Front.Comput.Sci.
DOI

Supplementary File

Self-Adaptive Label Filtering Learning for Unsupervised Domain Adaptation (Extended Version)

Qing TIAN(✉)[1,2], Heyang SUN[1], Shun PENG[1], Tinghuai MA[1,2]

1 School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China
2 Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing University of Information Science and Technology, Nanjing 210044, China

Abstract As an emerging machine learning paradigm, unsupervised domain adaptation (UDA) aims to train an effective model for unlabeled target domain by leveraging knowledge from related but distribution-inconsistent source domain. Most of the existing UDA methods align class-wise distributions resorting to target domain pseudo-labels, for which hard labels may be misguided by misclassifications while soft labels are confusing with trivial noises so that both of them tend to cause frustrating performance. In this article, we put forward a novel self-adaptive label filtering learning (SALFL) framework to address this issue. Specifically, we design a graph-based random walking strategy for SALFL to predict pseudo-labels and then refine them via self-adaptive label filtering mechanism. Further, we perform more general joint distribution adaptation on the refined labels and extend this framework to the deep network architecture. Finally, we optimize the proposed SALFL objective by designing an alternating algorithm, and demonstrate the superiority of the proposed approach by extensive experiments on cross-domain datasets.

Keywords unsupervised domain adaptation, self-adaptive label filtering, random walking, general joint distribution alignment

1 Introduction

In some machine learning applications, labeling data, *e.g.* pathological images, is usually cost expensive and even infeasible, thus it becomes advisable to train the required model on unlabeled data together with other related labeled data. In this setting, the performance of traditional machine learning models, however, likely tend to degenerate, since the assumption that the training data and testing data are drawn from the same distribution, cannot be satisfied. Fortunately, as an emerging machine learning paradigm, domain adaptation (DA) [1] was recently proposed to deal with this challenge by mitigating the distribution inconsistency between the training and testing data. As a significant branch of transfer learning [2], DA has been widely applied to cross-domain image classification [3, 4], object detection [5, 6] and natural language processing [7, 8], etc.

In DA, the knowledge from the source domain is leveraged to the target domain with limited or even none labeled data. In terms of whether labeled data is involved in target domain, the DA approaches can be categorized into semi-supervised DA and unsupervised DA (UDA) [1]. In this article, we focus on UDA which adapts knowledge from the source to the unsupervised target domains, which has wide applications and is more challenging.

The original foundation of UDA is to align the distributions between the source and target domains [9–11], therefore the choice of distribution discrepancy metric is critical. In recent years, a variety of distance metrics [12] have been successively applied in UDA, such as Kullback-Leibler divergence [13], Mahalanobis distance [14], Wasserstein distance [15], etc. Among them, the Maximum Mean Discrepancy (MMD) [16] has achieved wide success because of its effectiveness and simplicity. The MMD metric is a non-parametric distribution discrepancy measure which is typically modeled in the reproducing kernel Hilbert space (RKHS) [17] with

Received month dd, yyyy; accepted month dd, yyyy

E-mail: tianqing@nuist.edu.cn

solid theoretical foundation. It measures the marginal distribution discrepancy [18] by minimizing the distribution mean discrepancy between the source and target domains. Along this line, the conditional MMD (CMMD) [9] was designed to mitigate the class-wise mean discrepancy between the domains. Subsequently, many studies were conducted to generalize MMD through various strategies, like balancing between MMD and CMMD [10], integrating MMD with other losses [11], or increasing the discrimination of MMD [19].

In addition, in order to explore the natural structure information contained in UDA, the geometric structure learning [19–21] was introduced and widely achieved promising performance, since the data in UDA is typically distributed with neighbor-similarity relationships in some low-dimensional manifold space. According to this, the geometric structure knowledge from training data has been successfully exploited to guide pseudo-label prediction for the target domain.

In the UDA methods aforementioned, the target domain is usually assigned with pseudo-labels by a predictor trained on the source domain. In this setting, the pseudo-labels can be either encoded as hard labels [9] or soft labels [22]. For the hard type of labels, it assigns value 1 for the class to which an instance belongs, and 0 for the other classes. However, this kind of label coding easily tend to incur severe negative transfer if the encoded labels are wrong. In contrast, the kind of soft labels assign a probability, between 0 and 1, for each class to which the instance belongs.

Comparatively speaking, although the kind of soft labels can alleviate the negative influence by scattering the probability distributions, it may misdirect knowledge transfer since some unrelated classes may be assigned certain probabilities. To overcome such drawbacks, as shown in Fig. 1, we propose to achieve UDA by performing self-adaptive label filtering learning (SALFL) from both the statistical and the geometrical perspectives, which filters out the misclassified pseudo-labels to reduce negative transfer. Specifically, the proposed SALFL firstly predicts labels for the target domain instances by graph-based random walking and then filters out those noise labels by self-adaptive learning strategy. In addition, the proposed SALFL framework seeks a latent common space to align the joint feature distributions of the domains with more general form and is extended with deep network architecture. Finally, we optimize the SALFL objective by designing an alternating algorithm. In summary, the main contributions of this work are fourfold as follows.

- A self-adaptive label filtering learning (SALFL) framework is proposed for unsupervised domain adaptation (UDA) by aligning the source and target domains from both statistical and geometrical perspectives, which filters out the confusing probabilities by self-adaptive label filtering.
- In the SALFL framework, more general joint distribution adaptation form is proposed with the refined cross-domain labels and an efficient alternating optimization algorithm is built, with time complexity analysis.
- To verify the generality of the SALFL framework, we further extend it with the deep network architecture.
- Extensive evaluations and comparisons are conducted to demonstrate the superiority of the proposed method.

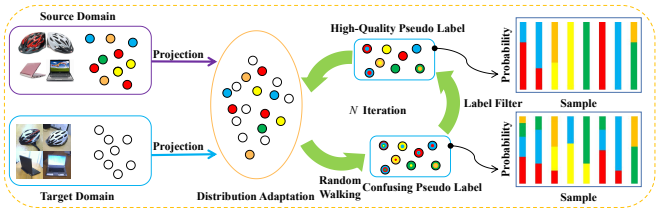


Fig. 1: Illustration of the proposed framework. The probability that a sample belongs to one class is represented by the area occupied by its corresponding color. The more colors in a sample, the more confusing that sample is.

The remainder of this article is organized as follows. Section 2 reviews the related work. Section 3 elaborates the proposed method. Section 4 presents the experiment settings and result analysis. Finally, Section 5 concludes this work and gives future research directions.

2 Related work

In this section, we review the UDA works that are most relevant to our method.

2.1 UDA with feature space alignment

Feature-based statistical alignment methods [23, 24] have been proven to be effective, which align the feature distributions through space transformation between the source and target domains. For instance, the subspace disagreement was adopted in GFK [23] by measuring its optimal dimensionality and then integrating the resulting cross-domain subspaces to model domain shift statistically. Although GFK is elegant and effective, it is inflexible on expansibility. Later, SA [24] was proposed to transform the source space to the target, which has ignored the redundancy of target space in spite of aligning the domain shift to a certain extent.

Actually, most UDA methods [9, 10, 18, 25, 26] attempt to mitigate domain shift in a shared subspace. Along this line, MMDE [25] was designed in an aligned low-dimensional latent feature space by integrating MMD into MVU [27]. Additionally, TCA [18] was designed to directly project the domains to a common space by minimizing their scatters with the MMD measure. Apart from the total distribution alignment, JDA [9] was proposed by incorporating the conditional MMD with the marginal MMD to perform alignment between the domains. To address the issue of imbalanced tasks in UDA, TJM [26] was built to reweigh the instances across domains together with joint feature matching. Later, BDA [10] achieved better results than TJM by introducing a balancing factor between the marginal and conditional distribution alignments. In order to make use of more discriminant knowledge, JGSA [11] modeled the distribution variance of the target domain, within-class and between-class scatters of source domain as well as cross-domain distribution divergence jointly in a unified objective. DICD [21] learned class discriminative representations for UDA by measuring both the intra-class similarity and inter-class dissimilarity.

Similarly, the idea of conditional distribution alignment is also widely used in deep models thanks to its effectiveness. MSTN [28] attempted to align the labeled source centroid and pseudo-labeled target centroid of semantic representations learned by the moving semantic transfer network. For example, MADA [29] aimed at fine-grained alignment of data distribution based on multiple domain discriminators. MRAN [30] extracted the multiple representations by a hybrid neural structure to realize the conditional distribution discrepancy minimization. Further, CDAN [31] enabled the alignment of multimodal distributions by conditioning the adversarial domain adaptation on discriminative information. Recently, DSAN [4] proposed a transfer network to align the local maximum mean discrepancy of the domain-special layer across different domains.

2.2 UDA with geometrical structure learning

Considering that the Geometrical structure learning [32] can explore the data structures, it has been adopted in UDA [19, 20, 33]. For instance, MEDA [33] was designed to learn a domain-invariant UDA classifier by minimizing the data structural risk on the Grassmann manifold. DGA-DA [19] was built to perform UDA by inferring the target domain labels through geometric structure learning. Nevertheless, the geometrical structure learning is usually implemented in iterative manner, such that the quality of the pseudo-labels generated in previous rounds is crucial to the subsequent rounds and it is prone to bring about cumulative errors. Unfortunately, such an issue is widely ignored in the methods discussed above.

2.3 UDA with label filtering learning

In UDA, the problem of unreliable pseudo-labels [34–36] has received certain attention but has not been resolved well. Specifically, CRST [34] encouraged the smoothness of soft pseudo-labels via the confidence regularization which could acquire more confident predictions but it is inconvenient to optimize the embedded regularization loss and also hard to transfer to other models. In addition, CAN [35] filtered the points and classes far away from the cluster centers or contain few target samples via zeroing them out. Though this method could mitigate the interference of noise pseudo-labels, it suffers from the problem of overconfidence pseudo-labels and the SALFL we proposed aims at handling both problems simultaneously. PFAN [36] utilized an Easy-to-Hard Transfer Strategy (EHTS) to select easy samples progressively which were used to align domains. The local domain consisted of selected samples has higher confidence and is easy to align but the confidence of each sample remains the same. Different from selecting samples iteratively and locally, our method attempts to make each sample more reliable by the self-adaptive label filtering learning strategy which works on all target samples.

3 Self-adaptive label filtering learning

In this section, we first define notations to be used and formalize the research problem to be addressed. Then we elaborate the proposed self-adaptive label filer learning (SALFL) frameowrk and present an optimization algorithm to solve it. Additionally, we analyze the time complexity of the model. Finally, we extend the model with deep network architecture.

3.1 Prelimination

In this article, matrices are written as boldface uppercase letters and vectors as boldface lowercase letters. $\|\boldsymbol{X}\|_F$ and $\mathrm{tr}(\boldsymbol{X})$ denote the Frobenius norm and trace of the matrix $\boldsymbol{X}$, respectively. For convenience, notations involved in this article are summarized in Table 1.

In UDA scenario, given a labeled source domain dataset $\mathcal{D}_S : \{\boldsymbol{X}_S, \boldsymbol{Y}_S\} = \left\{\boldsymbol{x}_S^i, \boldsymbol{y}_S^i\right\}_{i=1}^{n_S} \in \{\mathcal{X}_S, \mathcal{Y}_S\}$, where $\boldsymbol{x}_S^i \in \mathbb{R}^d$ denotes the ith source domain instance and $\boldsymbol{y}_S^i \in \mathbb{R}^C$ represents its corresponding label from totally C classes. UDA aims to assign labels for unlabeled target instances $\mathcal{D}_T : \{\boldsymbol{X}_T\} =$

Table 1: Summary of notation definitions involved in this article.

Notation	Meaning
d	the original feature dimension
p	the neighbor number
λ	the regularization parameter
C	the number of classes
T	the total iterations
n_S / n_T	the number of source/target domain instances
$\boldsymbol{W}$	the affinity matrix
$\boldsymbol{D}$	the degree matrix
$\boldsymbol{P}$	the path weight matrix
$\boldsymbol{M}_c$	the MMD matrix
$\boldsymbol{M}_d$	the discriminative loss matrix
$\boldsymbol{H}$	the centering matrix
$\boldsymbol{A}$	the projection matrix
$\boldsymbol{Y}_S$	the source domain hard labels
$\boldsymbol{F}_S / \boldsymbol{F}_T$	the source/target domain soft labels
$\boldsymbol{X}_S / \boldsymbol{X}_T$	the source/target domain instance features

$\left\{\boldsymbol{x}_T^i\right\}_{i=1}^{n_T}$ from the target domain $\mathcal{X}_T$, where $\boldsymbol{x}_T^i \in \mathbb{R}^d$ indicates the ith target domain instance. The source domain and target domain share same feature space $\mathcal{X}_S = \mathcal{X}_T$ and label space $\mathcal{Y}_S = \mathcal{Y}_T$. In addition, both the domains have different marginal and conditional distributions, i.e. $\mathcal{P}_S(\boldsymbol{x}_S) \neq \mathcal{P}_T(\boldsymbol{x}_T)$ and $\mathcal{P}_S(\boldsymbol{y}_S \mid \boldsymbol{x}_S) \neq \mathcal{P}_T(\boldsymbol{y}_T \mid \boldsymbol{x}_T)$.

3.2 Graph-based random walking

Considering a label prediction algorithm in the manner of a special random walking on the graph, which allows unlabeled points (i.e., target domain data) walking randomly to find label information from labeled points (i.e., source domain data) following the guidance of a neighborhood similarity graph. The affinity matrix $\boldsymbol{W} \in \mathbb{R}_+^{n \times n}$ of this graph structure is constructed with elements:

$$\boldsymbol{w}_{ij} := \begin{cases} \delta\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right), & \text{if } i \neq j \wedge \boldsymbol{x}_i \in \mathrm{NN}_k\left(\boldsymbol{x}_j\right) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $\delta\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = e^{-\left\|x_i - x_j\right\|^2 / \sigma^2}$ measures the non-negative similarity between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, σ^2 indicates the variance and NN_k denotes the set of k nearest neighbors in domains. As shown in Fig. 2, the transformation of each random walking can be defined as

$$\tilde{\boldsymbol{P}} = \boldsymbol{I}_\beta + \boldsymbol{I}_\alpha \boldsymbol{P} \tag{2}$$

where $\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{W}$ indicates the weights of paths in the nearest neighbors graph, the degree matrix $\boldsymbol{D}$ is a diagonal matrix with entries $d_i = \sum_j \boldsymbol{w}_{ij}$. Additionally, diagonal matrix $\boldsymbol{I}_\alpha$ and $\boldsymbol{I}_\beta$ is consisted of orientation coefficients α_i and β_i respectively, if the ith point is unlabeled (target domain data), let $\alpha_i = 1, \beta_i = 0$, which forces the point to walk around. And if the point is labeled (source domain data), let $\alpha_i = 0, \beta_i = 1$, which constrains the point staying put.

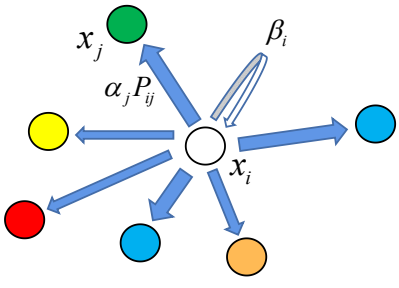


Fig. 2: Demonstration of graph-based random walking.

On the neighbors graph, each point walks randomly from starting point based on the transformation matrix $\tilde{\boldsymbol{P}}$ until it consecutively arrives any point twice. Let $\hat{\boldsymbol{P}} = \boldsymbol{I}_\alpha \boldsymbol{P}$ and $\boldsymbol{G}_{ij}$ represent the ith point stopping walking at the jth point. Then, the process of random walking can be described as following:

$$\boldsymbol{G} = \boldsymbol{I}_\beta + \hat{\boldsymbol{P}}\boldsymbol{I}_\beta + \hat{\boldsymbol{P}}^2\boldsymbol{I}_\beta + \cdots + \hat{\boldsymbol{P}}^k\boldsymbol{I}_\beta + \cdots \tag{3}$$

where $\left(\hat{\boldsymbol{P}}^k\boldsymbol{I}_\beta\right)_{ij}$ denotes the probability of the ith point stopping at the jth point on the kth step. Further, the soft label matrix $\boldsymbol{F}$ can be written as:

$$\boldsymbol{F} = \boldsymbol{G}\boldsymbol{Y} \tag{4}$$

3.3 Self-adaptive label filtering learning

The predicted labels for the target domain by random walking aforementioned is crucial to subsequent domain distribution alignment. The next step is to encode the labels with hard or soft labels. However, the hard labels are so confident that it easily tends to mislead the model, especially when encountered misclassifications. By comparison, the soft labels may also confuse the model by those minuscule label components. To overcome this problem, we put forward the self-adaptive label filtering learning to filter out wrong or noisy (pseudo) labels for the target domain.

Specifically, we denote respectively by $\boldsymbol{Z}_S, \boldsymbol{Z}_T$ the feature representations for the source and target domains, $(\boldsymbol{F}_T)_{i\cdot}$ the predicted pseudo-label coding vector of the ith instance in target domain by the random walking, with $(\boldsymbol{F}_T)_{ic}$ being the probability of it belonging to the cth class. At first, we relabel each instance for target domain as $\hat{y}_i$ via the nearest centroid classifier:

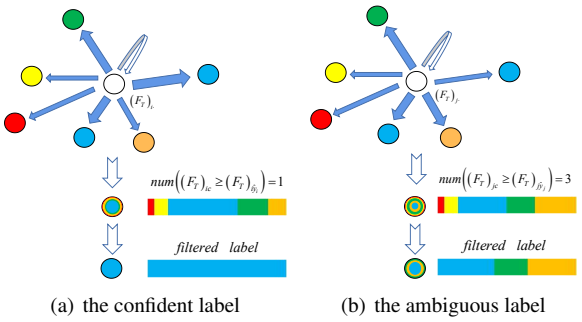
$$\hat{y}_i = \arg\min_k D_f\left((\boldsymbol{Z}_T)_i, c_k\right) \tag{5}$$

where $D_f(\cdot,\cdot)$ measures the Euclidean distance between two data points, c_k stands for the kth class centroid of the source

domain. Then, we update (refine) the pseudo-labels as follows:

$$(\boldsymbol{F}_T)^*_{ic} = \frac{(\boldsymbol{F}_T)_{ic}\,\varepsilon\left((\boldsymbol{F}_T)_{ic} - (\boldsymbol{F}_T)_{i\hat{\mathrm{y}}_i}\right)}{\sum_{j=1}^{C}(\boldsymbol{F}_T)_{ij}\,\varepsilon\left((\boldsymbol{F}_T)_{ij} - (\boldsymbol{F}_T)_{i\hat{\mathrm{y}}_i}\right)} \tag{6}$$

where the unit step function $\varepsilon(x) = 1$ when $x \geq 0$, and equals to 0 otherwise. Let us define function *nonzero* $(\boldsymbol{x})$ that returns the number of nonzero values in vector $\boldsymbol{x}$. As shown in Fig. 3, for one refined label $(\boldsymbol{F}_T)^*_i$, if *nonzero* $((\boldsymbol{F}_T)^*_i) = 1$, we consider it as confident label (Fig. 3(a)) which will benefit subsequent class-wise alignment. And if *nonzero* $((\boldsymbol{F}_T)^*_i) > 1$, we consider that it is an ambiguous label (Fig. 3(b)) which can eliminate the interference of trivial label probability components and thus mitigate possible negative transfer. We term the filtering process as self-adaptive label filtering, for which we use source domain distribution centroid to approximate the target domain centroid and it is a target-domain unsupervised manner. Actually, the label filtering learning process can be considered as combination of soft and hard label encodings based on label selection. Confident labels use the form of hard labels and ambiguous labels use the form of soft labels which benefits filtering out confusing labels.



(a) the confident label (b) the ambiguous label

Fig. 3: Demonstration of label filtering for the predicted pseudo-labels.

3.4 General form of domain adaptation with filtered labels

Although the proposed SALFL framework can be combined with most existing UDA methods, such as JDA [9], JGSA [11], DGA-DA [19], we need to mitigate the distributional shift based on filtered labels which could not be performed as these methods directly. Therefore, we formulate more general domain adaptation form. To align both the marginal and conditional distributions between the source domain and target domain, we resort to the maximum mean discrepancy (MMD) to measure their divergence. Mathematically, the marginal MMD between the domains is defined as:

$$\begin{aligned}&\mathrm{Dist}^{\,marginal}_{(\mathcal{D}_S,\mathcal{D}_T)}\\ &= \left\| \frac{1}{n_S}\sum_{\boldsymbol{x}_i\in\boldsymbol{X}_S}\boldsymbol{A}^T\boldsymbol{x}_i - \frac{1}{n_T}\sum_{\boldsymbol{x}_j\in\boldsymbol{X}_T}\boldsymbol{A}^T\boldsymbol{x}_j \right\|_F \\ &= \mathrm{tr}\left(\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{M}_0\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\right)\end{aligned} \tag{7}$$

where $\boldsymbol{A}$ is the subspace projection matrix. Let us define vectors $\boldsymbol{e}_S = (1/n_S)\,\mathbf{1}^{n_S\times 1}$, $\boldsymbol{e}_T = (-1/n_T)\,\mathbf{1}^{n_T\times 1}$, $\boldsymbol{e} = [\boldsymbol{e}_S; \boldsymbol{e}_T]$, the MMD matrix $\boldsymbol{M}_0 = \boldsymbol{e}\boldsymbol{e}^T$. Differently, in order to combine the filtered (refined) labels with conditional MMD measure effectively, we reformulate the conditional distribution distance across domains as follows:

$$\begin{aligned}&\mathrm{Dist}^{\,conditional}_{(\mathcal{D}_S,\mathcal{D}_T)}\\ &= \sum_{c=1}^{C}\left\| \frac{1}{\tilde{n}^c_S}\sum_{\boldsymbol{x}_i\in\boldsymbol{X}_S}(\boldsymbol{F}_S)_{ic}\,\boldsymbol{A}^T\boldsymbol{x}_i - \frac{1}{\tilde{n}^c_T}\sum_{\boldsymbol{x}_j\in\boldsymbol{X}_T}(\boldsymbol{F}_T)_{jc}\,\boldsymbol{A}^T\boldsymbol{x}_j \right\|_F \\ &= \mathrm{tr}\left(\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{M}_c\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\right)\end{aligned} \tag{8}$$

where $\tilde{n}^c_S = \sum_{i=1}^{n_S}(\boldsymbol{F}_S)_{ic}$, $\tilde{n}^c_T = \sum_{i=1}^{n_T}(\boldsymbol{F}_T)_{ic}$, $\boldsymbol{F}_S$ and $\boldsymbol{F}_T$ are labels of source and target domains obtained from the random walking, respectively. The reformulated MMD matrices $\boldsymbol{M}_c$ are calculated as follows:

$$(\boldsymbol{M}_c)_{ij} = \begin{cases} \sum_{c=1}^{C}\frac{(\boldsymbol{F}_S)_{ic}(\boldsymbol{F}_S)_{jc}}{\tilde{n}^c_S\tilde{n}^c_S}, & \boldsymbol{x}_i,\boldsymbol{x}_j\in\mathcal{D}_S \\ \sum_{c=1}^{C}-\frac{(\boldsymbol{F}_S)_{ic}(\boldsymbol{F}_T)_{jc}}{\tilde{n}^c_S\tilde{n}^c_T}, & \boldsymbol{x}_i\in\mathcal{D}_S,\boldsymbol{x}_j\in\mathcal{D}_T \\ \sum_{c=1}^{C}-\frac{(\boldsymbol{F}_T)_{ic}(\boldsymbol{F}_S)_{jc}}{\tilde{n}^c_T\tilde{n}^c_S}, & \boldsymbol{x}_i\in\mathcal{D}_T,\boldsymbol{x}_j\in\mathcal{D}_S \\ \sum_{c=1}^{C}\frac{(\boldsymbol{F}_T)_{ic}(\boldsymbol{F}_T)_{jc}}{\tilde{n}^c_T\tilde{n}^c_T}, & \boldsymbol{x}_i,\boldsymbol{x}_j\in\mathcal{D}_T \end{cases} \tag{9}$$

Although the joint domain distribution alignment could offer the model the discriminability on target domain, it merely focuses on the data from the same category or domain, but ignores the discriminability between different categories for the same or different domains. To this end, we need to maximize the discriminability between the category clusters in both domains. Specifically, in the joint domain $\mathcal{D} = \{\mathcal{D}_S, \mathcal{D}_T\}$, we denote by $\mathcal{D}^c$ the sub-domain of the cth category, and $\mathcal{D}^{-c}$ the sub-domains of all categories except the cth. Then, we formulate the between-category discriminative measure as:

$$\begin{aligned}&\mathrm{Dist}^{\,discriminative}\\ &= \sum_{c=1}^{C}\left\| \frac{1}{n^c}\sum_{\boldsymbol{x}_i\in\mathcal{D}^c}\boldsymbol{F}_{ic}\boldsymbol{A}^T\boldsymbol{x}_i - \frac{1}{n^{-c}}\sum_{\boldsymbol{x}_j\in\mathcal{D}^{-c}}\boldsymbol{F}_{jc}\boldsymbol{A}^T\boldsymbol{x}_j \right\|_F \\ &= \mathrm{tr}\left(\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{M}_d\boldsymbol{X}^{\mathrm{T}}\boldsymbol{A}\right)\end{aligned} \tag{10}$$

where $\boldsymbol{M}_d$ stands for the discriminative matrix, defined as:

$$\begin{aligned}(\boldsymbol{M}_d)_{ij} = &\sum_{c=1}^{C}\frac{(\boldsymbol{F})_{ic}\,(\boldsymbol{F})_{jc}}{\tilde{n}^c\tilde{n}^c} - \frac{(\boldsymbol{F})_{ic}\,(\boldsymbol{F})_{j(-c)}}{\tilde{n}^c\tilde{n}^{-c}} \\ &- \frac{(\boldsymbol{F})_{i(-c)}\,(\boldsymbol{F})_{jc}}{\tilde{n}^c\tilde{n}^{-c}} + \frac{(\boldsymbol{F})_{i(-c)}\,(\boldsymbol{F})_{j(-c)}}{\tilde{n}^{-c}\tilde{n}^{-c}}\end{aligned} \tag{11}$$

The maximization of (10) can effectively enlarge the distances between categories in both the source and target domains, which further enhances the discrimination ability of the proposed model.

3.5 Final objective formulation

By taking into account all the components above, we can consequently formulate the final objective of the SALFL method, as follows:

$$\min_{\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{H}\boldsymbol{X}^T\boldsymbol{A}=\boldsymbol{I}} \operatorname{tr}\left(\boldsymbol{A}^T\boldsymbol{X}\left(\boldsymbol{M}_0+\boldsymbol{M}_c-\boldsymbol{M}_d\right)\boldsymbol{X}^T\boldsymbol{A}\right)+\lambda\|\boldsymbol{A}\|_F^2 \quad (12)$$

whose first term denotes the designed discriminative module, while the second term limits the model complexity with λ being the tradeoff parameter. The marginal MMD matrix $\boldsymbol{M}_0 = ee^T$, the reformulated MMD matrix $\boldsymbol{M}_c$ and $\boldsymbol{M}_d$ are defined as (9) and (11) respectively, both of them are computed using the filtered pseudo-label matrix $\boldsymbol{F}$ calculated in (4). Additionally, in order to avoid trivial solutions, we embed the Principal Component Analysis (PCA) [37] criterion in the final model as data variance preservation constraint, that is $\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{H}\boldsymbol{X}^T\boldsymbol{A} = \boldsymbol{I}$, and the centering matrix $\boldsymbol{H} = \boldsymbol{I} - \left(\frac{1}{n}\right)\mathbf{1}\mathbf{1}^T$, where $n = n_S + n_T$.

3.6 Optimization

The optimization of objective (12) consists of two parts. First is label prediction by graph-based random walking. In (3), $\boldsymbol{G}(k)_{ij}$ denotes the ith point walk up to the jth point at kth random walking. When all the data instances stop the process of moving, according to the convergence criterion, we have:

$$\boldsymbol{F} = \lim_{k\to\infty}\boldsymbol{G}(k)\boldsymbol{Y} = (\boldsymbol{I}-\hat{\boldsymbol{P}})^{-1}\boldsymbol{I}_\beta\boldsymbol{Y} = (\boldsymbol{I}-\boldsymbol{I}_\alpha\boldsymbol{P})^{-1}\boldsymbol{I}_\beta\boldsymbol{Y} \quad (13)$$

In iterative optimization process of the model, the label matrix $\boldsymbol{Y}$ is actually the $\boldsymbol{F}$ obtained in the previous round.

The second part that needs to be optimized is about the label filtering-based discriminative domain adaptation. Integrating the PCA constraint through the augmented Lagrangian method [38], (12) can be rewritten as

$$\begin{aligned}\min_{\boldsymbol{A}} &\operatorname{tr}\left(\boldsymbol{A}^T\left(\boldsymbol{X}\left(\boldsymbol{M}_0+\boldsymbol{M}_c-\boldsymbol{M}_d\right)\boldsymbol{X}^T+\lambda\boldsymbol{I}\right)\boldsymbol{A}\right)\\ &+\operatorname{tr}\left(\left(\boldsymbol{I}-\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{H}\boldsymbol{X}^T\boldsymbol{A}\right)\Phi\right)\end{aligned} \quad (14)$$

where $\Phi = \operatorname{diag}(\lambda_1, \cdots, \lambda_k)$ is the Lagrange multiplier. Taking the partial derivative of (13) with respect to projection matrix $\boldsymbol{A}$, we obtain:

$$\left(\boldsymbol{A}^T\left(\boldsymbol{X}\left(\boldsymbol{M}_0+\boldsymbol{M}_c-\boldsymbol{M}_d\right)\boldsymbol{X}^T+\lambda\boldsymbol{I}\right)\boldsymbol{A}\right) = \boldsymbol{X}\boldsymbol{H}\boldsymbol{X}^T\boldsymbol{A}\Phi \quad (15)$$

Finally, (15) can be solved by generalized eigenvalue decomposition, the optimal projection matrix $\boldsymbol{A}$ is composed of the eigenvectors corresponding to the first smallest k eigenvalues. For clarity, we tabulate the complete process of SALFL in Algorithm 1.

Algorithm 1: Optimization Algorithm for SALFL

Input : Source domain $\{\boldsymbol{X}_S, \boldsymbol{F}_S\}$, target domain $\{\boldsymbol{X}_T\}$, regularization parameter λ, maximum iteration number T.
Output: Target domain labels $\boldsymbol{F}_T^*$.

```
Compute P by X;
repeat
    Compute F_T based on (13);
    Filter F_T as described in Section 3.3;
    Compute M_0 = ee^T;
    Compute M_c based on (9);
    Compute M_d based on (11);
    Compute A based on (15);
    Obtain Z = A^T X;
    Compute P by Z;
until iterations > T;
```

3.7 Time complexity analysis

We use the big O notation analyzing the time complexity of Algorithm 1 which mainly lies in iterative updating the variables. We denote T the number of iterations, n the number of samples and d the feature dimension. In one iteration, the time complexity of label prediction is $O\left(cn^2\right)$, the time complexity of label filtering is $O(n)$, and computing matrices $\boldsymbol{M}_0$, $\boldsymbol{M}_c$, $\boldsymbol{M}_d$ and $\boldsymbol{P}$ costs a complexity of $O\left(Cn^2+n^3\right)$. Finally, the singular-value decomposition (SVD) takes $O\left(n^3\right)$. Therefore, the total time complexity of Algorithm 1 is $O\left(Tn^3+TCn^2\right)$.

3.8 SALFL framework extension with the deep methods

The proposed SALFL framework is constructed by random walking, label filtering and domain adaptation where the part of domain adaptation can be replaced by other DA methods. Therefore, to illustrate the efficacy of our proposed SALFL with deep domain-invariant feature, we empower the representative deep UDA methods by replacing their pseudo label generators with our SALFL while keeping other settings unchanged.

4 Experiment

In this section, we verify the effectiveness of proposed SALFL model on seven real world cross-domain datasets.

Firstly, we describe the datasets, the baseline methods which the proposed SALFL is compared to and experimental setup. Then, we discuss the experimental results with hypothesis test. Finally, we analyze the construction effectiveness of the SALFL via ablation study and the parameter sensibility.

4.1 Datasets

We evaluate the proposed SALFL method on seven popular standard benchmark cross-domain datasets: Office-10 [39], Caltech-10 [40], Office-Home [41], USPS [42], MNIST [43], VOC-2007 [44], MSRC [45], Office-31 [39]. The summary of the details of these datasets is demonstrated in Table 2.

Table 2: Profiles of the used benchmark datasets.

Dataset	Sample	Feature(Dimension)	Class
Amazon-10	958	SURF(800) /DeCAF$_6$ (4096)	10
Webcam-10	295		
DSLR-10	157		
Caltech-10	1123		
Artistic	2427	ResNet-50(4096)	65
Clipart	4365		
Product	4439		
Real-World	4357		
USPS	1800	gray-scale pixel(256)	10
MNIST	2000		
VOC-2007	1530	Dense SIFT(128)	6
MSRC	1269		
Amazon-31	2817	image	31
Webcam-31	795		
DSLR-31	498		

Office-10 + Caltech-10: Office-10 is a real world dataset including three subdatasets. Amazon (A) images are photographs taken in a light-controlled studio environment. DSLR (D) and Webcam (W) images are both captured in the home with nature light and what is different is that the former is high-resolution while the latter is low-resolution. Caltech-10 contains standard object images collected from Google Image. The four datasets consist of 2533 images from 10 categories with two representations: 800-dimension SURF feature and 4096-dimension DeCAF$_6$ feature. Following the previous work [11], we construct 12 pairs of domain combinations for each feature representation, so we get 24 groups of UDA tasks such as "$A_{SURF} \rightarrow C_{SURF}$", "$A_{DeCAF_6} \rightarrow C_{DeCAF_6}$" *et al*.

Office-Home: Office-Home is a more challenging DA dataset with more than 15000 images from 65 categories consisting of four subdatasets: Artistic images (Ar), Clipart images (Cl), Product images (Pr) and Real-World images. We extract their 4096-dimension deep features by the standard ResNet-50 model. Using a similar manner, we design 12 pairs of UDA tasks, for example "Ar → Pr".

MNIST + USPS: MNIST and USPS are classical handwritten digit datasets. To construct UDA tasks, we choose randomly 1800 images form USPS, 2000 images form MNIST and reshape them to size 16×16 so that we could obtain the 256-dimension features of gray-scale pixel values. Finally, we get two cross-domain tasks: "MNIST → USPS" and "USPS → MNIST".

VOC-2007 + MSRC: VOC-2007 images are digital photos from Flickr[1)], and MSRC is a standard image dataset provided by Microsoft Research Cambridge. Following previous work [45], we construct UDA tasks "VOC-2007 → MSRC" and "MSRC → VOC-2007" by selecting six shared semantic classes including 1269 images from MSRC and 1530 images from VOC-2007 where 128-dimension Dense SIFT features are used.

Office-31: Office-31 is an extension of Office-10, it contains more than 4,000 images built in 31 categories. We evaluate our method across 6 transfer tasks, such as "A → W", which are commonly adopted in deep learning methods [35, 46].

4.2 Baseline methods and setup

We compare our method with the state-of-the-art related works: PCA [37], SA [24], GFK [23], TCA [18], TJM [26], JDA [9], BDA [10], JGSA [11], MEDA [33], DICD [21], DGA-DA [19]. The hyperparameters of these comparison methods are the same as the settings in their article while for settings not mentioned, we use grid search cross-validation to obtain. For fair comparison, we uniformly use ten-fold cross-vlidation. Additionally, to further demonstrate the superiority of proposed SALFL with deep-learning-based UDA methods: DAN [47], DANN [48], JAN [49], CDAN [31], CAN [35], DADA [46]. It could be unfair to compare with deep UDA methods directly so that we apply ResNet-50 features to evaluate.

For the aforementioned shallow methods involving subspace learning, we uniformly fix subspace dimension $k = 50$ in datasets Office-10, Caltech-10, VOC-2007 and MSRC, and $k = 100$ in others. As for other parameters involved in SALFL, we set the maximum iteration number $T = 10$, the neighbour number $p = 20$ and the only one non-fixed hyperparameter λ is searched in the range of $\{1e-4, 1e-3, 1e-2, 1e-1, 1e0, 1e1, 1e2\}$ by ten-fold cross-validation. Following the previous works [11, 19, 31], the accuracy of the target domain pseudo-labels is used as the evaluation measurement.

[1)] https://www.flickr.com/

Table 3: Recognition accuracy (%) on Office-10 + Caltech-10 with the SURF features.

	C→A	C→W	C→D	A→C	A→W	A→D	W→C	W→A	W→D	D→C	D→A	D→W	Average
PCA	36.78	31.98	37.54	34.82	35.93	27.52	26.34	31.44	77.87	29.52	31.69	75.8	39.77
SA	41.34	41.00	46.81	39.40	39.23	35.17	31.74	34.07	86.26	31.93	35.44	83.99	45.53
GFK	41.27	41.28	39.00	40.22	39.05	36.67	30.98	29.53	80.74	30.28	33.17	75.54	43.15
TCA	38.17	38.52	41.31	37.97	37.96	32.86	29.57	29.94	87.34	31.94	31.98	85.60	43.60
TJM	46.70	39.56	44.46	39.21	41.95	45.21	30.36	29.66	88.71	31.28	32.27	86.11	46.29
JDA	45.28	41.89	45.42	39.26	37.88	39.04	31.65	32.73	89.67	30.63	33.27	89.59	46.36
BDA	51.00	46.48	44.59	39.52	36.76	36.95	32.04	40.27	88.75	32.21	32.08	91.97	47.72
JGSA	53.07	48.21	48.60	41.66	44.91	45.15	33.47	40.87	88.69	30.50	**38.73**	93.74	50.63
MEDA	54.02	48.75	47.58	43.55	40.28	40.31	35.39	**43.53**	92.16	**35.10**	35.83	**95.14**	50.97
DICD	51.96	47.68	46.21	41.66	38.49	38.65	33.68	41.20	91.17	34.08	33.87	93.76	49.37
DGA-DA	52.15	47.34	45.84	41.36	38.38	38.34	33.25	41.56	90.04	33.60	33.56	93.26	49.06
SALFL	**56.22**	**60.77**	**59.05**	**45.95**	**50.53**	**48.34**	**37.03**	42.33	**93.79**	33.67	37.35	94.14	**54.93**

Table 4: Recognition accuracy (%) on Office-10 + Caltech-10 with the $DeCAF_6$ features.

	C→A	C→W	C→D	A→C	A→W	A→D	W→C	W→A	W→D	D→C	D→A	D→W	Average
PCA	85.20	70.39	74.32	69.97	57.43	65.32	60.42	62.87	99.07	50.94	62.74	88.45	70.59
SA	86.71	75.72	79.82	79.65	77.90	81.69	68.71	75.50	**100**	70.41	73.44	99.81	80.78
GFK	87.45	76.30	83.38	80.61	77.26	81.06	67.77	74.72	**100**	69.35	75.85	98.27	81.00
TCA	85.40	88.22	85.91	80.45	75.73	84.05	81.14	84.94	**100**	81.06	87.42	94.06	85.70
TJM	87.87	72.01	74.74	78.15	75.28	82.28	71.87	80.55	**100**	72.62	78.34	98.32	81.00
JDA	89.57	84.28	86.29	81.80	78.05	80.38	80.46	87.81	**100**	80.33	88.96	99.32	86.44
BDA	90.77	84.81	86.66	81.98	78.10	80.49	80.19	87.81	**100**	80.64	89.27	99.15	86.66
JGSA	91.45	86.62	93.84	84.84	80.81	88.68	85.28	89.55	**100**	86.59	92.17	99.57	89.95
MEDA	92.71	95.58	90.49	86.92	**87.31**	88.28	**93.21**	**99**	99.75	87.32	92.87	97.53	92.58
DICD	91.19	92.14	91.31	84.67	80.90	90.31	85.97	89.88	**100**	85.48	91.86	99.10	90.23
DGA-DA	91.18	93.49	91.89	84.97	81.31	89.7	86.07	90.5	**100**	86.93	92.72	**100**	90.73
SALFL	**95.1**	**96.71**	**96.57**	**88.85**	85.98	**92.8**	89.08	93.59	**100**	**90.78**	**96.33**	**100**	**93.81**

4.3 Results and analysis

1) **Results on Office-10 + Caltech-10 Datasets:** The classification results on shallow and deep features of Office-10 vs Caltech-10 are listed in Table 3 and Table 4 respectively (best in bold, second-best underlined). As can be seen, the proposed SALFL performs better than other comparison methods on the whole, where SALFL has the best average accuracy and achieves 3.96% improvement against the best baseline MEDA on SURF features. Compared with shallow features, all the methods improve a lot on deep features, demonstrating the power of the deep learning paradigm. In terms of one list, PCA performs the worst, it makes clear that domains alignment is crucial to boost the DA model. Besides, compared with GFK and SA which align source and target domains geometrically in subspaces, TCA, TJM, JDA and BDA show better performance thanks to apply MMD to measure distribution discrepancy statistically. JGSA combines joint MMD with Fisher discrimination criterion on the source domain enhancing the classification accuracy effectively. However, an interesting phenomenon is that DICD uses Fisher discrimination criterion on the target domain based on JGSA while obtaining worse performance. As we all know that conditional MMD and Fisher discrimination criterion on the target domain both rely on pseudo-labels of the target domain so that the quality of predicted labels determines the performance of models to a certain extent. DICD adds the target-category-relevant loss but performs more awful than JGSA due to this reason. And it also is the essential reason that we put forward the self-adaptive label filtering mechanism to enhance the quality of pseudo-labels.

2) **Results on Office-Home Dataset:** The results on Office-Home with ResNet-50 features are shown in Table 5 where we compare the proposed SALFL with deep UDA methods. Different from shallow UDA, deep UDA methods construct the end-to-end models which integrate feature extraction, knowledge transfer, labels prediction into a whole network so that the deep models could boost the recognition accuracy effectively. It can be observed that in the deep methods we compared, CDAN achieves 7.5% improvement against the second one JAN due to make full use of class-level conditional domain adversarial learning resorting to pseudo-labels. However, the performance of proposed SALFL improves further thanks to the high-quality pseudo-labels which promote conditional alignment more precisely obtained by the self-adaptive label filtering mechanism. The results on Office-Home testified that the proposed SALFL could achieve favorable results on large-scale datasets despite our model is independent of feature extraction compared with

deep methods.

Table 5: Recognition accuracy (%) on Office-Home with the ResNet-50 features.

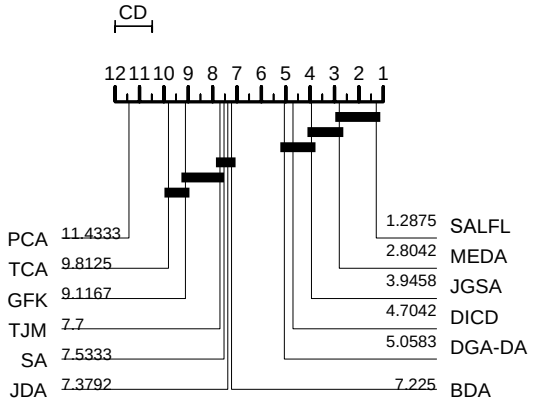
	DAN	DANN	JAN	CDAN	SALFL
Ar→Cl	43.6	45.6	45.9	50.7	**56.9**
Ar→Pl	57.0	59.3	61.2	70.6	**75.2**
Ar→Rw	67.9	70.1	68.9	76.0	**77.6**
Cl→Ar	45.8	47.0	50.4	57.6	**58.6**
Cl→Pr	56.5	58.5	59.7	70.0	**73.2**
Cl→Rw	60.4	60.9	61.0	70.0	**71.3**
Pr→Ar	44.0	46.1	45.8	57.4	**60.5**
Pr→Cl	43.6	43.7	43.4	50.9	**52.6**
Pr→Rw	67.7	68.5	70.3	77.3	**78.3**
Rw→Ar	63.1	63.2	63.9	**70.9**	69.1
Rw→Cl	51.5	51.8	52.4	56.7	**57.5**
Rw→Pr	74.3	76.8	76.8	81.6	**81.6**
Average	56.3	57.6	58.3	65.8	**67.7**

3) **Results on MNIST + USPS and VOC-2007 + MSRC Datasets:** We also conduct comparisons on MNIST + USPS and MSRC + VOC-2007 datasets, the results are reported respectively in Table 6 and Table 7. Similar observations can be found as on the Office-10 + Caltech-10 datasets, the proposed method generally achieves the best results. Additionally, MEDA optimizes joint distribution alignment and the classifier in a unified framework so that achieves the second-best results in all the four UDA tasks. Compared with our SALFL, the performance of MEDA is impacted by the quality of pseudo-labels, but unfortunately, we can hardly improve it resort to the labels filtering mechanism also for the reason that MEDA is an unified model.

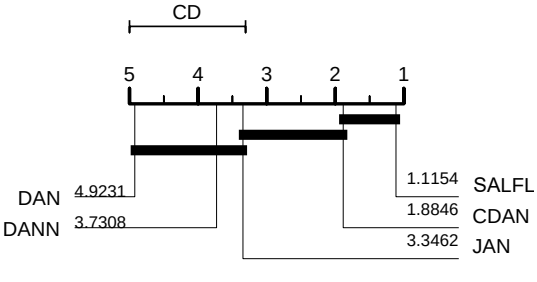
4) **Results of the SALFL framework with the deep methods on Office-31 Datasets:** We combine the proposed SALFL framework with the deep methods to further testify the efficacy of random walking and label filtering on deep DA methods. For the sake of comparison, we follow the settings of the deep methods respectively and conduct experiments on Office-31 Datasets. As shown in Table 8, the method "SALFL + CAN" achieves the best performance and all the five methods combined with the SALFL framework have significant improvement compared with original deep methods which demonstrate that the SALFL with the deep domain-invariant feature is effective. Additionally, a noteworthy phenomenon is that our framework improves the shallow methods more obviously than the deep methods. The main reason is that the pseudo labels obtained by the deep methods are more confident which leads to higher classification accuracy, so that making the performance improvement is quite difficult. As can also be seen from the results, on the one hand, the improvement of the SALFL framework on DAN, DANN and CDAN is significantly higher than that on DANA and CAN. On the other hand, the improvements of deep methods with SALFL framework are significantly higher in tasks with poor performance such as $D \rightarrow A$ and $W \rightarrow A$ than in tasks with good performance such as $D \rightarrow W$ and $W \rightarrow D$. This also shows that our framework has better performance for challenging tasks.

4.4 Hypothesis testing

In order to further evaluate the performance of proposed SALFL, we conduct the hypothesis testing [50] on the results from Table 3 to Table 7. The Friedman test (Hypothesis testing) results of shallow and deep methods are shown in Fig. 4(a) and Fig. 4(b) respectively. We can observe that the proposed SALFL has obvious superiority and enjoys a large crucial distance to the secondly-ranked method (i.e. MEDA and CDAN).



(a) Friedman Test of the compared shallow methods



(b) Friedman Test of the compared deep methods

Fig. 4: Hypothesis test among the compared methods.

4.5 Ablation study

We conduct ablation study additionally to evaluate the effectiveness of the modules of the proposed SALFL. Specifically, compared with other shallow models, the innovation of our method is consisted of three main components: graph-

Table 6: Recognition accuracy (%) on MNIST + USPS with the pixel features.

	PCA	SA	GFK	TCA	TJM	JDA	BDA	JGSA	MEDA	DICD	DGA-DA	SALFL
USPS→MNIST	44.63	47.85	46.98	51.25	52.03	59.48	58.96	68.15	71.01	65.22	65.35	**72.32**
MNIST→USPS	66.75	61.91	67.44	56.23	63.89	67.70	69.32	80.37	81.53	77.55	79.15	**85.89**
Average	55.69	54.88	57.21	53.74	57.96	63.59	64.14	74.26	76.27	71.39	72.25	**79.11**

Table 7: Recognition accuracy (%) on VOC-2007 + MSRC with the pixel features.

	PCA	SA	GFK	TCA	TJM	JDA	BDA	JGSA	MEDA	DICD	DGA-DA	SALFL
MSRC→VOC-2007	30.16	31.24	31.21	31.87	32.09	31.42	33.11	34.14	38.17	35.7	36.58	**42.37**
VOC-2007→MSRC	41.86	47.73	45.17	46.17	46.94	43.80	43.85	49.52	59.9	58.47	58.20	**61.95**
Average	36.01	39.49	38.19	39.02	39.52	37.61	38.48	41.83	49.04	47.09	47.39	**52.16**

Table 8: Recognition accuracy (%) of SALFL framework with the deep methods on Office-31.

	A→W	D→W	W→D	A→D	D→A	W→A	Average
DAN	80.5	97.1	99.6	78.6	63.6	62.8	80.4
DANN	82.0	96.9	99.1	79.7	68.2	67.4	82.2
CDAN	93.1	98.2	**100.0**	89.8	70.1	68.0	86.6
DANA	92.3	99.2	**100.0**	93.9	74.4	74.2	89.0
CAN	**94.5**	99.1	99.8	95.0	78.0	77.0	90.6
SALFL + DAN	82.7	97.2	99.9	79.7	68.9	66.8	82.5
SALFL + DANN	84.3	97.6	99.3	82.9	70.9	71.4	84.4
SALFL + CDAN	93.1	99.2	**100.0**	90.2	73.2	70.2	87.7
SALFL + DANA	93.2	**99.6**	**100.0**	94.9	76.2	75.7	89.9
SALFL + CAN	94.4	99.3	99.9	**95.6**	**79.3**	**78.7**	**91.2**

based random walking, self-adaptive label filtering mechanism, and discriminative loss term. Therefore, we design three comparison models get rid of the tree terms respectively. Without loss of generality, we herein show the results on Office-10 + Caltech-10 in Fig. 5. As we can see that, each component of our method is significant, especially the graph-based random walking and self-adaptive label filtering mechanism.

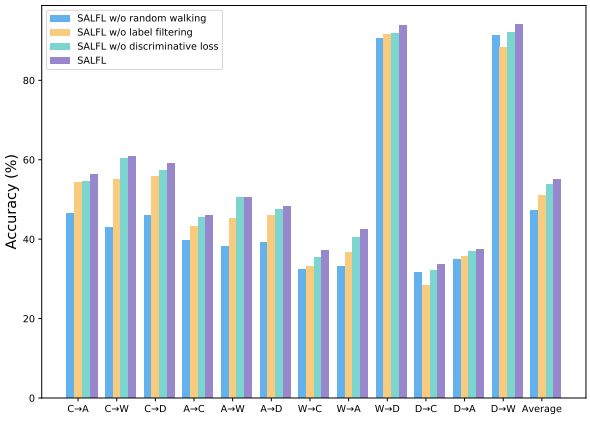


Fig. 5: Ablation study results on Office-10 + Caltech-10 with the SURF feature.

4.6 Parameter analysis

Finally, we analyze the sensitivity of the only one non-fixed parameter λ in SALFL. Due to space limitation, without loss of generality, we only report the results on "A→C" with SURF features, "A→C" with DeCAF$_6$ features, "Ar→Cl" with ResNet-50 features, "MNIST → USPS" with gray-scale pixel features and "VOC-2007 → MSRC" with Dense SIFT features and do not show the other similar results. As shown in Fig. 6, the performance of SALFL is sensitive to the regularization parameter λ and achieves better performance generally with λ between 0.01 and 10, which provides a reference in practice.

5 Conclusion

In this article, we proposed a self-adaptive label filtering learning (SALFL) method for unsupervised domain adaptation. On the one hand, SALFL leverages knowledge to learn an effective UDA model by combining geometrical graph-based random walking and statistical distribution measure with class-wise discriminative term. On the other hand, SALFL enhances the quality of pseudo-labels by the self-adaptive label filtering mechanism which solves the issue that hard labels may be overconfident and soft labels tend to be

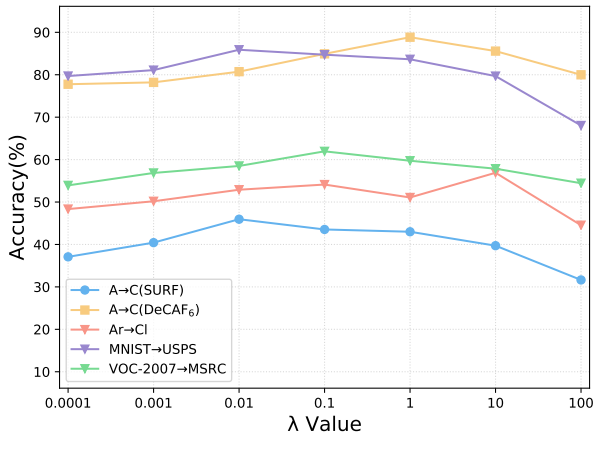


Fig. 6: Parameter sensitivity of λ.

confusing. Comprehensive experiments demonstrate the performance superiority of the proposed SALFL to state-of-the-art UDA approaches.

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grants 62176128 and 61702273, National key research and development program of China No.2021YFE0104400, the Natural Science Foundation of Jiangsu Province under Grant BK20170956, the Open Projects Program of National Laboratory of Pattern Recognition under Grant 202000007, the Fundamental Research Funds for the Central Universities No. NJ2019010, the Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund, the Postgraduate Research & Practice Innovation Program of Jiangsu Province KYCX21_1006, and was also sponsored by the Qing Lan Project.

References

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

[2] Pan S J, Yang Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 2009, 22(10):1345–1359.

[3] Tian Q, Ma C, Cao M, Chen S, Yin H. A convex discriminant semantic correlation analysis for cross-view recognition. *IEEE Transactions on Cybernetics*, DOI:10.1109/TCYB.2020.2988721.

[4] Zhu Y, Zhuang F, Wang J, Ke G, Chen J, Bian J, Xiong H, He Q. Deep subdomain adaptation network for image classification. *IEEE Transactions on Neural Networks and Learning Systems*, DOI: 10.1109/TNNLS.2020.2988928.

[5] Inoue N, Furuta R, Yamasaki T, Aizawa K. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5001–5009.

[6] Yao Z, Wang Y, Long M, Wang J, Philip S Y, Sun J. Multi-task learning of generalizable representations for video action recognition. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, 2020, pp. 1–6.

[7] Zhuang F, Luo P, Du C, He Q, Shi Z, Xiong H. Triplex transfer learning: Exploiting both shared and distinct concepts for text classification. *IEEE transactions on cybernetics*, 2013, 44(7):1191–1203.

[8] Guo H, Pasunuru R, Bansal M. Multi-source domain adaptation for text classification via distancenet-bandits. In *AAAI*, 2020, pp. 7830–7838.

[9] Long M, Wang J, Ding G, Sun J, Yu P S. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2200–2207.

[10] Wang J, Chen Y, Hao S, Feng W, Shen Z. Balanced distribution adaptation for transfer learning. In *2017 IEEE International Conference on Data Mining (ICDM)*, 2017, pp. 1129–1134.

[11] Zhang J, Li W, Ogunbona P. Joint geometrical and statistical alignment for visual domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1859–1867.

[12] Yang L, Jin R. Distance metric learning: A comprehensive survey. *Michigan State Universiy*, 2006, 2(2):4.

[13] Blitzer J, Crammer K, Kulesza A, Pereira F, Wortman J. Learning bounds for domain adaptation. In *Advances in neural information processing systems*, 2008, pp. 129–136.

[14] Ding Z, Fu Y. Robust transfer metric learning for image classification. *IEEE Transactions on Image Processing*, 2016, 26(2):660–670.

[15] Courty N, Flamary R, Habrard A, Rakotomamonjy A. Joint distribution optimal transportation for domain adaptation. In *Advances in Neural Information Processing Systems*, 2017, pp. 3730–3739.

[16] Gretton A, Borgwardt K, Rasch M, Schölkopf B, Smola A J. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, 2007, pp. 513–520.

[17] Borgwardt K M, Gretton A, Rasch M J, Kriegel H P, Schölkopf B, Smola A J. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics*, 2006, 22(14):49–57.

[18] Pan S J, Tsang I W, Kwok J T, Yang Q. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 2010, 22(2):199–210.

[19] Luo L, Chen L, Hu S, Lu Y, Wang X. Discriminative and geometry-aware unsupervised domain adaptation. *IEEE Transactions on Cybernetics*, 2020.

[20] Long M, Wang J, Ding G, Pan S J, Philip S Y. Adaptation regularization: A general framework for transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 2013, 26(5):1076–1089.

[21] Li S, Song S, Huang G, Ding Z, Wu C. Domain invariant and class discriminative feature learning for visual domain adaptation. *IEEE Transactions on Image Processing*, 2018, 27(9):4260–4273.

[22] Ding Z, Li S, Shao M, Fu Y. Graph adaptive knowledge transfer for unsupervised domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 37–52.

[23] Gong B, Shi Y, Sha F, Grauman K. Geodesic flow kernel for unsupervised domain adaptation. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2066–2073.

[24] Fernando B, Habrard A, Sebban M, Tuytelaars T. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2960–2967.

[25] Pan S J, Kwok J T, Yang Q et al. Transfer learning via dimensionality reduction. In *AAAI*, volume 8, 2008, pp. 677–682.

[26] Long M, Wang J, Ding G, Sun J, Yu P S. Transfer joint matching for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1410–1417.

[27] Weinberger K Q, Sha F, Saul L K. Learning a kernel matrix for nonlinear dimensionality reduction. In *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 106.

[28] Xie S, Zheng Z, Chen L, Chen C. Learning semantic representations for unsupervised domain adaptation. In *International conference on machine learning*, 2018, pp. 5423–5432.

[29] Pei Z, Cao Z, Long M, Wang J. Multi-adversarial domain adaptation. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[30] Zhu Y, Zhuang F, Wang J, Chen J, Shi Z, Wu W, He Q. Multi-representation adaptation network for cross-domain image classification. *Neural Networks*, 2019, 119:214–221.

[31] Long M, Cao Z, Wang J, Jordan M I. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*, 2018, pp. 1640–1650.

[32] Gallagher B. Matching structure and semantics: A survey on graph-based pattern matching. In *AAAI Fall Symposium: Capturing and Using Patterns for Evidence Detection*, 2006, pp. 45–53.

[33] Wang J, Feng W, Chen Y, Yu H, Huang M, Yu P S. Visual domain adaptation with manifold embedded distribution alignment. In *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 402–410.

[34] Zou Y, Yu Z, Liu X, Kumar B, Wang J. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5982–5991.

[35] Kang G, Jiang L, Yang Y, Hauptmann A G. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4893–4902.

[36] Chen C, Xie W, Huang W, Rong Y, Ding X, Huang Y, Xu T, Huang J. Progressive feature alignment for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 627–636.

[37] Wold S, Esbensen K, Geladi P. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 1987, 2(1-3):37–52.

[38] Hestenes M R. Multiplier and gradient methods. *Journal of optimization theory and applications*, 1969, 4(5):303–320.

[39] Saenko K, Kulis B, Fritz M, Darrell T. Adapting visual category models to new domains. In *European conference on computer vision*, 2010, pp. 213–226.

[40] Griffin G, Holub A, Perona P. Caltech-256 object category dataset. 2007.

[41] Venkateswara H, Eusebio J, Chakraborty S, Panchanathan S. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5018–5027.

[42] Hull J J. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, 1994, 16(5):550–554.

[43] LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998, 86(11):2278–2324.

[44] Huang Y, Huang K, Yu Y, Tan T. Salient coding for image classification. In *CVPR 2011*, 2011, pp. 1753–1760.

[45] Long M, Ding G, Wang J, Sun J, Guo Y, Yu P S. Transfer sparse coding for robust image representation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 407–414.

[46] Tang H, Jia K. Discriminative adversarial domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 2020, pp. 5940–5947.

[47] Long M, Cao Y, Wang J, Jordan M. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, 2015, pp. 97–105.

[48] Ganin Y, Ustinova E, Ajakan H, Germain P, Larochelle H, Laviolette F, Marchand M, Lempitsky V. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 2016, 17(1):2096–2030.

[49] Long M, Zhu H, Wang J, Jordan M I. Deep transfer learning with joint adaptation networks. In *International conference on machine learning*, 2017, pp. 2208–2217.

[50] Rust P F. Statistical methods of analysis. *Journal of the American Statistical Association*, 2003, 100(471):1094–1095.