Reliable Domain Adaptation with Classifiers Competition

Jingru Fu[0000−0003−4175−395X] and Lei Zhang*[0000−0002−5305−8543]

School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China
{jrfu,leizhang}@cqu.edu.cn

Abstract. Unsupervised domain adaptation (UDA) aims to transfer labeled source domain knowledge to the unlabeled target domain. Previous methods usually solve it by minimizing joint distribution divergence and obtaining the pseudo target labels via source classifier. However, those methods ignore that the source classifier always misclassifies partial target data and the prediction bias seriously deteriorates adaptation performance. It remains an open issue but ubiquitous in UDA, and to alleviate this issue, a Reliable Domain Adaptation (RDA) method is proposed in this paper. Specifically, we propose double task-classifiers and dual domain-specific projections to align those easily misclassified and unreliable target samples into reliable ones in an adversarial manner. In addition, the domain shift of both manifold and category space is reduced in the projection learning step. Extensive experiments on various databases demonstrate the superiority of RDA over state-of-the-art unsupervised domain adaptation methods.

Keywords: Domain Adaptation · Source Domain · Target Domain.

1 Introduction

Many algorithms in computer vision areas are based on a fundamental assumption that the training and test data are drawn from the same distribution [13]. However, this assumption generally does not hold in many real-world scenarios such that the trained model often does not fit the test data, since training and test images are obtained under very different conditions (e.g., different camera device parameters, varying illuminations, and viewpoints, etc.) [25].

To address this issue, domain adaptation was proposed to exploit the rich labeled source domain data to facilitate the learning of a different but semantic related unlabeled target domain [7, 11, 19]. This is a *unsupervised domain adaptation* (UDA) problem. A common strategy to handle such unsupervised scenario is to align the distributions across the source and target domain. Maximum Mean Discrepancy (MMD) is a favorite principle to measure the discrepancy between two distributions [15]. Pan et al proposed to learn a transferred subspace across domain by using MMD to measure the marginal distribution of domains [18]. However, the source label information with rich semantics is ignored. To solve it, Long et al proposed to jointly minimize both the marginal and conditional distributions [15]. Since there is no target label, an iterative pseudo target label updating strategy was used to compute the conditional distribution. Many works

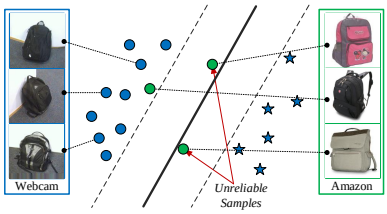


Fig. 1. The motivation of the proposed method. Source samples and target samples are denoted in blue and green, respectively. The classifier (solid line) is trained on the source samples (2 classes with different symbols for simplification). Target samples with large domain discrepancy have low classification confidence (within two dotted lines), which we define as unreliable samples.

[2, 9, 30, 31] have experimentally demonstrated that the pseudo target labels can significantly boost the performance of UDAs. However, none of them take into account the misclassified target samples and the prediction bias, which we view as *unreliable* target samples. In fact, the unreliable target samples deteriorate the clustering performance of adaptation, due to that incorrect target labels cannot well account for the class distribution discrepancy. As shown in Figure 1, our motivation is inspired by the fact that an easily misclassified sample generally closes to decision boundary and thus holds low confidence for a classifier. Apart from that, most of these methods assume there exists a common subspace between domains, which usually fails to extract domain-specific information from each domain.

To alleviate the pseudo target label prediction bias problem and preserve domain-specific information, we propose an RDA model composed of double task-classifiers and dual projections. The double task-classifiers are used to discover those unreliable target samples. Then two domain-specific projections are used to seek a reliable feature embedding that transforms those unreliable samples into reliable ones, in the meantime, they are forced to close to each other in order to reduce the distance across domains in the *Grassmann* manifold space [1]. Note that these two steps are trained in an adversarial manner.

Toward this end, we propose a **Reliable Domain Adaptation (RDA)** method for unsupervised domain adaptation, by discovering unreliable target samples with double classifiers and transforming the samples into new feature spaces, in an adversarial manner. We summarize the contributions of this paper as follows:

- We propose a RDA model to discover the unreliable target samples (i.e., easily-biased samples) via double task-classifiers and further transform the unreliable samples into reliable feature embedding, which effectively alleviates the clustering bias resulted from the incorrect pseudo target labels.
- We propose the dual subspace projections to reduce the discrepancy between domains in manifold space and preserve domain-specific information across domains.
- Extensive experiments on challenging benchmark datasets demonstrate that our method achieves the best performance by comparing to state-of-the-arts including shallow and deep learning methods.

2 Related Work

In this section, some related works are divided into three aspects:

Subspace-driven methods. Subspace alignment (SA) [8] aims at learning a linear mapping for aligning subspaces spanned by eigenvectors using principal component analysis (PCA) across domains. Geodesic flow kernel (GFK) [1] characterized the changes of geometric and statistical properties across domains by integrating numerous subspaces. CORAL [24] alleviated the domain shift by aligning the second-order statistics (e.g., covariance) between two domains. Those methods aligned the statistical features over domains in manifold space, where the global property of domains is well represented. The tolerance of noise is then improved. However, they ignored the distribution alignment.

Data-driven methods. Transfer component analysis (TCA) [18] learned the transfer components between domains using Maximum Mean Discrepancy (MMD). Domain invariant projection (DIP) [2] proposed to construct the MMD in the manifold space. Statistically invariant embedding (SIE) [3] used Hellinger distance on statistical manifolds to approximate the geodesic distance. Transfer Joint Matching (TJM) [16] matched the feature representations by re-weighting the instances. However, none of them utilized the semantic information that is beneficial to the discrimination of the model. So, joint distribution alignment (JDA) [15] proposed to reduce both the marginal distribution and conditional distribution measured by using MMD and pseudo target labels. However, due to the clustering bias, the predicted pseudo target labels are not reliable.

Adversarial learning methods. Generative adversarial networks (GAN) [10] was the first proposal for adversarial learning. It can be seen as a distribution matching method, for matching the generated data (i.e. generator) with the target data, supervised by a domain classifier (i.e. discriminator). Tzeng et al [26, 27] proposed adversarial domain adaptation models by enhancing the domain feature confusion, supervised by a domain classifier. Motivated by the theory proposed in [5], Saito et al [22] considered the decision boundaries between classes for the first time and aimed at aligning the distribution between classes. These methods are structured based on convolutional neural network (CNN), that well accelerates the discriminative feature representation. Our approach is based on a statistical learning framework that also uses the adversarial idea to achieve reliable unsupervised domain adaptation.

3 Reliable Domain Adaptation

In this section, we introduce the proposed method in detail. First, the problem and notations are defined, then the overall method is presented, and details of model and solution are finally introduced. Note that our approach is a statistical learning framework, *not* a CNN-based deep network.

3.1 Problem Definition

Given a labeled source domain $\mathcal{D}_s = \{(x_i, y_i)\}_{i=1}^{n_s}, x_i \in \mathbb{R}^D$ and an unlabeled target domain $\mathcal{D}_t = \{(x_j)\}_{j=1}^{n_t}, x_j \in \mathbb{R}^D$, where n_s and n_t indicate the number of samples in

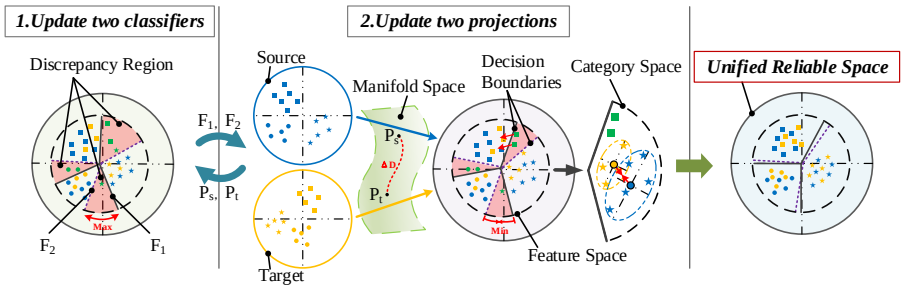


Fig. 2. Overview of the RDA approach. Two classifiers F_1 and F_2 (dotted line and solid line) are presented to discover the unreliable target samples by maximizing the discrepancy region (shadow region). Dual projections P_s and P_t aim to seek new reliable features by minimizing the shadow region and, simultaneously, align domains in both manifold and category spaces. The ultimate goal is to find a reliable space where clustering of the same class across domains is as good as possible. Note that solid circles define the ranges of subspaces, dotted circles define the ranges of features' distributions.

source and target domain, and D is the dimension of the original samples. We assume the label spaces between domains are the same, i.e. $\mathcal{Y}_s = \mathcal{Y}_t$, and the label space $\mathcal{Y}$ is a C-cardinality label set. $X_s \in \mathbb{R}^{D\times n_s}$ and $X_t \in \mathbb{R}^{D\times n_t}$ are domain-specific datasets drawn from distribution $\mathcal{P}_s(\cdot)$ and $\mathcal{P}_t(\cdot)$, respectively.

3.2 Model Formulation

Clearly in Fig 2, two fundamental steps are included in RDA in an adversarial manner:

Step 1. *Train two classifiers*. We introduce two *task-classifiers* aiming at discovering the unreliable target samples in this step, which are hard to be classified by the source-classifier due to distribution mismatch. Note that two *task-classifiers* are initially trained on source data, and are forced to classify source data as accurately as possible during the whole training processing. The double task-classifiers are trained by maximizing the discrepancy region, so that the target samples which are close to classifier boundaries (unreliable samples, green sample in Figure 2) can be discovered as much as possible.

We suppose the input features of classifiers to be $z_s = g_s(x_s)$ for source samples and $z_t = g_t(x_t)$ for target samples, respectively. $g_s(\cdot)$ and $g_t(\cdot)$ indicate the function of dual *domain-specific projections*. We train both classifiers $f_1(\cdot)$ and $f_2(\cdot)$ to classify the source samples as correct as possible and, simultaneously, maximize the discrepancy over classification outputs. The objective function of the first step is as follows:

$$\min_{f_1,f_2} \mathcal{L}_f(z_s, y_s) - \lambda \mathcal{L}_{adv}(z_t) \tag{1}$$

where $\mathcal{L}_f$ represents the classifiers' loss function for source features, $\mathcal{L}_{adv}$ represents the adversarial loss formulated as the discrepancy between two classifiers' outputs on target samples, and λ is the trade-off parameter.

Step 2. *Train dual projections*. The ultima goal of learning is to obtain dual generators that can generate reliable transferred features. We firstly propose to align conditional feature distribution over domains to ensure that the same classes of both domains can be clustered as well as possible and then introduce the pseudo label prediction strategy. The dual *domain-specific projections* are used to map data of both domains, respectively. The involved clustering bias is solved by minimizing the discrepancy in this step to align such unreliable target features. We define the pseudo target labels as $\hat{y}_t$. The objective function of the second step can be formulated as:

$$\min_{g_s,g_t} \mathcal{L}_{feat}(x_s, x_t, y_s, \hat{y}_t) + \alpha\mathcal{L}_{sub} + \lambda\mathcal{L}_{adv}(x_t) \tag{2}$$

where $\mathcal{L}_{feat}$ and $\mathcal{L}_{sub}$ represent the feature-align loss and the subspace-align loss, respectively, α is trade-off parameter. The subspace-align and adversarial loss are treated as the regularization terms in our model.

These two steps are updated alternately, and ultimately, a unified reliable transferred space can be obtained, where samples of the same category in two domains are clustered. In the next subsection, the technical details of RDA are presented.

Details in Step 1 This subsection explains the specific implementation of Step 1, which aims to train double *task-classifiers*. We formulate double classifiers as the coefficient vector $F \in \mathbb{R}^{d\times C}$, and dual projections as $P_s \in \mathbb{R}^{D\times d}$ and $P_t \in \mathbb{R}^{D\times d}$ according to the representer theorem [4], d donates the dimension of features.

1) Classification loss $\mathcal{L}_f$ on source domain. The loss function of the source classifiers is formulated as a regularized least-square loss:

$$\begin{aligned}\mathcal{L}_f(z_s, y_s) &= \sum_{i=1}^{2}(\sum_{j=1}^{n_s}(f_i(z_s^j) - y_s^j)^2 + \eta \left\|f_i\right\|^2) \\ &= \sum_{i=1}^{2}(\left\|Z_s^T F_i - Y_s\right\|_F^2 + \eta \left\|F_i\right\|_F^2),\end{aligned} \tag{3}$$

where $Z_s = [z_s^1, z_s^2, ..., z_s^{n_s}] \in \mathbb{R}^{d\times n_s}$ (Note that $Z_s = P_s^T X_s$) is the source domain feature set, η is trade-off parameter. $\left\|M\right\|_F = \sqrt{tr(M^T M)}$ is the Frobenius norm of matrix M, $tr(\cdot)$ is trace operator. We define the constructed source label matrix as $Y_s = [y_s^1, y_s^2, ..., y_s^{n_s}]^T \in \{-1, 1\}^{n_s\times C}$, and $y_s^i(c) = 1$ means that the i-th source sample is associated with the c-th class.

2) Adversarial loss $\mathcal{L}_{adv}$ on target domain. The unreliable target features are samples which close to classifier boundaries, inspired by a CNN-based deep method [22], which utilizes classifiers' difference to represent classifier boundaries, we also formal the outputs' discrepancy as adversarial loss term which can be formulated as:

$$\begin{aligned}\mathcal{L}_{adv}(z_t) &= \sum_{i=1}^{n_t}(f_1(z_t^i) - f_2(z_t^i))^2 \\ &= \left\|Z_t^T F_1 - Z_t^T F_2\right\|_F^2,\end{aligned} \tag{4}$$

where $Z_t = [z_t^1, z_t^2, ..., z_t^{n_t}] \in \mathbb{R}^{d \times n_t}$ ($Z_t = P_t^T X_t$) is target domain feature set. From Figure 1 we find that when we force to increase the difference between the two task-classifiers' outputs, target samples that are closing to the decision boundary can fall into the region between the two classifiers' decision boundaries and can then be found.

Details in Step 2 This subsection explains the specific implementation of Step 2.

1) Feature-align loss $\mathcal{L}_{feat}$. The proposed feature alignment loss aims at clustering the same class of the source and target domain in category space, such that the disparity between the conditional distributions across domains is reduced. The semantic guided MMD alike feature alignment loss is used to measure the dissimilarity of conditional distributions [15, 28, 30]. The pseudo target label is represented as $\hat{y}_t$. Then the feature-align loss is as:

$$\mathcal{L}_{feat} = \sum_{c=1}^{C} \| \frac{1}{n_s^{(c)}} \sum_{x_i \in \mathcal{D}_s^{(c)}} P_s^T x_i - \frac{1}{n_t^{(c)}} \sum_{x_j \in \mathcal{D}_t^{(c)}} P_t^T x_j \|^2, \tag{5}$$

where $\mathcal{D}_s^{(c)} = \{x_i | x_i \in \mathcal{D}_s | y_s(x_i) = c\}$ ($\mathcal{D}_t^{(c)} = \{x_j | x_j \in \mathcal{D}_s | \hat{y}_t(x_j) = c\}$) is the set of source (target) samples (a total of $n_s^{(c)}$ ($n_t^{(c)}$) samples) with respect to class c, $y_s(x_i)$ is the true source label of sample x_i.

The feature-align loss aims to reduce the class-wise distance between domains. As illustrated in Figure 2, the data points with the blue circle and yellow circle represent the class-wise center of target and source domain in the *category space* of the F_1 classifier.

2) Subspace-align loss $\mathcal{L}_{sub}$. Similar to [8], our goal is to decrease the distance (i.e. $\triangle D$ in Figure 2) between two domain-specific projections. [30] confirmed that the shift of subspace geometries can be reduced in this way. For better non-parameter learning, instead of learning an additional mapping function, we propose to minimize the following smooth subspace-align loss directly:

$$\mathcal{L}_{sub} = \|P_s - P_t\|_F^2, \tag{6}$$

3) Adversarial loss. For correcting the unreliable target samples found in Step 1, we expect to reduce the discrepancy in an adversarial way. Note that the dual projections (P_s, P_t) instead of classifiers (F_1, F_2) are trained to minimize the classifiers' difference. The following adversarial loss function is minimized:

$$\begin{aligned} \mathcal{L}_{adv}(x_t) &= \sum_{i=1}^{n_t} (f_1(g_t(x_t^i)) - f_2(g_t(x_t^i)))^2 \\ &= \left\| (P_t^T X_t)^T F_1 - (P_t^T X_t)^T F_2 \right\|_F^2, \end{aligned} \tag{7}$$

where the two task classifiers F_1 and F_2 have been solved in Step 1.

A deep adaptation method MCD_DA that is relevant to this paper was proposed by Saito et al [22], in which two classifiers are considered for solving UDA. Here, we briefly highlight the main differences between MCD_DA and RDA as following: 1) MCD_DA just tries to align source-unsupported target samples without considering to align conditional feature distribution, while RDA takes it into consideration. 2)

Algorithm 1 Reliable Domain Adaptation

Input: Data and source labels: X_s, X_t, y_s; Parameters: $d = 20$, $\eta = 1$, α, λ, T.
Output: Projection matrices: P_s and P_t; Predicted target labels: $\hat{y}_t$.

1: Initialize P_s and P_t using existing method. e.g. SA [8], PCA, etc.
   While iteration $t < T$ **do**
2: Update $\hat{y}_t$ using a base classifier, there is
   $\hat{y}_t = classifier(P_s^T X_s, y_s, P_t^T X_t)$.
3: Fix P_s and P_t, and update F_1 and F_2 by solving (8).
4: Fix F_1 and F_2, and update P_s and P_t by solving (9), calculate $Z_s = P_s^T X_s$, $Z_t = P_t^T X_t$.
5: $t = t + 1$.
   End while
6: **return** P_s, P_t, $\hat{y}_t$

MCD_DA only trains one shared generator between domains, but RDA considers the domain-specific generators (projections), and they are beneficial to reduce domain shifts in the manifold space. 3) MCD_DA is a deep adaptation method while RDA is a statistical learning framework. The necessity and effectiveness of the first two items are verified in the *ablation analysis* part.

Overall Model of RDA The ultimate model of RDA consist of two objectives:

$$\min_{F_1, F_2} \mathcal{L}_f(z_s, y_s) - \lambda \mathcal{L}_{adv}(z_t) \tag{8}$$

$$\min_{P_s, P_t} \mathcal{L}_{feat}(x_s, x_t, y_s, \hat{y}_t) + \alpha \mathcal{L}_{sub} + \lambda \mathcal{L}_{adv}(x_t) \tag{9}$$

where all terms in the minimax optimization model have been presented above.

In the optimization of the RDA model, we adopt the alternating optimization strategy, i.e., fix the projections in training the two task-classifiers and fix the task-classifiers in training the two projections. The predicted pseudo-labels of target data are updated in each loop. For each step, ADMM algorithm is considered [6]. The optimization of RDA is summarised in Algorithm 1.

4 Experiment

A number of experiments are conducted to evaluate the performance of RDA for unsupervised scenarios, which is closer to real-world applications. We compare our methods with state-of-the-art: 1) *Subspace-driven methods*: SA [8], GFK [1] and CORAL [24]; 2) *Data-driven methods*: JDA [15], DIP [2], JGSA [30] and TJM [16]; 3) *Adversarial learning methods*: Deep Domain Confusion (DDC) [27]; 4) *Deep transfer learning methods*: Domain Adaptation Networks (DAN) [14] and Residual Transfer Network (RTN) [17]. Notice that, it is unfair for RDA to compare directly against the deep DA methods, since RDA is a statistical shallow learning method. Therefore, deep features extracted using a pre-trained CNN are fed into RDA, and expect to further reduce the discrepancy of deep representation.

Table 1. Recognition accuracies (%) on Office+Caltech10 dataset with the deep feature from VGG-VD-16 model. * denotes deep transfer learning methods. **Red**: ranks the 1^{st}; **Blue**: ranks the 2^{nd}; **Green**: ranks the 3^{rd}.

Task	Raw	SA	JDA	GFK	JGSA	CORAL	DIP	TJM	DDC*	DAN*	RTN*	RDA
C→A	91.5	93.2	93.7	93.6	**94.2**	91.6	93.3	93.9	91.9	92.0	**94.4**	**96.0**
C→W	83.7	86.4	**94.6**	86.8	93.3	78.9	86.2	92.0	85.4	90.3	**96.6**	**99.0**
C→D	89.9	**95.0**	93.2	91.0	**94.4**	87.6	91.4	90.8	88.1	90.5	92.9	**94.3**
A→C	81.7	77.1	**90.1**	85.3	87.2	80.1	86.0	86.4	85.0	85.1	**88.5**	**93.2**
A→W	74.8	80.4	91.5	85.8	**95.7**	75.7	74.1	87.3	86.1	93.8	**97.0**	**98.6**
A→D	77.2	89.6	91.3	85.5	**94.1**	76.2	83.4	89.9	89.0	92.4	**94.6**	**96.8**
W→C	77.3	77.9	**86.7**	81.3	82.3	77.6	81.2	81.4	78.0	84.3	**88.4**	**92.6**
W→A	85.5	87.3	93.8	90.2	**94.9**	90.7	88.4	91.1	84.9	92.1	**93.1**	**96.0**
W→D	**99.0**	98.0	96.1	98.0	96.1	98.0	98.0	97.6	**100**	**100**	**100**	**99.4**
D→C	75.0	78.6	**84.8**	82.3	**85.2**	73.1	81.0	81.8	81.1	82.4	84.3	**91.3**
D→A	83.6	83.8	91.7	90.8	**93.8**	84.5	90.0	91.4	89.5	92.0	**95.5**	**94.5**
D→W	95.8	97.0	89.2	97.3	96.4	94.9	95.2	96.8	98.2	**99.0**	**98.8**	**99.7**
Average	84.6	87.0	91.4	89.0	**92.3**	84.1	87.4	90.0	88.2	91.2	**93.7**	**96.0**

4.1 Data Preparation

In experiments, five different visual benchmark datasets are exploited and tested.

1) **Office-10+Caltech-10 (4DA)** [1]: The Office data [21] contains three real-world object domains, including **A**mazon, **W**ebcam and **D**SLR. **C**altech-256 [12] is a standard database for object recognition. 4DA is formulated with 10 shared categories between Office and Caltech datasets. Two kinds of features, i.e. hand-crafted SURF feature and deep CNN features, are used. *First*, the SURF features [1] that are encoded with 800-dimension BoW features are used as the shallow feature. *Second*, the features extracted from a deep model (the FC7 activations of VGG-VD-16 model) [23] are exploited as the deep feature. By randomly selecting two different domains as the source and target domain, a total of 12 cross-domain tasks are constructed.

2) **MSRC+VOC2007** [16]: Six shared semantic classes from both datasets are formulated, and 1,269 images in MSRC and 1,530 images in VOC2007 are selected for domain adaptation. The 128-dimensional dense SIFT (DSIFT) features were extracted using the VLFeat open-source software package, and K-means clustering was used to obtain the 240-dimensional codebook. Following the experimental setting as [29], two cross-domain tasks are constructed: M vs. V and V vs. M.

3) **COIL20** [20]: Dataset contains 20 objects with 1440 gray scale images. Each image has 32×32 pixels and 256 gray levels per pixel. In experiments, the dataset is divided into two subsets COIL1 and COIL2 by following [29]. Specifically, the COIL1 (C1) and COIL2 (C2) contain the images taken in the directions of [0°, 85°] $\bigcup$ [180°, 265°] and [90°, 175°] $\bigcup$ [270°, 355°], respectively.

4.2 Experimental Setting

We strictly follow the experimental configuration for UDA as [1, 16, 29]. SVM is trained on the labeled source data for generating pseudo-target-labels. Three trade-off parame-

Table 2. Recognition accuracies (%) on Office+Caltech10 dataset with SURF features.

Task	Raw	SA	JDA	GFK	JGSA	CORAL	DIP	TJM	RDA
C→A	50.1	54.4	**59.8**	**56.6**	55.1	45.9	56.4	54.4	**59.4**
C→W	43.1	45.8	**50.1**	48.1	49.7	37.8	**51.2**	44.0	**57.6**
C→D	**47.8**	40.9	44.1	42.9	46.0	31.8	**46.9**	38.4	**51.6**
A→C	42.8	**44.8**	**44.9**	44.3	40.8	37.1	41.4	42.4	**49.2**
A→W	37.0	44.1	**47.0**	42.7	**59.0**	37.9	44.8	39.5	**45.1**
A→D	37.2	37.7	44.2	39.9	**49.4**	38.5	**47.8**	45.6	**50.3**
W→C	29.5	32.3	29.8	32.0	29.7	**32.5**	30.0	**33.3**	**40.9**
W→A	34.2	**43.3**	**42.0**	38.3	34.6	39.4	33.8	39.5	**45.6**
W→D	80.6	70.3	**86.3**	78.7	78.5	**80.9**	79.6	**83.6**	78.3
D→C	30.1	31.1	**34.4**	30.8	30.2	27.8	29.3	**32.3**	**36.6**
D→A	32.1	**40.8**	**44.6**	40.4	39.0	31.9	31.6	37.1	**46.5**
D→W	72.2	74.4	**83.3**	80.3	75.1	69.4	67.5	**83.7**	**80.7**
Average	44.7	46.7	**50.9**	47.9	**48.9**	42.6	46.7	47.8	**53.5**

Table 3. Recognition accuracies (%) on MSVC-VOC2007 and COIL20 datasets. * denotes the results of GFK based on 1-nearest neighbor (1-NN) classifier.

Task	Raw	SA	JDA	GFK*	JGSA	CORAL	TJM	RDA
M→V	37.1	31.8	38.2	28.8	**38.7**	33.9	**38.3**	**39.7**
V→M	**55.5**	46.0	**59.3**	48.9	49.3	54.1	54.1	**62.3**
C1→C2	82.7	**86.7**	**88.7**	72.5	85.1	84.9	83.1	**93.5**
C2→C1	84.0	**90.6**	**93.1**	74.2	83.9	87.9	88.5	**91.8**
Average	64.8	63.8	**69.8**	56.1	64.3	65.2	**66.0**	**71.8**

ters: α, λ and η are involved in the proposed method. We set $\eta = 1$ for all experiments to simplify the tuning steps. For fairness, α and λ are only tuned from the parameter set [0.1, 1, 10]. We empirically set the subspace dimension $d = 20$ for all experiments.

4.3 Experimental Results

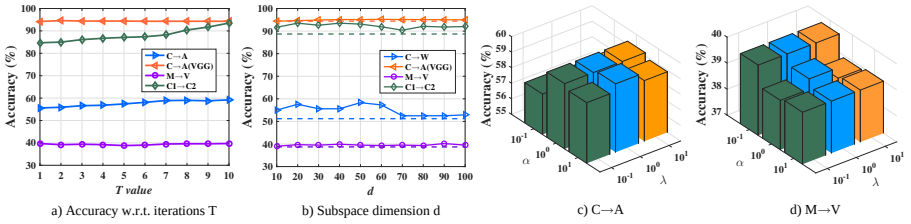
The recognition accuracies of RDA are shown in Tables 1, 2, and 3, respectively. From those results, we observe that RDA outperforms the state-of-the-art in a number of cross-domain tasks (21/28 tasks). Moreover, we achieve at least the second-best performance except for the three tasks: C→D, A→W and D→W. The average classification accuracy of RDA on the total 28 tasks is **74.3%**, which is **3.3%** higher than the state-of-the-art JDA (71.0%). Notice that, the results are obtained from a number of benchmark visual datasets, which can effectively demonstrate that RDA is capable of reducing the domain shift for UDA.

Second, for local comparisons, RDA generally outperforms the subspace-driven methods (i.e., SA, GFK and CORAL) and data-driven methods (i.e., DIP, JDA and TJM). The reason is that those methods do not reduce the cross-domain discrepancy in both category space and domain-specific subspace. Our approach considers both aspects. Above all, compared to the methods using the pseudo label strategy (e.g., JDA and

JGSA), our RDA alleviates the clustering bias resulted from unreliable pseudo target labels and guarantees the reliability.

Third, compared with shallow features (e.g., SURF features), deep features obtain significantly better results for all models. The proposed RDA shows significant improvement (**3.7%**) on average compared to the best shallow transfer method (i.e. JGSA), and **2.3%** comparing to the best deep transfer method (i.e. RTN) as shown in Table 1. The comparison shows that the proposed RDA, as a shallow learning method, is more effective but reliable.

4.4 Model Analysis and Discussion



a) Accuracy w.r.t. iterations T b) Subspace dimension d c) C→A d) M→V

Fig. 3. Convergence and parameter sensitivity analysis of RDA model on several datasets. Note that the dashed lines in b) show the best baseline results.

Parameter Sensitivity *First*, the recognition performance on several datasets with regard to the iterations T is shown in Figure 3 a). We set the maximum number of iteration T=10 in all experiments. From the results, it can be observed that classification performance rises slowly and tends to be smooth on some tasks (e.g., $C \rightarrow A(VGG)$ and $M \rightarrow V$), but shows a clear upward trend on other tasks (e.g., $C \rightarrow A$ and $C1 \rightarrow C2$). Empirically, we are able to get relatively good results with T=10. *Second*, we investigate the sensitivity of subspace dimension d with a wide range of $d \in \{10, 20, ..., 100\}$ to illustrate the relationship between d and the classification accuracy in Figure 3 b). From the results, it can be observed that RDA is robust and keeps stable with regard to the different numbers of d. *Third*, the two parameters are tuned from the given set $[10^{-1}, 10^{0}, 10^{1}]$. From the results on two tasks $C \rightarrow A$ and $M \rightarrow V$ shown in Figure 3 c) and d), we can observe that the parameter α has a relatively larger impact on the performance, which represents the importance of the subspace alignment loss. Generally, a larger λ contributes much to the unreliable target sample discovery and rectification with the adversarial loss. In general, the parameters can be easily tuned in experiments.

Ablation Analysis In RDA, three main components are involved: subspace alignment loss $\mathcal{L}_{sub}$ (SA), feature alignment loss $\mathcal{L}_{feat}$ (FA) and adversarial loss $\mathcal{L}_{adv}$ (Adv). For a better insight into the model, ablation analysis is presented. We randomly select several tasks and report the results in Figure 4 by using the model without (w/o) the associated loss terms. From the results, we observe that each loss is indispensable. In

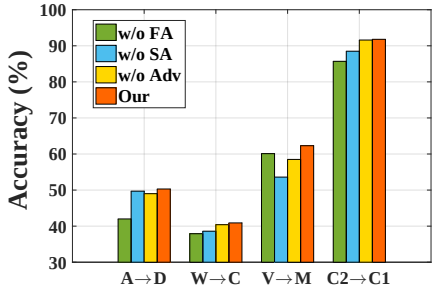


Fig. 4. Ablation analysis of RDA model.

general, the FA term has the greatest impact on performance. This is because there exists a large distribution divergence between two domains and most of the samples are misclassified. The SA term is also important, since it verifies that using domain-specific projections is more effective than a shared projection. The results also demonstrate that the adversarial loss term can further boost performance by improving the reliability of the model. The effectiveness of the adversarial regularization term is verified.

5 Conclusion

In this paper, we proposed a new Reliable Domain Adaptation (RDA) approach for UDA. RDA tries to simultaneously align the manifold and category space across domains through two dual projections. In order to address the prediction bias problem involved by pseudo labels, an adversarial learning strategy is introduced. Firstly, RDA focuses on the discovery of unreliable samples by maximizing the discrepancy between the two task-classifiers. Secondly, RDA focuses on the correction of those unreliable target samples by minimizing the classifiers discrepancy. Comprehensive experiments validate the superiority of RDA over state-of-the-arts.

References

1. B. Gong, Y. Shi, F.S., Grauman, K.: Geodesic flow kernel for unsupervised domain adaptation. ICCV **157**(10), 2066–2073 (2012)
2. Baktashmotlagh, M., Harandi, M.T., Lovell, B.C., Salzmann, M.: Unsupervised domain adaptation by domain invariant projection. In: ICCV. pp. 769–776 (2013)
3. Baktashmotlagh, M., Harandi, M.T., Lovell, B.C., Salzmann, M.: Domain adaptation on the statistical manifold. In: CVPR. pp. 2481–2488 (2014)
4. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. JMLR.org (2006)
5. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Machine Learning **79**(1-2), 151–175 (2010)
6. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations & Trends in Machine Learning **3**(1), 1–122 (2011)

7. Chu, W.S., Torre, F.D.L., Cohn, J.F.: Selective transfer machine for personalized facial expression analysis. In: IEEE Trans. Pattern Analysis and Machine Intelligence. vol. 39, pp. 529–545. (2017)
8. Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T.: Unsupervised visual domain adaptation using subspace alignment. In: ICCV. pp. 2960–2967 (2014)
9. Ghifary, M., Balduzzi, D., Kleijn, W.B., Zhang, M.: Scatter component analysis: A unified framework for domain adaptation and domain generalization. IEEE Trans. Pattern Analysis and Machine Intelligence **39**(7), 1414–1430 (2017)
10. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS. pp. 2672–2680 (2014)
11. Gopalan, R., Li, R., Chellappa, R.: Domain adaptation for object recognition: An unsupervised approach. In: ICCV (2011)
12. GriffinGS, HolubAD, PeronaP: Caltech-256 object category dataset. California Institute of Technology (2007)
13. Kan, M., Wu, J., Shan, S., Chen, X.: Domain adaptation for face recognition: Targetize source domain bridged by common subspace. International Journal of Computer Vision **109**(1-2), 94–109 (2014)
14. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: ICML. pp. 97–105 (2015)
15. Long, M., Wang, J., Ding, G., Sun, J., Yu, P.S.: Transfer feature learning with joint distribution adaptation. In: ICCV. pp. 2200–2207 (2014)
16. Long, M., Wang, J., Ding, G., Sun, J., Yu, P.S.: Transfer joint matching for unsupervised domain adaptation. In: CVPR. pp. 1410–1417 (2014)
17. Long, M., Wang, J., Jordan, M.I.: Unsupervised domain adaptation with residual transfer networks. NIPS (2016)
18. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. IEEE Trans. on Neural Networks **22**(2), 199–210 (2011)
19. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. on Knowledge and Data Engineering **22**(10), 1345–1359 (2010)
20. Rate, C., Retrieval, C.: Columbia object image library (coil-20). Computer (2011)
21. Saenko, K., Kulis, B., Fritz, M., Darrell, T.: Adapting visual category models to new domains. In: ECCV. pp. 213–226 (2010)
22. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. CVPR (2018)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Computer Science (2014)
24. Sun, B., Feng, J., Saenko, K.: Correlation alignment for unsupervised domain adaptation. In: Domain Adaptation in Computer Vision Applications, pp. 153–171. Springer (2017)
25. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: CVPR. pp. 1521–1528 (2011)
26. Tzeng, E., Hoffman, J., Saenko, K., Darrell, T.: Adversarial discriminative domain adaptation. In: CVPR. pp. 7167–7176 (2017)
27. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: ICCV. pp. 4068–4076 (2017)
28. Wang, J., Feng, W., Chen, Y., Yu, H., Huang, M., Yu, P.S.: Visual domain adaptation with manifold embedded distribution alignment. ACM MM (2018)
29. Xu, Y., Fang, X., Wu, J., Li, X., Zhang, D.: Discriminative transfer subspace learning via low-rank and sparse representation. IEEE Trans. image processing **25**(2), 850–863 (2016)
30. Zhang, J., Li, W., Ogunbona, P.: Joint geometrical and statistical alignment for visual domain adaptation. CVPR pp. 5150–5158 (2017)
31. Zhang, L., Zhang, D.: Robust visual knowledge transfer via extreme learning machine-based domain adaptation. IEEE Trans. image processing **25**(10), 4959–4973 (2016)