Multi-Source Domain Adaptation with Distribution Fusion and Relationship Extraction

1st Keqiuyin Li
Faculty of Engineering and IT
University of Technology Sydney
Sydney, NSW, Australia
keqiuyin.li@student.uts.edu.au

2nd Jie Lu
Faculty of Engineering and IT
University of Technology Sydney
Sydney, NSW, Australia
jie.lu@uts.edu.au

3rd Hua Zuo
Faculty of Engineering and IT
University of Technology Sydney
Sydney, NSW, Australia
hua.zuo@uts.edu.au

4th Guangquan Zhang
Faculty of Engineering and IT
University of Technology Sydney
Sydney, NSW, Australia
guangquan.zhang@uts.edu.au

***Abstract*—Transfer learning is gaining increasing attention due to its ability to leverage previously acquired knowledge to assist in completing a prediction task in a similar domain. While many existing transfer learning methods deal with single source and single target problem without considering the fact that a target domain maybe similar to multiple source domains, this work proposes a multi-source domain adaptation method based on a deep neural network. Our method contains common feature extraction, specific predictor learning and target predictor estimation. Common feature extraction explores the relationship between source domains and target domain by distribution fusion and guarantees the strength of similar source domains during training, something which has not been well considered in existing works. Specific predictor learning trains source tasks with cross-domain distribution constraint and cross-domain predictor constraint to enhance the performance of single source. Target predictor estimation employs relationship extraction and selective strategy to improve the performance of the target task and to avoid negative transfer. Experiments on real-world visual datasets show the performance of the proposed method is superior to other deep learning baselines.**

I. Introduction

Transfer learning [1], [2] has been explored for many years and gained success in a variety of applications in real-world scenarios, such as nature language processing, computer vision and biological problems. The main goal of transfer learning is to improve the performance of the target task using the knowledge learned from a similar source domain, since the target training data is often difficult to collect or expensive to label, especially where medicine is involved. Employing different transfer information, it can be divided into four categories: instance-based method [3], feature-based method [4], parameter-based method [5] and relationship-based method [6].

Based on feature and parameter transformation, one popular technique to achieve transfer learning is domain adaptation [7]–[9], which aims to tackle domain shift by reducing the discrepancy of distributions between the source and target domains in a latent feature space, including adaptations of marginal distribution [10], conditional distribution [11] and joint distribution [12].

Zellinger et al. [13] proposed a novel metric function named central moment discrepancy to measure the distance between probability distributions which can be solved without highly complex and costly kernel matrix computations. Zuo et al. [14], [15] used fuzzy system and granular computing to achieve regression transfer in homogeneous and heterogeneous feature spaces. Benefiting from the development of deep learning, recent surveys employed deep neural networks to extract common features of source and target domains. Based on adversarial learning, Long et al. [16] proposed joint adaptation networks which used joint maximum mean discrepancy to align the joint distributions across domains in hidden specific layers. Liu et al. [17] and Jang et al. [18] explored the transferability of deep feature representations and provided experiments and theory analysis on what and where to transfer in deep networks.

However, all these studies focused on single source domain adaptation while, in practice, a target domain can be similar to multiple source domains, and they may be different from each other but could provide richer information for transfer. It is easy to see why multi-source domain adaptation now attracts greater attention. Zhao et al. [19] and Wen et al. [20] measured the discrepancy by $\mathcal{H}$-divergence and proposed an adversarial strategy based deep framework to solve multiple sources domain adaptation both for classification and regression problems. Guo et al. [21] proposed a mixture-of-experts method to do text classification and speech tagging with multiple sources, where Mahalanobia distance and confidence score were used to extract the relationship between each source domain and target domain. Ding et al. [22] tried to achieve domain adaptation with multiple incomplete source domains via low-rank matrix which could recover missing labels in source domains based on latent features from a target

This work was supported by the Australian Research Council under grant FL190100149.(Corresponding author: Jie Lu.)

domain. Redko et al. [23] and Xu et al. [24] solved multiple sources domain adaptation under target shift and category shift, where source labels might not completely share labels of target domain or share labels with different proportions. Zhu et al. [25] proposed a two-stage alignment framework for multiple sources domain adaptation, in which domain-specific distribution alignment was used to reduce discrepancy between source domains and target domain; domain-specific classifier alignment was used to reduce difference among all classifiers.

The main idea of all described multi-source domain adaptation begins with extracting common features of source domains and target domain before training specific predictors of each source domain, finally combining all specific predictors as the target predictor. The popular combination rules include the average of source predictors and weighted average of source predictors. Although some of these approaches have considered the connections between different source domains and target domain, they still treat each source domain equally during training. It has been proven that adding irrelevant source samples may lead to negative transfer and reduce the performance of the learning task [26]. Measuring relationship among source domains and target domain is necessary to avoid negative performance. In this paper, in order to learn weights of multiple sources to get high performance of predictor in target domain, following work [25] and domain matching method proposed by Li et al. [27], we proposed a deep neural network based multi-source domain adaptation approach with distribution fusion and relationship extraction. This measures relatedness among sources and target during training and guarantees that more similar sources contribute more to target task learning. Our main contributions are as follows:

- We propose a method using distribution fusion and relationship extraction to guarantee the strength of similar source domains during learning a target task, something which is not well considered in existing methods;
- We use a selective strategy to choose the best performance of the target task and to avoid negative transfer of irrelevant sources.

The structure of this paper is designed as follows: Section II describes the proposed model. Section III carries a series of experiments on real-world datasets and analyses the results. Our conclusion is given in section IV.

II. Proposed Method

The proposed method contains the following three parts: (1) common feature extraction where pre-trained deep neural network, fine-tuning operation and distribution fusion strategy are employed to collect robust features; (2) specific predictor learning where specific domain adaptation and predictor of each source domain are learned with cross-domain constraints; (3) target predictor estimation where relationship extraction and selective strategy are used to obtain target predictor without negative transfer. The whole procedure is showed in Fig. 1.

A. Common Feature Extraction

In general, domain adaptation is an unsupervised learning task. For single source domain adaptation, given source domain $\mathcal{D}_s = \{(x_s^i, y_s^i)\}_{i=1}^{n_s}$ and target domain $\mathcal{D}_t = \{x_t^j\}_{j=1}^{n_t}$, where $x_s, x_t \in \mathcal{X}$ represent samples, $y_s \in \mathcal{Y}$ indicates corresponding label of x_s and n_s, n_t indicate the number of samples in source domain and target domain respectively. The main step is mapping distributions of source domain and target domain. One popular method to achieve this is maximum mean discrepancy (MMD) [10] which can be formulated as:

$$\mathcal{MMD}(X_s, X_t) = \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(x_s^i) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(x_t^j) \right\|_{\mathcal{H}}^2, \quad (1)$$

where $\|\cdot\|_{\mathcal{H}}$ is reproducing kernel Hillbert space (RKHS) norm, ϕ is kernel-induced feature map.

In our setting, when there are K source domains $\{\mathcal{D}_{s_k}\}_{k=1}^{K}$, to map original samples into a common feature space where domain adaptation can be achieved, we use a pre-trained deep neural network F_p to collect latent features of all domains. At the same time, fine-tuning block F_{fc}, which is optimized by distribution fusion block F_d and specific domain adaptation block (we will explain it in detail in section II-B), is added to fine-tune convolution layers in order to extract more robust latent representations. The extracted latent features are then used to complete specific domain adaptation. Common feature extraction can be formulated as:

$$\begin{aligned} f_{c_{s_k}}^i &= F_{fc}(F_p(x_{s_k}^i)), \\ f_{c_t}^j &= F_{fc}(F_p(x_t^j)), \\ i = 1, 2, \ldots, n_{s_k}, j &= 1, 2, \ldots, n_t. \end{aligned} \quad (2)$$

Rewrite (1) according to the proposed common feature extractor and express it as multi-source setting, that is:

$$\begin{aligned} &\mathcal{MMD}(X_{s_k}, X_t) = \\ &\left\| \frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} \phi(f_{c_{s_k}}^i) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(f_{c_t}^j) \right\|_{\mathcal{H}}^2. \end{aligned} \quad (3)$$

Before fine-tuning common features extracted by (2) using specific domain adaptation block and estimating RKHS distance between each source domain and target domain, we first add the distribution fusion block to optimize the parameters of the common feature fine-tuning block. So, corresponding loss function can be written as:

$$\begin{aligned} \mathcal{L}_{F_d} &= \mathcal{MMD}(\hat{X}_s, X_t) \\ &= \left\| F_d\big(\frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} \phi(f_{c_{s_k}}^i)\big) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(f_{c_t}^j) \right\|_{\mathcal{H}}^2 \\ &= \left\| \sum_{k=1}^{K} \omega_k \big(\frac{1}{n_{s_k}} \sum_{i=1}^{n_{s_k}} \phi(f_{c_{s_k}}^i)\big) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(f_{c_t}^j) \right\|_{\mathcal{H}}^2, \end{aligned} \quad (4)$$

where k is the kth source domain, $\hat{X}_s$ means weighted combination of source domains, which contributes the update

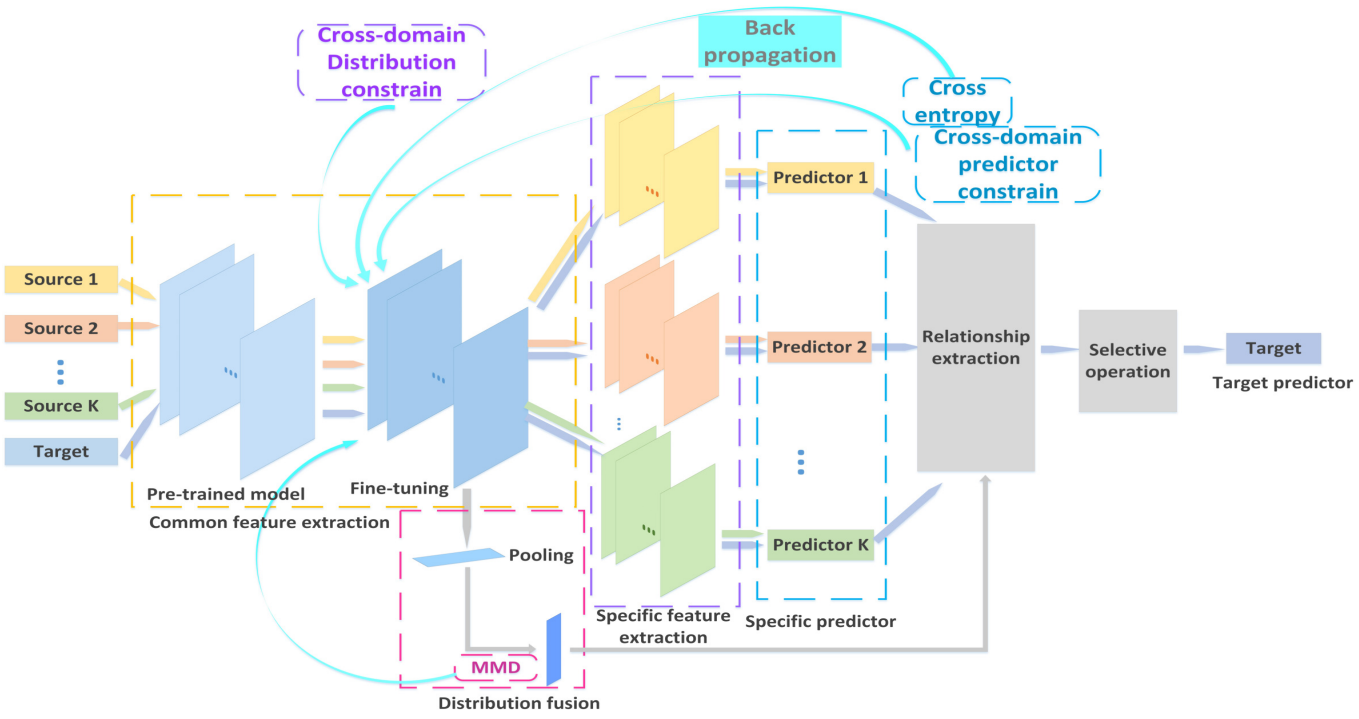

Fig. 1. The whole procedure of the proposed method (better view in color): the yellow dashed box shows common feature extraction; the red dashed box shows distribution fusion; the purple dashed box shows specific feature extraction and the blue dashed box shows specific predictor learning. The rounded dashed boxes which are in the same color as above mentioned dashed boxes represent corresponding losses. Gary boxes mean relationship extraction and selective operation.

of parameters of common feature fine-tuning block and gives more similar sources larger weights for subsequent training.

B. Specific Predictor Learning

After extracting latent features of all domains, which represent common knowledge of the sources and the target, specific domain adaptation blocks are used to learn specific features and predictors of each source domain simultaneously. The specific domain adaptation block of each source domain contains three fine-tuning layers and a fully connected layer. The specific fine-tuning layers $\{F_{fs}^k\}_{k=1}^K$ are added to learn specific latent features of corresponding kth source domain and the target domain, which guarantee that the predictor trained on the source domain can be used on the target domain. The fully connected layer is employed to learn high performance predictors of each source domain with cross-domain constraints. The final specific features of each source domain are:

$$\begin{aligned} f_{s_{s_k}}^i &= F_{fs}^k(f_{c_{s_k}}^i), \\ f_{s_{t_k}}^j &= F_{fs}^k(f_{c_t}^j), \\ i = 1, 2, \ldots, n_{s_k}, j &= 1, 2, \ldots, n_t. \end{aligned} \tag{5}$$

$f_{s_{t_k}}$ means feeding common representation f_{c_t} into the specific feature extractor of kth source domain.

To optimize F_{fs}, here we use the same strategy proposed in [25], cross-domain constraints contain two parts: cross-domain distribution and cross-domain predictor. It can be expressed as:

$$\begin{aligned} \mathcal{L}_{dc} =& \frac{1}{K}\sum_{k=1}^{K}\mathcal{MMD}(X_{s_k}, X_t) = \\ & \frac{1}{K}\sum_{k=1}^{K}\left\| \frac{1}{n_{s_k}}\sum_{i=1}^{n_{s_k}}\phi(f_{s_{s_k}}^i) - \frac{1}{n_t}\sum_{j=1}^{n_t}\phi(f_{s_{t_k}}^j) \right\|_{\mathcal{H}}^2, \end{aligned} \tag{6}$$

and

$$\begin{aligned} \mathcal{L}_{cp} =& \frac{2}{K(K-1)}\sum_{k_1=1}^{K-1}\sum_{k_2=k_1+1}^{K} \\ & \left(\frac{1}{n_t}\sum_{j=1}^{n_t}\left|F_{pres}^{k_1}(f_{s_{t_{k_1}}}^j) - F_{pres}^{k_2}(f_{s_{t_{k_2}}}^j)\right|\right). \end{aligned} \tag{7}$$

where F_{pres} represents a specific predictor of the source domain. By minimizing (6) and (7), specific domain-invariant features can be learned by taking generated cross-domain distributions into consideration. In addition, target samples close to class edges are more probably to get the same prediction results. It has been proven that adding these cited cross-domain constrains brings positive effects [25].

The final specific predictor F^k_{pres} of each source domain is controlled by cross entropy, that is:

$$\mathcal{L}_{pres} = \sum_{k=1}^{K}(-\sum_{i=1}^{n_{s_k}} y^i_{s_k} \log\left(F^k_{pres}(f^i_{s_{s_k}})\right). \tag{8}$$

The total loss function of proposed model is:

$$\mathcal{L} = \mathcal{L}_{pres} + \lambda\mathcal{L}_{F_d} + \gamma\mathcal{L}_{dc} + \mu\mathcal{L}_{cp}. \tag{9}$$

C. Target Predictor Estimation

After obtaining specific predictors of source domains, it is important to choose an appropriate combination rule to estimate the prediction function of the target domain. Since we have learned a relationship between latent features of target domain and the combination of weighted source domains using distribution fusion block, here we use the learned weights to extract the connection among source predictors and target predictor. To further strengthen the performance of outputs, the predictive accuracy of single source domain adaption is taken into account. The final target predictor is:

$$F_{pret} = \sum_{k=1}^{K} \alpha_k F^k_{pres}, \tag{10}$$

where

$$\begin{aligned} \alpha_k &= \frac{sigmoid(\omega_k)\cdot w_k}{\sum_{k'=1}^{K} sigmoid(\omega_{k'})\cdot w_{k'}}, \\ w_k &= \frac{accu_k}{\sum_{k'=1}^{K} accu_{k'}}. \end{aligned} \tag{11}$$

ω_k is the parameter of distribution fusion block which indicates the similarity between kth source domain and target domain and $accu_k$ means the prediction accuracy of kth source domain predictor.

To avoid negative transfer, source selective strategy is employed to obtain a high performance of final outputs of the target domain. Our proposed method compares single source outputs directly with predictions of target predictor (10). If there is negative transfer, the final target predictor equals to the single source predictor which achieves the best performance.

The whole method is summarized as follows.

Algorithm 1 Proposed multi-source domain adaptation

Input: Source domains $\{\mathcal{D}_{s_k}\}_{k=1}^K$, target domain $\mathcal{D}_t$, training iteration $\mathcal{I}$, pre-trained model F_p, ground truth label Y_t.
for $\epsilon = 1$, $\epsilon < \mathcal{I}$, $\epsilon++$, **do**
  $\{(x^i_{s_k}, y^i_{s_k})\}_{i=1}^m \leftarrow$ randomly collect m batch pairs from each source domain $\mathcal{D}_{s_k}$;
  $\{x^i_t\}_{i=1}^m \leftarrow$ randomly collect m batch pairs from target domain $\mathcal{D}_t$;
  $\{f_{c_{s_k}}\}_{k=1}^K \leftarrow F_{fc}(F_p(x_{s_k}))$;
  $f_{c_t} \leftarrow F_{fc}(F_p(x_t))$ according to (2);
  $\mathcal{L}_{F_d} \leftarrow \mathcal{MMD}(\hat{X}_s, X_t)$ according to (4);
  $\{f_{s_{s_k}}, f_{s_{t_k}}\}_{k=1}^K \leftarrow \{F^k_{fs}(f_{c_{s_k}}), F^k_{fs}(f_{c_t})\}_{k=1}^K$ according to (5);
  $\{Y_{s_k}\}_{k=1}^K \leftarrow \{F^k_{pres}(f_{s_{s_k}}\}_{k=1}^K$, estimate labels of each source domain;
  Compute $\mathcal{L}_{dc}$, $\mathcal{L}_{cp}$ and $\mathcal{L}_{pres}$ according to (6), (7) and (8);
  $\{Y_{t_k}\}_{k=1}^K \leftarrow \{F^k_{pres}(f_{s_{t_k}})\}_{k=1}^K$, estimate single source predictions of target domain respectively;
  Compute target label Y_{t_m} with multiple sources according to (10).
  Compare accuracy of single source prediction and multi-source prediction with ground truth label and use selective strategy to return best performance.
  Update common feature fine-tuning block F_{fc}, distribution fusion block F_d, specific feature fine-tuning blocks $\{F^k_{fs}\}_{k=1}^K$, specific predictors $\{F^k_{pres}\}_{k=1}^K$ by minimizing (9).
end for
Output: Feature fine-tuning block F_{fc}, distribution fusion block F_d, specific feature fine-tuning blocks $\{F^k_{fs}\}_{k=1}^K$, specific predictors $\{F^k_{pres}\}_{k=1}^K$, predictive label $\hat{Y}_t$.

III. Experiments

A. Datasets and Setting

To evaluate the efficiency of the proposed multi-source domain adaptation method, we take a series of experiments with some existing state-of-the-art baselines on datasets ImageCLEF-DA and Office-31.

ImageCLEF-DA is a balanced dataset which contains 1800 images from 12 categories shared by datasets Caltech-256 (C), ImageNet ILSVRC 2012 (I) and Pascal VOC 2012 (P) and each dataset is regarded as a domain. Every category contains 50 images and there are 600 images in each domain. We test the proposed model by building three tasks: $I, C \rightarrow P$; $I, P \rightarrow C$; $C, P \rightarrow I$.

Office-31 is an unbalanced dataset which comprises 4110 images from 31 categories shared by datasets Amazon (A), Webcam (W) and DSLR (D) and, again, each dataset is regarded as a domain. Amazon contains 2817 images from amazon.com, Webcam has 795 images taken by web camera and DSLR holds 498 images taken by digital SLR camera. The number of images in each category is different. We test the proposed model by building three tasks: $A, W \rightarrow D$; $A, D \rightarrow W$; $D, W \rightarrow A$.

Since there are few methods dealing with multi-source domain adaptation on real-world visual recognition, we compare the proposed method with the two most recent multiple source domain adaptation methods: Deep Cocktail Network (DCTN) [24] and Multiple Feature Spaces Adaptation Network (MFSAN) [25], along with other efficient single domain adaptation methods: ResNet [28], Deep Adaptation Network (DAN) [29], Deep Coral (D-CORAL) [30] and Reverse Gradient (RevGrad) [31]. For these single source domain adaptation models, we use two standards which are the same as used in previous surveys: simply combining all source domains as one domain, that is "Source Combine"; returning the best single source domain adaptation results, that is "Single Best". For multi-source domain adaptation models, the standard is labelled "Multi-Source", which indicates transfer results of multiple sources.

We choose $ResNet50$ as a pre-train model; common feature fine-tuning block contains 3 convolution layers, with kernel sizes of 1×1, 3×3, 1×1; distribution fusion block contains 1 fully connected layer; specific feature fine-tuning block includes 3 convolution layers, with kernel sizes of 1×1, 3×3,

1×1; specific predictor comprises 1 fully connected layer. The optimization algorithm is stochastic gradient descent (SGD), training batch size is 32; learning rate is 0.01; momentum is 0.9 and weight decay is $5e-4$. Trade-off parameters λ, γ and μ are set as the common value 0.5.

B. Comparison and Analysis

Since all baselines reported in previous survey are best performance, we also choose the best transfer results of the proposed method for evaluation. Tables I and II show the results on ImageCLEF-DA and Office-31 respectively. Tables III and IV show the results of the proposed method trained with and without distribution fusion or common feature fine-tuning on the two datasets.

TABLE I
COMPARISON OF CLASSIFICATION ACCURACY (%) OF THE PROPOSED WITH RESNET, DAN, D-CORAL, REVGRAD, DCTN AND MFSAN ON IMAGECLEF-DA DATASET

Standards	Method	I, C-P	I, P-C	P, C-I	Avg
Single Best	ResNet	74.8	91.5	83.9	83.4
	DAN	75.0	93.3	86.2	84.8
	D-CORAL	76.9	93.6	88.5	86.3
	RevGard	75.0	**96.2**	87.0	86.1
Source Combine	DAN	77.6	93.3	92.2	87.7
	D-CORAL	77.1	93.6	91.7	87.5
	RevGard	77.9	93.7	91.8	87.8
Multi-Source	DCTN	75.0	95.7	90.3	87.0
	MFSAN	79.1	95.4	93.6	89.4
	Proposed	**79.5**	95.8	**93.7**	**89.7**

TABLE II
COMPARISON OF CLASSIFICATION ACCURACY (%) OF THE PROPOSED WITH RESNET, DAN, D-CORAL, REVGRAD, DCTN AND MFSAN ON OFFICE31 DATASET

Standards	Method	A, W-D	A, D-W	W, D-A	Avg
Single Best	ResNet	99.3	96.7	62.5	86.2
	DAN	99.5	96.8	66.7	87.7
	D-CORAL	**99.7**	98.0	65.3	87.7
	RevGard	99.1	96.9	68.2	88.1
Source Combine	DAN	99.6	97.8	67.6	88.3
	D-CORAL	99.3	98.0	67.1	88.1
	RevGard	**99.7**	98.1	67.6	88.5
Multi-Source	DCTN	99.3	98.2	64.2	87.2
	MFSAN	99.5	98.5	72.7	90.2
	Proposed	99.6	**98.7**	**73.1**	**90.5**

It can be seen that the Source Combine results are better than Single Best results generally, getting a higher average accuracy than single source domain. Although the Single Best results of transfer tasks $I, P \rightarrow C$ using RevGard and $A, W \rightarrow D$ using D-CORAL outperform other methods and standards, enriching data still has a positive influence on transfer performance in most cases. Multi-Source results commonly achieve greater degree than Source Combine except for DCTN. That may result from different problem settings, where DCTN is directly aimed at dealing with domain adaptation with multiple sources and category shift. However, in most multi-source domain adaptation surveys, there is no category shift in source domains. The proposed method obtains the highest classification accuracy than most baselines and standards. This indicates that exploring any relationship between source domains and target domain is important, and guaranteeing the strength of more similar sources is worthy undertaking to achieve the desired performance.

TABLE III
COMPARISON OF CLASSIFICATION ACCURACY (%) OF PROPOSED METHOD ON DATASET IMAGECLEF-DA: M1: TRAINED WITH COMMON FEATURE FINE-TUNING BUT WITHOUT DISTRIBUTION FUSION; M2: TRAINED WITH DISTRIBUTION FUSION BUT WITHOUT COMMON FEATURE FINE-TUNING; M3: TRAINED WITH COMMON FEATURE FINE-TUNING AND DISTRIBUTION FUSION

Standards	Method	I, C-P	I, P-C	P, C-I	Avg
Proposed	M1	77.5	92.2	93.3	87.7
	M2	**79.5**	**96.2**	**94.0**	**89.9**
	M3	**79.5**	95.8	93.7	89.7

TABLE IV
COMPARISON OF CLASSIFICATION ACCURACY (%) OF PROPOSED METHOD ON DATASET OFFICE-31: M1: TRAINED WITH COMMON FEATURE FINE-TUNING BUT WITHOUT DISTRIBUTION FUSION; M2: TRAINED WITH DISTRIBUTION FUSION BUT WITHOUT COMMON FEATURE FINE-TUNING; M3: TRAINED WITH COMMON FEATURE FINE-TUNING AND DISTRIBUTION FUSION

Standards	Method	A, W-D	A, D-W	W, D-A	Avg
Proposed	M1	99.6	97.9	72.0	89.8
	M2	**100.0**	98.6	72.7	90.4
	M3	99.6	**98.7**	**73.1**	**90.5**

Tables III and IV test the effects of common feature fine-tuning and distribution fusion. It can be seen that training the proposed method with distribution fusion always achieves better performance than training the method with common feature fine-tuning only. For dataset ImageCLEF-DA which contains a small number of samples, adding depth of neural network may harm the performance. In contrast, for large scale dataset like Office-31, adding fine-tuning operation with distribution fusion can improve the performance.

IV. CONCLUSION AND FUTURE STUDY

In this paper, we proposed a domain adaptation method with multiple sources based on a pre-trained deep neural network. Our method uses fine-tuning operation and distribution fusion to explore the relationship between source domains and target domain while simultaneously guaranteeing that more similar sources contribute correspondingly more during training. Then the cross-domain distribution constraint and cross-domain predictor constraint are used to learn specific predictors of source domains. Target predictor is estimated by combining source predictors using relationship extraction. In conclusion, the selective strategy is used to avoid negative transfer. A series of experiments are designed and implemented on two real-world datasets. For more vigorous examination, the effects of

distribution fusion and fine-tuning are also tested. The results demonstrate the efficiency of our proposed.

The proposed method can still be improved and there are remaining problems that should be solved in future study. Initially, we currently use a fully connected layer to extract relationships between source domains and target domain, which requires significant memory space in a computer. Replacing it with convolution layers may improve calculation efficiency. This still needs to be explored in future research.

REFERENCES

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[2] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: a survey," *Knowledge-Based Systems*, vol. 80, pp. 14–23, 2015.

[3] B. Tan, Y. Song, E. Zhong, and Q. Yang, "Transitive transfer learning," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 1155–1164.

[4] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1859–1867.

[5] Y. Wei, Y. Zhu, C. W.-k. Leung, Y. Song, and Q. Yang, "Instilling social to physical: Co-regularized heterogeneous transfer learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

[6] J. Wang *et al.*, "Everything about transfer learning and domain adapation," http://transferlearning.xyz.

[7] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Advances in Neural Information Processing Systems*, 2007, pp. 137–144.

[8] W. M. Kouw, "An introduction to domain adaptation and transfer learning," *arXiv preprint arXiv:1812.11806*, 2018.

[9] L. Zhang, "Transfer adaptation learning: A decade survey," *arXiv preprint arXiv:1903.04687*, 2019.

[10] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2010.

[11] K. Saito, Y. Ushiku, and T. Harada, "Asymmetric tri-training for unsupervised domain adaptation," in *Proceedings of International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2988–2997.

[12] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2200–2207.

[13] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz, "Central moment discrepancy (cmd) for domain-invariant representation learning," *arXiv preprint arXiv:1702.08811*, 2017.

[14] H. Zuo, G. Zhang, W. Pedrycz, V. Behbood, and J. Lu, "Granular fuzzy regression domain adaptation in takagi–sugeno fuzzy models," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 847–858, 2017.

[15] H. Zuo, J. Lu, G. Zhang, and W. Pedrycz, "Fuzzy rule-based domain adaptation in homogeneous and heterogeneous spaces," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 2, pp. 348–361, 2018.

[16] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proceedings of International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2208–2217.

[17] H. Liu, M. Long, J. Wang, and M. I. Jordan, "Towards understanding the transferability of deep representations," *arXiv preprint arXiv:1909.12031*, 2019. [Online]. Available: https://arxiv.org/pdf/1909.12031

[18] Y. Jang, H. Lee, S. J. Hwang, and J. Shin, "Learning what and where to transfer," *arXiv preprint arXiv:1905.05901*, 2019.

[19] H. Zhao, S. Zhang, G. Wu, J. M. Moura, J. P. Costeira, and G. J. Gordon, "Adversarial multiple source domain adaptation," in *Advances in Neural Information Processing Systems*, 2018, pp. 8559–8570.

[20] J. Wen, R. Greiner, and D. Schuurmans, "Domain aggregation networks for multi-source domain adaptation," *arXiv preprint arXiv:1909.05352*, 2019.

[21] J. Guo, D. J. Shah, and R. Barzilay, "Multi-source domain adaptation with mixture of experts," *arXiv preprint arXiv:1809.02256*, 2018.

[22] Z. Ding, M. Shao, and Y. Fu, "Incomplete multisource transfer learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 2, pp. 310–323, 2016.

[23] I. Redko, N. Courty, R. Flamary, and D. Tuia, "Optimal transport for multi-source domain adaptation under target shift," *arXiv preprint arXiv:1803.04899*, 2018.

[24] R. Xu, Z. Chen, W. Zuo, J. Yan, and L. Lin, "Deep cocktail network: Multi-source unsupervised domain adaptation with category shift," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3964–3973.

[25] Y. Zhu, F. Zhuang, and D. Wang, "Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5989–5996.

[26] Y. Mansour, M. Mohri, and A. Rostamizadeh, "Domain adaptation with multiple sources," in *Advances in Neural Information Processing Systems*, 2009, pp. 1041–1048.

[27] Y. Li, D. E. Carlson *et al.*, "Extracting relationships by multi-domain matching," in *Advances in Neural Information Processing Systems*, 2018, pp. 6798–6809.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[29] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," *arXiv preprint arXiv:1502.02791*, 2015.

[30] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in *Proceedings of the European Conference on Computer Vision*. Springer, 2016, pp. 443–450.

[31] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proceedings of International Conference on International Conference on Machine Learning*. JMLR. org, 2015, pp. 1180–1189.