Debiased Prototype Network for Adversarial Domain Adaptation

Chunwei Wu[1], Guitao Cao*,[1]
[1]*Shanghai Key Laboratory of Trustworthy Computing*
East China Normal University
Shanghai 200062, China
Corresponding Author:
Guitao Cao(*gtcao@sei.ecnu.edu.cn)

Wenming Cao[2]
[2]*College of Information Engineering*
Shenzhen University
Shenzhen 518060, China
Co-Corresponding Author:
Wenming Cao(wmcao@szu.edu.cn)

Hong Wang[3], He Ren[3]
[3]*Ninth Research Office*
Shanghai Research Institute of Microwave Equipment
Shanghai 200062, China
Hong Wang(18016251333@189.cn)
He Ren(rh45@163.com)

2021 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-3900-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/IJCNN52387.2021.9533346

***Abstract*—Domain adaptation is an important and challenging task. Existing adversarial domain adaptation methods explore the relationship between the source and target domains, with the knowledge learned in the source domain supporting the target domain task. The quality of the knowledge will affect the task performance of the transfer, i.e., the higher the quality of the knowledge, the better the transfer task performance. To obtain better domain-invariant knowledge, we extract domain-invariant semantic information over the unit sphere via the prototype network. With the help of geometric constraints from the hypersphere, the features can be more tightly clustered with the estimated prototype (representatives of each class). Adaptation is achieved by adversarial learning to align the domain distribution, which enhances the transferability of the learned features and obtains the basic prototype. Since the basic prototypes dominantly computed from the source domain are biased against the expected domain-invariant prototype, a debiased method is further proposed to obtain the domain-invariant prototypes. Specifically, our method diminishes the intra- and inter- class bias to achieve the class-level alignment. Extensive experiments demonstrate that our model achieves state-of-the-art performance on several domain adaptation benchmark datasets. Our code is available at https://github.com/Chunweiwu-source/DPN.**

***Index Terms*—Domain adaptation, Prototype Network, Debiased method**

I. Introduction

Deep learning has significantly improved performance over traditional machine learning methods by extracting knowledge from large-scale data. However, collecting and labeling samples with the same distribution as the test set is expensive. When the training and test sets' distributions differ, the model obtained from the training set cannot achieve satisfactory prediction results on the test set. Unsupervised Domain adaptation (UDA) [1] is a machine learning technique under the condition that the training (source) and test (target) sets do not satisfy independent identical distribution. The core of UDA is to find the similarity between different domains, such that transfer the knowledge obtained in the source to the target domain.

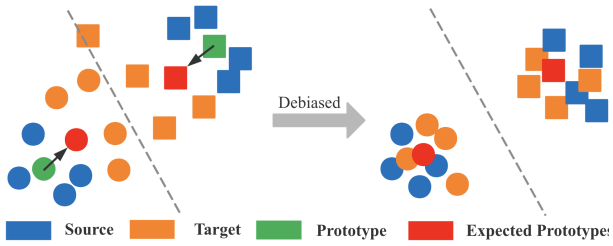


Fig. 1. Motivation of our method. In this figure, the color means the domain, and the shape means the class. We propose Debiased Prototype Network(DPN) that combines adversarial learning to compute a prototype for each class. DPN rectifies the basic prototype to find a more appropriate domain-invariant prototype (black arrow), thus achieving intra-class compactness and inter-class separability for the source and target domains. *Best viewed in color.*

An essential challenge in unsupervised domain adaptation is how to find a domain-invariant representation of the data [2], [3]. Previous domain adaptation methods measure the distribution differences between domains by the Maximum Mean Discrepancy (MMD) [4], [5], higher-order moments [6], [7]. As an extension of the minimizing divergence method, Yu et al. [8] directly operated on the cone of symmetric positive semi-definite (SPS) matrix to avoid the estimation of the underlying distributions. Recently, adversarial learning [9] has been successfully embedded into deep networks to reduce distribution discrepancy. The goal of domain-adversarial learning is to obtain domain-invariant representations, through encouraging domain confusion with an adversarial objective.

However, as discussed in Gu et al. [10], current domain adaptive approaches still face great challenges, including the design of effective embedding space to disentangle the features and leverage pseudo-labels in a more robust manner. Once the domain information is not completely removed, samples are distorted on the feature space, leading to under transfer (under-fitting) or negative transfer (over-fitting) [11]. To address the

978-0-7381-3366-9/21/$31.00 ©2021 IEEE

aforementioned issues, Cai et al. [11] attempted to extract the domain-invariant semantic information in the latent disentangled semantic representation of the data, while Chen et al. [12] proposed an Easy-to-Hard transfer strategy that gradually creates more reliable pseudo-labeled samples. These methods focus on exploiting the similarity between the different domains more sufficiently to reduce the generalization error.

In this paper, we propose a novel approach for UDA that tackle these two challenges in a unified model and significantly improves the accuracy of deep classifiers on domain adaptation task. Our goal is to find the expected domain-invariant prototypes which have the maximum cosine similarity to the source and target feature vectors in the same class. A prototype network based on cosine similarity learns the domain-invariant representations by domain-adversarial learning. In Debiased Prototype Network (DPN), we train a feature extractor with a cosine similarity based classifier on a unit sphere using labeled source domain samples. The cosine similarity based classifier has a strong ability that drives the feature extractor to learn discriminative features, and makes domain adaptation easier due to the geometric constraints of the hypersphere. Note that we project the samples onto the unit spherical, where the feature vectors of the same class are clustered more closely together. At the adaptation stage, we align the marginal distributions by adversarial training and compute the cosine similarity of basic prototypes and target feature vectors for classification. Since the neglect of conditional distribution, the basic prototypes learned from the source domain may not confidently distinguish the target samples.

To estimate domain-invariant prototypes, we propose a debiased method for reduced the intra- and inter- class bias, as shown in Fig. 1. The intra-class bias is the distance between the expectedly domain-invariant prototype and the prototype actually computed from the source samples of the class. We adopt the pseudo-labeling strategy to add target samples with high prediction confidence into our model's learning to reduce it. The inter-class bias refers to the discrepancy between different classes, which we maximize by conditional entropy regularization. The inner reason lies that conditional entropy is a measure of class overlap. If a sample can get a low prediction by conditional entropy, it can be considered as far from the decision boundary. Experiments will justify the effectiveness of the proposed debiased method. The main contributions of this paper are as follows:

- We propose a novel debiased prototype network to improve the transferability of image representations for adversarial domain adaptation. Our approach demonstrates that the prototype network extracts invariant features more efficiently.
- We propose a debiased method which is utilized to reduce the intra- and inter- class bias for prototype network.
- We have conducted extensive experiments on popular benchmark datasets and achieved state-of-the-art performance. The experimental results show that our proposed method can lead to a relatively large improvement in classification accuracy.

II. Related Work

A. Adversarial Domain Adaptation

Adversarial learning methods have been shown remarkably effective for UDA. The adversarial learning based methods can be classified into single-adversarial domain adaptation [13], [14] and multi-adversarial domain adaptation [15]–[17] according to the adversarial approach. In addition, there are some methods [18] that do not improve on the adversarial approach but introduced the attention mechanism into the adversarial domain adaptation. Wang et al. [14] proposed a self-adaptive re-weighted adversarial domain adaptation (SRADA) that tries to enhance domain alignment from the perspective of conditional distribution. Methods in [15], [16] reduced domain shift in raw pixel space by transforming source domain image to the style of target domain. Long et al. [17] presented a conditional adversarial domain adaptation (CDAN) that conditions the adversarial adaptation models on discriminative information conveyed in the classifier predictions. Wang et al. [18] applied local attention and global attention to focus the adaptation model on transferable regions or images.

These deep domain adaptation methods gain huge improvement, however, deep representations can only reduce, but not remove, the cross-domain discrepancy according to some recent research [19], [20]. To tackle this problem, our approach seamlessly integrate prototype learning and domain adaptation to learn more representative and transferable features.

B. Prototype Learning

To obtain the most discriminative features, various prototype networks have been explored. Unlike softmax-based CNN, prototype networks [21], [22] learn a metric space where the labeling is done by calculating the distance between feature vectors and each class's prototype. It is reported that prototype networks have achieved promising performance in open-set recognition [21], [23], few-shot learning [24]–[26] and zero-shot Learning [27]–[29]. Recently, Saito et al. [30] proposed a more universally applicable domain adaptation framework, which base on prototype networks, to handle arbitrary category shift. Xu et al. [31] exploited spatial prototypes of each class learned from labeled source data to assign the target samples a pseudo label.

As previous works [29], [32] have shown the importance of prototype rectification, we propose a novel debiased method to reduce the domain shift. Specifically, we employ the cosine similarity based classifier to learn the discriminative embedding space, making the computed prototypes more robust to represent a class. Considering the negative transfer caused by the distribution of target samples near the cluster edges, we diminish the class-level bias and encourage low-density separation to produce better domain-invariant prototypes. The previous prototype learning method focuses more on alignment at the class level and ignores alignment at the marginal distribution. To tackle this challenge, we simply minimize the Jensen-Shannon divergence of the source and target feature vectors via adversarial learning.

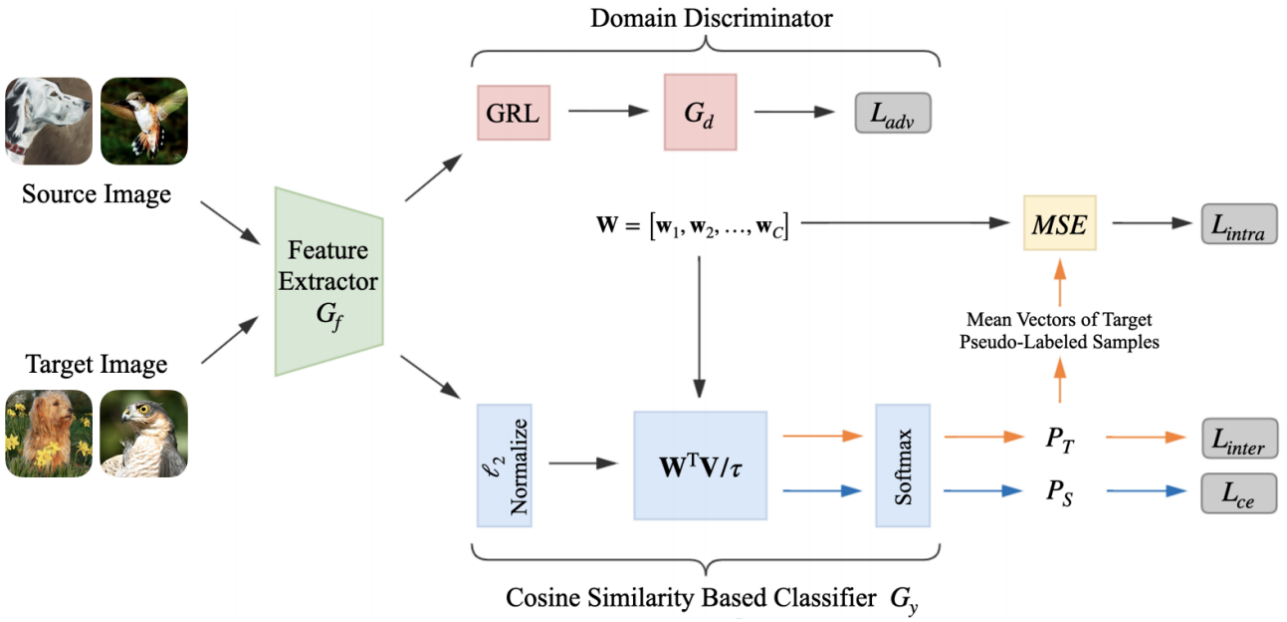


Fig. 2. The architecture of the proposed Debiased Prototype Network (DPN). DPN consists of a deep feature extractor (G_f, green), a cosine similarity based classifier (G_y, blue) and a domain discriminator (G_d, red). GRL and MSE stand for Gradient Reversal Layer and Mean Square Error, respectively. Adaptation is achieved via adversarial learning, which aligning marginal distributions to obtain an basic prototypes. Our aim is to find a more suitable class prototypes by diminishing the bias to achieve better transfer performance. The intra-class loss L_{intra} reduces the bias between the basic prototypes and expected domain-invariant prototypes, while the inter-class loss L_{inter} enforces the classifier to output low-entropy predictions on the data to obtain a larger inter-class distance. *Best viewed in color.*

III. Prototype Network for Adversarial Domain Adaptation

Overview of our method is illustrated in Fig. 2. In the unsupervised domain adaptation problem, we suppose $\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}$ and $\mathcal{D}_t = \{\mathbf{x}_j^t\}_{j=1}^{n_t}$ to be the labeled source domain and unlabeled target domain, respectively, with the same label space but different marginal distribution, i.e., $P(x^s) \neq P(x^t)$. In this paper, we utilize a cosine similarity based classifier that maps target domain samples close to their semantically identical source domain class centroids (prototypes). Then we propose a debiased method for prototype learning by which we target to diminish the intra- and inter-class bias.

A. Preliminaries: Domain Adversarial Network

In DA setting, domain adversarial networks borrow the idea of GAN [9] to help extract transferable features. The training process is a two-player game: the domain discriminator G_d learns to distinguish source and target domain features, while the feature extractor G_f learns domain-invariant feature representations to confuse the domain discriminator. After training, the network can extract both category-distinctive and domain-invariant feature representations. The objective function of domain adversarial network can be expressed as:

$$\begin{aligned} \mathcal{L}_{\mathrm{ce}}^{s}\left(\theta_f, \theta_y\right) &= \frac{1}{n_s} \sum_{\mathbf{x}_i \in \mathcal{D}_s} \mathcal{L}_y\left(G_y\left(G_f\left(\mathbf{x}_i\right)\right), \mathbf{y}_i\right) \\ \mathcal{L}_{\mathrm{adv}}\left(\theta_f, \theta_y, \theta_d\right) &= -\frac{\lambda}{n_s + n_t} \sum_{\mathbf{x}_i \in \left(\mathcal{D}_s \cup \mathcal{D}_t\right)} \mathcal{L}_d\left(G_d\left(G_f\left(\mathbf{x}_i\right)\right), d_i\right), \end{aligned} \tag{1}$$

where λ is a trade-off parameter and $\theta_f, \theta_d, \theta_y$ represent the parameters of feature network G_f, domain discriminator G_d and source classifier G_y. d_i means the domain label of sample $\mathbf{x}_i$ (e.g., d_i for source domain is 0, d_i for target domain is 1).

Note that our baseline model follows the standard adversarial domain adaptation procedure of the network. Since the training objective is to minimize the source domain classification error and maximize the domain classification error, we use labeled samples from the source domain to train a classifier G_y from minimizing the standard cross-entropy classification loss L_{ce}^s. The classifier G_y consists of a linear layer $\mathbf{W}_b^{\top} G_f(\mathbf{x}_i)$ followed by a softmax function.

B. Adversarial Learning based Prototype Network

Our base model is the same as the original baseline domain adversarial network except for the classifier design. To consider the adaptation of classifier, we employ a cosine similarity based classifier as done in [29]. Unlike [29], the prototype is not obtained by normalizing features of the same class, due to the computation inefficiency of set operation which will influence the parallelism in GPU. Compare to [33], the prototypes proposed by our method are learned by the network, and no manual boundary distribution is required to assign prototypes. As shown in Fig. 2, we still have a feature extractor G_f, a domain discriminator G_d and a classifier G_y.

For the feature extractor G_f, we employ a backbone CNN (e.g., ResNet [34]) to encode each image $\mathbf{x}$ as a feature vector $\mathbf{v}$. For the domain discriminator G_d, we employ a Gradient Reversal Layer (GRL) [35] to enable the network

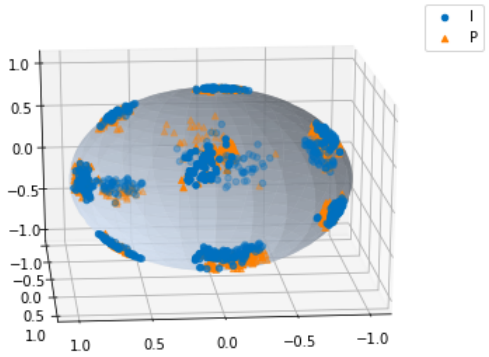


Fig. 3. A t-SNE visualization of the embeddings learned by DPN on the task I→P from ImageCLEF-DA dataset. Blue circle: source data, yellow triangle: target data. We leverage the geometric constraint induced by the hypersphere to make the domain adaptation more efficient. *Best viewed in color.*

to minimize the domain confusion loss while G_y maximizes the domain confusion loss in G_d. For the classifier G_y, we employ one linear layer without bias, which consists of weight vectors $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_C]$, where C represents the number of classes. Intuitively, the learned weight vectors $\mathbf{W}$ can be interpreted as prototypes for each class. We enforce $\|\mathbf{v}\| = \|\mathbf{w}\| = 1$ via a ℓ_2-normalization layer, which are projected to a 256-dimensional unit sphere (Fig. 3). Then the probability is denoted as:

$$P(\mathbf{x}) = \sigma\left(\mathbf{W}^{\mathrm{T}}\mathbf{V}/\tau\right), \tag{2}$$

where σ indicates a softmax function and the temperature parameter τ controls the concentration level of the distribution [36].

C. Debiased Method for Prototype Network

In DPN, we can obtain the basic prototypes by simply leverage domain adversarial learning. However, the prototypes are dominated by the source domain are biased against the expected domain-invariant prototypes we want to find. To correct the underlying domain invariant prototype, we propose a debiasing method to reduce intra-class bias and inter-class bias, as follows.

Intra-Class Bias is defined by (3):

$$B_{intra} = \mathbb{E}_{X\sim p_X}[X] - \mathbb{E}_{X_S\sim p_{X_S}}[X_S] \tag{3}$$

where $p_X = p_{X_S} \cup p_{X_T}$ is the distribution of all samples of a class and p_{X_S} is the distribution of the source domain labeled samples of a class. Obviously, the expectations of the two distributions are different. Since the basic prototype is obtained by adversarial learning, the distribution of features in the embedding space may have a low discrepancy, but the classes may not be correctly aligned in this space. The expected domain-invariant prototype should guarantee to minimize the difference between $\mathbb{E}_{X_S\sim p_{X_S}}[X_S]$ and $\mathbb{E}_{X_T\sim p_{X_T}}[X_T]$ while guaranteeing semantic consistency between the two domains. In the domain adaptation scenario, it is almost impossible to obtain the expected domain-invariant prototype since there is no label for the target domain and the conditional distribution is not known, which is to say that, the expected domain-invariant prototype is almost impossible to obtain.

To reduce the bias, we adopt the pseudo-labeling strategy to approximate $\mathbb{E}_{X\sim p_X}[X]$, which assigns temporary labels to the target samples based on the distance between the features and each prototype. In UDA, all target data belong to the category to which the source data belong. Under this assumption, the source and target samples are calculated together for the new prototype. Therefore, pseudo-labeled samples can improve the approximation to Eq. 3 such that the source and target distributions are forced to align class-wise conditionally in our unit sphere, and the target samples can be correctly classified. Specifically, we calculate the cosine similarity to basic prototypes for each mini-batch of target features. Let $\mathbf{V} \in R^{N_t\times d}$ denotes a memory bank which stores all target feature vectors, where d is the feature dimension in the last linear layer:

$$\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots \mathbf{v}_{N_t}], \tag{4}$$

where the $\mathbf{v}_i$ are ℓ_2-normalized. To consider target samples absent in the mini-batch, we employ a memory bank to store and use the feature vectors to calculate the similarity as done in [37]. Note that, since the fluctuations in sampling during learning are small, we update the memory by simply storing the feature vectors without considering the momentum of the vectors in previous epochs. Then, we can simply select top Z confidently predicted target domain samples per class to augment the source domain $\mathcal{D}_s$ with their pseudo labels. The proposed loss function for each prototype is:

$$\mathcal{L}_{intra} = \frac{1}{C}\sum_{i=1}^{C}\|\mathbf{w}_i - \frac{1}{Z}\sum_{j=1}^{Z}\bar{\mathbf{v}}_{j,i}\|_2^2. \tag{5}$$

As the loss of mean square error decreases, the debiased prototype is closer to the expected domain-invariant prototype. The experiments in Section V-B also demonstrate that a larger Z leads to better performance.

Inter-Class Bias. The underlying principle of the inter-class debiased loss is the classifier's decision boundary should not pass through high-density regions of the marginal data distribution. When the target samples are distributed near the discriminant boundary, or far from their corresponding prototypes, they are easily misclassified by the classifier. As discussed in [38], one way to enforce this is to require that the classifier output low-entropy predictions on unlabeled data. Entropy, as a measure of uncertainty, can encourage the estimated prototypes and the target features to cluster better. Therefore, we combine entropy minimization with the prototype network to reduce the distance between the feature vectors $\boldsymbol{v}$ in the target domain and the prototype $\mathbf{w}$ on our unit sphere, allowing the network to output more definite semantic label prediction. This debiased process enforces feature vectors

to better cluster around prototypes and away from dissimilar prototypes.

$$\mathcal{L}_{inter}\left(\theta_f, \theta_y\right) = -\frac{1}{n_t} \sum_{\mathbf{x}_i \in \mathcal{D}_t} p_t \log\left(p_t\right), \tag{6}$$

where n_t is the number of target domain data and p_t the probability of the target sample.

Overall objectives. The overall training objective for training DPN can be written as:

$$\mathcal{L} = \mathcal{L}_{\text{ce}}^{s} + \mathcal{L}_{adv} + \lambda_{intra}\mathcal{L}_{intra} + \lambda_{inter}\mathcal{L}_{inter} \tag{7}$$

where λ_{intra}, λ_{inter} are hyper-parameters. The optimization problem is to find the parameters $\hat{\theta}_f, \hat{\theta}_y$ and $\hat{\theta}_d$ that jointly satisfy:

$$\begin{aligned} \left(\hat{\theta}_f, \hat{\theta}_y\right) &= \arg\min_{\theta_f, \theta_y} \mathcal{L}\left(\theta_f, \theta_y, \theta_d\right) \\ \left(\hat{\theta}_d\right) &= \arg\max_{\theta_d} \mathcal{L}\left(\theta_f, \theta_y, \theta_d\right) \end{aligned} \tag{8}$$

Our method can be formulated as the iterative minimax training. To simplify training process, we use a Gradient Reversal Layer (GRL) [13] to ensure that the gradient direction is automatically inverted in the back-propagation process and the identity transformation is achieved in the forward propagation process, which is illustrated in Fig. 2.

IV. Experiments

In this section, we evaluate the proposed model with state-of-the-art domain adaptation methods on unsupervised domain adaptation problems. DPN is validated on two popular datasets: ImageCLEF-DA [39] and Office-Home [40].

A. Datasets

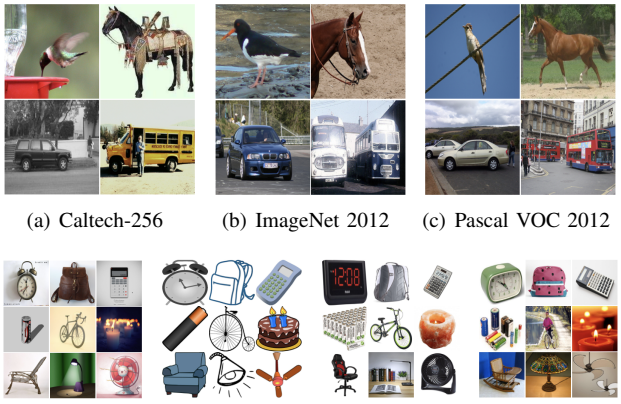
Examples of the two datasets are shown in Fig. 4.

ImageCLEF-DA is a benchmark for ImageCLEF 2014 domain adaptation challenge. It is collected by selecting the 12 common object categories in three different domains: Caltech-256 (C), ImageNet ILSVRC 2012 (I), Pascal VOC 2012 (P). There are 50 images in each category and 600 images in each domain.

Office-Home is more challenging dataset to evaluate domain adaptation algorithms. It contains around 15,500 images from 65 different categories of everyday objects. There are four significantly different domains: Artistic images (Ar), Clip Art (Cl), Product images (Pr) and Real-World images (Rw). The images for these domains were collected using a python web-crawler, so there are significant differences in appearance and background, and the number of images in each category is different, making it more difficult to transfer across domains.

B. Implementation details

We implement our all method base on the PyTorch, and the feature extractor G_f is fine-tune from ResNet-50 [34] models pre-trained on the ImageNet dataset [41], excluding the last FC layer. For all the unsupervised domain adaptation tasks as in [17], we train the new layers and classifier layer through back-propagation, where the classifier is trained from scratch with learning rate 10 times that of the lower layers. For baselines, we use their implementation. When optimizing G_f, G_y and G_d, all network parameters are updated by Stochastic Gradient Descent (SGD) with momentum of 0.9. We also following the results of [42], the value of temperature τ is set 0.05 in all settings. When modify the prototypes, we increasing λ_{intra} from 0 to 100 by multiplying to $100 \cdot \frac{1-\exp(-\delta p)}{1+\exp(-\delta p)}, \delta = 10$, and fix $\lambda_{inter} = 0.05, z = 8$ for our method. Other hyper-parameters are tuned via transfer cross validation [43].



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

(d) Art (e) Clipart (f) Product (g) Real World

Fig. 4. Datasets. Top: ImageCLEF-DA; Bottom: Office-Home.

C. Baselines

We evaluate our proposed Debiased Prototype Network (DPN) with the following state-of-the-art method for comparison. For ImageCLEF-DA datasets, our compared baseline methods include DDC [4], DAN [5], RTN [44], DANN [13], JAN [39], MADA [45], iCAN [46], CDAN [17], SAFN [47] and SRADA [14]. Besides, on Office-HOME dataset, we compare with MEDA [48], DWT-MEC [49], TADA [18], DSR [11] and SymNet [50].

D. Results

Table I shows the classification accuracy of the ResNet-50-based unsupervised domain adaptation on the Office-Home datasets. On most transfer tasks, DPN significantly outperforms the comparison methods. Our proposed method achieves state-of-the-art classification accuracy (70.7%), which is 1.8% higher than the second place SRADA. Notably, DPN is effective in improving class classification accuracy on hard transfer tasks, such as Ar→Cl and Pr→Cl. Table II shows the results of DPN on the CLEF-DA datasets. DPN also outperforms all comparison methods on most tasks. For a fair comparison, all baseline results were obtained from their original papers, or quoted from [14].

Comparison with adversarial based methods. Compared with DANN, which can be regarded as the baseline of adversarial learning methods, our DPN outperforms it by 13.1% on Office-Home. In contrast to other latest adversarial methods, especially CDAN, our DPN outperforms it by 4.9%, 1.7%

TABLE I
RECOGNITION ACCURACY(%) ON OFFICE-HOME DATASET.

Method	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pw	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw→Pr	AVG
ResNet-50	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	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	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
JAN	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
MEDA	46.6	68.9	68.8	49.0	66.4	66.1	51.8	45.0	72.9	61.2	50.3	76.0	60.2
CDAN	50.7	70.6	76.0	57.6	70.0	70.0	57.4	50.9	77.3	70.9	56.7	81.6	65.8
DWT-MEC	54.7	72.3	77.2	56.9	68.5	69.8	54.8	47.9	78.1	68.6	54.9	81.2	65.4
SAFN	52.0	71.7	76.3	**64.2**	69.9	71.9	63.7	51.4	77.1	70.9	57.1	81.5	67.3
TADA	53.1	72.3	77.2	59.1	71.2	72.1	59.7	53.1	78.4	72.4	60.0	82.9	67.6
SymNet	47.7	72.9	78.5	**64.2**	71.3	**74.2**	**64.2**	48.8	79.5	**74.5**	52.6	82.7	67.6
DSR	53.4	71.6	77.4	57.1	66.8	69.3	56.7	49.2	75.7	68.0	54.0	79.5	64.9
SRADA	55.5	73.5	78.7	60.7	74.1	73.1	59.5	55.0	80.4	72.4	60.3	84.3	68.9
DPN	**58.4**	**76.1**	**79.0**	64.0	**74.4**	72.6	63.5	**57.4**	**80.7**	**74.5**	**63.9**	**84.4**	**70.7**

TABLE II
RECOGNITION ACCURACY(%) ON IMAGECLEF-DA DATASET.

Method	I→P	P→I	I→C	C→I	C→P	P→C	AVG
ResNet-50	74.8	83.9	91.5	78.0	65.5	91.2	80.7
DDC	74.6	85.7	91.1	82.3	68.3	88.8	81.8
DAN	75.0	86.2	93.3	84.1	69.8	91.3	83.3
RTN	75.6	86.8	95.3	86.9	72.7	92.2	84.9
DANN	75.0	86.0	96.2	87.0	74.3	91.5	85.0
JAN	76.8	88.0	94.7	89.5	74.2	91.7	85.8
MADA	75.0	87.9	96.0	88.8	75.2	92.2	85.8
iCAN	**79.5**	89.7	94.7	89.9	78.5	92.0	87.4
CDAN	77.7	90.7	97.7	91.3	74.2	94.3	87.7
SAFN	78.0	91.7	96.2	91.1	77.0	94.7	88.1
SRADA	78.3	91.3	96.7	90.5	78.1	**96.2**	88.5
DPN	78.8	**92.8**	**96.8**	**92.8**	**78.8**	**96.2**	**89.4**

on Office-Home and ImageCLEF-DA respectively. Methodologically, as discussed in Section III-B, our approach utilizes a prototype network for UDA, which is different from the domain-adversarial learning based UDA approach described above.

V. DISCUSSION

A. Ablation Study

Our model benefits from a prototype network and a novel debiased method. To separate the transferable contributions of the different components of our model, the ablation analysis results under different model variants with some loss removed are presented in Table III. From this, we can compare these baselines in two aspects. On the one hand, we trained only source classifiers based on cross-entropy loss in the baseline of ResNet-50. Compared with the prototype network ResNet-50 (PN), based on the cosine similarity classifier, ResNet-50 (PN) improves the performance from 38.7% to 49.5%. Our results show that the prototype network has better transfer performance. On the other, we use the prototype network as the base model for DANN, and DANN(PN) can be considered as the baseline, and the performance is increased from 48.7% to 56.5%.

Experimental results reveal that both DPN (intra) and DPN (inter) gain significant improvements over baselines but DPN (inter) works better than DPN (intra). The reason is that DPN (inter) can better encourage a low-density separation between classes, thus allowing the knowledge to be better transferred in the case of complex datasets. More importantly, DPN (intra+inter) also shows a large improvement over DPN (intra) or DPN (inter), revealing that DPN (intra+inter) is the most efficient, while DPN (intra) and DPN (inter) are well complementary to each other.

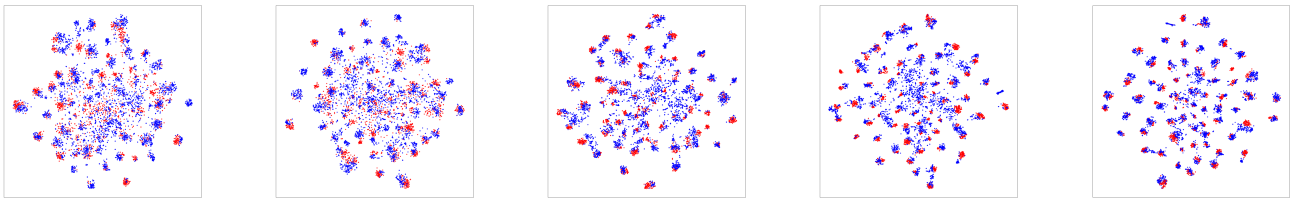
TABLE III
ABLATION STUDY ON THE OFFICE-HOME DATASET.

Method	Ar→Cl	Ar→Pr	Pr→Ar	Pr→Cl	AVG
ResNet-50	34.9	50.0	38.5	31.2	38.7
ResNet-50 (PN)	42.7	65.5	50.0	39.7	49.5
DANN	45.6	59.3	46.1	43.7	48.7
DANN (PN)	54.0	68.3	54.2	49.3	56.5
DPN (intra)	54.3	68.9	54.5	49.2	56.8
DPN (inter)	57.4	74.8	62.6	56.5	62.8
DPN (intra+inter)	**58.4**	**76.1**	**63.4**	**57.4**	**63.8**

B. Parameter Sensitivity

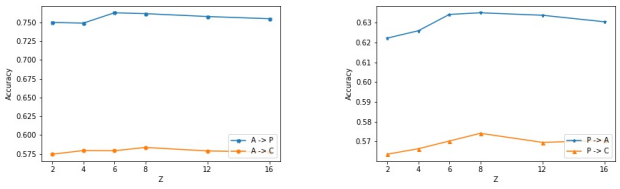
To illustrate the effectiveness of our proposed intra-class bias diminishing method, we empirically investigate the sensitivity of the parameters through extensive parametric experiments. Office-Home is a more challenging dataset compared with others and there exists an obvious gap between the accuracy of Office-Home and the accuracy of other datasets. To better demonstrate the variation of accuracy for different Z pseudo-tagged samples: (1) we choose the transfer tasks Ar→Pr and Ar→Cl to verify the robustness of our method in the case of less source domain data and large sample variation between two domains, as show in Fig. 6(a); (2) The transfer task Pr→Ar is chosen to verify the effect of the hyperparameter Z on the model in the case of having rich source domain data, as show in Fig. 6(b); (3) In addition, we choose the transfer task Pr→Cl as a comparison validation due to the relatively close number of samples in the Pr and Cl domains, as show in Fig. 6(b).

It can be shown that a coincident tendency that with larger Z, there is a growth of classification accuracy in each task.



(a) ResNet: Ar→Pr (b) ResNet (PN): Ar→Pr (c) DANN: Ar→Pr (d) DANN (PN): Ar→Pr (e) DPN: Ar→Pr

Fig. 5. The t-SNE visualization of features learned by (a) ResNet, (b) ResNet (PN), (c) DANN, (d) DANN (PN) and (e) DPN (red: Ar; blue: Pr).



(a) Ar→Pr and Ar→Cl task (b) Pr→Ar and Pr→Cl task

Fig. 6. Parameter sensitivity. Analysis of model parameter Z on Office-Home dataset.

However, the improvement space is limited when the source domain samples are poor, and better transfer performance can be obtained when the number of samples between domains is the same. Moreover, since we select Z pseudo-labeled samples on each class, this implicitly assumes that the class distribution is close to uniform. Nevertheless, our debiased method still improves over the baselines even when the classes are not well balanced, indicating that the method is robust to violations of the class balance assumption.

C. Feature Visualization

To further evaluate the performance of DPN, we visualize the network activations on task Ar→Pr (65 classes, from Office-Home dataset) learned by ResNet, ResNet (PN), DANN, DANN (PN) and DPN using t-SNE embeddings [51] in Fig. 5. The red dots are the source samples, and the blue dots are the target samples. From the left (ResNet) to the right (DPN), the source and target domain features become increasingly difficult to distinguish. From Fig. 5(a)-5(d), we observe that the prototype network based model can reserve better discrimination than the other two baselines. The reason is that in the spherical feature space, the source and target domain features can be better aligned by the cosine similarity based classifier. Although prototype learning can learn more transferable features, the class-level discrepancy of features are not improved because the limited distribution alignment capability of DANN can only align the marginal distribution between the domains.

In our method, the category information is introduced into the migration process to align the conditional distribution. Comparing with DANN(PN), our method can learn a more robust feature representation. From the classification accuracies in Table III, DPN (76.1%) has a significant improvement over DANN (PN) (68.3%). In particular, the representations generated by DPN formed exactly 65 clusters with clear boundaries. The above observations show that: (1) the features obtained from the prototype learning have stronger generalization ability and better transferability than traditional methods. And (2) our model achieves better intra-class compactness and inter-class separability by reducing class-level bias, resulting in a larger performance improvement.

VI. Conclusion And Future Work

This paper presented the Debiased Prototype Network for Adversarial Domain Adaptation, which we called DPN, a novel debiased method with reduced both intra- and inter-class bias. To obtain better domain-invariant knowledge, we extract domain-invariant semantic information over the unit sphere via the prototype network. With the help of geometric constraints from the hypersphere, the features can be more tightly clustered with the estimated prototype. Adaptation is achieved by adversarial learning to align the domain distribution, which enhances the transferability of the learned features and obtains the basic prototype. In addition, we rectify the prototypes by reducing intra- and inter- class bias. Experimental results demonstrate that DPN achieves superior performance compared to state-of-the-art methods.

DPN is a simple and efficient method for domain adaptation. With the widespread use of sensors in mobile phones, cars, buildings, roads, and computers, people are collecting larger and more diverse information. In the future, we plan to extend DPN to the problem of domain adaptation with inconsistent label space.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant 61871186 and 61771322, and in part by the Shanghai Natural Science Foundation under Grant 18ZR1411400.

References

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

[2] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *CVPR*. IEEE Computer Society, 2012, pp. 2066–2073.

[3] K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang, "Domain adaptation under target and conditional shift," in *ICML (3)*, ser. JMLR Workshop and Conference Proceedings, vol. 28. JMLR.org, 2013, pp. 819–827.

[4] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *CoRR*, vol. abs/1412.3474, 2014.

[5] M. Long, Y. Cao, J. Wang, and M. I. Jordan, "Learning transferable features with deep adaptation networks," in *ICML*, ser. JMLR Workshop and Conference Proceedings, vol. 37. JMLR.org, 2015, pp. 97–105.

[6] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *AAAI*. AAAI Press, 2016, pp. 2058–2065.

[7] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz, "Central moment discrepancy (CMD) for domain-invariant representation learning," in *ICLR (Poster)*. OpenReview.net, 2017.

[8] S. Yu, A. Shaker, F. Alesiani, and J. C. Príncipe, "Measuring the discrepancy between conditional distributions: Methods, properties and applications," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 2020, pp. 2777–2784.

[9] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014, pp. 2672–2680.

[10] X. Gu, J. Sun, and Z. Xu, "Spherical space domain adaptation with robust pseudo-label loss," in *CVPR*. IEEE, 2020, pp. 9098–9107.

[11] R. Cai, Z. Li, P. Wei, J. Qiao, K. Zhang, and Z. Hao, "Learning disentangled semantic representation for domain adaptation," in *IJCAI*. ijcai.org, 2019, pp. 2060–2066.

[12] 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 *CVPR*. Computer Vision Foundation / IEEE, 2019, pp. 627–636.

[13] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. S. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, pp. 59:1–59:35, 2016.

[14] S. Wang and L. Zhang, "Self-adaptive re-weighted adversarial domain adaptation," in *IJCAI*. ijcai.org, 2020, pp. 3181–3187.

[15] J. Hoffman, E. Tzeng, T. Park, J. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 2018, pp. 1994–2003.

[16] M. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *NIPS*, 2017, pp. 700–708.

[17] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *NeurIPS*, 2018, pp. 1647–1657.

[18] X. Wang, L. Li, W. Ye, M. Long, and J. Wang, "Transferable attention for domain adaptation," in *AAAI*. AAAI Press, 2019, pp. 5345–5352.

[19] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *ICML*. Omnipress, 2011, pp. 513–520.

[20] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *ICML*, ser. JMLR Workshop and Conference Proceedings, vol. 32. JMLR.org, 2014, pp. 647–655.

[21] H. Yang, X. Zhang, F. Yin, and C. Liu, "Robust classification with convolutional prototype learning," in *CVPR*. IEEE Computer Society, 2018, pp. 3474–3482.

[22] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, "Panet: Few-shot image semantic segmentation with prototype alignment," in *ICCV*. IEEE, 2019, pp. 9196–9205.

[23] Y. Shu, Y. Shi, Y. Wang, T. Huang, and Y. Tian, "P-ODN: prototype based open deep network for open set recognition," *CoRR*, vol. abs/1905.01851, 2019.

[24] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," in *NIPS*, 2017, pp. 4077–4087.

[25] T. Gao, X. Han, Z. Liu, and M. Sun, "Hybrid attention-based prototypical networks for noisy few-shot relation classification," in *AAAI*. AAAI Press, 2019, pp. 6407–6414.

[26] B. N. Oreshkin, P. R. López, and A. Lacoste, "TADAM: task dependent adaptive metric for improved few-shot learning," in *NeurIPS*, 2018, pp. 719–729.

[27] W. Xu, Y. Xian, J. Wang, B. Schiele, and Z. Akata, "Attribute prototype network for zero-shot learning," in *NeurIPS*, 2020.

[28] X. Li, M. Fang, H. Li, and J. Wu, "Zero shot learning based on class visual prototypes and semantic consistency," *Pattern Recognit. Lett.*, vol. 135, pp. 368–374, 2020.

[29] J. Liu, L. Song, and Y. Qin, "Prototype rectification for few-shot learning," in *ECCV (1)*, ser. Lecture Notes in Computer Science, vol. 12346. Springer, 2020, pp. 741–756.

[30] K. Saito, D. Kim, S. Sclaroff, and K. Saenko, "Universal domain adaptation through self supervision," in *NeurIPS*, 2020.

[31] R. Xu, P. Liu, L. Wang, C. Chen, and J. Wang, "Reliable weighted optimal transport for unsupervised domain adaptation," in *CVPR*. IEEE, 2020, pp. 4393–4402.

[32] C. Luo, Z. Li, K. Huang, J. Feng, and M. Wang, "Zero-shot learning via attribute regression and class prototype rectification," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 637–648, 2018.

[33] P. Mettes, E. van der Pol, and C. Snoek, "Hyperspherical prototype networks," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019, pp. 1485–1495.

[34] A. Verma, H. Qassim, and D. Feinzimer, "Residual squeeze CNDS deep learning CNN model for very large scale places image recognition," in *UEMCON*. IEEE, 2017, pp. 463–469.

[35] Y. Ganin and V. S. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *ICML*, ser. JMLR Workshop and Conference Proceedings, vol. 37. JMLR.org, 2015, pp. 1180–1189.

[36] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *CoRR*, vol. abs/1503.02531, 2015.

[37] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *CVPR*. IEEE Computer Society, 2018, pp. 3733–3742.

[38] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *NIPS*, 2004, pp. 529–536.

[39] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *ICML*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 2208–2217.

[40] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *CVPR*. IEEE Computer Society, 2017, pp. 5385–5394.

[41] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[42] R. Ranjan, C. D. Castillo, and R. Chellappa, "L2-constrained softmax loss for discriminative face verification," *CoRR*, vol. abs/1703.09507, 2017.

[43] M. Sugiyama, M. Krauledat, and K. Müller, "Covariate shift adaptation by importance weighted cross validation," *J. Mach. Learn. Res.*, vol. 8, pp. 985–1005, 2007.

[44] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *NIPS*, 2016, pp. 136–144.

[45] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in *AAAI*. AAAI Press, 2018, pp. 3934–3941.

[46] W. Zhang, W. Ouyang, W. Li, and D. Xu, "Collaborative and adversarial network for unsupervised domain adaptation," in *CVPR*. IEEE Computer Society, 2018, pp. 3801–3809.

[47] R. Xu, G. Li, J. Yang, and L. Lin, "Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation," in *ICCV*. IEEE, 2019, pp. 1426–1435.

[48] J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, and P. S. Yu, "Visual domain adaptation with manifold embedded distribution alignment," in *ACM Multimedia*. ACM, 2018, pp. 402–410.

[49] S. Roy, A. Siarohin, E. Sangineto, S. R. Bulò, N. Sebe, and E. Ricci, "Unsupervised domain adaptation using feature-whitening and consensus loss," in *CVPR*. Computer Vision Foundation / IEEE, 2019, pp. 9471–9480.

[50] Y. Zhang, H. Tang, K. Jia, and M. Tan, "Domain-symmetric networks for adversarial domain adaptation," in *CVPR*. Computer Vision Foundation / IEEE, 2019, pp. 5031–5040.

[51] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *ICML*, ser. JMLR Workshop and Conference Proceedings, vol. 32. JMLR.org, 2014, pp. 647–655.