IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 14, 2020, accepted October 27, 2020, date of publication November 2, 2020, date of current version November 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3035094

Deep Domain Adaptation Based on Adversarial Network With Graph Regularization

XU JIA[1], NA MA[2], AND FUMING SUN[3]
[1]School of Electronics and Information Engineering, Liaoning University of Technology, Jinzhou 121001, China
[2]School of Business Administration, Liaoning Technical University, Huludao 125105, China
[3]School of Electronics and Communication Engineering, Dalian Minzu University, Dalian 116026, China

Corresponding author: Xu Jia (gbjdjiaxu@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61976042, and in part by the Natural Science Foundation of Liaoning Province under Grant 2019-ZD-0700.

ABSTRACT Although most transfer learning methods can reduce the difference of the feature distributions between the source and target domains effectively, some classes in the two domains may still be misaligned after domain adaptation, especially for the classes with similar features such as "bicycle" and "motorcycle". Therefore, a graph regularization based adversarial network model is proposed, whose innovations mainly include the following two aspects: First, a constraint function which is used to measure the difference between the features belonged to different classes is proposed, whose purpose is that not only the training accuracy is taken into account during supervised training, but also the difference between classes should be enlarged as much as possible; Then, a graph regularization constraint function is proposed, which makes all the classes have good local preserving properties after domain adaptation, and further reduces the possibility of all classes being misaligned. Experimental results on several cross-domain benchmark datasets show that our newly proposed approach outperforms state of the art methods.

INDEX TERMS Adversarial network, domain adaptation, graph regularization, image classification, transfer learning.

I. INTRODUCTION

For image classification, most of traditional methods need a large number of labeled samples during training, and obtain the recognition model through supervised learning. However, it is very expensive and time-consuming to annotate these samples, therefore, we hope that even if a limited number of labeled samples are supplied, the recognition model with good universality can be also obtained. As we know, when there is a certain difference in feature distributions between two datasets, the knowledge can be still transferred from one dataset to the other dataset by domain adaptation, i.e., from the source domain to the target domain, so for small sample learning it is very necessary to propose an effective domain adaptation method [1].

It can be seen from the existing researches on domain adaptation that the most prominent problem we face is that some classes with strong similarity in the source and target domains are likely to be misaligned after domain adaptation, and the samples which belong to these classes in the target domain may be misclassified [2]. To address this challenge, we propose a graph regularization based adversarial network model, and its contributions mainly include the following two aspects: 1) When performing supervised learning on the dataset in the source domain, the classes with significant differences are usually easy to be classified correctly such as "person" and "airplane". However, if we only take the training accuracy into account, and don't consider how to enlarge the difference between classes, the samples in the target domain which belong to the classes with similar features are likely to be misclassified. To this end, we propose a new function to measure the discrimination between classes, thus all classes are strongly distinguishable from each other through imposing the discrimination constraint on the model as shown in Fig. 1; 2) To align all the classes in the source and target domains accurately, a graph regularization constraint is imposed on the feature layer in the proposed model, thus all the classes have good locality preserving properties after domain adaptation. As shown in Fig.2, the symbols with different shapes represent four classes of objects, and the length of the line between two classes represents the similarity of the classes, where the similarity of class A and class B

The associate editor coordinating the review of this manuscript and approving it for publication was Ioannis Schizas.

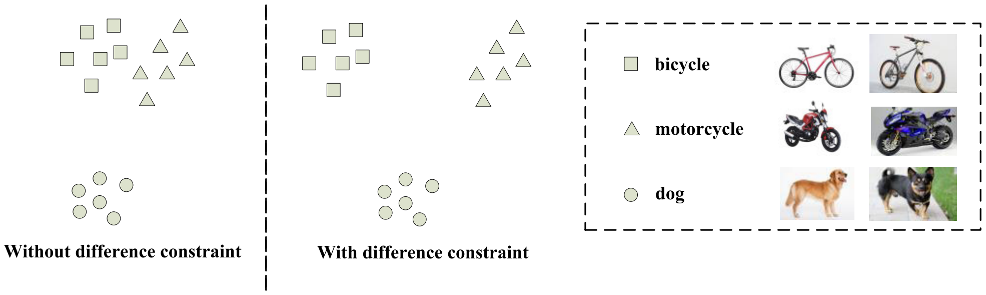


FIGURE 1. Schematic diagram of the feature distributions of the classes in source domain after supervised learning.

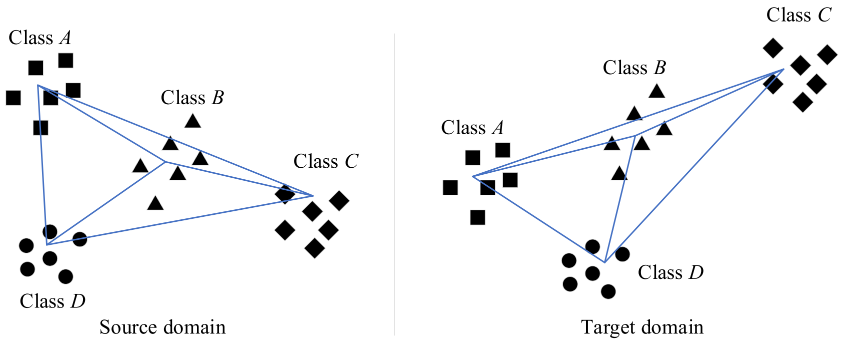


FIGURE 2. Schematic diagram of the feature distributions of the classes in the source and target domains after domain adaptation.

can be represented by S_{AB}. The purpose of imposing graph regularization constraint on the model is that after domain adaptation the feature similarities between each class and its adjacent classes in the source domain are almost the same as in the target domain. i.e., $S^s_{AB} \approx S^t_{AB}$, $S^s_{AC} \approx S^t_{AC}$, $S^s_{AD} \approx S^t_{AD}$, where s and t represent the source and target domains respectively.

The remainder of this article is organized as follows. In section II, some related works are addressed. In section III, the details of the proposed domain adaptation method are given. We have proved the effectiveness of the proposed method through the experiments that were conducted on some standard cross-domain recognition datasets in section IV. Finally, conclusions are drawn in section V.

II. RELATED WORK

In recent years, deep neural networks have been proved effectively in domain adaptation, and the existing methods based on deep neural network can be roughly divided into the following categories.

The first category is based on discrepancy, i.e., the discrepancy between the features which are extracted from the source and target domains should be as small as possible, where the commonly used functions for measuring feature discrepancy are shown as follows: Maximum Mean Discrepancy (MMD) [3], [4], Joint Maximum Mean Discrepancy (JMMD) [5], Weighted Maximum Mean Discrepancy (WMMD) [6], Wasserstein discrepancy [7], Sliced Wasserstein Discrepancy (SWD) [8], Orthogonal Discrepancy [9], Correlation Discrepancy [10]–[12], Source-Guided Discrepancy (SGD) [13], Contrastive Domain Discrepancy (CDD) [14] and pseudo-label differences [15], [16]. Besides the marginal distributions, the output class distributions are also considered in domain adaptation [17]. In addition, multi-domain adaptation can be achieved through moment matching [18], [19].

The second category is based on adversarial network, i.e., through using the extracted features to deceive the domain discriminator as much as possible, it is difficult to determine whether the features are from the source domain or the target domain [20]–[22]. In recent years, some scholars have carried out research on domain discriminator, for example, the discriminator can be optimized by using a gradient inversion layer in [23], Margin Disparity Discrepancy (MDD) is proposed to solve the distribution comparison with the asymmetric margin loss [24], and Batch Spectral Penalization (BSP) is proposed to boost the feature discriminability [25]. In addition, through improving single domain discriminator into multiple domain discriminators, the features of different levels [26] and the features of different classes [27], [28] can be aligned more accurately, and the domain discriminator can be also designed based on two-level domain confusion scheme [29]. The method is still effective when the number of classes in the source domain is more than in the target domains [30], [31], and the number of classes in the target domain is more than in the source domains [32].

The third category is based on reconstruction, where the existing reconstruction-based methods can be divided into two subcategories further, one is using encoder and decoder networks [33]–[36], and the other is also based on adversarial networks [37]–[39].

In addition, through using attention model idea, the transferable regions or images can be focused which are useful for domain adaptation [40], [41].

Although the effectiveness of the above methods has been verified on the specified datasets, there are still some problems that have not been solved well. For example, two classes with strong similarity may be misaligned after domain adaptation, such as "bicycle" and "motorcycle", which will make the samples that belong to these classes in the target domain be recognized incorrectly. To this end, we propose a new graph regularization based domain adaptation model, which can not only improve the discrimination of features between classes, but also make the feature distributions of all classes in the source and target domains similar sufficiently.

III. GRAPH REGULARIZATION BASED DOMAIN ADAPTATION

A. NETWORK MODEL

First, the source domain is defined as $\boldsymbol{D}_s = \left\{\left(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s\right)\right\}_{i=1}^{n_s}$, where n_s is the number of the samples in the source domain, $\boldsymbol{x}_i^s$ and $\boldsymbol{y}_i^s$ represent the ith sample and its label vector respectively, and the domain label vector of each sample is $\boldsymbol{d}_i = \left[\begin{matrix} 0 & 1 \end{matrix}\right]^T$, i.e., the sample $\boldsymbol{x}_i^s$ is from the source domain; Then, the target domain is defined as $\boldsymbol{D}_t = \left\{\boldsymbol{x}_j^t\right\}_{j=1}^{n_t}$, where n_t is the number of the samples in the target domain,$\boldsymbol{x}_j^t$ represents the jth sample, whose domain label is $\boldsymbol{d}_j = \left[\begin{matrix} 1 & 0 \end{matrix}\right]^T$, i.e., the sample $\boldsymbol{x}_j^t$ is from the target domain. The domain adaptation problem we need to solve is unsupervised, i.e., all the samples in the target domain are not annotated during training, and our goal is to obtain a feature extractor $\boldsymbol{f} = G_f\left(\boldsymbol{x}\right)$ and a feature classifier $\boldsymbol{y} = G_y\left(\boldsymbol{f}\right)$, which can enable the samples in the source and target domains to be recognized correctly.

To achieve knowledge transfer between domains, we hope that the features in the source and target domains should be as similar as possible, i.e., it is difficult to determine whether the feature comes from the source domain or the target domain. For this problem, some existing algorithms use adversarial networks to reduce the feature difference between the source and target domains. However, there are still some urgent issues that have not been considered, for example, the feature distributions in the source and target domains are sufficiently similar after domain adaptation, but it is likely that some classes with similar features between domains are misaligned such as class "dog" and class "cat", which may causes negative transfer. In addition, we can't determine whether the features which are extracted based on deep network are helpful enough for accurate classification. From the above analysis, we proposed an adversarial network with graph regularization based model, whose improvements are mainly reflected in the following two aspects.

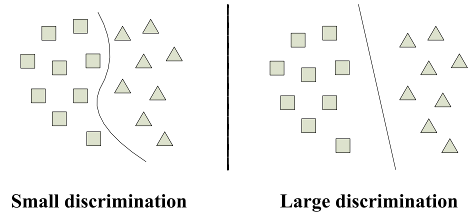


FIGURE 3. Schematic diagram of the optimal classifiers under different feature distributions.

1) As we known, if there are very small feature differences between classes, the algorithm will be easy to fall into local optimum during training classifier as shown in Fig. 3, so it is difficult to obtain a classifier with good robustness and generalization. On the contrary, if the feature of each class has good uniqueness, the above problem can be solved well. Therefore, we propose a new measurement function which can measure the discrimination of the features between classes as shown in (1), i.e., the uniqueness of the features,

$$L_s = \frac{2}{C^2} \sum_{1 \leq, i, j \leq C} \left\| \bar{\boldsymbol{f}}^{s,(i)} - \bar{\boldsymbol{f}}^{s,(j)} \right\|_2 \tag{1}$$

where C is the number of classes in the source domain,$\bar{\boldsymbol{f}}^{s,(i)}$ is the average feature vector of the ith class in the source domain in each epoch during training, which can be seen as the central feature of each class, and $\left\| \bar{\boldsymbol{f}}^{s,(i)} - \bar{\boldsymbol{f}}^{s,(j)} \right\|_2$ represents the discrimination between the ith and jth classes in the source domain. The purpose of imposing the constraint on the model is to improve the feature discriminations between classes as much as possible while acquiring high classification accuracy during training, therefore, the larger the value of the measurement function, the more helpful the extracted features are for classification.

2) Most of domain adaptation models fuse the features in the source and target domains by using domain discriminator and gradient reversal layer, however, some classes with similar features may be misaligned after domain adaptation such as "bicycle" and "motorcycle". To solve the problem, we impose a graph regularization constraint on the model, which enables the feature distributions of all classes in the source and target domains to be consistent after adaptation as shown in Fig. 2, in other words, each class has good local preserving property [42]. In addition, since the samples in the target domain are not annotated, we will give each sample a corresponding weight according to its pseudo label when imposing the graph regularization constraint, where the proposed graph regularization constraint function is shown in (2).

$$L_g = \sum_{1 \leq i, j \leq C} \left(w_{ij} \left\| \frac{\sum_{k=1}^{n_e} \left(\hat{y}_k^{(i)} \boldsymbol{f}_k^{t,(i)} \right)}{\sum_{k=1}^{n_e} \hat{y}_k^{(i)}} - \frac{\sum_{k=1}^{n_e} \left(\hat{y}_k^{(j)} \boldsymbol{f}_k^{t,(j)} \right)}{\sum_{k=1}^{n_e} \hat{y}_k^{(j)}} \right\|_2 \right) \tag{2}$$

$$w_{ij} = \frac{\left\|\bar{\boldsymbol{f}}^{s,(i)} - \bar{\boldsymbol{f}}^{s,(j)}\right\|_2}{\sum\limits_{1\le p,q\le C}\left\|\bar{\boldsymbol{f}}^{s,(p)} - \bar{\boldsymbol{f}}^{s,(q)}\right\|_2} \tag{3}$$

In (2), w_{ij} represents the correlation between the ith and jth classes which can be obtained by solving the normalized distance between the central features of the ith and jth classes as shown in (3), where $\bar{\boldsymbol{f}}^{s,(i)}$ is the average feature vector of the samples of the ith class in the source domain in each epoch, it is easy to draw that the more similar the features between classes in the source domain, the stronger their correlation, i.e., the greater the value of the parameter w_{ij}. Since the purpose of imposing the graph regularization constraint on the model is to make the classes in the target domain have the similar locality properties as the source domain after domain adaptation, i.e., if the features of two classes are similar in the source domain, the features of the two classes in the target domain should be also similar, where the central features of the ith and jth classes in the target domain can be solved by $\left(\sum_{k=1}^{n_e}\left(\hat{y}_k^{(i)}\boldsymbol{f}_k^{t,(i)}\right)\right)\Big/\sum_{k=1}^{n_e}\hat{y}_k^{(i)}$ and $\left(\sum_{k=1}^{n_e}\left(\hat{y}_k^{(j)}\boldsymbol{f}_k^{t,(j)}\right)\right)\Big/\sum_{k=1}^{n_e}\hat{y}_k^{(j)}$ respectively as shown in (2), $\boldsymbol{f}_k^{t,(i)}$ is the feature vector of the kth sample of the ith class in the target domain in each epoch during training, $\hat{y}^{(i)}$ is the ith component of the label vector $\hat{\boldsymbol{y}}$, $\hat{\boldsymbol{y}} = \left[\hat{y}^{(1)}\ \hat{y}^{(2)}\ \ldots\ \hat{y}^{(C)}\right]^T$, and n_e is the number of the samples in each epoch during training. From the above analysis, it is concluded that the graph regularization constraint function should be as small as possible.

In addition to the proposed two constraint terms, similar to other existing models, two cross-entropy functions as the other components of the final loss function [43] are used to measure the classification results of the samples in the source domain and the domain discrimination results of all samples which are shown in (4) and (5),

$$L_y = -\sum_{k=1}^{n_{es}}\left[\sum_{i=1}^{C}\left(y_k^{(i)}\log\hat{y}_k^{(i)} + \left(1-y_k^{(i)}\right)\log\left(1-\hat{y}_k^{(i)}\right)\right)\right] \tag{4}$$

$$L_d = -\sum_{k=1}^{n_e}\left[\sum_{i=1}^{2}\left(d_k^{(i)}\log\hat{d}_k^{(i)} + \left(1-d_k^{(i)}\right)\log\left(1-\hat{d}_k^{(i)}\right)\right)\right] \tag{5}$$

where $\boldsymbol{y}_k$ and $\hat{\boldsymbol{y}}_k$ are the real and pseudo labels of the kth sample which belongs to the source domain in each epoch, $\boldsymbol{y}_k = \left[y_k^{(1)}\ y_k^{(2)}\ \ldots\ y_k^{(C)}\right]^T$, $\hat{\boldsymbol{y}}_k = \left[\hat{y}_k^{(1)}\ \hat{y}_k^{(2)}\ \ldots\ \hat{y}_k^{(C)}\right]^T$, n_{es} is the number of the samples which belongs to the source domain in each epoch, and $\boldsymbol{d}_k$ and $\hat{\boldsymbol{d}}_k$ are the real and pseudo domain labels of the kth sample in each epoch, $\boldsymbol{d}_k = \left[d_k^{(1)}\ d_k^{(2)}\right]^T$, $\hat{\boldsymbol{d}}_k = \left[\hat{d}_k^{(1)}\ \hat{d}_k^{(2)}\right]^T$.

According to (1), (2), (4) and (5), it can be drawn that in our model the final loss function can be composed by 4 loss functions, which include the classification loss L_y of the samples in the source domain, the feature discrimination loss L_s of the samples in the source domain, the domain discrimination loss L_d of all samples, and the graph regularization loss L_g. The network architecture of the proposed method is shown in Fig. 4, where G_f and θ_f represent the feature extractor and its parameters, and $\boldsymbol{f}$ is the extracted feature; G_y and θ_y represent the feature classifier and its parameters, and $\hat{\boldsymbol{y}}$ is the classification result of the sample; G_d and θ_d represent the domain discriminator and its parameters, and $\hat{\boldsymbol{d}}$ is the domain discrimination result of the sample; ***GRL*** represents the gradient reversal layer. Our goal is to acquire the optimum parameters θ_f, θ_d and θ_y through training.

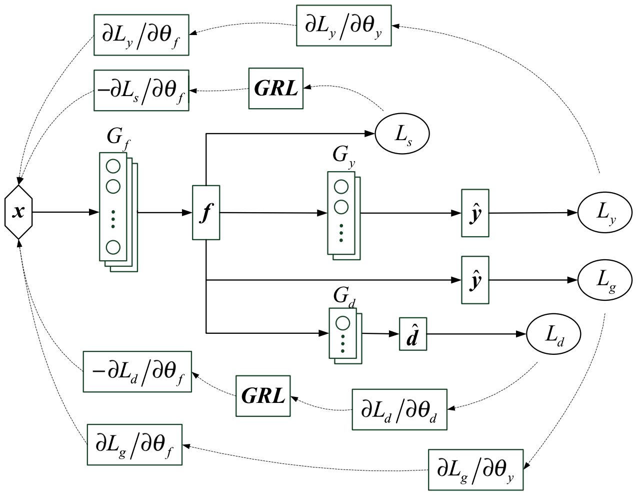

$\partial L_y/\partial\theta_f$
$\partial L_y/\partial\theta_y$
$-\partial L_s/\partial\theta_f$
GRL
L_s
G_f
G_y
x
f
$\hat{y}$
L_y
$\hat{y}$
L_g
G_d
$\hat{d}$
L_d
$-\partial L_d/\partial\theta_f$
GRL
$\partial L_d/\partial\theta_d$
$\partial L_g/\partial\theta_f$
$\partial L_g/\partial\theta_y$

FIGURE 4. **The proposed network architecture.**

Then, to obtain the optimum model parameters, a reasonable loss function should be given reasonably, which needs to meet the following requirements as much as possible:

1) Minimize L_y by seeking the parameters θ_y and θ_f;
2) Maximize L_s by seeking the parameters θ_f;
3) Minimize L_g by seeking the parameters θ_y, θ_d and θ_f;
4) Maximize L_d by seeking θ_d;

Therefore, the loss function can be written as (6),

$$\begin{aligned} L\left(\theta_f, \theta_y, \theta_d\right) = L_y\left(\theta_f, \theta_y, \boldsymbol{x}^s, \boldsymbol{y}^s\right) - \alpha L_s\left(\theta_f, \boldsymbol{x}^s\right) \\ - \beta L_d\left(\theta_f, \theta_d, \boldsymbol{x}, \boldsymbol{d}\right) \\ + \gamma L_g\left(\theta_f, \theta_y, \boldsymbol{x}, \boldsymbol{d}\right) \end{aligned} \tag{6}$$

where $\boldsymbol{x}^s$ and $\boldsymbol{y}^s$ represent the samples in the source domain and their labels, and $\boldsymbol{x}$ and $\boldsymbol{d}$ represent the samples in both source and target domains and their domain labels.

the optimization problem is to seek the best parameters $\hat{\theta}_y$, $\hat{\theta}_f$ and $\hat{\theta}_d$, which satisfy (7) and (8).

$$\left(\hat{\theta}_f, \hat{\theta}_y\right) = \underset{\theta_f,\theta_y}{\arg\min}\ L\left(\theta_f, \theta_y, \theta_d\right) \tag{7}$$

$$\hat{\theta}_d = \underset{\theta_d}{\arg\max}\ L\left(\theta_f, \theta_y, \theta_d\right) \tag{8}$$

B. OPTIMIZATION OF PARAMETERS BASED ON BATCH GRADIENT DESCENT

In the loss function of our proposed deep network, L_s and L_y depend on the samples in the source domain, and L_d and L_g depend on the samples in both the source and target domains. Therefore, it is necessary to select a certain number of samples from the source and target domains in each epoch during training, then use batch gradient descent method to optimize the model parameters, which is summarized below in Algorithm 1.

Algorithm 1 Model Parameters Optimization Process

```
Input $\boldsymbol{D}_s$, $\boldsymbol{D}_t$, $\alpha$, $\beta$, $\gamma$ and $\omega$, where $\boldsymbol{D}_s = \{(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s)\}_{i=1}^{n_s}$, $\boldsymbol{D}_t = \{(\boldsymbol{x}_i^t)\}_{i=1}^{n_t}$;
Initialize $\theta_y^{(0)}$, $\theta_f^{(0)}$ and $\theta_d^{(0)}$, $c = 0$;
for $epoch = 1, 2, \ldots, \omega$, do
    for $batch = 1, 2, \ldots$, do
        Update $\theta_y^{(c)}$, $\theta_f^{(c)}$ and $\theta_d^{(c)}$;
        $\theta_f^{(c+1)} \leftarrow \theta_f^{(c)} - \mu\left(\frac{\partial L_y^{(c)}}{\partial \theta_f} - \alpha\frac{\partial L_s^{(c)}}{\partial \theta_f} - \beta\frac{\partial L_d^{(c)}}{\partial \theta_f} + \gamma\frac{\partial L_g^{(c)}}{\partial \theta_f}\right)$
        $\theta_y^{(c+1)} \leftarrow \theta_y^{(c)} - \mu\left(\frac{\partial L_y^{(c)}}{\partial \theta_y} + \gamma\frac{\partial L_g^{(c)}}{\partial \theta_f}\right)$
        $\theta_d^{(c+1)} \leftarrow \theta_d^{(c)} - \mu\frac{\partial L_d^{(c)}}{\partial \theta_d}$
        $c = c + 1$
    end for
end for
Output $\hat{\theta}_y$, $\hat{\theta}_f$ and $\hat{\theta}_d$.
```

IV. EXPERIMENTS

A. DATASETS

In the experiments, we used the following three datasets, which include Office-31 dataset [44] and ImageCLEF-DA dataset [45], and Office-Caltech-10 dataset [46].

(a) Amazon
(b) DSLR
(c) Webcam

FIGURE 5. Some samples with the label "bottle" in Office-31 dataset.

1) There are three domains in the Office-31 dataset, which include "Amazon" (A), "Webcam" (W) and "DSLR" (D), and there are 31 classes of objects in each domain, where some samples with the label "bottle" are shown in Fig. 5.

2) There are three domains in the ImageCLEF-DA dataset, which include "Caltech-256" (C), "ImageNet ILSVRC 2012" (I) and "Pascal VOC 2012" (P), and there are 12 classes of objects in each domain, where some samples with the label "bicycle" are shown in Fig. 6.

(a) Caltech-256
(b) ImageNet ILSVRC 2012
(c) Pascal VOC 2012

FIGURE 6. Some samples with the label "bicycle" in ImageCLEF-DA dataset.

3) There are the following four domains in the Office-Caltech-10 dataset: "Amazon" (A), "Webcam" (W), "DSLR" (D) and "Caltech" (C), where "Amazon", "Webcam" and "DSLR" are the same as the domains in Office-31 dataset. There are 10 classes of objects in each domain contains, and some samples with the label "laptop computer" are shown in Fig. 7.

(a) Amazon
(b) DSLR
(c) Webcam
(d) Caltech

FIGURE 7. Some samples with the label "laptop computer" in Office-Caltech-10 dataset.

B. PARAMETERS SETTING

In our proposed network, AlexNet [33] is used for feature extraction and classification, and the three fully connected layers ($f \rightarrow 1024 \rightarrow 1024 \rightarrow 2$) [27] is used as the domain discriminator. In addition, through comparing the average classification accuracies under different combinations of parameters on the dataset "ImageCLEF-DA", the best classification results can be acquired under $\alpha = 0.5$, $\beta = 0.1$, $\gamma = 0.01$, where the comparison results are shown in Table 1.

TABLE 1. Comparison of recognition accuracy using Office-Caltech-10 dataset.

	$\gamma=0.01$	$\gamma=0.1$	$\gamma=0.5$	$\gamma=1$		$\gamma=0.01$	$\gamma=0.1$	$\gamma=0.5$	$\gamma=1$
$\alpha=0.01,\beta=0.01$	72.0	69.8	68.4	67.1	$\alpha=0.5,\beta=0.01$	80.1	80.4	79.6	76.6
$\alpha=0.01,\beta=0.1$	70.1	70.1	69.6	67.4	$\alpha=0.5,\beta=0.1$	81.3	79.3	79.1	69.0
$\alpha=0.01,\beta=0.5$	69.7	68.9	70.2	68.4	$\alpha=0.5,\beta=0.5$	78.2	78.1	76.2	70.9
$\alpha=0.01,\beta=1$	70.1	70.6	69.8	68.9	$\alpha=0.5,\beta=1$	77.1	72.8	71.0	68.6
$\alpha=0.1,\beta=0.01$	76.2	72.2	71.8	70.6	$\alpha=1,\beta=0.01$	74.2	71.4	70.5	70.1
$\alpha=0.1,\beta=0.1$	75.1	72.9	72.5	71.1	$\alpha=1,\beta=0.1$	73.1	71.4	69.5	67.5
$\alpha=0.1,\beta=0.5$	76.6	75.2	73.1	70.6	$\alpha=1,\beta=0.5$	74.2	72.5	70.9	68.4
$\alpha=0.1,\beta=1$	75.7	74.2	73.7	68.6	$\alpha=1,\beta=1$	73.7	72.9	70.1	70.1

TABLE 2. Comparison of recognition accuracy using ImageCLEF-DA dataset.

Model	Training accuracy	Transfer accuracy
	$C \rightarrow C$	$C \rightarrow I$
Model 1	99.4	47.8
Model 2	98.8	62.4
Ours	99.8	82.2

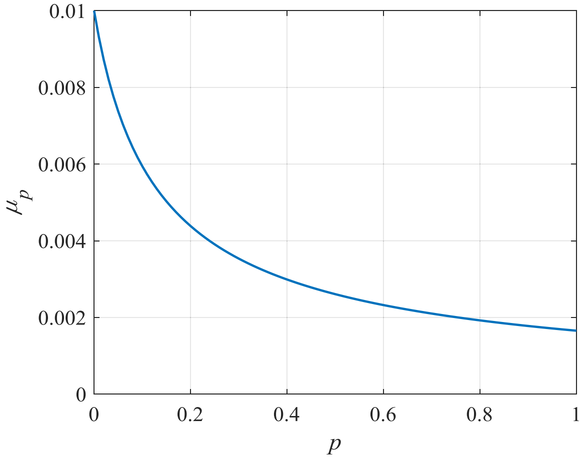


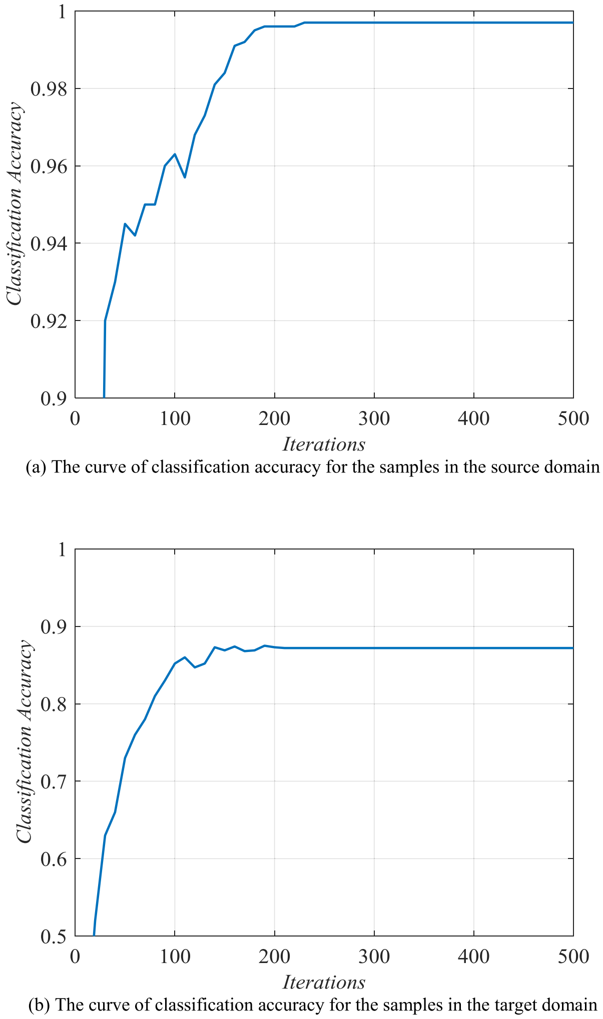
FIGURE 8. The variation curve of learning rate during training.

Besides the parameters in the model, the learning rate μ is another important parameter for training. In our proposed model, μ is not fixed during training, but should be adjusted according to (9), whose variation curve is shown in Fig. 8.

$$\mu_p = \frac{\mu_0}{(1+ap)^b} \tag{9}$$

$$p = \frac{N_{iter}}{N_{total}} \tag{10}$$

where p is the ratio of training process, N_{iter} and N_{total} represent the current numbers of iterations and the total number of iterations, $a = 10$, $b = 0.75$, and $\mu_0 = 0.01$.



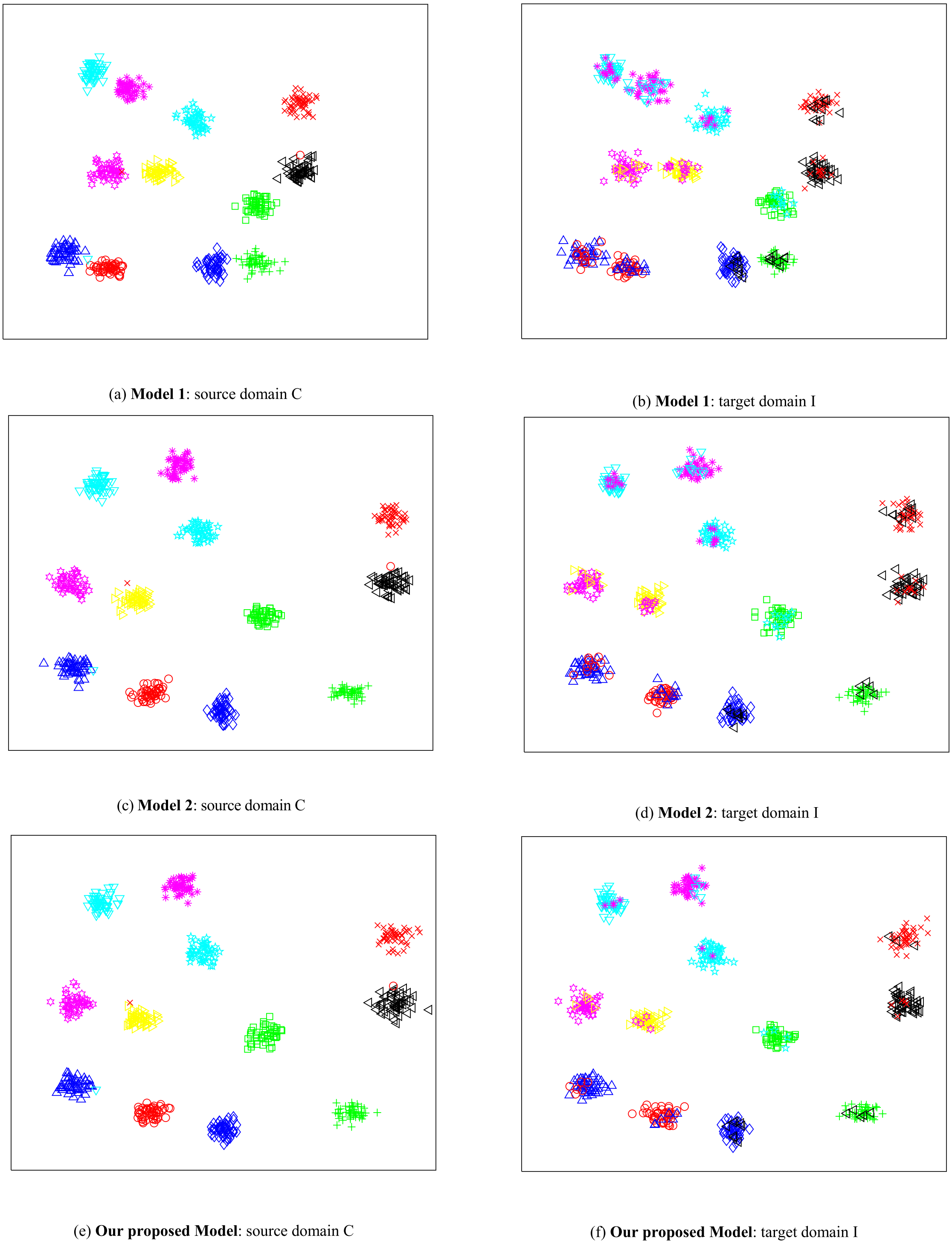
(a) The curve of classification accuracy for the samples in the source domain

(b) The curve of classification accuracy for the samples in the target domain

FIGURE 9. The curves of classification accuracies during training.

C. RESULTS AND ANALYSIS

After determining all parameters in the experiment, we will train the proposed model on the above three datasets. To observe the changes of multiple indicators during training,



(a) **Model 1**: source domain C

(b) **Model 1**: target domain I

(c) **Model 2**: source domain C

(d) **Model 2**: target domain I

(e) **Our proposed Model**: source domain C

(f) **Our proposed Model**: target domain I

FIGURE 10. The t-SNE visualization of deep features on the ImageCLEF-DA dataset.

the experimental results which are obtained from the dataset "ImageCLEF-DA" $P \rightarrow I$ are displayed as examples, where the curves of training and testing accuracies are shown in Fig. 9.

As can be seen from Fig. 9, when the number of iterations is about 200, the classification results of the samples in the source and target domains can converge to two fixed values respectively, i.e., 0.997 and 0.872. In summary, it is concluded that the proposed model is effective in dealing with domain adaptation problem.

To verify the effectiveness of our innovations, the proposed model will be modified to the following two models using

TABLE 3. The classified results of some easily misclassified samples.

The samples and these label	 aircraft	 bottle	 bicycle	 car	 aircraft
Model 3	bus	√	motorcycle	bus	car
Model 2	bus	√	motorcycle	√	car
Model 1	√	√	√	√	√

TABLE 4. Comparison of recognition accuracy using Office-31 dataset.

Method	$A \to W$	$D \to W$	$W \to D$	$A \to D$	$D \to A$	$W \to A$	Average
TCA [47]	59.0	90.2	88.2	57.8	51.6	47.9	65.8
DDC [4]	61.0	95.0	98.5	64.9	47.2	49.4	69.3
DAN [15]	68.5	96.0	99.0	66.8	50.0	49.8	71.7
RevGrad [23]	73.0	96.4	99.2	-	-	-	-
RTN [3]	73.3	96.8	99.6	71.0	50.5	51.0	73.7
MADA [27]	78.5	99.8	100.0	74.1	56.0	54.5	77.1
WAN [2]	79.2	99.8	99.8	74.8	56.0	54.5	77.3
MDD [28]	80.7	99.8	100.0	76.6	58.2	55.6	78.5
Ours	81.2	99.8	99.8	76.8	60.5	57.2	79.2

TABLE 5. Comparison of recognition accuracy using ImageCLEF-DA dataset.

Method	$I \to P$	$P \to I$	$I \to C$	$C \to I$	$C \to P$	$P \to C$	Average
DAN	67.3	80.5	87.7	76.0	61.6	88.4	76.9
RTN	67.4	82.3	89.5	78.0	63.0	90.1	78.4
RevGrad	66.5	81.8	89.0	79.8	63.5	88.7	78.2
MADA	68.3	83.0	91.0	80.7	63.8	92.2	79.8
WAN	69.7	82.2	90.3	81.9	63.6	92.4	80.0
MDD	71.5	82.9	91.0	81.8	64.5	92.8	80.7
Ours	71.7	83.2	91.0	82.2	66.2	93.3	81.3

different ways. First, the first modified model is obtained by removing the constraint terms L_s and L_g from the proposed model, i.e., **Model 1**; Then, the second modified model is obtained by only removing the graph regularization term L_g, i.e., **Model 2**. The parameters in the above modified models are the same as the parameters in the proposed model. After domain adaptation, the classification results based on different models are visualized in Fig.10 by using t-SNE and in Table 2.

In Fig.10, the points with different color and shape represent different classes of objects. As can be seen from Fig.10(a) and Fig.10(b), though the good classification accuracy can be obtained through using **Model 1** to perform supervised learning on the source domain, the feature discriminations between some similar classes are poor, and it is easy to misclassify the samples without labels in the target domain. For example, for the classes with strong similarity such as "bicycle" and "motorcycle", if only the training accuracy is taken into account, and the impact of the discrimination between the classes is ignored, the samples in these classes in the target domain are easily to be misclassify as shown in Table 3. In addition, from the results which are obtained using **Model** 2 in Fig.10(c) and Fig.10(d), after the constraint L_s is imposed on the model, the similar classes can be distinguished from each other well, however, the classification results of the samples in the target domain are still not satisfactory. At last, when using the proposed model, all classes have better local preserving properties after domain

TABLE 6. Comparison of recognition accuracy using Office-Caltech-10 dataset.

Method	$A \to C$	$A \to D$	$A \to W$	$C \to A$	$C \to D$	$C \to W$	$D \to A$	$D \to C$	$D \to W$	$W \to A$	$W \to C$	$W \to D$	Average
TCA	81.2	82.8	84.4	92.1	87.9	88.1	90.4	79.6	96.6	85.6	75.5	99.4	87.0
DDC	83.5	88.4	83.1	91.9	88.8	85.4	89.0	79.2	98.1	84.9	73.4	100	87.1
DAN	84.1	91.7	91.8	92.0	89.3	90.6	90.0	80.3	98.5	92.1	81.2	100	90.1
RTN	88.1	95.5	95.2	93.7	94.2	96.9	93.8	84.6	99.2	92.5	86.6	100	93.4
WAN	89.2	95.5	95.2	93.4	94.2	96.9	93.8	86.4	99.2	93.1	86.8	100	93.6
MDD	89.4	95.3	95.8	93.2	94.8	97.0	94.7	87.2	99.2	93.2	88.3	100	94.0
Ours	89.4	95.5	96.2	94.4	96.6	97.2	95.2	88.4	99.2	93.6	89.2	100	94.6

adaptation, which indicates that the feature distributions of all classes in the two domains have good consistency, and it can be seen from Fig.10(f) that the knowledge in the source domain can be transferred to the target domain well.

Then, we will compare the proposed method with some existed models using the following three datasets: Office-31 dataset, ImageCLEF-DA, dataset, and Office-Caltech-10 dataset, and the comparison results are shown in Table 4, Table 5, and Table 6, where $A \to W$ means that "A" and "W" represent the source domain and the target domain respectively, i.e., the samples in "A" and in "W" are used as the training and testing samples respectively in the experiment.

Through using the proposed model, the distributions of all classes in the source and target domains are almost the same, and even for the classes with strong similarity, they can be also aligned well. It can be seen from Table 4, Table 5 and Table 6 that the proposed method can achieve higher recognition accuracies, and we can conclude that our innovations are of great significance.

V. CONCLUSION

In this article, we proposed an adversarial network with graph regularization based domain adaptation method. On the one hand, to achieve a high recognition accuracy, we have obtained effective feature extractor and feature classifier through supervised learning; on the other hand, through imposing the graph regularization constraint on the adversarial network, the distributions of all classes in source and target domains have good consistency, i.e., the classes in the two domains can be aligned well after domain adaptation, and the images in the target domain can be classified accurately. The experimental results on the used three datasets showed the improvements in accuracy.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their helpful comments and suggestions. This work was supported by National Natural Science Foundation of China (No. 61976042) and Natural Science Foundation of Liaoning Province (No.2019-ZD-0700).

REFERENCES

[1] X. Jia, F. Sun, H. Li, Y. Cao, and X. Zhang, "Image multi-label annotation based on supervised nonnegative matrix factorization with new matching measurement," *Neurocomputing*, vol. 219, pp. 518–525, Jan. 2017.

[2] X. Jia and F. Sun, "Unsupervised deep domain adaptation based on weighted adversarial network," *IEEE Access*, vol. 8, pp. 64020–64027, 2020.

[3] M. S. Long, H. Zhu, J. M. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Barcelona Spain, 2016, pp. 136–144.

[4] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014, *arXiv:1412.3474*. [Online]. Available: http://arxiv.org/abs/1412.3474

[5] M. Long, H. Zhu, J. M. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70. Sydney, NSW, Australia, 2017, pp. 2208–2217.

[6] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 2272–2281.

[7] J. Shen, Y. R. Qu, W. N. Zhang, and Y. Yu, "Wasserstein distance guided representation learning for domain adaptation," *Proc. 32nd AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, 2018, pp. 4058–4065.

[8] C.-Y. Lee, T. Batra, M. H. Baig, and D. Ulbricht, "Sliced wasserstein discrepancy for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 10285–10295.

[9] G.-Y. Zhou and J. X. Huang, "Modeling and mining domain shared knowledge for sentiment analysis," *ACM Trans. Inf. Syst.*, vol. 36, no. 2, pp. 1–36, Sep. 2017.

[10] B. C. Sun and K. Saenko, "Deep CORAL: Correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, 2016, pp. 443–450.

[11] X. Peng and K. Saenko, "Synthetic to real adaptation with generative correlation alignment networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Lake Tahoe, NV, USA, Mar. 2018, pp. 1982–1991.

[12] C. Chen, Z. Chen, B. Jiang, and X, Jin, "Joint domain alignment and discriminative feature learning for unsupervised deep domain adaptation," in *Proc. 33rd AAAI Conf. Artif. Intell.*, Palo Alto, HI, USA, 2019, pp. 3296–3303.

[13] S. Kuroki, N. Charoenphakdee, H. Bao, J. Honda, I. Sato, and M. Sugiyama, "Unsupervised domain adaptation based on source-guided discrepancy," in *Proc. 33rd AAAI Conf. Artif. Intell.*, Palo Alto, HI, USA, 2019, pp. 4122–4129.

[14] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 4893–4902.

[15] M. S. Long, Y. Cao, J. M. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.*, vol. 37. Ithaca, NY, USA, 2015, pp. 97–105.

[16] X. Zhang, F. Xinnan Yu, S.-F. Chang, and S. Wang, "Deep transfer network: Unsupervised domain adaptation," 2015, *arXiv:1503.00591*. [Online]. Available: http://arxiv.org/abs/1503.00591

[17] M. Kim, P. Sahu, B. Gholami, and V. Pavlovic, "Unsupervised visual domain adaptation: A deep max-margin Gaussian process approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 4380–4390.

[18] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 1406–1415.

[19] S. Cicek and S. Soatto, "Unsupervised domain adaptation via regularized conditional alignment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 1416–1425.

[20] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," *Neurocomputing*, vol. 312, pp. 135–153, Oct. 2018.

[21] H. Zou, Y. X. Zhou, J. F. Yang, H. H. Liu, H. P. Das, and C. J. Spanos, "Consensus Adversarial Domain Adaptation," in *Proc. 33rd AAAI Conf. Artif. Intell.*, Palo Alto, HI, USA, 2019, pp. 5997–6004.

[22] C. Chen, W. Xie, W. Huang, Y. Rong, X. Ding, Y. Huang, T. Xu, and J. Huang, "Progressive feature alignment for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 627–636.

[23] Y. Ganin, V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. 32nd Int. Conf. Int. Conf. Mach. Learn.*, vol. 37. Lille, France, 2015, pp. 1180–1189.

[24] Y. Zhang, T. Liu, M. Long, and M. I. Jordan, "Bridging theory and algorithm for domain adaptation," 2019, *arXiv:1904.05801*. [Online]. Available: http://arxiv.org/abs/1904.05801

[25] X. Y. Chen, S. N. Wang, M. S. Long, and J. M. Wang, "Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation," in *Proc. 36th Int. Conf. Mach. Learn.*, Long Beach, CA, USA, 2019, pp. 1081–1090.

[26] W. Zhang, W. Ouyang, W. Li, and D. Xu, "Collaborative and adversarial network for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 3801–3809.

[27] Z. Y. Pei, Z. J. Cao, M. S. Long, and J. M. Wang, "Multi-adversarial domain adaptation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, vol. 2018, pp. 3934–3941.

[28] Y. C. Zhang, T. L. Liu, M. S. Long, and M. I. Jordan, "Bridging theory and algorithm for domain adaptation," *Int. Conf. Mach. Learn.*, Long Beach, CA, USA, 2019, pp. 7404–7413.

[29] Y. Zhang, H. Tang, K. Jia, and M. Tan, "Domain-symmetric networks for adversarial domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 5031–5040.

[30] J. Zhang, Z. Ding, W. Li, and P. Ogunbona, "Importance weighted adversarial nets for partial domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 8156–8164.

[31] Z. Cao, K. You, M. Long, J. Wang, and Q. Yang, "Learning to transfer examples for partial domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 2985–2994.

[32] K. You, M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Universal domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 2720–2729.

[33] M. Ghifary, W. B. Kleijn, M. Zhang, and D. Balduzzi, "Domain generalization for object recognition with multi-task autoencoders," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 2551–2559.

[34] M. Ghifary, W. B. Kleijn, M. J. Zhang, D. Balduzzi, and W. Li, "Deep reconstruction-classification networks for unsupervised domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, Amsterdam, The Netherlands, 2016, pp. 597–613.

[35] F. Z. Zhuang, X. H. Cheng, P. Luo, S. J. Pan, and Q. He, "Supervised representation learning: Transfer learning with deep autoencoders," in *Proc. 24th Int. Conf. Artif. Intell.*, Buenos Aires, Argentina, 2015, pp. 4119–4125.

[36] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 343–351.

[37] 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. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2223–2232.

[38] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70. Sydney, NSW, Australia, 2017, pp. 1857–1865.

[39] Z. Yi, H. Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised dual learning for Image-to-Image translation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2849–2857.

[40] X. M. Wang, L. Li, W. R. Ye, M. S. Long, and J. M. Wang, "Transferable attention for domain adaptation," in *Proc. 33rd AAAI Conf. Artif. Intell.*, Palo Alto, HI, USA, 2019, pp. 5345–5352.

[41] V. K. Kurmi, S. Kumar, and V. P. Namboodiri, "Attending to discriminative certainty for domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 491–500.

[42] F. Feng, X. He, J. Tang, and T.-S. Chua, "Graph adversarial training: Dynamically regularizing based on graph structure," *IEEE Trans. Knowl. Data Eng.*, early access, Dec. 5, 2019, doi: 10.1109/TKDE.2019.2957786.

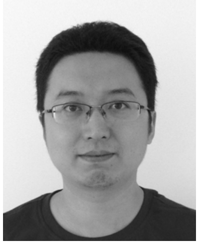
[43] A. Bietti, G. Mialon, D. X. Chen, and J. Mairal, "A kernel perspective for regularization deep neural networks," in *Proc. 36th Int. Conf. Mach. Learn.*, Stockholm, Sweden, 2019, pp. 664–674.

[44] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," in *Proc. Eur. Conf. Comput. Vis.*, Crete, Greece, 2010, pp. 213–226.

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

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[47] 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.



XU JIA was born in Kaiyuan, Liaoning, China, in 1983. He received the B.S. degree in automation from Shenyang Aerospace University, Liaoning, China, in 2005, and the M.S. and Ph.D. degrees in pattern recognition and intelligent system from Northeastern University, Liaoning, in 2009 and 2012, respectively.

From 2013 to 2015, he was a Lecturer with the Liaoning University of Technology, where he has been an Assistant Professor with the School of Electronics and Information Engineering, since 2016. He is the author of more than 30 articles. His research interests include machine learning and image processing.

NA MA was born in Chaoyang, Liaoning, China, in 1985. She received the B.S. degree in automation from Shenyang University, Liaoning, in 2008, and the M.S. degree in educational economic and management from Northeastern University, Liaoning, in 2010. She is currently pursuing the Ph.D. degree with Liaoning Technical University. Her research interests include machine learning and intelligent decision.

FUMING SUN was born in Dalian, Liaoning, China, in 1972. He received the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2007.

From September 2012 to July 2013, he was a Visiting Scholar with the Department of Automation, Tsinghua University. He was a Professor with the School of Electronics and Information Engineering, Liaoning University of Technology, from 2004 to 2018. He is currently a Professor with the School of Information and Communication Engineering, Dalian Minzu University, Dalian, China. His current research interests include content-based image retrieval, image content analysis, and pattern recognition.