Guide Subspace Learning for Unsupervised Domain Adaptation

Lei Zhang, *Senior Member, IEEE*, Jingru Fu, Shanshan Wang, *Student Member, IEEE*, David Zhang, *Fellow, IEEE*, ZhaoYang Dong, *Fellow, IEEE*, C.L. Philip Chen, *Fellow, IEEE*

***Abstract*—A prevailing problem in many machine learning tasks is that the training (i.e. source domain) and test data (i.e. target domain) have different distribution (i.e. non i.i.d). Unsupervised domain adaptation (UDA) was proposed to learn the unlabeled target data by leveraging the labeled source data. In this paper, we propose a Guide Subspace Learning (GSL) method for UDA, in which an invariant, discriminative and domain agnostic subspace is learned by three guidance terms through a two-stage progressive training strategy. First, the subspace-guided term reduces the discrepancy between domains by moving the source closer to the target subspace. Second, the data-guided term uses the coupled projections to map both domains to a unified subspace, where each target sample can be represented by the source samples with a low-rank coefficient matrix that can preserve the global structure of data. In this way, the data from both domains can be well interlaced and the domain-invariant features can be obtained. Third, for improving the discrimination of the subspaces, the label-guided term is constructed for prediction based on source labels and pseudo-target labels. To further improve the model tolerance to label noise, a label relaxation matrix is introduced. For the solver, a two-stage learning strategy with *teacher teaches and student feedbacks* mode is proposed to obtain the discriminative domain-agnostic subspace. Additionally, for handling nonlinear domain shift, a nonlinear guide subspace learning (NGSL) framework is formulated with kernel embedding, such that the unified subspace is imposed with nonlinearity. Experiments on various cross-domain visual benchmark databases show that our methods outperform many state-of-the-art UDA methods. The source code is available at https://github.com/Fjr9516/GSL.**

***Index Terms*—Subspace Learning, Domain Adaptation, Transfer Learning**

I. Introduction

CONVENTIONAL machine learning algorithms are established by supposing that the training and test data lie in the same feature space with independent identical distribution (i.i.d.) [1]. However, this assumption generally does not hold in many real-world scenarios. In computer vision areas, owing to various factors such as camera device parameter, illuminations, viewpoint, background, etc., the visual datasets show serious distribution mismatch and domain shift [2], [3], [4], [5], such that a dramatic performance drop of the conventional machine learning methods is caused. The domain mismatch of images is visually described in Fig. 1 and the semantic similarity but domain mismatch is clearly shown. A general strategy for pattern recognition is to sampling a large number of labeled data from a specific domain and train a generalized classifier. However, labeling a large number of data specially consumes a lot of human resources, and this is cost-ineffective and even unrealistic in many areas. Therefore, how to achieve cross-domain learning by leveraging another distribution different but semantic related domain is becoming an increasingly important topic. Recently, transfer learning/domain adaptation methods have been proposed to relieve such domain mismatch problem [6], [7] by transferring the rich knowledge from the source domain (training set) to the target domain (test set). In this paper, we address a unsupervised domain adaptation scenario where the labels of target domain are unavailable.

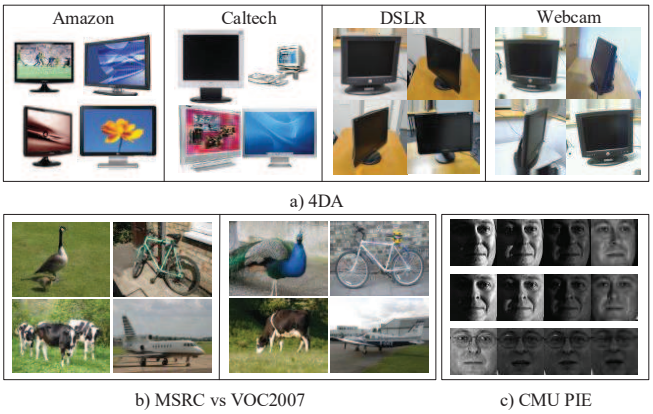

Fig. 1. Some examples from different domains. a) 4DA object dataset. b) MSRC (left) and VOC2007 (right) image dataset. c) CMU PIE face dataset with different illuminations, poses, expressions and occlusion.

Conventional TL/DA methods can be divided into classifier-based methods and representation-based methods [6], [8]. The *classifier-based* methods tend to solve the domain disparity problem by adapting classifiers to the data of different distributions, such as A-SVM [9], DSM [10], and DTMKL [11]. These classifier adaptation methods achieved a success and promoted the progress of domain adaptation. However, they may not well utilize the intrinsic structure of the data but strongly

This work is supported by National Natural Science Fund of China (Grant 61771079), Chongqing Natural Science Fund (No. cstc2018jcyjAX0250) and Chongqing Youth Talent Program. (*Corresponding author: Lei Zhang*).

L. Zhang, J. Fu, and S. Wang are with the School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China. (E-mail: leizhang@cqu.edu.cn, jrfu@cqu.edu.cn, wangshanshan@cqu.edu.cn).

D. Zhang is with School of Science and Engineering, Chinese University of Hong Kong (Shenzhen), Shenzhen 518172, China. (E-mail: csdzhang@comp.polyu.edu.hk).

Z.Y. Dong is with School of Electrical Engineering and Telecommunications, The University of New South Wales, Sydney, NSW 2052, Australia. (E-mail: Joe.Dong@unsw.edu.au).

C. L. Philip Chen is with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau 99999, China. (E-mail: philip.chen@ieee.org).

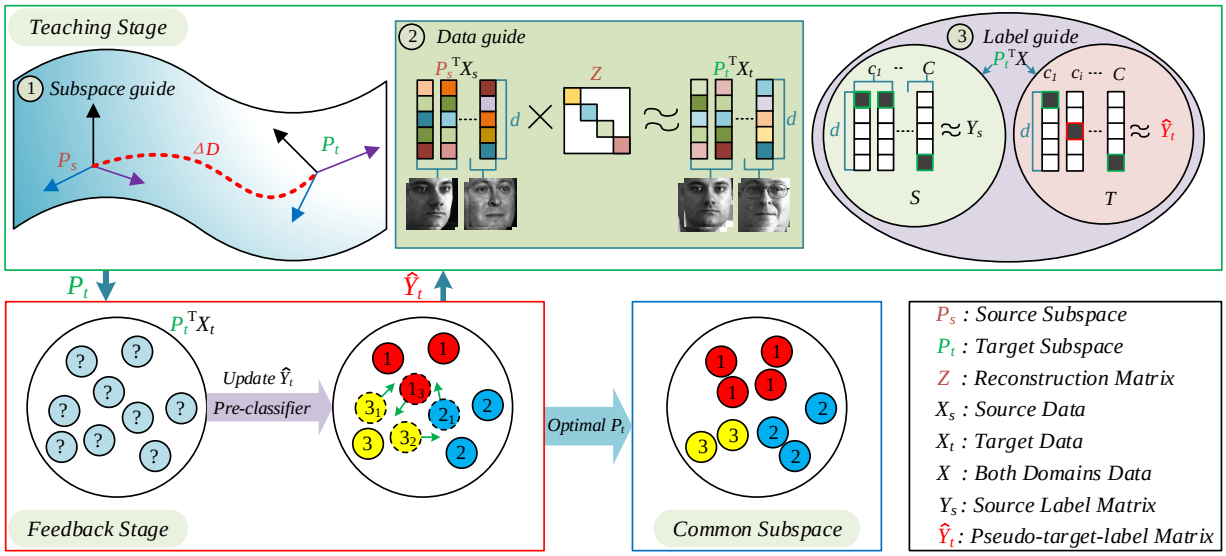

Fig. 2. Schematic of the proposed GSL method, which includes two stages: tutor teaching and student feedback. In teaching stage, the domain disparity is progressively removed via three guiding steps: subspace guidance, data guidance and label guidance, then an invariant and discriminative target projection P_t is obtained. In feedback stage, the P_t from the teaching stage converts the target domain data into discriminative features, and obtains the pseudo-target-label through the pre-classifier. The progressively correct pseudo labels are then feeding back to the teaching stage. Each circle represents a sample, the number l_c in the dotted circle represents that the sample from class c is misclassified as class l. With alternative learning of the two stages, a common and optimal target projection is obtained, where the target data can be well classified (*the student surpasses the master*).

depend on a certain type of classifier, and cannot easily adapt to other classifier types. Therefore, in recent years researchers focus on *representation-based* methods, which tend to learn a better domain invariant and agnostic feature by exploiting the intrinsic structure of data, such as RDALR [12], TSL [13], LTSL [8], LSDT [14], DTLC [15], MCTL [16] and CDSL [17]. However, these methods have not well exploited the label information, that is helpful to improve the classification performance. Therefore, the algorithms that combine the advantages of both classifier and representation-based models were proposed, such as DTSL [18], MMDT [19], EMFS [20] and CoE [21], which then effectively improve the discrimination of transfer models. However, the existing methods only consider the domain adaptation at the data level, which are sensitive to noise (e.g. outliers). Also, another problem is that these methods share a common projection matrix for both domains, which may not effectively measure the difference between the domain-specific subspaces. SA [22] and CORAL [23] stand in another perspective, by aligning the first-order and second-order statistical global features such as principal component space and covariance space, respectively. Further, a common characteristic of these classifier and representation based TL/DA methods is the "one-stage" formulation, which is generally hard to get the optimal projection and does not work well if the domain disparity is large.

Considering the above problems of the existing methods, in this paper we propose a couple projection and soft label based domain adaptation model for domain agnostic subspace learning with guide learning mechanism. Specifically, in order to guide the subspace projection towards extracting the invariant, discriminative and common features across domains, we propose a "two-stage" progressive training strategy for unsupervised domain adaptation. The proposed "two-stage" domain adaptation model consists of *teaching* and *feedback* stages, which follows that the experienced tutor not only transfers/teaches domain knowledge to the students, but also progressively guide the students toward a high-level performance according to the students' feedback. The teaching stage is composed of three guidance losses: subspace guidance, data guidance and label guidance. The feedback stage aims to improve the model transferring capability by progressively learning better subspace and soft target labels. Therefore, the proposed transfer model is called Guide Subspace Learning (GSL). For clarity, the *Guide Learning* mechanism with teaching and feedback stages is defined as follows.

Definition (Guide Learning) *Guide learning is a learning mechanism consisting of teacher teaching and student feedback, and the students' learning performance can be gradually improved with the guidance of feedback information.*

Concretely, the schematic of GSL is shown in Fig. 2, from which we see that in the teaching stage we learn a target projection (i.e., P_t) through the three guidance terms and in the feedback stage we update the "feedback information" (i.e., $\hat{Y}_t$) under the new target subspace, and then transfer the updated feedback information to the teaching step.

In summary, the key contributions of this work are threefold:

- We propose a Guide Subspace Learning (GSL) method for unsupervised domain adaptation, which consists of three elements: subspace guidance, data guidance and label guidance. Then an invariant, discriminative and common target subspace is obtained by a "two-stage" guide learning mechanism. To the best of our knowledge, this is the first work for domain adaptation and transfer learning by formulating a guide learning model.
- In order to overcome the nonlinear domain shift problem, we further generalize the proposed GSL method into a

kernel-based nonlinear framework in reproduced kernel Hilbert space (RKHS), and a nonlinear guided subspace learning (NGSL) is formulated.
- Our model can be easily adapted to supervised and semi-supervised settings by freely replacing the pseudo target labels with true labels, and then degenerates into a "one-stage" method. Extensive experiments show the superiority of the proposed GSL over state-of-the-arts.

The remainder of this paper is organized as follows. Section II presents a brief review of related work. The proposed guide subspace learning (GSL) method is formulated in Section III. The proposed nonlinear GSL (NGSL) is presented in Section IV. The experiments and discussions are presented in Section V. Finally, Section VI concludes the paper.

II. Related Work

A. Active learning/Curriculum learning/Self-paced learning

Active learning (AL) focuses on actively selecting and annotating the most informative unlabeled samples from the data pool, towards avoiding redundant annotations and achieving a high classification performance [24], [25]. It is generally used in semi-supervised tasks as shown in the Fig. 3 a). Inspired by the cognitive principle of humans/animals, the curriculum learning (CL) proposed by Bengio et al. [26] tends to train the model by including the samples from simple to complex progressively. However, a problem of CL is encountered due to the difficult identification of simple and complex samples in a given training dataset with human intervention. In order to alleviate the deficiency, Kumar et al. [27], Lu et al. [28], [29], and Jiang et al. [30] introduced self-paced learning (SPL) models which simultaneously selects easy samples and iteratively updates the parameters in a progressive manner. It is often used in supervised tasks as illustrated in Fig. 3 b). The proposed GSL is essentially different from them in learning tasks and methodologies. Specifically, as shown in Fig. 3 c), the proposed GSL aims to solve unsupervised domain adaptation tasks by learning a transferrable subspace across domains with progressively-qualified pseudo target labels, while AL/CL/SPL aim to solve generic machine learning problems by handling the samples with different strategies.

B. DA/TL Methods

This work focuses on the feature-based DA/TL methods that learn a common representation for cross-domain classification. To this end, we divide the representational DA/TL methods into two categories: data level and subspace level.

- **Data level approaches**

Data level denotes the domain shift alignment between the source and target domains through feature representation in the raw or projected space. Under the low-rank representation theory [31], [32], Jhuo et al. [12] proposed a RDALR method by imposing a low-rank constraint on the cross-domain feature reconstruction matrix to reduce the domain shift, such that the intrinsic relationship in data is interpreted. Shao et al. [8] also proposed a low-rank constraint derived LTSL method for transfer learning, in which a subspace projection was learned for better cross-domain low-rank modeling. Differently, Zhang and Zhang [14] proposed a latent sparse domain transfer (LSDT) model by imposing sparsity constraint on the reconstruction matrix in a projected data space, such that outliers in source data can be effectively prohibited from transferring to target domain. Further, Fang et al. proposed a DTSL [18] model by imposing both low-rank and sparse constraints on the domain reconstruction matrix to guarantee the global and local property during transfer. Besides the feature reconstruction methods, Si et al. [13] proposed transfer subspace learning (TSL) for solving the domain mismatch problem by minimizing the Bregman divergence between domain distributions in a common subspace. Hoffman et al. [19] proposed a MMDT method by jointly combining the classifier-based and representation-based methods for invariant image representation. A flaw of these data level methods is that they depend heavily on the reconstruction and transformation matrix, which easily causes negative transfer effect.

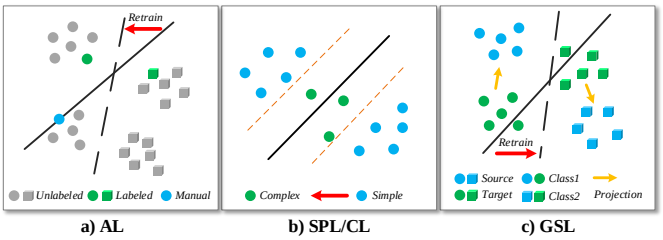

Fig. 3. Comparison with other learning mechanisms. Black solid lines and dashed lines indicate classification boundaries. a) AL aims to find the most informative sample points for manual labeling; b) SPL/CL aims to gradually incorporate samples from simple to complex for training; c) GSL is applied to unsupervised domain adaptation, and it progressively learns the subspace projection to reduce domain discrepancy. (Best viewed in colors)

- **Subspace level approaches**

Subspace level approaches tend to align the statistical feature distribution of two domains. Subspace alignment (SA) [22] aims at learning a linear mapping for aligning subspaces spanned by eigenvectors using principal component analysis (PCA) between two different domains. These principal components show global information of domains with noise removal, and therefore the subspace level approach becomes more robust. It is worth mentioning that SA can be interpreted from the *Grassmann* manifold perspective. Sun and Saenko [33] proposed a subspace distribution alignment (SDA) method for reducing the subspace distribution difference, which also proved that SA can be extended to geodesic flow kernel (GFK) [34] in the case of an infinite subspaces distribution alignment. GFK characterized the changes of geometric and statistical properties across domains by integrating numerous subspaces. Pan et al. [35] proposed a transfer component analysis (TCA) method, which exploits the Maximum Mean Discrepancy (MMD) to measure the difference between domains and derives a simple solution via eigenvalue decomposition. Besides the first-order statistical information, Sun et al. [23] proposed a CORAL method for alleviating the domain shift by aligning the second-order statistics (e.g. covariance) between two domains. Long et al. [36] proposed a joint distribution alignment (JDA) method by adapting both the

marginal distribution and conditional distribution, and simultaneously exploring the pseudo labels of the target data. Wang et al. [37] proposed a balanced distribution adaptation (BDA) by simply introducing a parameter to measure different domain distributions. Zhang et al. [38] proposed a joint geometrical and statistical alignment (JGSA) method, which considered the geometrical shift and distribution shift simultaneously.

III. The Proposed Guide Subspace Learning

A. Mathematical Notations

Given the source domain $\mathcal{S} = \{X_s, y_s\}$ and target domain $\mathcal{T} = \{X_t, y_t\}$, where $X_s \in \mathbb{R}^{D\times n_s}$ and $X_t \in \mathbb{R}^{D\times n_t}$ are domain-specific datasets, y_s and y_t are class labels (note that the target label y_t is unavailable for unsupervised setting), D is the dimensionality of the raw data, and n_s and n_t indicate the number of source and target samples, respectively. Let $P_s \in \mathbb{R}^{D\times d}$ and $P_t \in \mathbb{R}^{D\times d}$ be the projection of the source and target domain, respectively, where d is the dimensionality of the invariant and unified subspace. Define $Z \in \mathbb{R}^{n_s\times n_t}$ as the reconstruction matrix across domains, I as the identity matrix, and $\|M\|_F$ and $\|M\|_\infty$ are the Frobenius norm and infinity norm of matrix M, respectively. Let $\|M\|_* = \sum_i \delta_i(M)$ denote the nuclear norm of matrix M, where $\delta_i(M)$ is the i^{th} singular value of matrix M. $tr(\cdot)$ denotes the trace operator of matrix.

B. Model Formulation

In order to achieve domain adaptation, it is often assumed that there is a common subspace, where the distributions of the two domains are approximately the same. We follow this assumption but suppose that there exist two projections P_s and P_t, instead of one common projection. The proposed GSL aims at learning an invariant, discriminative, and domain agnostic subspace, in which the domain distribution discrepancy is reduced. Specifically, GSL is composed of three parts in the teaching stage: 1) subspace guidance; 2) data guidance; 3) label guidance, which are elaborated as follows.

1) Subspace Guidance: The advantage of couple subspaces is that they can better characterize the global information of the domain and improve the robustness to noise (e.g. outliers). For clarity, we define P_s and P_t as the subspace projection of the source and target domain, respectively. They are imposed to be orthogonal, since they can be thought as two points in the *Grassmann* manifold $\mathbb{G}(d, D)$ [39]. Therefore, the domain discrepancy can be reduced by moving the two points closer as shown in Fig. 2. Specifically, we expect that the source subspace P_s guides the learning of the target subspace P_t in an interactive manner, such that the subspaces of the two domains can be effectively aligned for reducing domain disparity. Instead of learning an additional transformation as other subspace level methods did, our method directly minimizes the following Bregman divergence between subspaces:

$$\min_{P_s,P_t} \|P_s - P_t\|_F^2 \tag{1}$$

We see from Eq.(1) that it has a simple yet effective mathematical formulation and treats the two subspaces equally. The orthogonality of subspaces can often be initialized by existing subspace learning models (e.g. PCA).

2) Data Guidance: In order to narrow the distribution gap between source and target domains, we expect to use the intrinsic information of data to guide the learning of target subspace P_t. The data reconstruction between domains can effectively reflect the intrinsic information of the data. Therefore, in data guidance, we tend to seek an invariant subspace by forcing the target data to be linearly represented by source data, such that the domain distribution gap is minimized. For revealing and interpreting the underlying structure of domain data, we require that each target data can be reconstructed by its similar neighbors in the source domain. Mathematically, the objective can be achieved by imposing a low-rank constraint on the reconstruction matrix Z. Low-rank has been extensively discussed in machine learning community due to its impact on subspace recovery [31]. Specifically, the data guidance can be formulated as follows

$$\min_{P_s,P_t,Z} \left\|P_t^T X_t - P_s^T X_s Z\right\|_F^2 + \alpha \cdot rank(Z) \tag{2}$$

where $rank(\cdot)$ denotes the rank operator of a matrix. However, due to the non-convex property of rank function, an effective solution is not easy to be optimized directly. Therefore, we obtain a tractable optimization problem by relaxing the problem and replacing the rank with the nuclear norm [31], [40], and yield the following convex surrogate as

$$\min_{P_s,P_t,Z} \left\|P_t^T X_t - P_s^T X_s Z\right\|_F^2 + \alpha \|Z\|_* \tag{3}$$

where $\|\cdot\|_*$ represents the nuclear norm that is computed as the summation of singular values of a matrix. The parameter α controls the intrinsic correlation of the reconstruction matrix. By combining the Eq.(3) with the Eq.(1), an invariant target subspace with domain disparity reduction can be obtained.

3) Label Guidance: Although an invariant subspace can be obtained by solving problem (3), the subspace discrimination is still weak and does not benefit the classification due to that the rich label information of source data is neglected. For label distribution alignment, another important idea is that although the true target label is unavailable, the pseudo-labels can be generated through implicit student feedback. Therefore, we further introduce label guidance strategy to improve the discrimination by considering the source labels and pseudo target labels (feedback information), that can bridge the feedback stage and the teaching stage.

We expect that the learned invariant target subspace projection P_t can also serve as a label mapping function by forcing $P_t^T X_t$ to be close to the pseudo label matrix $\hat{Y}_t \in \mathbb{R}^{d\times n_t}$ ($d \geq c$, where c indicates the number of classes) with category information. For unification of the subspace P_t in both domains, we also force $P_t^T X_s$ to be close to the source label matrix Y_s. For obtaining pseudo target labels, we propose to initialize the pseudo labels by using existing classifiers (e.g. SVM) and then update the labels with the progressive learning of target subspace. Consequently, a discriminative target subspace can be learned progressively by alternatively updating P_t and $\hat{Y}_t$. Further, considering that each sample shows different classification confidence reflected in the coding value of the label, some varying degree of labels is allowed for improving the generalization. Therefore, we introduce a label relaxation

matrix M to alleviate the effect of label consolidation while increasing the flexibility of the model. For convenience, we define the label matrix $Y = \left[Y_s, \hat{Y}_t\right] \in \mathbb{R}^{d\times n}$ as:

$$Y_{\{i,j\}} = \begin{cases} 1, & if\ x_j \in c_i \\ -1, & otherwise \end{cases} \tag{4}$$

where $Y_{\{i,j\}}(i = 1, \cdots, d; j = 1, \cdots, n)$ denotes the $\{i, j\}$-th element of the matrix Y with one-hot coding, $n = n_s + n_t$ indicates the total number of samples in both domains, and c_i represents the cluster of the i^{th} class. Note that we assume $d \geq c$. The reason is that if we modeling the task as a c-dimensional problem, too much information will be lost when c is too small. Note that the entry of Y beyond the c dimension is set to -1 as shown in Eq.(4).

The purpose of label guidance strategy is to seek a discriminative P_t by using label relaxation and progressive pseudo-label updating mechanism. Specifically, the proposed label guidance is mathematically formulated as

$$\min_{P_t, M} \left\|P_t^T X - Y \circ M\right\|_F^2 \quad s.t.\ M \succcurlyeq 0 \tag{5}$$

where $X = [X_s, X_t] \in \mathbb{R}^{D\times n}$, $M \in \mathbb{R}^{d\times n}$ represents the positive relaxation matrix, and $\circ$ is the hadamard product operator. Actually, the label guidance strategy makes the classification task more conducive by progressively and alternatively improving the pseudo target label quality and the unified target subspace. Also, the label distribution across domains can be aligned in this work in addition to aligning the feature distribution. To some extent, the introduction of a relaxation matrix M also allows a certain label noise and further improve the model generalization.

Finally, by incorporating the three terms Eq.(1), Eq.(3) and Eq.(5) together, we obtain the ultimate objective function of the proposed GSL model, which is formulated as

$$\begin{aligned} \min_{P_s, P_t, M, Z} & \beta \left\|P_s - P_t\right\|_F^2 + \left\|P_t^T X_t - P_s^T X_s Z\right\|_F^2 \\ & + \alpha \left\|Z\right\|_* + \frac{1}{2}\left\|P_t^T X - Y \circ M\right\|_F^2 \\ & s.t.\ M \succcurlyeq 0 \end{aligned} \tag{6}$$

where β and α are trade-off parameters. We progressively update the pseudo labels $\hat{Y}_t$ of target data using the learned invariant and discriminative target subspace P_t (teaching stage) and update the target subspace P_t using the new pseudo labels $\hat{Y}_t$ (feedback stage). In the following, the solution is shown.

C. Optimization

As can be seen from the model in Eq.(6), four variables P_s, P_t, Z and M are involved. Note that Y is a semi-variable that can be computed using SVM classifier. To solve the model, an inexact augmented Lagrange multiplier method (IALM) [31] is introduced. In general, by introducing an auxiliary variable L, the problem (6) can be converted into:

$$\begin{aligned} \min_{P_s, P_t, M, Z, L} & \beta \left\|P_s - P_t\right\|_F^2 + \left\|P_t^T X_t - P_s^T X_s Z\right\|_F^2 + \alpha \left\|L\right\|_* \\ & + \frac{1}{2}\left\|P_t^T X - Y \circ M\right\|_F^2 \\ & s.t.\ M_{\{i,j\}} \geq 0,\ Z = L \end{aligned} \tag{7}$$

Algorithm 1 The first stage of GSL for solving problem (8)

Input: $X_s \in \mathbb{R}^{D\times n_s}$, $X_t \in \mathbb{R}^{D\times n_t}$, $X \in \mathbb{R}^{D\times n}$, $Y \in \mathbb{R}^{d\times n}$, α, β

Output: P_t

Initialization: $M = \mathbf{1}$, $Z = L = \mathbf{0}$, $Y_1 = \mathbf{0}$, $\mu_{max} = 10^6$, $\rho = 1.01$, $\epsilon = 10^{-7}$;

Initialize P_s via existing method, e.g., PCA;

While not converged **do**

Fix other variables and update P_t by solving (10);

Fix other variables and update P_s by solving (11);

Fix other variables and update Z by solving (13);

Fix other variables and update L by solving (18);

Fix other variables and update M by solving (21);

Update multiplier Y_1 and penalty parameter μ by (22);

Check convergence: $\left\|Z - L\right\|_\infty < \epsilon$.

End while

return P_t

The augmented Lagrange multipliers (ALM) method is generally used for solving the nuclear norm optimization problems [40], [41], and the problem (7) can be reformulated as:

$$\begin{aligned} \mathcal{L}_{P_s, P_t, Z, L, M} = & \beta \left\|P_s - P_t\right\|_F^2 + \left\|P_t^T X_t - P_s^T X_s Z\right\|_F^2 \\ & + \alpha \left\|L\right\|_* + \frac{1}{2}\left\|P_t^T X - Y \circ M\right\|_F^2 \\ & + tr(Y_1^T (Z - L)) + \frac{\mu}{2}\left\|Z - L\right\|_F^2 \end{aligned} \tag{8}$$

where Y_1 denotes the Lagrange multiplier and $\mu > 0$ is a penalty parameter. Specifically, by using variable alternating strategy, the detailed solution of each variable in the proposed GSL model can be derived as follows.

- **Update** P_t:

For solving P_t, by fixing the irrelevant terms with respect to P_t, we can have the following convex model:

$$\begin{aligned} P_t = \arg\min_{P_t} & \beta \left\|P_s - P_t\right\|_F^2 + \left\|P_t^T X_t - P_s^T X_s Z\right\|_F^2 \\ & + \frac{1}{2}\left\|P_t^T X - Y \circ M\right\|_F^2 \end{aligned} \tag{9}$$

By setting the derivative of the above model with respect to P_t to be zero, a closed-form solution P_t^* can be solved as:

$$P_t^* = (2\beta I + 2X_t X_t^T + X X^T)^{-1}(2\beta P_s + 2X_t Z^T X_s^T P_s + A_1) \tag{10}$$

where $A_1 = X(Y \circ M)^T$ is a pre-computed matrix.

- **Update** P_s:

Similar to the solving process of P_t, with other variables frozen , the model is differentiable to P_s. Therefore, by setting the derivative with respect to P_s as zero, the closed-form solution can be derived as:

$$P_s^* = (2\beta I + 2X_s Z Z^T X_s^T)^{-1}(2\beta P_t + 2X_s Z X_t^T P_t) \tag{11}$$

- **Update** Z:

By dropping those terms without containing the variable Z in Eq.(8), the problem with respect to Z becomes

$$Z = \arg\min_{Z} \left\|P_t^T X_t - P_s^T X_s Z\right\|_F^2 + tr(Y_1^T(Z-L)) + \frac{\mu}{2}\left\|Z-L\right\|_F^2 \tag{12}$$

By setting the derivative of Eq.(12) with respect to Z as zero, the closed-form solution of (12) can be computed as:

$$Z^* = (2Xs^T P_s P_s^T X_s + \mu I)^{-1}(2Xs^T P_s P_t^T X_t + \mu A_2) \tag{13}$$

where $A_2 = L - Y_1/\mu$ is a pre-computed matrix.

- **Update** L:

By removing the terms irrelevant to the variable L in Eq.(8), the following formulation can be deduced:

$$L = \arg\min_{L} \alpha\left\|L\right\|_* + \frac{\mu}{2}\left\|L-(Z+Y_1/\mu)\right\|_F^2 \tag{14}$$

The optimal solution of problem (14) can be computed via the singular value thresholding (SVT) algorithm [42]. Specifically, given a matrix $Q \in \mathbb{R}^{n_1 \times n_2}$ with rank r, the singular value decomposition (SVD) of matrix Q is:

$$Q = U\Sigma V^T \tag{15}$$

where $\Sigma = diag((\sigma_i)_{1 \leqslant i \leqslant r})$, U and V represent $n_1 \times r$ and $n_2 \times r$ matrices with orthogonal column vectors, σ_i denotes the positive singular value, and $diag(\cdot)$ denotes the diagonal operator of matrix. Given $\tau \geqslant 0$, we introduce the singular value shrinkage operator as follows [42]:

$$\mathcal{D}_\tau(Q) = U\mathcal{D}_\tau(\Sigma)V^T,\ \mathcal{D}_\tau(\Sigma) = diag((\sigma_i - \tau)_+) \tag{16}$$

where $(t)_+ = \max(0,t)$ denotes the positive value operator. The solution of problem (14) can be derived in Theorem 1.

Theorem 1. For each $\tau \geqslant 0$ and $P \in \mathbb{R}^{n_1 \times n_2}$, the singular value shrinkage operator in Eq.(16) obeys [42]:

$$\mathcal{D}_\tau(P) = \arg\min_{Q} \tau\left\|Q\right\|_* + \frac{1}{2}\left\|Q-P\right\|_F^2 \tag{17}$$

According to *Theorem 1*, the optimal solution of problem (14) can be easily derived as:

$$L^* = \mathcal{D}_{\frac{\alpha}{\mu}}(Z + Y_1/\mu) \tag{18}$$

where the operator $\mathcal{D}_{\frac{\alpha}{\mu}}(\cdot)$ can be computed by using Eq.(16).

- **Update** M:

By removing the terms irrelevant to M in Eq.(8), we have:

$$M = \arg\min_{M} \frac{1}{2}\left\|P_t^T X - Y \circ M\right\|_F^2,\quad s.t.\ M_{\{i,j\}} \geq 0 \tag{19}$$

By defining $A_3 = P_t^T X$ and considering the optimization problem element by element with $M_{\{i,j\}}$, the above problem (19) can be further written as:

$$M_{\{i,j\}} = \min_{M_{\{i,j\}}} \frac{1}{2}(A_{3\{i,j\}} - Y_{\{i,j\}}M_{\{i,j\}})^2,\quad s.t.\ M_{\{i,j\}} \geq 0 \tag{20}$$

Then, by calculating the derivative with respect to M, the optimal solution of $M_{\{i,j\}}$ can be derived as:

$$M^*_{\{i,j\}} = \max(A_{3\{i,j\}}/Y_{\{i,j\}}, 0) \tag{21}$$

Algorithm 2 The complete GSL method

Input: Source data and labels: $X_s \in \mathbb{R}^{D \times n_s}$, y_s;
Target data: $X_t \in \mathbb{R}^{D \times n_t}$; The maximum iteration T.
Output: P_t^* and $\hat{y}_t$
Initialization : $P_t = \mathbf{I}$;
While not converged **or** iteration $t < T$ **do**
1: Update $\hat{y}_t$ using existing classifier, there is $\hat{y}_t = classifier(P_t^T X_s, y_s, P_t^T X_t)$;
2: Construct Y using Eq.(4);
3: Fix Y, and solve P_t in problem (8) using Algorithm 1;
4: Check convergence by (23);
5: $t = t + 1$;
End while
6: **return** P_t^*, $\hat{y}_t$

- **Update** Y_1, μ:

The multiplier Y_1 and step-size μ are updated by:

$$\begin{cases} Y_1 = Y_1 + \mu(Z-L) \\ \mu = \min(\rho\mu, \mu_{max}) \end{cases} \tag{22}$$

Specifically, the optimization process of problem (8) by using IALM is described in Algorithm 1, which shows the first stage (teacher teaches) of the proposed GSL for seeking a unified subspace P_t. In the second stage (student feedback), an existing classifier (e.g. SVM) is trained based on the newly projected source data by using P_t from Algorithm 1 for computing the pseudo labels $\hat{Y}_t$ of target data. Then an alternative algorithm between the first stage and the second stage is constructed in Algorithm 2, which is the complete GSL. As can be seen from Algorithm 2, two loops including inner loop (i.e. Algorithm 1) and outer loop are involved. To check the convergence of GSL, we define

$$\triangle P_t = \left\|P_t^{(t)} - P_t^{(t-1)}\right\|_F \Big/ \left\|P_t^{(t-1)}\right\|_F \tag{23}$$

where t indicates the iteration index of the outer-loop. The Algorithm 2 converges if the condition $\triangle P_t < \varepsilon$ is satisfied. Note that $\varepsilon > 0$ indicates an extremely small value. The experiment proves that this condition can effectively measure the convergence of the outer loop.

Finally, an invariant, discriminative and domain agnostic target subspace P_t can be achieved to extract features across domains in a progressive manner.

D. Computational Complexity and Convergence Analysis

1) Computational Complexity Analysis: For convenience, X_s and X_t are supposed to be $D \times n$ matrices. The main computations of Algorithm 1 include:

- Matrix inversion and multiplication in steps 2, 3 and 4, which involve a computational cost of $\mathcal{O}(kn^3)$;
- Singular value decomposition (SVD) in step 5, which involves a computational cost of $\mathcal{O}(n^3)$.

Suppose that the number of iterations for Algorithm 1 and Algorithm 2 is T_1 and T, respectively, the total computational complexity of GSL can be expressed as $\mathcal{O}(TT_1(k+1)n^3)$. Note that our algorithm is not suitable for large-scale data,

but it is fast enough on small-scale data sets. We suggest that for large-scale tasks, off-the-shelf CNNs trained on big data (e.g. ImageNet) can be used as feature extractor for small-scale data. The performance has been verified as shown in our experiments (e.g. Table III).

2) Convergence Analysis of Algorithm 1 and 2:

First, the convergence of Algorithm 1 is discussed as follows. The convergence of the exact ALM algorithm has been proved in [43]. In Algorithm 1, the inexact ALM proposed for solving the robust PCA [40] has been used in step 5 for solving (18). However, so far, the convergence of IALM with three or more variables (e.g. five variables are involved in Algorithm 1) is generally difficult to be theoretically guaranteed [44]. Fortunately, as claimed in [45], three sufficient conditions can ensure the convergence.

- The dictionary matrix D (i.e. X_s in our GSL model) should be of full column rank;
- The optimality gap ϵ_k shown in Eq.(24) generated in each iteration step is monotonically decreasing.

$$\epsilon_k = \|(Z_k, L_k) - (Z^*, L^*)\|_F^2 \tag{24}$$

  where Z_k and L_k denote the solutions obtained at the k-th iteration, respectively. Z^* and L^* represent the optimal solutions of the model $\arg\min_{Z,L} \mathcal{L}$.
- The penalty parameter μ in step 7 of Algorithm 1 should be upper bounded.

Experimentally, the convergence curves of Algorithm 1 are shown in Section V, which demonstrates a good convergence, even if the above conditions are difficult to hold constantly.

Second, for Algorithm 2, although the student feedback process in step 1 and the teacher teaching process in step 3 seem to be independent, the convergence and performance curves of the outer-loop on several datasets as shown in Section V provide evidences that the interaction between teacher and student can progressively promote each other. Intuitively, the label guidance contributes to raising the model to a higher "stair", while the convergence condition $\triangle P_t$ is amount to the height of each "stair" in each iteration. The upper bound of the algorithm is achieved when $\triangle P_t$ approaches ε. Ultimately, a domain agnostic subspace P_t with domain gap between "*teacher*" and "*student*" relieved can be achieved.

IV. Nonlinear Guide Subspace Learning

In many computer vision tasks, nonlinear domain transfer is often encountered. Therefore, nonlinear model under the linear GSL framework can be deduced. Recently, there are several approaches handling nonlinear distribution alignment [46], [47] by mapping the raw data into a reproducing kernel Hilbert space (RKHS). Therefore a nonlinear version of GSL, i.e. NGSL model, is also derived through kernel embedding.

A. Formulation

Let ϕ: $x \to \phi(x)$ be a nonlinear mapping from the raw feature space $\mathbb{R}^D$ into a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$. Then we have $\phi(X) = [\phi(x_1), \phi(x_2), ..., \phi(x_n)]$. In RKHS, the kernel Gram matrix $\mathcal{K}$ is defined as $[\mathcal{K}]_{i,j} = < \phi(x_i), \phi(x_j) >_{\mathcal{H}} = \phi(x_i)^T\phi(x_j) = k(x_i, x_j)$, where $k(\cdot)$ is a kernel function, such as sigmoid and RBF functions.

Algorithm 3 The first stage of NGSL for solving (26)

Input: $X_s \in \mathbb{R}^{D\times n_s}$, $X_t \in \mathbb{R}^{D\times n_t}$, $X = [X_s, X_t] \in \mathbb{R}^{D\times n}$, $Y \in \mathbb{R}^{d\times n}$, α, β, λ;

Output: Φ_t;

Initialisation: $M = 1$, $Z = L = 0$, $Y_1 = 0$, $\mu_{max} = 10^6$, $\rho = 1.01$, $\epsilon = 10^{-7}$;

1: Initialize P_s using existing method, e.g. PCA.
2: Initialize $\Phi_s = \phi(X)^{-1}P_s$.
3: Compute $K := \phi(X)^T\phi(X)$, $K_s := \phi(X)^T\phi(X_s)$ and $K_t := \phi(X)^T\phi(X_t)$.
While not converged **do**
4: Fix other variables and update Φ_t by solving (27);
5: Fix other variables and update Φ_s by solving (28);
6: Fix other variables and update Z by: $Z^* = (2K_s^T\Phi_s\Phi_s^TK_s + \mu I)^{-1}(2K_s^T\Phi_s\Phi_t^TK_t + \mu A_2)$
7: Fix other variables and update L by solving (18);
8: Fix other variables and update M by solving: $M = \arg\min_M \frac{1}{2}\left\|\Phi_t^TK - Y \circ M\right\|_F^2$
9: Update multiplier Y_1 and penalty parameter μ by (22);
10: Check convergence: $\|Z - L\|_\infty < \epsilon$.
End while
11: **return** Φ_t

Let $K = \phi(X)^T\phi(X)$, $K_s = \phi(X)^T\phi(X_s)$ and $K_t = \phi(X)^T\phi(X_t)$ denote the kernel Gram matrix. Then for representing the source and target subspaces P_s and P_t, one proposition is provided as follows.

Proposition 1: There exist optimal P_s and P_t that can be intuitively represented as a linear combination of $\phi(X)$ as

$$P_s = \phi(X)\Phi_s; P_t = \phi(X)\Phi_t \tag{25}$$

where Φ_s and Φ_t are the representation coefficient matrices.

Therefore, with kernel mapping, by substituting the Eq.(25) into Eq.(6), the proposed NGSL model can be formulated as:

$$\begin{aligned}\min_{\Phi_s,\Phi_t,M,Z} &\beta\left\|\phi(X)\Phi_s - \phi(X)\Phi_t\right\|_F^2 + \left\|\Phi_t^TK_t - \Phi_s^TK_sZ\right\|_F^2 \\ &+ \alpha\left\|Z\right\|_* + \frac{1}{2}\left\|\Phi_t^TK - Y\circ M\right\|_F^2 \\ &s.t.\ M \succcurlyeq 0\end{aligned} \tag{26}$$

where K, K_s, and K_t are kernel Gram matrices.

B. Optimization

The optimization algorithm of NGSL is similar with GSL as shown in Algorithm 1 and 2. However, in updating Φ_s and Φ_t, the matrix may not be full-rank and becomes irreversible. To this end, we introduce a small constant $\lambda > 0$ in order to obtain a numerically stable solution. Therefore, there are

$$\begin{aligned}\Phi_t^* =&(2\beta K + 2K_tK_t^T + KK^T + \lambda I)^{-1} \\ &(2\beta K\Phi_s + 2K_tZ^TK_s^T\Phi_s + K(Y\circ M)^T)\end{aligned} \tag{27}$$

$$\Phi_s^* = (2\beta K + 2K_sZZ^TK_s^T + \lambda I)^{-1}(2\beta K\Phi_t + 2K_sZK_t^T\Phi_t) \tag{28}$$

Algorithm 4 The complete NGSL method

Input: Source data and labels: $X_s \in \mathbb{R}^{D\times n_s}$, y_s;
Target data: $X_t \in \mathbb{R}^{D\times n_t}$; The maximum iteration T.
Output: $P_t = \phi(X)\Phi_t^*$ and $\hat{y}_t$
Initialization : $\Phi_t = \mathbf{I}$;
While not converged **or** iteration $t < T$ **do**
1: Update $\hat{y}_t$ using existing classifier, there is $\hat{y}_t = classifier(\Phi_t^T K_s, y_s, \Phi_t^T K_t)$;
2: Construct Y using Eq.(4);
3: Fix Y, and solve Φ_t using Algorithm 3;
4: Check convergence using Eq.(23);
5: $t = t + 1$;
End while
6: **return** P_t^*, $\hat{y}_t$

Due to the space limitation, the deduction and optimization details of NGSL for Z and M are provided in **Appendix A** and **B**. In NGSL, the continuously differentiable functions such as Gaussian RBF kernel, linear kernel, etc. can be used as kernel function. The optimization of the first stage in NGSL (*Teacher teaches*) is shown as Algorithm 3. The complete NGSL with pseudo target label learning is presented in Algorithm 4.

V. Experiment

In this section, extensive experiments are conducted to evaluate the effectiveness of the proposed methods for unsupervised domain adaptation scenarios, which is much closer to real-world applications. Note that, we focus on close set problem instead of open set problem [48], therefore both domains have the same class labels. We compare our methods with state-of-the-arts, including: 1) data level methods: TSL [13], RDALR [12], LTSL [8] and DTSL [18]; 2) subspace level methods: SA [22], GFK [34] and CORAL [23]; 3) pseudo-label strategy based methods: JDA [36], JGSA [38] and LDADA [49]. In addition, we compare our methods with several recent deep transfer learning methods, such as Deep Domain Confusion (DDC) [50], Domain Adaptation Networks (DAN) [51] and Residual Transfer Network (RTN) [52]. Further discussions on parameter sensitivity, ablation analysis and convergence are also presented.

A. Databases

In experiments, four visual benchmark datasets, including 4DA object dataset [34], MSRC-VOC2007 image dataset [53], CMU PIE face dataset [36] and COIL20 3D object dataset [54] are exploited and tested.

1) **4DA Dataset:** 4DA consists of Office data and Caltech-256 data [55]. The Office data contains three real-world object domains, including Amazon, Webcam and DSLR. 4DA is formulated with 10 shared categories between Office and Caltech datasets. Therefore, 4 domains including **A** (Amazon), **C** (Caltech-256), **D** (DSLR) and **W** (Webcam) are constructed. In feature representation, two kinds of features i.e. shallow and deep features are used separately. *First*, the SURF feature [34] encoded with 800-dimension BoW features is used as shallow feature. *Second*, the feature extracted from a deep model (i.e. the FC7 activations of VGG-VD-16 model) [56] is exploited as deep feature. By deploying pairwise domains such as source domain and target domain alternatively, totally 12 cross-domain tasks are constructed. Some example images in 4DA dataset are illustrated in Fig. 1 a).

TABLE I
Experimental Data Description

Dataset	Subsets	Abbr.	images	Feature (dim)	Classes
4DA	**A**mazon	A	958	SURF(800) VGG-FC7(4,096)	10
	Caltech	C	1,123		
	DSLR	D	157		
	Webcam	W	295		
PIE	C05(←)	P1	3,332	Pixel(1,024)	68
	C27(⊙)	P4	3,329		
	C29(→)	P5	1,632		
MV	**M**SRC	M	1,269	Codebook(240)	6
	VOC2007	V	1,530		
COIL20	COIL1	C1	720	Pixel(1,024)	20
	COIL2	C2	720		

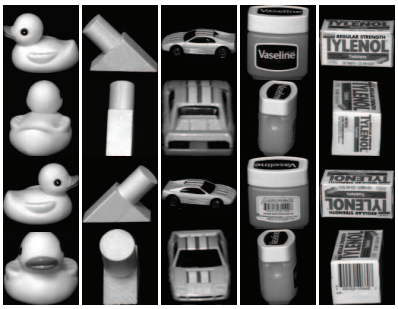

Fig. 4. Some examples from COIL20 3D object dataset. Each column denotes one object across different poses, which shows significant domain shift.

2) **MSRC and VOC2007 (MV) Dataset:** MSRC data contains 4,323 images of 18 classes, which was released by Microsoft Research Cambridge. VOC2007 contains 5011 images of 20 classes. In our experiment, 6 shared semantic classes including aeroplane, bicycle, bird, car, cow, and sheep from both datasets are formulated, with each image cropped with 256 pixels. The 128-dimensional dense SIFT (DSIFT) feature was 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 [18], two cross-domain tasks are constructed: MSRC vs. VOC2007 and VOC2007 vs. MSRC. Some images of MV data are illustrated in Fig. 1 b).

3) **CMU PIE Face Dataset:** PIE contains 68 individuals with 41,368 face images of size 32×32. Five sessions including PIE1 (C05, left pose), PIE2 (C07, upward pose), PIE3 (C09, downward pose), PIE4 (C27, frontal pose), and PIE5 (C29, right pose) are involved. The face images were captured by 13 different poses and 21 different illuminations and/or expressions. Alternatively, we construct 4 cross-domain face recognition tasks: PIE1 vs. PIE4, PIE4 vs. PIE1, PIE4 vs. PIE5, and PIE5 vs. PIE4. Some example images of the face dataset are illustrated in Fig. 1 c).

4) **COIL20 Dataset:** The COIL20 dataset contains 20 objects with 1440 gray scale images (i.e. 72 multi-pose images per object). Each image has 32×32 pixels and 256 gray

TABLE II
RECOGNITION ACCURACIES (%) ON 4DA DATASET WITH SURF FEATURE. NA DENOTES NO ADAPTATION, THE BEST IS TYPED IN BOLDFACE, THE SECOND BEST IS UNDERLINED, AND * DENOTES THE METHODS WITH PSEUDO-LABEL STRATEGY.

Data Set	Compared TL/DA Methods													
	NA	SA	JDA*	TSL	RDALR	LTSL	DTSL	GFK	JGSA*	CORAL	LDADA*	GSL	NGSL$_{linear}$	NGSL$_{rbf}$
C→A(1)	50.1	54.4	**59.8**	52.3	52.5	24.1	53.3	56.6	55.1	45.9	54.8	56.6	58.7	59.3
C→W(2)	43.1	45.8	50.1	40.3	40.7	22.9	45.8	48.1	49.7	37.8	60.2	55.9	59.7	**63.4**
C→D(3)	47.8	40.9	44.1	49.0	45.2	14.6	**51.0**	42.9	46.0	31.8	41.5	49.7	49.7	49.0
A→C(4)	42.8	44.8	44.9	43.3	43.6	21.4	44.7	44.3	40.8	37.1	38.4	45.4	**46.0**	45.6
A→W(5)	37.0	44.1	47.0	34.6	35.9	18.2	38.3	42.7	**59.0**	37.9	49.3	41.7	44.1	45.1
A→D(6)	37.2	37.7	44.2	38.9	36.9	22.3	39.5	39.9	**49.4**	38.5	39.1	44.0	47.1	47.1
W→C(7)	29.5	32.3	29.8	31.4	28.1	34.6	30.3	32.0	29.7	32.5	31.7	35.3	**37.9**	37.8
W→A(8)	34.2	**43.3**	42.0	34.7	31.2	39.5	34.7	38.3	34.6	39.4	35.1	40.7	41.8	42.1
W→D(9)	80.6	70.3	86.3	79.6	83.4	72.6	82.8	78.7	78.5	80.9	74.6	88.5	88.5	**89.8**
D→C(10)	30.1	31.1	34.4	33.1	32.3	35.4	30.7	30.8	30.2	27.8	29.9	31.8	35.3	**37.6**
D→A(11)	32.1	40.8	**44.6**	32.6	33.7	39.4	33.2	40.4	39.0	31.9	40.6	34.8	40.6	43.7
D→W(12)	72.2	74.4	83.3	72.5	72.5	74.9	76.6	80.3	75.1	69.4	74.7	84.1	85.8	**86.1**
Average	44.7	46.7	50.9	45.2	44.7	35.0	46.7	47.9	48.9	42.6	47.5	50.7	52.9	**53.9**

TABLE III
RECOGNITION ACCURACIES (%) ON 4DA DATASET WITH THE DEEP FEATURE FROM VGG-VD-16 MODEL. NA DENOTES NO ADAPTATION, THE BEST IS TYPED IN BOLDFACE, THE SECOND BEST IS UNDERLINED, AND * DENOTES DEEP TRANSFER LEARNING METHODS.

Data Set	Compared TL/DA Methods												
	NA	SA	JDA	GFK	JGSA	CORAL	LDADA	DDC*	DAN*	RTN*	GSL	NGSL$_{linear}$	NGSL$_{rbf}$
C→A(1)	91.5	93.2	93.7	93.6	94.2	91.6	95.1	91.9	92.0	94.4	95.2	**95.9**	95.8
C→W(2)	83.7	86.4	94.6	86.8	93.3	78.9	94.4	85.4	90.3	96.6	96.6	**99.0**	98.6
C→D(3)	89.9	95.0	93.2	91.0	94.4	87.6	93.2	88.1	90.5	92.9	94.9	**98.7**	**98.7**
A→C(4)	81.7	77.1	90.1	85.3	87.2	80.1	88.7	85.0	85.1	88.5	91.2	**93.6**	93.1
A→W(5)	74.8	80.4	91.5	85.8	95.7	75.7	92.5	86.1	93.8	97.0	94.2	**98.6**	**98.6**
A→D(6)	77.2	89.6	91.3	85.5	94.1	76.2	90.0	89.0	92.4	94.6	95.5	95.5	**96.2**
W→C(7)	77.3	77.9	86.7	81.3	82.3	77.6	88.3	78.0	84.3	88.4	90.5	**92.9**	92.2
W→A(8)	85.5	87.3	93.8	90.2	94.9	90.7	94.3	84.9	92.1	93.1	93.1	95.9	**96.0**
W→D(9)	99.0	98.0	96.1	98.0	96.1	98.0	99.6	**100**	**100**	**100**	**100**	**100**	**100**
D→C(10)	75.0	78.6	84.8	82.3	85.2	73.1	84.8	81.1	82.4	84.3	86.2	**91.5**	91.4
D→A(11)	83.6	83.8	91.7	90.8	93.8	84.5	94.2	89.5	92.0	95.5	88.1	**96.2**	96.0
D→W(12)	95.8	97.0	89.2	97.3	96.4	94.9	95.0	98.2	99.0	98.8	**99.7**	**99.7**	**99.7**
Average	84.6	87.0	91.4	89.0	92.3	84	92.6	88.2	91.2	93.7	93.8	**96.5**	96.4

levels per pixel. In experiments, by following the experimental protocol in [18], the dataset is divided into two subsets COIL1 and COIL2. Specifically, the COIL1 set contains all images taken in the directions of [0°, 85°] $\bigcup$ [180°, 265°], so the total number of images is 720. Similarly, the COIL2 set contains all images taken in the directions of [90°, 175°] $\bigcup$ [270°, 355°]. Several example images are illustrated in Fig. 4.

The detailed information of the 4 benchmark datasets in the experiments is summarized in Table I.

B. Experimental Setting

We strictly follow the same experimental configuration as [36], [57], [58], which exploited all the source instances during training process. Two trade-off parameters α and β are involved in the proposed GSL model, and an extra parameter λ is introduced in NGSL model. For fairness, the three parameters are tuned from the parameter set [0.1, 1, 10]. Additionally, for NGSL, two kernel functions such as linear kernel function $k(x, y) = x^T y$ and Gaussian kernel function (i.e. Radial Basis Function, RBF) $k(x, y) = exp(-\gamma \|x - y\|^2)$ are considered. The Gaussian parameter γ is set as 1 for all experiments of NGSL. In experiments, SVM is used to progressively generate pseudo target labels $\hat{y_t}$. In order to simplify the parameter tuning, the dimensionality d of the invariant subspace is constantly set as the number c of classes for each dataset.

C. Experimental Results

Results on the 4DA Dataset with Shallow SURF Feature. The 4DA dataset is still a challenging benchmark, which therefore attracts many competitive approaches for evaluation and comparison. The recognition accuracies are reported in Table II, from which we observe that the proposed GSL ranks the second (50.7%) in average and slightly inferior to JDA (50.9%), but the proposed NGSL is 3% higher than JDA and shows state-of-the-art performance. The reason is that the combination of the global information alignment through the proposed subspace guiding learning mechanism in this paper can effectively narrow the domain gap between source and target domains. Also, we observe that the methods with pseudo-label strategy such as JDA, JGSA, LDADA and our methods outperform others, and the proposed *student feedback* stage with pseudo target labels is confirmed to be effective. In addition, the proposed NGSL shows superior performance and proves that the domain gap is generally caused by nonlinear shifts in real applications. Notably, we find that a proper feature pre-processing can improve the performance of the proposed models. Two kinds of pre-processing methods including l_2-normalization and z-score plus l_2-norm normalization are considered. The comparison results on 4DA tasks with SURF features are shown in Fig. 5, from which we observe that z-

TABLE IV
RECOGNITION ACCURACIES (%) ON MSVC-VOC2007 DATASET. NA DENOTES NO ADAPTATION, THE BEST IS TYPED IN BOLDFACE, THE SECOND BEST IS UNDERLINED AND * DENOTES THE RESULTS OF GFK BASED ON 1-NN CLASSIFIER.

Data Set	Compared TL/DA Methods													
	NA	SA	JDA	TSL	RDALR	LTSL	DTSL	GFK*	JGSA	LDADA	CORAL	GSL	NGSL$_{linear}$	NGSL$_{rbf}$
M→V(1)	37.1	31.8	38.2	32.4	37.5	38.0	38.0	28.8	38.7	25.1	33.9	41.8	<u>40.7</u>	**42.0**
V→M(2)	55.5	46.0	59.3	43.2	62.3	<u>67.1</u>	56.4	48.9	49.3	43.2	54.1	66.4	64.7	**68.2**
Average	46.3	38.9	48.8	37.8	49.9	52.6	47.2	38.9	44.0	34.2	44.0	<u>54.1</u>	52.7	**55.1**

TABLE V
RECOGNITION ACCURACIES (%) ON COIL20 DATASET. NA DENOTES NO ADAPTATION, THE BEST IS TYPED IN BOLDFACE, THE SECOND BEST IS UNDERLINED, AND * DENOTES THE RESULTS OF GFK BASED ON 1-NN CLASSIFIER.

Data Set	Compared TL/DA Methods													
	NA	SA	JDA	TSL	RDALR	LTSL	DTSL	GFK*	JGSA	LDADA	CORAL	GSL	NGSL$_{linear}$	NGSL$_{rbf}$
C1→C2(1)	82.7	86.7	88.7	80.0	80.7	75.4	84.6	72.5	85.1	77.9	84.9	88.8	**92.9**	<u>92.1</u>
C2→C1(2)	84.0	<u>90.6</u>	**93.1**	75.6	78.8	72.2	84.2	74.2	83.9	81.5	87.9	89.2	89.3	90.3
Average	83.3	88.7	90.9	77.8	79.7	73.8	84.4	73.3	84.5	79.7	86.4	89.0	<u>91.1</u>	**91.2**

TABLE VI
RECOGNITION ACCURACIES (%) ON PIE DATASET. NA DENOTES NO ADAPTATION, THE BEST IS TYPED IN BOLDFACE, THE SECOND BEST IS UNDERLINED, AND * DENOTES THE RESULTS OF GFK BASED ON 1-NN CLASSIFIER.

Data Set	Compared Transfer Learning Methods													
	NA	SA	JDA*	TSL	RDALR	LTSL	DTSL	GFK*	JGSA	LDADA	CORAL	GSL	NGSL$_{linear}$	NGSL$_{rbf}$
P1→P4(1)	51.8	42.8	<u>84.5</u>	46.7	41.7	20.0	81.3	31.2	76.1	35.6	26.0	**84.8**	83.7	75.1
P4→P1(2)	65.9	51.4	80.6	59.2	48.1	52.8	79.7	34.2	73.3	39.5	36.6	**83.9**	<u>83.1</u>	81.1
P4→P5(3)	52.0	47.9	54.6	45.2	48.8	47.0	<u>71.0</u>	37.4	55.3	26.9	40.8	**71.8**	65.2	67.0
P5→P4(4)	53.4	43.1	57.0	53.1	44.5	23.6	<u>66.1</u>	31.3	64.4	29.3	30.2	63.2	64.4	**70.0**
Average	55.8	53.8	69.2	51.1	45.8	35.9	<u>74.5</u>	33.5	67.3	32.8	33.4	**75.9**	74.1	73.3

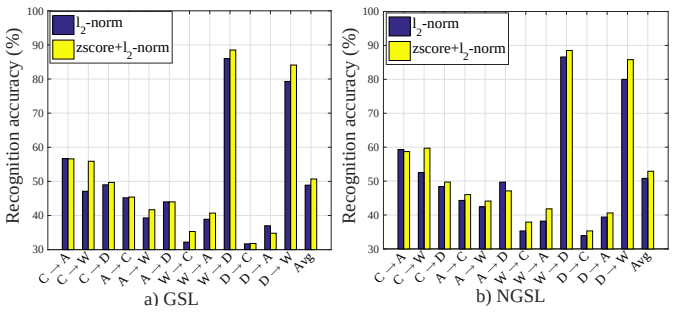

Fig. 5. Comparison between feature pre-processing methods on 4DA dataset with SURF features. a) denotes the results of GSL model. b) denotes the results of NGSL model (linear kernel). The figure is better viewed in color.

score+l_2 normalization shows better performance.

Results on the 4DA Dataset with Deep Feature. Following [49], the experimental comparisons on deep features are presented in Table III, from which we observe the significantly better results than SURF feature for all methods. From the results, we have the following observations:

- The proposed GSL outperforms other state-of-the-art non-deep transfer learning methods, such as JDA, JGSA.
- The proposed NGSL shows significant improvement (3.9%) in average over state-of-the-art models.
- By comparing with deep transfer learning methods denoted with *, the proposed GSL is merely slightly better than RTN [52], while the proposed NGSL outperforms state-of-the-art RTN with 2.8% in accuracy.
- The comparison shows that the proposed GSL, as a shallow learning method, has attractive competitiveness.

Results on the MSRC-VOC2007 (MV) Dataset. The results on MV dataset are shown in Table IV, from which we can observe that GSL outperforms state-of-the-art LTSL method with 1.5% in average accuracy. Moreover, the proposed NGSL with Gaussian kernel shows further improvement and achieves state-of-the-art performance (55.1%) than other competitive methods. The results demonstrate that the proposed GSL and NGSL models can effectively help reduce the distribution discrepancy across different visual domains.

Results on the COIL20 Dataset. The results on COIL20 dataset are shown in Table V, from which we can see that the proposed GSL shows slightly inferior results than the competitive JDA method. This may be due to that the COIL20 data consists of 3D objects with nonlinear transfer function across poses. Therefore, the proposed NGSL shows the best performance (91.2%), and the effectiveness of the proposed NGSL in nonlinear domain shift is demonstrated.

Results on the PIE Dataset. The experimental results are shown in Table VI. Compared with second-best method DTSL [18], GSL wins 3 out of 4 tasks and achieves state-of-the-art performance over others. Regarding the baseline without adaptation (i.e. NA), the average accuracy increases from 55.8% to 75.9%. The average accuracies of both NGSL$_{linear}$ and NGAL$_{rbf}$ perform slightly inferior to GSL. However, for the task of P5→P4, NGSL is 1.2% and 6.8% higher than GSL. The reason may be that PIE dataset is encoded with 1024-dimensional pixel feature, and both GSL and NGSL set the dimensionality of the invariant subspace as the number c ($c = 68$ on PIE dataset) of classes. GSL maps the dimensions of the original space D (1,024) to c (68), and the dimension loss is $(1024-68)/1024 \approx 93.4\%$. However, NGSL maps the

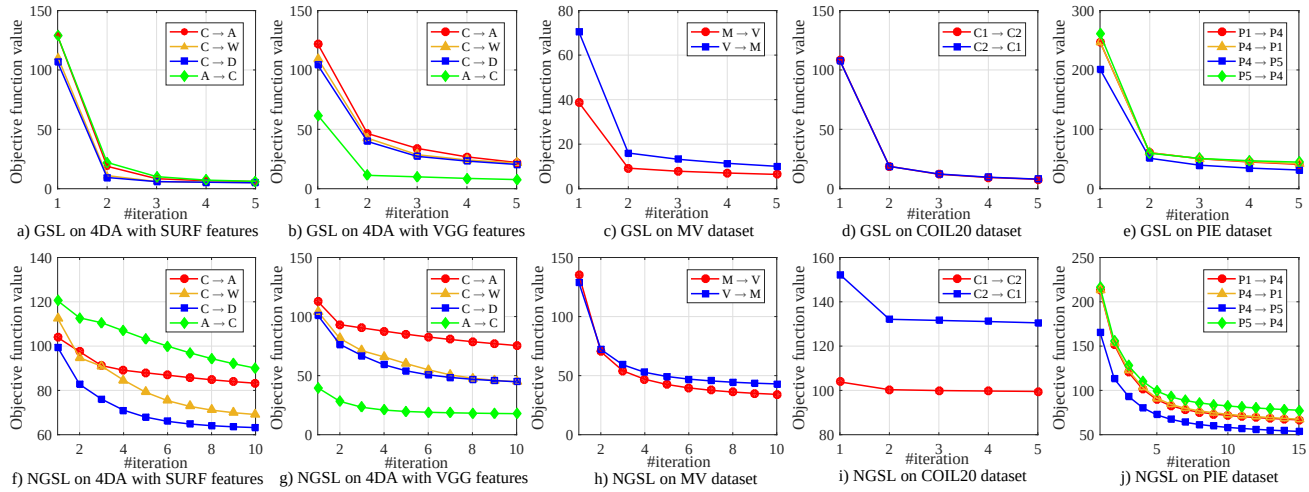

Fig. 6. Convergence of Algorithm 1 and Algorithm 3 on benchmark cross-domain datasets. The 1^{st} row represents the convergence of the GSL model and the 2^{nd} row represents the convergence of the NGSL model.

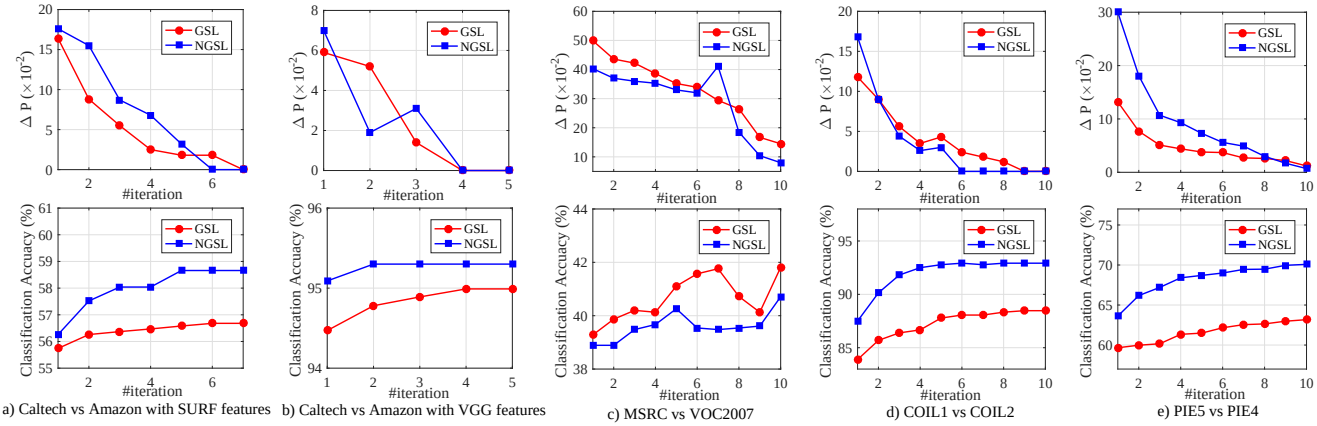

Fig. 7. Convergence and performance variation (%) of Algorithm 2 (GSL) and Algorithm 4 (NGSL) on benchmark cross-domain datasets. The 1^{st} row represents the algorithmic convergence and the 2^{nd} row shows the performance variation with iterations.

dimension n (the number of images) to c, which results in a higher dimension loss of 98.6%∼99.0%. Note that, n=1632, 3329, and 3332 for P5, P4, and P1, respectively. Obviously, NGSL loses more information, such that NGSL does not perform well. Due to the number $n = 1632$ of P5 is close to D, therefore, a better cross-domain recognition performance is achieved by NGSL for the task P5→P4.

VI. Model Analysis and Discussion

In this section, the convergence, parameter sensitivity and ablation analysis of models are presented and discussed.

A. Convergence

The algorithms for solving the proposed GSL and NGSL models consist of two stages: inner-loop (Algorithm 1 and Algorithm 3) and outer-loop (Algorithm 2 and Algorithm 4).

1) Convergence of Algorithm 1 and Algorithm 3 (Inner-loop): We empirically show the convergence of inner-loop in Fig. 6 by running Algorithm 1 (GSL) and Algorithm 2 (NGSL) on several datasets. Fig. 6 a)-Fig. 6 e) show the convergence of the proposed GSL inner loop (Algorithm 1) on 4DA (C→A, C→W, C→D, and A→C), MV, COIL20 and PIE datasets, respectively. Similarly, Fig. 6 f)-Fig. 6 j) show the convergence of the proposed NGSL inner loop (Algorithm 3). We can observe that the proposed GSL model converges at the 5-th iteration and the proposed NGSL model converges after more iterations. In order to reduce the training time, we uniformly set the number of iterations of inner-loop in experiments as 5 (except the PIE data that requires 15 iterations). Note that the optimality gap in each iteration is proved to be monotonically decreasing in Section III.D.

2) Convergence of Algorithm 2 and Algorithm 4 (Outer-loop): As discussed in Fig. 6, the first stage (inner loop) of the proposed GSL and NGSL can quickly converge to an optimum after several iterations. Due to that the first stage (inner loop) and the second stage (outer loop) are relatively independent, therefore, it is also necessary to check the convergence and performance variation with iterations by running the complete GSL and NGSL algorithms on several datasets. Although the convergence cannot be theoretically guaranteed, the optimality gap $\triangle P_t$ of the proposed models on 4DA (C→A), MSRC-

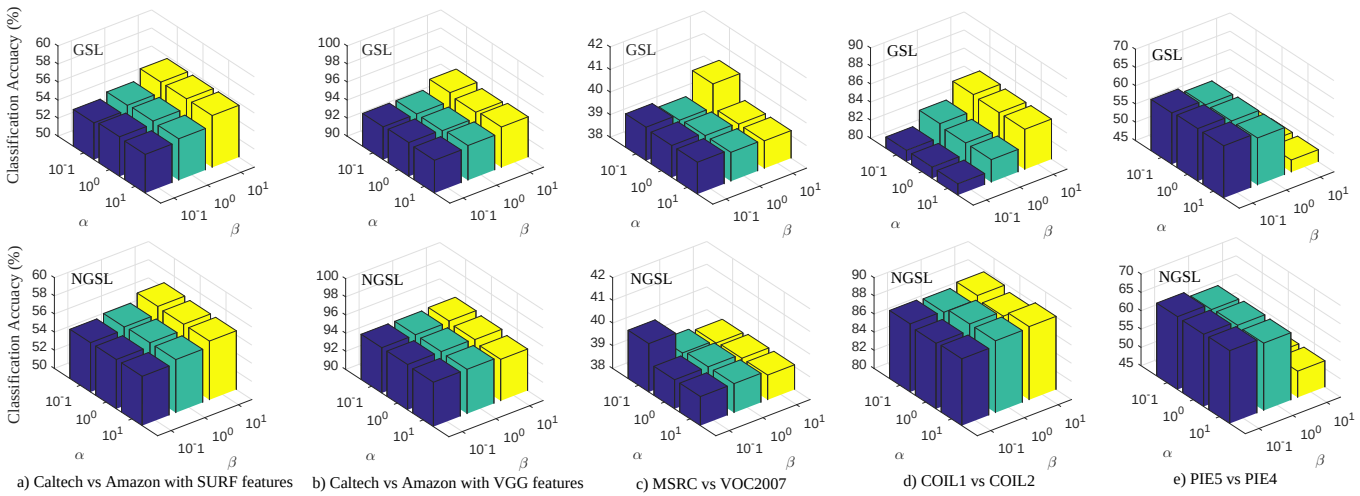

Fig. 8. Performance variations of the proposed GSL and NGSL with respect to different values of α and β on benchmark cross-domain datasets.

VOC (M→V), COIL20 (COIL1→COIL2), and PIE (P5→P4) is generally decreasing as shown in the first row of Fig. 7. Therefore, the proposed models can progressively converge to an optimal solution by alternatively optimizing the two stages. Further, we show the performance variation on several datasets with iterations in the second row of Fig. 7. From the variations, we can observe that the classification accuracy is increasing with the alternative and progressive learning between the first stage and the second stage for both GSL and NGSL, which can be analogized to the process of "climb the stairs". Note that when $\triangle P_t$ becomes extremely small, the classification accuracy becomes stable and unchanged, therefore $\triangle P_t$ can intuitively interpret the height of the "stairs". The key idea of guide learning, i.e. "*the students surpasses the master*" is validated. In experiments, we set the maximum iteration number T in Algorithm 2 and Algorithm 4 as 10.

B. Parameter Sensitivity

In order to explore the model's sensitivity to trade-off parameters, we conduct experiments to study the parameter sensitivity of our models. In GSL, two trade-off parameters are involved, i.e. α and β. Specifically, the two parameters are tuned from the given set $[10^{-1}, 1, 10^{1}]$. For NGSL, an extra parameter λ is introduced for avoiding the irreversibility of matrix. For easier analysis, we focus on the two trade-off parameters α and β. Note that in NGSL analysis, we set $\lambda = 10^{1}$ for 4DA and COIL20 datasets, $\lambda = 1$ for MV dataset, and $\lambda = 10^{-1}$ for PIE dataset. The classification accuracies with respect to different trade-off parameters by using GSL and NGSL are shown in Fig. 8, from which we can observe that the parameter β has a relatively larger impact on the performance. The reason is that the parameter β reflects the importance of the data dependent subspace guidance term. We can see from Fig. 8 that a larger β could result in better performance except the MSRC-VOC data in Fig. 8 c) and the PIE data in Fig. 8 e). In fact, this commonly happens in computer vision and machine learning tasks, due to the data characteristic. As shown in Fig. 8 e), a smaller β is better, which denotes that the subspace guidance term contributes less. This may be due to the intrinsic similarity between faces in PIE dataset, such that the reconstruction based data guidance term contributes more to domain adaptation. Generally, a larger β contributes much on the domain discrepancy reduction and a smaller α makes more relaxation for reconstruction. The parameters can be easily tuned in experiments.

TABLE VII
RESULTS OF ABLATION ANALYSIS.

Tasks	GSL	w/o LRC	w/o SG	w/o FB
C→A	**56.6**	55.6	53.9	55.7
M→V	**41.8**	**41.8**	**41.8**	39.3
C1→C2	**88.8**	86.7	81.0	83.9
Average	**62.4**	61.4	58.9	59.6

C. Ablation Analysis

In the proposed GSL models, subspace guidance (SG), feedback stage (FB) and low-rank constraint (LRC) are presented. For better insight of the model, the ablation analysis is presented. *First*, by setting the two parameter α and β as 0, respectively, the ablation analysis of SG and LRC can be discussed. The experimental results on three datasets (4DA: C→A, MSRC-VOC: M→V, and COIL20: C1→C2;) by using the model without (w/o) SG term and without (w/o) LRC term are presented in Table VII. From the results, we can observe that the performance is significantly degraded from 62.4% to 58.9% without SG term, and the effect of the proposed subspace guidance is validated. Also, by dropping the LRC term, the accuracy is 1% decreased. *Second*, for verifying the effectiveness of the proposed two-stage feedback strategy (FB), as can be seen in Table VII, by removing the pseudo label update stage (w/o FB), the average accuracy is significantly degraded from 62.4% to 59.6%. From the ablation analysis of each part in GSL, we can conclude that the performance benefits from the proposed subspace and label guidance mechanism. The significance of the proposed progressive guide learning paradigms is verified.

VII. Conclusion

We propose a new transfer learning framework called Guide Subspace Learning (GSL) for unsupervised domain adaptation, which consists of couple projections based subspace guidance, low-rank reconstruction based data guidance and pseudo-label relaxation based label guidance. The proposed GSL is inspired by the "teacher teaches (source domain) and student feedback (target domain)" learning mode in human world, which aims to progressively seek an invariant, discriminative and domain agnostic target subspace. The teacher teaching stage proposes to learn the domain-specific projections. The student feedback stage proposes to provide the "feedback information" (i.e. the pseudo target labels) for the teacher teaching stage model. The proposed GSL holds the ultimate objective of "*the student surpasses the master*". Furthermore, the kernel embedding is introduced and a nonlinear GSL method called NGSL is derived, which aims to handle the nonlinear domain shift in high dimensional feature spaces (i.e. RKHS). In the training phase, a two-stage guide learning mechanism by intuitively following the "*climb the stairs*" process of growing step by step is proposed. Experimental results on challenging benchmark datasets demonstrate that our methods outperform many state-of-the-art TL/DA methods.

Appendix A
Deduction of The Proposed NGSL

Let ϕ: $x \to \phi(x)$ be a nonlinear mapping from the raw feature space $\mathbb{R}^D$ into a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$. The kernel Gram matrix $\mathcal{K}$ is defined as $[\mathcal{K}]_{i,j} =< \phi(x_i), \phi(x_j) >_{\mathcal{H}} = \phi(x_i)^T\phi(x_j) = k(x_i, x_j)$, where $k(\cdot)$ is a kernel function, such as sigmoid and RBF functions.

Let $K = \phi(X)^T\phi(X)$, $K_s = \phi(X)^T\phi(X_s)$ and $K_t = \phi(X)^T\phi(X_t)$ denote the kernel Gram matrix. Then for representing the source and target subspaces P_s and P_t, one proposition is provided as follows.

Proposition 1: There exist optimal P_s and P_t that can be intuitively represented as a linear combination of $\phi(X)$ as

$$P_s = \phi(X)\Phi_s; P_t = \phi(X)\Phi_t \tag{29}$$

where Φ_s and Φ_t are the representation coefficient matrices. Therefore, by substituting P_s and P_t into Eq.(6), there is

$$\begin{aligned}\min_{\Phi_s,\Phi_t,M,Z} &\beta \left\|\phi(X)\Phi_s - \phi(X)\Phi_t\right\|_F^2 \\ &+ \left\|\Phi_t^T\phi(X)^T\phi(X_t) - \Phi_s^T\phi(X)^T\phi(X_s)Z\right\|_F^2 \\ &+ \alpha\left\|Z\right\|_* + \frac{1}{2}\left\|\Phi_t^T\phi(X)^T\phi(X) - Y\circ M\right\|_F^2 \\ &s.t.\ M \succcurlyeq 0\end{aligned} \tag{30}$$

We have the kernel Gram matrix $K = \phi(X)^T\phi(X)$, $K_s = \phi(X)^T\phi(X_s)$ and $K_t = \phi(X)^T\phi(X_t)$, then the problem (30) can be further transformed as follows:

$$\begin{aligned}\min_{\Phi_s,\Phi_t,M,Z} &\beta \left\|\phi(X)\Phi_s - \phi(X)\Phi_t\right\|_F^2 + \left\|\Phi_t^TK_t - \Phi_s^TK_sZ\right\|_F^2 \\ &+ \alpha\left\|Z\right\|_* + \frac{1}{2}\left\|\Phi_t^TK - Y\circ M\right\|_F^2 \\ &s.t.\ M \succcurlyeq 0\end{aligned} \tag{31}$$

Appendix B
Optimization Process of NGSL Model

The optimization algorithm is similar with the GSL presented in the main text. Specifically, the optimization details of NGSL are clearly derived as follows.

By introducing an auxiliary variable L in problem (31), which is then reformulated as:

$$\begin{aligned}\min_{\Phi_s,\Phi_t,M,Z,L} &\beta \left\|\phi(X)\Phi_s - \phi(X)\Phi_t\right\|_F^2 + \left\|\Phi_t^TK_t - \Phi_s^TK_sZ\right\|_F^2 \\ &+ \alpha\left\|L\right\|_* + \frac{1}{2}\left\|\Phi_t^TK - Y\circ M\right\|_F^2 \\ &s.t.\ M \succcurlyeq 0,\ Z = L\end{aligned} \tag{32}$$

Therefore, the Augmented Lagrange multipliers function can be written as

$$\begin{aligned}\mathcal{L}_{\Phi_s,\Phi_t,Z,L,M} =&\beta \left\|\phi(X)\Phi_s - \phi(X)\Phi_t\right\|_F^2 \\ &+ \left\|\Phi_t^TK_t - \Phi_s^TK_sZ\right\|_F^2 + \alpha\left\|L\right\|_* \\ &+ \frac{1}{2}\left\|\Phi_t^TK - Y\circ M\right\|_F^2 \\ &+ tr(Y_1^T(Z-L)) + \frac{\mu}{2}\left\|Z-L\right\|_F^2\end{aligned} \tag{33}$$

where Y_1 denotes the Lagrange multiplier and $\mu > 0$ is a penalty parameter. Then, by using variables alternating strategy, we can derive the solution of each variable as follows.

Updating Φ_t: For Φ_t, by ignoring the irrelevant terms with respect to Φ_t, we obtain:

$$\begin{aligned}\Phi_t = arg\min_{\Phi_t} &\beta \left\|\phi(X)\Phi_s - \phi(X)\Phi_t\right\|_F^2 \\ &+ \left\|\Phi_t^TK_t - \Phi_s^TK_sZ\right\|_F^2 + \frac{1}{2}\left\|\Phi_t^TK - Y\circ M\right\|_F^2\end{aligned} \tag{34}$$

It is easy to get the closed-form solution of problem (34) as

$$\Phi_t = (2\beta K + 2K_tK_t^T + KK^T)^{-1}(2\beta K\Phi_s + 2K_tA_1^T + KA_2^T) \tag{35}$$

where $A_1 = \Phi_s^TK_sZ$ and $A_2 = Y\circ M$.

However, the matrix that needs inverse operation in Eq.(35) may not be full rank and irreversible. So, we introduce a small positive constant λ in order to obtain numerically stable solution of Φ_t, there is:

$$\Phi_t^* = (2\beta K + 2K_tK_t^T + KK^T + \lambda I)^{-1}(2\beta K\Phi_s + 2K_tA_1^T + KA_2^T) \tag{36}$$

Updating Φ_s: Similar to the update procedure of Φ_t, by setting the derivative with respect to Φ_s as zero. With a small constant $\lambda > 0$, the closed-form solution can be solved as:

$$\Phi_s^* = (2\beta K + 2K_sZZ^TK_s^T + \lambda I)^{-1}(2\beta K\Phi_t + 2K_sZK_t^T\Phi_t) \tag{37}$$

Updating Z: By dropping those terms without containing variable Z in (33), we get:

$$\begin{aligned}Z = arg\min_{Z} &\left\|\Phi_t^TK_t - \Phi_s^TK_sZ\right\|_F^2 + tr(Y_1^T(Z-L)) \\ &+ \frac{\mu}{2}\left\|Z-L\right\|_F^2\end{aligned} \tag{38}$$

The closed-form solution of (38) can be easily obtained as:

$$Z^* = (2Ks^T\Phi_s\Phi_s^TK_s + \mu I)^{-1}(2Ks^T\Phi_s\Phi_t^TK_t + \mu A_3) \tag{39}$$

where $A_3 = L - Y_1/\mu$.

Updating L: By dropping those terms without containing variable L in (33), we can deduce the following form:

$$L = arg\min_{L} \alpha \|L\|_* + \frac{\mu}{2}\|L - (Z + Y_1/\mu)\|_F^2 \quad (40)$$

The solution of problem (40) is the same as Eq.(18) by using Theorem 1 and singular value thresholding (SVT) algorithm.

Updating M: By dropping those terms irrelevant to M in (33), we obtain:

$$M = arg\min_{M} \frac{1}{2}\left\|\Phi_t^T K - Y \circ M\right\|_F^2, \quad s.t.\ M \succcurlyeq 0 \quad (41)$$

By defining $A_4 = \Phi_t^T K$ and considering the $\{i,j\}$-th element $M_{\{i,j\}}$ of the matrix M, the above problem can be further written as:

$$M_{\{i,j\}} = \min_{M_{\{i,j\}}} \frac{1}{2}(A_{4\{i,j\}} - Y_{\{i,j\}}M_{\{i,j\}})^2, \quad s.t.\ M_{\{i,j\}} \geq 0 \quad (42)$$

By computing the derivative of the problem, the optimal solution of $M_{\{i,j\}}$ can be solved as:

$$M^*_{\{i,j\}} = max(A_{4\{i,j\}}/Y_{\{i,j\}}, 0) \quad (43)$$

Update Y_1, μ:

$$\begin{cases} Y_1 = Y_1 + \mu(Z - L) \\ \mu = min(\rho\mu, \mu_{max}) \end{cases} \quad (44)$$

The above optimization process of the NGSL can be referred to as Algorithm 3 in the main body of the paper.

REFERENCES

[1] M. Kan, J. Wu, S. Shan, and X. Chen, "Domain adaptation for face recognition: Targetize source domain bridged by common subspace," *International Journal of Computer Vision*, vol. 109, no. 1-2, pp. 94–109, 2014.

[2] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *CVPR*, 2011, pp. 1521–1528.

[3] X. Shen, F. Shen, Q. Sun, Y. Yang, Y. Yuan, and H. Shen, "Semi-paired discrete hashing: Learning latent hash codes for semi-paired cross-view retrieval," *IEEE Trans. Cybernetics*, vol. 47, no. 12, pp. 4275–4288, 2017.

[4] J. Li, K. Lu, Z. Huang, L. Zhu, and H. Shen, "Heterogeneous domain adaptation through progressive alignment," *IEEE Trans. Neural Networks and Learning Systems*, vol. 30, no. 5, pp. 1381–1391, 2019.

[5] L. Niu, W. Li, D. Xu, and J. Cai, "Visual recognition by learning from web data via weakly supervised domain generalization," *IEEE Trans. Neural Networks and Learning Systems*, vol. 28, no. 9, pp. 1985–1999, 2017.

[6] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowledge & Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[7] L. Yang, L. Jing, J. Yu, and M. Ng, "Learning transferred weights from co-occurrence data for heterogeneous transfer learning," *IEEE Trans. Neural Networks and Learning Systems*, vol. 27, no. 11, pp. 2187–2200, 2016.

[8] M. Shao, D. Kit, and Y. Fu, "Generalized transfer subspace learning through low-rank constraint," *International Journal of Computer Vision*, vol. 109, no. 1-2, pp. 74–93, 2014.

[9] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive svms," in *ACM MM*, 2007, pp. 188–197.

[10] L. Duan, D. Xu, and S. F. Chang, "Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach," in *CVPR*, 2012, pp. 1338–1345.

[11] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 465–479, 2012.

[12] I. H. Jhuo, D. Liu, D. T. Lee, and S. F. Chang, "Robust visual domain adaptation with low-rank reconstruction," in *CVPR*, 2012.

[13] S. Si, D. Tao, and B. Geng, "Bregman divergence-based regularization for transfer subspace learning," *IEEE Trans. Knowledge & Data Engineering*, vol. 22, no. 7, pp. 929–942, 2010.

[14] L. Zhang, W. Zuo, and D. Zhang, "Lsdt: Latent sparse domain transfer learning for visual adaptation," *IEEE Trans. Image Processing*, vol. 25, no. 3, pp. 1177–1191, 2016.

[15] Z. Ding and Y. Fu, "Deep transfer low-rank coding for cross-domain learning," *IEEE Trans. Neural Networks and Learning Systems*, vol. 30, no. 6, pp. 1768–1779, 2019.

[16] L. Zhang, S. Wang, G. Huang, W. Zuo, J. Yang, and D. Zhang, "Manifold criterion guided transfer learning via intermediate domain generation," *IEEE Trans. Neural Networks and Learning Systems*, 2019.

[17] L. Zhang, Y. Liu, and P. Deng, "Odor recognition in multiple e-nose systems with cross-domain discriminative subspace learning," *IEEE Trans. Instrumentation and Measurement*, vol. 66, no. 7, pp. 1679–1692, 2017.

[18] Y. Xu, X. Fang, J. Wu, X. Li, and D. Zhang, "Discriminative transfer subspace learning via low-rank and sparse representation." *IEEE Trans. Image Processing*, vol. 25, no. 2, pp. 850–863, 2016.

[19] J. Hoffman, E. Rodner, J. Donahue, T. Darrell, and K. Saenko, "Efficient learning of domain-invariant image representations," *arXiv*, 2013.

[20] W. Deng, A. Lendasse, Y. Ong, I. Tsang, L. Chen, and Q. Zheng, "Domain adaption via feature selection on explicit feature map," *IEEE Trans. Neural Networks and Learning Systems*, vol. 30, no. 4, pp. 1180–1190, 2019.

[21] X. Shen, W. Liu, I. Tsang, Q. Sun, and Y. Ong, "Multilabel prediction via cross-view search," *IEEE Trans. Neural Networks and Learning Systems*, vol. 29, no. 9, pp. 4324–4338, 2018.

[22] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, "Unsupervised visual domain adaptation using subspace alignment," in *ICCV*, 2014, pp. 2960–2967.

[23] B. Sun, J. Feng, and K. Saenko, "Correlation alignment for unsupervised domain adaptation," in *Domain Adaptation in Computer Vision Applications*. Springer, 2017, pp. 153–171.

[24] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," *SIGIR*, 1994.

[25] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification." in *ICML*, 2000, pp. 999–1006.

[26] Y. Bengio, R. Collobert, and J. Weston, "Curriculum learning," in *ICML*, 2009, pp. 41–48.

[27] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *NIPS*, 2010, pp. 1189–1197.

[28] J. Lu, D. Meng, T. Mitamura, and A. G. Hauptmann, "Easy samples first:self-paced reranking for zero-example multimedia search," in *ACM MM*, 2014, pp. 547–556.

[29] J. Lu, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann, "Self-paced curriculum learning," in *AAAI*, 2015, pp. 2694–2700.

[30] L. Jiang, D. Meng, S. I. Yu, Z. Lan, S. Shan, and A. Hauptmann, "Self-paced learning with diversity," *NIPS*, pp. 2078–2086, 2014.

[31] J. Wright, A. Ganesh, S. Rao, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices," *NIPS*, 2009.

[32] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *ICML*, 2010, pp. 663–670.

[33] B. Sun and K. Saenko, "Subspace distribution alignment for unsupervised domain adaptation," in *BMVC*, 2015, pp. 24.1–24.10.

[34] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *CVPR*, 2012, pp. 2066–2073.

[35] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[36] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *ICCV*, 2014, pp. 2200–2207.

[37] J. Wang, Y. Chen, S. Hao, W. Feng, and Z. Shen, "Balanced distribution adaptation for transfer learning," in *ICDM*, 2017, pp. 1129–1134.

[38] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," *CVPR*, pp. 5150–5158, 2017.

[39] J. Hamm and D. D. Lee, "Grassmann discriminant analysis:a unifying view on subspace-based learning," in *ICML*, 2008, pp. 376–383.

[40] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *Arxiv*, 2010.

[41] A. Hansson, L. Zhang, and L. Vandenberghe, "Subspace system identification via weighted nuclear norm optimization," in *CDC*, 2012.

[42] J. F. Cai, Cand, E. J. S, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *Siam Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[43] D. P. Bertsekas, *Constrained optimization and Lagrange Multiplier methods*. Academic Press, 1982.
[44] Y. Zhang, "Recent advances in alternating direction methods: Practice and theory," *IPAM Workshop on Continuous Optimization*, 2010.
[45] J. Eckstein and D. P. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*. Springer-Verlag New York, Inc., 1992.
[46] G. Zhang, H. Sun, G. Xia, and Q. Sun, "Multiple kernel sparse representation-based orthogonal discriminative projection and its cost-sensitive extension," *IEEE Trans. Image Processing*, vol. 25, no. 9, pp. 4271–4285, 2016.
[47] K. R. Mller, S. Mika, G. Rtsch, K. Tsuda, and B. Schlkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Networks*, vol. 12, no. 2, pp. 181–201, 2001.
[48] P. P. Busto and J. Gall, "Open set domain adaptation," in *ICCV*, 2017, pp. 754–763.
[49] H. Lu, C. Shen, Z. Cao, Y. Xiao, and A. D. H. Van, "An embarrassingly simple approach to visual domain adaptation." *IEEE Trans. Image Processing*, vol. 27, no. 7, pp. 3403–3417, 2018.
[50] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko, "Simultaneous deep transfer across domains and tasks," *ICCV*, pp. 4068–4076, 2015.
[51] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *ICML*, 2015, pp. 97–105.
[52] M. Long, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," *NIPS*, 2016.
[53] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *CVPR*, 2014, pp. 1410–1417.
[54] C. Rate and C. Retrieval, "Columbia object image library (coil-20)," *Technical Report*, 2011.
[55] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," *California Institute of Technology*, 2007.
[56] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv*, 2014.
[57] M. Long, J. Wang, D. Shen, D. Shen, and Q. Yang, "Transfer learning with graph co-regularization," in *AAAI*, 2012, pp. 1033–1039.
[58] B. Gong, K. Grauman, and F. Sha, "Connecting the dots with landmarks: discriminatively learning domain-invariant features for unsupervised domain adaptation," in *ICML*, 2014.

Lei Zhang (M'14-SM'18) received his Ph.D degree in Circuits and Systems from the College of Communication Engineering, Chongqing University, Chongqing, China, in 2013. He worked as a Post-Doctoral Fellow with The Hong Kong Polytechnic University, Hong Kong, from 2013 to 2015. He is currently a Professor/Distinguished Research Fellow with Chongqing University. He has authored more than 90 scientific papers in top journals, such as IEEE T-NNLS, IEEE T-IP, IEEE T-MM, IEEE T-IM, IEEE T-SMCA, and top conferences such as ICCV, AAAI, ACM MM, ACCV, etc. His current research interests include machine learning, pattern recognition, computer vision and intelligent systems. He is a Senior Member of IEEE.

Jingru Fu received the B.S. degree from Fuzhou University, Fuzhou, China, in 2017. She is currently working towards the M.S. degree in Learning Intelligence and Vision Essential (LiVE) group at Chongqing University, Chongqing, China. Her current research interests include machine learning, transfer learning and computer vision.

Shanshan Wang received BE and ME from the Chongqing University in 2010 and 2013, respectively. She is a visiting Ph.D student at Singapore University of Technology and Design from 2018 to 2019. She is currently pursuing the Ph.D. degree at Chongqing University. Her current research interests include transfer learning, pattern recognition, computer vision.

David Zhang (F'09) graduated in Computer Science from Peking University in 1974. He received his MSc in 1982 and his PhD in 1985 in Computer Science from the Harbin Institute of Technology (HIT), respectively. From 1986 to 1988 he was a Postdoctoral Fellow at Tsinghua University and then an Associate Professor at the Academia Sinica, Beijing. In 1994 he received his second PhD in Electrical and Computer Engineering from the University of Waterloo, Ontario, Canada. He is a Chair Professor since 2005 at the Hong Kong Polytechnic University where he is the Founding Director of the Biometrics Research Centre (UGC/CRC) supported by the Hong Kong SAR Government in 1998. Professor Zhang is a Croucher Senior Research Fellow, Distinguished Speaker of the IEEE Computer Society, and a Fellow of both IEEE and IAPR.

Zhaoyang Dong obtained Ph.D. from the University of Sydney, Australia in 1999. He is with the University of NSW, Sydney, Australia. His immediate role is Professor and Head of the School of Electrical and Information Engineering, The University of Sydney. He was Ausgrid Chair and Director of the Ausgrid Centre for Intelligent Electricity Networks (CIEN) under the Smart Grid, Smart City national demonstration project. His research interest includes smart grid, power system planning, power system security, load modeling, renewable energy systems, and electricity market. He is an editor of IEEE Transactions on Smart Grid, IEEE Transactions on Sustainable Energy, IEEE PES Transaction Letters and IET Renewable Power Generation. He is Fellow of IEEE.

C. L. Philip Chen (S'88-M'88-SM'94-F'07) is a chair professor of the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau, China. The University of Macaus Engineering and Computer Science programs receiving Hong Kong Institute of Engineers (HKIE) accreditation and Washington/Seoul Accord is his utmost contribution in engineering/computer science education for Macau as the former Dean of the Faculty. His current research interests include systems, cybernetics, and computational intelligence. Dr. Chen was the IEEE SMC Society President from 2012 to 2013 and is now a Vice President of Chinese Association of Automation (CAA). He is a Fellow of IEEE, AAAS, IAPR, CAA, and HKIE. He is the editor-in-chief of the IEEE Transaction on Systems, Man, and Cybernetics: Systems, and an associate editor of several IEEE Transactions. He received 2016 Outstanding Electrical and Computer Engineers award from his alma mater, Purdue University after he graduated from the University of Michigan, Ann Arbor, Michigan, USA.