Adaptive Component Embedding for Domain Adaptation

Mengmeng Jing, Jidong Zhao, Jingjing Li, Lei Zhu, Yang Yang, and Heng Tao Shen

***Abstract*—Domain adaptation is suitable for transferring knowledge learned from one domain to a different but related domain. Considering the substantially large domain discrepancies, learning a more generalized feature representation is crucial for domain adaptation. On account of this, we propose an adaptive component embedding (ACE) method, for domain adaptation. Specifically, ACE learns adaptive components across domains to embed data into a shared domain-invariant subspace, in which the first-order statistics is aligned and the geometric properties are preserved simultaneously. Furthermore, the second-order statistics of domain distributions is also aligned to further mitigate domain shifts. Then, the aligned feature representation is classified by optimizing the structural risk functional in the reproducing kernel Hilbert space (RKHS). Extensive experiments show that our method can work well on six domain adaptation benchmarks, which verifies the effectiveness of ACE.**

***Index Terms*—Distribution alignment, domain adaptation, subspace learning, transfer learning.**

I. Introduction

IN REAL-WORLD applications of visual recognition, data in the source domain on which classifiers are trained may significantly differ from data in the target domain to which those classifiers are applied. For example, in the sketch–photograph recognition task, photographs of a suspect are not provided and the best substitutes are sketched drawings based on the recollection of eyewitnesses [1]. The task is to train a model with sketches and apply it to identify the suspect. If we view sketches as the source domain, and photographs of the suspects as the target domain, then the sketch–photograph recognition task becomes a typical cross-domain task. The traditional machine-learning methods would fail to solve this kind of problem as the source- and target-domain data are drawn from different probability distributions. In recent years, domain adaptation has shown excellent performance in solving such a challenge by reducing the mismatches between the source and target domains. Domain adaptation is widely applied in image classification [2]–[13]; cross-language text classification [14], [15]; sentiment analysis [16], [17]; cold-start recommendation [18]; and so on.

Domain adaptation has two different settings: 1) semisupervised and 2) unsupervised. In the semisupervised setting, all of the source-domain data have labels, while only a small part of the target-domain data is labeled. In the unsupervised setting, none of the target labels are provided, which is more challenging compared with the semisupervised setting. In this article, we handle the unsupervised domain adaptation. In this setting, performance would degrade in most cases if we directly apply the classifier trained on the source domain to the target domain. Thus, it is important and indispensable to reduce the domain disparities in the beginning.

Most of the existing unsupervised domain adaptation methods can be roughly categorized into three categories [19]: 1) feature-based methods [2], [3], [20]–[26]; 2) classifier-based methods [27]–[29]; and 3) neural-network-based methods [4], [5], [30]–[32]. Classifier-based [27]–[29] methods first train a classifier with the source samples as prior knowledge. Then, the learned parameters and samples from both domains are utilized to train the target classifier. Nevertheless, when the discrepancies between domains are significantly large, classifiers learned in the source domain cannot be applied to the target domain. Recently, plenty of neural-network-based methods [4], [5], [30]–[32] are proposed, and have achieved excellent performance in domain adaptation. These methods usually manage to train deep neural networks and perform knowledge transfer in a unified architecture. However, training deep neural networks is very time consuming. Feature-based methods [2], [3], [20]–[24] learn a new feature representation where discrepancies between domains, for example, distribution discrepancies and geometry discrepancies, are significantly reduced.

The proposed method falls into the feature-based group, which assumes that there exists a feature space where discrepancies between domains are reduced. There are two main strategies to reduce the domain discrepancies: 1) statistics alignment [2], [3], [20], [21] and 2) geometry alignment [9], [13], [24]. Statistics alignment methods [2], [3], [20], [21] minimize the statistics of samples to match two domains. Usually, these sample statistics include the first-order statistics, that is, mean, and the second-order statistics, that is, variance and covariance [21]. For example,

Manuscript received October 14, 2019; revised January 19, 2020; accepted February 8, 2020. Date of publication March 6, 2020; date of current version June 23, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61806039 and Grant 61832001, and in part by the Sichuan Department of Science and Technology under Grant 2019YFG0141. This article was recommended by Associate Editor Y. S. Ong. *(Corresponding author: Jingjing Li.)*

Mengmeng Jing, Jidong Zhao, Jingjing Li, Yang Yang, and Heng Tao Shen are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: lijin117@yeah.net).

Lei Zhu is with the School of Information Science and Engineering, Shandong Normal University, Jinan 250358, China.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2020.2974106.

Digital Object Identifier 10.1109/TCYB.2020.2974106

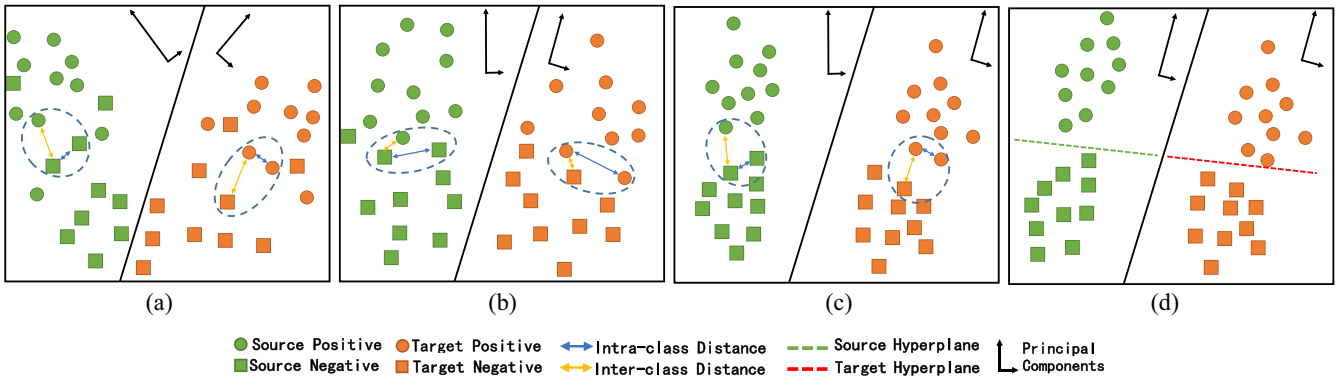


Fig. 1. Main idea of our method. (a) Distribution discrepancies of the source and target data in their original space are substantially large. (b) Previous methods usually project data onto a subspace to align the first-order statistics, for example, marginal and conditional distributions, which could dramatically reduce the distribution discrepancies. However, comparing the neighborhood relations among points circled by the blue circles in (a) with that in (b), the intraclass distance may become farther and the interclass distance may become closer in the process of the first-order alignment. (c) Our method attempts to preserve the local relations among data points when aligning the first-order statistics. Thus, the intraclass distance becomes closer, and the interclass distance becomes farther. (d) Our method further aligns the second-order statistics. Then, a classifier can be easily learned to classify the target labels.

TCA [20] uses the maximum mean discrepancy (MMD) as a metric to empirically measure the distribution discrepancy between domains and optimizes it to minimize the statistical discrepancies. CORAL [21] recolored whitened source features with the second-order statistics of the target distribution, that is, covariance, to align the two domains [21]. Geometry alignment methods [9], [13], [24] are based on the manifold assumption that data are originally lying on a low-dimensional manifold, but are embedded into a high-dimensional space [33]. The geometric structures of the manifold represent neighborhood relations among samples, that is, samples close to each other usually belong to the same class while samples far away from each other usually belong to different classes, which are crucial if we predict sample labels based on the distance. In geometry alignment methods, undirected graphs are constructed to maintain the locality consistency when projecting data onto the feature subspace [24].

However, when the discrepancies between domains are substantially large, either of the statistics alignment or geometry alignment alone would fail. For example, if we only apply the statistics alignment, the common subspace where distribution discrepancies are reduced may not exist [23]. Furthermore, if we only employ the geometry alignment, the neighborhood relations among samples cannot represent similarities in categories since the distributions are not aligned. In other words, samples close to each other geometrically may belong to different categories. Fortunately, the two strategies are complementary to each other. On the one hand, after projecting data onto the subspace where the statistical discrepancies are reduced, the geometric structures underlying the data manifold may be broken. Hence, we need to employ the geometry alignment to maintain the locality consistency after aligning the statistics. On the other hand, the distribution divergences may still exist after preserving the geometric structures.

To take full advantage of both strategies, in this article, we propose a novel method, called adaptive component embedding (ACE), for domain adaptation. ACE contains two steps to learn a new feature representation. Concretely, in the first step, some adaptive components are learned to embed data into a subspace where the first-order statistics of distributions are aligned, and the local neighborhood relations residing in the data manifold are preserved. More concretely, the source and target domains are different but related. There may exist some components that make discrepancies between two domains larger. There may also exist some components that make two domains closer. We try to discover the adaptive components that can maximize knowledge transfer from the source domain to the target domain. With these adaptive components, the new feature representation can be learned via embedding data of both domains into a domain-invariant subspace. To guarantee the effectiveness of the new feature representation, we impose distribution adaptation and manifold regularization. In the second step, we further align the second-order statistics of distributions. Finally, a classifier is learned by performing structural risk minimization (SRM) in reproducing kernel Hilbert space (RKHS). The main idea of our method is illustrated in Fig. 1.

The contributions of this article are listed as follows.

1) Aiming at reducing the large difference between domains, we propose a two-step method to significantly reduce the distribution discrepancies through aligning not only the first-order statistics but also the second-order statistics.
2) To avoid damaging the geometric structures in the process of distribution matching, we carefully design two graphs that cannot only preserve the neighborhood relations between intraclass samples but also save the discriminant information between interclass samples.
3) We carry out comprehensive experiments on six datasets. The selected datasets cover the standard dataset, large-scale dataset, traditional feature dataset, deep feature dataset, and even the "synthetic-to-real" dataset. The experimental results show that our method outperforms state-of-the-art methods across all of the evaluations, which fully verifies the effectiveness of our method.

II. Related Work

A. Feature-Based Methods

This kind of methods tries to find a new feature representation, where the domain distribution is aligned and some important properties, for example, geometrical or statistical properties, are preserved [2], [3], [20]–[24]. For example, GFK [22] embeds the source and target data into a Grassmann manifold where the geodesic flow kernel is constructed to derive the low-dimensional representation that is invariant to both domains. CORAL [21] aligns the source domain with the target one by matching the second-order statistics, that is, covariance. TCA [20] attempts to learn the transfer components in RKHS by MMD so that the source and target domain are obtained closer in the subspace spanned by these transfer components. JDA [3] extends TCA by jointly aligning the marginal and conditional distributions to minimize the distribution divergences between domains.

In all the feature-based methods, the most related work with our method is ARTL [24], which also optimizes statistical alignment, geometry alignment, and SRM simultaneously. However, our method is different from ARTL in at least three aspects as follows.

1) Both ARTL and our method construct an intrinsic graph to preserve the local geometric structures in the feature space. Moreover, our method constructs a penalty graph to preserve the global discriminating information.
2) ARTL optimizes SRM with statistical alignment and geometry alignment in one step. Our method first learns an adaptive feature representation by employing statistical alignment and geometry alignment. Then, with the new feature representation, we learn a classifier by optimizing SRM. Obviously, our two-step optimization process is more flexible. For example, ARTL can only learn a classifier through optimizing SRM. Our method, however, can learn different kinds of classifiers (SVM, kNN, etc.) with the new feature representation. The new feature representation allows high-order statistics alignment.
3) Both ARTL and our method reduce discrepancies between the sample means of the two domains. Apart from this, our method also aligns the sample covariance of the two domains.

The recent work JGSA [2] is a little similar to our method. The difference between our method and JGSA is as follows.

1) Both JGSA and our method learn the within-class matrix to make samples belonging to the same class closer and the between-class matrix to make samples with different labels farther in the feature space. However, the discriminating ability of JGSA is based on intraclass and interclass scatters, which is optimal only in cases where the data of each class are approximately Gaussian distributed, a property that cannot always be satisfied in real-world applications [33].
2) JGSA only aligns the first-order statistics while our method could align the first- and second-order statistics simultaneously.

B. Classifier-Based Methods

These methods are mainly SVM-based methods [27]–[29]. Bruzzone and Marconcini [27] extended the TSVM to gradually predict the target-domain labels, and remove the labeled source-domain data in the meantime. Yang *et al.* [28] learned an adaptive support vector machine (A-SVM) for the target domain, which is adapted from classifiers trained with the source-domain data. Schweikert *et al.* [29] trained multiple auxiliary classifiers with the source-domain data, and one classifier with the target-domain data. Then they combine the auxiliary classifiers with the target one to help classify the target data.

C. Neural-Network-Based Methods

Recently, the literature has witnessed the appealing performance of deep neural networks on many fields, including domain adaptation [4], [5], [30]–[32]. DDC [5] uses the linear-kernel MMD to regularize the adaptation layer of a tailored AlexNet to increase the domain invariance. DAN [4] uses the multikernel MMD to reduce the distribution difference when embedding the deep features into RKHS. DANN [31] trains generalized adversarial networks to learn features that are both discriminative and invariant to the change of domains. DAH [32] uses a unique loss function to train a deep neural network which outputs binary hash codes instead of probability values for classification.

Another line of works [34]–[37] tried to minimize the JS-divergence or KL-divergence of the distributions to learn a deep generative model which can image-to-image translate samples across domains, so that the classifier trained on the labeled source domain can be adapted to classify the unlabeled target domain. For instance, CyCADA [35] adopts CycleGAN [34] to reduce the JS-divergence of the distributions, while ensuring cycle consistency and semantic consistency when translating images across domains. UNIT [36] learns a pair of GANs and VAEs to translate images into a shared latent space where the distribution divergences between domains are minimized. However, the aforementioned two methods only involve the deterministic one-to-one mappings, which would fail if the cross-domain mapping is multimodal. MUNIT [37] improves UNIT by disentangling the shared latent space into a common content space and a specific style space.

D. Representation Transfer Methods

The representation transfer methods can use a large number of unlabeled samples in the source domain to improve performance in the target domain. Raina *et al.* [38] used sparse coding to obtain the high-level representation of unlabeled samples to perform the self-taught learning. RD-STL [39] improves Raina's work by enhancing the robustness of the model for outliers and noisy samples in the unlabeled samples. However, the aforementioned methods depend heavily on the compatibility between domains, which is not guaranteed in practical applications [40]. To alleviate this problem, a TPL [40] not only learns the adaptation factors for the higher

level representation but also learns supplementary knowledge in the target domain.

III. Proposed Method

A. Problem Definition

Definition 1 [9]: A domain $\mathcal{D}$ contains three portions: 1) a feature space $\mathcal{X}$; 2) its probability distribution $\mathcal{P}(\mathbf{X})$; and 3) the corresponding label set $\mathcal{Y}$, where $\mathbf{X} \in \mathcal{X}$. Subscripts s and t are used to denote variables of the source and target domains, respectively.

Task 1 [9]: Given a labeled source domain $\mathcal{D}_s = \{\mathcal{X}_s, \mathcal{P}(\mathbf{X}_s), \mathcal{Y}_s\}$ and an unlabeled target domain $\mathcal{D}_t = \{\mathcal{X}_t, \mathcal{P}(\mathbf{X}_t), \mathcal{Y}_t\}$, where $\mathbf{X}_s \in \mathcal{X}_s$, $\mathbf{X}_t \in \mathcal{X}_t$, $\mathcal{D}_s \neq \mathcal{D}_t$, and $\mathcal{Y}_s = \mathcal{Y}_t$, the marginal probability distribution $\mathcal{P}(\mathbf{X}_s) \neq \mathcal{P}(\mathbf{X}_t)$ and the conditional probability distribution $\mathcal{P}(\mathbf{Y}_s|\mathbf{X}_s) \neq \mathcal{P}(\mathbf{Y}_t|\mathbf{X}_t)$, learn a set of adaptive components to embed data into a d-dimensional ($d \ll m$) subspace in which the discrepancies between domains are reduced both statistically and geometrically.

For clarity, the frequently used notations and corresponding descriptions are shown in Table I.

TABLE I
Notations and Descriptions

Notation	Size	Description
n_s, n_t	-	number of source/target samples
m, d	-	dimensionality of original space/subspace
$\mathbf{X}_s$, $\mathbf{X}_t$	$m \times n_s / m \times n_t$	source/target samples
$\mathbf{Z}_s$, $\check{\mathbf{Z}}_s$, $\mathbf{Z}_t$	$d \times n_s / d \times n_s / d \times n_t$	source/aligned source/target feature
$\mathbf{L}$, $\tilde{\mathbf{L}}$, $\mathbf{W}$	$(n_s+n_t) \times (n_s+n_t)$	Laplacian/normalized Laplacian/weight matrix
$\mathbf{M}_0$, $\mathbf{M}_c$	$(n_s+n_t) \times (n_s+n_t)$	marginal/conditional MMD matrix
K, $\tilde{\mathbf{K}}$	$(n_s+n_t) \times (n_s+n_t)$	kernel function/matrix
$\mathbf{C}_s$, $\check{\mathbf{C}}_s$, $\mathbf{C}_t$	$d \times d$	source/aligned source/target covariance matrix
$\mathbf{P}$	$m \times d$	adaptive components
$\mathbf{V}$	$d \times d$	covariance alignment matrix

B. Problem Formulation

Algorithm Overview: The proposed method is motivated by the domain adaptation theory [41], which bounds the target expected error by three terms: 1) the source expected error; 2) the discrepancy distance between two distributions; and 3) the inadaptability of two domains. In view of this, we minimize the three terms, respectively. First and foremost, we learn several adaptive components to obtain the new feature representation where the first-order statistics of samples are aligned and, at the same time, the geometric structures underlying the data manifold are preserved as much as possible. In this step, the first-order statistics alignment could reduce the discrepancy distance between two distributions, and the geometric structure preservation could alleviate the inadaptability of two domains [24]. Next, we align the second-order statistics of the features to further minimize the domain discrepancies. Eventually, a classifier is learned by optimizing SRM to reduce the source expected error. For a global understanding, we report the objective function as follows:

$$\begin{aligned}\arg\min_{f \in \mathcal{H}_{\mathcal{K}}, \boldsymbol{P}} \ & \sum_{i=1}^{n} \ell\big(f\big(\boldsymbol{P}^T x_i\big), y_i\big) + \eta \Omega(f) \\ & + R_1(\boldsymbol{X}_s, \boldsymbol{X}_t, \boldsymbol{P}) + R_2\big(\boldsymbol{P}^T X_s, \boldsymbol{P}^T X_t\big). \end{aligned} \tag{1}$$

In (1), $\mathbf{P}$ is the set of the adaptive components that can project data onto a domain-invariant subspace. R_1 and R_2 are the first-order and second-order statistics alignment regularizer terms, respectively. ℓ is a loss function which can be one of the regularized least-square loss and SVM loss. Ω is a regularizer in the ambient space. η is a tradeoff parameter to balance the importance of the loss function and the ambient regularizer. In the rest of this section, we will explain each portion of the objective function.

First-Order Statistics Alignment: Since the source- and target-domain data are usually drawn from different distributions, that is, $\mathcal{P}(\mathbf{X}_s) \neq \mathcal{P}(\mathbf{X}_t)$ and $\mathcal{P}(\mathbf{Y}_s|\mathbf{X}_s) \neq \mathcal{P}(\mathbf{Y}_t|\mathbf{X}_t)$, the brute-force knowledge transfer from the source domain to the target one will cause poor performance. Hence, the priority is to reduce the significant distribution discrepancies between domains. However, in practice, the distributions of the source and target domains are usually unknown. Alternatively, we resort to align the first-order statistics, that is, mean, to reduce the distribution discrepancies. Specifically, we employ the empirical MMD as a metric to estimate the differences between domains. It is noteworthy that MMD is defined in the RKHS space. For the sake of computing efficiency, we compute the simplified MMD in the original space. The marginal distribution discrepancies can be optimized by

$$\arg\min_{\boldsymbol{P}} \operatorname{Tr}\big(\boldsymbol{P}^T \boldsymbol{X} \boldsymbol{M}_0 \boldsymbol{X}^T \boldsymbol{P}\big) \tag{2}$$

where $\mathbf{X} = [\mathbf{X}_s, \mathbf{X}_t]$ is the source- and target-domain data and Tr denotes the trace operation. $\mathbf{M_0}$ can be computed as

$$(\mathbf{M}_0)_{ij} = \begin{cases} \frac{1}{n_s n_s}, & x_i, x_j \in \mathbf{X}_s \\ \frac{1}{n_t n_t}, & x_i, x_j \in \mathbf{X}_t \\ \frac{-1}{n_s n_t}, & \text{otherwise.} \end{cases} \tag{3}$$

Computing the conditional MMD requires labels of samples. However, the labels of the target domain are unknown. Here, we use the pseudolabels of the target domain, which can be simply predicted by adopting the basic classifiers, for example, SVM and kNN. Then, the conditional MMD can be computed by

$$\arg\min_{\mathbf{P}} \sum_{c=1}^{C} \operatorname{Tr}\big(\mathbf{P}^T \mathbf{X} \mathbf{M}_c \mathbf{X}^T \mathbf{P}\big) \tag{4}$$

where C is the number of classes, and $\mathbf{M}_c$ can be computed as

$$(\mathbf{M}_c)_{ij} = \begin{cases} \frac{1}{n_s^{(c)} n_s^{(c)}}, & x_i, x_j \in \mathbf{X}_s^{(c)} \\ \frac{1}{n_t^{(c)} n_t^{(c)}}, & x_i, x_j \in \mathbf{X}_t^{(c)} \\ \frac{-1}{n_s^{(c)} n_t^{(c)}}, & \begin{cases} x_i \in \mathbf{X}_s^{(c)}, & x_j \in \mathbf{X}_t^{(c)} \\ x_j \in \mathbf{X}_s^{(c)}, & x_i \in \mathbf{X}_t^{(c)} \end{cases} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Geometry Alignment: After aligning the first-order statistics, there are two significant challenges that need to be addressed. The first challenge is a damaged intrinsic geometry. The geometric structures underlying the data manifold represent

neighborhood relations among samples, which is crucial when predicting labels based on the local similarity. However, these geometric structures may be damaged when aligning the first-order statistics. The second challenge is indistinguishable category information. In some datasets, for example, PIE, it is difficult to distinguish samples of categories with high similarity. For images in the PIE dataset, the Euclidean distance between samples from one domain but different classes may be smaller than that of samples within one class but from different domains. Then, samples from the same domain but with different labels are easily misclassified as the same class.

In this article, we employ the graph embedding theory [33], [42] to address both the challenges of the damaged intrinsic geometry and indistinguishable category information. We try to achieve the following two goals.

Goal 1: Samples belonging to the same class are close to each other after being projected onto a domain-invariant subspace.

Goal 2: Samples belonging to different classes are far away from each other after being projected onto a domain-invariant subspace.

In the graph embedding theory [33], a graph $\mathbf{G}$ and its corresponding weight matrix $\mathbf{W}$ are usually constructed to represent neighboring relations among samples. Specifically, if $\mathbf{x}_i,\ \mathbf{x}_j \in \mathbf{X}\,(j \neq i)$ belong to the same class, and $\mathbf{x}_i$ is one of the k-nearest neighbors of $\mathbf{x}_j$, then $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected in $\mathbf{G}$, and $\mathbf{W}_{ij} = 1$. Otherwise, $\mathbf{W}_{ij} = 0$. The neighboring relations could be preserved by minimizing

$$\begin{aligned}\sum_{ij} \mathbf{W}_{ij} \left\|\mathbf{P}^T x_i - \mathbf{P}^T x_j\right\|^2 &= \sum_i \mathbf{P}^T x_i \boldsymbol{D}_{ii} x_i^T \mathbf{P} - \sum_{ij} \mathbf{P}^T x_i \mathbf{W}_{ij} x_i^T \mathbf{P} \\ &= \mathrm{Tr}\big(\mathbf{P}^T \mathbf{X}(\mathbf{D}-\mathbf{W})\mathbf{X}^T\mathbf{P}\big) = \mathrm{Tr}\big(\mathbf{P}^T \mathbf{X}\boldsymbol{L}\boldsymbol{X}^T\mathbf{P}\big) \quad (6)\end{aligned}$$

where $\mathbf{D}$ is a diagonal matrix whose diagonal entry is $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$ and $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the Laplacian matrix. In (6), for a certain W, a heavy penalty will be imposed if adjacent samples are projected far away. Therefore, by minimizing (6), the neighborhood relations among samples could be preserved as much as possible. Hall [43] proposed a normalized Laplacian, which can be computed as $\tilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$. Using the normalized Laplacian matrix $\tilde{\mathbf{L}}$ instead of the conventional one $\mathbf{L}$ provides certain theoretical guarantees and seems to perform as well or better in many practical tasks [44].

In this article, to achieve Goal 1 and Goal 2, sample relations represented by (6) are not limited to neighboring relations, it can also represent distant relations. Specifically, we design two graphs, one is the intrinsic graph $\{\mathbf{G}_w,\ \mathbf{W}_w\}$ to preserve neighboring relations, and the other is the penalty graph $\{\mathbf{G}_b,\ \mathbf{W}_b\}$ to preserve distant relations. In graph $\mathbf{G}_w$, if $\mathbf{x}_i,\ \mathbf{x}_j \in \mathbf{X}(j \neq i)$ belong to the same class, and $\mathbf{x}_i$ is one of the k-nearest neighbors of $\mathbf{x}_j$, then $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected. In graph $\mathbf{G}_b$, if $\mathbf{v}_i,\ \mathbf{v}_j \in \mathbf{X}(j \neq i)$ belong to different classes, and the distance between them is one of the k-nearest distances among all the interclass distances, then $\mathbf{v}_i$ and $\mathbf{v}_j$ are connected. If two samples are connected, $\mathbf{W}_{ij} = 1$. Otherwise, $\mathbf{W}_{ij} = 0$. To achieve Goal 1, (6) should be minimized with $\mathbf{L} = \tilde{\mathbf{L}}_w$. To achieve Goal 2, (6) should be maximized with $\mathbf{L} = \tilde{\mathbf{L}}_b$. These two goals could be achieved simultaneously by optimizing the following objective:

$$\begin{aligned}\arg\min_{\mathbf{P}} \quad & \mathrm{Tr}\Big(\mathbf{P}^T \mathbf{X}\tilde{\mathbf{L}}_w \mathbf{X}^T \mathbf{P}\Big) \\ \text{s.t.} \quad & \mathbf{P}^T \mathbf{X}\tilde{\mathbf{L}}_b \mathbf{X}^T \mathbf{P} = \mathbf{I}. \end{aligned} \quad (7)$$

ACE: To obtain the new feature representation, we combine (2), (4), and (7), and obtain the optimization objective. Moreover, following [20], we add a regularization term $\mathrm{Tr}(\mathbf{P}^T\mathbf{P})$ to control the complexity of $\mathbf{P}$, and add a constraint term $\mathbf{P}^T\mathbf{X}\mathbf{H}\mathbf{X}^T\mathbf{P} = \mathbf{I}$ to avoid the trivial solution for $\mathbf{P}$, where $\mathbf{H} = \mathbf{I} - \frac{1}{n}\mathbf{1}$ is the centering matrix, $\mathbf{1} \in \mathbb{R}^{n\times n}$ is the matrix with all 1s, $\mathbf{I} \in \mathbb{R}^{n\times n}$ is the identity matrix, and $n = n_s + n_t$ is the number of all data. Thus, the objective of ACE is

$$\begin{aligned}R_1 = \arg\min_{\mathbf{P}} \quad & \mathrm{Tr}\big(\mathbf{P}^T\mathbf{X}\mathbf{M}\mathbf{X}^T\mathbf{P}\big) + \rho\mathrm{Tr}\Big(\mathbf{P}^T\mathbf{X}\tilde{\mathbf{L}}_u\mathbf{X}^T\mathbf{P}\Big) \\ & + \lambda\mathrm{Tr}\big(\mathbf{P}^T\mathbf{P}\big) \\ \text{s.t.} \quad & \mathbf{P}^T\mathbf{X}\mathbf{H}\mathbf{X}^T\mathbf{P} = \mathbf{I} \end{aligned} \quad (8)$$

where $\mathbf{M} = \sum_{c=0}^{C} \mathbf{M}_c$ is the MMD matrix, and $\tilde{\mathbf{L}}_u = \tilde{\mathbf{L}}_w - \tilde{\mathbf{L}}_b$ is the uniform Laplacian matrix. λ and ρ are two parameters to weigh the importance of different terms.

The Lagrange function of (8) is

$$\begin{aligned}\mathcal{L} = & \mathrm{Tr}\Big(\mathbf{P}^T\Big(\mathbf{X}\mathbf{M}\mathbf{X}^T + \rho\mathbf{X}\tilde{\mathbf{L}}_u\mathbf{X}^T + \lambda\mathbf{I}\Big)\mathbf{P}\Big) \\ & + \mathrm{Tr}\big(\big(\mathbf{P}^T\mathbf{X}\mathbf{H}\mathbf{X}^T\mathbf{P} - \mathbf{I}\big)\Phi\big) \end{aligned} \quad (9)$$

where Φ is the Lagrange multiplier. We let $(\partial\mathcal{L}/\partial\mathbf{P}) = 0$ and obtain

$$\Big(\mathbf{X}\mathbf{M}\mathbf{X}^T + \rho\mathbf{X}\tilde{\mathbf{L}}_u\mathbf{X}^T + \lambda\mathbf{I}\Big)\mathbf{P} = \mathbf{X}\mathbf{H}\mathbf{X}^T\mathbf{P}\Phi \quad (10)$$

where $\Phi = \mathrm{diag}(\phi_1, \ldots, \phi_d)$ contains the d largest eigenvalues, and $\mathbf{P} = [p_1, \ldots, p_d]$ consists of the corresponding eigenvectors, which are just the adaptive components and can be solved through eigendecomposition.

Finally, we obtain the distribution-aligned feature representation $\mathbf{Z}_s \in \mathbb{R}^{d\times n_s}$ and $\mathbf{Z}_t \in \mathbb{R}^{d\times n_t}$ by

$$\mathbf{Z}_s = \mathbf{P}^T\mathbf{X}_s,\ \mathbf{Z}_t = \mathbf{P}^T\mathbf{X}_t. \quad (11)$$

Second-Order Statistics Alignment: Up to now, we have already aligned the first-order statistics of the distributions. At the same time, the geometric structures residing in the data manifold are preserved. However, Sun *et al.* [21] emphasized that aligning the second-order statistics, that is, covariance, is also crucial and can help the classifiers work well on predicting the target-domain data. We compute the covariance matrices of the source and target feature representations and denote them by $\mathbf{C}_s = \mathrm{cov}(\mathbf{Z}_s)$ and $\mathbf{C}_t = \mathrm{cov}(\mathbf{Z}_t)$, respectively, where cov is the covariance. We design a transformation matrix $\mathbf{V} \in \mathbb{R}^{d\times d}$ to match $\mathbf{C}_s$ with $\mathbf{C}_t$. Then, the discrepancy between the covariances is

$$\left\|\tilde{\mathbf{C}}_s - \mathbf{C}_t\right\|_F^2 \quad (12)$$

where $\tilde{\mathbf{C}}_s$ is the matched covariance matrix of the source feature representation and can be written as

$$\tilde{\mathbf{C}}_s = \mathbf{V}^T\mathbf{C}_s\mathbf{V}. \quad (13)$$

Algorithm 1 ACE

Input: Source and target domain data: $\mathbf{X}_s, \mathbf{X}_t$, ground truth labels of source domain:$\mathbf{Y}_s$, pseudolabels of the target domain:$\mathbf{Y}_t$, subspace dimension d, regularization parameters ρ, η and λ

Output: Classification result $\tilde{\mathbf{Y}}_t$

begin

1: Optimize (10) and get the new representation $\mathbf{Z}_s = \mathbf{P}^T\mathbf{X}_s$ and $\mathbf{Z}_t = \mathbf{P}^T\mathbf{X}_t$;
2: Compute the alignment matrix $\mathbf{V}$ through (15), and get $\tilde{\mathbf{Z}}_s = \mathbf{Z}_s\mathbf{V}$;
3: Construct the kernel matrix $\tilde{\mathbf{K}}$ by using $\tilde{\mathbf{Z}}_s$, $\mathbf{Z}_t$;
4: Compute the coefficient β by solving (26);
5: Get classification result $\tilde{\mathbf{Y}}_t$ through (20);

end

Then, the second-order statistics alignment objective is

$$R_2 = \arg\min_{\mathbf{V}} \left\|\mathbf{V}^T\mathbf{C}_s\mathbf{V} - \mathbf{C}_t\right\|_F^2. \tag{14}$$

In practice, optimizing (14) is time consuming and unstable. Alternatively, we follow [21] and compute the transformation matrix $\mathbf{V}$ as

$$\mathbf{V} = (\mathbf{C}_s + \mathbf{I})^{-\frac{1}{2}}(\mathbf{C}_t + \mathbf{I})^{\frac{1}{2}} \tag{15}$$

where $\mathbf{I} \in \mathbb{R}^{d\times d}$ is the identity matrix. Considering the following derivation:

$$\tilde{\mathbf{C}}_s = \mathbf{V}^T\mathbf{C}_s\mathbf{V} = \mathbf{V}^T\mathrm{cov}(\mathbf{Z}_s)\mathbf{V} = \mathrm{cov}\left(\mathbf{V}^T\mathbf{Z}_s\right) \tag{16}$$

we can match $\mathbf{Z}_s$ with $\mathbf{Z}_t$ by

$$\tilde{\mathbf{Z}}_s = \mathbf{V}^T\mathbf{Z}_s. \tag{17}$$

Structural Risk Minimizing: Though the first-order and second-order statistics have been aligned, and in the meantime, the geometric structures are preserved, the difference between domains still exists. With the new feature representation, samples may not be linearly separable. Alternatively, we make a nonlinear transformation to project the new feature representation onto RKHS, whose theory is well developed.

Given a Mercer kernel $K : \mathbf{X}\times\mathbf{X} \to \mathbb{R}$, there exists a related RKHS induced by functions $\mathbf{X} \to \mathbb{R}$ with the corresponding norm $\|\cdot\|_K$. With the new source feature representation $\tilde{\mathbf{Z}}_s = [z_1, \ldots, z_i, \ldots, z_{n_s}]$, we can learn a classifier f in RKHS by minimizing the structural risk functional

$$\arg\min_{f\in\mathcal{H}_K} \quad \sum_{i=1}^{n_s} \ell(f(z_i), y_i) + \eta\|f\|_K^2 \tag{18}$$

where $\mathcal{H}_K$ is the RKHS, $\|f\|_K^2$ is a regularizer, and η is a tradeoff parameter. $\|f\|_K^2$ is added to impose smoothness constraints on possible solutions, which can control the complexity of the classifier in the ambient space.

However, RKHS is an infinite-dimensional, in which it is extremely difficult to learn a classifier. Wahba [45] proposed the representer theorem in RKHS. Using this theorem, the solution to (18) exists in RKHS and can be written as expansions in terms of the training samples. Therefore, (18) becomes

$$f^*(z) = \sum_{i=1}^{n_s} \beta_i K(z_i, z) \tag{19}$$

where $K(\cdot,\cdot)$ is the radial basis function (RBF), and β_i is a coefficient. Then, the infinite-dimensional optimization problem is reduced to learn finite coefficients of the samples. Since (19) only involves the source-domain samples, we instead use an extended representer theorem [46] to exploit the target-domain samples as follows:

$$f^*(z) = \sum_{i=1}^{n} \beta_i K(z_i, z). \tag{20}$$

Using (20), the ambient regularizer for the classifier f in (18) becomes

$$\begin{aligned}\|f\|_K^2 = \langle f, f\rangle_K &= \left\langle \sum_{i=1}^{n}\beta_i K(z_i,\cdot), \sum_{j=1}^{n}\beta_j K\left(z_j,\cdot\right)\right\rangle_K \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}\beta_i^T\beta_j\langle K(z_i,\cdot), K\left(z_j,\cdot\right)\rangle_K. \end{aligned}\tag{21}$$

Considering the reproducing kernel property in RKHS: $\langle K(x,\cdot), K(y,\cdot)\rangle_K = K(x,y)$, (21) can be further derived as

$$\|f\|_K^2 = \sum_{i=1}^{n}\sum_{j=1}^{n}\beta_i^T\beta_j K\left(z_i, z_j\right) = \mathrm{Tr}\left(\beta^T\tilde{\mathbf{K}}\beta\right) \tag{22}$$

where $\tilde{\mathbf{K}} \in \mathbb{R}^{n\times n}$ is the kernel matrix with $\tilde{\mathbf{K}}(i,j) = K(z_i, z_j)$. Next, by selecting different kinds of loss functions, we can obtain different solutions for (18). Here, we use the regularized least squares $(y_i - f(z_i))^2$ as the loss function $\ell(f(z_i), y_i)$ in (18), then we obtain

$$\begin{aligned}&\arg\min_{f\in\mathcal{H}_K} \sum_{i=1}^{n}\ell(f(z_i), y_i) + \eta\|f\|_K^2 \\ &= \arg\min_{f\in\mathcal{H}_K} \sum_{i=1}^{n}\Lambda_{ii}(y_i - f(z_i))^2 + \eta\mathrm{Tr}\left(\beta^T\tilde{\mathbf{K}}\beta\right)\end{aligned}\tag{23}$$

where Λ is a sparse matrix to filter out the unlabeled target samples. It can be written as

$$\Lambda = \begin{pmatrix}\mathbf{I}_{n_s\times n_s} & \mathbf{0}_{n_s\times n_t}\\ \mathbf{0}_{n_t\times n_s} & \mathbf{0}_{n_t\times n_t}\end{pmatrix}. \tag{24}$$

Substituting (20) into (23), we obtain

$$\arg\min_{f\in\mathcal{H}_K} \quad \|\left(\mathbf{Y} - \beta^T\tilde{\mathbf{K}}\right)\Lambda\|_F^2 + \eta\mathrm{Tr}\left(\beta^T\tilde{\mathbf{K}}\beta\right) \tag{25}$$

where $\mathbf{Y} = [\tilde{\mathbf{Y}}_s, \mathbf{0}_{C\times n_t}]$ is an indicator matrix, where $\tilde{\mathbf{Y}}_s \in \mathbb{R}^{C\times n_s}$ are the soft labels for the source domain. If x_s^i belongs to class j, $\tilde{\mathbf{Y}}_s(i,j) = 1$; otherwise, $\tilde{\mathbf{Y}}_s(i,j) = 0$.

Obviously, (25) is convex and differentiable with respect to β. Then, by getting the derivative of (25) to $\mathbf{0}$, β can be computed as

$$\beta = \left(\Lambda\tilde{\mathbf{K}} + \eta\mathbf{I}\right)^{-1}\Lambda\mathbf{Y}^T. \tag{26}$$

It is noteworthy that we are solving the multiclass classification problems, so the classifier coefficient β here is a matrix, whose size is $\mathbb{R}^{n\times C}$ instead of $\mathbb{R}^{n\times 1}$.

Soft Labels Inference: Using the classifier coefficient β and the kernel matrix $\tilde{\mathbf{K}}$, we obtain the prediction for all samples by

$$\tilde{\mathbf{Y}} = \left[\tilde{\mathbf{Y}}_s; \tilde{\mathbf{Y}}_t\right] = \tilde{\mathbf{K}}\beta. \tag{27}$$

In (27), $\tilde{\mathbf{Y}}_t$ are the soft labels for the target samples, whose row entries are the probability values of a sample predicted as different classes.

The proposed method is summarized in Algorithm 1.

C. Time Complexity

In this section, we analyze the time complexity of the proposed method. As stated in the previous sections, m is the dimensionality of the original feature, d is the dimensionality of subspace, where $d \ll m$. n is the number of all the source and target domains. There are two main time-consuming parts as follows.

1) $O(dm^2)$ for solving eigendecomposition problems in step 1 when d nonzero eigenvectors are preserved [47].
2) $O(n^{2.3757})$ for solving matrix inverse and multiplication in step 2–4 by the Coppersmith–Winograd method [48].

In practice, the most time-consuming part is to solve the eigendecomposition problem in step 1. It is worth noting that the time complexity in step 1 is independent of the number of data, which reveals the applicability of our method in the large-scale datasets.

D. Kernelized Feature Alignment

In (8), features are aligned in linear space. For nonlinear problems, ACE could be easily extended to Kernel-ACE using a kernel mapping: $\psi : x \to \psi(x)$. Then, (8) becomes

$$\begin{aligned} R_1 = \arg\min_{\mathbf{P}} \quad & \mathrm{Tr}\left(\mathbf{P}^T\mathbf{K}_f\mathbf{M}\mathbf{K}_f^T\mathbf{P}\right) + \rho\mathrm{Tr}\left(\mathbf{P}^T\mathbf{K}_f\tilde{\mathbf{L}}_u\mathbf{K}_f^T\mathbf{P}\right) \\ & + \lambda\mathrm{Tr}\left(\mathbf{P}^T\mathbf{P}\right) \\ \text{s.t.} \quad & \mathbf{P}^T\mathbf{K}_f\mathbf{H}\mathbf{K}_f^T\mathbf{P} = \mathbf{I} \end{aligned} \tag{28}$$

where $\mathbf{K}_f = \psi(\mathbf{X})^T\psi(\mathbf{X}) \in \mathbb{R}^{n\times n}$ is the kernel matrix of the original features, and $P \in \mathbb{R}^{n\times d}$ is the set of the adaptive components for Kernel-ACE.

E. Theoretical Analysis

As learning the domain-invariant features and training the adaptive classifier are two separate processes, we present the theoretical analysis of them, respectively.

The Bound of the Classifier Error: For a domain adaptation task, one needs to minimize the empirical risk so as to bound the actual risk. However, Vapnik [49] had proved the following theory.

Theorem 1 [49]: With probability at least 1-η simultaneously for all classifiers from the set of totally bounded classifiers $0 \le f(x) \le B$, with finite VC dimension q the following inequality holds true:

$$\epsilon(\beta) \le \epsilon_{\mathrm{emp}}(\beta) + \frac{AG(n)}{2}\left(1 + \sqrt{1 + \frac{4\epsilon_{\mathrm{emp}}(\beta)}{AG(n)}}\right) \tag{29}$$

where n is the number of samples, and $G(n)$ can be computed as

$$G(n) = 4\frac{q\left(ln\frac{2n}{q} + 1\right) - ln\frac{\eta}{4}}{n}. \tag{30}$$

Given Theorem 1, when q/n is small, the actual risk $\epsilon(\beta)$ can be bounded by the empirical risk $\epsilon_{\mathrm{emp}}(\beta)$. Unfortunately, in domain adaptation, generally, the number of samples is very limited. For a fixed VC dimension, q/n is large, then the second term in (29) is also large, a small $\epsilon_{\mathrm{emp}}(\beta)$ cannot guarantee a small $\epsilon(\beta)$. Alternatively, we resort to find a classifier with a small VC dimension q. However, to reduce the empirical risk and to use a classifier with a small VC dimension are contradictive. On the one hand, we need to find a classifier with a small VC dimension q to make $\epsilon_{\mathrm{emp}}(\beta)$ bound $\epsilon(\beta)$. On the other hand, we need to find a classifier in a large range of VC dimensions to minimize the empirical risk. ACE obtains out of this dilemma by employing the SRM learning framework in (18). SRM controls the VC dimension of the classifier by introducing regularization into the empirical risk minimization framework. It achieves a balance between minimizing the empirical risk and reducing the capacity of the classifier to minimize the upper bound of the actual risk [49].

The Bound of the Domain Adaptation Error: Now, we demonstrate the connections between Ben-David's domain adaptation theory [41] and our method.

Theorem 2 (Domain Adaptation Theory [41]): For the hypothesis space H, $\forall h \in H$, we have

$$\epsilon_T(h) \le \epsilon_S(h) + \frac{1}{2}d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t) + \lambda \tag{31}$$

where $\epsilon_S(h)$ and $\epsilon_T(h)$ are the source expected error and the target expected error, respectively, $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t)$ is the $\mathcal{H}\Delta\mathcal{H}$-distance between two domains and λ is the error of the ideal joint hypothesis which can be computed as $\lambda = \min_{h\in H}[\epsilon_S(h) + \epsilon_T(h)]$.

In ACE, $\epsilon_S(h)$ could be explicitly minimized by SRM in (18). $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}_s, \mathcal{D}_t)$ is the discrepancy distance between two distributions D_s and D_t with respect to H. It cound be mininized by the first-order statistics alignment and the second-order statistics alignment. The experimental results in Fig. 3(b) also show that ACE could significantly reduce distribution divergences between domains. λ is used to measure the inadaptability of the source and target domains. When λ is large, one cannot expect to learn a classifier that performs well on both the source and target domains [41]. Usually, λ is considered sufficient low [50], [51]. Otherwise, one should consider choosing a different source domain for adaptation. Long *et al.* [24] derived that geometry alignment in (7) could implicitly minimize λ. Theoretically speaking, ACE could

minimize the upper bound of the expected error on the target domain.

IV. Experiments

In this section, we conduct elaborate experiments on six datasets. Then, we give the parameter sensitivity and effectiveness analysis. Finally, we report the runtime of the state-of-the-art methods.

A. Evaluation Protocol

Following the previous work [20], [22], we use the classification accuracy on target data as the evaluation metric. Specifically, we compute the classification accuracy as $|x : x \in X_t \wedge \hat{y}_t = y_t|/|x : x, \in X_t|$, where x is a sample in target domain, y_t is the real label for x, and $\hat{y}_t$ is the predicted label for x.

B. Compared Baselines

ACE is compared with three kinds of state-of-the-art methods: 1) nontransfer method; 2) traditional methods; and 3) deep methods. The nontransfer method is PCA. The traditional methods contain: GFK [22], TCA [20], JDA [3], TJM [23], CORAL [21], ARTL [24], JGSA [2], and SCA [52]. The deep methods include: AlexNet [53], DDC [5], DAN [4], DANN [31], and DAH [32].

For all the compared baselines, we report results in the original paper or the best we can achieve. For all the methods that need pseudolabels, we uniformly adopt kNN to obtain these pseudolabels. To show the effectiveness of the new feature representations $\tilde{\mathbf{Z}}_s$ and $\mathbf{Z}_t$, we run ACE with the kNN classifier, instead of optimizing SRM. For ACE, there are four hyperparameters to tune: λ, η, d, and ρ. To find the best hyperparameters, we perform the importance-weighted cross-validation (IWCV) method. We fix $\lambda = 10$ and $\eta = 0.5$ in all the tasks. Values of d and ρ are set according to the specific tasks.

C. Datasets

The Office+Caltech [54], [55] dataset includes four different domains, that is, Amazon, Webcam, DSLR, and Caltech. The original images are processed and transformed into several types of features, for example, SURF, DeCAF$_6$, and VGG$_6$. In this experiment, SURF and DeCAF$_6$ are considered. SURF features are encoded with an 800-bin histogram with codebooks trained from a subset of Amazon images [22]. DeCAF$_6$ are activation features of the 6th fully connected layer of a convolutional network constructed by [56]. As for the hyperparameters, we set $d = 21$ and $\rho = 2$ for evaluations on SURF features, and set $d = 12$ and $\rho = 15$ for evaluations on DeCAF$_6$ features.

The CMU PIE [57] dataset includes 41 368 face images from 68 individuals. To conduct face recognition experiments, five domains involving different poses are selected, that is, C05 (left pose), C07 (upward pose), C09 (downward pose), C27 (frontal pose), and C29 (right pose). Hyperparameters for evaluations on PIE datasets are set as $d = 200$ and $\rho = 50$.

TABLE II
Datasets Statistics. sf. and df. Represent SURF and DeCAF$_6$, Respectively

Dataset	Num.	Dim.	Cls.	Domains
Office+Caltech (sf.)	2533	800	10	C,A,W,D
Office+Caltech (df.)	2533	4096	10	C,A,W,D
CMU PIE	11554	1024	68	05,07,09,27,29
ImageNet+VOC2007	10717	4096	5	I,V
MSRC+VOC2007	2799	240	6	M,V
Office-Home	15588	4096	65	Ar,Cl,Pr,Rw
VisDA	207785	2048	12	train,validation

ImageNet+VOC2007 are datasets for visual object recognition challenges. In this benchmark, five common classes of two datasets are selected, that is, person, cat, dog, bird, and chair. We regard each dataset as a domain and thus obtain two evaluations: 1) ImageNet $\rightarrow$ VOC2007 and 2) VOC2007 $\rightarrow$ ImageNet. DeCAF$_6$ features are extracted in these two evaluations. As for the hyperparameters, we set $d = 20$ and $\rho = 50$.

MSRC+VOC2007 have six common classes, that is, sheep, bicycle, car, bird, airplane, and cow. The numbers of images in MSRC and VOC2007 are 1269 and 1530, respectively. To extract features from the raw pixels, images are uniformly rescaled to 256 pixels in length. Then, using the VLFeat open-source package, 128-D dense SIFT (DSIFT) features are extracted. In the last step, K-means clustering is used to obtain the codeworks and thus create a 240-D codebook. Hyperparameters are set as $d = 60$ and $\rho = 5$.

Office-Home [32] includes four domains, that is, Art, Clipart, Product, and Real World. There are about 15 500 images in total. Each domain contains 65 classes. We use a VGG-F model pretrained using the ImageNet2012 to retrieve the 4096-D deep features, which are the fc7 layer output of the model. As for the hyperparameters, we set $d = 200$ and $\rho = 5$.

VisDA [58] is a large-scale dataset which contains three domains and each domain has the common 12 classes. There are 152 397 synthetic images in the training set and 55 388 real images in the validation set. We use the training set as the source domain, and the validation set as the target domain. Thus, there is a synthetic-to-real domain shift between the source and target domains. We use a Resnet-50 model pretrained on ImageNet2012 to extract the 2048-D features. For the hyperparameters, we set $d = 50$ and $\rho = 8$.

For a clear observation, We present dataset statistics on Table II.

D. Experimental Results and Analysis

Experimental Results: We report the experimental results of ACE and the compared baselines in Tables III–IX.

On Office+Caltech with SURF features, ACE outperforms the state-of-the-art methods on most of the tasks. The only two exceptions are A $\rightarrow$ D and D $\rightarrow$ W, on which ACE still obtains the second-best results. On Office+Caltech with DeCAF$_6$ features, the results of all methods have been significantly improved over that on SURF features. Obviously, deep neural networks can extract discriminative features more

TABLE III
ACCURACY (%) ON OFFICE+CALTECH WITH SURF FEATURES. CRA IS SHORT FOR CORAL. ACE* REPRESENTS ACE WITH THE 1NN CLASSIFIER. ACE^k AND JGSA^k REPRESENT THE RESULTS OF THESE METHODS THAT LEARN FEATURES WITH RBF KERNEL

X_s	X_t	PCA	GFK	TCA	CRA	JDA	TJM	SCA	ARTL	JGSA	JGSA^k	ACE*	ACE	ACE^k
C	D	44.6	40.8	45.9	40.7	49.0	44.6	47.1	47.1	45.9	48.4	54.1	58.0	**58.6**
	W	34.6	37.0	39.3	39.2	39.3	39.0	40.0	46.8	45.4	48.5	42.7	**54.6**	54.2
	A	39.5	46.0	45.6	47.2	43.1	46.8	45.6	52.0	51.5	53.1	53.7	**59.6**	58.7
A	D	33.8	40.1	35.7	38.3	42.0	45.2	39.5	46.5	**47.1**	45.2	42.0	45.2	45.2
	W	35.9	37.0	40.0	38.7	38.0	42.0	34.9	41.4	45.8	45.1	46.1	46.8	**50.8**
	C	39.0	40.7	42.0	40.3	40.9	39.5	39.7	42.9	41.5	41.5	45.0	**46.0**	45.8
W	D	89.2	85.4	91.1	84.9	92.4	89.2	87.3	88.5	90.5	88.5	91.1	**93.0**	**93.0**
	A	29.1	27.6	30.5	37.8	29.8	30.0	30.0	39.1	39.9	40.8	37.8	40.3	**40.9**
	C	28.2	24.8	31.5	34.6	33.0	30.2	31.1	32.7	33.2	33.6	34.7	**35.3**	34.9
D	W	86.1	80.3	87.5	85.9	89.2	85.4	84.4	87.1	91.9	**93.2**	91.9	91.2	90.2
	A	33.2	28.7	32.8	38.1	33.4	32.8	31.6	30.8	38.0	38.7	37.8	40.5	**42.8**
	C	29.7	29.3	33.0	34.2	31.2	31.4	30.7	30.9	29.9	30.3	34.7	35.8	**37.8**
Avg.		43.6	43.1	46.2	46.7	46.8	46.3	45.2	48.8	50.0	50.6	51.0	53.9	**54.4**

TABLE IV
ACCURACY (%) ON OFFICE+CALTECH WITH DECAF$_6$ FEATURES. CRA IS SHORT FOR CORAL. ACE* REPRESENTS ACE WITH THE 1NN CLASSIFIER. ACE^k AND JGSA^k REPRESENT THE RESULTS OF THESE METHODS THAT LEARN FEATURES WITH RBF KERNEL

X_s	X_t	Traditional Methods								Deep Methods			ACE*	ACE	ACE^k
		GFK	TCA	CRA	JDA	SCA	ARTL	JGSA	JGSA^k	AlexNet	DAN	DDC			
C	D	86.6	85.4	84.7	89.8	87.9	86.6	93.6	92.4	87.1	89.3	88.8	88.5	92.4	**94.3**
	W	77.6	78.3	80.0	85.1	85.4	87.8	86.8	83.4	83.7	90.6	85.4	89.5	94.2	**94.6**
	A	88.2	89.8	92.0	89.6	89.5	92.4	91.4	91.1	91.9	92.0	91.9	92.1	92.9	**93.5**
A	D	82.2	81.5	84.1	80.3	85.4	85.4	88.5	84.7	87.4	91.7	89.0	86.0	**94.3**	93.6
	W	70.9	74.2	74.6	78.3	75.9	88.5	81.0	80.0	79.5	**91.8**	86.1	87.8	**91.8**	90.2
	C	79.2	82.6	83.2	83.6	78.8	87.4	84.9	84.9	83.0	84.1	85.0	84.5	**87.5**	87.3
W	D	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100**	**100.0**	**100.0**	**100.0**
	A	76.8	84.1	81.2	90.3	86.1	92.3	90.7	91.3	83.8	92.1	84.9	92.3	**92.8**	92.4
	C	69.8	80.4	75.5	84.8	74.8	**88.2**	85.0	84.5	73.0	81.2	78.0	84.5	86.3	86.3
D	W	99.3	99.7	99.3	99.7	98.6	**100.0**	99.7	98.6	97.7	98.5	98.2	**100.0**	**100.0**	**100.0**
	A	76.3	89.1	85.5	91.7	90.0	92.7	92.0	92.0	87.1	90.0	89.5	91.9	91.9	**93.1**
	C	71.4	82.3	76.8	85.5	78.1	87.3	86.2	84.8	79.0	80.3	81.1	86.3	87.6	**87.9**
Avg.		81.5	85.6	84.7	88.2	85.9	90.7	90.0	89.0	86.1	90.1	88.2	90.3	92.6	**92.8**

TABLE V
ACCURACY (%) ON PIE. ACE* REPRESENTS ACE WITH THE 1NN CLASSIFIER

X_s	X_t	GFK	JDA	TJM	JGSA	ARTL	ACE*	ACE
C05	C07	45.9	58.8	46.4	62.8	60.5	**68.8**	67.2
	C09	49.7	54.2	53.6	59.9	62.5	62.4	**66.2**
	C27	65.5	84.5	75.1	82.9	83.6	84.9	**86.2**
	C29	41.5	49.8	45.9	57.3	**55.2**	54.4	50.4
C07	C05	46.5	57.6	52.0	68.0	**67.4**	66.8	66.7
	C09	56.6	62.9	56.7	69.4	71.4	72.7	**73.8**
	C27	70.6	75.8	73.1	85.9	85.5	85.5	**90.4**
	C29	41.0	39.9	40.4	58.0	54.2	52.0	**56.3**
C09	C05	48.4	51.0	53.8	69.0	66.6	64.5	**67.9**
	C07	55.9	58.0	50.3	69.6	67.6	69.4	**72.0**
	C27	74.7	68.5	75.9	80.1	86.7	87.9	**89.8**
	C29	49.8	40.0	46.8	68.0	65.1	62.3	**65.9**
C27	C05	70.9	80.6	76.1	83.0	89.5	88.2	**90.1**
	C07	80.9	82.6	78.0	85.1	90.0	91.7	**93.5**
	C09	86.4	87.3	84.3	79.7	91.3	92.6	**93.2**
	C29	58.9	54.7	56.3	70.5	75.9	77.3	**80.1**
C29	C05	39.3	46.5	44.3	63.2	**63.4**	55.3	57.4
	C07	38.1	42.1	37.9	60.9	51.5	54.5	**57.0**
	C09	48.9	53.3	44.8	64.3	66.3	62.1	**67.4**
	C27	54.3	57.0	55.9	72.7	74.4	70.7	**79.4**
Avg.		56.2	60.2	57.4	70.5	71.4	71.2	**73.5**

TABLE VI
ACCURACY (%) ON IMAGENET+VOC2007. V AND I REPRESENT VOC2007 AND IMAGENET, RESPECTIVELY. CRA IS SHORT FOR CORAL. ACE* REPRESENTS ACE WITH THE 1NN CLASSIFIER

X_s	X_t	TCA	JDA	CRA	ARTL	JGSA	ACE*	ACE
I	V	63.7	63.4	59.6	62.4	52.3	65.2	**69.1**
V	I	64.9	70.2	70.3	72.2	70.6	76.8	**83.4**
Avg.		64.3	66.8	65.0	67.3	61.5	71.0	**76.3**

TABLE VII
ACCURACY (%) ON MSRC+VOC2007. M AND V REPRESENT MSRC AND VOC2007, RESPECTIVELY. ACE* REPRESENTS ACE WITH THE 1NN CLASSIFIER

X_s	X_t	GFK	TJM	SCA	JGSA	ARTL	ACE*	ACE
M	V	28.8	32.5	32.8	33.8	37.1	32.2	**38.3**
V	M	48.9	46.3	48.9	48.2	55.2	54.3	**64.7**
Avg.		38.8	39.4	40.8	41.1	46.1	43.3	**51.5**

effectively than the traditional methods. Even though, ACE is still the best of all the methods. Office-Home is another dataset with deep features. On this dataset, ACE obtains a significant advantage over the best traditional methods ARTL. Even compared with the best deep method DAH, ACE still achieves a 0.8% improvement. On the VisDA dataset, features used by ACE and other traditional methods are extracted by ResNet-50, which is shallower than ResNet-101 utilized by deep methods. Even though, ACE still obtains the best performance across all the traditional and deep methods. Performance of ACE on ImageNet+VOC2007,

TABLE VIII
ACCURACY (%) ON OFFICE-HOME. AR, CL, PR, AND RW REPRESENT ART, CLIPART, PRODUCT, AND REAL-WORLD, RESPECTIVELY. CRA IS SHORT FOR CORAL

X_s	X_t	Traditional Methods						Deep Methods			ACE
		GFK	TCA	CRA	JDA	JGSA	ARTL	DAN	DANN	DAH	
Ar	Cl	21.6	19.9	27.1	25.3	25.5	30.5	30.7	**33.3**	31.6	32.6
	Pr	31.7	32.1	36.2	36.0	38.2	40.5	42.2	43.0	40.8	**49.2**
	Rw	38.8	35.7	44.3	42.9	44.1	48.5	54.1	54.4	51.7	**54.9**
Cl	Ar	21.6	19.0	26.1	24.5	28.0	35.3	32.8	32.3	34.7	**36.8**
	Pr	34.9	31.4	40.0	40.2	38.7	47.7	47.6	49.1	**51.9**	51.1
	Rw	34.2	31.7	40.3	40.9	40.2	48.1	49.8	49.8	**52.8**	51.2
Pr	Ar	24.5	21.9	27.8	26.0	29.2	34.6	29.1	30.5	29.9	**35.7**
	Cl	25.7	23.6	30.5	32.7	28.1	34.0	34.1	38.1	**39.6**	32.7
	Rw	42.9	42.1	50.6	49.3	50.6	56.2	56.7	56.8	**60.7**	58.3
Rw	Ar	32.9	30.7	38.5	35.1	40.0	47.1	43.6	44.7	45.0	**47.4**
	Cl	29.0	27.2	36.4	35.4	36.5	40.5	38.3	**42.7**	45.1	39.5
	Pr	50.9	48.7	57.1	55.4	57.0	63.8	62.7	64.7	62.5	**65.9**
Avg.		32.4	30.3	37.9	37.0	38.0	43.9	43.5	44.9	45.5	**46.3**

TABLE IX
ACCURACY (%) ON VISDA. (50) AND (101) REPRESENT DEEP FEATURES EXTRACTED BY RESNET-50 AND RESNET-101 NETWORKS, RESPECTIVELY

Methods	aero.	bike	bus	car	horse	knife	moto.	person	plant	sktbrd	train	truck	Avg.
GFK (50)	77.8	58.7	60.7	42.0	88.3	44.2	80.0	60.3	72.5	42.1	54.2	19.3	58.3
TCA (50)	87.1	65.8	62.5	51.4	92.3	28.7	78.2	**77.8**	86.5	36.2	72.0	**27.8**	63.9
JDA (50)	85.3	46.5	55.0	56.3	88.7	**45.1**	85.4	77.5	68.0	33.5	70.1	19.3	60.9
JGSA (50)	**91.1**	63.8	64.2	47.5	92.6	42.3	79.8	76.5	84.3	50.1	72.2	26.5	65.8
ResNet (50)	67.7	36.6	48.4	**68.2**	76.9	5.3	65.8	38.0	72.5	29.1	82.1	3.7	49.5
ResNet (101)	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
DANN (101)	81.9	77.7	**82.8**	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
DAN (101)	87.1	63.0	76.5	42.0	90.3	42.9	**85.9**	53.1	49.7	36.3	85.8	20.7	61.1
ACE (50)	86.7	**69.2**	68.3	59.7	**94.3**	38.9	85.5	73.6	**91.0**	**54.7**	**86.2**	18.6	**68.9**

MSRC+VOC2007, and PIE is also remarkable, which further verifies the effectiveness of ACE on these benchmarks.

To show that the experimental results are statistically significant, we select four different tasks and perform McNemar's test with the results of ACE and ARTL. Test results are reported in Table X. All the *p*-values are less than 0.05. Then, we have great confidence that ACE can perform better than ARTL. Therefore, the experimental results are statistically significant.

We report the results of ACE based on the 1NN classifier in Tables III–VII. ACE with the 1NN classifier performs worse than ACE with optimizing SRM on all the datasets, which verifies that SRM could separate features better than the 1NN classifier. In addition, ACE with the 1NN classifier still outperforms JGSA. Note the results of JGSA are also based on the 1NN classifier.

We report the results of the kernelized JGSA and ACE in Tables III and IV. The results reveal that the kernelized methods can improve performance on some tasks.

Discussions: Based on the above experimental results, we can obtain the following observations.

1) When the distribution difference between domains is not significant, almost all the baseline methods can achieve good performance, for example, W→ D (SURF), Rw→ Pr, and C27→ C09. However, when the evaluation tasks are challenging, for example, C→ D (SURF), Ar→ Pr, and V→M, most of the baseline methods cannot work well, only ACE still obtains appealing performance. TCA [20], TJM [23], SCA [52], and JDA [3] try to find a latent subspace where the domain distributions are aligned and some important properties are preserved. Unfortunately, when the discrepancies between domains are substantially large, this common subspace may not exist, which leads to performance degradation of these methods. As for the other methods, CORAL [21] only aligns the covariance of the source and target domains. GFK [22] constructs a geodesic flow that can connect the source and target domains in a Grassmann manifold, which lacks the process of distribution alignment. JGSA [2] attempts to align the domain distributions and maximumly preserve the statistical properties simultaneously, and thus achieves a second-best average accuracy in total. However, all of the above baselines do not consider preserving the geometric structures underlying the manifold when projecting data onto a latent subspace.
2) In all the compared baselines, ARTL [24] is the only method that uses the graph Laplacian to regularize the

TABLE X
SIGNIFICANT TEST. sf. AND df. REPRESENT SURF AND DECAF$_6$, RESPECTIVELY

Tasks	W→C (sf.)	C→A (df.)	C27→C29	Cl→Ar
p-values	0.0056	0.0318	3.8×10^{-21}	0.0018

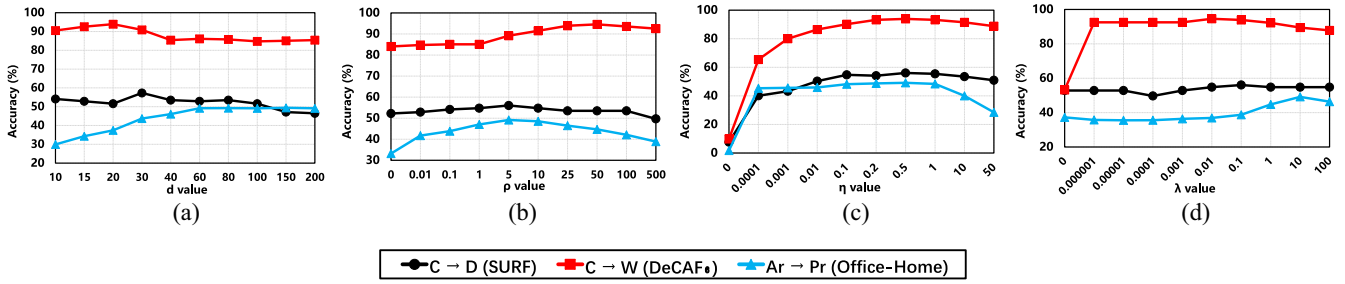


Fig. 2. Parameter sensitivity. Four hyperparameters d, ρ, η, and λ are evaluated on C→D (SURF), C→W (DeCAF$_6$), and Ar→Pr (Office-Home).

TABLE XI
RUNTIME PERFORMANCE (SECOND). sf. AND df. REPRESENT SURF AND DECAF$_6$, RESPECTIVELY

Runtime	C→A(sf.)	C→W(df.)	C05→C27	Ar→Cl
ACE	1.88	7.19	25.65	46.68
JGSA	2.12	27.98	23.16	87.04
ARTL	1.60	1.70	35.49	63.22
CORAL	0.55	29.13	5.62	53.15
GFK	0.80	10.28	2.89	72.24
JDA	0.62	6.15	3.99	18.00
TCA	2.11	1.23	28.66	35.78

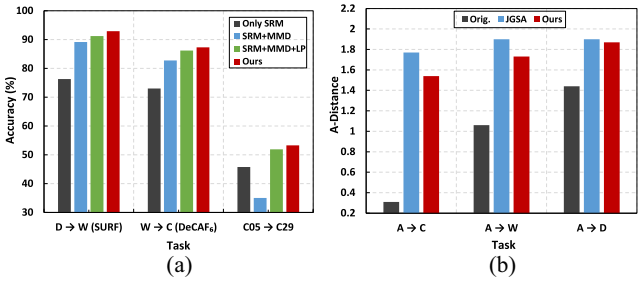


Fig. 3. Effectiveness analysis. (a) Effectiveness of every strategy in promoting the performance. (b) $\mathcal{A}$-distance of the original DeCAF$_6$ features, JGSA features, and ours.

manifold and thus preserve the geometric structures. However, ARTL does not construct the penalty graph to enhance the discriminative ability of the feature representation. Besides, ARTL does not align the second-order statistics to further reduce the distribution discrepancies. ACE exceeds all the compared methods with a significant advantage in virtue of jointly optimizing the distribution alignment and geometric structures preservation.

3) In deep methods, processes of training deep neural networks and performing knowledge transfer are carried out simultaneously. Compared with most of the traditional methods, this joint optimizing strategy could significantly improve performance. Our method, instead, extracts the deep features through a pretrained neural networks first. Then, these extracted features are utilized to conduct knowledge transfer. According to the experimental results on several datasets, for example, Office-Home and Office+Caltech (DeCAF$_6$), our ACE could achieve better performance than the deep domain adaptation methods, that is, DCC [5], DAN [4], and DAH [32], which reveals that this separate optimizing strategy could also transfer knowledge across domains. Using the extracted deep features, our proposed method could perform well, which mainly derives from our distribution alignment and the geometric structure preservation strategy to significantly reduce the domain discrepancies. The most significant advantage of our method is that, this separate optimizing strategy is very flexible, for we can perform the alignment operation to reduce the disparities of some properties (statistical properties or geometrical properties) between domains. From the experimental results, we observe that the knowledge transfer part is of great importance to an effective yet efficient domain adaptation. In contrast, the deep methods could not easily align these properties when training the deep neural networks. Furthermore, our method does not need to fine-tune plenty of hyperparameters, which substantially reduces the training time.

E. Runtime Analysis

We report the average runtime (training) of seven methods on all benchmarks in Table XI. All of the test codes are executed in the same software and hardware environment. For the iterative methods, both JDA and JGSA iterate only one time. For fairness, we set the subspace dimensions of all subspace learning methods as 30. For each result in Table XI, we run the algorithms ten times and report the means. From Table XI, we can obtain the following observations.

1) Compared with the complex methods, that is, JGSA, ACE not only saves plenty of time but also obtains the higher performance.
2) Compared with the simple methods, for example, GFK and CORAL, ACE achieves a remarkable improvement at the cost of a little more running time. Therefore, ACE is a balanced method and has a strong application value in the real world.

F. Parameter Sensitivity

We carry out the parameter sensitivity analysis to verify that ACE can perform pretty well in a wide range of hyperparameters. The evaluated hyperparameters are d, η, ρ, and λ. Specifically, we first choose three evaluations from three different datasets: 1) C→D (SURF); 2) C→W (DeCAF$_6$); and 3) Ar→Cl (Office-Home). Then, for every hyperparameter, the

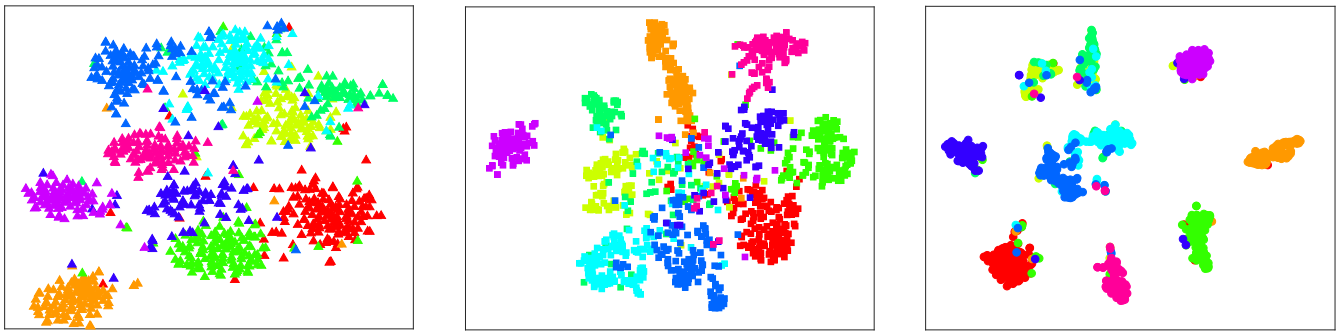


Fig. 4. Feature analysis. *t*-SNE of features extracted by different methods on A → C (DeCAF_6) task. (a) Original features. (b) JGSA. (c) Ours.

selected three evaluations are run under ten different values. When one hyperparameter is being evaluated, other hyperparameters are set the same as that in Section IV-C. Experimental results are shown in Fig. 2. In Fig. 2(a), for evaluations in the Office+Caltech dataset, that is, C→D (SURF) and C→W (DeCAF_6), accuracies vary in a very small range with different values of d. For evaluations in the Office-Home dataset, that is, Ar→Cl, performance will degrade if d is too small. Since features of different datasets have different intrinsic dimensions, correspondingly different subspace dimensions d need to be selected. In Fig. 2(b), the best value of ρ varies in different evaluations. For DeCAF_6, the optimal ρ appears between 25 and 500. For SURF and Office-Home, ρ should be chosen between 1 and 10. In Fig. 2(c), when $\eta \to 0$, all of the accuracies reduce dramatically. When $\eta > 1$, accuracies also reduce. Therefore, the optimal range of η is [0.1, 1]. In Fig. 2(d), when $\lambda \to 0$, accuracies are unsatisfactory. λ influences performance little when $\lambda > 1e-6$. When $\lambda > 0.01$, evaluations on all of the datasets can obtain appealing performance.

G. Effectiveness Analysis

Effectiveness Verification: To verify the contributions of different parts in our method, we conduct experiments on four different settings: 1) method that only optimizes SRM (only SRM); 2) method that optimizes both SRM and MMD (SRM + MMD); 3) method that in addition optimizes graph Laplacian based on method in 2) (SRM + MMD + LP); and 4) our method. The experimental results are reported in Fig. 3(a). We can make the following observations.

1) Performance is improved significantly after optimizing MMD.
2) The result of method 2 (SRM + MMD) in C05 → C29 task is inferior to the result of other methods, even compared with the result of method 1 (only SRM), which reveals that only optimizing MMD may damage the geometric structures and thus reduce the performance.
3) Our method outperforms method 3 (SRM + MMD + LP) in virtue of the alignment of the second-order statistics.

Feature Analysis: We compare features extracted by different methods in Figs. 3(b) and 4.

We first show the domain discrepancy of DeCAF_6 tasks in Fig. 3(b). Following Ben-David *et al.* [41], we use the $\mathcal{A}$-distance as a metric to measure the domain discrepancy. As it is difficult to compute the exact $\mathcal{A}$-distance, we instead compute the empirical $\mathcal{A}$-distance as $2(1-2\epsilon)$, where ϵ is the generalization error of a basic classifier (SVM in our experiment) to distinguish samples between the source and target domains. From Fig. 3(b), we surprisingly observe that $\mathcal{A}$-distance of the original DeCAF_6 features is far less than that of JGSA and our method.

In Fig. 4(a)–(c), we plot t-SNE of the original DeCAF_6 features, JGSA features, and our features. It is obvious that features extracted by our methods can be clearly classified into ten different groups, which further verify that our method can extract discriminative features. As a comparison, features extracted by JGSA are not distinguishable enough to classify different groups.

V. Conclusion

In this article, we propose a novel method for unsupervised domain adaptation, referred to as ACE. ACE learns a set of adaptive components to embed the source and target domains into a domain-invariant subspace, and thus obtains the new feature representation. ACE aligns the first- and second-order statistics, that is, mean and covariance, to reduce the domain discrepancies. At the same time, the geometric structures residing in the data manifold are well preserved. Comprehensive experiments on six datasets validate the effectiveness of ACE compared with several state-of-the-art methods.

Acknowledgment

The authors sincerely thank the editors and the reviewers for their valuable comments and suggestions.

References

[1] X. Tang and X. Wang, "Face photo recognition using sketch," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, 2002, p. 1.
[2] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 5150–5158.
[3] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 2200–2207.
[4] M. Long, Y. Cao, Z. Cao, J. Wang, and M. I. Jordan, "Transferable representation learning with deep adaptation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 12, pp. 3071–3085, Dec. 2019.
[5] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014. [Online]. Available: arXiv:1412.3474.
[6] M. Uzair and A. Mian, "Blind domain adaptation with augmented extreme learning machine features," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 651–660, Mar. 2017.

[7] S. Khalighi, B. Ribeiro, and U. J. Nunes, "Importance weighted import vector machine for unsupervised domain adaptation," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3280–3292, Oct. 2017.

[8] Y. Chen, S. Song, S. Li, L. Yang, and C. Wu, "Domain space transfer extreme learning machine for domain adaptation," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1909–1922, May 2019.

[9] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, "Transfer independently together: A generalized framework for domain adaptation," *IEEE Trans. Cybern.*, vol. 49, no. 6, pp. 2144–2155, Jun. 2019.

[10] J. Li, M. Jing, K. Lu, L. Zhu, and H. T. Shen, "Locality preserving joint transfer for domain adaptation," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 6103–6115, Jun. 2019.

[11] J. Li, K. Lu, Z. Huang, L. Zhu, and H. T. Shen, "Heterogeneous domain adaptation through progressive alignment," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1381–1391, May 2018.

[12] J. Li, Y. Wu, J. Zhao, and K. Lu, "Low-rank discriminant embedding for multiview learning," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3516–3529, Nov. 2017.

[13] M. Jing, J. Li, J. Zhao, and K. Lu, "Learning distribution-matched landmarks for unsupervised domain adaptation," in *Database Systems for Advanced Applications*. Cham, Switzerland: Springer, 2018, pp. 491–508.

[14] W. Dai, Y. Chen, G.-R. Xue, Q. Yang, and Y. Yu, "Translated learning: Transfer learning across different feature spaces," in *Proc. Neural Inf. Process. Syst.*, 2009, pp. 353–360.

[15] P. Prettenhofer and B. Stein, "Cross-language text classification using structural correspondence learning," in *Proc. Annu. Meeting Assoc. Comput. Linguist.*, 2010, pp. 1118–1127.

[16] M. Chen, K. Q. Weinberger, and J. Blitzer, "Co-training for domain adaptation," in *Proc. Neural Inf. Process. Syst.*, 2011, pp. 2456–2464.

[17] V. Sindhwani, P. Niyogi, and M. Belkin, "A co-regularization approach to semi-supervised learning with multiple views," in *Proc. ICML*, 2005, pp. 74–79.

[18] J. Li, K. Lu, Z. Huang, and H. T. Shen, "On both cold-start and long-tail recommendation with social data," *IEEE Trans. Knowl. Data Eng.*, early access, doi: 10.1109/TKDE.2019.2924656.

[19] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

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

[21] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. AAAI*, vol. 6, 2016, p. 8.

[22] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2012, pp. 2066–2073.

[23] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, 2014, pp. 1410–1417.

[24] M. Long, J. Wang, G. Ding, S. J. Pan, and P. S. Yu, "Adaptation regularization: A general framework for transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 5, pp. 1076–1089, May 2014.

[25] M. Jiang, W. Huang, Z. Huang, and G. G. Yen, "Integration of global and local metrics for domain adaptation learning via dimensionality reduction," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 38–51, Jan. 2017.

[26] C.-X. Ren, X.-L. Xu, and H. Yan, "Generalized conditional domain adaptation: A causal perspective with low-rank translators," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 821–834, Feb. 2020.

[27] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A DASVM classification technique and a circular validation strategy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 770–787, May 2010.

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

[29] G. Schweikert, G. Rätsch, C. Widmer, and B. Schölkopf, "An empirical analysis of domain adaptation algorithms for genomic sequence analysis," in *Proc. Neural Inf. Process. Syst.*, 2009, pp. 1433–1440.

[30] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Proc. ECCV Workshops*, 2016, pp. 443–450.

[31] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096–3030, 2016.

[32] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 5018–5027.

[33] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph embedding and extensions: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 40–51, Jan. 2007.

[34] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2242–2251.

[35] J. Hoffman *et al.*, "CYCADA: Cycle consistent adversarial domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1994–2003.

[36] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 700–708.

[37] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proc. ECCV*, 2018, pp. 172–189.

[38] R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. ACM Int. Conf. Mach. Learn.*, 2007, pp. 759–766.

[39] H. Wang, F. Nie, and H. Huang, "Robust and discriminative self-taught learning," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 298–306.

[40] S. N. Tran and A. D. Garcez, "Adaptive transferred-profile likelihood learning," in *Proc. IEEE IJCNN*, 2016, pp. 2687–2692.

[41] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, nos. 1–2, pp. 151–175, 2010.

[42] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Neural Inf. Process. Syst.*, 2002, pp. 585–591.

[43] F. J. Hall, "The adjacency matrix, standard Laplacian, and normalized Laplacian, and some eigenvalue interlacing results," vol. 16, Dept. Math. Stat., Georgia State Univ., Atlanta, GA, USA, 2010.

[44] U. Von Luxburg, M. Belkin, and O. Bousquet, "Consistency of spectral clustering," *Ann. Stat.*, vol. 36, no. 2, pp. 555–586, 2008.

[45] G. Wahba, *Spline Models for Observational Data*, vol. 59. Philadelphia, PA, USA: SIAM, 1990.

[46] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, Nov. 2006.

[47] D. C. Sorensen, "Implicitly restarted Arnoldi/Lanczos methods for large scale eigenvalue calculations," in *Parallel Numerical Algorithms*. Dordrecht, The Netherlands: Springer, 1997, pp. 119–165.

[48] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," in *Proc. ACM STOC*, 1987, pp. 1–6.

[49] V. N. Vapnik, *Statistical Learning Theory*. Hoboken, NJ, USA: Wiley-Intersci., 1998.

[50] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. Comput. Vis. Pattern Recognit.*, 2018, pp. 3723–3732.

[51] M. Long, C. Yue, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 97–105.

[52] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, "Scatter component analysis: A unified framework for domain adaptation and domain generalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1414–1430, Aug. 2017.

[53] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[54] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. ECCV*, 2010, pp. 213–226.

[55] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Rep., 2007. [Online]. Available: https://authors.library.caltech.edu/7694/

[56] J. Donahue *et al.*, "DECAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 647–655.

[57] T. Sim, S. Baker, and M. Bsat, "The CMU pose, illumination, and expression (PIE) database," in *Proc. IEEE FG*, 2002, pp. 53–58.

[58] X. Peng *et al.*, "VISDA: A synthetic-to-real benchmark for visual domain adaptation," in *Proc. CVPR Workshops*, 2018, pp. 2021–2026.

Mengmeng Jing received the B.Eng. and M.Sc. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 2015 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering.

His current research interests include machine learning, especially transfer learning, subspace learning, and recommender systems.

Jidong Zhao received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 1999, 2003, and 2006, respectively.

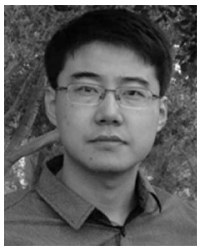
He is currently a Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China. His current research interests include pattern recognition, multimedia, and computer vision.

Jingjing Li received the M.Sc. and Ph.D. degrees in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2013 and 2017, respectively.

He is currently a National Postdoctoral Program for Innovative Talents Research Fellow with the School of Computer Science and Engineering, University of Electronic Science and Technology of China. He has a great interest in machine learning, especially transfer learning, subspace learning, and recommender systems.

Lei Zhu received the B.Sc. degree from the Wuhan University of Technology, Wuhan, China, in 2009, and the Ph.D. degree from the Huazhong University of Science and Technology, Wuhan, in 2015.

He is currently a Full Professor with the School of Information Science and Engineering, Shandong Normal University, Jinan, China. He was a Research Fellow with the University of Queensland, Brisbane, QLD, Australia, from 2016 to 2017, under the supervision of Prof. H. T. Shen. He was with the Singapore Management University, Singapore, from 2015 to 2016. His current research interests include large-scale multimedia content analysis and retrieval.



Yang Yang received the bachelor's degree in computer science from Jilin University, Changchun, China, in 2006, the master's degree in computer science from Peking University, Beijing, China, in 2009, and the Ph.D. degree in computer science from the University of Queensland, Brisbane, QLD, Australia, in 2012, under the supervision of Prof. H. T. Shen and Prof. X. Zhou.

He is currently with the University of Electronic Science and Technology of China, Chengdu, China. He was a Research Fellow with the National University of Singapore, Singapore, from 2012 to 2014, under the supervision of Prof. T.-S. Chua. His current research interests include multimedia content analysis, computer vision, and social media analytics.

Heng Tao Shen received the B.Sc. (Hons.) and Ph.D. degrees from the Department of Computer Science, National University of Singapore, Singapore, in 2000 and 2004, respectively.

He is currently a Professor of the National Thousand Talents Plan, the Dean of the School of Computer Science and Engineering, and the Director of the Center for Future Media, University of Electronic Science and Technology of China, Chengdu, China. He is also an Honorary Professor with the University of Queensland, Brisbane, QLD, Australia, where he joined as a Lecturer, a Senior Lecturer, a Reader, and became a Professor in late 2011. He has published over 200 peer-reviewed papers, most of which appeared in top-ranked publication venues, such as ACM Multimedia, CVPR, ICCV, AAAI, IJCAI, SIGMOD, VLDB, ICDE, ACM TIS, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and *VLDB Journal*. His current research interests include multimedia search, computer vision, artificial intelligence, and big data management.

Prof. Shen was a recipient of six best paper awards from international conferences, including the Best Paper Award from ACM Multimedia 2017 and the Best Paper Award Honorable Mention from ACM SIGIR 2017. He is currently an Associate Editor of the IEEE TRANSACTION KNOWLEDGE AND DATA ENGINEERING. He has served as a PC Co-Chair for ACM Multimedia 2015.