Cross-Domain Object Recognition via Input-Output Kernel Analysis

Zhenyu Guo and Z. Jane Wang, *Senior Member, IEEE,*

***Abstract*—It is of great importance to investigate the domain adaptation problem of image object recognition since now image data is available from a variety of source domains. To understand the changes in data distributions across domains, we study both the input and output kernel spaces for cross-domain learning situations, where most of the labeled training images are from a source domain and the testing images are from a different target domain. To address the feature distribution change issue in the Reproducing Kernel Hilbert Space induced by vector-valued functions, we propose a Domain Adaptive Input-Output Kernel Learning (DA-IOKL) algorithm, which simultaneously learns both the input and output kernels with a discriminative vector-valued decision function by reducing the data mismatch and minimizing the structural error. We also extend the proposed method to the cases of having multiple source domains. We demonstrate the ability of the proposed model to adapt across domains by examining two cross-domain object recognition benchmark data sets. The proposed method consistently outperforms the the state-of-art domain adaptation and multiple kernel learning methods.**

***Index Terms*—Domain adaptation, Object Recognition, Output Kernel, Multiple Kernel Learning**

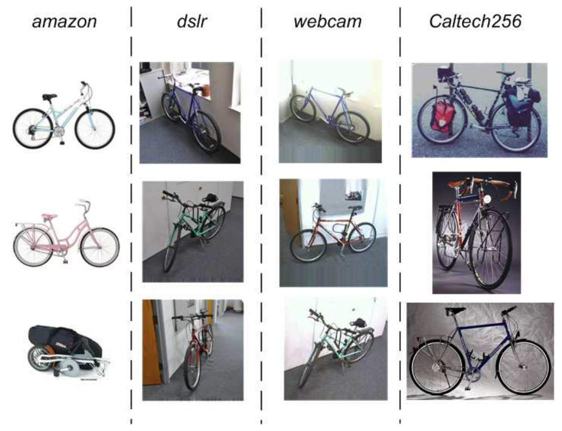


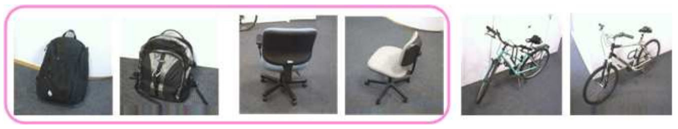
Fig. 1: Example images of the same ‘bike’ category from four different domains: *amazon*, *dslr*, *webcam* and Caltech256.

I. Introduction

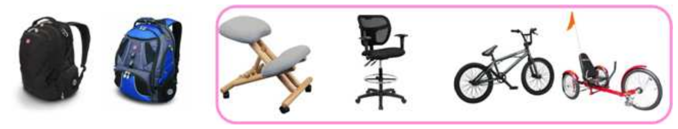
In traditional visual object recognition systems, a model is always trained based on data from the same domain as the testing data, where the implicit assumption is that the training and testing distributions are the same. However, we often face the situations that additional data (with or without labels) from other similar but different domains can be available for training. How to train a model that can take benefits from the extra data in similar source domain(s) by overcoming the side-effects introduced by domain shift is called *domain adaptation* (DA), a research topic drawing increasing research attentions in computer vision and machine learning fields recently. In a domain adaptation problem, a source domain is different from, but somewhat similar to the target domain. One example in the text classification area is that the email spam data sets [1] collected from different individual users form different domains, characterizing individual users' preferences. One example in visual object recognition is given in [2], where the Caltech-256 data set is considered as one domain of object images, and the other domain is the collection of the results from the image search engine Bing when using the category names from Caltech-256 as search queries. Label quality is a key characteristic of different domains. Although the above two domains both contain images for the same object categories, the images in Caltech-256 are manually labeled (i.e., noiseless in label information), while on the other hand, the labels for Bing images are noisy and unreliable. As another example, a specific data set for cross-domain object recognition is adopted in [3]–[5], where the images are acquired by *dslr*, *webcam* and from the *amazon* website. Figure 1 shows some example images for the ‘bike’ category from the above 3 domains, along with images from the Caltech256 domain. As we can see, though the 4 domains are somewhat similar (i.e., representing the same object categories), there are differences across these four domains in terms of pose, lighting, resolution, camera peculiarities and other factors. In reality, the available data from target domain usually comes with no labels or with very limited label information. In the literature, the problem involving with available training data from target domain with no labels is referred as *unsupervised domain adaptation*; the problem involving with target training data with a small number of labels is called *semi-supervised domain adaptation*. In the area of computer vision and multimedia, the *semi-supervised domain adaptation* is attracting increasing attentions, and it is also the focus of this paper. In this paper, we tackle the domain adaptation problem of image object recognition that aims at efficiently leveraging the labeled images from different but related source and target domains to derive better hypothesis testing in the target domain.

Though it is intuitive to believe that extra labeled data from other source domains can potentially increase the recognition accuracy in the target domain, it is not trivial to make good use

Zhenyu Guo and Z.Jane Wang are with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada. E-mail: {zhenyug,zjanew}@ece.ubc.ca.



(a) Sample images from domain *webcam*.
$sim(desk\_chair, back\_pack) > sim(desk\_chair, bike)$.



(b) Sample images from domain *amazon*.
$sim(desk\_chair, back\_pack) < sim(desk\_chair, bike)$.

Fig. 2: From left to right in each row are images from categories: *back_pack, desk_chair, bike*. The proposed output kernel space analysis shows that the categories *desk_chair* and *back_pack* are more similar than *desk_chair* and *bike* in the *webcam* domain, while the opposite is observed in the *amazon* domain. These observations are consistent with the visual inspection results.

of the extra labeled data because of the distribution differences in the feature spaces across domains. Experimental results in [3], [6], [7] show that standard classifiers directly trained on the combination of the source and target domains perform poorly on the test data in the target domain, when compared with the classifiers trained on a large number of labeled data from the target domain. Let $P_{\mathcal{A}}(\mathbf{x}, y)$ and $P_{\mathcal{B}}(\mathbf{x}, y)$ denote joint distributions of feature-label data from source domain and target domain respectively, where $\mathbf{x}$ is the feature vector and label y is the label. Semi-supervised domain adaptation is to use a small number of training samples from $P_{\mathcal{B}}(\mathbf{x}, y)$ and many from $P_{\mathcal{A}}(\mathbf{x}, y)$ to build a learning model for classification. The shift from $P_{\mathcal{A}}$ to $P_{\mathcal{B}}$ causes troubles when training a standard classifier. We denote the data set from source domain as $S_{\mathcal{A}} = \{X_{\mathcal{A}}, Y_{\mathcal{A}}\}$ and data set from target domain as $S_{\mathcal{B}} = \{X_{\mathcal{B}}, Y_{\mathcal{B}}\}$. In detail, the feature data points from two domains can be described as $\{\mathbf{x}_1^{\mathcal{A}}, \ldots, \mathbf{x}_{n_A}^{\mathcal{A}}\}$, and $\{\mathbf{x}_1^{\mathcal{B}}, \ldots, \mathbf{x}_{n_B}^{\mathcal{B}}\}$.

In order to train a domain adaptive classifier by efficiently leveraging the information from different domains, different DA techniques have been proposed to learn a function $\hat{y} = f(\mathbf{x}_{test}^{\mathcal{B}} | S_{\mathcal{A}}, S_{\mathcal{B}})$ that predicts the class label of an unseen testing sample $\mathbf{x}_{test}$ from target domain with high probability, $p_{\mathcal{B}}(Y = \hat{y} | X = \mathbf{x}_{test}^{\mathcal{B}})$, by forcing $p_{\mathcal{A}}(\mathbf{x}, y)$ and $p_{\mathcal{B}}(\mathbf{x}, y)$ to be close. By definition, $p(\mathbf{x}, y) = p(y|\mathbf{x})p(\mathbf{x})$, therefore the mismatch between distributions can be handled in terms of $p(y|\mathbf{x})$ or $p(\mathbf{x})$ under different assumptions [8]. Among those, one trend is to directly or indirectly reduce the mismatch of data distributions across domains by projection [5], [9]–[12], kernel analysis [13]–[15], or metric learning [3], [4]. Another trend is to combine the decision functions of different classifiers trained on the data from different domains to obtain a more powerful augmented classifier for testing data in the target domain [2], [6], [15].

In the previous work, label data y is usually treated as a scalar, and multi-class classification is achieved based on binary classification by adopting one-versus-all or one-versus-one principle. The use of scalar functions as classifiers works well in the traditional classification tasks. However, in practice, we observe that the distribution shift from $P_{\mathcal{A}}$ to $P_{\mathcal{B}}$ actually results in changes in between-category similarity. For instance, Figure 2 shows that the categories *desk_chair* and *back_pack* appear more similar in the *webcam* domain, while *desk_chair* and *bike* appear more similar in the *amazon* domain. Such between-category similarity information plays an important role in multi-class classification in domain adaptation, and it also defines the structure of the intrinsic manifold where the multi-class data points embed into. Correspondingly, a scalar class label cannot capture the shift from $P_{\mathcal{B}}(y|\mathbf{x})$ to $P_{\mathcal{A}}(y|\mathbf{x})$ correctly. Therefore, we propose modeling the class label as a vector $\mathbf{y}$, using the binary coding scheme(1 stands for presence and 0 stands for the absence of a class instance). Recalling the theories of functional analysis on vector-valued functions [16], [17], we can consider a multi-class classifier as a vector-valued function with the structured output which induces a vector-valued Reproducing Kernel Hilbert Space (RKHS). In this case, the **Input** kernel space (the scalar kernel space on the input features) of this RKHS is related to the mismatch of data distributions $P_{\mathcal{B}}(\mathbf{x})$ and $P_{\mathcal{A}}(\mathbf{x})$, and the **Output** kernel space (the matrix kernel space on the structured output of the function) actually corresponds to between-category similarities, which also contributes to a better estimation of $P(\mathbf{y}|\mathbf{x})$. The input and output kernels together form a more comprehensive kernel space for vector-valued decision functions than the kernel space used in [13]–[15] for scalar functions. It is also more comprehensive and richer than the intermediate spaces introduced by linear projections [5], [9]–[12] and metric learning [3], [4].

Following the above inspiration, in this paper, we will investigate both the input and output kernel spaces to overcome the distribution mismatch issue caused by domain shift. More specifically, we propose a Domain Adaptive Input-Output Kernel Learning (DA-IOKL) algorithm for cross-domain image object recognition, i.e., learning an separable RKHS for vector-valued functions (consisting of input kernel and output kernel) where the mismatch between the data distributions could be reduced. The contributions of this paper can be summarized as follows:

1) For the first time, we introduce the analysis of the output kernel space induced by a vector-valued decision function into the domain adaptation problem;
2) We propose a particular objective function in the DA-IOKL model to learn the optimal input and output kernels jointly. We adopt Maximum Mean Discrepancy (MMD) [18] as a regularizer to minimize the structural classification error and the mismatch between data domains, and to avoid large computational cost and optimization difficulty, we propose a multiple kernel form to parameterize the input kernel. In addition, DA-IOKL also provides a vector-valued function as a true multi-class classifier;

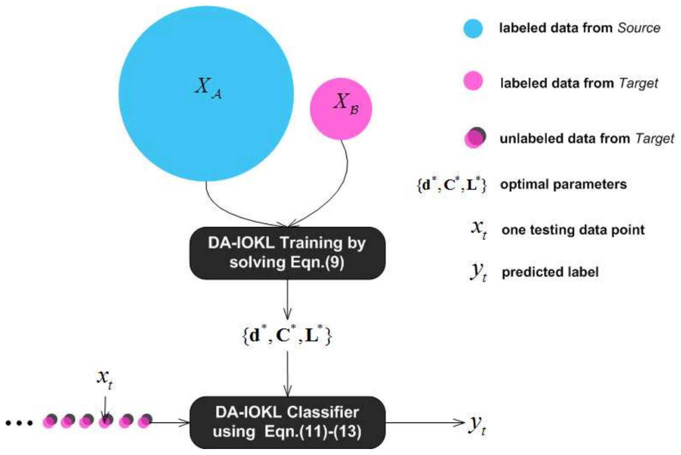

$X_{\mathcal{A}}$
$X_{\mathcal{B}}$
labeled data from Source
labeled data from Target
unlabeled data from Target
$\{\mathbf{d}^*, \mathbf{C}^*, \mathbf{L}^*\}$ optimal parameters
x_t one testing data point
y_t predicted label
DA-IOKL Training by solving Eqn.(9)
$\{\mathbf{d}^*, \mathbf{C}^*, \mathbf{L}^*\}$
x_t
DA-IOKL Classifier using Eqn.(11)-(13)
y_t

Fig. 3: The proposed DA-IOKL framework for cross-domain object recognition. Each circle represents a set of data, and its size indicates the number of data points.

3) We present an efficient algorithm to solve the DA-IOKL optimization problem. Although the objective function of DA-IOKL is not jointly convex w.r.t. all parameters, since it is invex [17] over the output kernel and convex over the input kernel, it can converge to the global minimum. By adopting box constraints, we apply efficient off-the-shelf optimization approaches such as L-BFGS-B [19] to compute the DA-IOKL solution.

We illustrate the cross-domain object recognition process in Figure 3. We can see that a relatively smaller number of labeled training data from the target domain is available during training, compared with the number of labeled training data from the source domain. And a classifier learned jointly based on the data from both source and target domains is used to classify the unlabeled testing data in the target domain. For the details about DA-IOKL and the learned multi-class classifier, please refer to Section III. In addition, a list of major abbreviations used in this paper is given below:

DA: Domain Adaption
DA-IOKL: Domain Adaptive Input & Output Kernel Learning
RKHS: Reproducing Kernel Hilbert Space
MMD: Maximum Mean Discrepancy
MKL: Multiple Kernel Learning

The rest of the paper is organized as follows. Section II summarizes previous works in cross-domain learning and general multiple kernel learning. Section III presents the proposed DA-IOKL algorithm. In Section IV, experiments for cross-domain classification are conducted on two data sets: the object domain adaptation (*DA*) data set [3] and Caltech256+domain adaptation (*cal+DA*) data set [12]. Finally, we conclude the paper in Section V.

II. Related Work

The mismatch problem of data distributions was first investigated in the Natural Language Processing (NLP) community, where intensive researches have been conducted to handle the domain adaptation problem. To capture the shift in feature spaces, $\mathcal{A}$-distance is introduced in [7], [9], [20] for Structural Correspondence Learning (SCL), which presents an approximate estimation of the total variance distance between two distributions. Although this method could measure the shift in the feature spaces, it is hard to estimate and it is not clear how to extend it to computer vision adaptation tasks. Other approaches used in vision researches such as [14], [15], [21] employ a domain similarity measure based on MMD [18], which is non-parametric, easy to estimate empirically and also flexible for the choice of kernel functions. Due to its good performance [18] and its compatibility with kernel methods, we adopt MMD as a penalty term over the input kernel space in this paper.

There are **two major categories** of domain adaptation methods based on various domain shift criteria: 1) reducing the mismatch of data distributions in the feature spaces by projections (or equivalently kernel functions or metrics); 2) combining decision functions trained from different domains. In the **first category**, following the idea of reducing the mismatch of data distributions, the approaches proposed in [7], [9], [20], [21] aim at finding a feature space which can minimize the divergence of distributions between domains based on a specific measure. In addition, [9] provides a theoretical analysis on the feature representation function and classification error. Besides these methods, within the scope of data distributions in feature spaces, several intuitive methods are proposed for recognition purposes. [10] presents a feature augmented way to construct a common feature space. Instead of looking for a subspace by projection, Saenko et al. [3] and Kulis et al. [4] propose learning a metric that can minimize the distance between *similar* data pairs and maximize the distance between *dissimilar* data pairs across two domains, by applying a regularized metric learning model [22]. This approach needs to solve a Semi-Definite Programming (SDP) [23] problem during learning subject to a large number of linear constraints, and thus it is computationally expensive and hard to scale up to high dimensional data. It is also limited to two-domain adaptation scenarios and requires data correspondences between two domains for better performance. [5] adopts a general subspace approach to learn intermediate feature spaces by sampling points along the geodesic of a Grassmann manifold formed by two different domains. More recently, [12] introduces a geodesic flow kernel to extend the idea from [5] and also proposes a way to automatically determine the optimal dimensionality of the subspace. Similarly, [11] proposes finding an intermediate space where the data points from the source domain could be well reconstructed by the data points from the target domain. The intermediate space is obtained by searching for an optimal linear operator and removing noise and outliers simultaneously. In particularly, among the above methods, the methods in [3]–[5], [10]–[12] are investigated for cross-domain image object recognition tasks. It is worth noting that most methods [5], [10]–[12] in the first category can be used in both an unsupervised way and semi-supervised way, since it is not necessary to have labels for certain subspace methods. The small number of labels from the target domains could further improve the performance on the unsupervised subspace learning. In this paper, we include these methods [5], [11], [12] for comparison under the semi-supervised learning setting for our domain adaptation problem.

In the **second category**, several methods are proposed, with focus on the decision functions of the classifiers. [6] employs the adaptive SVM to adapt the decision function f^S trained on the source domain into the target SVM classifier f^T by formulating $f^T = f^S + \triangle f$, where $\triangle f$ is trained based on data from the target domain. Transductive SVM [2], [24], Domain Adaptive SVM and cross-domain SVM [25], and other variants of SVMs are also explored in domain adaptation problems by defining a new decision function incorporating data from both the target and source domains.

Since the first category of methods could yield better performances in cross-domain object recognition, we decide to follow the idea of searching for one or several subspaces to reduce the mismatch of data distributions. Recall that the feature spaces or kernel spaces studied in the previous methods [3]–[5], [10]–[12] are the spaces for scalar functions, which can be considered as the **Input** spaces for the vector-valued functions. Therefore, those methods don't consider the **Output** space which is highly related to multi-class classification. Instead of dealing only with data in the input space, we investigate the domain shift in the RKHS for vector-valued functions, which contains both **Input** and **Output** kernel spaces and is more comprehensive than the input space only. To reduce the data mismatch in the RKHS of vector-valued functions, as stated in Section I, we propose an Input-Output kernel learning algorithm, DA-IOKL, to jointly learn an input and output kernel space for the pooled data from source and target domains. Equivalently it can be considered as searching for a better input-output kernel space where the domain shift is minimized. The proposed DA-IOKL also provides a vector-valued function as the optimal multi-class classifier.

The proposed DA-IOKL contains the learning process for an input kernel matrix and an output kernel matrix. The dimension of the output kernel matrix is usually small, since it only depends on the number of categories. But the dimension of the input kernel matrix is usually large and grows with the number of data points used in training and testing. Therefore, we propose learning the output kernel matrix directly with a non-parametric formulation [17] and learning the input kernel matrix with a parametric form similar to Multiple Kernel Learning (MKL) [26]–[28]. We plan to learn a convex combination of kernel bases as the optimal input kernel function, instead of learning the kernel matrix directly as in [29].

III. Proposed Method

In this section we will present the proposed DA-IOKL algorithm. Different from previous works that model the class label as scalar and use a scalar function as the predictive function, the proposed algorithm represents the first attempt to investigate domain adaptation with the analysis of the RKHS for vector-valued functions, inspired by [16], [17], [30]. With the assumption of a multivariate distribution on class labels, we propose using MMD [18] to constrain the mismatch in the marginal distributions. Following a geometric intuition described in [31], we assume the conditional probability distributions should be similar if the marginal distributions are shifted close to each other. Therefore, we propose learning a vector-valued function in the RKHS that can give a best classification performance on the target domain data, where we put an MMD regularization on the parametric input kernel estimation and adopt output kernel estimation approach presented in [17]. Since the output kernel estimation is based on the specific input kernel function, and the input kernel function is constrained by MMD regularization, we first fix the input kernel function and present the topics related to output kernel learning in Section III-A, Section III-C and Section III-D. Section III-B gives a detailed analysis on the advantage of choosing vector-valued function and how this choice could benefit domain adaptation. Section III-E present the whole DA-IOKL that learns the input and output kernel together in an alternating optimizing way. At last, Section III-F proposes a new domain similarity measure based on the output kernel matrix.

A. Background on RKHS of Vector-valued Functions

Let $\mathcal{Y}$ be a real Hilbert space with inner product $(\cdot,\cdot)_{\mathcal{Y}}$, $\mathcal{X}$ a set, and $\mathcal{H}$ is a linear space of functions on $\mathcal{X}$ with values in $\mathcal{Y}$. We assume $\mathcal{H}$ is also a Hilbert space with inner product$\langle\cdot,\cdot\rangle$. Apparently, if $\mathcal{Y} = \mathbb{R}^m$, $\mathcal{H}$ is the space of vector-valued functions. We call a function $g \in \mathcal{H}$ a $\mathcal{Y}$-valued function, and we denote the kernel associated with the RKHS of g as a $\mathcal{Y}$-kernel. We give the definitions for RKHS of $\mathcal{Y}$-valued functions and $\mathcal{Y}$-kernel as follows.

Definition 3.1: (RKHS of $\mathcal{Y}$-valued functions). A RKHS of a $\mathcal{Y}$-valued function $g : \mathcal{X} \to \mathcal{Y}$ is a Hilbert space $\mathcal{H}$ such that, for all $\mathbf{x} \in \mathcal{X}$ there exists $C_x \in \mathbb{R}$,

$$||g(\mathbf{x})||_{\mathcal{Y}} \leq C_x ||g||_{\mathcal{H}}, \forall g \in \mathcal{H}.$$

Definition 3.2: (Positive semidefinite $\mathcal{Y}$-kernel). We say that $\mathbf{H} : \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ is a positive semidefinite $\mathcal{Y}$-kernel if it satisfies the following property for any finite integer l:

$$\sum_{i=1}^{l}\sum_{j=1}^{l} (\mathbf{y}_i, \mathbf{H}(\mathbf{x}_i,\mathbf{x}_j)\mathbf{y}_j)_{\mathcal{Y}} \geq 0, \forall (\mathbf{x}_i,\mathbf{y}_i) \in (\mathcal{X},\mathcal{Y}).$$

In [17], it states that a *unique* positive semidefinite $\mathcal{Y}$-kernel H is associated with a given RKHS of a $\mathcal{Y}$-valued function from $\mathcal{H}$, which is defined over data set $\mathcal{X}$. We assume $\mathcal{Y} = \mathbb{R}^m$, which is the output space in our object recognition problem with m categories. Therefore $\mathcal{L}(\mathcal{Y})$ is the space of m ordered square matrices. Given a basis $\{b_i\}_{i\in\mathcal{T}}$ with $\mathcal{T} = \{1,...,m\}$, a kernel R over $\mathcal{X} \times \mathcal{T}$ can be defined as

$$(b_i, \mathbf{H}(\mathbf{x}_1,\mathbf{x}_2)b_j)_{\mathcal{Y}} = R((\mathbf{x}_1,i),(\mathbf{x}_2,j)).$$

Similarly, given a $\mathcal{Y}$-valued function $g : \mathcal{X} \to \mathcal{Y}$, we can uniquely define a function $h : \mathcal{X} \times \mathcal{T} \to \mathbb{R}$ such that

$$g(\mathbf{x}) = \sum_{i\in\mathcal{T}} h(\mathbf{x},i) b_i.$$

More details on RKHSs of vector-valued functions could be found in [16], [17], [30].

1) Output Kernel : From Definition 3.2, we know that a $\mathcal{Y}$-kernel $\mathbf{H}$ is a function defined on $\mathcal{X} \times \mathcal{X}$. For $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$, the value for $\mathbf{H}(\mathbf{x}_i, \mathbf{x}_j)$ is a linear operator in $\mathcal{L}(\mathcal{Y})$, which is a square matrix. Recall the theory for RKHS of scalar functions, a kernel K for a scalar function is defined on $\mathcal{X} \times \mathcal{X}$ and its value $K(\mathbf{x}_i, \mathbf{x}_j)$ is a scalar, where $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$. In multi-class classification tasks, we denote $\mathcal{Y} = \mathbb{R}^m$ as the output space of m categories. If a data point $\mathbf{x}_j$ belongs to category i, its label is denoted by $\mathbf{y}_j \in \mathbb{R}^m$, which has $+1$ on the the ith element and 0 on others. Let $\mathcal{H}$ be the RKHS of the function $g : \mathcal{X} \to \mathcal{Y}$ associated with the $\mathcal{Y}$-kernel $\mathbf{H}$. $\mathbf{H}$ could be decomposed as [17]:

$$\mathbf{H}(\mathbf{x}_i, \mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j) \cdot \mathbf{L}, \forall \mathbf{x}_i, \mathbf{x}_j \in \mathcal{X} \tag{1}$$

where $\mathbf{L}$ is a symmetric positive semidefinite matrix that measures the relationships between output components (the category similarity information in the multi-class recognition case) of function $g(\mathbf{x})$. $\mathbf{L}$ is called the **Output Kernel**; the kernel K is the scalar part of $\mathbf{H}$, and it measures the similarity between data points in the input space of the function $g(\mathbf{x})$, so the kernel K is called the **Input Kernel**. Given the function $g(\mathbf{x})$, the predicted category label of a testing data $\mathbf{x}$ could be determined by $y_{pre} = \arg\max_{s \in \mathcal{T}} g^{(s)}(\mathbf{x})$, where $\mathcal{T} = \{1, ..., m\}$ and $g^{(s)}(\mathbf{x})$ is the s-th element of the vector-valued function $g(\mathbf{x})$. We can see that $g(\mathbf{x})$ is a also a true multi-class classification decision function.

B. Vector-valued Functions for Domain Adaptation

In this section, we will explain why it is important to use vector-valued functions and how the input and output kernels help domain adaptation.

1) Why Vector-Valued Functions?: Since the data distribution can be decomposed as $p(\mathbf{x}, \mathbf{y}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$, we focus on the conditional probability to illustrate the advantages of using a vector-valued function over a scalar function (e.g. used in previous methods [5], [10]–[12]) in domain adaptation. For the convenience of expression, we suppose the predictive function is a draw from a Gaussian Process. Accordingly, for a scalar predictive function, the corresponding $p(y|\mathbf{x})$ is a univariate Gaussian distribution characterized by a mean and a variance. When extending the scalar predictive function from binary classification to multi-class classification (where a vector $\mathbf{y}$ is used for the class label) by adopting the one-against-one or one-against all rule, it is equivalent to model the conditional probability $p(\mathbf{y}|\mathbf{x})$ with a covariance matrix whose diagonal entries are the variances of $p(y|\mathbf{x})$ for different classes and off-diagonal entries are zeros. While for a vector-valued predictive function, the resulting $p(\mathbf{y}|\mathbf{x})$ is a general Gaussian Vector distribution characterized by a mean vector and a covariance matrix. The covariances between components of $\mathbf{y}$ reflect the similarities between object categories, which can be distinct across domains as discussed in Section I, shown in Figure 2.

Let Gaussian distributions p_A and p_B denote the conditional probability distributions of domain $\mathcal{A}$ and domain $\mathcal{B}$. Suppose we have a way to project the data into a new space where the two conditional distributions could be matched. For the scalar y and the scalar predictive function, only the mean vectors and variances of p_A and p_B could be matched (what most of the previous methods try to do) and there is still a large portion of mismatch between the two distributions due to the differences in covariances. However, for the vector $\mathbf{y}$ and the vector-valued predictive function, the mean vectors and the covariance matrices (both variances and covariances) could be matched, which leads to a better match between p_A and p_B. Therefore, to fully estimate the data distribution, we propose using a vector $\mathbf{y}$ to model the class label and using a vector-valued function $\mathbf{f}$ to model the predictive function.

2) How to Use Vector-valued Functions for Domain Adaptation?: Recalling that data samples are drawn from the distribution $p(\mathbf{x}, \mathbf{x}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x})$, we can tackle the distribution shift issue from the marginal and conditional distributions separately. According to [32], from the geometric perspective, the connection between $p(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{x})$ could be assumed as follows: If two points $\mathbf{x}_1, \mathbf{x}_2 \in X$ are *close* in the intrinsic geometry of $p(X)$, then the conditional distributions $p(\mathbf{y}|\mathbf{x}_1)$ and $p(\mathbf{y}|\mathbf{x}_2)$ are similar. Basically, the conditional probability $p(\mathbf{y}|\mathbf{x})$ varies smoothly along the geodesics in the intrinsic geometry of $p(X)$.

In Section III-A1, we decompose the kernel for a vector-valued function as $\mathbf{H} = K \cdot \mathbf{L}$, where K is the input kernel. Therefore, we propose minimizing MMD [18] of $p_{\mathcal{A}}(\mathbf{x})$ and $p_{\mathcal{B}}(\mathbf{x})$, the marginal distributions for two domains, to reduce distribution shift between the two marginal distributions, by learning a proper input kernel function K in the RKHS. The details of MMD regularization can be found in SectionIII-E1. According to the above geometric assumption, with making the marginal distributions of both domains similar, the conditional distributions for the two domains, $p_{\mathcal{A}}(\mathbf{y}|\mathbf{x})$ and $p_{\mathcal{B}}(\mathbf{y}|\mathbf{x})$, should be similar. Since we use a vector $\mathbf{y}$ for the class label and a vector-valued function $\mathbf{f}$ for the predictive function, the covariance matrix of $\mathbf{y}$ is already taken into consideration. For the convenience of expression, we first fix the input kernel function K for the estimation of the output kernel $\mathbf{L}$ in Section III-C and Section III-D. Then in Section III-E, we incorporate a parametric form of K with a MMD regularizer that can learn an optimal K to reduce the mismatch between $p_{\mathcal{A}}(\mathbf{x})$ and $p_{\mathcal{B}}(\mathbf{x})$.

C. Output Kernel Learning

Here we describe the algorithm proposed in [17] to learn an output kernel from input data, which involves the learning of a $\mathcal{Y}$-valued function $g : \mathcal{X} \to \mathcal{Y}$ and an output kernel matrix $\mathbf{L}$. The basic assumption here is that an output kernel matrix describes the data structure best if the associated function could achieve the minimum classification error on the training data. Let $\mathbb{S}_+$ denote the positive semidefinite matrix space. The objective function for obtaining the proper output kernel could be written as

$$\min_{\mathbf{L} \in \mathbb{S}_+^m} \left[\min_{g \in \mathcal{H}} \left(\sum_{i=1}^{l} \frac{||g(\mathbf{x}_i) - \mathbf{y}_i||_2^2}{2\lambda} + \frac{||g||_{\mathcal{H}}^2}{2} + \frac{||\mathbf{L}||_F^2}{2} \right) \right], \tag{2}$$

where $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{X} \times \mathcal{Y}$ are data-label pairs in multi-class classification. According to the representer theorem [16], the

Algorithm 1 Output Kernel Learning
```
1: $\mathbf{L}, \mathbf{C}, \mathbf{E}, \mathbf{Z} \leftarrow 0$
2: while $||\mathbf{Z} + \lambda\mathbf{C} - \mathbf{Y}||_F \geq \delta$ do
3:     $\mathbf{C} \leftarrow$ Solution to $\mathbf{KCL} + \lambda\mathbf{C} = \mathbf{Y}$,
4:     $\mathbf{E} \leftarrow \mathbf{KC}$
5:     $\mathbf{P} \leftarrow \frac{1}{2}\mathbf{E}^T\mathbf{C} - \mathbf{L}$
6:     $\mathbf{Q} \leftarrow$ Solution to $(\mathbf{E^T E} + \lambda\mathbf{I})\mathbf{Q} = \mathbf{P}$
7:     $\mathbf{L} \leftarrow \mathbf{L} + \lambda\mathbf{Q}$
8:     $\mathbf{Z} \leftarrow \mathbf{EL}$
9: end while
```

TABLE I: An algorithm for learning a vector-valued function and an output kernel simultaneously with block coordinate descent. [17]

optimal solution for the inner minimization has the form

$$g^*(\mathbf{x}) = \sum_{i=1}^{l} H(\mathbf{x}, \mathbf{x}_i)c_i = \mathbf{L}\sum_{i=1}^{l} c_i K(\mathbf{x}, \mathbf{x}_i), \tag{3}$$

where K is the input kernel function. By setting the (i, j) entry of $\mathbf{K} \in \mathbb{S}_+^l$ as $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$, and $\mathbf{Y}, \mathbf{C} \in \mathbb{R}^{l\times m}$ as

$$\mathbf{Y} = (y_1, \ldots, y_l)^T, \mathbf{C} = (c_1, \ldots, c_l)^T, \tag{4}$$

we can see that the objective function in Eqn. (2) becomes

$$Q(\mathbf{L}, \mathbf{C}) := \frac{||\mathbf{Y} - \mathbf{KCL}||_F^2}{2\lambda} + \frac{\langle \mathbf{C^T KC}, \mathbf{L}\rangle_F}{2} + \frac{||\mathbf{L}||_F^2}{2}. \tag{5}$$

Dinuzzo et al. shows in [17] that $Q(\cdot, \cdot)$ is an invex function over the open set $\mathbb{S}_+^m \times \mathbb{R}^{l\times m}$ and proposes an efficient block-wise coordinate descent optimization algorithm, which is described in Table I.

D. Domain Weighted Output Kernel Learning

In domain adaptation, one straightforward but effective method is domain weighting [10], [33], [34], where the decision functions or loss functions corresponding to individual domains are weighted according to their "contributions" to the task in the target domain. Here we adopt the formulation of convex combination of decision functions [34] to address different importance of training data from different domains. The weight parameters can be estimated by cross-validation or empirical studies.

For data set $X_\mathcal{A}$ from the source domain $\mathcal{A}$ and data set $X_\mathcal{B}$ from the target domain $\mathcal{B}$, the joint data set is $X_{\mathcal{AB}} = [X_\mathcal{A}, X_\mathcal{B}]$, which results in an output kernel $L_{\mathcal{AB}}$ that captures category relationship of the joint set $X_{\mathcal{AB}}$. Also, the associated function $g_{\mathcal{AB}}$ should minimize the classification errors of training data from both domains. Now the loss function in Eqn. (2) could be split as

$$\sum_{x_i\in\mathcal{A}} \alpha \cdot \frac{||g(\mathbf{x}_i) - \mathbf{y}_i||_2^2}{2\lambda} + \sum_{x_j\in\mathcal{B}} \frac{||g(\mathbf{x}_j) - \mathbf{y}_j||_2^2}{2\lambda}, \tag{6}$$

where the first item represents the total training error for the source domain training data, and the second item represents the total training error for the target domain training data. In domain adaptation recognition, the goal is to achieve better testing performance in the target domain, therefore we propose a weighted loss to emphasize the importance of target training error during optimization. By introducing a weight $\alpha \in [0, 1]$ into the convex combination of the loss function, we could weight the importance of training data from the source domain. Let $\mathbf{D}$ be a diagonal matrix whose diagonal elements are α for the first $n_\mathcal{A}$ ones, and 1 for the rest. Similar to Algorithm 1, we can develop an efficient algorithm for domain weighted output kernel learning by using $\tilde{\mathbf{Y}} = \mathbf{DY}$ to replace $\mathbf{Y}$ and using $\tilde{\mathbf{K}} = \mathbf{DK}$ to replace $\mathbf{K}$ in Algorithm 1, shown in Table I. The parameter α could be chosen by cross-validation based on the target domain training data.

E. Domain Adaptive Input-Output Kernel Learning

We now present the proposed DA-IOKL algorithm which jointly learns the input and output kernels to reduce the domain shift in both kernel spaces. We first briefly revisit the MMD measure [18] for regularizing domain shift in the input kernel space, we then present the structural risk function and the corresponding optimization solution for the proposed DA-IOKL.

1) Domain Shift Measure : To reduce the domain shift in cross-domain recognition, we first need to define a domain shift measure based on the data from both domains. An efficient nonparametric criterion was proposed by Borgwardt et al. [18], which is referred as MMD, to compare the data distributions based on the distance between the sample means from two domains in a RKHS induced by a certain kernel function. Please refer to [18] for the details about MMD. It is worth noting that, MMD can only measure the domain shift in the input space of the vector-valued function.

2) Structural Risk with Multiple Kernel Formulation: Here we introduce a multiple kernel parameterization for the DA-IOKL algorithm for the domain adaptation purpose. Recall that Eqn. (5) defines the classification error on the training data, which is actually a function of $\mathbf{L}$ and $\mathbf{C}$. To learn an optimal input kernel function, we use a convex combination of base kernel functions to parameterize it, so that the new input kernel is $K_\mathbf{d}$ that is a function respect to coefficients $\mathbf{d}$ and has the form $K_\mathbf{d} = \sum_{m=1}^{M} d_m K_m, d_m \leq 0, m = 1, 2, \ldots, M$, as used in [26]–[28]. d_m's are the combination coefficients and form a column vector $\mathbf{d}$. K_m is the m-th base kernel function and M is the total number of base kernels used. Accordingly, the matrix formulation of the input kernel becomes

$$\mathbf{K_d} = \sum_{m=1}^{M} d_m \mathbf{K}_m, \tag{7}$$

then Eqn. (5) can be rewritten as,

$$Q(\mathbf{L}, \mathbf{C}, \mathbf{d}) := \frac{||\mathbf{Y} - \mathbf{K_d CL}||_F^2}{2\lambda} + \frac{\langle \mathbf{C^T K_d C}, \mathbf{L}\rangle_F}{2} + \frac{||\mathbf{L}||_F^2}{2},$$

where the objective function $Q(\cdot, \cdot, \cdot)$ is also a function of the kernel combination coefficients $\mathbf{d}$. It is easy to see that we could get an 'optimal' input kernel function by minimizing $Q(\cdot, \cdot, \cdot)$ w.r.t. $\mathbf{d}$.

In cross-domain recognition, reducing the domain shift is one critical concern to ensure the generalizability of the model when tested in the target domain, since the number of training samples from the target domain is usually small (e.g., 3 per category in our experiments). Reducing the domain shift could make the training samples from the source domain more similar to the samples from the target domain. In addition to reducing domain shift, minimizing classification error is the most important concern to ensure the model's discriminative ability. Therefore, by addressing the two concerns jointly, we include a MMD regularizing term into the structural risk function along with the total classification error, which also has a multiple kernel parameterization for the input kernel learning. Now the MMD regularizer could be written as,

$$\Omega(tr(\mathbf{K}_{\mathbf{d}}\mathbf{S})) = \frac{1}{2}tr(\sum_{m=1}^{M} d_m \mathbf{K}_m \mathbf{S})^2 = \frac{1}{2}\mathbf{d}^T\mathbf{p}\mathbf{p}^T\mathbf{d}, \quad (8)$$

where $\mathbf{p} = [p_1, \ldots, p_M]^T, p_m = tr(\mathbf{K}_m\mathbf{S})$, and $\mathbf{d} = [d_1, \ldots, d_M]^T$. $\mathbf{K}_m$ is the positive definite kernel matrix associated with the m-th base kernel function K_m.

At last, for the desired kernel function $K_{\mathbf{d}} = \sum_{m=1}^{M} d_m K_m$, we put a simple box constraint on the coefficients, which is $d_m \geq 0$ and $d_m \leq ub$, for $m = 1, 2, \ldots, M$, with $ub > 0$, instead of using the simplex constraint in other works [27]. We found that the box constraints are easier to solve for large-scale data set or high dimensional data, and could obtain good performances. In addition, in order to control the model complexity (preventing $||\mathbf{d}||$ to be too large), we place another regularizing term to bound the norm of the coefficient. We use $||\mathbf{d}||_1$ for the parameter selection purpose for a single image feature, and use $||\mathbf{d}||_2$ for cases where multiple image features are available.

Therefore, the final objective function for our DA-IOKL is:

$$\min_{\mathbf{d}\in\mathcal{D}} \min_{\mathbf{L},\mathbf{C}} \frac{1}{2}\mathbf{d}^T\mathbf{p}\mathbf{p}^T\mathbf{d} + \theta Q(\mathbf{L}, \mathbf{C}, \mathbf{d}) + \eta R(\mathbf{d}), \quad (9)$$

where the feasible set $\mathcal{D}$ is $\{\mathbf{d}|0 \preceq \mathbf{d} \preceq \mathbf{ub}, \}$, and $R(\mathbf{d})$ means the l_1 or l_2 norm of $\mathbf{d}$, and θ, η are penalty coefficients. By solving this optimization problem, we could learn the optimal input kernel function $k_{\mathbf{d}}$ and the output kernel matrix $\mathbf{L}$ jointly. In the next subsection, we propose an computationally efficient algorithm to solve Eqn. (9) in an alternating way.

3) Learning Algorithm: To solve the optimization problem of DA-IOKL, we make use of the Output Kernel Learning algorithm described in Table I and build our solution based on it. By defining $J(\mathbf{d}) = \min_{\mathbf{L},\mathbf{C}} Q(\mathbf{L}, \mathbf{C}, \mathbf{d})$, we can rewrite Eqn. (9) as,

$$\min_{\mathbf{d}\in\mathcal{D}} f(\mathbf{d}) = \min_{\mathbf{d}\in\mathcal{D}} \frac{1}{2}\mathbf{d}^T\mathbf{p}\mathbf{p}^T\mathbf{d} + \theta J(\mathbf{d}) + \eta R(\mathbf{d}). \quad (10)$$

For the above objective function $f(\mathbf{d})$, it is easy to show that the first part is linear w.r.t. $\mathbf{d}$ and $J(\mathbf{d})$ is quadratic w.r.t. $\mathbf{d}$. So that the objective function is convex w.r.t. $\mathbf{d}$, $\mathbf{C}$ and $\mathbf{L}$ separately, although not jointly convex. It is shown in [17] that $J(\mathbf{d})$ is invex w.r.t. $\mathbf{C}$ and $\mathbf{L}$ jointly, so we could infer that f is invex w.r.t $(\mathbf{d}, \mathbf{C}, \mathbf{L})$ jointly and thus has the global minima coinciding with a local minima. Therefore, we can employ an efficient alternating optimization algorithm to solve Eqn. (10).

Algorithm 2 DA-IOKL
```
1: $\mathbf{d} \leftarrow \frac{1}{M}\mathbf{1}_M$
2: while {not converges}
3:     $\mathbf{C}, \mathbf{L} \leftarrow$ Solution to $\min_{\mathbf{C},\mathbf{L}} Q(\mathbf{L}, \mathbf{C}, \mathbf{d})$
           by Algorithm I
4:     $\mathbf{d} \leftarrow$ Solution to Eq. (10) by L-BFGS-B
5: end while
```

TABLE II: The proposed Domain Adaptive Input-Output Kernel Learning Algorithm

The proposed DA-IOKL algorithm is iterative and each iteration contains two optimization steps. For a given $\mathbf{d}$, $J(\mathbf{d})$ can be computed and $\mathbf{C}$ and $\mathbf{L}$ can be estimated by using Algorithm I described in Table I. We first initialize $\mathbf{d}$ to get $\mathbf{C}$ and $\mathbf{L}$ by solving Eqn. (5). We then minimize the function $f(\mathbf{d})$ over $\mathbf{d}$ with the fixed $\mathbf{C}$ and $\mathbf{L}$. We repeat this two optimization steps for several iterations until convergence or the maximum number of iterations is reached. This DA-IOKL algorithm is summarized in Table II.

For the 2nd step of each iteration, to minimize $f(\mathbf{d})$ over $\mathbf{d}$, we first compute the gradient of $f(\mathbf{d})$ as,

$$\triangledown f = \mathbf{p}\mathbf{p}^T + \theta\triangledown_{\mathbf{d}} J + \eta\triangledown_{\mathbf{d}} R,$$

where $\triangledown_{\mathbf{d}} R$ is the gradient of the l_1 or l_2 norm. And $\triangledown_{\mathbf{d}} J$ is the gradient w.r.t $\mathbf{d}$ with fixed $\mathbf{L}$ and $\mathbf{C}$, which is given by

$$\begin{aligned}\frac{\partial J}{\partial d_m} &= \frac{1}{\lambda}[\frac{1}{2}tr(\mathbf{K}_m(\mathbf{C}\mathbf{L})^2(\mathbf{K}_d^T + \mathbf{K}_m)) - tr(\mathbf{K}_m\mathbf{C}\mathbf{L}\mathbf{Y}^T)] \\ &+ \frac{1}{2}tr(\mathbf{C}\mathbf{L}\mathbf{K}_m\mathbf{C}),\end{aligned}$$

where $\frac{\partial J}{\partial d_m}$ is the m-th element of $\triangledown_{\mathbf{d}} J$, $\mathbf{K}_{\mathbf{d}}$ is the fused kernel matrix given certain kernel coefficient $\mathbf{d}$, and $\mathbf{K}_m$ is the mth base kernel matrix.

After obtaining the gradient, we can use quasi-Newton methods with reasonable memory size to optimize $f(\mathbf{d})$ under a simple box constraint. Therefore, the Limited-memory BFGS with Bound constraints(L-BFGS-B) [19] is the natural choice for us. L-BFGS-B converges faster than the previous first-order methods [27] since it uses an approximate second-order Hessian update and is suitable for large-scale real world data due to its low rank approximation with limited memory size.

4) A True Multi-class Classifier : As illustrated in Figure 3, DA-IOKL consists of two major components for cross-domain object recognition: DA-IOKL training and DA-IOKL classification. Based on the joint set $X_{\mathcal{AB}}$, the DA-IOKL model can be trained by solving Eqn. (9) using the efficient algorithm described above. We now describe the corresponding DA-IOKL classifier. As stated in Section III-A1, for a certain RKHS, the associated vector-valued function $g(\cdot)$ is a true multi-classifier. According to the representer theory [16], the non-parametric form (matrix form) of the corresponding function in the RKHS of $g(\cdot)$ is given as $\mathbf{G} = \mathbf{C}\mathbf{L}$. And $\mathbf{G}$ is a $N \times m$ linear operator, where N is the number of training data and m is the number of categories. Let $\mathbf{d}^*, \mathbf{C}^*$ and $\mathbf{L}^*$ be the optimal solution learned by DA-IOKL, then the operator is $\mathbf{G}^* = \mathbf{C}^*\mathbf{L}^*$, and the input kernel function is $K^*_{\mathbf{d}}$. For an unlabeled data point x_t to be tested in the target domain, its row kernel vector computed based on the training data is given as

$$\mathbf{k}^N_{\mathbf{d}^*}(\mathbf{x}_t) = [K_{\mathbf{d}^*}(\mathbf{x}_t, \mathbf{x}_1), \ldots, K_{\mathbf{d}^*}(\mathbf{x}_t, \mathbf{x}_i), \ldots, K_{\mathbf{d}^*}(\mathbf{x}_t, \mathbf{x}_N)], \tag{11}$$

where $\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_N$ are N labeled training data points. For the testing data point x_t, its predicted label vector by using $\mathbf{G}^*$ is

$$\mathbf{y}_t = \mathbf{k}^N_{\mathbf{d}^*}(\mathbf{x}_t)\mathbf{G}^*, \tag{12}$$

where $\mathbf{y}_t$ is a m dimensional row vector. According to Section III-A1, the category label for testing data x_t can be determined as

$$y_t = \arg\max_{s \in \mathcal{T}} \mathbf{y}_t^{(s)}, \mathcal{T} = \{1, \ldots, m\}, \tag{13}$$

where $\mathcal{T}$ is the set of category labels, and $\mathbf{y}_t^{(s)}$ is the s-th element of the row vector $\mathbf{y}_t$. Therefore, Eqns. (11)(12)(13) together form a true multi-class classifier, the DA-IOKL classifier.

5) Extension to Multi-Source Domain Adaptation : Until now, we have only talked about adaptation from a single source domain, but it is straightforward to extend the proposed DA-IOKL to multi-source domain cases. By investigating the objective function of DA-IOKL, we can see that the MMD term and the domain weighting parameter α are the two items related to different domains. Take the case of having two source domains as an example, in Section IV-B2, we could simply employ two penalty weights α_1 and α_2 for each of the two source domains, which could be estimated by cross-validation on a grid of $[0, 1] \times [0, 1]$. For the MMD term, since we only want to reduce the domain shifts between the target domain and each of the source domains, we use the summation of the MMD's between two domains as the regularizer. Let $\mathcal{A}$ and $\mathcal{C}$ be two source domains, and $\mathcal{B}$ be the target domain, the MMD regularizer in the final objective function is

$$MMD_K^2(\mathcal{A}, \mathcal{B}) + MMD_K^2(\mathcal{C}, \mathcal{B}).$$

Similarly, we could also extend DA-IOKL to multiple source domains if needed.

F. Domain Shift Measure Based on Output Kernel Matrix

After proposing the specific DA-IOKL model, we look back to ask an important question: which domain should we adapt from? The selection of source domains should be based on domain shift or similarity between two domains. The smaller shift two domains have, the better performance the adaptation could get. Although the MMD introduced in Section III-E1 provides a measure of shift between $p_{\mathcal{A}}(\mathbf{x})$ and $p_{\mathcal{B}}(\mathbf{x})$, it cannot reflect the shift from $p_{\mathcal{A}}(\mathbf{y}|\mathbf{x})$ to $p_{\mathcal{B}}(\mathbf{y}|\mathbf{x})$, which is in the output kernel space. In addition, since in our proposed algorithm the output kernel $\mathbf{L}$ is estimated based on the input kernel K, we believe the domain shift information in K could also be reflected in $\mathbf{L}$. Therefore, we will introduce a domain shift measure based on the divergence of output kernel matrices in multi-class classification tasks. Besides serving as a domain shift measure, this divergence can intuitively explain how the proposed DA-IOKL handles shift in the output kernel space. From SectionIII-A, we know that a RKHS of $\mathcal{Y}$-valued functions can be associated with a unique $\mathcal{Y}$-kernel H, meaning that by carefully choosing a $\mathcal{Y}$-valued function we could use the associated H to characterize the underlying structure of data from a certain domain. In other words, if could use a pair of $(\mathbf{H}, g(\mathbf{x}))$ to describe the RKHS, we could use this pair to represent the domain. According to Equation (1), the $\mathcal{Y}$- kernel H can be further decomposed into an input part and an output part. For a fixed input kernel function, a unique output kernel $\mathbf{L}$ could be calculated which can reflect the category level structure. Therefore, by fixing a common input kernel function, the corresponding output kernel matrices for different domains can be seen as a domain signature.

We now proceed to define a metric on the signatures to measure the domain shift. Since the output kernel $\mathbf{L}$ is fix-ordered positive semi-definite (PSD), we have considered several popular metrics proposed for PSD matrices, including Riemannian Metric [35], Affine Invariant Riemannian Metric [36], Log-Euclidean Riemannian Metric [37], Bregman Divergence (so called $LogDet$) [38] and Jensen-Bregman $LogDet$ (JBLD) [39]. Among these, JBLD, a symmetric extended version of Bregman Divergence, is much easier to compute than the others. In this paper, we adopt the JBLD as a measure between output kernels $L_{\mathcal{A}}$ and $L_{\mathcal{B}}$, and we refer it as Output Kernel Divergence (OKD). The OKD between domains $\mathcal{A}$ and $\mathcal{B}$ is defined as the JBLD between $L_{\mathcal{A}}$ and $\mathcal{B}$,

$$\begin{aligned} J_{OKD}(\mathcal{A}, \mathcal{B}) &= J_{jbld}(\mathbf{L}_{\mathcal{A}}, \mathbf{L}_{\mathcal{B}}) \\ &= log|\tfrac{\mathbf{L}_{\mathcal{A}}+\mathbf{L}_{\mathcal{B}}}{2}| - \tfrac{1}{2}log|\mathbf{L}_{\mathcal{A}}\mathbf{L}_{\mathcal{B}}|, \end{aligned} \tag{14}$$

which is symmetric, nonnegative and invariant under congruence transformation [40]. This proposed output kernel divergence could be used as shift measure between domains $\mathcal{A}$ and $\mathcal{B}$. Suppose that we have multiple source domains, the divergence measures between domain pairs could be used as a criterion for domain selection, i.e., selecting the source domain with the smallest OKD value.

Unlike MMD which measures the shift in the input kernel space, OKD actually measures the shift in the output kernel space. We also notice that a Rand of Domain (ROD) metric was proposed in [12] for the domain selection purpose. Similar to MMD, the ROD is limited to the input kernel space (i.e. the distribution shift on $p(X)$), which fails to capture the shift in the output space. In Section IV, we will report illustrative OKD values computed based on real data. The results not only provide a criterion for source domain selection, but also demonstrate that the proposed DA-IOKL method could learn a RHKS where the data distributions from different domains are closer than in the original feature spaces. The results provide a direct and intuitive evidence that the proposed method can reduce domain shift in the output kernel space, which leads to better performances than previous state-of-art methods which mainly focus on the shift in the input feature space.

IV. EXPERIMENTS

In this section, we will evaluate the proposed DA-IOKL algorithm on the *DA* data set [3] and the *Cal+DA* data set [12]. Experiments on the *DA* data set are conducted for multi-class classification under three scenarios: single source adaptation, multi-source adaptation and multi-feature adaptation. The experiments on *Cal+DA* follow the protocol described in [12]. We also compute the OKD value between domains on *DA* dataset to demonstrate how our proposed method handles the domain shift in output kernel space. We compare our results with the state-of-art methods for all experiments. The results show that the proposed DA-IOKL consistently outperforms the state-of-art methods, which demonstrates the ability and robustness of DA-IOKL for cross-domain object recognition tasks.

A. Data Sets and Features

1) Object Domain Adaptation Data Set (DA): This benchmark data set for domain adaptation used in our experiments is released by Saenko et al. [3], which contains 31 object categories of images from the following 3 domains: *amazon*, *dslr* and *webcam*. In average, the *amazon* domain contains 90 instances for each category, whereas *dslr* domain and *webcam* domain have around 30 instances for each category. Moreover, for *dslr* and *webcam* domains, images are taken for 5 corresponding objects in each category. There are 4652 images in total in the data set.

To compare with the state-of-art results on this benchmark data set, we first use the same SURF [41] feature file released by Saenko et al. in [3]. All images are resized to the same width and converted to gray-scale. When abstracting SURF descriptors, the blob response threshold is set to be 1000, with other parameters left to be default values. A 64-dimensional non-rotationally invariant SURF descriptor is used to describe the patch surrounding each detected interest point. Then a codebook of size 800 is constructed by K-means clustering on a random subset of descriptors generated from images in the *amazon* domain. Finally, all images in the data set are represented by bag-of-word histograms formed by vector quantization using the 800 dimensional codebook.

To further study the feature combination ability of DA-IOKL granted by the multiple kernel parameterization in the input kernel part, in addition to SURF, we also abstract GIST [42], dense-SIFT [43], HoG [44] (signed and unsigned) and Local Self Similarity (LSS) [45]. We thus investigate 6 types of features in total.

2) Caltech + Domain Adaptation Data Set(Cal+DA): This data set is introduced in [12] to reduce the potential bias in the *DA* data set by adding the Caltech256 data set as the fourth domain in addition to the *amazon, dlsr* and *webcam* domains. They select 10 common categories between Caltech256 and *DA*: BACKPACK, TOURING-BIKE, CALCULATOR, HEAD-PHONE, COMPUTER-KEYBOARD, LAPTOP-101, COMPUTER-MONITOR, COMPUTER-MOUSE, COFFEE-MUG, and VIDEO-PROJECTOR. There are 8 to 151 images per category per domain, and 2533 images in total. Following Section IV-A1, the same SURF features are abstracted from images in Caltech domain and quantized into 800 dimensional bag-of-word histograms using the same dictionary.

B. Results on DA Data Set

In this section, we will describe the experiment details on the *DA* data set and discuss the results for each of the experimental settings. For the convenience of comparing our results with state-of-art results on this data set reported in [3]–[5], [11], we first follow the same experimental protocol by using the same SURF feature file as in these works. In addition, we also conduct experiments with multiple image descriptors abstracted from the same data set to demonstrate the ability of our model for feature combination. Specifically, we use only SURF feature in SectionIV-B1 and SectionIV-B2, and use multiple types of features in Section IV-B3. We also want to mention another point: Since there are images taken from the same objects in the domains *dslr* and *webcam*. To make a fair comparison, the same test objects are **held out** of training. In other words, if an image of a certain object is a test object, the images of the same object cannot be used during training. All the results reported for all methods in SectionIV-B1 and SectionIV-B2 are obtained by following this rule. However, the experiments in SectionIV-B3 don't leave out the images from the same objects.

1) Single Source Domain Adaptation : Following the settings in [3], [4], for single source domain adaptation experiments, there are labeled training images available for all categories in both the source and target domains at the training time. In every trial of the experiments, we randomly select 8 labeled images per category if *dslr* or *webcam* is used as the source domain, or 20 labeled images per category if *amazon* is used as the source. We also select 3 labeled images per category from the target domain. Note that in our experiments, images of the same test object are generally **held out** of training. For the exceptional cases, we use the mark $*$ to indicate 'without holding out the images of the same test object'. In particular, for each category, we use images of objects with IDs $\{1, 2, 3\}$ for training and $\{4, 5\}$ for testing in the *webcam* domain, and use images of objects with ID $\{1\}$ for training and $\{4, 5\}$ for testing in the *dslr* domain.

We use kernel functions with the form $k_m(x_i, x_j) = exp(-\gamma_m dist_m(x_i, x_j))$ as the kernel basis for the input kernel. We use the χ^2 distance, l_1 norm and l_2 norm as the $dist(\cdot,\cdot)$ function, leading to the χ^2 kernel, Laplacian kernel and Gaussian kernel functions respectively. And we select $\gamma_m = \frac{1}{dim}$, where dim is the dimensions of the feature vectors. For the parameters used in the proposed DA-IOKL algorithm, we use l_1 norm regularization as $R(\mathbf{d})$ in the objective function to enforce sparsity in the kernel coefficients for the purpose of selecting kernel parameters. We set θ to 1×10^{-4} and η to 1×10^{-3} empirically. Further, we cross-validate the loss penalty α on the training set and λ is set proportioned to the norm of the output matrix $\mathbf{Y}$. For single source adaptation experiments, the optimal α is 1 for *dslr/webcam* and for *webcam/dslr*, and 0.2 for *amazon/dslr*. We repeat the experiments 10 times and report the average accuracy with standard deviation. Results are shown in Table III.

To demonstrate the accuracy improvements brought by the proposed DA-IOKL algorithm, we also report the results from several baseline methods. The baseline methods are described as follows.

- **NC** stands for naive combination of the training data from source and target domains. A standard SVM is applied

Source	*Target*	NC	A-SVM	ITML [22]	symm [3]	ARC-t [4]	RDALR [11]	**DA-IOKL**
$dslr$	$webcam$	32.2 ± 1.4	33.0 ± 0.8	23	35.1	36.1	36.9 ± 1.2	**39.9** ± 1.1
$webcam$	$dslr$	22.1 ± 1.1	26.0 ± 0.7	18	27.5	25.3	30.1 ± 0.8	**34.4** ± 1.0
$amazon$	$dslr^*$	41.3 ± 1.3	42.2 ± 0.9	41	49.5	50.4	50.7 ± 0.8	**72.2** ± 2.0

TABLE III: *DA* Data Set: Classification accuracy results for single source adaptation. The average accuracy in % is reported and the corresponding standard deviation is included. Here $*$ means 'without holding out the images of he same test object'.

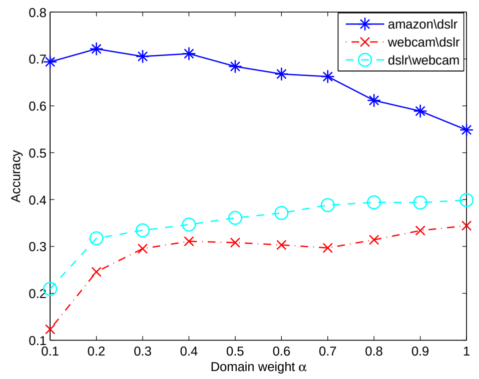


Fig. 4: Classification accuracy v.s. the penalty term α curves under three experimental settings.

to the combined training data.

- **A-SVM** applies Adaptive SVM [6] on the training data from both domains.
- **ITML** applies metric learning [22] on combined training data from both domains to learn a discriminative metric.
- **symm** applies metric learning [3] on the corresponding pairs of training samples between two domains to reduce the domain shift.
- **ARC-t** is presented in [4], which puts an asymmetric regularizer on the cross-domain metric learning problem. The metric is trained on all the data from both domains.
- **RDALR** is presented in [11], which learns an optimal linear operator to project the data from the source domain into an intermediate space by satisfying reconstruction constraints using the target domain data.

In Table III, the results for previous methods are directly quoted from the related papers published in the literature. From the table we can see that the proposed DA-IOKL outperforms all previous methods in all studied experimental settings, and generally the performance improvement is not negligible. Since the experimental case $amazon/dslr^*$ is conducted **without** holding out the images of the same test objects, we can see clearly from the table that all methods yield much better results than other experimental settings and that the proposed DA-IOKL provide the best performance on $amazon/dslr^*$. Since we only use the SURF type of features in the experiments in this section, the coefficients for the input kernel we learned are quite sparse and thus help select the best kernel function.

During the experiments, we also note that the domain weight α plays an important role in cross-domain object recognition. We visualize the recognition accuracy plots as a function of α for three experimental settings in Figure 4. From the figure we can see that *dslr* and *webcam* domains are close to each other, since $\alpha = 1$ gives the best performance, meaning that the algorithm tends to treat these two domains as the same. The best choice of α actually reveals the domain similarity for the training data, we therefore could use the MMD measure to guide the selection of α, if cross-validation is not feasible. For the rest of the paper, we choose the same α as used in this section for simplicity. In Table III, the best performance in *amazon/dslr* is achieved by setting $\alpha = 0.2$. To demonstrate the performance improvement by using domain weighting in the input and output kernel learning of DA-IOKL, we also conduct experiments with $\alpha = 1$ for $amazon/dslr^*$ and compare the results of different methods. We note that DA-IOKL still yields a much better accuracy at $63.5\% \pm 2.1\%$, which is 12.8% better than the state-of-art result in [11].

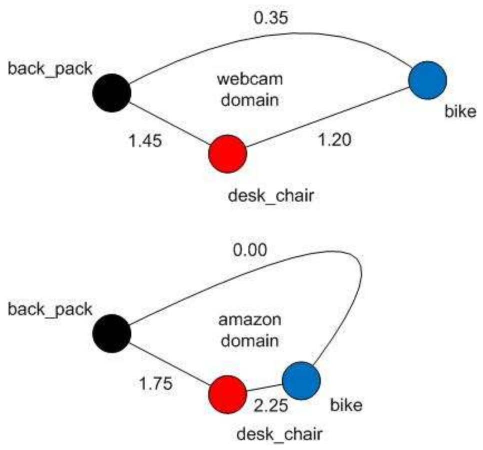


Fig. 5: Illustrations of the output kernels learned from the *webcam* domain (upper) and the *amazon* domain (lower), where the nodes represent the object categories and each edge represents the pair-wise distance between categories measured by a kernel score on it. The larger the output kernel score is, the more similar two categories are.

To understand the output kernel shift illustrated in Figure 2 in Section I, we visualize a part of the output kernel matrix in Figure 5. The output kernel scores are normalized for comparison. And the larger the kernel score is, the closer two nodes are. We can see from Figure 5 that the observations in the output kernel space are consistent with the observations in visual appearances shown in Figure 2 in Section I. We believe that the correct estimation of the relationship between categories by the proposed DA-IOKL leads to the improved performances in domain adaptation.

Source	*Target*	NC	A-SVM [6]	RDALR [11]	**DA-IOKL**
$amazon, dslr$	$webcam$	20.6 $\pm$ 1.8	30.4 $\pm$ 0.6	36.9 $\pm$ 1.1	**39.2** $\pm$ 2.0
$amazon, webcam$	$dslr$	16.4 $\pm$ 1.1	25.3 $\pm$ 1.1	**31.2** $\pm$ 1.3	**31.6** $\pm$ 1.9
$dslr, webcam$	$amazon$	16.9 $\pm$ 0.7	17.3 $\pm$ 0.9	20.9 $\pm$ 0.9	**25.2** $\pm$ 1.1

TABLE IV: *DA* Data Set: Classification accuracy results for multiple sources adaptation. The average accuracy in % and the corresponding standard deviation are reported.

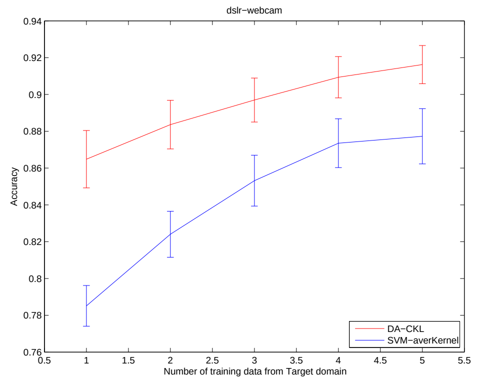
Source	*Target*	SimpleMKL [27]	SVM	**DA-IOKL**
$dslr$	$webcam$	88.7 $\pm$ 1.8	88.6$\pm$ 1.5	**91.4** $\pm$ 1.4
$webcam$	$dslr$	88.7 $\pm$ 1.4	90.3 $\pm$ 1.8	**92.7** $\pm$ 1.1
$amazon$	$dslr$	68.5 $\pm$ 1.9	65.7 $\pm$ 2.0	**76.5** $\pm$ 1.5

TABLE V: *DA* Data Set: Classification accuracy results for multiple features. The average accuracy in % is reported and the corresponding standard deviation is included.

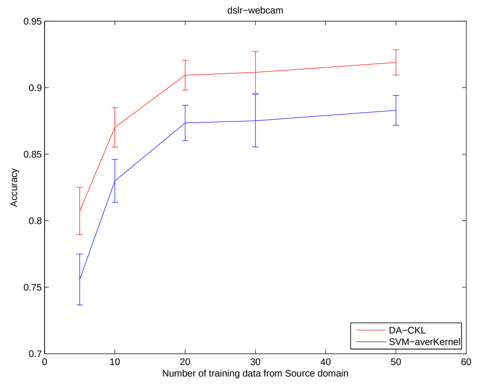
2) Multi-Source Domain Adaptation: We also study the performances of the proposed DA-IOKL for multi-source domain adaptation, where multiple different source domains are available during training. More specifically, following previous works, we conduct experiments for 2-source domain adaptation and evaluate DA-IOKL by comparing its performances with state-of-art methods: A-SVM [6] and RDALR [11]. The settings of training/testing samples follow Section IV-B1, where the same testing objects are held out during training. We report the average accuracy with standard deviation for 5 trials of experiments. Results are shown in Table IV. We can see that the proposed DA-IOKL consistently outperforms other methods. The performance improvement is significant in the cases *amazon,dslr/webcam* and *dslr,webcam/amazon*. In *amazon,webcam/dslr*, the result is slightly better than that of RDALR, though it is much better than that of NC and A-SVM. The results in Table IV demonstrate that the proposed DA-IOKL could successfully adapt from multiple source domains to improve the overall object recognition in the target domain.

3) Multi-Feature Domain Adaptation: To validate the proposed algorithm for the feature combination purpose, we conduct the experiments with using multiple features. The base kernels are the same as in previous sections, and all 6 types of features are used. We conduct classification experiments for *dslr/webcam*, *webcam/dslr* and *amazon/dslr* cases without holding out the images from the same objects during training. Results are shown in Table V. We compare with the multiple kernel learning method SimpleMKL [27] and a standard SVM using an average kernel. In DA-IOKL, we use the l_2 norm of $\mathbf{d}$ as regularizer R, to search for a meaningful combination of kernel bases. From Table V, we can see that the proposed DA-IOKL learns the optimal kernel coefficients successfully and the learned input and output kernels lead to better performances than the baseline methods. Its better performances than SimpleMKL demonstrate that DA-IOKL could combine multiple features efficiently for the domain adaptation tasks.

To show the impacts by varying the numbers of training samples from the source domain / target domain, we conduct the experiments where the number of training samples from the source domain is fixed to 20 and the number of training samples from the target domain is set as $\{1, 2, 3, 4, 5\}$ respectively. Also, we conduct the experiments where the number of training samples from the target domain is fixed to 4 and the number of training samples from the source domain is set as $\{5, 10, 20, 30, 50\}$ respectively. The results are shown in Figure 6. Here for illustration, we only show the results for *dslr/webcam* and only use the SVM with an average kernel as the baseline method, though similar observations are noted for other cases. From Figure 6, as expected, we note that the accuracy keeps increasing as the number of training samples from the target or source domain increases, and the accuracy saturates as the number of training samples from the source domain is sufficiently large. The DA-IOKL consistently outperforms the baseline method when employing different numbers of training samples.



(a) Accuracy v.s. the number of training samples from the target domain.



(b) Accuracy v.s. the number of training samples from the source domain.

Fig. 6: Accuracy results for the *dslr/webcam* case when varying the numbers of training samples from the target and source domains.

Method	C → A	C → D	A → C	A → W	W → C	W → A	D → A	D → W
NC	23.1 ± 0.4	26.5 ± 0.7	24.0 ± 0.3	31.6±0.6	20.8±0.5	30.8±0.6	31.3±0.7	55.5±0.7
symm [3]	33.7 ± 0.8	35.0 ± 1.1	27.3 ± 0.7	36.0±1.0	21.7±0.5	32.3±0.8	30.3±0.8	55.6±0.7
SGF [5]	40.2 ± 0.7	36.6 ± 0.8	37.7 ± 0.5	37.9±0.7	29.2±0.7	38.2±0.6	39.2±0.7	69.5±0.9
GFK(PCA,PCA) [12]	42.0 ± 0.5	49.5 ± 0.8	37.8 ± 0.4	53.7±0.8	32.8±0.7	42.8±0.7	45.0±0.7	78.7±0.5
GFK(PLS,PCA) [12]	46.1 ± 0.6	55.0 ± 0.9	39.6 ± 0.4	56.9±1.0	32.1±0.7	46.2±0.7	46.2±0.6	80.2±0.4
GFK(PLS,PLS) [12]	38.7 ± 0.6	38.6 ± 1.4	36.6 ± 0.4	36.3±0.9	28.6±0.6	36.3±0.5	35.0±0.4	74.6±0.5
DA-IOKL	**63.7** ± 1.5	**67.1** ± 4.2	**46.6** ± 1.5	**71.0**±3.4	**33.8**±1.7	**54.8**±2.3	**54.8**±2.8	**83.3**±1.6

TABLE VI: *Cal+DA* data set: Classification accuracy results, where the average accuracy in % and the corresponding standard deviation are reported.

C. Results on Cal+DA Data Set

Since the *DA* data set is a medium-scale data set and consists images from similar objects for the same categories across two different domains, this may lead to bias for the methods evaluated on this data set. To further demonstrate the robustness of the proposed DA-IOKL, we conduct domain adaptation experiments on the *Cal+DA* data set in this section, which contains the 4th domain consisting of images from Caltech256, in addition to *amazon, dslr* and *webcam* domains. When Caltech256 domain serves as a source domain, we randomly sample 20 images per category for training. When Caltech256 domain serves as the target domain, we randomly select 3 images per category for training. And the experiments follow the protocol mentioned in previous sections for other three domains. We report the average accuracy with standard deviation for 20 trials of experiments. We summarize the results for DA-IOKL and the stat-of-art methods in Table VI. We use the initials of the domain names in the table to describe the adaptation between two domains, e.g. $\mathbf{C} \rightarrow \mathbf{A}$ means that the source domain is Caltech256 and the target domain is *amazon*.

In the table, the methods **NC** and **symm** [3] are described in Section IV-B1. **SGF** is proposed in [5], which samples subspaces along the geodesic line between two domains to search for an intermediate space where the domain shift could be reduced. **GFK** [12] is a kernelized method based on geodesic flow presented by **SGF** and it is able to determine the optimal dimensions of subspaces automatically. From Table VI we could clearly see that the proposed DA-IOKL consistently outperforms the previous methods, generally with large improvement margins.

The methods **symm**, **SGF** and **GFK** are somehow similar to each other in the sense that they all look for a ‘better’ subspace by projecting the data from the input feature space using a linear projection. Although some of them also adopt non-linear projections using the kernel trick, these methods are still within the scope of the **Input** kernel space for scalar functions. In the contrast, the proposed DA-IOKL reduces the domain shift in the RKHSs of vector-valued functions, which is more comprehensive and contains both **Input** and **Output** kernels. Therefore, it is not surprising that the proposed DA-IOKL consistently provides the best results. We also note that DA-IOKL usually shows larger variances, especially for $\mathbf{C} \rightarrow \mathbf{D}$ and $\mathbf{A} \rightarrow \mathbf{W}$ cases. We believe it is caused by the randomness of selecting training samples from the source domain. Since DA-IOKL could make better use of the source training data, the randomness introduced by the source domain affects DA-IOKL more.

D. Domain Shift Measure in Output Kernel Space

Here we present one of the first empirical studies on *domain shift measure* between visual data domains in the output kernel spaces. Given the *DA* dataset, for a classification task on the target domain *dslr*, we would like to choose a more similar domain from the rest two as the source. To measure the domain shift, first we randomly select 20 samples from each domain, we then perform the DA-IOKL to learn the output kernel for each domain separately. We also learn the output kernels for all possible combinations of any domain pairs. In all the experiments in this subsection, we use linear kernel as the input kernel function. We use OKD to compute distances between these output kernels as the shift measures between domain pairs. The results are shown in Table VII.

Before computing the J_{OKD} divergence, we normalize each kernel by its largest diagonal entry to make all kernels comparable. J_{OKD} between two identical matrices is a constant, denoting by J_{self}. Therefore we scale the resulting similarity table by dividing all entries by J_{self} . The smallest J_{OKD} is 1.000, meaning that the two domains are identical. The smaller the corresponding J_{OKD} is, the more similar two domains are.

Now for the domain selection problem, according to Table VII, if *dslr* is the target domain, it is better to choose *webcam* as the source, since $J_{OKD}(webcam, dslr)$ is smaller given *amazon, dslr* as source candidates. This conclusion is supported by experimental results in Section IV-B1, indicating that the proposed OKD is powerful for domain shift measure. Another interesting observation noted from Table VII is that the *monotonicity* holds for OKD. In the experiments of domain adaptation, we actually learn a RKHS from a mixed data collection. For the 3 cases *dslr/webcam, webcam/dslr* and *amazon/dslr,* from Table VII, we know that the proposed DA-IOKL essentially learns an intermediate RKHS between the source and target domains, which proves that DA-IOKL indeed addresses the domain shift in the output kernel space.

For comparison, we also compute the domain similarity using the Maximum Mean Discrepancy (MMD) measure [18]. Table VIII shows the MMD measure results, from which we can see that MMD mis-judges the similarity between *webcam-dslr* and *webcam-amazon,* and it does not reveal the *monotonicity* along domain changes. In addition, [12] reports the results of ROD metric on these three domains, which coincide with our OKD results and the MMD results for the

Domain	webcam	dslr	amazon	webcam+dslr	amazon+dslr	amazon+webcam	webcam + dslr + amazon
webcam	1.000	1.293	1.314	1.165	1.301	1.247	1.353
dslr	1.293	1.000	1.410	1.195	1.280	1.375	1.683
amazon	1.314	1.410	1.000	1.270	1.150	1.107	1.201
webcam+dslr	1.165	1.195	1.270	1.000	1.189	1.209	1.125
amazon+dslr	1.300	1.280	1.150	1.189	1.000	1.133	1.070
amazon+webcam	1.247	1.375	1.107	1.209	1.133	1.000	1.053
webcam+dslr+amazon	1.353	1.683	1.202	1.125	1.070	1.053	1.000

TABLE VII: Domain shift measurements between all possible domain pairs by the proposed OKD measure.

Domain	webcam	dslr	amazon	webcam+dslr	amazon+dslr	amazon+webcam	webcam+dslr+amazon
webcam	0.000	1.281	1.026	0.374	0.437	0.271	0.180
dslr	1.281	0.000	2.861	0.303	0.744	1.801	0.872
amazon	1.026	2.861	0.000	1.718	0.756	0.309	0.754
webcam+dslr	0.374	0.303	1.718	0.000	0.299	0.788	0.200
amazon+dslr	0.437	0.744	0.756	0.299	0.000	0.336	0.071
amazon+webcam	0.271	1.801	0.309	0.788	0.336	0.000	0.203
webcam+dslr+amazon	0.180	0.872	0.754	0.200	0.071	0.203	0.000

TABLE VIII: Domain similarity results between all possible domain pairs by MMD [18]. The MMD value is 0 between two identical distributions. And the smaller the MMD is, the similar two distributions are.

three domains. However, they didn't include the results on transiting domains (e.g. *webcam-amazon*), we didn't include them here for further comparison.

V. Conclusion

In this paper, we introduce the output kernel analysis into the domain adaptation problem. We also propose a Domain Adaptive Input-Output Kernel Learning algorithm, referred as DA-IOKL, to learn an optimal input and output kernel space for cross-domain image object recognition. The proposed DA-IOKL is generally applicable to single source, multiple sources and multiple features domain adaptation tasks, and our experiment results show that it consistently outperforms the state-of-art methods when tested on two standard benchmark data sets. For the future work, to get more compact representations and a more efficient algorithm, we plan to study RKHSs for operator-valued functions.

Acknowledgment

This work was supported by the Canadian Natural Sciences and Engineering Research Council (NSERC) under Grant STPGP 365164-08 and Grant 11R82396.

References

[1] "Ecml/pkdd discovery challenge," 2006. [Online]. Available: http://www.ecmlpkdd2006.org/challenge.html

[2] A. Bergamo and L. Torresani, "Exploiting weakly-labeled web images to improve object classification: A domain adaptation approach," in *Advances in Neural Information Processing Systems(NIPS)*, 2010.

[3] K. Saenko, B. Kulis, M. Fritz, and T. Darell, "Adapting visual category models to new domains," in *European Conference on Computer Vision (ECCV)*, 2010.

[4] B. Kulis, K. Saenko, and T. Darrell, "What you saw is not what you get: Domain adaptation using asymmetric kernel transforms," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[5] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *International Conference on Computer Vision (ICCV)*, 2011.

[6] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive svms," in *ACM Multimedia*, 2007.

[7] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *Conference of the Association for Computational Linguistics (ACL)*, 2007.

[8] E. Zhong, W. Fan, J. Peng, K. Zhang, J. Ren, D. Turaga, and O. Verscheure, "Cross domain distribution adaptation via kernel mapping," in *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2009.

[9] S. Ben-david, J. Blitzer, K. Crammer, and P. M. Sokolova, "Analysis of representations for domain adaptation," in *Advances in Neural Information Processing Systems(NIPS)*, 2007.

[10] H. Daumé III, "Frustratingly easy domain adaptation," in *Conference of the Association for Computational Linguistics (ACL)*, 2007.

[11] I.-H. Jhuo, D. Liu, D. Lee, and S.-F. Chang, "Robust visual domain adaptation with low-rank reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012. [Online]. Available: http://www.ee.columbia.edu/ln/dvmm/publications/12/DomainAdaptation.pdf

[12] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012. [Online]. Available: http://www.cs.utexas.edu/∼grauman/papers/subspace-cvpr2012.pdf

[13] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Schölkopf, "Correcting sample selection bias by unlabeled data," in *Advances in Neural Information Processing Systems(NIPS)*, 2006.

[14] L. Duan, I. W. Tsang, D. Xu, and S. J. Maybank, "Domain transfer svm for video concept detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[15] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[16] C. A. Micchelli and M. A. Pontil, "On learning vector-valued functions," *Neural Computation*, 2005.

[17] F. Dinuzzo, C. S. Ong, P. Gehler, and G. Pillonetto, "Learning output kernels with block coordinate descent," in *International Conference on Machine Learning (ICML)*, 2011.

[18] K. M. Borgwardt, A. A. Gretton, B. M. J. Rasch, H. peter Kriegel, A. B. Schölkopf, and B. A. J. S. D, "Integrating structured biological data by kernel maximum mean discrepancy," in *In ISMB*, 2006.

[19] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on Scientific and Statistical Computing*, 1995.

[20] D. Kifer, S. Ben-David, and J. Gehrke, "Detecting change in data streams," in *International Conference on Very Large Data Bases (VLDB)*, 2004.

[21] S. J. Pan, I. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, 2011.

[22] J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon, "Information-theoretic metric learning," in *International Conference on Machine Learning (ICML)*, 2007.

[23] P. Jain, B. Kulis, J. V. Davis, and I. S. Dhillon, "Metric and kernel learning using a linear transformation," *Journal of Machine Learning Research (JMLR)*, 2012.

[24] A. G. an V. Vovk and V. Vapnik, "Learning by transduction," in *Conference on Uncertainty in Artificial Intelligence (UAI)*, 1998.

[25] W. Jiang, E. Zavesky, and S.-F. Chang, “Cross-domain learning methods for high-level visual concept classification,” in *International Conference on Image Processing (ICIP)*, 2008.

[26] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, “Large scale multiple kernel learning,” *Journal of Machine Learning Research (JMLR)*, 2006.

[27] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, “Simplemkl,” *Journal of Machine Learning Research*, 2008.

[28] M. Varma and B. R. Babu, “More generality in efficient multiple kernel learning,” in *International Conference on Machine learning (ICML)*, 2009.

[29] G. R. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, “Learning the kernel matrix with semidefinite programming,” *Journal of Machine Learning Research (JMLR)*, 2004.

[30] C.Carmelie, E. D. Vito, and A. Toigo, “Vector valued reproducing kernel hilbert spaces of integrable functions and mercer theorem,” *Analysis and Applications*, 2006.

[31] M. A. Álvarez, L. Rosasco, and N. D. Lawrence, “Kernels for vector-valued functions: a review,” *Foundations and Trends in Machine Learning*, vol. 4, pp. 195–266, 2012.

[32] M. Belkin, P. Niyogi, and V. Sindhwani, “Mannifold regularization: A geometric framework for leanring from labeled and unlabeled examples,” *Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

[33] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, “Learning bounds for domain adaptation,” in *Advances in Neural Information Processing Systems(NIPS)*, 2007.

[34] G. Schweikert, C. Widmer, B. Schölkopf, and G. Rätsch, “An empirical analysis of domain adaptation algorithms for genomic sequence analysis,” in *Advances in Neural Information Processing Systems(NIPS)*, 2008.

[35] W. Förstner and B. Moonen, “A metric for covariance matrices,” *Tech. Report of the Dpt of Geodesy and Geoinformatics, Stuttgart University*, p. 113128, 1999.

[36] X. Pennec, P. Fillard, and N. Ayache, “A riemannian framework for tensor computing,” *International Journal of Computer Vision(IJCV)*, 2006.

[37] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, “Log euclidean metrics for fast and simple calculus on diffusiontensors,” *Magnetic Resonance in Medicine*, 2006.

[38] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh, “Clustering with bregman divergences,” in *Journal of Machine Learning Research*, 2005.

[39] A. Cherian, S. Sra, A. Banerjee, and N. Papanikolopoulos, “Efficient similarity search for covariance matrices via the jensen-bregman logdet divergence,” in *International Conference on Computer Vision (ICCV)*, 2011.

[40] A. Banerjee, D. Boley, and S. Acharyya, “Symmetrized bregman divergences and metrics,” in *The Learning Workshop*, 2009.

[41] H. Bay, T. Tuytelaars, and L. V. Gool, “Surf: Speeded up robust features,” in *Computer Vision and Image Understanding (CVIU)*, 2008.

[42] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *International Journal of Computer Vision*, 2001.

[43] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[44] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2005.

[45] E. Shechtman and M. Irani, “Matching local self-similarities across images and videos,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007.