Cross-Domain Pattern Classification With Distribution Adaptation Based on Evidence Theory

Lin-Qing Huang, Zhun-Ga Liu, Jean Dezert

Cross-Domain Pattern Classification With Distribution Adaptation Based on Evidence Theory

Lin-Qing Huang, Zhun-Ga Liu, *Member, IEEE*, and Jean Dezert

***Abstract*—In pattern classification, there may not exist labeled patterns in the target domain to train a classifier. Domain adaptation (DA) techniques can transfer the knowledge from the source domain with massive labeled patterns to the target domain for learning a classification model. In practice, some objects in the target domain are easily classified by this classification model, and these objects usually can provide more or less useful information for classifying the other objects in the target domain. So a new method called distribution adaptation based on evidence theory (DAET) is proposed to improve the classification accuracy by combining the complementary information derived from both the source and target domains. In DAET, the objects that are easy to classify are first selected as easy-target objects, and the other objects are regarded as hard-target objects. For each hard-target object, we can obtain one classification result with the assistance of massive labeled patterns in the source domain, and another classification result can be acquired based on the easy-target objects with confidently predicted (pseudo) labels. However, the weights of these classification results may vary because the reliabilities of the used information sources are different. The weights are estimated by mean difference reflecting the information source quality. Then, we discount the classification results with the corresponding weights under the framework of the evidence theory, which is expert at dealing with uncertain information. These discounted classification results are combined by an evidential combination rule for making the final class decision. The effectiveness of DAET for cross-domain pattern classification is evaluated with respect to some advanced DA methods, and the experiment results show DAET can significantly improve the classification accuracy.**

***Index Terms*—Belief functions (BFs), cross-domain pattern classification, evidence theory, information fusion, transfer learning.**

I. Introduction

IN PATTERN classification, a common assumption is that the training and test data are drawn from the same distribution, that is, they satisfy the condition of independence and identical distribution (i.i.d.). Unfortunately, this assumption usually cannot be satisfied in many applications, and the standard machine learning methods are unable to work well when this assumption is not valid. For example, we can collect rich labeled data in an existing domain (called the source domain), and there are no labeled patterns in a new domain (called the target domain). The distributions of patterns in the source and target domains are not close to each other, so the standard classification models cannot be directly employed. The classification performance will be poor if one directly uses the classifier learnt by labeled patterns in the source domains to classify unseen objects in the target domain. It is called the domain adaptation (DA) problem, and the major issue is how to effectively reduce the distribution discrepancy between the source and target domains for building a more reliable classification model. In recent years, many DA techniques [1]–[3] have been proposed to solve this issue. These methods can be divided into two main categories: 1) instance-based methods [4]–[7] and 2) feature-based methods [8]–[17].

For instance-based methods, the patterns in the source domain are reused with appropriate weights to reduce the distribution difference for learning a classification model to classify unseen objects. The TrAdaBoost method proposed by Dai *et al.* [4] decreased the weights if the patterns in the source domain are not correctly classified, and increases them if the patterns in the target domain are committed into wrong classes. The Kernel mean match (KMM) [5] method directly produces the resampling weights. It accounts for the difference by reweighting the training points such that the means of the training and test data in a reproducing Kernel Hilbert space (RKHS) are drawn sufficiently close in some mathematical sense (i.e., for a chosen distance metric). Sugiyama *et al.* [6] proposed a Kullback–Leibler importance estimation procedure (KLIEP) method, which consists of patterns importance estimation and a natural model selection procedure based on Kullback–Leibler divergence. The metric transfer learning framework (MTLF) [7] proposes to simultaneously learn the instance weights and the Mahalanobis distance to maximize intraclass distances, and to minimize interclass distances.

The principle of featured-based methods is to discover a new feature representation of patterns in the source and target domains for reducing the distribution discrepancy. The source and target data are transformed into a new feature space to make the distributions closer, and the standard machine learning models can be successfully employed for these patterns in this new feature space. Pan *et al.* [8] proposed a transfer component analysis (TCA) method to learn the transformation

Manuscript received March 23, 2021; revised July 27, 2021 and October 19, 2021; accepted November 26, 2021. This work was supported in part by National Natural Science Foundation of China under Grant U20B2067, Grant 61790552, and Grant 61790554; and in part by the Innovation Foundation for Doctor Dissertation of Northwestern Polytechnical University under Grant 06090-21GH010201. This article was recommended by Associate Editor Y. Xia. *(Corresponding author: Zhun-Ga Liu.)*

Lin-Qing Huang and Zhun-Ga Liu are with the School of Automation, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: huanglinqing95@gmail.com; liuzhunga@nwpu.edu.cn).

Jean Dezert is with the DTIS Department, ONERA-The French Aerospace Lab, 91761 Palaiseau, France (e-mail: jean.dezert@onera.fr).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2021.3133890.

matrix by minimizing the marginal distribution discrepancy. The stratified transfer learning (STL) [9] method aims to reduce the conditional distribution discrepancy between the source and target domains. Long *et al.* [10] presented a joint distribution adaptation (JDA) method to discover a new feature representation, which simultaneously adapts both the marginal and conditional distributions. The balanced distribution adaptation (BDA) method proposed by Wang *et al.* [11] used one tradeoff parameter to balance the importance of marginal and conditional distributions when learning the new feature representation. The visual DA (VDA) [12] method reduces the distribution discrepancy across domains in an unsupervised manner, and constructs domain invariant clusters in the embedding representation to separate various classes. The joint geometrical and statistical alignment (JGSA) [13] method is a unified framework to geometrically and statistically reduce the shifts between domains. Wei *et al.* [14] proposed the transfer with manifolds discrepancy alignment (TMDA) method to couple the discovery of data manifolds with the minimization of manifold maximum mean discrepancy (MMD). The dynamic double classifiers approximation (DDCA) [15] method integrates the feature representation learning and classifier learning into a unified optimization objective to guarantee the good classification performance.

Many deep feature transformation methods have been proposed to adapt the distribution because of the powerful feature representation capability of the convolutional neural network (CNN). Long *et al.* [17] proposed a deep adaptation network (DAN) framework to embed the deep features of all task-specific layers into RKHSs and optimally match distribution. The adversarial tight match (ATM) method proposed by Li *et al.* [18] used the maximum density divergence (MDD) to measure the distribution discrepancy between the source and target domains, and the MDD loss in ATM can simultaneously minimize the interdomain divergence and maximize the intraclass density.

The aforementioned methods are in the setting of transductive transfer learning [19], and they require that the source and target tasks are the same. They also require that all unlabeled objects in the target domain must be available at training time. In applications, the unlabeled objects in the target domain can provide some extra useful information for classification. Thus, we want to exploit the entire useful information contained in both the source and target domains. Some objects in the target domain are easily classified into the correct classes after distribution alignment, and we call them the easy-target objects. The classification accuracy of the remaining objects in the target domain is usually very low, and they are called the hard-target objects. In other words, the classification model learned from the knowledge in the source domain cannot perform very well on the hard-target objects. Actually, the easy-target objects also can train a classifier to classify these hard-target objects. The effective combination of complementary knowledge in the source domain patterns and easy-target objects can provide better classification performance on the hard-target objects. By doing this, the classification accuracy of total unseen objects in the target domain (i.e., easy-target objects and hard-target objects) is expected to be improved.

The feature-based methods are frequently employed because of their good performance for solving the DA problem. Thus, we adapt the distributions between the source and target domains using feature-based methods. In previous feature-based works [10], [11], the developed methods usually employ an expectation maximization-like (EM-like) mechanism to predict the labels of target-domain objects for estimating the conditional distribution under an unsupervised manner. If the pseudolabels of the unseen objects predicted by this mechanism do not change at each iteration, the labels correspond to the ground truth with a very high confidence. Inspired by this phenomenon, these objects can be used to provide some extra information for classifying the hard-target objects. The evidence theory is a good mathematical framework to represent and combine uncertain information, and it has been successfully used for information fusion [20], [21]. That is why we propose a method called distribution adaptation based on evidence theory (DAET) for effectively extracting and combining the complementary information contained in both the source and target domains. Based on this new approach, we will improve the classification accuracy of unseen objects. The main principles of DAET are as follows.

1) The exploitation of the knowledge is available in the target domain. For this, we identify the easy-target objects whose predicted labels never change during iterations because they are very likely correctly classified. These easy-target objects usually can bring some extra useful information for classifying the hard-target objects.
2) For one unseen hard-target object, we integrate the multiple classification results, thanks to the labeled source patterns at multistep iterations. This procedure aims to generate one high-quality item of evidence as much as possible using the knowledge in source-domain patterns. The easy-target objects under the new and original feature representations provide some complementary knowledge to classify this unseen hard-target object. Two pieces of the classification results will be obtained based on these easy-target objects under the original and new feature representations. The classification results are also integrated to effectively extract the useful information in the easy-target objects.
3) Two integrated classification results yielded by source-domain patterns and easy-target objects are combined, thanks to the framework of the evidence theory. The reliabilities/weights of these classification results are usually different because of the discrepancy between datasets. We use the mean discrepancy between the hard-target objects and the source-domain patterns, and that of the hard-target objects and easy-target objects to estimate the weighting factors. A discounting operation will make the combination results close to ground truth.

The remainder of this article is organized as follows. In Section II, some background knowledge, that is, transfer learning and evidence theory, is briefly introduced. The DAET method for cross-domain pattern classification is presented in Section III, and several experiments to test the effectiveness of DAET are reported in Section IV. Finally, Section V concludes this article.

II. Background Knowledge

This article focuses on the DA problem, which is essential to the transfer learning technique. The combination of useful knowledge in both the source and target domains is expected to improve the classification accuracy of unseen objects in the target domain. The evidence theory is an interesting mathematical framework to represent and fuse uncertain information [22]–[24]. In [24], the belief-based bidirectional heterogeneous transfer classification method is proposed to reduce the uncertainty using evidence theory and achieve the good classification performance. Thus, evidence theory is also employed here, and we briefly introduce some important concepts of transfer learning and evidence theory.

A. Transfer Learning

In transfer learning [2], there exist two important concepts: 1) domain $\mathcal{D}$ and 2) task $\mathcal{T}$. A domain contains two elements, that is: 1) a feature space $\mathcal{X}$ and 2) a marginal distribution $P(\boldsymbol{X})$, where $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ is the set of patterns. The domain is usually denoted by $\mathcal{D} = \{\mathcal{X}, P(\boldsymbol{X})\}$. The task $\mathcal{T}$ also has two components: 1) a label space $\mathcal{Y}$ and 2) a predictive function $f(\cdot)$, where $f(\cdot)$ is a function to predict the labels of unseen objects. Similarly, the task is often denoted by $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$.

Given one source domain $\mathcal{D}_s = \{\mathcal{X}_s, P(\boldsymbol{X}_s)\}$, one source task $\mathcal{T}_s = \{\mathcal{Y}_s, f_s(\cdot)\}$, one target domain $\mathcal{D}_t = \{\mathcal{X}_t, P(\boldsymbol{X}_t)\}$, and one target task $\mathcal{T}_t = \{\mathcal{Y}_t, f_t(\cdot)\}$, transfer learning aims to improve the learning of $f_t(\cdot)$ using the knowledge in the source domain $\mathcal{D}_s$ and source task $\mathcal{T}_s$ when $\mathcal{D}_s \neq \mathcal{D}_t$ or $\mathcal{T}_s \neq \mathcal{T}_t$. The preprocessing of original data or modification of the classification model should be done to transfer knowledge from the source domain $\mathcal{D}_s$ into the target domain $\mathcal{D}_t$.

Transfer learning becomes a standard/traditional machine learning problem when $\mathcal{D}_S = \mathcal{D}_T$ and $\mathcal{T}_S = \mathcal{T}_T$. In such case, patterns in the source domains (i.e., training labeled data) are used to learn a predictive function $f(\cdot)$, and it is directly used to predict the label of one unseen object $\boldsymbol{x}$ in the target domain (i.e., test data) as $f(\boldsymbol{x})$. The traditional machine learning methods try to learn a classifier from scratch for each domain (i.e., one should collect labeled data for building classification models when a new domain arises), while transfer learning tries to use the knowledge in related domains to build the classification models in the target domain when the source and target domains have different feature spaces or distributions.

When $\mathcal{D}_s \neq \mathcal{D}_t$, there exist two cases to consider.

1) $\mathcal{X}_s = \mathcal{X}_t$ and $P(\boldsymbol{X}_s) \neq P(\boldsymbol{X}_t)$, which is called DA or homogeneous transfer learning (HoTL).
2) $\mathcal{X}_s \neq \mathcal{X}_t$, which is known as heterogeneous transfer learning (HeTL).

The DA technique uses the knowledge in the source domain to help the learning of the target predictive function when the feature spaces of source and target domains are the same but the distributions are different, which corresponds to the aforementioned case 1. It has been successfully employed in many real applications [1], that is, cross-domain pattern classification [25], clustering [26], regression [27], and so on. Papers [1]–[3] provide detailed presentations of transfer learning and DA techniques.

The multisource DA is also an interesting research topic, and we have developed the combination transferable classification (CTC) [28] and evidential combination of augmented multisource of information (ECAMI) [29] methods for improving the classification accuracy of unseen objects in the target domains. The two methods combine information in multiple source domains by the evidence theory to obtain a higher classification performance. Our aim in this work is to effectively extract and combine useful information contained in the singleton source domain and the target domain to improve the classification accuracy.

B. Evidence Theory

The mathematical theory of evidence or evidence theory is a mathematical framework developed by Shafer [30] to reason about uncertainty, which is based on the belief functions (BFs) and on the rule of combination proposed by Dempster [31]. This theory is also known as the Dempster–Shafer theory (DST), or evidential reasoning (ER). It has been widely used in many real-world applications [32], [33] because of its ability to represent and combine imprecise/uncertain information in different fields, for example, data classification [34], [35], data clustering [36], [37], information fusion [20], [21], decision making [38], [39], and so on.

Let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_C\}$ be a finite set of answers of some questions, called the frame of discernment (FoD). In pattern classification, the element $\{\omega_c\}_{c=1}^{C}$ and Ω denote the actual category of one pattern and the label space, respectively. The power-set 2^{Ω} is the set of all subsets of FoD Ω, for example, if the FoD is $\Omega = \{\omega_1, \omega_2, \omega_3\}$, its power set is $2^{\Omega} = \{\emptyset, \{\omega_1\}, \{\omega_2\}, \{\omega_3\}, \{\omega_1, \omega_2\}, \{\omega_1, \omega_3\}, \{\omega_2, \omega_3\}, \Omega\}$.

The basic belief assignment (BBA), also called a mass function, is defined as a mapping $m(\cdot)$ from the power set 2^{Ω} to the interval [0, 1], and it satisfies the conditions: $\sum_{A \in 2^{\Omega}} m(A) = 1$ and $m(\emptyset) = 0$. For any $A \in 2^{\Omega} \backslash \{\emptyset\}$, $m(A)$ represents the belief that one is willing to commit exactly to A, given a certain piece of evidence. A is a focal element of the BBA $m(\cdot)$ when $m(A) > 0$. If all the focal elements are singletons, $m(\cdot)$ is said to be the simple Bayesian BBA. In a c-class pattern classification problem, $m(A)$ denotes the belief of one object belonging to the singleton class (e.g., $A = \{\omega_1\}$) or the disjunction (union) of several classes (e.g., $A = \{\omega_1, \omega_2\}$).

In DST, two BBA's $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ induced by two distinct sources of evidence are combined by Dempster's rule (a.k.a., DS rule) to provide a new BBA denoted by $\boldsymbol{m}_{12} = \boldsymbol{m}_1 \oplus \boldsymbol{m}_2$. The DS combination rule is called the orthogonal sum of $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$, and it is defined by

$$\begin{cases} m_{12}(\emptyset) = 0 \\ m_{12}(A) = m_1 \oplus m_2(A) = \dfrac{\sum_{B,C \in 2^{\Omega} | B \cap C = A} m_1(B) m_2(C)}{1 - K} \end{cases} \tag{1}$$

where $K = \sum_{B,C \in 2^{\Omega} | B \cap C = \emptyset} m_1(B) m_2(C) \in [0, 1]$ reflects the conflicting mass between two BBA's, and the symbol $\oplus$ is the DS combination operator. The computation of $\boldsymbol{m}_{12}$ is possible if and only if $K \neq 1$ because, otherwise, there is a logical contradiction when $K = 1$. The DS rule may produce unreasonable results when the sources of evidence highly conflict. Many other methods [30], [40] have been developed to solve

this problem. Nevertheless, DS rule has the important commutativity and associativity properties, and it often achieves pretty good combination performance when the sources of evidence are with low conflict. Thus, it is widely employed to combine the BBA's in many applications [20], [21].

III. Proposed Method

Let $\mathcal{D}_s = \{\mathcal{X}_s, P(\boldsymbol{X}_s)\}$ and $D_s = \{(\boldsymbol{x}_p^s, y_p^s)\}_{p=1}^{n_s}$ be the source domain and source-domain dataset, where $\mathcal{X}^s$ is the feature space of patterns $\{\boldsymbol{x}_p^s\}_{p=1}^{n_s}$; $\boldsymbol{X}_s = [\boldsymbol{x}_1^s, \boldsymbol{x}_2^s, \ldots, \boldsymbol{x}_{n_s}^s] \in \mathbb{R}^{k\times n_s}$ and k is the dimension of feature space; $P(\boldsymbol{X}_s)$ is the marginal distribution of source patterns; n_s is the number of patterns in the source domain. Similarly, the target domain and target-domain dataset are denoted by $\mathcal{D}_t = \{\mathcal{X}_t, P(\boldsymbol{X}_t)\}$ and $D_t = \{\boldsymbol{x}_q^t\}_{q=1}^{n_t}$, respectively, where $\mathcal{X}^t$ is the feature space of target patterns $\{\boldsymbol{x}_q^t\}_{q=1}^{n_t}$; $\boldsymbol{X}_t = [\boldsymbol{x}_1^t, \boldsymbol{x}_2^t, \ldots, \boldsymbol{x}_{n_t}^t] \in \mathbb{R}^{k\times n_t}$; $P(\boldsymbol{X}_t)$ is the marginal distribution of target patterns, and n_t is the number of unseen objects in the target domain. In this work, we consider the cross-domain classification problem where the source-domain patterns and the target-domain objects are in the same feature space but drawn from different distributions, that is, $\mathcal{X}_s = \mathcal{X}_t$ but $P(\boldsymbol{X}_s) \neq P(\boldsymbol{X}_t)$.

The distribution shift affects the classification accuracy of unseen objects in the target domain, and the classification performance in the target domain is low if we directly use the classification model learned by labeled patterns in the source domain. Many DA methods for solving such issue propose to first align the distribution, and then train classifiers based on labeled patterns in the source domain to classify unseen objects in the target domain. However, the information in the target domain is usually not considered because the target domain does not have any labeled patterns. The supervised information in the source domain is only used to classify unseen objects in the target domain. If we can take full advantage of useful knowledge in both the source and target domains, the classification accuracy should be substantially improved. In order to extract the useful information in the target domain, the classification model learned by knowledge in the source domain can be used to predict the (pseudo) labels of unseen objects in the target domain. The objects whose predicted labels are very likely correct can provide some (pseudo) supervised information for the classification. The combination of complementary information in both the source and target domains should improve the classification performance with respect to the only using of the knowledge in the source domain. A method to effectively extract and combine the useful information in both the source and target domains is presented in the sequel.

A. Easy-Target Objects Selection

The feature-based DA methods usually learn a new feature representation of patterns in both the source and target domains for reducing the distribution discrepancy. One classical feature-based method called JDA [10] adapts both the marginal and conditional distribution difference. JDA proposes an EM-like iteration mechanism to estimate the conditional distribution difference for learning the robust new feature representations. It makes sense to align the distributions and to transfer information as much as possible. In this work, we propose to extract the knowledge in the target domain based on this EM-like mechanism.

Let $\boldsymbol{X} = [\boldsymbol{X}_s, \boldsymbol{X}_t] \in \mathbb{R}^{k\times(n_s+n_t)}$ be the set of patterns in the source domain $\mathcal{D}_s$ and the target domain $\mathcal{D}_t$. The learning goal is to find a transformation matrix $\boldsymbol{A} \in \mathbb{R}^{k\times\tilde{k}}(\tilde{k} \leq k)$ such that the distributions of the source and target domain data under the new feature representation (i.e., $\boldsymbol{A}^T\boldsymbol{X}_s \in \mathbb{R}^{\tilde{k}\times n_s}$ and $\boldsymbol{A}^T\boldsymbol{X}_t \in \mathbb{R}^{\tilde{k}\times n_t}$), where $\tilde{k}$ is the dimension of the new feature space, are very close.

The discrepancy between the marginal probability distributions $P_s(\boldsymbol{X}_s)$ and $P_t(\boldsymbol{X}_t)$ is computed by

$$\begin{aligned}\mathcal{P} &= \left\| \frac{1}{n_s} \sum_{\boldsymbol{x}_p^s \in D_s} \boldsymbol{A}^T\boldsymbol{x}_p^s - \frac{1}{n_t} \sum_{\boldsymbol{x}_q^t \in D_t} \boldsymbol{A}^T\boldsymbol{x}_q^t \right\|^2 \\ &= \mathrm{tr}\big(\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{M}_0\boldsymbol{X}^T\boldsymbol{A}\big) \end{aligned} \tag{2}$$

with

$$\boldsymbol{M}_0 = \begin{bmatrix} \frac{1}{n_s n_s}\mathbf{1}_{n_s}\mathbf{1}_{n_s}^T & -\frac{1}{n_s n_t}\mathbf{1}_{n_s}\mathbf{1}_{n_t}^T \\ -\frac{1}{n_t n_s}\mathbf{1}_{n_t}\mathbf{1}_{n_s}^T & \frac{1}{n_t n_t}\mathbf{1}_{n_t}\mathbf{1}_{n_t}^T \end{bmatrix} \tag{3}$$

where $\mathrm{tr}(\cdot)$ is the trace of matrix; and $\mathbf{1}_{n_s}$ and $\mathbf{1}_{n_t}$ are vertical vectors of ones with n_s and n_t elements, respectively.

The discrepancy between the conditional probability distributions $Q_s(\boldsymbol{X}_s|y = \omega_c)$ and $Q_t(\boldsymbol{X}_t|y = \omega_c)$ w.r.t.[1] class ω_c is similarly computed by

$$\begin{aligned}\mathcal{Q}_c &= \left\| \frac{1}{n_s^c} \sum_{\boldsymbol{x}_p^s \in D_s^c} \boldsymbol{A}^T\boldsymbol{x}_p^s - \frac{1}{n_t^c} \sum_{\boldsymbol{x}_q^t \in D_t^c} \boldsymbol{A}^T\boldsymbol{x}_q^t \right\|^2 \\ &= \mathrm{tr}\big(\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{M}_c\boldsymbol{X}^T\boldsymbol{A}\big) \end{aligned} \tag{4}$$

with

$$\boldsymbol{M}_c = \begin{bmatrix} \frac{1}{n_s^c n_s^c}\tilde{\mathbf{1}}_{n_s}^c\tilde{\mathbf{1}}_{n_s}^{cT} & -\frac{1}{n_s^c n_t^c}\tilde{\mathbf{1}}_{n_s}^c\tilde{\mathbf{1}}_{n_t}^{cT} \\ -\frac{1}{n_t^c n_s^c}\tilde{\mathbf{1}}_{n_t}^c\tilde{\mathbf{1}}_{n_s}^{cT} & \frac{1}{n_t^c n_t^c}\tilde{\mathbf{1}}_{n_t}^c\tilde{\mathbf{1}}_{n_t}^{cT} \end{bmatrix} \tag{5}$$

where $D_s^c = \{\boldsymbol{x}_p^s | \boldsymbol{x}_p^s \in D_s \wedge y_p^s = \omega_c\}$ and $D_t^c = \{\boldsymbol{x}_q^t | \boldsymbol{x}_q^t \in D_t \wedge \hat{y}_q^t = \omega_c\}$ are the set of patterns belonging to class ω_c; y_p^s is the true label of source-domain pattern $\boldsymbol{x}_p^s$, and $\hat{y}_q^t$ is the predicted (pseudo) label of target object $\boldsymbol{x}_q^t$ by the classifier learned by knowledge in the source domain; $n_s^c = |D_s^c|$ and $n_t^c = |D_t^c|$ are the number of patterns in the source and target domains with class ω_c; $\tilde{\mathbf{1}}_{n_s}^c$ and $\tilde{\mathbf{1}}_{n_t}^c$ are indicator vectors with n_s and n_t elements. The pth (qth) element of vector $\tilde{\mathbf{1}}_{n_s}^c$ ($\tilde{\mathbf{1}}_{n_t}^c$) will be 1 if $\boldsymbol{x}_p^s$ ($\boldsymbol{x}_q^t$) belongs to class ω_c, or 0 otherwise.

To obtain the robust new feature representation, one has to minimize the objective function

$$\mathcal{P} + \sum_{c=1}^{C} \mathcal{Q}_c = \sum_{c=0}^{C} \mathrm{tr}(\boldsymbol{A}^T\boldsymbol{X}\boldsymbol{M}_c\boldsymbol{X}^T\boldsymbol{A}) + \lambda\|\boldsymbol{A}\|_F. \tag{6}$$

Let $\{\boldsymbol{z}_p^s = \boldsymbol{A}^T\boldsymbol{x}_p^s\}_{p=1}^{n_s}$ and $\{\boldsymbol{z}_q^t = \boldsymbol{A}^T\boldsymbol{x}_q^t\}_{q=1}^{n_t}$ be the patterns in the source and target domains under the new feature representation. The pseudolabels (i.e., $\hat{y}_q^t$) for one unseen

[1] with respect to.

object z_q^t can be obtained by the classification model learned by $\{z_p^t\}_{p=1}^{n_s}$. The predicted (pseudo) labels $\{\hat{y}_q^t\}_{q=1}^{n_t}$ are used for computing the discrepancy between conditional distributions when the true labels $\{y_q^t\}_{q=1}^{n_t}$ of target objects are not available. Then, an EM-like iteration mechanism [10] is employed to refine the predicted (pseudo) label quality. After multiple iterations, a robust new feature representation for the source-domain data and target-domain data is obtained.

If the distribution of some unseen objects is quite different from that of source-domain patterns, the predicted (pseudo) labels of these unseen objects will frequently change during iterations, and they are usually committed into a wrong class even though aligning the distributions. In practice, some unseen objects in the target domain may be easily classified with few steps of iteration because the distribution discrepancy between these unseen objects and patterns in the source domain is small. Moreover, the class is very likely to be the ground truth when its corresponding probability value is close to 1. Thus, the target-domain objects whose predicted labels never change and the corresponding class has a probability close to 1 at each iteration are very likely to be correctly classified. We call these objects the easy-target objects. Similarly, the unseen objects changing their pseudolabels a lot during iterations are usually hard to correctly classify, and we distinguish them as the hard-target objects.

Let $\{\omega_q^l\}_{l=1}^L$ and $\{m(\omega_q^l)\}_{l=1}^L$ be the predicted label and corresponding probability value at the lth iteration for the unseen object $\boldsymbol{x}_q^t$, where L is the number of iterations. The pattern will be regarded as easy-target object if and only if $\omega_q^1 = \omega_q^2 = \cdots = \omega_q^L$ and $\{m(\omega_q^l)\}_{l=1}^L \geq \varepsilon$, where ε is a positive threshold value (i.e., a tuning parameter) to control the selection of easy-target objects. This operation can successfully extract the knowledge in the target domain, and the selected easy-target objects will provide some extra (pseudo) supervised information for classification.

In order to obtain higher classification accuracy of unseen objects in the target domain (i.e., to correctly classify both the easy-target objects and hard-target objects), the classification performance on hard-target objects should be improved because they degrade the total classification accuracy a lot. In applications, the hard-target objects usually cannot be correctly classified based only on the information in the source domain. Fortunately, an extra classifier can be trained by easy-target objects with their pseudolabels, that is, the easy-target objects can provide some pseudosupervised information in the target domain for classifying hard-target objects. The classification results of hard-target objects using the classifiers trained by easy-target objects usually have more or less complementary knowledge w.r.t. the classification results yielded by the assistance of source-domain patterns. The combination of supervised information in the source domain and pseudosupervised information in the target domain is expected to improve the classification accuracy of hard-target objects.

B. Weighted Combination of Classification Results

For one unseen hard-target object $\boldsymbol{x}$, let $\hat{\boldsymbol{m}}_{1,l}$ be the soft classification result yielded by the auxiliary of source-domain patterns at the lth iteration. The new feature representations of patterns in the source and target domains are different at each iteration. Thus, the information contained in the source-domain data under different feature representations is more or less diverse. To obtain the useful information in L iterations, one can integrate the L classification results. The L pieces of classification results are not distinct because the preserved information on the new feature representations with L iterations overlaps to some degree, and the average fusion (AF) method can be naturally used here to integrate them. It is worth noting that the reliabilities/weights $\{\hat{\boldsymbol{m}}_{1,l}\}_{l=1}^L$ are usually different because the performance (i.e., classification accuracy) of the classifier learned at each iteration is diverse. It is not reasonable to directly use the AF method, and some weighting factors should be considered. Let θ_l be the classification accuracy in the source domain at the lth iteration, we propose to integrate the L classification results using AF with weights estimated by the classification accuracy as

$$\boldsymbol{m}_1 = \sum_{l=1}^{L} w_l \cdot \hat{\boldsymbol{m}}_{1,l} \tag{7}$$

with $w_l = [\theta_l / (\sum_{l=1}^L \theta_l)]$. This operation aims to extract useful information in the source domain as much as possible.

Let $D_t^{ea} = \{(\boldsymbol{x}_i^t, \hat{\boldsymbol{y}}_i^t)\}_{i=1}^{N_e}$ and $D_t^{ha} = \{\boldsymbol{x}_j^t\}_{j=1}^{N_h}$ be the dataset of easy-target objects and hard-target objects under original feature representation, respectively, and one can obtain $D_t = D_t^{ea} \cup D_t^{ha}$. The classification result for the hard-target object $\boldsymbol{x}$ can be denoted by $\tilde{\boldsymbol{m}}_2$ using the knowledge in D_t^{ea}. Let $\hat{D}_{t,l}^{ea} = \{(\boldsymbol{z}_{i,l}^t, \hat{\boldsymbol{y}}_i^t)\}_{i=1}^{N_e}$ and $\hat{D}_t^{ha} = \{\boldsymbol{z}_{j,l}^t\}_{j=1}^{N_h}$ be the dataset of easy-target and hard-target objects with new feature representation at the lth iteration, and one can also obtain $\hat{D}_{t,l} = \hat{D}_{t,l}^{ha} \cup \hat{D}_{t,l}^{ea}$. The classification result for object $\boldsymbol{x}$ using information in $\hat{D}_{t,l}^{ea}$ can be denoted by $\hat{\boldsymbol{m}}_{2,l}$. There may exist some complementary knowledge in the original and new feature representations at the lth iteration. The information contained in the original and new feature representation may overlap to some extent, so we also integrate the classification results using the AF method. Whereas, the performance of the classifier learned by easy-target objects under the original and new feature representations affects the reliabilities/weights of classification results. Thus, the weights estimated by classification accuracy must be considered when integrating the classification results $\tilde{\boldsymbol{m}}_2$ and $\hat{\boldsymbol{m}}_{2,l}$. Let $\tilde{\xi}$ and $\hat{\xi}_l$ be the classification accuracy on easy-target objects under the original and new feature representations at the lth iteration, and the integration result can be computed by

$$\boldsymbol{m}_2 = \frac{\tilde{\xi}}{\tilde{\xi} + \sum_{l=1}^L \hat{\xi}_l} \tilde{\boldsymbol{m}}_2 + \sum_{l=1}^{L} \frac{\hat{\xi}_l}{\tilde{\xi} + \sum_{l=1}^L \hat{\xi}_l} \hat{\boldsymbol{m}}_{2,l}. \tag{8}$$

This operation aims to obtain a robust classification result by effectively extracting knowledge in the easy-target objects.

In applications, the classification results yielded by the assistance of source-domain patterns and easy-target objects are regarded as two pieces of evidence for combination. The knowledge contained in the source-domain patterns and easy-target objects comes from different information sources, so the classification results are considered as distinct and

complementary. The proper combination of the two pieces of classification results should be able to further improve the classification accuracy. The evidence theory provides an efficient tool to characterize and combine the uncertain information, and it is also employed here for combining the classification results. The two pieces of classification results obtained by different classifiers usually have different reliabilities/weights. The classification result represented by BBA will be discounted by Shafer's discounting operation [30] with the corresponding weight to control its influence on the combination.

Let $\boldsymbol{m}$ and $\beta \in [0, 1]$ be one BBA and its corresponding weight, respectively, and the discounted BBA ${}^{\beta}\boldsymbol{m}$ is computed by

$$\begin{cases} {}^{\beta}m(A) = \beta \cdot m(A), A \in 2^{\Omega} \backslash \Omega \\ {}^{\beta}m(\Omega) = 1 - \beta + \beta \cdot m(\Omega). \end{cases} \tag{9}$$

In this discounting operation, the belief of each focal element will be partially discounted to the ignorance Ω according to the weight β. If the evidence is completely unreliable, we take $\beta = 0$. Thus, the discounted BBA will be ${}^{\beta}m(A) = 0$, $A \in 2^{\Omega} \backslash \Omega$ and ${}^{\beta}m(\Omega) = 1$. If the source of evidence is completely reliable, we take $\beta = 1$, and hence, we have ${}^{\beta}m(A) = m(A), A \in 2^{\Omega}$. After the discounting operation, the two pieces of classification results usually are not in very high conflict, because some beliefs are committed to the ignorance, which plays a neutral role in combination. The DS rule satisfies the commutativity and associativity law, and it generally produces pretty good performance in applications [34]. So it is employed here to combine the discounted classification results.

The weight of the classification result will be determined depending on the discrepancy between the hard-target objects and source-domain patterns (or easy-target objects). If the discrepancy of source-domain patterns (or easy-target objects) is high, the quality of classification result obtained based on the source-domain patterns (or easy-target objects) should be low, and the weight should be small. In applications, the mean difference is usually used to measure the discrepancy across datasets [8]. Thus, we use mean discrepancy between the hard-target objects and source-domain patterns, and the hard-target objects and easy-target objects to estimate the reliabilities/weights of the two classification results. There exist much useful knowledge when the mean discrepancy is small, and the corresponding classification results have high reliabilities/weights for combination. If the mean discrepancy is big, the useful knowledge is scarce, and one cannot obtain reliable classification results by fusing. In other words, the bigger the mean discrepancy, the smaller the weights.

The information in the patterns under the original and new feature representation is diverse, and the mean difference is also different. In order to obtain robust weights, the discrepancy between datasets under the original and new feature representation should be both considered when estimating the weighting factors. The weights can be determined by the above-mentioned mechanism as

$$\begin{cases} \beta_1 = \frac{\min\{d_1, d_2\}}{d_1} \\ \beta_2 = \frac{\min\{d_1, d_2\}}{d_2} \end{cases} \tag{10}$$

with

$$d_1 = \left\| \frac{1}{n_s} \sum_{z_p^s \in \hat{D}_s} z_p^s - \frac{1}{N_h} \sum_{z_q^t \in \hat{D}_t^{ha}} z_q^t \right\|^2 + \left\| \frac{1}{n_s} \sum_{\boldsymbol{x}_p^s \in D_s} \boldsymbol{x}_p^s - \frac{1}{N_h} \sum_{\boldsymbol{x}_q^t \in D_t^{ha}} \boldsymbol{x}_q^t \right\|^2 \tag{11}$$

and

$$d_2 = \left\| \frac{1}{N_e} \sum_{z_i^t \in \hat{D}_t^{ea}} z_i^t - \frac{1}{N_h} \sum_{z_j^t \in \hat{D}_t^{ha}} z_j^t \right\|^2 + \left\| \frac{1}{N_e} \sum_{\boldsymbol{x}_i^t \in D_t^{ea}} \boldsymbol{x}_i^t - \frac{1}{N_h} \sum_{\boldsymbol{x}_j^t \in D_t^{ha}} \boldsymbol{x}_j^t \right\|^2 \tag{12}$$

where $\| \cdot \|$ is the L_2 (Euclidean) norm; n_s, N_e, and N_h are the number of source-domain patterns, easy-target objects, and hard-target objects; D_s, D_t^{ea}, and D_t^{ha} are the datasets of source-domain patterns, easy-target objects, and hard-target objects under the original feature representation; $\hat{D}_s, \hat{D}_t^{ea}$, and $\hat{D}_t^{ha}$ denote the datasets of source-domain patterns, easy-target objects, and hard-target objects under the new representation.

The classification results obtained from the source-domain patterns and easy-target objects will be discounted by the estimated weighting (i.e., discounting) factors β_1 and β_2. Because the minimum mean discrepancy will induce a maximum weight (whose value is 1), the corresponding classification result will have an important impact for the combination. Let ${}^{\beta_1}\boldsymbol{m}_1$ and ${}^{\beta_2}\boldsymbol{m}_2$ be the discounted classification results by (9), and the combination of them using the DS rule as (1) is computed by

$$\boldsymbol{m} = {}^{\beta_1}\boldsymbol{m}_1 \oplus {}^{\beta_2}\boldsymbol{m}_2. \tag{13}$$

In applications, the combination result is usually transformed into the pignistic probability BetP$(\cdot)$ [41] for making the final class decision, and it is defined by

$$\text{BetP}(\omega_c) = \sum_{X \in 2^{\Omega}, \omega_c \in X} \frac{m(X)}{|X|} \tag{14}$$

where $|X|$ denotes the cardinality of X. The unseen object $\boldsymbol{x}$ is committed to the class with the biggest BetP$(\cdot)$ value as

$$\omega = \max_{\omega_c} \text{BetP}(\omega_c). \tag{15}$$

The flowchart to show the principle of DAET is shown in Fig. 1, and the pseudocode is given in Algorithm 1. In order to illustrate how DAET works, we also give a simple example.

Example: Assume the class space is $\Omega = \{\omega_1, \omega_2, \omega_3\}$, and there, respectively, exist five patterns in the source and target domains. The source- and target-domain datasets are given by

$$D_s = \left\{ \left(\boldsymbol{x}_1^s, \omega_1\right), \left(\boldsymbol{x}_2^s, \omega_2\right), \left(\boldsymbol{x}_3^s, \omega_1\right), \left(\boldsymbol{x}_4^s, \omega_2\right), \left(\boldsymbol{x}_5^s, \omega_3\right) \right\}$$
$$D_t = \left\{ \boldsymbol{x}_1^t, \boldsymbol{x}_2^t, \boldsymbol{x}_3^t, \boldsymbol{x}_4^t, \boldsymbol{x}_5^t \right\}.$$

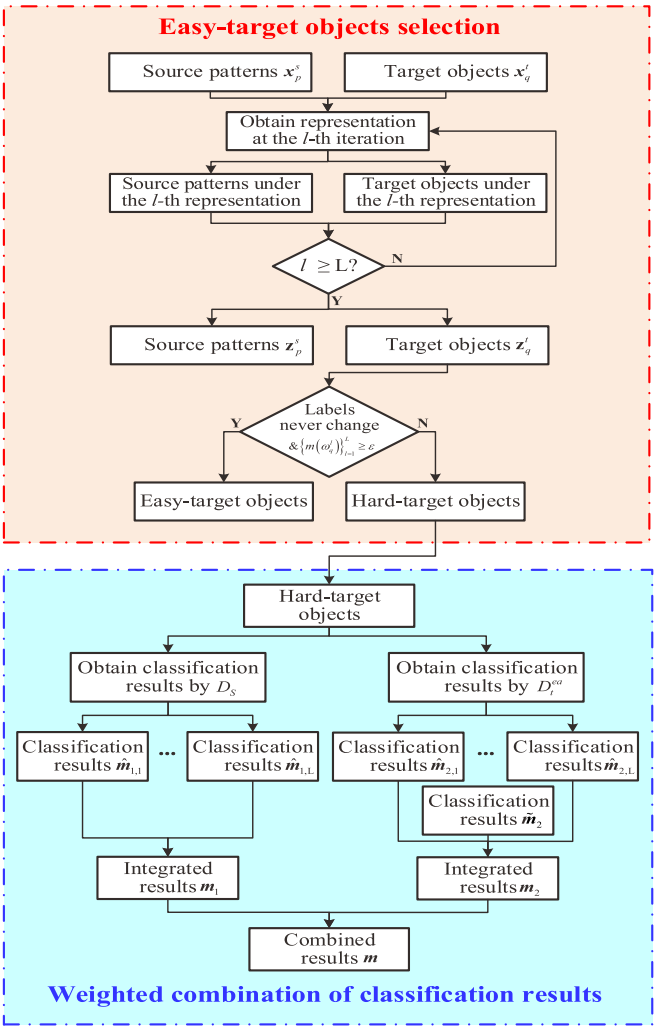

Fig. 1. Principle of the proposed DAET method.

It is assumed the easy-target objects with predicted (pseudo) labels are $(\boldsymbol{x}_1^t, \hat{\omega}_1)$, $(\boldsymbol{x}_4^t, \hat{\omega}_3)$, and $(\boldsymbol{x}_5^t, \hat{\omega}_3)$, and the hard-target objects are $\boldsymbol{x}_2^t$ and $\boldsymbol{x}_3^t$. For the hard-target object $\boldsymbol{x}_2$, two pieces of evidences obtained by effectively extracting information in five source-domain patterns and three easy-target objects are

$$m_1(\omega_1) = 0.50, m_1(\omega_2) = 0.40, m_1(\omega_3) = 0.10$$
$$m_2(\omega_1) = 0.20, m_2(\omega_2) = 0.50, m_2(\omega_3) = 0.30.$$

Assume the estimated weights are $\beta_1 = 0.8$ and $\beta_2 = 1$, and then the discounted resulted using (9) is

$$^{\beta_1}m_1(\omega_1) = 0.40, {}^{\beta_1}m_1(\omega_2) = 0.32, {}^{\beta_1}m_1(\omega_3) = 0.08$$
$$^{\beta_1}m_1(\Omega) = 0.20$$
$$^{\beta_2}m_2(\omega_1) = 0.20, {}^{\beta_2}m_2(\omega_2) = 0.50, {}^{\beta_2}m_2(\omega_3) = 0.30$$
$$^{\beta_2}m_2(\Omega) = 0.$$

The DS combination of these discounted classification results using (1) will be

$$m(\omega_1) = 0.2586, m(\omega_2) = 0.5603, m(\omega_1) = 0.1811$$
$$m(\Omega) = 0.$$

Algorithm 1 DAET

Input:

Data:

The source domain data set: $D_S = \{(\boldsymbol{x}_p^S, y_p^S)\}_{p=1}^{N_S}$ and the target domain data set: $D_T = \{\boldsymbol{x}_q^T\}_{q=1}^{N_T}$.

Parameter:

ε: Parameter to control the selection of easy-target objects.

1: Adapt distributions between the source and target domains, and obtain the new feature representation;
2: **for** $i = 1$ to N_h **do**
3: Select easy-target objects in the target domain by method proposed in Section III-A, and regard the rest of objects as the hard-target objects;
4: Obtain the classification results for hard-target objects by the knowledge in source domain patterns and easy-target objects using Eq. (7) and Eq. (8);
5: Compute the mean discrepancy using Eqs. (11)-(12) to estimate the weighting factors as Eq. (10);
6: Discount the classification results using Eq. (9) with estimated weighting factors;
7: Fuse the discounted results by DS rule as Eq. (13);
8: Make the class decision using Eq. (15).
9: **end for**

Output:

Labels for objects in the target domain.

One can see that the pignistic probability $\text{BetP}(\omega_2) = 0.5603$ is the biggest value, so the hard-target object $\boldsymbol{x}_2^t$ is committed to the class ω_2.

C. Discussion of the Tuning Parameter

The parameter $\varepsilon \in [0, 1]$ is used to control the selection of easy-target objects, and it must be tuned in real applications for good classification performance. It affects the number of easy-target objects a lot. If the value of ε is too big, the selected easy-target objects are correctly classified with very high confidence but few objects will be regarded as the easy-target objects. In this case, the number of easy-target objects is small, which means that the useful knowledge in easy-target objects will have a marginal impact in the improvement of the classification performance. There will be some wrong classified objects in easy-target objects when setting the ε value too small. Some objects which are wrongly classified are regarded as easy-target objects, and the wrong information usually has a negative influence on the combination. To summarize, the big ε value will lead to little useful information, while there will exist much wrong information in the easy-target objects when setting a small ε value. In other words, the determination of ε should not be too big or too small. In applications, if we want to obtain a good classification performance, we must select an appropriate value to guarantee that the number of selected easy-target objects is big, and the predicted (pseudo) labels of all easy-target objects are very likely to be the ground truth (i.e., the classification accuracy of easy-target objects is high). The influence of ε on classification accuracy is reported in the experiment. According to our heuristical test, we find that a

good classification performance is often obtained when setting $\varepsilon = 0.8$, so we recommend 0.8 as the default value for ε.

The iteration number L also affects the selection of easy-target objects, and the L times EM-like iteration mechanism aims to minimize both the marginal and conditional distribution. Thus, the determination of L depends on the the objective function given in (6), that is, the iteration number L should guarantee that the objective function can achieve the minimum value. In previous works [10]–[13], we have observed that the minimum objective function value is usually found after ten iterations, so we set $L = 10$ in our proposed DAET method.

IV. Experiments

A. Benchmark Datasets

Several classical benchmark datasets have been used to test the effectiveness of our proposed DAET method, and they are briefly described here.

Office+Caltech-10 contains four different domains, that is: 1) Amazon (A); 2) Caltech (C); 3) DSLR (D); and 4) Webcam (W) and it has 2533 images with ten categories, and the feature dimension is 800. We can define $4 \times 3 = 12$ cross-domain classification tasks as C→A, D→A, W→A, ..., A→W, C→W, D→W.

PIE are the subsets of CMU-PIE, which is a popular dataset for face recognition, and the feature dimension is 1024. It involves five different domains (i.e., face orientations) with 41 368 faces of 68 different identities, that is, PIE_C05 (PIE1, left pose), PIE_C07 (PIE2, upward pose), PIE_C09 (PIE3, downward pose), PIE_C25 (PIE4, front pose), and PIE_C29 (PIE5, right pose). Thus, $5 \times 4 = 20$ cross-domain classification tasks can be constructed as PIE2→PIE1, PIE3→PIE1, ..., PIE3→PIE5, PIE4→PIE5.

VLSC is a large image dataset with 4096 D features. It contains four domains: 1) VOC2007 (V); 2) LabelMe (L); 3) SUN09 (S); and 4) Caltech (C), and 10 729 samples w.r.t. five categories, that is: 1) bird; 2) cat; 3) chair; 4) dog; and 5) human. Thus, we define the cross-domain classification under $4 \times 3 = 12$ tasks: L→V, S→V, C→V, ..., V→C, L→C, S→C.

VisDA [42] is a very large benchmark dataset, which consists of a training domain, a validation domain, and a test domain. It contains over 280 000 images from 12 classes. We regard the training and validation sets as two (source and target) domains according to [18]. Two cross-domain classification tasks are defined: 1) training→Validation and 2) validation→Training.

B. Ablation Experiment

The proposed DAET method consists of two parts: 1) easy-target objects selection and 2) weighted combination of classification results. In order to test the effectiveness of our method, we do the ablation experiment[2] for performance evaluation of several variants of DAET with different fusion methods.

[2]Ablation experiment can be used to test the performance of a system by removing a chosen component to understand its contribution in the entire system. It is often employed to isolate the contributions of new method [16]–[18]. So we do ablation experiment here.

1) *DAET (CSETD):* In this method, the source-domain patterns and easy-target objects are simply merged as training data, and one classifier is learned based on these data to classify the hard-target objects.
2) *DAET (AF):* We can obtain two classifiers, respectively, trained by the source-domain patterns and easy-target objects. For one hard-target object, two pieces of classification results yielded by the classifiers are integrated by the AF method. The AF result is computed by $\boldsymbol{m} = (\boldsymbol{m}_1 + \boldsymbol{m}_2)/2$.
3) *DAET (DS):* In this method, two pieces classification results yielded by two classifiers using the source-domain patterns and easy-target objects are combined by Dempster's (DS) rule. The DS combination result is computed by (1) as $\boldsymbol{m} = \boldsymbol{m}_1 \oplus \boldsymbol{m}_2$.
4) *DAET (Murphy):* Two classification results obtained by information in the source and target domains are combined by the Murphy average rule [40]. The Murphy AF result is computed by $\boldsymbol{m} = \bar{\boldsymbol{m}} \oplus \bar{\boldsymbol{m}}$, where $\bar{\boldsymbol{m}} = (\boldsymbol{m}_1 + \boldsymbol{m}_2)/2$.
5) *DAET (WDS):* This method use the DS rule with the weighting factors to combine the two pieces of classification results, and the weighted DS combination result is computed by (13) as $\boldsymbol{m} = {}^{\beta_1}\boldsymbol{m}_1 \oplus {}^{\beta_2}\boldsymbol{m}_2$.

C. Comparison Methods

Several related methods are employed to construct the experiments for evaluating the effectiveness of DAET. The brief introduction of them are given as follows.

1) *K Nearest Neighbor (KNN) [43]:* The source-domain data are directly employed to learn the KNN model for classifying the unseen objects in the target domain without any data preprocessing or model modification.
2) *Correlation Alignment (CORAL) [44]:* It minimizes the distribution discrepancy by aligning the second-order statistics feature of the source and target domains.
3) *Geodesic Flow Kernel (GFK) [45]:* GFK embeds data into Grassmann manifolds and constructs geodesic flows between them to model domain shift. It integrates an infinite number of subspaces to learn new representations.
4) *Transfer Component Analysis [8]:* TCA adapts the marginal distribution discrepancy by learning a new representation. One classifier learned by source patterns under new representation is used to classify unseen objects.
5) *Adaptation Regularization-Based Transfer Learning (ARTL) [46]:* The method learns an adaptive classifier by simultaneously optimizing the structural risk, the joint distribution matching between domains, and the manifold consistency underlying marginal distribution.
6) *Landmarks Selection-Based Subspace Alignment (LSSA) [47]:* LSSA selects the landmarks to reduce the discrepancy between domains, and then the data are projected in the same space to make an efficient subspace alignment.

7) *Scatter Component Analysis (SCA) [48]:* SCA finds the representation that trades between maximizing the separability of classes, minimizing the shifts between domains, and maximizing the separability of data.
8) *Easy Transfer Learning (EasyTL) [49]:* It exploits the intradomain structures features to learn nonparametric transfer features and classifiers, and it does not need model selection and hyperparameter tuning.
9) *Stratified Transfer Learning [9]:* STL exploits the intraaffinity of each class between the source and target domains, and the intraclass knowledge are transferred into the same subspaces.
10) *Joint Distribution Adaptation [10]:* The marginal and conditional distributions across domains are both matched using an EM-like iteration mechanism to obtain the domain-invariant feature representation.
11) *Manifold Embedded Distribution Alignment (MEDA) [50]:* It learns a domain-invariant classier in Grassmann manifold with structural risk minimization, and performs dynamic distribution alignment to account for the relative importance of marginal and conditional distributions.
12) *Centroid Matching and Manifold Self-Learning CMMS [51]:* It makes label prediction for unseen objects by class centroid matching in order to exploit the data distribution structure, and a local manifold self-learning strategy is employed to capture the inherent local connectivity.
13) *Joint Distinct Subspace Classification (JDSC) [52]:* This method finds two coupled feature transformation matrixes for the source- and target-domain data to minimize the distribution shift, which considers the different importance of marginal and conditional distributions.
14) *Guide Subspace Learning (GSL) [53]:* The projection-based subspace guide, low-rank reconstruction-based data guidance, and relaxation-based label guidance are integrated to learn the domain-invariant feature.
15) *Joint Probability Domain Adaptation (JPDA) [54]:* It presents the joint MMD to replace the frequently used MMD, and this operation improves the transferability across domains and the discriminability across classes.
16) *Combined Source and Easy-Target Data (CSETD) [28]:* The source-domain patterns and easy-target objects are simply merged as new training data to provide information for classifying unseen objects in the target domain.

The code of these methods can be downloaded from the link http://transferlearning.xyz/.

D. Implementation Details

For fair comparison, the KNN classification method is regarded as the base classifier to classify unseen objects in the target domain after aligning the distribution by [8]–[10], [44], and [46]–[50], that is, the comparison methods also use the same KNN classifier. We set $K = 5$ according to our previous work on CTC [28], and the influence of K value has been discussed in CTC. The parameters of all comparison methods are set according to their values given in the related original papers. As for DAET, the iteration number is set to $L = 10$, and the regularization parameter where it is used in (6) is set to $\lambda = 1$. We set the dimension of new feature representation $\tilde{k} = 20, 50, 20, 20$ for **Office+Caltech-10**, **PIE**, **VLSC**, and **VisDA** datasets, which are the same taken for the comparison methods. The reported results are the classification accuracy in the target domain

$$\text{Accuracy} = \frac{\left|\boldsymbol{x} : \boldsymbol{x} \in \mathcal{D}_t \wedge \hat{y}_t = y_t\right|}{\boldsymbol{x} : \boldsymbol{x} \in \mathcal{D}_t} \tag{16}$$

where $\hat{y}_t$ is the predicted label of unseen object in the target domain, and y_t is the ground truth.

TABLE I
CORRECT CLASSIFICATION RATE (%) OF DIFFERENT DAET VARIANTS ON DIFFERENT CROSS-DOMAIN PATTERN CLASSIFICATION TASKS

Method	PIE2→PIE1	PIE1→PIE2	C→A	A→C	L→C	C→L	Average
Adaptation without DAET	44.90	42.54	46.97	39.63	50.37	48.96	45.56
DAET (CSETD)	46.06	30.86	37.21	17.01	32.31	42.99	34.41
DAET (AF)	41.62	37.50	31.42	37.22	50.79	51.27	41.64
DAET (DS)	44.96	43.80	43.22	39.73	50.34	51.98	45.67
DAET (Murphy)	44.99	44.08	38.70	39.98	50.98	52.01	41.12
DAET (WDS)	**46.73**	**44.58**	**47.70**	**40.61**	**51.71**	**52.93**	**47.38**

E. Experimental Results and Analysis

The classification accuracy of ablation study on **PIE**, **Office+Caltech-10**, and **VLSC** are shown in Table I. One can observe that the classification performance degrades if the source-domain patterns and easy-target objects are simply merged as new training data to classify hard-target objects. This is because the pseudolabels of the easy-target objects are not completely correct, and they have a negative influence on training the classifier. In the DAET (AF/DS/WDS) method, two classifiers are, respectively, trained using source-domain patterns and easy-target objects. This operation can reduce the negative influence of easy-target objects with wrong pseudolabels more or less. The DS rule is an interesting combination method to integrate the knowledge in different information sources, so the DAET (DS) method usually can achieve the higher classification accuracy compared with DAET (AF). One can see that the classification accuracy of DAET (Murphy) is higher than DS and AF method in some cases. It demonstrates that the Murphy average rule can reduce the negative influence of conflicts, and usually achieve a relative good classification performance compared with DS and AF. In applications, the classifier learned by easy-target objects with wrong pseudolabels is not completely reliable, and the classifier trained by source-domain patterns with somewhat distribution discrepancy also may be unreliable. Thus, two pieces of classification results $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ should be taken in account with different weights. The proposed method DAET (WDS) can further reduce the harmful influence, and combine the complementary information contained in source-domain patterns and easy-target objects. To summarize, the two parts working together can achieve a better classification performance than other methods. The detailed experiment results and analysis on these benchmarks are given in the sequel.

TABLE II
CLASSIFICATION ACCURACY OF DIFFERENT METHODS ON DATASET **PIE** (%)

Task	KNN	CORAL [44]	GFK [45]	TCA [8]	ARTL [46]	LSSA [47]	SCA [48]	EasyTL [49]	CSETD	STL [9]	JDA [10]	MEDA [50]	CMMS [51]	JDSC [52]	JPDA [53]	GSL [54]	DAET
PIE2→PIE1	20.00	21.02	23.88	42.45	32.65	31.21	33.16	42.35	46.06	42.08	44.90	42.25	39.20	43.55	42.55	42.40	**46.73**
PIE3→PIE1	24.29	26.12	26.53	27.55	25.99	35.02	28.06	44.34	46.77	42.92	33.06	39.58	42.90	40.55	47.23	38.67	**47.35**
PIE4→PIE1	33.27	33.27	24.08	50.61	48.47	43.46	44.90	60.46	36.88	57.75	54.69	52.86	51.74	57.59	61.01	60.92	**61.02**
PIE5→PIE1	24.29	23.88	17.55	39.50	26.56	21.79	38.78	41.28	32.40	33.75	32.65	24.58	**46.30**	41.90	35.00	44.69	34.08
PIE1→PIE2	30.83	31.25	28.33	39.58	24.49	22.96	37.50	43.75	30.86	12.65	42.54	42.21	40.00	35.73	39.79	42.32	**44.58**
PIE3→PIE2	20.42	20.42	31.52	42.75	27.50	31.00	40.10	44.27	40.98	43.85	42.50	42.92	44.20	36.77	36.92	43.18	**44.58**
PIE4→PIE2	37.08	37.50	60.00	72.50	49.36	42.23	54.17	66.67	57.73	75.83	72.50	51.10	64.50	55.92	68.96	72.49	**72.80**
PIE5→PIE2	17.08	17.08	35.42	53.33	22.28	25.35	32.29	36.98	40.64	**54.17**	51.25	40.00	47.50	39.23	30.42	40.65	49.58
PIE1→PIE3	42.50	42.50	44.58	40.83	25.49	28.43	36.98	42.92	33.33	11.88	42.50	41.90	41.40	39.15	39.85	44.52	**45.00**
PIE2→PIE3	23.33	22.92	27.50	47.92	29.41	30.33	38.02	37.50	57.82	62.92	51.67	50.83	**57.90**	44.73	50.62	51.04	55.42
PIE4→PIE3	39.17	38.75	26.67	82.92	46.81	50.49	47.40	55.21	66.18	21.84	86.25	80.41	76.50	53.80	64.58	81.67	**87.08**
PIE5→PIE3	20.83	20.42	26.25	45.42	24.02	25.00	31.77	33.85	40.56	48.33	46.25	30.42	47.50	42.83	37.71	45.52	**50.00**
PIE1→PIE4	46.73	46.73	46.21	58.37	38.63	38.78	45.66	42.35	50.77	44.49	58.37	47.76	56.40	50.44	59.80	53.86	**60.41**
PIE2→PIE4	30.20	31.02	41.48	66.12	49.32	40.73	50.51	61.99	50.22	64.25	66.53	42.08	62.90	63.17	60.92	64.69	**67.76**
PIE3→PIE4	25.71	26.12	28.37	64.69	51.16	46.53	42.35	45.41	50.85	59.58	66.12	62.29	55.98	62.24	53.47	61.93	**68.57**
PIE5→PIE4	22.04	22.45	22.24	50.20	41.45	37.99	32.65	48.98	38.83	51.67	53.67	51.70	43.31	54.18	40.71	50.75	**54.90**
PIE1→PIE5	25.83	25.42	27.50	29.17	19.91	27.51	30.50	**35.94**	15.53	27.96	30.00	30.99	32.05	17.64	32.08	27.50	31.25
PIE2→PIE5	16.67	17.50	34.17	46.25	25.74	28.73	25.00	35.42	38.76	41.67	45.42	37.08	45.78	32.55	24.17	29.79	**48.75**
PIE3→PIE5	12.92	12.50	26.25	44.17	27.45	30.21	40.10	43.75	35.58	45.42	45.42	27.08	45.32	29.90	36.46	40.52	**48.75**
PIE4→PIE5	30.00	30.42	20.00	70.00	39.77	35.72	31.25	50.52	63.24	62.58	70.00	62.42	68.97	43.32	41.67	51.56	**72.50**
Average	27.16	27.36	30.93	50.72	33.82	33.67	42.06	45.83	46.24	45.28	51.96	45.02	50.52	44.26	45.19	49.43	**55.81**
Win/Total	0/20	0/20	0/20	1/20	0/20	0/20	0/20	1/20	0/20	1/20	1/20	0/20	2/20	0/20	0/20	0/20	**16/20**

TABLE III
CLASSIFICATION ACCURACY OF DIFFERENT METHODS ON DATASET **OFFICE+CALTECH-10** (%)

Task	KNN	CORAL [44]	GFK [45]	TCA [8]	ARTL [46]	LSSA [47]	SCA [48]	EasyTL [49]	CSETD	STL [9]	JDA [10]	MEDA [50]	CMMS [51]	JDSC [52]	JPDA [53]	GSL [54]	DAET
C→A	22.76	20.15	41.02	36.12	38.00	38.31	43.47	42.90	37.21	40.62	46.97	43.32	47.50	42.44	46.24	40.54	**47.70**
D→A	26.62	30.69	32.05	30.90	32.15	25.16	29.50	32.25	36.00	20.04	32.46	32.25	35.60	**36.43**	29.75	34.34	32.15
W→A	22.65	26.20	30.48	29.54	30.72	28.60	29.63	29.33	16.57	18.06	31.32	34.55	37.30	28.39	31.63	**38.31**	30.69
A→C	24.04	23.06	40.25	32.95	34.82	39.00	30.99	36.60	17.01	16.92	39.63	37.49	35.40	21.99	40.16	40.51	**40.61**
D→C	26.09	32.06	30.10	28.05	28.76	27.34	26.84	31.60	26.16	20.31	29.39	32.59	31.10	26.71	32.06	**33.04**	29.92
W→C	18.08	25.73	30.72	18.88	22.80	26.80	26.52	28.32	18.23	14.87	31.26	30.28	30.20	25.11	31.70	26.46	**32.59**
A→D	23.57	30.75	36.61	28.03	42.04	33.12	40.48	31.21	15.11	19.75	**42.68**	38.85	39.50	31.85	39.49	40.13	40.76
C→D	24.84	26.75	41.40	35.03	36.31	35.03	37.30	39.49	41.72	39.11	47.77	43.31	45.42	44.20	45.86	47.13	**49.04**
W→D	44.59	73.25	77.90	57.96	80.89	78.34	62.70	64.97	79.61	57.78	81.53	67.52	65.00	67.50	80.70	80.89	**81.53**
A→W	27.12	26.10	40.00	32.54	32.20	33.89	39.92	32.20	35.56	34.92	42.71	37.29	34.20	34.53	44.75	40.68	**43.73**
C→W	26.10	19.66	40.68	24.04	33.56	32.54	36.97	28.81	39.54	36.58	44.41	44.75	40.89	43.73	43.73	46.44	**47.12**
D→W	43.73	63.56	64.41	58.64	73.92	71.86	57.73	66.10	67.88	69.17	77.97	63.05	68.80	59.66	69.75	70.98	**77.97**
Average	28.08	31.07	42.25	34.39	40.51	39.17	38.50	38.65	35.88	32.34	44.56	42.10	42.58	38.30	44.65	44.95	**46.20**
Win/Total	0/12	1/12	0/12	0/12	0/12	0/12	0/12	0/12	1/12	0/12	1/12	0/12	0/12	1/12	0/12	2/12	**8/12**

TABLE IV
CLASSIFICATION ACCURACY OF DIFFERENT METHODS ON DATASET **VLSC** (%)

Task	KNN	CORAL [44]	GFK [45]	TCA [8]	ARTL [46]	LSSA [47]	SCA [48]	EasyTL [49]	CSETD	STL [9]	JDA [10]	MEDA [50]	CMMS [51]	JDSC [52]	JPDA [53]	GSL [54]	DAET
L→V	24.22	35.07	40.42	50.07	49.17	35.51	45.47	42.94	32.31	24.22	50.37	50.64	49.90	40.21	50.67	44.43	**51.71**
S→V	27.64	37.44	45.91	50.07	49.03	35.66	47.99	48.14	46.26	26.94	54.98	53.81	54.50	46.51	52.45	44.43	**57.21**
C→V	54.09	52.01	47.55	54.38	55.57	49.18	26.30	53.64	32.81	51.39	55.13	56.75	53.80	52.60	54.23	44.43	**56.46**
V→L	50.28	50.47	47.83	41.21	45.77	42.72	50.09	47.07	42.99	44.14	48.96	52.15	41.30	48.54	49.45	46.69	**52.93**
S→L	27.64	39.70	44.05	43.29	47.64	44.05	43.86	51.80	36.76	42.60	**52.55**	36.55	49.20	51.76	50.09	46.69	51.98
C→L	34.09	31.80	25.14	27.98	35.35	19.85	32.65	37.81	43.80	**47.55**	34.03	40.75	44.50	32.38	29.49	46.96	36.48
V→S	54.53	51.80	50.50	47.91	52.66	53.81	42.16	53.96	44.77	50.30	52.37	53.19	53.80	45.61	52.66	46.26	**55.97**
L→S	29.21	34.53	32.66	36.40	36.58	30.22	28.27	30.00	44.78	23.48	35.54	30.26	29.90	36.12	32.37	36.62	**36.69**
C→S	42.01	42.16	21.51	40.29	45.47	31.51	40.07	36.26	46.16	32.41	46.33	47.57	41.30	44.75	41.73	36.26	**48.78**
V→C	51.46	63.84	74.41	73.57	79.71	62.73	44.37	79.42	57.99	55.56	80.39	78.23	79.70	74.83	75.80	80.24	**80.81**
L→C	16.83	22.25	27.26	48.54	65.65	23.92	34.35	60.36	42.33	32.62	**70.10**	65.60	55.80	43.37	41.45	44.28	65.51
S→C	21.56	32.82	34.24	46.04	49.10	38.39	33.24	47.58	23.53	27.31	47.57	34.10	49.10	46.59	51.32	44.20	**49.93**
Average	38.89	43.23	47.53	46.65	50.96	38.96	39.07	49.08	38.21	38.21	51.62	49.97	50.23	46.93	48.45	47.35	**53.76**
Win/Total	0/12	0/12	0/12	0/12	0/12	0/12	0/12	0/12	0/12	1/12	2/12	0/12	0/12	0/12	0/12	0/12	**9/12**

The classification performance (i.e., accuracy) of the proposed DAET method and other tested methods is shown in Tables II–V. We can see that the performance of standard machine learning method (i.e., KNN) is poor because the training and test data are drawn from different distributions. It is unreasonable to directly employ the standard machine learning methods without any data preprocessing. One can also see that the DA methods improve the classification performance more or less compared with the standard machine learning method. The distribution discrepancy is successfully reduced by these DA methods, that is, the knowledge in the source domain is effectively employed to classify unseen objects in the target domain. The performance of CSETD is not very good compared with other related methods because the information quality in

TABLE V
CLASSIFICATION ACCURACY OF DIFFERENT METHODS ON DATASET **VISDA** (%)

Task	KNN	CORAL [44]	GFK [45]	TCA [8]	ARTL [46]	LSSA [47]	SCA [48]	EasyTL [49]	CSETD	STL [9]	JDA [10]	MEDA [50]	CMMS [51]	JDSC [52]	JPDA [53]	GSL [54]	DAET
Training→Validation	32.46	52.50	53.40	47.70	47.10	48.75	42.70	48.30	49.59	52.32	57.30	57.60	57.54	56.43	55.68	58.08	**58.80**
Validation→Training	38.50	46.60	65.80	73.20	49.10	57.25	34.50	46.10	58.86	63.12	70.00	71.20	72.96	73.53	70.25	74.50	**75.00**
Average	35.48	49.55	59.60	60.45	48.10	53.00	38.60	47.20	54.23	57.72	63.65	64.40	65.25	64.98	60.97	66.29	**66.90**
Win/Total	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	**2/2**

the source patterns and easy-target objects is different, and the simplistic merging operation usually yields a bad influence on the classification. The performance of DAET on different datasets is globally higher than other related DA techniques because these DA techniques only use the information in the source domain to classify unseen objects in the target domain, and some useful information in the target domain is ignored.

In DAET, the unseen objects whose pseudolabels do not change during iterations and corresponding classes have very high probability values are regarded as easy-target objects to provide some extra knowledge. The integration of the classification results through L-steps iteration can effectively extract the useful information in the source-domain patterns. The integration of the classification results obtained by the assistance of easy-target objects under original and new feature representation provides robust Bayesian BBA's. The two integration procedures can successfully extract useful information in the source-domain patterns and easy-target objects.

Our experiment shows that the reliabilities/weights must be considered when combining information in source-domain patterns and easy-target objects. The weights estimated by the mean discrepancy discount the BBA's to reduce the negative influence of poor reliability. Finally, the combination of discounted classification results yielded by the auxiliary of source-domain patterns and easy-target objects makes the fusion result closer to the ground truth. The experimental results in Tables II-V demonstrate that the selection of easy-target objects can successfully work for providing some extra useful information for the classification of hard-target objects, and the estimated weights reduce the bad influence of reliability on the fusion. In some extreme cases, the classification accuracy of DAET is not higher than related methods because the predicted labels of easy-target objects are not completely correct, and the discounting operation cannot completely eliminate their negative influence on the combination even though a small weight is given. Overall, the classification accuracy of DAET is the highest compared with related methods in general, that is, DAET can successfully work to improve the classification accuracy in the target domain.

F. Parameter Influence

The parameter ε plays an important role to control the selection of easy-target objects, and we have evaluated its influence on the classification performance. The accuracy and number of easy-target objects, and the accuracy of the cross-domain classification tasks on benchmark datasets w.r.t. the ε value are shown in Figs. 2–4.

One can see that the accuracy of easy-target objects is improved with the increasing of ε, the bigger the ε value,

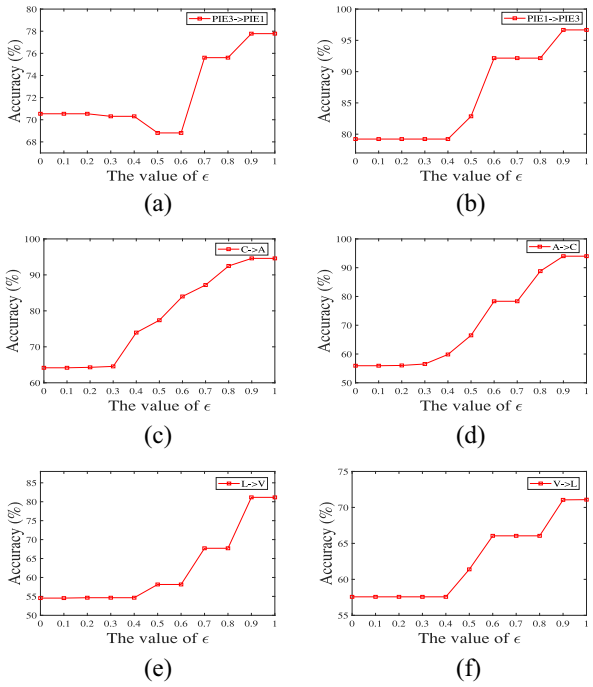

Fig. 2. Accuracy of easy-target objects w.r.t. ε value on different datasets. (a) **PIE**: PIE3→PIE1. (b) **PIE**: PIE1→PIE3. (c) **Office+Caltech-10**: C→A. (d) **Office+Caltech-10**: A→C. (e) **VLSC**: L→V. (f) **VLSC**: V→L.

the higher the accuracy of easy-target objects. This is a reasonable result because the predicted label with a very high probability value (i.e., mass value) is usually the ground truth. However, the number of easy-target objects decreases as the ε value increases because few objects will be regarded as the easy-target objects if the ε value is very big (i.e., less and close to one). The number of easy-target objects often drops a lot when one sets a bigger ε value than 0.8. The classification accuracy of cross-domain classification tasks also varies with the ε value, whereas one cannot achieve the best performance when the setting of ε is too big (i.e., close to one) or too small (i.e., close to zero). When selecting a small ε value, the accuracy of easy-target objects is low. There will be some wrong information in easy-target objects for classification, and it has a negative influence on the combination result. Thus, the total classification accuracy in the target domain is not very high. If one commits a big value to ε, the easy-target objects are almost corrected classified. However, the number of easy-target objects will be very small, that is, the information involved in them is not rich. We cannot effectively extract the knowledge in the target domain with a big ε value, so the total classification accuracy in the target domain is not the highest. One sees that the proposed method usually produces a good

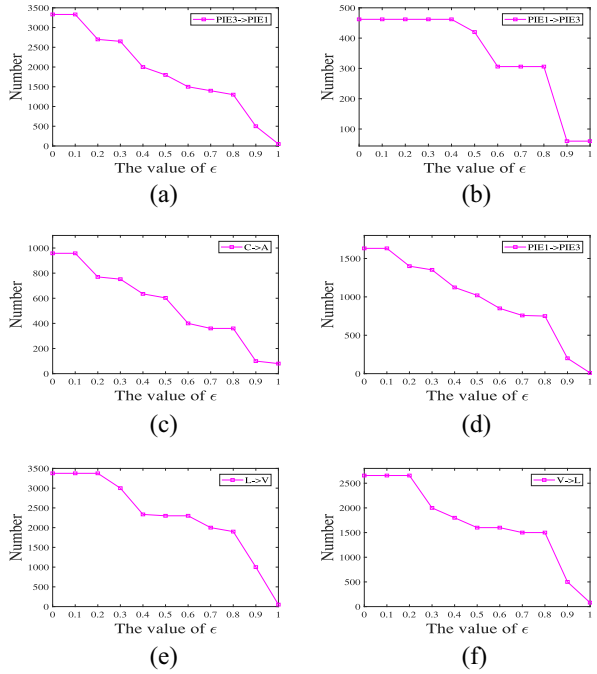

Fig. 3. Number of easy-target objects w.r.t. ε value on different datasets. (a) **PIE**: PIE3→PIE1. (b) **PIE**: PIE1→PIE3. (c) **Office+Caltech-10**: C→A. (d) **Office+Caltech-10**: A→C. (e) **VLSC**: L→V. (f) **VLSC**: V→L.

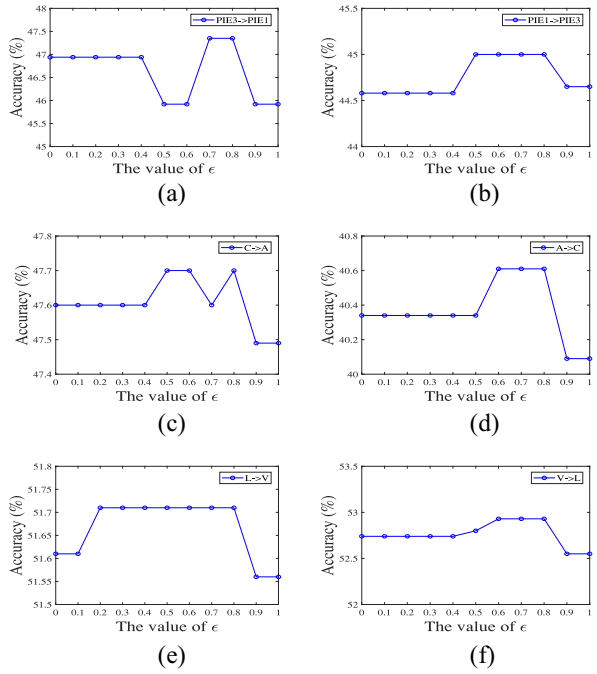

Fig. 4. Classification accuracy of DAET w.r.t. ε value on different datasets. (a) **PIE**: PIE3→PIE1. (b) **PIE**: PIE1→PIE3. (c) **Office+Caltech-10**: C→A. (d) **Office+Caltech-10**: A→C. (e) **VLSC**: L→V. (f) **VLSC**: V→L.

classification performance when $\varepsilon \in [0.6, 0.8]$, and that is why we recommend $\varepsilon = 0.8$ as the default value.

V. Conclusion

In this article, we have proposed a new DAET method to solve the cross-domain classification problem, and to improve the classification accuracy in the target domain. The unseen objects in the target domain whose predicted (pseudo) labels never change during iterations can be easily classified, and their pseudolabels correspond to the real classes with a very high confidence. So, we distinguished them as easy-target objects, and the remaining objects as hard-target objects. The easy-target objects are used to provide useful knowledge for classifying the hard-target objects. In contrary to classical methods of classification that are based only on the knowledge drawn in the source-domain patterns, we have shown in this article how to exploit the information of the easy-target objects to improve the classification accuracy. To effectively extract knowledge in the source domain, the classification results characterized by BBAs for the hard-target objects at multistep iterations are integrated by AF with the estimated weights. Similarly, the classification results yielded by easy-target objects under the original and new feature representation are also integrated. This processing makes the BBA's obtained by information in the source-domain patterns and easy-target objects robust. The final combination of complementary knowledge contained in the source and target domains improves in general the classification accuracy compared with only using knowledge in the source domain. Several classical cross-domain pattern classification benchmark datasets have been used to evaluate the effectiveness of this new DAET method. The experimental results show DAET can successfully improve the classification performance.

In future research works, we want to extend the application of DAET in other scenarios such as synthetic aperture radar automatic target recognition. Moreover, we also attempt to develop new cross-domain classification methods for dealing with multiple (more than two) source domains. It is expected that the classification accuracy could be further improved by combining complementary information in different source domains, and by using more advanced effective fusion rules.

References

[1] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: A survey," *Knowl. Based Syst.*, vol. 80, pp. 14–23, May 2015.

[2] 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.

[3] S. Sun, H. Shi, and Y. Wu, "A survey of multi-source domain adaptation," *Inf. Fusion*, vol. 24, pp. 84–92, Jul. 2015.

[4] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, "Boosting for transfer learning," in *Proc. Int. Conf. Mach. Learn.*, 2007, pp. 193–200.

[5] J. Huang, A. Smola, A. Gretton, K. Borgwardt, and B. Schölkopf, "Correcting sample selection bias by unlabeled data," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2006, pp. 601–608.

[6] M. Sugiyama, S. Nakajima, H. Kashima, P. V. Bünau, and M. Kawanabe, "Direct importance estimation with model selection and its application to covariate shift adaptation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2007, pp. 1433–1440.

[7] Y. Xu *et al.*, "A unified framework for metric transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 6, pp. 1158–1171, Jun. 2017.

[8] 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.

[9] J. Wang, Y. Chen, L. Hu, X. Peng, and P. S. Yu, "Stratified transfer learning for cross-domain activity recognition," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, 2018, pp. 1–10.

[10] 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.

[11] J. Wang, Y. Chen, S. Hao, W. Feng, and Z. Shen, "Balanced distribution adaptation for transfer learning," in *Proc. IEEE Int. Conf. Data Min.*, 2017, pp. 1129–1134.

[12] J. Tahmoresnezhad and S. Hashemi, "Visual domain adaptation via transfer feature learning," *Knowl. Inf. Syst.*, vol. 50, no. 2, pp. 1–21, 2016.

[13] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1859–1867.

[14] P. Wei, Y. K. Ke, X. Qu, and T.-Y. Leong, "Subdomain adaptation with manifolds discrepancy alignment," *IEEE Trans. Cybern.*, early access, May 13, 2021, doi:10.1109/TCYB.2021.3071244.

[15] X. Fang, N. Han, G. Zhou, S. Teng, Y. Xu, and S. Xie, "Dynamic double classifiers approximation for cross-domain recognition," *IEEE Trans. Cybern.*, early access, Jul. 15, 2020, doi:10.1109/TCYB.2020.3004398.

[16] J. Yang, J. Yang, S. Wang, S. Cao, H. Zou, and L. Xie, "Advancing imbalanced domain adaptation: Cluster-level discrepancy minimization with a comprehensive benchmark," *IEEE Trans. Cybern.*, early access, Aug. 16, 2021, doi:10.1109/TCYB.2021.3093888.

[17] 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.

[18] J. Li, E. Chen, Z. Ding, L. Zhu, and H. Shen, "Maximum density divergence for domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 3918–3930, Nov. 2021.

[19] A. Arnold, R. Nallapati, and W. W. Cohen, "A comparative study of methods for transductive transfer learning," in *Proc. Int. Conf. Data Min. Workshops*, 2007, pp. 77–82.

[20] Z.-G. Liu, Q. Pan, J. Dezert, J.-W. Han, and Y. He, "Classifier fusion with contextual reliability evaluation," *IEEE Trans. Cybern.*, vol. 48, no. 5, pp. 1605–1618, May 2018.

[21] Z.-G. Liu, J.-F. Duan, L.-Q. Huang, and J. Dezert, "Combination of classifiers with incomplete frames of discernment," *Chin. J. Aeronaut.*, to be published, doi:10.1016/j.cja.2021.04.020.

[22] Z.-G. Liu, Q. Pan, G. Mercier, and J. Dezert, "A new incomplete pattern classification method based on evidential reasoning," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 635–646, Apr. 2015.

[23] Z.-W. Zhang, H.-P. Tian, L.-Z. Yan, A. Martin, and K. Zhou, "Learning a credal classifier with optimized and adaptive multiestimation for missing data imputation," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Jun. 25, 2021, doi:10.1109/TSMC.2021.3090210.

[24] Z.-G. Liu, G.-H. Qiu, S.-Y. Wang, and T.-C. Li, "A new belief-based bidirectional transfer classification method," *IEEE Trans. Cybern.*, early access, Feb. 18, 2021, doi:10.1109/TCYB.2021.3052536.

[25] L. Wen, L. Gao, and X. Li, "A new deep transfer learning based on sparse auto-encoder for fault diagnosis," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 1, pp. 136–144, Jan. 2019.

[26] Z. Deng, Y. Jiang, F. L. Chung, H. Ishibuchi, K. S. Choi, and S. Wang, "Transfer prototype-based fuzzy clustering," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 5, pp. 1210–1232, Oct. 2016.

[27] H. Zuo, J. Lu, G. Zhang, and W. Pedrycz, "Fuzzy rule-based domain adaptation in homogeneous and heterogeneous spaces," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 2, pp. 348–361, Feb. 2019.

[28] Z.-G. Liu, L.-Q. Huang, K. Zhou, and T. Denœux, "Combination of transferable classification with multisource domain adaptation based on evidential reasoning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 5, pp. 2015–2029, May 2021.

[29] L.-Q. Huang, Z.-G. Liu, Q. Pan, and J. Dezert, "Evidential combination of augmented multi-source of information based on domain adaptation," *Sci. China Inf. Sci.*, vol. 63, no. 11, pp. 1–14, 2020.

[30] G. Shafer, *A Mathematical Theory of Evidence*, vol. 42. Princeton, NJ, USA: Princeton Univ. Press, 1976.

[31] A. P. Dempster, "Upper and lower probabilities induced by a multivalued mapping," *Ann. Math. Stat.*, vol. 38, no. 2, pp. 325–339, 1967.

[32] T. Denœux, "Logistic regression, neural networks and Dempster–Shafer theory: A new perspective," *Knowl. Based Syst.*, vol. 176, no. 3, pp. 54–67, 2019.

[33] C. Gong, P.-H. Wang, and Z.-G. Su, "An interactive nonparametric evidential regression algorithm with instance selection," *Soft Comput.*, vol. 24, pp. 3125–3140, Jan. 2020.

[34] T. Denœux, "A k-nearest neighbor classification rule based on dempster-shafer theory," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, no. 5, pp. 804–813, May 1995.

[35] Z.-W. Zhang, Z. Liu, Z. Ma, J. He, and X. Zhu, "Evidence integration credal classification algorithm versus missing data distributions," *Inf. Sci.*, vol. 569, no. 2, pp. 39–54, 2021.

[36] Z.-W. Zhang, Z. Liu, A. Martin, Z.-G. Liu, and K. Zhou, "Dynamic evidential clustering algorithm," *Knowl. Based Syst.*, vol. 213, no. 4, pp. 1–13, 2021.

[37] C. Gong, Z. Su, P. Wang, and Q. Wang, "Cumulative belief peaks evidential k-nearest neighbor clustering," *Knowl. Based Syst.*, vol. 200, no. 4, pp. 1–13, 2020.

[38] T. Denœux, "Decision-making with belief functions: A review," *Int. J. Approx. Reason.*, vol. 109, pp. 87–110, Jun. 2019.

[39] S. Huang, X. Su, H. Yong, S. Mahadevan, and D. Yong, "A new decision-making method by incomplete preferences based on evidence distance," *Knowl. Based Syst.*, vol. 56, no. 3, pp. 264–272, Jan. 2014.

[40] C. K. Murphy, "Combining belief functions when evidence conflicts," *Decis. Support Syst.*, vol. 29, no. 1, pp. 1–9, 2000.

[41] P. Smets, "Decision making in the TBM: The necessity of the pignistic transformation," *Int. J. Approx. Reason.*, vol. 38, no. 2, pp. 133–147, 2005.

[42] X. Peng, B. Usman, N. Kaushik, J. Hoffman, and K. Saenko, "VisDA: The visual domain adaptation challenge," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 2102–2107.

[43] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[44] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 2058–2065.

[45] B. Gong, S. Yuan, S. Fei, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2066–2073.

[46] 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.

[47] R. Aljundi, R. Emonet, D. Muselet, and M. Sebban, "Landmarks-based kernelized subspace alignment for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 56–63.

[48] 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, Jul. 2017.

[49] J. Wang, Y. Chen, H. Yu, M. Huang, and Q. Yang, "Easy transfer learning by exploiting intra-domain structures," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2019, pp. 1210–1215.

[50] J. Wang, W. Feng, Y. Chen, H. Yu, and P. S. Yu, "Visual domain adaptation with manifold embedded distribution alignment," in *Proc. ACM Int. Conf. Multimedia*, 2018, pp. 1210–1215.

[51] L. Tian, Y. Tang, L. Hu, Z. Ren, and W. Zhang, "Domain adaptation by class centroid matching and local manifold self-learning," *IEEE Trans. Image Process.*, vol. 29, pp. 9703–9718, 2020.

[52] S. N. Saray and J. Tahmoresnezhad, "Joint distinct subspace learning and unsupervised transfer classification for visual domain adaptation," *Signal Image Video Process.*, vol. 15, no. 2, pp. 279–287, 2020.

[53] L. Zhang, J. Fu, S. Wang, D. Zhang, Z. Dong, and C. L. P. Chen, "Guide subspace learning for unsupervised domain adaptation," *IEEE Trans. Image Process.*, vol. 29, pp. 2424–2438, 2020.

[54] W. Zhang and D. Wu, "Discriminative joint probability maximum mean discrepancy (DJP-MMD) for domain adaptation," in *Proc. Int. Joint Conf. Neural Netw.*, 2020, pp. 9703–9718.

Lin-Qing Huang was born in Xinyang, China. He received the bachelor's degree from Henan University, Kaifeng, China, in 2017, and the master's degree from Northwestern Polytechnical University, Xi'an, China, in 2020, where he is currently pursuing the Ph.D. degree.

His research interests mainly focus on pattern recognition, domain adaptation, and information fusion.

Zhun-Ga Liu (Member, IEEE) was born in Luoyang, China. He received the bachelor's, master's, and Ph.D. degrees from Northwestern Polytechnical University (NPU), Xi'an, China in 2007, 2010, and 2013, respectively.

He has been a Full Professor with the School of Automation, NPU since 2017. His current research interests mainly focus on pattern recognition, information fusion, and belief functions.

Jean Dezert was born in France, in 1962. He received the Electrical Engineering and Ph.D. degrees from the University of Paris XI, Orsay, France, in 1985 and 1990, respectively.

Since 1993, he has been a Senior Research Scientist with ONERA-The French Aerospace Laboratory, Palaiseau, France. His current research interests include information fusion, decision-making support, and belief function theory.