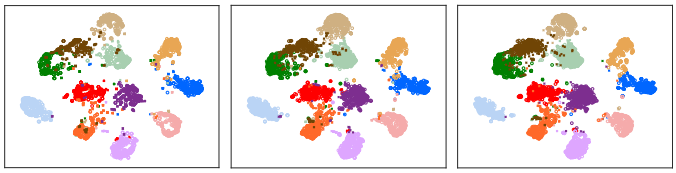
(a) Weights of ResNet-50 (b) Weights of DMP (c) A→W (d) W→A

Fig. 8. (a)-(b): Class weights computed by different methods on the VisDA-2017 dataset under the partial setting. (c)-(d): Accuracy curves by different numbers of shared classes on the Office-31 dataset under the partial setting. Best viewed in color.

Class-wise Confusion and Weight Visualization. In this section, we provide the confusion matrices of the ResNet baseline (before adaptation) and DMP on VisDA-2017 under vanilla and partial settings in Figure 7. This shows the quantity of misclassified samples, which cannot be seen in the accuracy table and t-SNE visualization. ResNet-101 and ResNet-50 were employed as backbone networks for the vanilla and partial adaptations, respectively. In Figure 7(a), the off-diagonal elements of ResNet (before adaptation) are larger than DMP (after adaptation) in Figure 7(c), which demonstrates that DMP alleviates the confusion between these classes. For the partial adaptation in Figure 7(b) and 7(d), DMP improves the accuracies of *car* (4th diagonal element) and *horse* (5th diagonal element) to about 95%. From the perspective of mitigating negative transfer, we can take the misclassified mass of the outlier classes, i.e. the last 6 classes in the partial setting, as a performance evaluation. The density of the outlier region in Figure 7(d) is much lower than that in Figure 7(b), which demonstrates that DMP is effective in identifying the outlier classes and alleviating negative transfer.

To visualize the weighting strategy proposed in Section 4.1, we also plot histograms of the class weights computed from ResNet and DMP in Figure 8(a)-(b) under the partial setting. The ResNet model assigns excessive weights to the outlier classes (in orange). DMP assists weight learning by aligning the domains partially and embedding the shared classes discriminatively.

Experiment of Shared Classes. To evaluate the robustness of methods under different shared class schemes, we varied the target class number from 10 to 31 on the Office 31 dataset while fixing the source class number as 31. The shared classes are extracted in alphabetical order, so the 10 shared classes here are different from the random selection in previous protocols [6], [7]. We compared the performance of DMP with ResNet baseline, DANN [18] and PADA [7]. Figure 8(c) shows the accuracies in task A→W.



(a) $\alpha = 0.1$ (b) $\alpha = 0.3$ (c) $\alpha = 0.5$

Fig. 9. Visualization of interpolations using 2-D t-SNE [55] on VisDA-2017 with different values of the interpolation parameter α. The features are colored by classes. '$\circ$' and '$\times$' represent the correctly and incorrectly classified samples, respectively. Best viewed in color.

TABLE 4
Class-wise recognition rates (%) of interpolations on VisDA-2017.

α	Plane	bcycl	bus	car	horse	knife	mcyle	person	plant	sktbrd	train	truck	Mean
0	98.0	85.0	86.0	83.0	94.0	84.0	95.0	76.0	95.0	69.0	91.0	48.0	83.7
0.1	98.0	87.3	87.7	83.8	94.0	88.3	96.0	75.9	96.0	72.0	91.6	50.4	85.1
0.3	99.9	91.1	90.2	90.3	94.8	93.0	98.7	80.1	96.9	81.3	93.4	53.4	88.6
0.5	100.0	92.1	90.9	92.4	95.3	95.8	99.8	81.8	97.7	84.0	98.2	56.1	90.4

We observed that the accuracy curves of all methods tended to decrease with the increasing number of of shared classes. DMP outperforms the other methods for all shared class cases. When the number of shared classes is small, the accuracies of DANN are lower than other methods as it aligns the target domain with the entire source domain and this results in negative transfer. In Figure 8(d), all curves seem more stable and DMP achieves the highest results in all cases.

Disccriminability of Interpolation. To further evaluate the discriminability of DMP, we computed unseen interpolations from 1200 randomly selected target images (100 images per class) as $\alpha * \mathbf{h}_i^t + (1-\alpha) * \mathbf{h}_j^t \ (i \neq j)$, where $\mathbf{h}_i$ and $\mathbf{h}_j$ are from the same class. The interpolation parameter α was selected from $\{0.1, 0.3, 0.5\}$. The quantitative results are shown in Table 4. We

TABLE 5
TOP: Recognition rates (%) under different sample size settings on ImageCLEF. **Bottom**: Time comparison on four experimental datasets. Batch and Epoch are abbreviated as *B* and *E* in units.

n_c^s \ n_c^t	I→P			P→I		
	10	30	50	10	30	50
10	80.0	78.4	77.6	89.5	90.3	90.1
30	83.5	80.7	79.4	91.4	91.8	91.2
50	84.3	81.4	80.7	91.4	92.1	92.5

Dataset	Office-31	Office-Home	ImageCLEF	VisDA-2017
Task	A→W	Ar→Cl	I→P	S→R
ResNet [53]	6.7*s*/*E*	60.6*s*/*E*	5.1*s*/*E*	0.453*s*/*B*
CDAN+E [23]	10.7*s*/*E*	72.0*s*/*E*	7.7*s*/*E*	1.238*s*/*B*
DMP	10.7*s*/*E*	70.8*s*/*E*	6.8*s*/*E*	0.767*s*/*B*

observed that the accuracies of interpolations are higher than the data before interpolation (i.e., $\alpha = 0$), and the accuracies of larger α are higher, which validates the discriminability of the interpolated embedding features. There are two possible reasons for this phenomenon. First, the fully-connected layer based classification rule guarantees that the predictions of interpolated embedding features are just the interpolations of predictions. Second, the interpolations with larger α are closer to the class centers, thus they are likely to have more discriminative ability. The qualitative analysis through t-SNE visualization of interpolations is presented in Figure 9. Note there are 50k+ interpolations, thus many points overlap. We observed that with the increase of α, the interpolations gradually move to the class centers. For many misclassified samples, which are scattered at the wrong side of decision boundary, their interpolations are pulled back to their ground-truth clusters and correctly classified. We also observed that the interpolations under all α settings are still inter-class separable and intra-class compact. This is mainly because the embedding features are discriminative.

Experiment on Sample Size and Time Comparison. As the performance of discriminative structure learning is in reference to the sample size, we chose $\{10, 30, 50\}$ samples per class for each domain in ImageCLEF to form several subsets. The proposed model was trained and tested on these subsets under the standard UDA protocol. The results are shown in top of Table 5, where $n_c^{s/t}$ represents the amount of samples per class on the source/target domain. We observed that a larger source sample size n_c^s provides a higher accuracy since more ground-truth labels are used to learn discriminative structure. For some cases in task **P→I**, a larger n_c^t also increases the precision. This demonstrates the target predictive information is also helpful in mining the discriminative information. The accuracies at $(n_c^s, n_c^t) = (30, 30)$ are slightly lower than that at $(50, 50)$, it indicates that DMP is still effective when the sample size is small. This is because DMP enhances the discriminability of the target features by simultaneously using the source labels and target predictive information.

We also evaluated the efficiency of DMP by comparing the training time. Results in the bottom of Table 5 show that DMP is faster than CDAN and slightly slower than ResNet. It validates that the time efficiency is mainly decided by the architecture of the baseline network and the post alignment network (e.g., adversarial alignment). As DMP only adds three low-dimensional fully connected layers for discriminative alignment, it does not introduce many parameters relative to the backbone network and adversarial alignment while improving the recognition rates significantly.

6 Conclusion

In this paper, we proposed a discriminative manifold propagation method for both vanilla and partial UDA problems. Both transferability and discriminability are simultaneously reached by the manifold alignment and discriminative embedding. To optimize the structure of the target domain, the source labels and target predictive information were encoded probabilistically and transductively into the discriminant criterion. A global discriminative structure was approximated via the pre-built prototypes. The theoretical error bounds, which are guaranteed to find the optimal dimensions for the Grassmann and affine Grassmann manifolds during the alignment, were derived. Numerical simulation and extensive comparisons demonstrated the effectiveness of the proposed method.

References

[1] J. Hu, J. Lu, and Y.-P. Tan, "Deep transfer metric learning," in CVPR, 2015, pp. 325–333.
[2] M. Long, Y. Cao, Z. Cao, J. Wang, and M. I. Jordan, "Transferable representation learning with deep adaptation networks," IEEE TPAMI, vol. 35, no. 8, pp. 1798–1828, 2013.
[3] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," IEEE TNNLS, vol. 26, no. 5, pp. 1019–1034, 2014.
[4] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," IEEE TNN, vol. 22, no. 2, pp. 199–210, 2010.
[5] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," Machine Learning, vol. 79, no. 1-2, pp. 151–175, 2010.
[6] J. Zhang, Z. Ding, W. Li, and P. Ogunbona, "Importance weighted adversarial nets for partial domain adaptation," in CVPR, 2018, pp. 8156–8164.
[7] Z. Cao, L. Ma, M. Long, and J. Wang, "Partial adversarial domain adaptation," in ECCV, 2018, pp. 135–150.
[8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein et al., "Imagenet large scale visual recognition challenge," IJCV, vol. 115, no. 3, pp. 211–252, 2015.
[9] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007, http://www.vision.caltech.edu/Image_Datasets/Caltech256/.
[10] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes (VOC) challenge," IJCV, vol. 88, no. 2, pp. 303–338, 2010.
[11] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in ICCV, 2013, pp. 2200–2207.
[12] C. X. Ren, J. Feng, D.-Q. Dai, and S. Yan, "Heterogeneous domain adaptation via covariance structured feature translators," IEEE TCYB, 2019, Accepted.
[13] J. Moon, D. Das, and C. G. Lee, "Multi-step online unsupervised domain adaptation," in ICASSP. IEEE, 2020, pp. 41 172–41 576.
[14] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in ICCV, 2011, pp. 999–1006.
[15] S. Shekhar, V. M. Patel, H. Van Nguyen, and R. Chellappa, "Coupled projections for adaptation of dictionaries," IEEE TIP, vol. 24, no. 10, pp. 2941–2954, 2015.
[16] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in NeurIPS, 2014, pp. 3320–3328.
[17] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in ECCV, 2016, pp. 443–450.
[18] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," JMLR, vol. 17, no. 1, pp. 2096–2030, 2016.
[19] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in CVPR, 2018, pp. 8503–8512.
[20] P. Ge, C. X. Ren, D. Q. Dai, J. Feng, and S. Yan, "Dual adversarial autoencoders for clustering," IEEE TNNLS, 2019, Accepted.
[21] Z. Cao, K. You, M. Long, J. Wang, and Q. Yang, "Learning to transfer examples for partial domain adaptation," in CVPR, 2019, pp. 2985–2994.
[22] Z. Cao, M. Long, J. Wang, and M. I. Jordan, "Partial transfer learning with selective adversarial networks," in CVPR, 2018, pp. 2724–2732.

[23] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in NeurIPS, 2018, pp. 1640–1650.

[24] C. X. Ren, X. L. Xu, and H. Yan, "Generalized conditional domain adaptation: A causal perspective with low-rank translators," IEEE TCYB, vol. 50, no. 2, pp. 821–834, 2020.

[25] P. O. Pinheiro, "Unsupervised domain adaptation with similarity learning," in CVPR, 2018, pp. 8004–8013.

[26] D. Das and C. G. Lee, "Graph matching and pseudo-label guided deep unsupervised domain adaptation," in ICANN. Springer, 2018, pp. 342–352.

[27] X. Chen, S. Wang, M. Long, and J. Wang, "Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation," in ICML, 2019, pp. 1081–1090.

[28] R. Xu, G. Li, J. Yang, and L. Lin, "Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation," in ICCV, 2019, pp. 1426–1435.

[29] Z. Ding and Y. Fu, "Deep transfer low-rank coding for cross-domain learning." IEEE TNNLS, vol. 30, no. 6, pp. 1768–1779, 2019.

[30] C. Chen, Z. Chen, B. Jiang, and X. Jin, "Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation," in AAAI, 2019.

[31] Y. Pan, T. Yao, Y. Li, Y. Wang, C.-W. Ngo, and T. Mei, "Transferrable prototypical networks for unsupervised domain adaptation," in CVPR, 2019, pp. 2239–2247.

[32] Z. Huang, R. Wang, S. Shan, L. Van Gool, and X. Chen, "Cross euclidean-to-riemannian metric learning with application to face recognition from video," IEEE TPAMI, vol. 40, no. 12, pp. 2827–2840, 2017.

[33] Y. W. Luo, C. X. Ren, P. Ge, K. kun Huang, and Y.-F. Yu, "Unsupervised domain adaptation via discriminative manifold embedding and alignment," in AAAI, 2020.

[34] C. X. Ren, P. Ge, D. Q. Dai, and H. Yan, "Learning kernel for conditional moment-matching discrepancy-based image classification," IEEE TCYB, 2019, Accepted.

[35] C. X. Ren, B. Liang, P. Ge, Y. Zhai, and Z. Lei, "Domain adaptive person re-identification via camera style generation and label propagation," IEEE TIFS, vol. 15, pp. 1290–1302, 2020.

[36] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in ICML, vol. 70, 2017, pp. 2208–2217.

[37] W.-G. Chang, T. You, S. Seo, S. Kwak, and B. Han, "Domain-specific batch normalization for unsupervised domain adaptation," in CVPR, 2019, pp. 7354–7362.

[38] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in CVPR, 2018, pp. 3723–3732.

[39] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," IEEE TPAMI, no. 1, pp. 40–51, 2007.

[40] C. X. Ren, Y. W. Luo, X. L. Xu, D. Q. Dai, and H. Yan, "Discriminative residual analysis for image set classification with posture and age variations," IEEE TIP, 2019, Accepted.

[41] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," SIAM Journal on Matrix Analysis and Applications, vol. 20, no. 2, pp. 303–353, 1998.

[42] L. Zwald and G. Blanchard, "On the convergence of eigenspaces in kernel principal component analysis," in NeurIPS, 2006, pp. 1649–1656.

[43] L.-H. Lim, K. Sze-Wai Wong, and K. Ye, "Numerical algorithms on the affine grassmannian," SIAM Journal on Matrix Analysis and Applications, vol. 40, no. 2, pp. 371–393, 2019.

[44] X. Pennec, P. Fillard, and N. Ayache, "A riemannian framework for tensor computing," IJCV, vol. 66, no. 1, pp. 41–66, 2006.

[45] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Geometric means in a novel vector space structure on symmetric positive-definite matrices," SIAM Journal on Matrix Analysis and Applications, vol. 29, no. 1, pp. 328–347, 2007.

[46] T. Papadopoulo and M. I. Lourakis, "Estimating the jacobian of the singular value decomposition: Theory and applications," in ECCV, 2000, pp. 554–570.

[47] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in ECCV, 2010, pp. 213–226.

[48] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in CVPR, 2017, pp. 5018–5027.

[49] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein et al., "Imagenet large scale visual recognition challenge," IJCV, vol. 115, no. 3, pp. 211–252, 2015.

[50] A. Bergamo and L. Torresani, "Exploiting weakly-labeled web images to improve object classification: a domain adaptation approach," in NeurIPS, 2010, pp. 181–189.

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

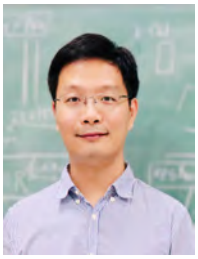
[52] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in ECCV, 2014, pp. 740–755.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in CVPR, 2016, pp. 770–778.

[54] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht, "Sliced wasserstein discrepancy for unsupervised domain adaptation," in CVPR, 2019, pp. 10 285–10 295.

[55] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," JMLR, vol. 9, no. Nov, pp. 2579–2605, 2008.

You-Wei Luo received the B.S. degree in statistics from China University of Mining and Technology, Xuzhou, China, in 2018. He is currently pursuing the Ph.D. degree with the School of Mathematics, Sun Yat-sen University, Guangzhou, China. His research interests include image processing, manifold learning and transfer learning.



Chuan-Xian Ren received the PhD degree from Sun Yat-Sen University, Guangzhou, China, in 2010. He is currently Associate professor of the School of Mathematics, Sun Yat-Sen University. His research interests include image processing, pattern recognition and machine learning.

Dao-Qing Dai (M'07-SM'19) received the Ph.D. degree in mathematics from Wuhan University, Wuhan, China, in 1990. From 1998 to 1999, he was an Alexander von Humboldt Research Fellow with Free University, Berlin, Germany. He is currently a Professor with the School of Mathematics, Sun Yat-Sen University, China. He has authored or co-authored over 100 refereed technical papers. His current research interests include image processing, wavelet analysis, and pattern recognition.

Hong Yan (S'88-M'89-SM'93-F'06) received his PhD degree from Yale University. He was Professor of Imaging Science at the University of Sydney and currently is Chair Professor of Computer Engineering and Wong Chung Hong Professor of Data Engineering at City University of Hong Kong. Professor Yan's research interests include image processing, pattern recognition, and bioinformatics. He has over 600 journal and conference publications in these areas. Professor Yan is an IEEE Fellow and IAPR Fellow. He received the 2016 Norbert Wiener Award from the IEEE Systems, Man and Cybernetics Society for contributions to image and biomolecular pattern recognition techniques. He is a member of the European Academy of Sciences and Arts.