Class-specific Reconstruction Transfer Learning for Visual Recognition Across Domains

Shanshan Wang, *Student Member, IEEE,* Lei Zhang, *Senior Member, IEEE,* Wangmeng Zuo, *Senior Member, IEEE,* Bob Zhang, *Member, IEEE*

***Abstract*—Subspace learning and reconstruction have been widely explored in recent transfer learning work. Generally, a specially designed projection and reconstruction transfer functions bridging multiple domains for heterogeneous knowledge sharing are wanted. However, we argue that the existing subspace reconstruction based domain adaptation algorithms neglect the class prior, such that the learned transfer function is biased, especially when data scarcity of some class is encountered. Different from those previous methods, in this paper, we propose a novel class-wise reconstruction-based adaptation method called Class-specific Reconstruction Transfer Learning (CRTL), which optimizes a well modeled transfer loss function by fully exploiting intra-class dependency and inter-class independency. The merits of the CRTL are three-fold. 1) Using a class-specific reconstruction matrix to align the source domain with the target domain fully exploits the class prior in modeling the domain distribution consistency, which benefits the cross-domain classification. 2) Furthermore, to keep the intrinsic relationship between data and labels after feature augmentation, a projected Hilbert-Schmidt Independence Criterion (pHSIC), that measures the dependency between data and label, is first proposed in transfer learning community by mapping the data from raw space to RKHS. 3) In addition, by imposing low-rank and sparse constraints on the class-specific reconstruction coefficient matrix, the global and local data structure that contributes to domain correlation can be effectively preserved. Extensive experiments on challenging benchmark datasets demonstrate the superiority of the proposed method over state-of-the-art representation-based domain adaptation methods. The demo code is available in https://github.com/wangshanshanCQU/CRTL**

***Index Terms*—Transfer learning, cross-domain learning, semi-supervised learning, image classification.**

I. Introduction

In statistical machine learning, image classification methods aim to build a classification model from training samples and then apply it to classify test samples. Generally, with the fundamental assumption of machine learning, the fixed model can work well only if the test samples are in similar distribution with the training samples [22]. However, in real world, it is impossible to guarantee that the data with similar semantics has the same feature distribution.

This work was supported by the National Science Fund of China under Grants (61771079) and Chongqing Youth Talent Program, and the Fundamental Research Funds of Chongqing (No. cstc2018jcyjAX0250). (*Corresponding author: Lei Zhang*)

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

W.M. Zuo is with School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China. (E-mail: wmzuo@hit.edu.cn).

B. Zhang is with Department of Computer and Information Science, University of Macau, China. (E-mail: bobzhang@umac.mo).

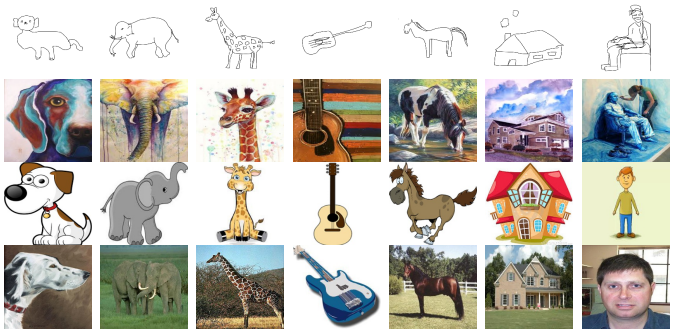

Fig. 1: Different distributions from different domain subjects

Various sampling factors, such as resolutions, illuminations, background, etc., can lead to different distributions, which, in machine learning, is labeled as domain mismatch. Therefore, conventional learning methods fail to handle such issues because the basic assumption of independent identical distribution that general machine learning needs is violated. Fig.1 shows some examples with similar semantics but different distribution. It is not difficult to understand that if the images in the first row are used to train a classifier, the model cannot work well when classifying the images of the second row.

To solve the problem, one straightforward method is to collect a large amount of labeled source data that have shown diverse distribution as the unseen testing data and use them to retrain or fine-tune the classifier model. This is called $data\text{-}driven$ transfer learning (TL), which implies that deep learning is a special case of TL. However, in many real-world applications, collecting and labeling sufficient data is too expensive, and the scarcity of the training data prohibits the model training (e.g., classifier). Therefore, it is essential to make full use of the data from another source.

To this end, transfer learning and domain adaptation were proposed by leveraging a number of data from target domains for knowledge sharing. This is called $model\text{-}driven$ transfer learning, which tends to explore the knowledge transfer from source domain to target domain by exploiting their structural and similar high-level semantic relationship. Generally, one can use distribution different yet semantic relevant domain data to enhance the classification performance by fully exploiting the commonality between domains. By combining $data\text{-}driven$ and $model\text{-}driven$ mind together, deep transfer learning is resulted, which is another quite effective and understudied method, but it is not the focus of this paper.

Transfer learning, that is proposed to leverage the prior

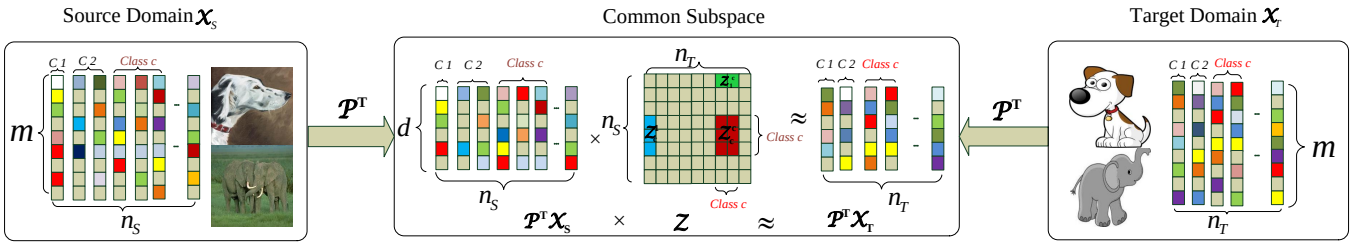

Fig. 2: Illustration of our proposed Class-specific Reconstruction Transfer Learning (CRTL). A structural and class-specific reconstruction matrix $\boldsymbol{\mathcal{Z}}$ is expected, in which $\boldsymbol{\mathcal{Z}}_c^1$ represents reconstructing the 1^{st} class data in target domain by the c^{th} class data in source domain. Intuitively, we wish that the data of class c in target domain can only be represented by the data of the same class in source domain using $\boldsymbol{\mathcal{Z}}_c^c$, so that a more structural reconstruction matrix $\boldsymbol{\mathcal{Z}}$ can be obtained.

knowledge of other different but related *domain data*, is often referred to as *domain adaptation* (DA) [17], [37], [41], [45], [46], [57], [59] in computer vision. Domain adaptation is one of the most promising techniques for cross-domain learning with a well-labeled source domain and a few labeled target domain. The domain data generally share the same task yet different distributions [56]. In order to address the problem of distribution discrepancy, progresses on domain adaptation and transfer learning have been made by researchers in this community. In general, DA methods can be approximately divided into three categories [58]: (1) feature representation based adaptation; (2) classifier adaptation; (3) deep adaptation.

In this paper, we focus on the representation based feature adaptation, and propose a class-specific reconstruction transfer learning (CRTL) model. The proposed method aims at constructing a class-specific and statistical dependence preserved model across domains in reproducing kernel Hilbert space (RKHS). The ultimate goal is to align the feature distribution between domains in some projected subspace by modeling a mutual representation. The general framework of the proposed CRTL method is described in Fig. 2, in which the correspondence matrix is posed to be class-wise such that the class-independency in mutual representation is enhanced.

Maximum Mean Discrepancy (MMD) [15], that acts as a discrepancy metric or criterion to evaluate the distribution mismatch across domains, has been widely used in many unsupervised domain adaptation methods and works well in aligning the global distribution. However, the information of sample categories that benefits to classification is neglected. In this paper, inspired by classifier adaptation, to enhance the correlation between the projected feature and labels, a statistical method that can describe such intrinsic relationship is proposed. Specifically, a Hilbert-Schmidt Independence Criterion (HSIC) [16] formulated with Hilbert-Schmidt norm in RKHS is used to measure the dependency between data and labels. Instead of MMD, the HSIC is introduced in our CRTL by projecting the data from the raw space $\mathbb{R}^D$ to RKHS $\mathbb{H}$, that can be mathematically defined as $\varphi : \mathbb{R}^D \rightarrow \mathbb{H}$.

Due to the domain difference between the source and target domain, a latent projection [11], [30] is generally expected for projecting the source and target data into a common subspace, where the commonality can be easily captured. However, if only learn a common subspace without domain correspondence, the domain knowledge transfer performance can be seriously restricted. To this end, a latent subspace and a reconstruction (correspondence) matrix are simultaneously modeled in our CRTL.

For pursuit of model discrimination, the existing domain adaptation methods [43], [52], [60] attempt to make the subspace projection discriminative by constructing some regularizer and discriminative constraints, rather than considering the class-wise characteristic of the reconstruction matrix. In the existing reconstruction based transfer learning algorithms [18], [43], [52], [63], the class prior distributions that is beneficial to construct a well-designed reconstruction transfer loss function is generally ignored, such that these models are class-biased. Different from those methods, we have an idea to make reconstruction matrix class-specific, which holds a similar assumption with Yang et al. [55] that the data can be better represented by the data of the same class. Therefore, in CRTL, the intra-class dependency and inter-class independency in domain adaptation have been fully exploited and modeled from two aspects. $First$, in modeling, the low-rank and sparsity constraints can also be imposed for enhancing such intra-class dependency and inter-class independency. For example, the sparsity constraint expects that the source data of class c can robustly reconstruct the target data of the same class and the low-rank constraint improves the domain correlation. Low-rank representation (LRR) [29] was originally suggested for block diagonal solution in subspace segmentation. Different from LRR, sparse subspace clustering (SSC) [7] was suggested for data points lying in a union of low-dimensional subspaces, which not only handles the data points near the intersections of subspaces, but also avoids the trivial solution. Benefits from both regularization constraints, the global and local structures can be captured during domain correspondence. $Second$, most importantly, a class-specific reconstruction transfer loss function is specially constructed, such that the learnt transfer matrix is more structural and explanatory. Essentially, when labeled data is deployed by categories, the correspondence matrix shows an intrinsic block-diagonal structure [8], [35].

This work is substantially an extended version of our conference paper [49]. We have completely rewritten the paper to explain the motivation and principle of the proposed CRTL method. For clearly elaborating the optimization details, the solving algorithm and learning procedure have been formulated in the paper. Additionally, more experiments and algorithmic analysis are presented in our manuscript. The

contributions of this work are summarized as follows.

- To keep the intrinsic relationship between domain data and labels, a Hilbert-Schmidt Independence Criterion (HSIC) instead of the MMD criterion is introduced to preserve the $data$-$label$ dependency in reproducing kernel Hilbert space (RKHS). Specifically, a projected HSIC (pHSIC) is proposed with feature augmentation.
- In order to model the class prior distributions across domains, a class-specific reconstruction transfer loss function that fully exploits the intra-class dependency and inter-class independency is proposed. The class discrimination in data reconstruction between source and target domains is then guaranteed.
- Using both LRR and SSC based regularization constraints, the global and local structures are effectively preserved with better block diagonal characteristic that strengthens the domain correlation and stronger robustness that weakens the domain outliers.
- A joint learning framework of the reconstruction transfer matrix and the pHSIC-based common subspace is formulated and extensive experiments demonstrate the superiority over other state-of-the-art techniques.

The rest of paper is organized as follows. Section II revisits the related work and preliminaries. Section III presents the proposed CRTL with optimization. Section IV presents the experiments and comparisons. The discussion is presented in Section V, and finally Section VI concludes this paper.

II. Related Work and Preliminaries

In recent years, a number of transfer learning methods have been proposed, which can be summarized as three categories [58]: classifier adaptation, feature adaptation, and deep domain adaptation. In $classifier\ adaptation$, one representative method called ASVM proposed by Yang et al. [54] tends to learn the perturbation term for adapting the source classifier to the target classifier. Xue et al. [53] proposed a method exploiting the common knowledge to share model parameters across domains based on Dirichlet process prior. Zhang et al. [61] proposed a domain adaptation ELM method for classifier adaptation, and also a robust extreme domain adaptive classifier [62] by using Laplacian graph regularization for local structure preservation. Duan et al. [5] proposed an adaptive multiple kernel learning (AMKL) for cross-domain recognition. Since it is impossible to eliminate the domain disparity between the source and target domain by using classifier adaptation, it is rational to consider the domain adaptation in feature-level.

In $feature\ adaptation$, subspace projection and learning is an appropriate way to achieve the goal, and the classifier trained on the projected source data is also adaptive to the projected target data. Hoffman et al. [17] proposed a feature transformation method for domain shift alignment. Gong et al. [13] proposed a GFK by using geodesic flow kernel to modeling domain shift. Shekhar et al. [44] proposed a shared domain dictionary learning (SDDL) method, which assumes that one common dictionary can be learned for both domains. Another way is representation (reconstruction) based feature adaptation. Shao et al. [43] proposed a LTSL method for reconstruction transfer based on low-rank constraint, in which the subspace and reconstruction matrix are learnt separately. Zhang et al. [63] proposed a latent sparse domain transfer (LSDT) method for visual adaptation, which jointly pursuits a latent subspace and domain correspondence based on sparsity constraint. Xu et al. [52] proposed a supervised discriminative domain transfer learning method (DTSL) based on the joint constraint of low-rank and sparsity, which pursuits a classifier and a reconstruction transfer matrix by adding label information. Recently, deep learning is widely recognized to be a very effective high-level discriminative feature representation technique, which is also introduced in DA/TL community.

In $deep\ domain\ adaptation$, data-driven transfer learning method has witnessed a great achievements [47], [12], [38], [51]. However, when solving domain data problems by using deep learning technology, massive labeled training data are required. The data amount is increased with the increase of convolutional neural network (CNN) parameters [2]. For the tasks of small data, deep learning may not work well. Constructing joint data-driven and model-driven deep transfer learning is an effective way to face with domain data challenge. The number of required data is not as much as deep learning needs by exploiting transfer learning method [36]. On one hand, a network with fewer parameters and smaller structure can be easily re-trained from scratch. On the other hand, a large number of data easily causes overfitting, while transfer learning allows the model to see different domain data. To this end, Tzeng et al. [47] proposed a DDC method which simultaneously achieves knowledge transfer between domains and tasks by using CNN. Long et al. [31] proposed a deep adaptation network (DAN) method by imposing MMD loss on the high-level features across domains. Additionally, Long et al. [33] also proposed a residual transfer network (RTN) which tends to learn a residual classifier based on softmax loss. Very recently, GAN inspired adversarial domain adaptation has been preliminarily studied for domain confusion. For example, Tzeng et al. proposed a novel ADDA method [48] for adversarial domain adaptation based on CNN.

A. HSIC Criterion

In this section, we explicitly minimize the distribution difference between domains to facilitate the information transfer. Different from MMD, from another point of view, Hilbert-Schmidt Independence Criterion [16] is proposed in our paper. HSIC is an independence criterion based on the eigen spectrum of cross-covariance operators in reproducing kernel Hilbert space, which is used to measure the dependency between two sets $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$. Let k_x and k_y denote the kernel function with respect to the RKHS $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{G}}$. According to [16], HSIC independence Criterion is shown in Equation (1).

$$\begin{aligned} &\mathbf{HSIC}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{G}}) \\ &= \parallel C_{\mathcal{XY}} \parallel^2_{HS} = (N-1)^{-2} Tr(\boldsymbol{\mathcal{K}}_{\boldsymbol{\mathcal{X}}} \mathbf{H} \boldsymbol{\mathcal{K}}_{\boldsymbol{\mathcal{Y}}} \mathbf{H}) \\ &s.t. \mathbf{H} = \mathbf{I} - N^{-1} \mathbf{1}_{N\times 1} \mathbf{1}^T_{N\times 1} \end{aligned} \tag{1}$$

where N denotes the size of two sets $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$, $\parallel C_{\mathcal{XY}} \parallel^2_{HS}$ is Hilbert-Schmidt norm of the cross-covariance operator. $\boldsymbol{\mathcal{K}}_{\boldsymbol{\mathcal{X}}}$

and $\boldsymbol{\mathcal{K}_{\mathcal{Y}}}$ are two kernel Gram matrix w.r.t. $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{G}}$ in RKHS, respectively. $\mathbf{H}$ is the centering matrix. With characteristic kernels k_x and k_y, it can be proved that the value of HSIC is zero if and only if $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ are independent [16].

HSIC consists of an empirical estimation of the Hilbert-Schmidt norm of the cross-covariance operator and it has remarkable simplicity advantage compared with previous kernel-based independence criteria. Also, HSIC do not suffer from slow learning rate. In this paper, we exploit this criterion in a semi-supervised manner to match object pairs from two sets by minimizing their dependence based on the HSIC. A projected HSIC criterion is proposed for improving the dependency between enhanced features and labels during transfer.

B. Sparse and Low-rank based Reconstruction Transfer

Three representative reconstruction transfer models proposed by Shao et al. [43], Zhang et al. [63], [60], and Xu et al. [52], respectively, have a common characteristic that the target domain data is expected to be represented by source domain data through a transfer loss function. The key difference lies in the constraint on the reconstruction matrix. Specifically, Shao et al. [43] imposed the transfer a low-rank characteristic. Low-rank representation is advantageous in getting the block diagonal solution for subspace segmentation, so that the global structure can be preserved. Zhang et al. [63] tends to sparsely model the reconstruction such that the outliers from the source domain data can be prevented from transferring to the target domain by fully exploiting the local structure, and robustness is guaranteed. Zhang et al. [60] proposed to model the reconstruction transfer with row sparse by using l_{21}-norm, such that the outliers are better to be prevented for eliminating trivial solution. Besides, the discrimination on the subspace projection is also studied which fully exploits the domain-class consistency instead of domain difference only. Xu et al. [52] considers a more general scenario by jointly modeling the low-rank and sparsity constraint, which takes into account the global and local structure across domains. Also, for discrimination, the label information is considered in guiding the transfer learning phase. The rationale behind of feature reconstruction transfer lies in the domain correlation enhancement through the low-rank and sparsity modeling.

In this paper, the proposed CRTL is closely related with the above four methods, but different in essence. The focus of this paper is the reconstruction transfer loss function construction which aims at modeling the intra-class dependency and inter-class independency. The relation to the above four methods is specially presented in the following section.

C. Relation to Existing Reconstruction based DA/TL Models

Although a series of feature representation based domain adaptation models have been proposed for cross-domain learning [43], [63], [60], [52], they did not consider the fine-grained transfer loss function construction, the class-wise feature representation and the data-label dependency, such that the high-level semantic information is neglected and the transfer is understudied. Generally, the proposed CRTL belongs to feature representation based adaptation, but it is essentially different from these mentioned approaches in several aspects. 1) a fine-grained domain reconstruction loss with class prior information considered is constructed by fully exploiting the intra-class domain dependency and inter-class domain independency. Then, a class-specific reconstruction matrix with domain transfer is resulted. 2) In modeling the subspace projection, the data-label dependency is fully exploited by proposing a projected HSIC criterion that interprets the statistical dependency between two sets in RKHS. To our best knowledge, there is few work on the HSIC criterion [50] for DA/TL problems instead of the over-studied MMD criterion [20]. Although the class discrimination is considered in [60], [52], they only focus on the subspace projection instead of the reconstruction transfer loss function and the reconstruction matrix. Owing to the new perspectives, the proposed CRTL approach yields state-of-the-art performance on challenging benchmark cross-domain visual datasets.

III. The Proposed Class-specific Reconstruction Transfer Learning

A. Notations

In this paper, the source and target domain are defined by subscript S and T. The training set of source and target domain is defined as $\boldsymbol{\mathcal{X}_S} \in \mathbb{R}^{m\times n_S}$ and $\boldsymbol{\mathcal{X}_T} \in \mathbb{R}^{m\times n_T}$, where m denotes dimension of data, n_S and n_T denote the number of samples in source and target domain, respectively. Let $\boldsymbol{\mathcal{X}} = [\boldsymbol{\mathcal{X}}_S, \boldsymbol{\mathcal{X}}_T]$, then $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{m\times N}$, where $N = n_S + n_T$. $\boldsymbol{\mathcal{Y}}$ denotes the data labels. We let $\boldsymbol{\mathcal{P}}$ denote the transformation matrix. $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{n_S\times n_T}$ represents the reconstruction coefficient matrix and $\mathbf{I}$ denotes the identity matrix. $\|\cdot\|_p$, $\|\cdot\|_F$ and $\|\cdot\|_*$ denote l_p -norm, Frobenius norm and nuclear norm, respectively. The superscript T denotes transpose operator, and $Tr(\cdot)$ denotes trace operator of matrix. The kernel Gram matrix $\boldsymbol{\mathcal{K}}$ is defined as $\left[\boldsymbol{\mathcal{K}}\right]_{i,j} =< \varphi(\mathbf{x}_i), \varphi(\mathbf{x}_j) >_{\mathcal{H}} = \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j)$, where $k(\cdot)$ is a kernel function.

B. The Proposed Projected HSIC

What we expect is that after projection, the intrinsic correlation between the projected features and labels can be well exploited for better preservation of the data-label dependency. Thus, the observations $Z_H := \{(x_1, y_1)\dots(x_n, y_n)\}$ can be used to construct Hilbert-Schmidt Independency Criterion after feature augmentation. Note that, the HSIC after feature augmentation is called projected HSIC (pHSIC), which is formulated with the same principle as HSIC, but different in that a projection $\boldsymbol{\mathcal{P}}$ is integrated for knowledge transfer. As described in [16], the proposed pHSIC can be formulated as

$$\begin{aligned} &\mathbf{pHSIC}(Z_H, \boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{G}}) \\ =&(N-1)^{-2}Tr(\boldsymbol{\mathcal{K}}\mathbf{H}\boldsymbol{\mathcal{L}}\mathbf{H}) \\ =&(N-1)^{-2}Tr(k(\boldsymbol{\mathcal{P}}^T\varphi(\boldsymbol{\mathcal{X}}), \boldsymbol{\mathcal{P}}^T\varphi(\boldsymbol{\mathcal{X}}))\mathbf{H}\boldsymbol{\mathcal{L}}\mathbf{H}) \\ &s.t. \mathbf{H} = \mathbf{I} - N^{-1}\mathbf{1}_{N\times 1}\mathbf{1}_{N\times 1}^T \end{aligned} \tag{2}$$

where $\boldsymbol{\mathcal{K}}, \boldsymbol{\mathcal{L}} \in \mathcal{R}^{N\times N}$, $\boldsymbol{\mathcal{K}}_{i,j} = k(x_i', x_j')$, $\boldsymbol{\mathcal{L}}_{i,j} = l(y_i, y_j)$, $\mathbf{H}_{i,j} = \delta_{i,j} - N^{-1}$. $k(\cdot)$ and $l(\cdot)$ denote kernel functions and Gaussian kernel function is considered in this paper. $\boldsymbol{\mathcal{K}} = k(\boldsymbol{\mathcal{X}'}, \boldsymbol{\mathcal{X}'})$, $\boldsymbol{\mathcal{X}'} = [\boldsymbol{\mathcal{X}'_S}, \boldsymbol{\mathcal{X}'_T}]$ denotes the projected data, $\boldsymbol{\mathcal{L}} =$

$l(\boldsymbol{\mathcal{Y}}, \boldsymbol{\mathcal{Y}})$. $\mathbf{H}$ is a centering matrix. φ is a nonlinear function for feature augmentation, which maps the data from the raw space $\mathbb{R}^D$ to RKHS $\mathbb{H}$ defined as $\varphi : \mathbb{R}^D \rightarrow \mathbb{H}$. In this paper, by using Mercer kernel theorem, the nonlinear function φ does not need to be explicit. Thus, by maximizing the pHSIC, the dependency between data and labels can be well preserved and improved in domain transfer learning.

C. The Proposed Transfer Loss with Class Dependency

As described in Fig. 2, in this paper, a better reconstruction transfer matrix $\boldsymbol{\mathcal{Z}}$ as well as the discriminative subspace $\boldsymbol{\mathcal{P}}$ are expected. By leveraging class prior information, we wish to learn a structural and class-specific reconstruction matrix $\boldsymbol{\mathcal{Z}}$, instead of learning a general reconstruction matrix $\boldsymbol{\mathcal{Z}}$ based on the whole dataset. As is shown in Fig. 2, the sub-matrix $\boldsymbol{\mathcal{Z}}_j^i$ in the reconstruction matrix represents the coefficients that the i^{th} class data in target domain is reconstructed by the j^{th} class data in source domain. Intuitively, we wish that the data of class i in target domain can only be represented by the data of the same class in source domain using $\boldsymbol{\mathcal{Z}}_i^i$, such that a more structural and class-wise reconstruction matrix $\boldsymbol{\mathcal{Z}}$ can be obtained. Thus, a class-specific transfer loss function that fully exploits the intra-class dependency and inter-class independency across domains can be constructed.

For the labeled data $\boldsymbol{\mathcal{X}}_T$ and $\boldsymbol{\mathcal{X}}_S$, we expect that after projection with $\boldsymbol{\mathcal{P}}$ and the nonlinear mapping with φ, the intra-class dependency between the source and target data of the same class can be effectively enhanced, while the inter-class dependency of different classes is weakened. To this end, we achieve this goal, that is, the data of class c in target domain can be better expressed by the data of the same class in source domain, by minimizing the domain reconstruction error $\| \boldsymbol{\mathcal{P}}^T\varphi(\boldsymbol{\mathcal{X}}_T^c) - \boldsymbol{\mathcal{P}}^T\varphi(\boldsymbol{\mathcal{X}}_S^c)\boldsymbol{\mathcal{Z}}_c^c \|_F^2$, where $\boldsymbol{\mathcal{X}}_T^c$ represents the target data of class c, $\boldsymbol{\mathcal{X}}_S^c$ represents the source data of class c, and $\boldsymbol{\mathcal{Z}}_c^c$ represents the class-specific representation coefficient sub-matrix with respect to class c. Furthermore, for avoiding the inter-class impact from the data of other classes during domain adaptation, we also consider to minimize the representation error between classes across domains. Specifically, we wish that the target data of class c cannot be expressed by the source data of class k (excluding class c). Therefore, the class representation error can be formulated as $\sum_{k=1,k\neq c} \| \boldsymbol{\mathcal{P}}^T\varphi(\boldsymbol{\mathcal{X}}_S^c)\boldsymbol{\mathcal{Z}}_k^c \|_F^2$, where $\boldsymbol{\mathcal{Z}}_k^c$ represents the reconstruction coefficient sub-matrix between the source data of class k and the target data of class c. With the above analysis, the proposed transfer loss function consisting of domain reconstruction loss (intra-class dependency) and class representation loss (inter-class independency) can be formulated as follows.

$$\begin{aligned} &E(\boldsymbol{\mathcal{X}}_S, \boldsymbol{\mathcal{X}}_T, \boldsymbol{\mathcal{P}}, \boldsymbol{\mathcal{Z}}) \\ =&\sum_{c=1}^{C}(\| \boldsymbol{\mathcal{P}}^T\varphi(\boldsymbol{\mathcal{X}}_T^c) - \boldsymbol{\mathcal{P}}^T\varphi(\boldsymbol{\mathcal{X}}_S^c)\boldsymbol{\mathcal{Z}}_c^c \|_F^2) \\ &+\sum_{c=1}^{C}\sum_{k=1,k\neq c}^{C} \| \boldsymbol{\mathcal{P}}^T\varphi(\boldsymbol{\mathcal{X}}_S^c)\boldsymbol{\mathcal{Z}}_k^c \|_F^2 \end{aligned} \tag{3}$$

In the loss function, we observe that there are two variables $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Z}}$, which represents that domain adaptation is achieved by performing class-wise domain reconstruction under some latent subspace with class-dependency exploited.

D. Model Formulation

As mentioned in [52], [63], the sparsity constraint can help preserve the local structure of data such that each target sample can be well reconstructed by a few very associated samples from the source domain. Furthermore, the sparse subspace clustering (SSC) theory effectively accounts for the noise in data corruption and outliers removal with their intrinsic relevance preserved. In addition, SSC ensures that the data from different domains can be well interlaced and significantly reduce the disparity of the domain distributions. Different from sparsity constraint, low-rank property can better preserve the global structure of data, and it is advantageous to reveal a block-diagonal structure. In constructing the reconstruction matrix $\boldsymbol{\mathcal{Z}}$, in this paper, a joint sparse and low-rank regularizer is used to better account for the local and global characteristics, simultaneously. Eventually, we have imposed the joint sparse plus low-rank constraints on the reconstruction matrix $\boldsymbol{\mathcal{Z}}$. By combining the projected HSIC and the class-wise transfer loss together, the general objective function of the proposed CRTL model can be formulated as follows.

$$\begin{aligned} &\min_{\boldsymbol{\mathcal{P}},\boldsymbol{\mathcal{Z}}} E(\boldsymbol{\mathcal{X}}_S, \boldsymbol{\mathcal{X}}_T, \boldsymbol{\mathcal{P}}, \boldsymbol{\mathcal{Z}}) + \| \boldsymbol{\mathcal{Z}} \|_* \\ &\qquad - \mathbf{pHSIC}(Z_H, \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{L}}) + \| \boldsymbol{\mathcal{Z}} \|_1 \\ &s.t. \boldsymbol{\mathcal{P}}^\mathbf{T}\boldsymbol{\mathcal{P}} = \mathbf{I} \end{aligned} \tag{4}$$

Note that, for solving a convex optimization problem, the sparsity and low-rank property are shown with l_1-norm and nuclear norm, respectively.

Specifically, by substituting the pHSIC criterion in Eq. (2) and the class-wise transfer loss function in Eq. (3), the general CRTL model proposed in Eq. (4) can be rewritten as

$$\begin{aligned} &\min_{\boldsymbol{\mathcal{P}},\boldsymbol{\mathcal{Z}}} \sum_{c=1}^{C}(\| \boldsymbol{\mathcal{P}}^T\varphi(\mathbf{X}_T^c) - \boldsymbol{\mathcal{P}}^T\varphi(\boldsymbol{\mathcal{X}}_S^c)\boldsymbol{\mathcal{Z}}_c^c \|_F^2) + \| \boldsymbol{\mathcal{Z}} \|_* \\ &\qquad + \sum_{c=1}^{C}\sum_{k=1,k\neq c}^{C} \| \boldsymbol{\mathcal{P}}^T\varphi(\boldsymbol{\mathcal{X}}_S^c)\boldsymbol{\mathcal{Z}}_k^c \|_F^2 + \| \boldsymbol{\mathcal{Z}} \|_1 \\ &\qquad - \frac{1}{(N-1)^2} Tr(k(\boldsymbol{\mathcal{P}}^T\varphi(\boldsymbol{\mathcal{X}}), \boldsymbol{\mathcal{P}}^T\varphi(\boldsymbol{\mathcal{X}}))\mathbf{H}\boldsymbol{\mathcal{L}}\mathbf{H}) \\ &s.t. \boldsymbol{\mathcal{P}}^\mathbf{T}\boldsymbol{\mathcal{P}} = \mathbf{I}, \boldsymbol{\mathcal{X}} = [\boldsymbol{\mathcal{X}}_S, \boldsymbol{\mathcal{X}}_T], \mathbf{1}^{1\times n_S}\boldsymbol{\mathcal{Z}} = \mathbf{1}^{1\times n_T} \end{aligned} \tag{5}$$

In this model, the subspace projection $\boldsymbol{\mathcal{P}}$ is imposed an orthogonal constraint and the class-wise reconstruction matrix $\boldsymbol{\mathcal{Z}}$ is imposed a normalization constraint for better solutions.

Generally, we claim that the optimal mapping $\boldsymbol{\mathcal{P}}^*$ can be represented as $(\boldsymbol{\mathcal{P}}^*)^T = \boldsymbol{\Phi}^T\varphi(\boldsymbol{\mathcal{X}})^T$, that is, the projection $\boldsymbol{\mathcal{P}}$ is a linear representation of the data $\varphi(\boldsymbol{\mathcal{X}})$ by using $\boldsymbol{\Phi}$. Therefore, with Mercer kernel theorem, by substituting $\boldsymbol{\mathcal{P}}^*$

into the objective function (5), the proposed CRTL model can be finally reformulated as

$$\begin{aligned}\min_{\boldsymbol{\Phi},\boldsymbol{\mathcal{Z}}} &\sum_{c=1}^{C}(\parallel \boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}_T^c - \boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}_S^c\boldsymbol{\mathcal{Z}}_c^c \parallel_F^2)+ \parallel \boldsymbol{\mathcal{Z}} \parallel_* \\ &+\sum_{c=1}^{C}\sum_{k=1,k\neq c}^{C} \parallel \boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}_S^c\boldsymbol{\mathcal{Z}}_k^c \parallel_F^2 + \parallel \boldsymbol{\mathcal{Z}} \parallel_1 \\ &-\frac{1}{(N-1)^2}Tr(\boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}\mathbf{H}\boldsymbol{\mathcal{L}}\mathbf{H}\boldsymbol{\mathcal{K}}\boldsymbol{\Phi}) \\ s.t.&\boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}\boldsymbol{\Phi}=\mathbf{I}, \mathbf{1}^{1\times n_S}\boldsymbol{\mathcal{Z}}=\mathbf{1}^{1\times n_T}\end{aligned} \tag{6}$$

In the CRTL model, the sub-blocks $\boldsymbol{\mathcal{Z}}_k^c$ and $\boldsymbol{\mathcal{Z}}_c^c$ of $\boldsymbol{\mathcal{Z}}$ are modeled, therefore, in the following we show the details of optimization by introducing some constant matrix.

E. Optimization

In CRTL model, although it seems that three variables are involved in model (6), both $\boldsymbol{\mathcal{Z}}_c^c$ and $\boldsymbol{\mathcal{Z}}_k^c$ are sub-blocks of $\boldsymbol{\mathcal{Z}}$, and therefore can be expressed by using some easily designed constant matrix which is used to represent the sub-matrix by using $\boldsymbol{\mathcal{Z}}$. Then, the model can be solved with respect to two variables $\boldsymbol{\Phi}$ and $\boldsymbol{\mathcal{Z}}$ in (6), respectively. Further, to solve the problem, we adopt a variable alternating optimization strategy, i.e. solving one variable while fixing the other one. With the two updating steps for $\boldsymbol{\Phi}$ and $\boldsymbol{\mathcal{Z}}$, the complete optimization of the proposed method is illustrated as follows.

First, we construct the block matrix $\boldsymbol{\mathcal{A}}_c$, $\boldsymbol{\mathcal{A}}_k$, $\boldsymbol{\mathcal{B}}_c$ as

$$\begin{aligned}\boldsymbol{\mathcal{A}}_c &= \begin{bmatrix}\boldsymbol{\mathcal{A}}_{T1} & \boldsymbol{\mathcal{A}}_{T2} & \dots & \boldsymbol{\mathcal{A}}_{Tc} & \dots & \boldsymbol{\mathcal{A}}_{TC}\end{bmatrix}^T, \\ \boldsymbol{\mathcal{A}}_k &= \begin{bmatrix}\boldsymbol{\mathcal{A}}_{T1} & \boldsymbol{\mathcal{A}}_{T2} & \dots & \boldsymbol{\mathcal{A}}_{Tk} & \dots & \boldsymbol{\mathcal{A}}_{TC}\end{bmatrix}^T, \\ \boldsymbol{\mathcal{B}}_c &= \begin{bmatrix}\boldsymbol{\mathcal{B}}_{S1} & \boldsymbol{\mathcal{B}}_{S2} & \dots & \boldsymbol{\mathcal{B}}_{Sc} & \dots & \boldsymbol{\mathcal{B}}_{SC}\end{bmatrix},\end{aligned}$$

where $\boldsymbol{\mathcal{A}}_c \in \boldsymbol{\mathcal{R}}^{n_T\times n_{Tc}}$, $\boldsymbol{\mathcal{A}}_k \in \boldsymbol{\mathcal{R}}^{n_T\times n_{Tk}}(k\neq c)$, and $\boldsymbol{\mathcal{B}}_c \in \boldsymbol{\mathcal{R}}^{n_{Sc}\times n_S}$ are block matrix, among which $\boldsymbol{\mathcal{A}}_{Tc} \in \mathbb{R}^{n_{Tc}\times n_{Tc}}$, $\boldsymbol{\mathcal{A}}_{Tk} \in \boldsymbol{\mathcal{R}}^{n_{Tk}\times n_{Tk}}$, and $\boldsymbol{\mathcal{B}}_{Sc} \in \mathbb{R}^{n_{Sc}\times n_{Sc}}$ are identity matrix, and others are all $\mathbf{0}$ matrix.

With the exact definition of $\boldsymbol{\mathcal{A}}_c$, $\boldsymbol{\mathcal{A}}_k$, and $\boldsymbol{\mathcal{B}}_c$, we can have $\boldsymbol{\mathcal{Z}}_c = \boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{A}}_c$, $\boldsymbol{\mathcal{Z}}_k^c = \boldsymbol{\mathcal{B}}_c\boldsymbol{\mathcal{Z}}_k = \boldsymbol{\mathcal{B}}_c\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{A}}_k$, and $\boldsymbol{\mathcal{Z}}_c^c = \boldsymbol{\mathcal{B}}_c\boldsymbol{\mathcal{Z}}_c = \boldsymbol{\mathcal{B}}_c\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{A}}_c$. By substituting $\boldsymbol{\mathcal{Z}}_c^c$ and $\boldsymbol{\mathcal{Z}}_k^c$ into the model, then the model (6) can be reformulated as follows.

$$\begin{aligned}\min_{\boldsymbol{\Phi},\boldsymbol{\mathcal{Z}}} &\sum_{c=1}^{C}(\parallel \boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}_T^c - \Phi^T\boldsymbol{\mathcal{K}}_S^c\boldsymbol{\mathcal{B}}_c\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{A}}_c \parallel_F^2)+ \parallel \boldsymbol{\mathcal{Z}} \parallel_* \\ &+\sum_{c=1}^{C}\sum_{k=1,k\neq c}^{C} \parallel \boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}_S^c\boldsymbol{\mathcal{B}}_c\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{A}}_k \parallel_F^2 + \parallel \boldsymbol{\mathcal{Z}} \parallel_1 \\ &-\frac{1}{(N-1)^2}Tr(\boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}\mathbf{H}\boldsymbol{\mathcal{L}}\mathbf{H}\boldsymbol{\mathcal{K}}\boldsymbol{\Phi}) \\ s.t.&\boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}\boldsymbol{\Phi}=\mathbf{I}, \mathbf{1}^{1\times n_S}\boldsymbol{\mathcal{Z}}=\mathbf{1}^{1\times n_T}\end{aligned} \tag{7}$$

Further, to solve the problem (7), a variable alternating optimization strategy is considered, i.e., one variable is solved by frozen the other one. In addition, the inexact augmented Lagrangian multiplier (IALM) and alternating direction method of multipliers (ADMM) can be used to efficiently solve each variable, respectively. With the two updating steps for $\boldsymbol{\Phi}$ and $\boldsymbol{\mathcal{Z}}$, the optimization details of the proposed method are illustrated as follows.

First, by introducing two auxiliary variables $\boldsymbol{\mathcal{J}}$ and $\boldsymbol{\mathcal{G}}$ with respect to the correspondence matrix $\boldsymbol{\mathcal{Z}}$, the minimization problem (7) with new equality constraints introduced can be re-written as follows

$$\begin{aligned}\min_{\boldsymbol{\Phi},\boldsymbol{\mathcal{Z}}} &\sum_{c=1}^{C}(\parallel \boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}_T^c - \boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}_S^c\boldsymbol{\mathcal{B}}_c\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{A}}_c \parallel_F^2) \\ &+\sum_{c=1}^{C}\sum_{k=1,k\neq c}^{C} \parallel \boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}_S^c\boldsymbol{\mathcal{B}}_c\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{A}}_k \parallel_F^2 \\ &-\frac{1}{(N-1)^2}Tr(\boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}\mathbf{H}\boldsymbol{\mathcal{L}}\mathbf{H}\boldsymbol{\mathcal{K}}\boldsymbol{\Phi}) \\ &+\parallel \boldsymbol{\mathcal{J}} \parallel_* + \parallel \boldsymbol{\mathcal{G}} \parallel_1 \\ s.t.&\boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}\boldsymbol{\Phi}=\mathbf{I}, \mathbf{1}^{1\times n_S}\boldsymbol{\mathcal{Z}}=\mathbf{1}^{1\times n_T}, \boldsymbol{\mathcal{Z}}=\boldsymbol{\mathcal{J}}, \boldsymbol{\mathcal{Z}}=\boldsymbol{\mathcal{G}}\end{aligned} \tag{8}$$

Furthermore, with the augmented Lagrange function [27], the above model (8) can be converted into the following minimization problem

$$\begin{aligned}\min_{\boldsymbol{\Phi},\boldsymbol{\mathcal{Z}},\boldsymbol{\mathcal{J}},\boldsymbol{\mathcal{G}}} &\sum_{c=1}^{C}(\parallel \boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}_T^c - \boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}_S^c\boldsymbol{\mathcal{B}}_c\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{A}}_c \parallel_F^2)+ \parallel \boldsymbol{\mathcal{J}} \parallel_* + \parallel \boldsymbol{\mathcal{G}} \parallel_1 \\ &+\sum_{c=1}^{C}\sum_{k=1,k\neq c}^{C} \parallel \boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}_S^c\boldsymbol{\mathcal{B}}_c\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{A}}_k \parallel_F^2 + Tr(\boldsymbol{\mathcal{R}}_1^T(\boldsymbol{\mathcal{Z}}-\boldsymbol{\mathcal{J}})) \\ &-\frac{1}{(N-1)^2}Tr(\boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}\mathbf{H}\boldsymbol{\mathcal{L}}\mathbf{H}\boldsymbol{\mathcal{K}}\boldsymbol{\Phi}) + Tr(\boldsymbol{\mathcal{R}}_2^T(\boldsymbol{\mathcal{Z}}-\boldsymbol{\mathcal{G}})) \\ &+Tr(\boldsymbol{\mathcal{R}}_3^T(\mathbf{1}^{1\times n_S}\boldsymbol{\mathcal{Z}}-\mathbf{1}^{1\times n_T})) + \frac{\mu}{2}(\parallel \boldsymbol{\mathcal{Z}}-\boldsymbol{\mathcal{J}} \parallel_F^2) \\ &+\frac{\mu}{2}(\parallel \boldsymbol{\mathcal{Z}}-\boldsymbol{\mathcal{G}} \parallel_F^2) + \frac{\mu}{2}(\parallel \mathbf{1}^{1\times n_S}\boldsymbol{\mathcal{Z}}-\mathbf{1}^{1\times n_T} \parallel_F^2)\end{aligned} \tag{9}$$

where $\boldsymbol{\mathcal{R}}_1$, $\boldsymbol{\mathcal{R}}_2$, $\boldsymbol{\mathcal{R}}_3$ denote the Lagrange multipliers and μ is a penalty parameter. The above model can be divided into two sub-problems. For clarity and easy following, we then present the two updating steps for $\boldsymbol{\Phi}$ and $\boldsymbol{\mathcal{Z}}$ separately.

- **Step 1 (Update $\boldsymbol{\Phi}$)**

By fixing $\boldsymbol{\mathcal{Z}}$, $\boldsymbol{\mathcal{J}}$ and $\boldsymbol{\mathcal{G}}$, $\boldsymbol{\Phi}$ can be updated by solving the following optimization problem

$$\begin{aligned}\boldsymbol{\Phi}^* = arg\min_{\boldsymbol{\Phi}} &\sum_{c=1}^{C}(\parallel \boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}_T^c - \boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}_S^c\mathbf{B}_c\boldsymbol{\mathcal{Z}}\mathbf{A}_c \parallel_F^2) \\ &+\sum_{c=1}^{C}\sum_{k=1,k\neq c}^{C} \parallel \boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}_S^c\mathbf{B}_c\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{A}}_k \parallel_F^2 \\ &-\frac{1}{(N-1)^2}Tr(\boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}\mathbf{H}\boldsymbol{\mathcal{L}}\mathbf{H}\boldsymbol{\mathcal{K}}\boldsymbol{\Phi}) \\ s.t.&\boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}\boldsymbol{\Phi}=\mathbf{I}\end{aligned} \tag{10}$$

We can derive the solution of the projection vectors in $\boldsymbol{\Phi}_K$ one by one. To obtain the i^{th} column in $\boldsymbol{\Phi}_K$ [denoted as $\boldsymbol{\Phi}_{K(:,i)}$], we can rewrite the problem (10) as

$$\begin{aligned}\boldsymbol{\Phi}_{K+1(:,i)} = \min_{\boldsymbol{\Phi}_{K(:,i)}} &\sum_{c=1}^{C}(\parallel \boldsymbol{\Phi}_{K(:,i)}^T\mathbf{H}_{1c} \parallel_2^2) \\ &+\sum_{c=1}^{C}\sum_{k=1,k\neq c}^{C} \parallel \boldsymbol{\Phi}_{K(:,i)}^T\mathbf{H}_{2c}^k \parallel_2^2 \\ &-\frac{1}{(N-1)^2}Tr(\boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}\mathbf{H}\boldsymbol{\mathcal{L}}\mathbf{H}\boldsymbol{\mathcal{K}}\boldsymbol{\Phi}) \\ &+\alpha_i(\boldsymbol{\Phi}_{K(:,i)}^T\boldsymbol{\mathcal{K}}\boldsymbol{\Phi}_{K(:,i)}-\mathbf{I})\end{aligned} \tag{11}$$

Algorithm 1 The Proposed CRTL

Input: $\boldsymbol{\mathcal{X}}_S \in \mathcal{R}^{m\times n_S}$, $\boldsymbol{\mathcal{X}}_T \in \mathcal{R}^{m\times n_T}$,
$\boldsymbol{\mathcal{Y}}_S \in \mathcal{R}^{n_S\times 1}$, $\boldsymbol{\mathcal{Y}}_T \in \mathcal{R}^{n_T\times 1}$
Procedure:
1. Compute $\boldsymbol{\mathcal{K}}_T = \varphi(\boldsymbol{\mathcal{X}})^T\varphi(\boldsymbol{\mathcal{X}}_T)$, $\boldsymbol{\mathcal{K}}_S = \varphi(\boldsymbol{\mathcal{X}})^T\varphi(\boldsymbol{\mathcal{X}}_S)$,
$\boldsymbol{\mathcal{X}} = [\boldsymbol{\mathcal{X}}_S, \boldsymbol{\mathcal{X}}_T]$, $\boldsymbol{\mathcal{K}} = \varphi(\boldsymbol{\mathcal{X}})^T\varphi(\boldsymbol{\mathcal{X}})$
2. Construct constant matrixs $\boldsymbol{\mathcal{A}}_c$, $\boldsymbol{\mathcal{A}}_k$, $\boldsymbol{\mathcal{B}}_c$, there is
$\boldsymbol{\mathcal{Z}}_c = \boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{A}}_c$
$\boldsymbol{\mathcal{Z}}_k^c = \boldsymbol{\mathcal{B}}_c\boldsymbol{\mathcal{Z}}_k = \boldsymbol{\mathcal{B}}_c\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{A}}_k$
$\boldsymbol{\mathcal{Z}}_c^c = \boldsymbol{\mathcal{B}}_c\boldsymbol{\mathcal{Z}}_c = \boldsymbol{\mathcal{B}}_c\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{A}}_c$
3. Initialize: add auxiliary variable $\boldsymbol{\mathcal{J}}$, $\boldsymbol{\mathcal{G}}$, where $\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{J}} = \boldsymbol{\mathcal{G}}$
add Lag-multipliers $\boldsymbol{\mathcal{R}}_1$, $\boldsymbol{\mathcal{R}}_2$, $\boldsymbol{\mathcal{R}}_3$ and penalty parameter μ.
4. **While** not converge **do**
4.1 **Step1**: Fix $\boldsymbol{\mathcal{J}}$, $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{\mathcal{Z}}$, and update $\boldsymbol{\Phi}$ by solving eigenvalue decomposition.
4.2 **Step2**: Fix $\boldsymbol{\Phi}$, and update $\boldsymbol{\mathcal{Z}}$ using ADMM;
Fix $\boldsymbol{\mathcal{Z}}$ and $\boldsymbol{\mathcal{G}}$, and update $\boldsymbol{\mathcal{J}}$ by using the singular value thresholding (SVT) [1] operator on problem (14).
Fix $\boldsymbol{\mathcal{Z}}$ and $\boldsymbol{\mathcal{J}}$, and update $\boldsymbol{\mathcal{G}}$ by shrinkage operator on problem (17).
Fix $\boldsymbol{\mathcal{J}}$ and $\boldsymbol{\mathcal{G}}$, and update $\boldsymbol{\mathcal{Z}}$ according to Gradient descent operator on problem (19).
4.3 Update the multipliers $\boldsymbol{\mathcal{R}}_1$, $\boldsymbol{\mathcal{R}}_2$ and $\boldsymbol{\mathcal{R}}_3$
$\boldsymbol{\mathcal{R}}_1 = \boldsymbol{\mathcal{R}}_1 + \mu(\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{J}})$
$\boldsymbol{\mathcal{R}}_2 = \boldsymbol{\mathcal{R}}_2 + \mu(\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{G}})$
$\boldsymbol{\mathcal{R}}_3 = \boldsymbol{\mathcal{R}}_3 + \mu(\mathbf{1}^{1\times N_S}\boldsymbol{\mathcal{Z}} - \mathbf{1}^{1\times N_T})$
4.4 Update the parameter μ
$\mu = min(\mu \times 1.01, max_\mu)$
4.5 Check convergence
end while
Output: $\boldsymbol{\Phi}$ and $\boldsymbol{\mathcal{Z}}$.

where $\mathbf{H}_{1c} = \boldsymbol{\mathcal{K}}_T^c - \boldsymbol{\mathcal{K}}_S^c\boldsymbol{\mathcal{B}}_c\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{A}}_c, \mathbf{H}_{2c}^k = \boldsymbol{\mathcal{K}}_S^c\boldsymbol{\mathcal{B}}_c\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{A}}_k$. Further, by setting the derivative of problem (11) with respect to $\boldsymbol{\Phi}_{K(:,i)}$ to be zero, we have

$$\begin{aligned}&(\sum_{c=1}^{C}(\mathbf{H}_{1c}\mathbf{H}_{1c}^{\mathbf{T}}) + \sum_{c=1}^{C}\sum_{k=1,k\neq c}^{C}\mathbf{H}_{2c}^k(\mathbf{H}_{2c}^k)^{\mathbf{T}}\\ &- \frac{1}{(N-1)^2}\boldsymbol{\mathcal{K}}\mathbf{H}\boldsymbol{\mathcal{L}}\mathbf{H}\boldsymbol{\mathcal{K}})\boldsymbol{\Phi}_{K(:,i)}\\ &= -\alpha_i\boldsymbol{\mathcal{K}}\boldsymbol{\Phi}_{K(:,i)}\end{aligned} \tag{12}$$

It is clear that $\boldsymbol{\Phi}_K$ can be obtained by solving an Eigen-decomposition problem, and $\boldsymbol{\Phi}_{K(:,i)}$ is the i^{th} eigenvector corresponding to the i^{th} smallest eigenvalue.

- **Step 2 (Update $\boldsymbol{\mathcal{Z}}$)**

First, the updates of $\boldsymbol{\mathcal{J}}_{K+1}$ and $\boldsymbol{\mathcal{G}}_{K+1}$ are introduced.

After dropping out the irrelevant terms with respect to $\boldsymbol{\mathcal{J}}_{K+1}$, problem (9) can be rewritten as

$$\begin{aligned}\boldsymbol{\mathcal{J}}_{K+1} =& \min_{\boldsymbol{\mathcal{J}}_K} \| \boldsymbol{\mathcal{J}}_K \|_* + Tr(\boldsymbol{\mathcal{R}}_{1K}^T(\boldsymbol{\mathcal{Z}}_K - \boldsymbol{\mathcal{J}}_K))\\ &+ \frac{\mu_K}{2}\| \boldsymbol{\mathcal{Z}}_K - \boldsymbol{\mathcal{J}}_K \|_F^2\end{aligned} \tag{13}$$

It can be further rewritten as

$$\boldsymbol{\mathcal{J}}_{K+1} = \min_{\boldsymbol{\mathcal{J}}_K} \| \boldsymbol{\mathcal{J}}_K \|_* + \frac{\mu_K}{2}\| \boldsymbol{\mathcal{J}}_K - (\boldsymbol{\mathcal{Z}}_K + \frac{\boldsymbol{\mathcal{R}}_{1K}}{\mu_K}) \|_F^2 \tag{14}$$

The problem (14) can be effectively solved by the singular value thresholding (SVT) operator [1].

After dropping out the irrelevant terms with respect to $\boldsymbol{\mathcal{G}}_{K+1}$, the problem (9) can be rewritten as

$$\begin{aligned}\boldsymbol{\mathcal{G}}_{K+1} =& \min_{\boldsymbol{\mathcal{G}}_K} \| \boldsymbol{\mathcal{G}}_K \|_1 + Tr(\boldsymbol{\mathcal{R}}_{2K}^T(\boldsymbol{\mathcal{Z}}_K - \boldsymbol{\mathcal{G}}_K))\\ &+ \frac{\mu_K}{2}\| \boldsymbol{\mathcal{Z}}_K - \boldsymbol{\mathcal{G}}_K \|_F^2\end{aligned} \tag{15}$$

It can be further simplified as

$$\boldsymbol{\mathcal{G}}_{K+1} = \min_{\boldsymbol{\mathcal{G}}_K} \| \boldsymbol{\mathcal{G}}_K \|_1 + \frac{\mu_K}{2}\| \boldsymbol{\mathcal{G}}_K - (\boldsymbol{\mathcal{Z}}_K + \frac{\boldsymbol{\mathcal{R}}_{2K}}{\mu_K}) \|_F^2 \tag{16}$$

According to the shrinkage operator [28], the solution of (16) can be obtained as

$$\boldsymbol{\mathcal{G}}_{K+1} = shrink(\boldsymbol{\mathcal{Z}}_K + \frac{\boldsymbol{\mathcal{R}}_{2K}}{\mu_K}, \frac{1}{\mu_K}) \tag{17}$$

By dropping out those terms independent of $\boldsymbol{\mathcal{Z}}$ in the problem (9), we can have

$$\begin{aligned}\min_{\boldsymbol{\mathcal{Z}}} &\sum_{c=1}^{C}(\| \boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}_T^c - \boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}_S^c\boldsymbol{\mathcal{B}}_c\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{A}}_c \|_F^2) + Tr(\boldsymbol{\mathcal{R}}_1^T(\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{J}}))\\ &+ \sum_{c=1}^{C}\sum_{k=1,k\neq c}^{C}\| \boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}_S^c\boldsymbol{\mathcal{B}}_c\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{A}}_k \|_F^2 + Tr(\boldsymbol{\mathcal{R}}_2^T(\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{G}}))\\ &+ Tr(\boldsymbol{\mathcal{R}}_3^T(\mathbf{1}^{1\times n_S}\boldsymbol{\mathcal{Z}} - \mathbf{1}^{1\times n_T})) + \frac{\mu}{2}(\| \boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{J}} \|_F^2)\\ &+ \frac{\mu}{2}(\| \boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{G}} \|_F^2) + \frac{\mu}{2}(\| \mathbf{1}^{1\times n_S}\boldsymbol{\mathcal{Z}} - \mathbf{1}^{1\times n_T} \|_F^2)\end{aligned} \tag{18}$$

We can see from problem (18) that it is hard to get the closed-form solution of $\boldsymbol{\mathcal{Z}}$. According to gradient descent operator [40], the expression of $\boldsymbol{\mathcal{Z}}_{K+1}$ is solved as

$$\boldsymbol{\mathcal{Z}}_{K+1} = \boldsymbol{\mathcal{Z}}_K - \alpha \cdot \frac{\bigtriangledown(\boldsymbol{\mathcal{Z}})}{\| \bigtriangledown(\boldsymbol{\mathcal{Z}}) \|}, \tag{19}$$

where the derivative with respect to $\boldsymbol{\mathcal{Z}}$ is expressed as

$$\begin{aligned}\bigtriangledown(\boldsymbol{\mathcal{Z}}) =& \sum_{c=1}^{C}(-\boldsymbol{\mathcal{B}}_c^T(\boldsymbol{\mathcal{K}}_S^c)^T\boldsymbol{\Phi}(\boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}_T^c - \boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}_S^c\boldsymbol{\mathcal{B}}_c\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{A}}_c)\boldsymbol{\mathcal{A}}_c^T)\\ &+ \sum_{c=1}^{C}\sum_{k=1,k\neq c}^{C}(\boldsymbol{\mathcal{B}}_c^T(\boldsymbol{\mathcal{K}}_S^c)^T\boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{\mathcal{K}}_S^c\boldsymbol{\mathcal{B}}_c\boldsymbol{\mathcal{Z}}\boldsymbol{\mathcal{A}}_k\boldsymbol{\mathcal{A}}_k^T)\\ &+ \frac{\mu}{2}(\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{J}}) + \frac{\mu}{2}(\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{G}})\\ &+ \frac{\mu}{2}\mathbf{1}^{n_S\times 1}(\mathbf{1}^{1\times n_S}\boldsymbol{\mathcal{Z}} - \mathbf{1}^{1\times n_T})\\ &+ \frac{\boldsymbol{\mathcal{R}}_1}{2} + \frac{\boldsymbol{\mathcal{R}}_2}{2} + \frac{\mathbf{1}^{n_S\times 1}\boldsymbol{\mathcal{R}}_3}{2}\end{aligned} \tag{20}$$

In detail, the iterative optimization procedure of the proposed CRTL is summarized in **Algorithm 1**.

F. Classification

In this paper, the superiority of the proposed method is shown through the cross-domain classification performance on the projected source data and target data, which can be represented as $\boldsymbol{\mathcal{X}}_S{}' = \boldsymbol{\Phi}^T\varphi(\boldsymbol{\mathcal{X}})^T\varphi(\boldsymbol{\mathcal{X}}_S)$ and $\boldsymbol{\mathcal{X}}_T{}' = \boldsymbol{\Phi}^T\varphi(\boldsymbol{\mathcal{X}})^T\varphi(\boldsymbol{\mathcal{X}}_T)$, respectively. Then, the general classifiers (e.g. SVM, least square method, SRC) can be used for training on the augmented training data $[\boldsymbol{\mathcal{X}}_S{}', \boldsymbol{\mathcal{X}}_T{}']$ with label $\boldsymbol{\mathcal{Y}} = [\boldsymbol{\mathcal{Y}}_S, \boldsymbol{\mathcal{Y}}_T]$. Notably, for the COIL-20 experiment, in order to keep the same experimental setting with DTSL [52], the classifier is trained only on $\boldsymbol{\mathcal{X}}_S{}'$ with label $\boldsymbol{\mathcal{Y}}_S$. Finally, the recognition performance is verified and compared based on the unseen target test data $\boldsymbol{\mathcal{X}}_{Tu}{}' = \boldsymbol{\Phi}^T\varphi(\boldsymbol{\mathcal{X}})^T\varphi(\boldsymbol{\mathcal{X}}_{Tu})$.

TABLE I: Recognition accuracy (%) of different domain adaptation on Office-31 recognition

Tasks	A-SVM	GFK [13]	SGF [14]	SA [9]	RDALR [21]	LTSL [43]	JDA [32]	CRTL
$A \rightarrow W$	42.2	46.4	45.1	48.4	50.7	**53.5**	34.8	46.2
$D \rightarrow W$	33.0	61.3	61.4	**61.8**	36.9	54.4	52.1	59.5
$W \rightarrow D$	26.0	66.3	63.4	**65.7**	32.9	59.1	47.3	60.1
$AD \rightarrow W$	30.4	34.3	31.0	54.4	36.9	30.2	42.6	**59.9**
$AW \rightarrow D$	25.3	52.0	25.0	37.5	31.2	43.0	44.4	**59.0**
$DW \rightarrow A$	17.3	**21.7**	15.0	16.5	20.9	17.1	18.3	20.3
Average	29.0	47.0	40.2	47.4	34.9	42.9	39.9	**50.8**

IV. Experiments

In this section, for evaluating the proposed method, extensive experiments have been conducted on cross-domain visual recognition tasks with many challenging benchmark DA datasets. Specifically, we have performed cross-domain object recognition (Office-31 dataset, 4DA object dataset and COIL-20 object dataset), heterogeneous image classification (MSRC-VOC2007 datasets and PACS datasets), cross-pose face recognition (Multi-PIE face dataset), and cross-domain handwritten digit recognition (USPS dataset, SEMEION dataset and MNIST dataset). Several closely related methods, such as SGF [14], GFK [13], SA [9], LTSL [43], DTSL [52], LSDT [63],and JDA [32] have been compared. Additionally, we have also compared with deep domain adaptation methods, such as DDC [47], DAN [31], RTN [33], JAN [34] based on some deep learned features. Further, two recent deep adversarial domain adaptation methods, including DANN [10] and ADDA [48] are compared to demonstrate the superiority of our model.

A. Cross-domain Object Recognition

The benchmark Office-31 dataset, 4DA dataset, 4DA-CNN office dataset with deep feature representation and COIL-20 object dataset have been tested.

TABLE II: Comparisons (%) with deep transfer and deep adversarial transfer models on Office-31 dataset

Office-31	A→W	D→W	W→D	A→D	D→A	W→A	Avg.
Source Only	68.4	96.7	99.3	68.9	62.5	60.7	76.1
DDC [47]	75.6	96.0	98.2	76.5	62.2	61.5	78.3
DAN [31]	80.5	**97.1**	**99.6**	78.6	63.6	62.8	80.4
RTN [33]	84.5	96.8	99.4	77.5	66.2	64.8	81.6
DANN [10]	**82.0**	96.9	99.1	**79.7**	68.2	67.4	82.2
ADDA [48]	86.2	96.2	98.4	77.8	69.5	68.9	82.9
JAN [34]	85.4	97.4	99.8	84.7	68.6	70.0	84.3
CRTL	77.4	95.7	97.6	79.5	**81.9**	**81.8**	**85.6**

Results on Office-31 dataset (Amazon, DSLR and Webcam[1]**)** [63]: It contains three domains including Amazon (A), DSLR (D) and Webcam (W). 31 classes are contained in each domain. Some examples are shown in Fig. 3. By following the setting in [63], 20 samples per class are selected if Amazon is chosen as the source domain and 8 samples per class are selected if other domain is chosen as the source domain. For target domain, 3 samples in each class are chosen for training and others are used for testing. The experiments are employed in single source domain and multiple source domains, respectively. The experimental results are shown in Table I. From the results, we can observe that our performance outperforms several state-of-the-art methods.

Fig. 3: Some images from 4DA datasets

Additionally, Office-31 dataset is a common dataset in deep domain adaptation methods. MMD and adversarial learning are two mainstreams in deep DA problems. In our method, we adopt the HSIC criterion instead of the MMD metric. Additionally, we have mentioned the adversarial learning in related work. Therefore, we show the comparisons between ours and deep methods (MMD based and adversarial models) on Office-31 dataset, including DAN [31] (MMD based method), DANN [10] and ADDA [48] (adversarial learning based methods) and some other deep methods such as RTN [33] and JAN [34]. Specifically, for fair comparison, we extract the deep features of Office-31 from the ResNet50, then compare with some famous deep methods From Table II, we can observe that the proposed CRTL, as a shallow transfer learning method, has shown very good competitiveness. This also demonstrates that the shallow transfer learning model can be accompanied with deeply learned features for better addressing domain discrepancy with less resources.

Results on 4DA dataset (Amazon, DSLR, Webcam and Caltech 256[2]**)** [13]: In 4DA dataset, four domains coming from Office-31 and an extra Caltech 256 simplified as A, D, W, and C are included, with each domain 10 object classes are contained. Some images are shown in Fig. 3. In the experiment, the standard experimental protocol is used by following [13]. Specifically, 20 samples per class are selected from Amazon and 8 samples per class from DSLR, Webcam

[1]http://www.eecs.berkeley.edu/~mfritz/domainadaptation/

[2]http://www.vision.caltech.edu/Image_Datasets/Caltech256/

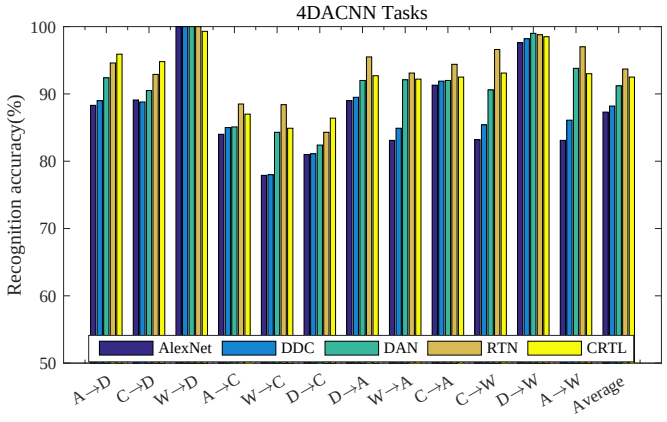

Fig. 4: Comparison with deep transfer learning methods

and Caltech are randomly chosen when they are treated as source domain. 3 samples per class are chosen when they are used as target domain, while the rest data in target domain is used for performance test. With the domain adaptation setting, 12 cross-domain tasks are tested, e.g., $A \rightarrow D$, $C \rightarrow D$, etc. Note that, the 800-bin SURF features in [63] are used. Compared with several state-of-the-art methods, as shown in Table III, our method shows an improvement in average.

Results on 4DA-CNN dataset (Amazon, DSLR, Webcam and Caltech 256) [3]: In 4DA-CNN dataset, the features are extracted by feeding the raw 4DA data into the pre-trained convolutional neural network (i.e., AlexNet) on ImageNet [23], with 8 layers consisting of 5 convolutional layers and 3 fully connected layers. The features with dimension of 4096 from the 6^{th} and 7^{th} layers (i.e., DeCAF-f_6 and DeCAF-f_7 features [3]) are explored. We have highlighted the best results in Table IV, from which we can observe that the average recognition accuracy of the proposed method outperforms other state-of-the-art models, and the superiority is demonstrated.

The compared methods in Table IV are shallow transfer learning. It is interesting but challenging to compare with the deep transfer learning methods, such as AlexNet [23], DDC [47], DAN [31] and RTN [33]. In this paper, the features of the 7^{th} layer are experimented. The comparison is described in Fig. 4, from which we can observe that our proposed method ranks the second in average performance (92.5%), which is inferior to the deep residual transfer network (RTN), but still better than other three deep transfer learning models. The comparison shows that the proposed CRTL, as a shallow transfer learning method, has very good competitiveness.

Results on COIL-20 data: Columbia Object Image Library [39]: The COIL-20 dataset[3] as described in Fig. 5 contains 20 objects with 1440 gray scale images (72 multi-pose images per object). Each image has 128×128 pixels with 256 gray levels per pixel. In the experiment, by following the experimental protocol in [52], the size of each image is cropped into 32×32. The dataset is divided into two subsets COIL1 (C1) and COIL2 (C2), with each 2 quadrants are contained. Specifically, the C1 set contains quadrants 1 and 3, including the directions of [0°, 85°] and [180°, 265°] and

[3]http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php

Fig. 5: Some examples from COIL-20 dataset

the C2 contains quadrants 2 and 4, including the directions of [90°, 175°] and [270°, 355°]. The two subsets are distribution different but relevant in semantic, and result in a DA problem. We have used two settings in constructing the source and target data: COIL1 (source) vs COIL2 (target) ($C1 \rightarrow C2$) and COIL2 (source) vs COIL1 (target) ($C2 \rightarrow C1$).

The experimental results of cross-domain 3D object recognition are shown in Table V, from which we observe that our proposed CRTL method achieves the second best performance over other related methods in both tasks (i.e., 87.0% for $C1 \rightarrow C2$ and 86.5% for $C2 \rightarrow C1$). The recognition accuracy of CRTL is lower than the JDA in this benchmark. However, the improvements over other reconstruction based methods, such as RDALR, LTSL, LSDT, and DTSL demonstrate that learning a class-specific reconstruction matrix can effectively promote the domain adaptation performance. The superiority of our proposed class-specific transfer loss is demonstrated.

Noteworthily, the performances of our method on 4DA dataset and COIL-20 data do not have a high improvement compared with other methods especially with JDA. The reason may be that JDA considers not only the marginal distribution but also conditional distribution, while the category discrepancy is large in these two benchmark datasets.

B. Cross-domain Image Classification

In this section, the experiments on MSRC-VOC 2007 dataset and PACS dataset have been conducted for cross-domain image classification.

Results on MSRC[4] and VOC2007[5] datasets [52]: The MSRC dataset contains 4323 images with 18 classes and the VOC2007 dataset contains 5011 images with 20 concepts. Generally, MSRC consists of standard images for benchmark evaluation, while VOC2007 composes of arbitrary photos from Flickr. Therefore, they follow significantly different distributions. In experiment, for cross-domain classification, 6 common semantic classes: airplane, bicycle, bird, car, cow and sheep from both datasets have been explored. Several example images are shown in Fig. 6, which shows the heterogeneous image feature. For fairness, we follow [52] to construct the cross-domain image dataset MSRC vs. VOC ($M \rightarrow V$) by selecting 1269 images from MSRC as the source domain, and 1530 images from VOC2007 as the target domain. Then

[4]http://research.microsoft.com/en-us/projects/objectclassrecognition
[5]http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007

TABLE III: Recognition accuracy (%) over 10 object categories on 4DA-SURF with hand-crafted feature representation by using different domain adaptation algorithms

4DA Tasks	A-SVM	HFA [4]	ARC-t [24]	SGF [14]	GFK [13]	LTSL [43]	JDA [32]	CRTL
$A \rightarrow D$	**55.9**	52.7	50.2	46.9	50.9	50.4	44.2	52.1
$C \rightarrow D$	**55.8**	51.9	50.6	50.2	55.0	49.5	44.1	54.0
$W \rightarrow D$	55.1	51.7	71.3	78.6	75.0	82.6	**86.3**	72.5
$A \rightarrow C$	32.0	31.1	37.0	37.5	39.6	41.5	**44.9**	43.0
$W \rightarrow C$	30.4	29.4	31.9	32.9	32.8	36.7	29.8	**37.5**
$D \rightarrow C$	31.7	31.0	33.5	32.9	33.9	36.2	34.4	**37.4**
$D \rightarrow A$	45.7	45.8	42.5	44.9	**46.2**	45.7	44.6	45.6
$W \rightarrow A$	45.6	45.9	43.4	43.0	46.2	41.9	42.0	**47.0**
$C \rightarrow A$	45.3	45.5	44.1	42.0	46.1	49.3	**59.8**	48.4
$C \rightarrow W$	60.3	**60.5**	55.9	54.2	57.0	50.4	50.1	54.5
$D \rightarrow W$	62.1	62.1	78.3	78.6	80.2	81.0	**83.3**	79.0
$A \rightarrow W$	**62.4**	61.8	55.7	54.2	56.9	52.3	47.0	53.5
Average	48.5	47.4	49.5	49.7	51.6	51.5	50.9	**52.0**

TABLE IV: Recognition accuracy (%) over 10 object categories on 4DA-CNN with deep feature representation by using different domain adaptation algorithms

Tasks	SourceOnly		Naive Comb		SGF [14]		GFK [13]		SA [9]		LTSL [43]		LSDT [63]		JDA [32]		CRTL	
	f_6	f_7	f_6	f_7	f_6	f_7	f_6	f_7	f_6	f_7	f_6	f_7	f_6	f_7	f_6	f_7	f_6	f_7
$A \rightarrow D$	80.8	81.3	94.5	94.1	90.5	92.0	92.6	94.3	94.2	92.8	95.5	94.5	**96.4**	**96.0**	93.8	94.0	**96.4**	95.8
$C \rightarrow D$	76.6	77.6	92.9	92.8	93.1	92.4	92.0	91.9	93.0	92.1	93.6	93.5	**95.4**	94.6	93.7	93.0	95.2	**94.8**
$W \rightarrow D$	96.1	96.2	99.1	98.9	97.7	97.6	97.8	98.5	98.6	98.5	99.1	98.8	**99.4**	**99.3**	98.9	99.1	**99.4**	**99.3**
$A \rightarrow C$	79.3	79.3	84.0	83.4	77.1	77.4	78.9	79.1	83.1	83.3	85.3	85.4	85.9	**87.0**	84.5	83.3	**86.2**	**87.0**
$W \rightarrow C$	59.5	68.1	81.7	81.2	74.1	76.8	77.5	76.1	81.1	81.0	82.3	82.6	83.1	84.2	82.1	82.6	**83.6**	**84.9**
$D \rightarrow C$	67.3	74.3	83.0	82.7	75.9	78.2	78.8	77.5	82.4	82.9	84.4	84.8	85.2	86.2	84.5	82.8	**85.5**	**86.4**
$D \rightarrow A$	77.0	81.8	90.5	90.9	88.0	88.0	88.9	90.1	90.4	90.7	91.1	91.9	92.2	92.5	91.9	91.7	**92.5**	**92.7**
$W \rightarrow A$	66.8	73.4	90.1	90.6	87.2	86.8	86.2	85.6	89.8	90.9	90.6	91.0	91.0	91.7	**91.3**	90.8	**91.3**	**92.2**
$C \rightarrow A$	85.8	86.5	89.9	90.3	88.5	89.3	87.5	88.4	89.5	89.9	90.4	90.9	**92.1**	**92.5**	91.2	91.0	92.0	**92.5**
$C \rightarrow W$	67.5	67.8	91.6	90.6	89.4	87.8	87.7	86.4	91.2	89.0	91.8	90.8	**93.3**	**93.5**	91.4	90.4	92.7	93.1
$D \rightarrow W$	95.4	95.1	97.9	98.0	96.8	95.7	97.0	96.5	97.5	97.5	98.2	97.8	98.7	98.3	**98.9**	**98.7**	98.7	98.5
$A \rightarrow W$	70.5	71.6	90.4	91.1	87.2	88.1	89.5	88.6	90.3	87.8	92.2	91.5	92.1	92.9	90.8	90.6	**92.3**	**93.0**
Average	76.9	79.4	90.5	90.4	87.1	87.5	87.9	87.8	90.1	89.7	91.2	91.1	92.1	92.4	91.1	90.7	**92.2**	**92.5**

TABLE V: Recognition accuracy (%) of different domain adaptation on COIL-20

Tasks	SVM	TSL	RDALR [21]	LTSL [43]	DTSL [52]	LSDT [63]	JDA [32]	CRTL
$C1 \rightarrow C2$	82.7	80.0	80.7	75.4	84.6	81.7	**89.3**	87.0
$C2 \rightarrow C1$	84.0	75.6	78.8	72.2	84.2	81.5	**88.5**	86.5
Average	83.3	77.8	79.7	73.8	84.4	81.6	**88.9**	86.8

we switch the two datasets: VOC vs. MSRC ($V \rightarrow M$). All images are uniformly re-scaled to 256 pixels, and 128-dimensional dense SIFT (DSIFT) features using the VLFeat open source package are extracted. Then K-means clustering is used to obtain a 240-dimensional codebook.

In experiments, the source training data set contains all the labeled samples in the source domain, the labeled target training data contains 4 labeled examples per class randomly selected from the target domain and the rest unlabeled examples are recognized as the target testing data. The experimental results by using different domain adaptation methods are shown in Table XI, from which we can observe that the proposed method outperforms other state-of-the-art methods.

Results on P-A-C-S dataset[6] **[25]**: The PACS dataset, as described in Fig. 1, is a recently proposed dataset for cross-domain image classification tasks. This dataset is practically relevant, and harder (bigger domain shift) than existing benchmarks. It is developed by intersecting the classes in Caltech256 (Photo), Sketchy (Photo, Sketch) [42], TU-Berlin (Sketch) [6] and Google Images (Art painting, Cartoon, Photo). Eventually, this new benchmark includes 4 domains (Photo, Art painting, Cartoon, Sketch) which can be simplified as P, A, C and S, respectively. Generally, it contains 7 common categories: dog, elephant, giraffe, guitar, horse, house and person. The total number of images is 9991. This benchmark dataset brings two important advancements over the previous ones: (1) it extends the previously photo-only setting in DA community, and uniquely includes domains that are maximally distinct from each other. It spans a wide spectrum of visual abstraction, from photos with the least abstract to human sketches with the most abstract; (2) it better approaches real-world scenario where a

Fig. 6: Some samples from MSRC and VOC2007 datasets

[6]http://sketchx.eecs.qmul.ac.uk

TABLE VI: Recognition accuracy (%) of different domain adaptation on MSRC and VOC2007

Tasks	SVM	MMDT [17]	KMM [19]	GFK [13]	LSDT [63]	JDA [32]	CRTL
$M \rightarrow V$	36.3	36.0	36.1	29.5	36.9	**38.2**	37.3
$V \rightarrow M$	64.3	62.1	**64.8**	50.7	59.3	59.3	**64.8**
Average	50.3	49.1	50.5	40.1	48.1	48.8	**51.1**

target domain (e.g., sketch) is rare, and domain generalization from an abundant domain (e.g., photos) is necessary.

In experiment, the deep features are extracted by using a fine-tuned CNN using the data from multiple domains. Note that the CNN is the pre-trained AlexNet based on ImageNet. By following the same experimental setting as [25], three domains are used as source domain for training and the rest domain is used as target domain for testing. Therefore, 4 groups of experiments are conducted, alternatively. Note that, one sample per class from target domain is randomly chosen as the target training sample. With above settings, the recognition accuracies of 4 groups by using different DA methods have been shown in Table VII. It is obvious that the proposed method outperforms other state-of-the-art DA methods.

C. Cross-poses Face Recognition

Pose alignment is challenging due to the highly non-linear changes induced by 3D rotation of a face. The cross-pose face recognition, as a standard DA problem, is therefore conducted. The CMU Multi-PIE face dataset[7] is a popular dataset consisting of 337 subjects, which contains 4 different sessions with 15 poses, 20 illuminations, and 6 expressions. In our experiment, we select the first 60 subjects from Session 1 and Session 2. As a result, a smaller session 1 ($S1$) of 7 images with different poses per class under neutral expression and a smaller session 2 ($S2$) that is similar to $S1$ but under smile expression are constructed. In this way, two tasks with neutral and smile expression have been formulated. The example images of one subject in $S1$ and $S2$ are illustrated in the 1^{st} and 2^{nd} row of Fig. 7, respectively. Specifically, the experimental configurations are as follows.

$S1$: For the faces in Session 1, one frontal face per subject is used as the source training data, one 60° posed face is used as the target training data, and the rest 5 face images are used as the target test data.

$S2$: The experimental configuration is the same as $S1$, which is conducted on the faces in Session 2.

$S1+S2$: The two frontal faces and the two 60° posed faces under neutral and smile expression are used as source training data and target training data, respectively. The rest 10 face images are used as target test data.

$S1 \rightarrow S2$: The faces per subject in $S1$ under neural expression are used as source training data, the frontal and 60° posed faces in $S2$ are used as the target training data, and the rest data are used as test data.

With above settings, the recognition accuracies of different experimental configurations have been shown in Table VIII. It is obvious that the proposed method performs significantly better over other DA methods in handling such pose change

[7]http://www.cs.cmu.edu/afs/cs/project/PIE/MultiPie/Multi-Pie/Home.html

Fig. 7: Facial images of one person from CMU Multi-PIE

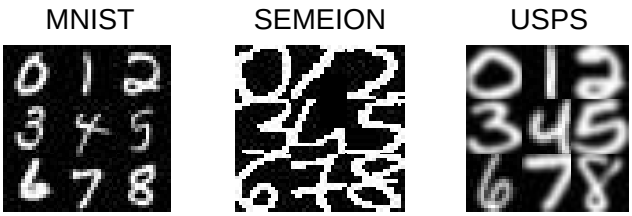

Fig. 8: Some images from handwritten digits datasets

based nonlinear domain transfer problem, which is generally recognized to be a difficult problem in computer vision.

D. Cross-domain Handwritten Digits Recognition

In this experiment, three handwritten digits datasets: MNIST (M)[8], USPS (U)[9] and SEMEION (S)[10], as described in Fig. 8 with 10 classes from digit $0 \sim 9$, have been used for evaluating the proposed CRTL method. The MNIST dataset consists of 70,000 instances with image size of 28×28, the USPS dataset consists of 9298 examples with image size of 16×16, and the SEMEION dataset consists of 2593 images with size of 16×16. In experiments, for dimension consistency, the images in MNIST dataset are cropped into 16×16. For DA setting, by following [63], each dataset is used as the source and target domain alternatively, and 6 cross-domain tasks are obtained. Also, 100 samples per class from source domain and 10 samples per class from target domain are randomly selected for training. To this end, 5 random splits are used, and the average classification accuracies are reported in Table IX. From the results, we observe that our CRTL(81.8%) significantly outperforms other state-of-the-art representation based DA methods, and the superiority is therefore proved.

V. Discussion

A. Parameter Setting and Ablation Analysis

Parameter Setting. In CRTL, Gaussian kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(- \parallel \mathbf{x}_i - \mathbf{x}_j \parallel^2 / 2\sigma^2)$ is considered,

[8]http://yann.lecun.com/exdb/mnist/

[9]http://www-i6.informatik.rwth-aachen.de/~keysers/usps.html

[10]http://archive.ics.uci.edu/ml/datasets/Semeion+Handwritten+Digit

TABLE VII: Recognition accuracy (%) of different domain adaptation models on P-A-C-S dataset

Tasks	SVM	SGF [14]	SA [9]	LSDT [63]	JDA [32]	CRTL
$ACP \rightarrow S$	33.0	31.2	31.8	34.7	30.2	**37.4**
$CPS \rightarrow A$	50.5	41.7	42.5	54.2	39.6	**54.6**
$PSA \rightarrow C$	**55.5**	43.2	45.5	53.6	41.4	53.6
$SAC \rightarrow P$	80.8	70.5	70.0	80.9	64.4	**82.4**
$Average$	54.9	46.6	47.5	55.8	43.9	**57.0**

TABLE VIII: Recognition accuracy (%) of different domain adaptation methods on face recognition across poses

Tasks	Naive Comb	A-SVM	SGF [14]	GFK [13]	SA [9]	LTSL [43]	LSDT [63]	JDA [32]	CRTL
$S1$ ($0° \rightarrow 60°$)	61.0	57.0	53.7	56.0	51.3	56.0	59.7	33.3	**65.7**
$S2$ ($0° \rightarrow 60°$)	62.7	62.7	55.0	58.7	62.7	60.7	63.3	39.0	**69.0**
$S1 + S2$ ($0° \rightarrow 60°$)	60.2	60.1	53.8	56.3	61.7	60.7	61.7	36.5	**68.5**
$S1 \rightarrow S2$	93.6	94.3	92.5	96.7	98.3	96.7	95.8	92.0	**98.7**
$Average$	69.4	68.5	63.8	67.0	68.5	68.5	70.1	50.2	**75.5**

TABLE IX: Recognition accuracy (%) of different domain adaptation methods on handwritten digits recognition

Tasks	Naive Comb	A-SVM	SGF [14]	GFK [13]	SA [9]	LTSL [43]	LSDT [63]	JDA [32]	CRTL
$M \rightarrow U$	78.8	78.3	79.2	82.6	78.8	83.2	79.3	79.8	**85.4**
$S \rightarrow U$	83.6	76.8	77.5	82.7	82.5	83.6	84.7	77.8	**86.2**
$M \rightarrow S$	51.9	70.5	51.6	70.5	74.4	72.8	69.1	62.2	**76.2**
$U \rightarrow S$	65.3	74.5	70.9	76.7	74.6	65.3	67.4	68.4	**82.6**
$U \rightarrow M$	71.7	73.2	71.1	74.9	72.9	71.7	70.5	75.0	**82.0**
$S \rightarrow M$	67.6	69.3	66.9	74.5	72.9	67.6	70.0	73.2	**78.4**
Average	69.8	73.8	69.5	77.0	76.0	74.0	73.5	72.7	**81.8**

where σ is the kernel parameter tuned for different tasks empirically from an appropriate range (0.1,2) in experiments. Specifically, $\sigma = 1.2$ for 4DA-CNN and MSRC-VOC2007 datasets, $\sigma = 0.5$ for COIL-20 dataset, $\sigma = 0.2$ for CMU Multi-PIE dataset and $\sigma = 1.0$ for handwritten digits dataset. The dimension of common subspace is set as $d = N$ for better recognition performance. Note that the least square classifier is used in DA experiments except that in COIL-20 experiment, the SVM classifier is used because of its good performance.

Ablation Analysis. For better insight of the impact of the loss terms including domain reconstruction loss (RECO) term, class representation loss (REPR) term and the pHSIC term, the ablation analysis is provided in 4DA-CNN dataset and CMU Multi-PIE dataset. The ablation analysis results are shown in Table X and Table XI, respectively, in which "w/o RECO" denotes that the reconstruction loss is dropped, "w/o REPR" denotes that the representation loss is dropped, and "w/o pHSIC" denotes that the data-label dependency preservation term is dropped. From Table X and Table XI, we can observe that both the the transfer loss with domain reconstruction and class representation, and the pHSIC item in our method have contributed to reducing the domain discrepancy.

B. Dimensionality and Computational Complexity Analysis

Dimensionality Analysis. In CRTL model, a latent common subspace $\boldsymbol{\mathcal{P}}$ is learned. Therefore, the performance variation with varying subspace dimensions is studied on 4 tasks of 4DA-CNN (i.e., $C \rightarrow D$, $W \rightarrow A$, $C \rightarrow W$, and $D \rightarrow W$) and 3 tasks of PIE face datasets (i.e., $S1$, $S2$, and $S1 + S2$). Specifically, the performance curve with decreasing dimensionality d is shown in Fig. 9 (a) and (b), respectively. Generally, higher dimension leads to better performance.

TABLE X: Ablation analysis on 4DA-CNN dataset

Dataset	CRTL	Transfer Loss Function		w/o pHSIC
		w/o RECO	w/o REPR	
$A \rightarrow D$	95.91	95.91	96.02	95.91
$C \rightarrow D$	94.49	93.94	94.44	94.45
$W \rightarrow D$	99.13	98.66	99.13	99.13
$A \rightarrow C$	87.00	87.03	86.90	87.00
$W \rightarrow C$	84.67	84.56	84.59	84.73
$D \rightarrow C$	86.12	86.19	86.18	86.17
$D \rightarrow A$	92.63	92.45	92.60	92.64
$W \rightarrow A$	91.94	91.91	91.86	91.93
$C \rightarrow A$	92.41	92.45	92.48	92.51
$C \rightarrow W$	92.68	92.42	92.49	92.62
$D \rightarrow W$	98.42	97.28	98.51	98.53
$A \rightarrow W$	93.34	93.34	93.08	93.32
$Average$	92.40	92.18	92.35	92.41

Computational Complexity Analysis. In this section, we present the computational complexity analysis of the **Algorithm 1**. In general, two steps: update $\boldsymbol{\mathcal{Z}}$ and update $\boldsymbol{\Phi}$ are involved. The computation of $\boldsymbol{\Phi}$ involves eigen-decomposition and matrix multiplication, and the complexity is $O(N^3)$. The computation of updating $\boldsymbol{\mathcal{Z}}$ involves updating of $\boldsymbol{\mathcal{J}}$, $\boldsymbol{\mathcal{G}}$ and $\boldsymbol{\mathcal{Z}}$. Thus the complexity of computing $\boldsymbol{\mathcal{Z}}$ is $O(N^2)$. Suppose that the number of iterations is T, then the total computation complexity is $O(TN^3) + O(TN^2)$. Note that the complexity of kernel Gram matrix computation is excluded.

C. Visualization and Convergence

In this section, the visualization of the learned feature distribution and reconstruction matrix as well as the convergence analysis have been discussed.

Visualization of Feature Distribution. For better insight of the CRTL model, the distribution visualization is explored. We

TABLE XI: Ablation analysis on CMU PIE dataset

Dataset	CRTL	Transfer Loss Function		w/o pHSIC
		w/o RECO	w/o REPR	
$S1$ ($0^\circ \rightarrow 60^\circ$)	65.00	65.00	65.33	66.67
$S2$ ($0^\circ \rightarrow 60^\circ$)	69.33	69.33	67.67	67.00
$S1+S2$ ($0^\circ \rightarrow 60^\circ$)	68.33	68.33	67.50	68.50
$S1 \rightarrow S2$	99.33	99.33	99.00	99.33
$Average$	75.62	75.62	74.88	75.38

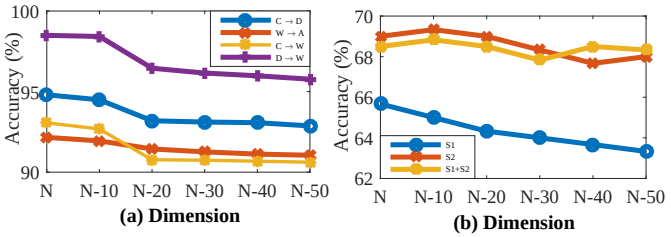

Fig. 9: Performance analysis with decreasing dimension in 4DA-CNN (a) and Multi-PIE (b). Generally, higher dimension leads to better performance.

have shown the visualization of 4DA-CNN feature [3] in $A \rightarrow W$ task by using the t-SNE embedding tool in Fig. 10. We can observe the better domain confusion characteristic after projection from the range of abscissa axis, that is, the two domains become much closer.

Visualization of Reconstruction Matrix. The reconstruction matrix $\boldsymbol{\mathcal{Z}}$ is imposed with class-specific, sparse and low-rank characteristic, such that the domain correlation between the source and target domain data can be improved with block-diagonal property. That is, the target data of class c can be better linearly represented by the source data of the same class. Fig. 11 shows the visualization of the matrix $\boldsymbol{\mathcal{Z}}$ in 4DA-CNN and PIE datasets, and the block-diagonal structure of $\boldsymbol{\mathcal{Z}}$ is observed. Therefore, it is effective to preserve the class-wise characteristic by exploiting the intra-class dependency and inter-class independency, which helps improve the discrimination when domain data is badly corrupted [26].

Convergence. The convergence of CRTL method is studied by conducting the experiments on COIL-20 ($C1 \rightarrow C2$) and PIE (S1+S2), respectively. In experiments, the number of iterations is set to be 100, and the objective function (i.e., F_{min}) as described in Fig. 12 decreases to a constant value after several iterations. Also, the convergence of regularization terms in CRTL, i.e., $\| \boldsymbol{\mathcal{Z}} \|_1$ and $\| \boldsymbol{\mathcal{Z}} \|_*$ are also presented. However, small perturbation still exists which is not strange in non-convex optimization.

VI. Conclusion

In this paper, we propose a class-specific reconstruction transfer learning (CRTL) model, which fully exploits the intra-class dependency and inter-class independency of the reconstruction transfer matrix. For pursuit of a latent subspace where the transfer can be better achieved, we propose a projected HSIC criterion for exploring statistical dependency between features and labels. The merits of CRTL are three-fold. $First$, we cast the transfer learning problem as a fine-grained reconstruction modeling and optimization problem. $Second$, in order to keep the intrinsic statistical dependency between the domain data and labels after feature projection, a Projected Hilbert-Schmidt Independency Criterion (**pHSIC**) in RKHS is explored in CRTL. $Third$, the joint low-rank and sparse constraints are imposed for characterizing global and local structure. Extensive experiments on challenging DA datasets demonstrate the superiority the proposed method over other state-of-the-art methods. This paper brings a new perspective that shallow statistical transfer learning models and deeply learned features extracted with a pre-trained deep network can be accompanied and promoted with each other, rather than only over-depending on deep transfer models trained from scratch or fine-tuned with domain data of interest.

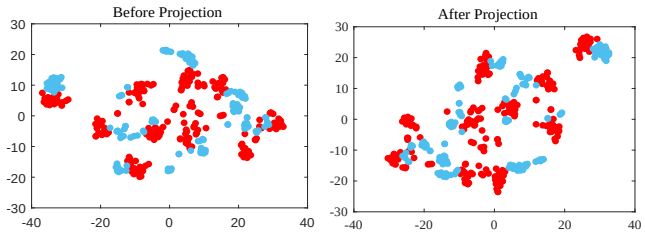

Fig. 10: Visualization of distribution on 4DA-CNN (A→W) task. The red points represents the Amazon feature and the blue points means the Webcam feature. We see the better domain confusion characteristic after projection from the range of abscissa axis. That is, the two domains become closer.

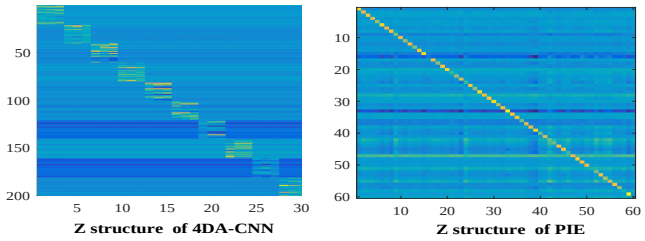

Fig. 11: Visualization of reconstruction matrix $\boldsymbol{\mathcal{Z}}$, which shows a block-diagonal structure. The reason is that the target data of class c can be better linearly represented by the source data of the same class, such that the domain correlation between the source and target domain data can be improved.

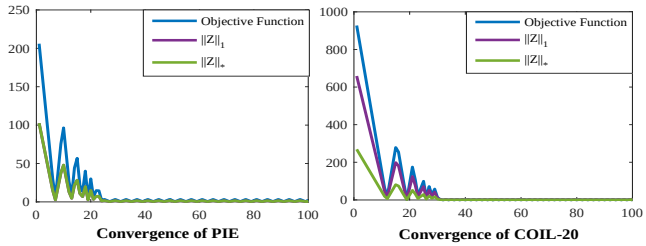

Fig. 12: Convergence analysis of CRTL model. The objective function (i.e., F_{min}) and the regularization terms in CRTL decrease to a constant value after several iterations.

We suggest that for improving real-world application, some seamless bridging between shallow and deep models can be studied in the future work.

REFERENCES

[1] J. F. Cai, E. J. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. *Siam Journal on Optimization*, 20(4):1956–1982, 2008.

[2] Z. Ding, M. Shao, and Y. Fu. Deep low-rank coding for transfer learning. In *IJCAI*, 2015.

[3] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *Computer Science*, 50(1):815–830, 2013.

[4] L. Duan, D. Xu, and I. Tsang. Learning with augmented features for heterogeneous domain adaptation. *arXiv*, 2012.

[5] L. Duan, D. Xu, I. W. Tsang, and J. Luo. Visual event recognition in videos by learning from web data. In *CVPR*, pages 1959–1966, 2010.

[6] M. Eitz, J. Hays, and M. Alexa. How do humans sketch objects? *ACM Trans. Graphics (TOG)*, 31(4):1–10, 2012.

[7] E. Elhamifar and R. Vidal. Sparse subspace clustering: algorithm, theory, and applications. *PAMI*, 35(11):2765–2781, 2012.

[8] J. Feng, Z. Lin, H. Xu, and S. Yan. Robust subspace segmentation with block-diagonal prior. In *CVPR*, pages 3818–3825, 2014.

[9] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, 2013.

[10] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016.

[11] J. Ghosn and Y. Bengio. Bias learning, knowledge sharing. *IEEE Trans. Neural Networks*, 14(4):748–765, 2003.

[12] X. Glorot, A. Bordes, and Y. Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *ICML*, pages 513–520, 2011.

[13] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pages 2066–2073, 2012.

[14] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, 2011.

[15] A. Gretton, K. Borgwardt, M. J. Rasch, B. Scholkopf, and A. J. Smola. A kernel method for the two-sample problem. *NIPS*, abs/0805.2368(2007):513 – 520, 2008.

[16] A. Gretton, O. Bousquet, A. Smola, and B. Scholkopf. Measuring statistical dependence with hilbert-schmidt norms. In *ALT*, volume 16, pages 63–78. Springer, 2005.

[17] J. Hoffman, E. Rodner, J. Donahue, B. Kulis, and K. Saenko. Asymmetric and category invariant feature transformations for domain adaptation. *IJCV*, 109(1):28–41, 2014.

[18] T. Hsu, W. Chen, and C. Hou. Unsupervised domain adaptation with imbalanced cross-domain data. In *ICCV*, pages 4121–4129, 2015.

[19] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Scholkopf. Correcting sample selection bias by unlabeled data. In *NIPS*, 2006.

[20] A. Iyer, J. S. Nath, and S. Sarawagi. Maximum mean discrepancy for class ratio estimation: Convergence bounds and kernel selection. In *ICML*, pages 530–538, 2014.

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

[22] M. Kan, J. Wu, S. Shan, and X. Chen. Domain adaptation for face recognition: Targetize source domain bridged by common subspace. *IJCV*, 109(1):94–109, 2014.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *NIPS*, 25(2):1097–1105, 2012.

[24] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, pages 1785–1792, 2011.

[25] D. Li, Y. Yang, Y.-Z. Song, and T. Hospedales. Deeper, broader and artier domain generalization. In *ICCV*, 2017.

[26] Y. Li, J. Liu, H. Lu, and S. Ma. Learning robust face representation with classwise block-diagonal structure. *IEEE Trans. Information Forensics and Security*, 9(12):2051–2062, 2014.

[27] Z. Lin, R. Liu, and Z. Su. Linearized alternating direction method with adaptive penalty for low-rank representation. *NIPS*, 2011.

[28] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013.

[29] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *ICML*, pages 663–670, 2010.

[30] J. Liu and L. Zhang. Sparse softmax vector coding based deep cascade model. In *CCCV*, pages 603–614, 2017.

[31] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105, 2015.

[32] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu. Transfer feature learning with joint distribution adaptation. In *ICCV*, 2013.

[33] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *NIPS*, 2016.

[34] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, pages 2208–2217, 2017.

[35] C. Y. Lu, H. Min, Z. Q. Zhao, L. Zhu, D. S. Huang, and S. Yan. Robust and efficient subspace segmentation via least squares regression. In *ECCV*, pages 347–360, 2012.

[36] G. Mesnil, Y. Dauphin, X. Glorot, S. Rifai, Y. Bengio, I. Goodfellow, E. Lavoie, X. Muller, G. Desjardins, and D. Warde-Farley. Unsupervised and transfer learning challenge: a deep learning approach. *Workshop on Unsupervised and Transfer Learning*, 7:1–15, 2012.

[37] H. V. Nguyen, H. T. Ho, V. M. Patel, and R. Chellappa. Dash-n: Joint hierarchical domain adaptation and feature learning. *IEEE Trans. Image Processing*, 24(12):5479–5491, 2015.

[38] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, pages 1717–1724, 2014.

[39] C. Rate and C. Retrieval. Columbia object image library (coil-20). *Tech. Rep.*, 2011.

[40] L. Rosasco, A. Verri, M. Santoro, S. Mosci, and S. Villa. Iterative projection methods for structured sparsity regularization. *MIT Tech. Rep.*, 2009.

[41] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, 2010.

[42] P. Sangkloy, N. Burnell, C. Ham, and J. Hays. The sketchy database: learning to retrieve badly drawn bunnies. *Acm Trans. Graphics*, 35(4):1–12, 2016.

[43] M. Shao, D. Kit, and Y. Fu. Generalized transfer subspace learning through low-rank constraint. *IJCV*, 109(1):74–93, 2014.

[44] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa. Generalized domain-adaptive dictionaries. In *CVPR*, pages 361–368, 2013.

[45] B. Sun, J. Feng, and K. Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, volume 6, page 8, 2016.

[46] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. *arXiv preprint arXiv:1607.01719*, 2016.

[47] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *ICCV*, pages 4068–4076, 2015.

[48] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. 2017. CVPR.

[49] S. Wang, L. Zhang, and W. Zuo. Class-specific reconstruction transfer learning via sparse low-rank constraint. In *ICCV*, 2017.

[50] M. Xiao and Y. Guo. Feature space independent semi-supervised domain adaptation via kernel matching. *IEEE Trans. PAMI*, 37(1):54–66, 2015.

[51] M. Xie, N. Jean, M. Burke, D. Lobell, and S. Ermon. Transfer learning from deep features for remote sensing and poverty mapping. *arXiv*, 2015.

[52] Y. Xu, X. Fang, J. Wu, X. Li, and D. Zhang. Discriminative transfer subspace learning via low-rank and sparse representation. *IEEE Trans. Image Process*, 25(2):850–863, 2015.

[53] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal of Machine Learning Research*, 8(1):35–63, 2007.

[54] J. Yang, R. Yan, and A. Hauptmann. Cross-domain video concept detection using adaptive svms. In *ACM MM*, 2007.

[55] M. Yang, L. Zhang, X. Feng, and D. Zhang. Fisher discrimination dictionary learning for sparse representation. In *ICCV*, 2011.

[56] Y. Yao and G. Doretto. Boosting for transfer learning with multiple sources. In *CVPR*, pages 1855–1862, 2010.

[57] H. Zhang, V. M. Patel, S. Shekhar, and R. Chellappa. Domain adaptive sparse representation-based classification. In *AFGR*, pages 1–8, 2015.

[58] L. Zhang. Transfer adaptation learning: A decade survey. *arXiv*, 2019.

[59] L. Zhang, S. Wang, G. B. 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.

[60] L. Zhang, J. Yang, and D. Zhang. Domain class consistency based transfer learning for image classification across domains. *Information Sciences*, 418-419:242–257, 2017.

[61] L. Zhang and D. Zhang. Domain adaptation extreme learning machines for drift compensation in e-nose systems. *IEEE Trans. Instrumentation and Measurement*, 64(7):1790–1801, 2015.

[62] L. Zhang and D. Zhang. Robust visual knowledge transfer via extreme learning machine based domain adaptation. *IEEE Trans. Image Process*, 25(10):1–1, 2016.

[63] L. Zhang, W. Zuo, and D. Zhang. Lsdt: Latent sparse domain transfer learning for visual adaptation. *IEEE Trans. Image Process*, 25(3):1177–1191, 2016.