TRANSFERABLE DISCRIMINATIVE FEATURE MINING FOR UNSUPERVISED DOMAIN ADAPTATION

Lingjun Zhao[1] *Wanxia Deng*[1*] *Gangyao Kuang*[1] *Dewen Hu*[2] *Li Liu*[2,3†]

[1]CEMEE, College of Electronic Science, National University of Defense Technology, China;
[2] National University of Defense Technology, China; [3]Univeristy of Oulu, Finland;

ABSTRACT

Unsupervised Domain Adaptation (UDA) aims to seek an effective model for unlabeled target domain by leveraging knowledge from a labeled source domain with a related but different distribution. Many existing approaches ignore the underlying discriminative features of the target data and the discrepancy of conditional distributions. To address these two issues simultaneously, the paper presents a Transferable Discriminative Feature Mining (TDFM) approach for UDA, which can naturally unify the mining of domain-invariant discriminative features and the alignment of class-wise features into one single framework. To be specific, to achieve the domain-invariant discriminative features, TDFM jointly learns a shared encoding representation for two tasks: supervised classification of labeled source data, and discriminative clustering of unlabeled target data. It then conducts the class-wise alignment by decreasing intra-class variations and increasing inter-class differences across domains, encouraging the emergence of transferable discriminative features. When combined, these two procedures are mutually beneficial. Comprehensive experiments verify that TDFM can obtain remarkable margins over state-of-the-art domain adaptation methods.

Index Terms— Domain adaptation, unsupervised learning, transfer learning, image classification.

1. INTRODUCTION

Domain Adaptation (DA) aims to leverage labeled data from one or more similar domains (named source domain) to improve the learning of the interested domain (named target domain) that has a distribution different from but related to the source distribution, *i.e.,* the domain shift [1]. We tackle one category of DA, *i.e.,* , the problem of Unsupervised DA (UDA) where the source domain contains abundant labeled data while the target domain is fully unlabeled. Mainstream approaches either explicitly extract the domain-invariant features via minimizing the domain discrepancy [2, 3, 4, 5, 6, 7] or implicitly learn them via adversarial learning [8, 9, 10, 11, 12, 13, 14]. Despite their general efficacy, these methods may still be constrained by two bottlenecks. First, these methods overlook the underlying discriminative features of the unlabeled target domain. Mining such information can contribute to accelerating the domain-invariant features learning, as well as to reducing the class-wise distribution discrepancy. Second, many existing methods tend only to minimize the divergence only between the marginal distributions and ignore the difference of conditional distributions. This may result in misclassification as shown in Figure 1 (b).

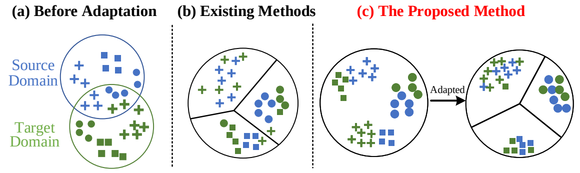


Fig. 1. (a) The before adaptation of source and target domains. (b) Existing methods ignore the intrinsic discriminative information and the class-level structures, which directly align the marginal distributions. (c) The proposed method mines the intrinsic discriminative features and adapts the class-level discriminative features corporately.

Accordingly, to tackle the two challenges outlined above simultaneously, we propose a new approach, named Transferable Discriminative Feature Mining (TDFM), in which we naturally unify the discriminative information exploration and class-distinguishable features adaptation in one framework. Figure 1 (c) simply shows our motivation. The left picture of the Figure 1 (c) depicts that we firstly promote the domain-invariant class-distinguishable structure by cooperatively learning the class boundaries in the source domain and the clusters boundaries in the target domain. Then we apply the obtained discriminative features to conduct class-wise alignment as shown in the right picture of the Figure 1 (c).

Specifically, in order to learn the discriminative, yet domain-invariant features, we propose to jointly learn a shared encoding representation for two tasks: supervised classification of labeled source data, and discriminative clustering of unlabeled target data. The shared encoding is trained through the simultaneous guidance of the supervised classification loss of the source domain and the unsupervised cluster-

[*] indicates the equal contribution with the first author.
[†] indicates the corresponding author. li.liu@oulu.fi

ing loss of the target domain, among which the clustering loss is defined as the KL divergence between the model's predictive label distribution and an auxiliary distribution [15, 16]. In this way, the learned representation can guarantee both domain invariance and categorical distinguishability.

In addition, as discussed above, the second challenge is how to ensure the corresponding categories of the two domains are correctly matched. With this in mind we conduct the class-wise alignment which explicitly minimizes the Maximum Mean Discrepancy (MMD) [17] distances of class-level distributions across domains. To be exact, the category labels of unlabeled samples in the target domain are first predicted and adopted as pseudo-labels during the training process. We then incorporate class-discriminative information by encouraging the intra-class minimum distance and inter-class maximum distance into the class-level distribution alignment.

The goal of the proposed TDFM is to integrate the mining of domain-invariant discriminative features and the adaptation of class-distinguishable features into a unified framework. The learning of domain-invariant discrimination features and the adaptation of the class-distinguishable information can be coupled in a mutually beneficial manner. In detail, the exploration of the class-discriminatory information can aid in keeping different classes of each domain far away from each other and the identical classes of each domain closer to each other, while the learning approach significantly diminishes the cross-domain marginal distribution discrepancy via the shared encoding. The good class-separability features obtained will also facilitate the class-level distribution adaptation. The contributions can be summarized as follows:

(1) We propose a novel TDFM approach to address two bottlenecks for UDA simultaneously. First, the learned domain-invariant representations can be equipped with category-discriminative knowledge. Second, the class-level distribution can be adapted by keeping different classes far away from each other and the identical classes closer to each other.

(2) The proposed TDFM explicitly unifies the learning of domain-invariant discriminative features and the adaptation of class-distinguishable features into a unified framework. TDFM can fully exploit the discriminative information of both domains and effectively minimize the marginal and conditional divergences simultaneously, thereby facilitate the learning process and boost the classification performance.

(3) Comprehensive experiments on the Office31, Office-Home and VisDA-C datasets demonstrate that TDFM outperforms existing methods by a large margin. Ablation studies prove the mining of the discriminative information and class-level features alignment can benefit from each other.

2. PROPOSED METHODOLOGY

2.1. The UDA Problem Formulation

We focus on the problem of UDA in image classification, where we consider two different domains defined with different but related probability distributions. The domain of interest is dubbed the target domain while the available domain with labeled data is called the source domain. The goal is to predict the labels of samples drawn from a target domain as accurately as possible, given N_s labeled samples $X^s = \{\boldsymbol{x}_i^s\}_{i=1}^{N_s}$, with the annotations $Y^s = \{y_i^s\}_{i=1}^{N_s}$ drawn from a source domain and N_t unlabeled samples $X^t = \{\boldsymbol{x}_i^t\}_{i=1}^{N_t}$ sampled from the target domain, and we have $y_i^s \in 1, 2, ..., C$. We define the feature extractor as f with parameters θ and the embedding classifier as g with parameters ϕ. We denote the whole network as $h = f \circ g$.

2.2. Exploration of Discriminative Features

We first introduce the exploration of discriminative features of the proposed Transferable Discriminative Feature Mining (TDFM). The supervised classification of the source domain and the unsupervised clustering of the target domain are learned cooperatively via shared encoding in order to extract the domain-invariant discriminative features. From a technical perspective, we define the supervised classification loss of the source domain as:

$$\mathcal{L}_{cls}(\theta, \phi) = \frac{1}{N_s}\sum_{i=1}^{N_s} \ell_{ce}(h(\boldsymbol{x}_i^s; \theta, \phi), y_i^s), \tag{1}$$

where ℓ_{ce} denotes the cross-entropy loss. Meanwhile, we consider the clustering learning of the target domain $X^t = \{\boldsymbol{x}_i^t\}_{i=1}^{N_t}$ which is clustered into C clusters in the output probability space. We define the prediction of the network, following a multinomial logistic regression operation (*i.e.,* ., softmax), as $\{p_i^t\}_{i=1}^{N_t}$ which we abbreviate to P^t. Similar to [16], we first introduce an auxiliary target variable Q^t. The clustering objective function can thus be defined as:

$$\mathcal{L}^{'}_{clu}(\theta, \phi) = \frac{1}{N_t}\sum_{i=1}^{N_t}\sum_{c=1}^{C} q_{ic}^t log\frac{q_{ic}^t}{p_{ic}^t} + q_{ic}^t log\varrho_k^t, \tag{2}$$

where the first term denotes the KL divergence between the model prediction probability P^t and the auxiliary target variable Q^t. The second term is used to enforce the balanced assignments, where $\varrho_k^t = \frac{1}{N_t}\sum_{i=1}^{N_t} q_{ic}^t$. q_{ic}^t denotes the element of Q^t, which is defined as follows:

$$q_{ic}^t = \frac{p_{ic}^t / (\sum_{j=1}^{N_t} p_{jc}^t)^{\frac{1}{2}}}{\sum_{c'=1}^{C} p_{ic'}^t / (\sum_{j=1}^{N_t} p_{jc'}^t)^{\frac{1}{2}}}. \tag{3}$$

To strengthen the discriminative ability of the learned feature, we further introduce the clustering learning in the latent feature space Z^t which is defined in the last layer output of the feature extractor, *i.e.*, $\boldsymbol{z}_i^t = f(\boldsymbol{x}_i^t; \boldsymbol{\theta}) \in Z^t$. Similarly, we partition the data in the latent feature space Z^t into C clusters, each of which is represented by a centroid $\boldsymbol{\mu}_c^t,\ c = 1, ..., C$, where $\boldsymbol{\mu}_c^t \in Z^t$ and the cluster centroid is learnable. Then we cluster the target samples using spherical K-means to obtain their pseudo labels. Following [18], we use the Student's t-distribution as a kernel to measure the distance from embedded point $\boldsymbol{z}_i^t$ to centroid $\boldsymbol{\mu}_j$, as follows:

$$\tilde{p}_{ic}^t = \frac{exp((1 + \|\boldsymbol{z}_i^t - \boldsymbol{\mu}_c\|^2))^{-1}}{\sum_{c'=1}^{C} exp((1 + \|\boldsymbol{z}_i^t - \boldsymbol{\mu}_{c'}\|^2))^{-1}}, \tag{4}$$

where $\tilde{p}_{ic}^t$ represents the probability of soft cluster assignments based on instance-to-centroid distances in the latent feature space Z^t. We collectively write $\tilde{p}_{ic}^t$ as $\tilde{P}^t$. Following [15], we introduce the auxiliary distribution $\tilde{Q}^t$ to conduct the clustering learning. Similar to Equation 3, the element of $\tilde{Q}^t$ is defined as follows:

$$\tilde{q}_{ic}^t = \frac{\tilde{p}_{ic}^t / (\sum_{j=1}^{N_t} \tilde{p}_{jc}^t)^{\frac{1}{2}}}{\sum_{c'=1}^{C} \tilde{p}_{ic'}^t / (\sum_{j=1}^{N_t} \tilde{p}_{jc'}^t)^{\frac{1}{2}}}. \tag{5}$$

Similar to Equation 2, the clustering loss of the latent feature can be defined as:

$$\mathcal{L}_{clu}^{''}(\theta, \phi) = \frac{1}{N_t} \sum_{i=1}^{N_t} \sum_{c=1}^{C} \tilde{q}_{ic}^t log \frac{\tilde{q}_{ic}^t}{\tilde{p}_{ic}^t} + \tilde{q}_{ic}^t log \tilde{\varrho_k}^t. \tag{6}$$

By combining Equations 2 and 6, we can obtain the overall clustering loss:

$$\mathcal{L}_{clu}(\theta, \phi) = \mathcal{L}_{clu}^{'}(\theta, \phi) + \mathcal{L}_{clu}^{''}(\theta, \phi). \tag{7}$$

By training the classification loss (*i.e.,* ., Equation 1) and the clustering loss (*i.e.,* ., Equation 7) together, we can obtain domain-invariant category-discriminative representations.

2.3. Adaptation of Discriminative Features

The adaptation of the discriminative features step will explicitly minimize the distances of conditional distributions by encouraging small within-class compactness and large between-class dispersion across domains.

Accordingly, similar to [4], we minimize the Maximum Mean Discrepancy (MMD) [17] distance of the intra-class sample pairs while maximizing the distance of the inter-class sample pairs in the probability output space. In more detail, the loss between two class conditional distributions with their mean embeddings in the Reproducing Kernel Hilbert space (RKHS) can be written as follows:

$$\begin{aligned} \mathcal{L}'_{ada} = \frac{1}{C}\sum_{c=1}^{C}(&\sum_{\boldsymbol{x}_i, \boldsymbol{x}_j \in X_{(c)}} \|\psi(h(\boldsymbol{x}_i)) - \psi(h(\boldsymbol{x}_j))\|_{\mathcal{H}} \\ &- \sum_{\boldsymbol{x}_i \in X_{(c)}} \sum_{\boldsymbol{x}_v \notin X_{(c)}} \|\psi(h(\boldsymbol{x}_i)) - \psi(h(\boldsymbol{x}_v))\|_{\mathcal{H}}), \end{aligned} \tag{8}$$

where $X_{(c)} = X_{(c)}^s \cup \hat{X}_{(c)}^t$. $X_{(c)}^s$ denotes source samples in class c. $\hat{X}_{(c)}^t$ denotes target samples in pseudo-class c. The pseudo-label of the target domain is predicted by the spherical K-means clustering mentioned above. $\psi(\cdot)$ is the kernel feature map of RKHS. As computing the Equation 8 directly is intractable, we use the kernel trick to rewrite the first term as follows:

$$d^{intra} = \frac{1}{C}\sum_{c=1}^{C}(o_1 + o_2 - 2o_3), \tag{9}$$

where $o_1 = \sum_{i=1}^{N_s} \sum_{j=1}^{N_s} \frac{\mathbf{1}_c(y_i^s, y_j^s) k(h(\boldsymbol{x}_i^s), h(\boldsymbol{x}_j^s))}{\sum_{i=1}^{N_s} \sum_{j=1}^{N_s} \mathbf{1}_c(y_i^s, y_j^s)}$, $o_2 = \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} \frac{\mathbf{1}_c(\hat{y}_i^t, \hat{y}_j^t) k(h(\boldsymbol{x}_i^t), h(\boldsymbol{x}_j^t))}{\sum_{i=1}^{N_t} \sum_{j=1}^{N_t} \mathbf{1}_c(\hat{y}_i^t, \hat{y}_j^t)}$, and $o_3 = \sum_{i=1}^{N_s} \sum_{j=1}^{N_t} \frac{\mathbf{1}_c(y_i^s, \hat{y}_j^t) k(h(\boldsymbol{x}_i^s), h(\boldsymbol{x}_j^t))}{\sum_{i=1}^{N_s} \sum_{j=1}^{N_t} \mathbf{1}_c(y_i^s, \hat{y}_j^t)}$. $\hat{y}_i^t$ and $\hat{y}_j^t$ denote the pseudo-labels of the target domain, while k represents the kernel function [2]. Each element of $\mathbf{1}_c(y, y')$ is defined as: $\mathbf{1}_c(y, y') = 1$ if $y = y' = c$; $\mathbf{1}_c(y, y') = 0$, otherwise. In the same way, the second term of Equation 8 can be written as follows:

$$d^{inter} = \frac{1}{C(C-1)} \sum_{c=1}^{C} \sum_{c'=1, c' \neq c}^{C} (o'_1 + o'_2 - 2o'_3), \tag{10}$$

where $o'_1 = \sum_{i=1}^{N_s} \sum_{j=1}^{N_s} \frac{\mathbf{1}_{cc'}(y_i^s, y_j^s) k(h(\boldsymbol{x}_i^s), h(\boldsymbol{x}_j^s))}{\sum_{i=1}^{N_s} \sum_{j=1}^{N_s} \mathbf{1}_{cc'}(y_i^s, y_j^s)}$, $o'_2 = \sum_{i=1}^{N_t} \sum_{j=1}^{N_t} \frac{\mathbf{1}_{cc'}(\hat{y}_i^t, \hat{y}_j^t) k(h(\boldsymbol{x}_i^t), h(\boldsymbol{x}_j^t))}{\sum_{i=1}^{N_t} \sum_{j=1}^{N_t} \mathbf{1}_{cc'}(\hat{y}_i^t, \hat{y}_j^t)}$, and $o'_3 = \sum_{i=1}^{N_s} \sum_{j=1}^{N_t} \frac{\mathbf{1}_{cc'}(y_i^s, \hat{y}_j^t) k(h(\boldsymbol{x}_i^s), h(\boldsymbol{x}_j^t))}{\sum_{i=1}^{N_s} \sum_{j=1}^{N_t} \mathbf{1}_{cc'}(y_i^s, \hat{y}_j^t)}$. The element of $\mathbf{1}_{cc'}(y, y')$ is defined as: $\mathbf{1}_{cc'}(y, y') = 1$, if $y = c, y' = c'$; $\mathbf{1}_{cc'}(y, y') = 0$, otherwise. By Combining Equations 9 and 10, Equation 8 can be rewritten as follows:

$$\mathcal{L}'_{ada} = d^{intra} - d^{inter}. \tag{11}$$

To strengthen the transferability of the class discriminative features, we further introduce the adaptation of class-level distribution in the last layer output of the feature extractor. We conduct the adaptation learning of the latent feature $\boldsymbol{z}$:

$$\mathcal{L}''_{ada} = d_f^{intra} - d_f^{inter}, \tag{12}$$

where the definition of d_f^{intra} and d_f^{inter} is similar to d^{intra} and d^{inter}, respectively.

By combining Equations 11 and 12, we can obtain the overall loss of the conditional distributions:

$$\mathcal{L}_{ada} = \mathcal{L}'_{ada} + \mathcal{L}''_{ada} \tag{13}$$

The entire loss can thus be obtained via the equation:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda_1 \mathcal{L}_{clu} + \lambda_2 \mathcal{L}_{ada}, \tag{14}$$

where λ_1 and λ_2 are applied to balance the loss function.

3. EXPERIMENTS

3.1. Datasets

Office-31 [19] consists of three domains: Amazon (A), Dslr (D), and Webcam (W), and includes 4,652 images in 31 classes. **Office-Home** [20] contains around 15,500 images divided into 65 classes. The dataset comprises four domains: Artistic (Ar), Clip Art (Cl), Product (Pr) and Real-World (Rw). **VisDA-C** [21] has two domains and 12 classes where the Synthetic one, consisting of 152,397 synthetic 2D renderings, and Real one, consisting of 55,388 real images.

Table 1. Classification accuracies (%) on Office31 dataset. The bold numbers denote the best results for each column.

	A→W	D→W	W→D	A→D	D→A	W→A	Average
ResNet-50 [22]	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
DAN [2]	80.5±0.4	97.1±0.2	99.6±0.1	78.6±0.2	63.6±0.3	62.8±0.2	80.4
DANN [9]	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
CDAN+E [10]	94.1±0.1	98.6±0.1	**100.0**±0.0	92.9±0.2	71.0±0.3	69.3±0.3	87.7
CAN [4]	94.5±0.3	99.1±0.2	99.8±0.2	95.0±0.3	78.0±0.3	77.0±0.3	90.6
SRDC [23]	95.7±0.2	**99.2**±0.1	**100.0**±0.0	**95.8**±0.2	76.7±0.3	77.1±0.1	90.8
TDFM	**96.1** ±0.3	**99.2** ±0.0	**100.0**±0.0	95.5 ±0.2	**79.2** ±0.3	**78.1** ±0.2	**91.4**

Table 2. Classification accuracies (%) on the VisDA-C dataset. The bold numbers denote the best result.

Methods	ResNet-50 [22]	DAN [2]	DANN [9]	CDAN+E [10]	TAT [11]	TDFM
Average	60.0	63.1	63.7	70.0	71.9	**75.4**

Table 3. Classification results (%) on the Office-Home dataset. The bold numbers denote the best result.

	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Ave.
ResNet-50 [22]	34.9	50.0	58.0	37.4	41.9	46.2	38.5	31.2	60.4	53.9	41.2	59.9	46.1
DAN [2]	43.6	57.0	67.9	45.8	56.5	60.4	44.0	43.6	67.7	63.1	51.5	74.3	56.3
DANN [9]	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
CDAN+E [10]	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	69.3	53.6	82.0	65.8
SRDC [23]	52.3	76.3	81.0	69.5	76.2	78.0	68.7	53.8	81.7	**76.3**	57.1	85.0	71.3
TDFM	**62.2**	**78.1**	**83.2**	**69.7**	**77.9**	**78.6**	**70.1**	**59.7**	**83.4**	75.9	**63.4**	**85.9**	**74.0**

Table 4. Ablation experiments on Office31 and Office-Home dataset. Bold numbers denote the best results for each column.

	A→W	D→W	W→D	A→D	D→A	W→A	Ave.
TDFM (w/o ada)	84.9±0.1	98.2±0.1	100.0±0.0	84.7±0.1	75.3±0.3	73.0±0.1	86.0
TDFM (w/o clu)	94.5±0.3	99.1±0.2	99.8±0.2	95.0±0.3	78.0±0.3	77.0±0.3	90.6
TDFM	**96.1** ±0.3	**99.2** ±0.0	**100.0**±0.0	**95.5** ±0.2	**79.2** ±0.3	**78.1** ±0.2	**91.4**

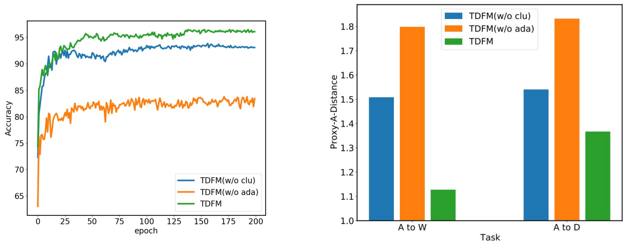
	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	Ave.
TDFM (w/o ada)	53.2	71.3	75.9	62.4	71.2	69.4	61.6	51.1	75.8	68.4	51.3	79.8	66.0
TDFM (w/o clu)	60.3	77.2	79.4	68.5	74.9	72.9	69.6	58.4	79.4	72.7	57.5	82.2	71.1
TDFM	**62.2**	**78.1**	**83.2**	**69.7**	**77.9**	**78.6**	**70.1**	**59.7**	**83.4**	**75.9**	**63.4**	**85.9**	**74.0**

3.2. Implementation Details

We applied ResNet-50 [22], pretrained on ImageNet [24] as the backbone. The network was trained using the mini-batch SGD optimizer. The learning rate annealing strategy was adopted as [4]: $\eta_p = \eta_0(1 + \alpha p)^{-\beta}$, where p denotes the training progress changing from 0 to the maximum number of iterations. For Office-31 and Office-Home, $\alpha = 0.001$, $\beta = 0.75$, while for VisDA-C, $\alpha = 0.001$ and $\beta = 2.25$. η_0 denotes the initial learning rate, 1e-3 for the convolutional layers and 1e-2 for the task-specific FC layer. The tradeoff parameter λ_1 annealing strategy is $\lambda_1 = \lambda'_1(1+0.001*p)^{-0.75}$, where λ'_1 was set to 0.1. The λ_2 was set to 0.3.

3.3. Results

To facilitate fair comparison, the results of other comparison methods are directly quoted from their original papers. Experimental results are presented in Table 1, 2 and 3. In a word, we can observe that our proposed method outperforms the state-of-the-art methods on all transfer tasks, which strongly confirms the effectiveness of our proposed TDFM in mining the transferable discriminative features. TDFM exceeds the excellent results obtained by CAN [4] in terms of its ability, which demonstrates that clustering learning could promote the following of following class-level feature adaptation. Moreover, TDFM also outperforms SRDC [23], which improves the discriminability via clustering learning; this proves that focusing only on data clustering is insufficient to guarantee the transferability of class-level information.



(a) (b)

Fig. 2. (a) Accuracy curve. (b) Distribution discrepancy.

3.4. Analysis

Ablation Studies Table 4 presents the results of our ablation studies on Office-31 and Office-Home datasets. We remove the clustering loss and the class-level adaptation loss from the overall training objective, respectively; the training settings are denoted as TDFM (w/o clu) and TDFM (w/o ada), respectively. TDFM (w/o clu), similar to CAN [4], performs much better than TDFM (w/o ada), which shows the adaptation of class-level features is more important than the clustering learning. TDFM also significantly outperforms TDFM (w/o clu), which verifies the mining of discriminative information via clustering learning plays an important role in this process. Importantly, the ablation experiments reveal that the clustering learning and the adaptation of class-level features can promote each other and work better cooperatively. **Convergence and Distribution Discrepancy** Figure 2 (a) illustrates the test accuracy of TDFM (w/o ada), TDFM (w/o clu) and TDFM on the A→W task. We can see TDFM achieves optimal performance more quickly. We further analyze the proxy $\mathcal{A}$-distance (PAD) [25] on the A→W and A→D tasks as shown in Figure 2 (b). We can observe the PAD of TDFM is smaller than TDFM (w/o ada) and TDFM (w/o clu), which suggests that our features can more effectively reduce the cross-domain gap.

4. CONCLUSION

We develop a novel TDFM approach for UDA. The proposed TDFM incorporates domain-invariant discriminative features learning and class-level features adaptation into a single framework. The domain-invariant discriminative features are achieved via joint learning of supervised classification of the source domain and unsupervised clustering of the target domain. The class-level feature adaptation is obtained via the maximization of inter-class distances and the minimization of intra-class distances. These two procedures work cooperatively to significantly improve the target classification accuracy. Comprehensive experiments demonstrate that TDFM substantially outperforms the state-of-the-art methods.

5. REFERENCES

[1] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira, "Analysis of representations for domain adaptation," in *NeurIPS*, 2007, pp. 137–144.

[2] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan, "Learning transferable features with deep adaptation networks," in *ICML*, 2015.

[3] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan, "Deep transfer learning with joint adaptation networks," in *ICML*, 2017, pp. 2208–2217.

[4] Kang Guoliang, Jiang Lu, Yang Yi, and Hauptmann Alexander, G, "Contrastive adaptation network for unsupervised domain adaptation," in *CVPR*, 2019, pp. 4893–4902.

[5] Yan Hongliang, Li Zhetao, Wang Qilong, Li Peihua, Xu Yong, and Zuo Wangmeng, "Weighted and class-specific maximum mean discrepancy for unsupervised domain adaptation," *TMM*, 2019.

[6] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz, "Central moment discrepancy (cmd) for domain-invariant representation learning," 2017.

[7] Baochen Sun and Kate Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *ECCV*, 2016, pp. 443–450.

[8] Yaroslav Ganin and Victor Lempitsky, "Unsupervised domain adaptation by backpropagation," in *ICML*, 2014.

[9] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky, "Domain-adversarial training of neural networks," *JMLR*, vol. 17, no. 1, pp. 2096–2030, 2016.

[10] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan, "Conditional adversarial domain adaptation," in *NeurIPS*, 2018, pp. 1640–1650.

[11] Hong Liu, Mingsheng Long, Jianmin Wang, and Michael Jordan, "Transferable adversarial training: A general approach to adapting deep classifiers," in *ICML*, 2019, pp. 4013–4022.

[12] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht, "Sliced wasserstein discrepancy for unsupervised domain adaptation," in *CVPR*, 2019, pp. 10285–10295.

[13] Chen Qingchao, Liu Yang, Wang Zhaowen, Wassell Ian, and Chetty Kevin, "Re-weighted adversarial adaptation network for unsupervised domain adaptation," in *CVPR*, 2018, pp. 7976–7985.

[14] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell, "Adversarial discriminative domain adaptation," in *CVPR*, 2017, pp. 7167–7176.

[15] Junyuan Xie, Ross B. Girshick, and Ali Farhadi, "Unsupervised deep embedding for clustering analysis," in *ICML*, 2016, pp. 478–487.

[16] Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang, "Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization," in *ICCV*, 2017, pp. 5747–5756.

[17] Gretton Arthur, Borgwardt Karsten, Rasch Malte, Schoelkopf Bernhard, and Smola Alex, "A kernel two-sample test," *JMLR*, vol. 13, pp. 723–773, 2012.

[18] Van Der Maaten Laurens and Geoffrey Hinton, "Visualizing data using t-sne," *JMLR*, vol. 9, pp. 2579–2605, 2008.

[19] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell, "Adapting visual category models to new domains," in *ECCV*, 2010, pp. 213–226.

[20] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *CVPR*, 2017, pp. 5018–5027.

[21] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko, "Visda: The visual domain adaptation challenge," *arXiv preprint arXiv:1710.06924*, 2017.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[23] Hui Tang, Ke Chen, and Kui Jia, "Unsupervised domain adaptation via structurally regularized deep clustering," in *CVPR*, 2020, pp. 8722–8732.

[24] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.

[25] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan, "A theory of learning from different domains," *Machine Learning*, vol. 79, pp. 151–175, 2010.