Regularized Deep Transfer Learning: When CNN Meets kNN

Shanshan Wang, *Student Member, IEEE,* Lei Zhang, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a regularized deep transfer learning architecture composed of a softmax classifier and a *k*-nearest neighbor (*k*NN) classifier. We aim to generalize the softmax classifier possibly well under the regularization effect of the *k*NN classifier. That is, given a training sample, the *k*NN classifier assigns a soft label vector according to its distance to the center of each class. The working mechanism of the *k*NN classifier is attributed to both the source and target domains. On the one hand, it gradually becomes stronger by backpropagating the cross-entropy classification loss on the source images. On the other hand, for target data, we enforce to minimize the discrepancy between the label vectors produced by the *k*NN classifier and the softmax classifier. Using the *k*NN classifier, we are able to reduce the intra-class variations on the source domain and meanwhile pull close the source and target feature distributions, which can better bound the expected error of target domain. In experiment, we demonstrate that our method compares favorably with the state-of-the-arts on benchmarks.**

***Index Terms*—*k*NN, CNN, classification, domain adaptation.**

I. Introduction

UNSUPERVISED domain adaptation (UDA) aims to recognize the unlabeled target domain data by leveraging only the labeled, related but different source domain instances. In literature, a large body of works enforce the feature distribution of the target and source domain to be globally aligned [10], [21], [7]. Maximum Mean Discrepancy (MMD), as a non-parametric metric, is commonly used to measure the dissimilarity of distributions [8], [10]. After MMD guided distribution alignment, domain-invariant or domain-confused feature representation can be learned [3], [17], [22], [15], [20]. Nevertheless, it is not *sufficient* to guarantee a class-level alignment for classification.

In this paper, we adopt the broad idea of class-level alignment and propose to leverage the regularization effect of a *k*NN classifier on the CNN softmax classifier. Basically, given a training sample, *k*NN classifier assigns a soft label vector according to its distance to the center of its class. The key to the successful collaboration of two classifiers in domain adaptation depends on three essential bases. First, the CNN based source classifier can recognize source samples correctly [14]. Second, in order to be an effective auxiliary force, the discriminative ability of the *k*NN classifier should be sufficiently strong. Without a strong *k*NN classifier, learning domain invariant features with intra-class compactness and inter-class separability cannot be guaranteed. Third, given a relatively stronger *k*NN classifier, it is critical to find an effective way to regularize the CNN-based source classifier. The relatively reliable labels produced by the *k*NN classifier can be used as complementary cues. This effect is particularly important for the target domain training samples for which the *k*NN classifier predicts soft label vectors. These soft vectors may regularize softmax classifier for domain adaptation.

In light of the above analysis, this paper proposes a new deep architecture that depends on the regularization effect of a *k*NN classifier on the softmax one (the classifier-of-interest). The pipeline of our method is illustrated in Fig. 1. In our method, the working mechanism of the *k*NN classifier is attributed to both the source and target domains. On the one hand, its discriminative ability gradually becomes stronger by backpropagating a cross-entropy loss on the source images, which we name as self-regularization term. With the joint supervision of the self-regularization term and the softmax loss in the CNN-based classifier, the intra-class feature variation is reduced, and the inter-class feature variation is enlarged. In this manner, the *k*NN classifier can be viewed as a good priori condition for class-level alignment across domains. On the other hand, with a stronger *k*NN classifier, we aim to learn a *k*NN regularized CNN classifier by minimizing the *k*NN-CNN confidence gap and the soft label guided distribution discrepancy simultaneously. The confidence gap reflects the key idea that *k*NN regularizes CNN and we adopt the ℓ_1 loss to measure the gap. Specifically, we note that the confidence of *k*NN classifier can also be promoted by global distribution alignment including marginal and conditional distribution. In this sense, the predicted soft target labels are able to guide the formulation of conditional distribution, and then progressively improve the confidence of soft labels. We summarize the main contributions as follows.

- An idea that *kNN regularizes CNN* is proposed to alleviate domain bias of CNN classifier in unsupervised domain adaptation. By taking the class prior into consideration, a ℓ_1 loss based on the results of experiment for CNN classifier regularization is proposed.
- Using the soft target labels produced by the *k*NN classifier, a more confident conditional distribution of target data is formulated and can be incorporated into CNN for feature distribution alignment across domains. Domain discrepancy in both feature and category space is reduced.
- Experiment on various datasets verifies the superiority of the ℓ_1 loss between *k*NN and CNN to other CNN-based domain adaptation methods.

This work is supported by National Natural Science Fund of China (No. 61771079), Natural Science Fund of Chongqing (No. cstc2018jcyjAX0250) and Chongqing Youth Talent Program. (*Corresponding author: Lei Zhang*)

S. Wang and L. Zhang are with the School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China. (E-mail: wangshanshan@cqu.edu.cn, leizhang@cqu.edu.cn).

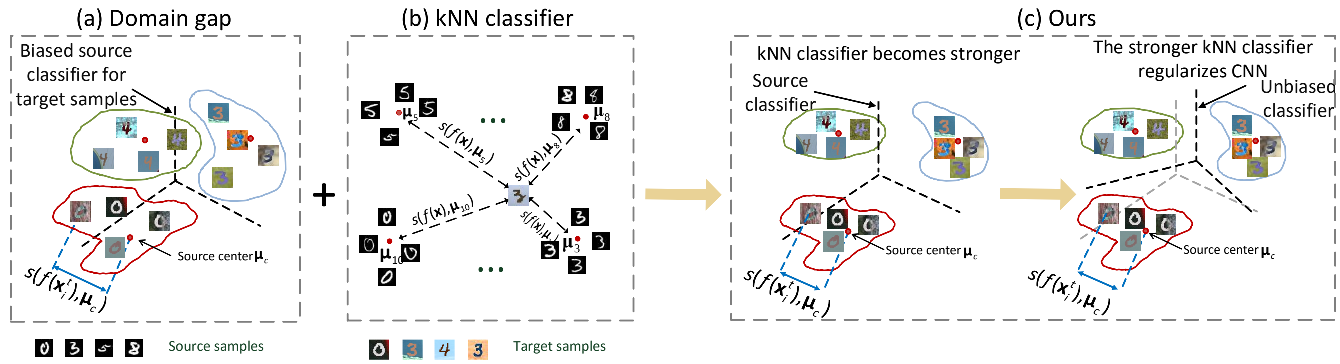

Fig. 1: Illustration of our approach. (a): When only considering DA in feature alignment, target data may be mispredicted by the biased source classifier due to domain gap. (b): A kNN classifier assigns labels according to the k-nearest neighbors of a sample-of-interest. The classifier is combined with source classifier to reduce domain gap. (c): First, the discriminative ability of kNN classifier should be sufficiently strong. Then, with a stronger kNN classifier, we aim to learn a kNN regularized CNN by simultaneously minimizing the kNN-CNN confidence gap and the soft label guided distribution discrepancy. Domain discrepancy in both the feature and category space is reduced.

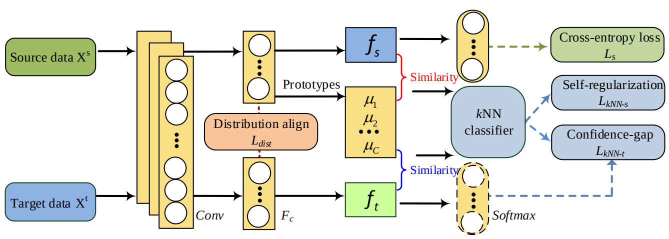

Fig. 2: The framework of our kNN regularized CNN method. Three parts are referred. 1) The cross-entropy loss $\mathcal{L}_s$ is optimized for source classifier training. 2) To minimize the kNN-CNN confidence gap, ℓ_1 loss ($\mathcal{L}_{kNN-t}$) is used to regularize the source classifier. 3) The distribution alignment loss ($\mathcal{L}_{dist}$) is used for distribution match.

II. The Proposed Method

A. Notation

We suppose $\mathcal{D}_s = \{(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s)\}_{i=1}^{n_s}$ and $\mathcal{D}_t = \{\boldsymbol{x}_j^t\}_{j=1}^{n_t}$ to be the labeled source training set and unlabeled target data, drawn from distributions $P(\boldsymbol{x}, \boldsymbol{y})$ and $Q(\boldsymbol{x}, \boldsymbol{y})$, respectively. Clearly, $P \neq Q$. Our goal is to predict the target label $\hat{\boldsymbol{y}}_t = \arg\max_{y_t} G(f(\boldsymbol{x}_t))$, where $G(\cdot)$ represents the softmax output and $f(\cdot)$ refers to the feature representation. Our method aims to reduce the domain gap by minimizing the source risk $\epsilon_s(G) = \mathbb{E}_{(\boldsymbol{x}^s, \boldsymbol{y}^s)\sim P}[G(f(\boldsymbol{x}^s)) \neq \boldsymbol{y}^s]$ regularized by kNN classifier and feature alignment across domains, such that the target risk $\epsilon_t(G) = \mathbb{E}_{(\boldsymbol{x}^t, \boldsymbol{y}^t)\sim Q}[G(f(\boldsymbol{x}^t)) \neq \boldsymbol{y}^t]$ can be minimized.

B. Model Formulation

An overview of our proposed method is depicted in Fig. 2. The goal of *kNN regularizes CNN* is to produce a softmax classifier $G(\cdot)$ that operates on the feature representation $f(\boldsymbol{x})$ to recognize target samples. The proposed model is composed of three parts: source classifier training, kNN regularizes CNN, and feature distribution alignment. For each part, the loss function is formulated as follows.

Source Classifier Training. We address unsupervised domain adaptation with sufficient source labels but none target label available. We suppose that there are C categories for both domains. For training the source classifier in CNN, the standard cross-entropy loss with softmax function is used,

$$\mathcal{L}_s(\boldsymbol{x}^s, \boldsymbol{y}^s; f; G) = -\sum_c \mathbf{1}[\boldsymbol{y}^s = c] \log p_s^c, \tag{1}$$

where $p_s = G(f(\boldsymbol{x}^s))$ is the softmax probability. This loss can guarantee the discrimination of source features.

***k*NN regularizes CNN.** It is known to us that by optimizing the loss function Eq. (1), the CNN classifier is seriously biased toward the source domain. The prediction confidence by testing the CNN on target data is undoubtedly low. To alleviate the model bias problem, in this section, we describe how kNN regularizes CNN. The basic idea is to reduce the confidence gap between kNN and CNN. In this paper, we suppose that kNN can basically reflect the category similarity between target data and the class-wise prototypes (*e.g.*, class centers) of source data.

Suppose there are C class-wise prototypes $\boldsymbol{\mu} = [\mu_1, \cdots, \mu_C]$, where the mean representation μ_c of the source samples with respect to class c $(c = 1, \cdots, C)$ is computed as,

$$\mu_c = \frac{1}{n_c^s} \sum_{\boldsymbol{x}_i^s \in \mathbf{X}_c^s} f(\boldsymbol{x}_i^s), \tag{2}$$

where $\mathbf{X}_c^s$ is the source training set with respect to class c.

In kNN classifier, the cosine similarity $s(\mathbf{p}, \mathbf{q})$ between $\mathbf{p}$ and $\mathbf{q}$ is computed as,

$$s(\mathbf{p}, \mathbf{q}) = \frac{<\mathbf{p}, \mathbf{q}>}{\|\mathbf{p}\|_2 \|\mathbf{q}\|_2}. \tag{3}$$

To reduce the confidence gap between kNN and CNN on the unlabeled target data, we leverage a ℓ_1 loss, i.e., $\mathcal{L}_{kNN-t}$ between the kNN soft predictions and CNN soft prediction of the target samples. The rational behind is that the two samples with high similarity score in feature level should have similar predictions. The loss function is formulated as,

$$\mathcal{L}_{kNN-t} = \frac{1}{C} \sum_{c=1}^{C} \|{p^t}_{knn,c} - {p^t}_{cnn,c}\|_1, \tag{4}$$

where $p_{cnn}^t = G(f(\boldsymbol{x}^t))$ is the output probability of CNN classifier and $p_{knn}^t = softmax(s(f(\boldsymbol{x}^t), \boldsymbol{\mu}))$ is the softmax activated output probability of kNN classifier. We adopt sparse ℓ_1 loss which assumes the errors obey Laplacian distribution, and shows better performance than Gaussian distribution of ℓ_2 loss. As can be seen from Eq. (4), kNN aims to regularize CNN by using the target data, instead of the source data. Although the CNN can be trained on the supervised source data with cross-entropy loss in Eq. (1), the self-regularization based kNN classifier can also be imposed on source data for similarity learning and classifier learning simultaneously. Specifically, under the traditional cross-entropy loss, kNN based similarity classifier is jointly trained. Therefore, the kNN regularized source classifier loss, *ie*, $\mathcal{L}_{kNN-s}$ is formulated as,

$$\mathcal{L}_{kNN-s}(\boldsymbol{x}^s, \boldsymbol{y}^s; f) = -\sum_c \mathbf{1}[\boldsymbol{y}^s = c] \log p^s_{knn,c}, \tag{5}$$

where $p^s_{knn} = softmax(s(f(\boldsymbol{x}^s), \boldsymbol{\mu}))$ is the softmax activated output probability of kNN classifier.

Up to now, the details on how kNN regularizes CNN in this paper are presented. The loss $\mathcal{L}_{kNN}$ is written as,

$$\mathcal{L}_{kNN} = \mathcal{L}_{kNN-t} + \mathcal{L}_{kNN-s}, \tag{6}$$

Feature Distribution Alignment. By far, we have presented how kNN classifier regularizes CNN for joint similarity and classifier learning without unbias across domains. However, there are two remained issues to be addressed. On one hand, the marginal distribution should be formulated and matched. On the other hand, the conditional distribution with category supervision should also be aligned. But the target data is unlabeled, therefore, the pseudo target labels with progressive updated strategy are computed in model optimization. We employ the maximum mean discrepancy (MMD) [5] to measure the marginal and conditional distribution.

First, the marginal distribution discrepancy between source and target data is formulated as,

$$MMD_{marg} = \| \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(\boldsymbol{x}^s_i) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(\boldsymbol{x}^t_j) \|^2_{\mathcal{H}_k}, \tag{7}$$

where $\phi(\cdot)$ denotes the feature map associated with the kernel map $k(\boldsymbol{x}^s, \boldsymbol{x}^t) = \langle \phi(\boldsymbol{x}^s), \phi(\boldsymbol{x}^t) \rangle$ from original space to a reproducing kernel Hilbert space $\mathcal{H}_k$ (RKHS).

Second, the conditional distribution discrepancy between source and target data is formulated as,

$$MMD_{cond} = \sum_{c=1}^{C} \| \frac{1}{n^c_s} \sum_{i=1}^{n^c_s} \phi(\boldsymbol{x}^s_i | \boldsymbol{y}^c_s) - \frac{1}{n^c_t} \sum_{j=1}^{n^c_t} \phi(\boldsymbol{x}^t_j | \hat{\boldsymbol{y}}^c_t) \|^2_{\mathcal{H}_k}, \tag{8}$$

where $\hat{\boldsymbol{y}}_t$ is the pseudo target labels. By jointly minimizing the marginal and conditional distribution discrepancy, the feature distribution alignment loss $\mathcal{L}_{dist}$ is given by,

$$\mathcal{L}_{dist} = MMD_{marg} + MMD_{cond}, \tag{9}$$

Overall Training Loss. Considering the Eq. (1), Eq. (6) and Eq. (9), the overall training loss of our model is given by,

$$\mathcal{L} = \mathcal{L}_s + \lambda_1 \mathcal{L}_{dist} + \lambda_2 \mathcal{L}_{kNN}, \tag{10}$$

where λ_1 and λ_2 are hyper-parameters. The optimization procedure is following the basic CNN protocol, since the gradients of the three terms in Eq. (10) are computable. In our model, no extra CNN variables are introduced. The CNN network parameters can be solved with standard mini-batch SGD. The pseudo-label $\hat{y}^i_t$ of x^i_t based on maximum posterior probability using CNN softmax classifier is progressively updated.

Office-31	A→W	D→W	W→D	A→D	D→A	W→A	Avg.
Source Only	68.4	96.7	99.3	68.9	62.5	60.7	76.1
TCA	72.7	96.7	99.6	74.1	61.7	60.9	77.6
GFK	72.8	95.0	98.2	74.5	63.4	61.0	77.5
DDC	75.6	96.0	98.2	76.5	62.2	61.5	78.3
DAN	80.5	97.1	99.6	78.6	63.6	62.8	80.4
RTN	84.5	96.8	99.4	77.5	66.2	64.8	81.6
DANN	82.0	96.9	99.1	79.7	68.2	67.4	82.2
ADDA	86.2	96.2	98.4	77.8	69.5	68.9	82.9
JAN	85.4	97.4	99.8	84.7	68.6	70.0	84.3
CAN	81.5	98.2	99.7	85.5	65.9	63.4	82.4
EM	86.8	**99.3**	**100.0**	87.2	71.2	**71.8**	86.1
SimNet	88.6	98.2	99.7	85.3	**73.4**	**71.8**	86.2
Ours	**92.2**	97.2	99.8	**88.0**	71.1	71.7	**86.7**

TABLE I: Recognition accuracy (%) on the Office-31 dataset. All models utilize ResNet-50 as base architecture.

ImageCLEF-DA	I→P	P→I	I→C	C→I	C→P	P→C	Avg.
Source Only	74.8	83.9	91.5	78.0	65.5	91.2	80.7
DAN	74.5	82.2	92.8	86.3	69.2	89.8	82.5
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
CAN	78.2	87.5	94.2	89.5	75.8	89.2	85.7
Ours	**78.3**	**89.7**	**95.3**	**91.5**	**77.2**	**92.3**	**87.4**

TABLE II: Recognition accuracy (%) on the ImageCLEF-DA dataset. All models utilize ResNet-50 as base architecture.

III. Experiment

In this section, the experiments on several benchmark datasets, Office-31 dataset [16], ImageCLEF-DA [10] dataset and Office-Home [19] dataset, are exploited for evaluation.

In all experiment, we follow the standard evaluation protocol for unsupervised domain adaptation [10]. Our implementation is based on the PyTorch platform. We utilize the pre-trained ResNet-50 as base model and re-train the parameters of high-level layers. We conduct our model to select parameters λ_1 and λ_2 and we fix $\lambda_1 = 1$ and $\lambda_2 = 1$ throughout all experiment. For MMD-based methods, we adopt Gaussian kernel with the bandwidth set as the median pairwise squared distances in training set. We evaluate the rank-1 classification accuracy and compare with DAN [8], DANN [3] and JAN [10]. Besides, on Office-31 dataset, we also compare with TCA [13], GFK [4], DDC [18], RTN [9], ADDA [17], EM [6], SimNet [15] and CAN [22]. On ImageCLEF-DA dataset, CAN [22] is used for comparison. Note that the backbone model of all methods is ResNet-50.

Results on Office-31 Dataset [16]. This dataset is a challenging benchmark for cross-domain object recognition. Three domains such as Amazon (A), Webcam (W) and Dslr (D) are included in this dataset, which contains 4,652 images from 31 object classes. With each domain worked as source and target alternatively, 6 cross-domain tasks are tested, *e.g.*, $A \rightarrow D$, $W \rightarrow D$, *etc.* In experiment, we follow the same experimental protocol as [8]. The recognition accuracies are

OfficeHome	Ar→Cl	Ar→Pr	Ar→Rw	Cl→Ar	Cl→Pr	Cl→Rw	Pr→Ar	Pr→Cl	Pr→Rw	Rw→Ar	Rw→Cl	Rw →Pr	Avg.
Source Only	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
Ours	**46.4**	**66.4**	**72.1**	**51.2**	**61.5**	**62.5**	**54.4**	**44.6**	**73.0**	**64.2**	50.4	**77.0**	**60.3**

TABLE III: Recognition accuracy (%) on Office-Home dataset. All models utilize ResNet-50 as base architecture.

reported in Table I. From the results, we observe that our method (86.7% in average) outperforms state-of-the-arts.

Results on ImageCLEF-DA Dataset [10]. The ImageCLEF-DA is a benchmark for ImageCLEF 2014 DA challenge, which contains 12 common categories shared by three public datasets: Caltech-256 (C), ImageNet ILSVRC 2012 (I) and Pascal VOC 2012 (P). In each domain, there are 50 images per class and totally 600 images are constructed. We evaluate all methods on 6 cross-domain tasks: *e.g.*, $I \rightarrow P$, $P \rightarrow I$, *etc.* We compare our method with the baseline model (ResNet-50) and the existing deep domain adaptation methods. The results are shown in Table II, from which we observe that our method (87.4% in average) still outperforms state-of-the-arts.

Results on Office-Home Dataset [19]. This is a new and challenging dataset for domain adaptation, which consists of 15,500 images from 65 categories coming from four very different domains: Artistic images (Ar), Clip Art (Cl), Product images (Pr) and Real-World images (Rw). There are 12 DA tasks on this dataset. We follow the same experimental protocol as before and compare against several recently reported results of well-known deep domain adaptation methods. The results are shown in Table III, from which we observe that our method achieves the best performance.

IV. Discussion

Ablation Study. We propose to utilize kNN to regularize CNN by minimizing a discrepancy loss $\mathcal{L}_{kNN}$, which includes two parts: confidence gap ℓ_1 loss on target data ($\mathcal{L}_{kNN-t}$) and self-regularization term on source data ($\mathcal{L}_{kNN-s}$). For better insight of the importance of the $\mathcal{L}_{kNN}$ loss, ablation analysis is presented. The results under different model variants with some terms removed are presented in Table IV. The baseline of Source Only denotes that only the source classifier based on cross-entropy loss is trained. DAN is another baseline with cross-entropy loss and marginal distribution alignment, and the performance is increased from 63.9% to 69.0%. From the results, we can observe that the performance is significantly increased from 69.0% to 76.5% in our method by jointly matching marginal and pseudo-conditional distribution. To demonstrate the effectiveness of kNN classifier regularization, the performance is decreased from 78.3% to 76.5% after removing the $\mathcal{L}_{kNN}$. The performance is decreased from 78.3% to 76.9% and 77.3% without $\mathcal{L}_{kNN-t}$ and $\mathcal{L}_{kNN-s}$, respectively. The proposed kNN regularized CNN model is verified.

Quantitative Distribution Discrepancy. In this section, we use the $\mathcal{A}$-distance [1] that jointly formulates source and target risk to measure the distribution discrepancy after domain adaptation. The proxy $\mathcal{A}$-distance is defined as $d_{\mathcal{A}} = 2(1-2\epsilon)$,

Office-31	A → W	D → A	W → A	Avg.
Source Only	68.4	62.5	60.7	63.9
DAN	80.5	63.6	62.8	69.0
Ours (w/o $\mathcal{L}_{kNN}$)	88.1	70.7	70.9	76.5
Ours (w/o $\mathcal{L}_{kNN-t}$)	90.3	69.4	71.1	76.9
Ours (w/o $\mathcal{L}_{kNN-s}$)	90.0	70.2	71.6	77.3
Ours	**92.2**	**71.1**	**71.7**	**78.3**

TABLE IV: Ablation study on the Office-31 dataset.

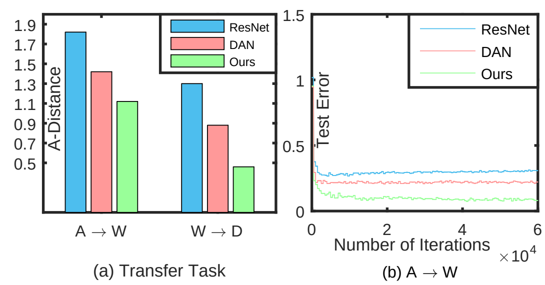

Fig. 3: Illustration of model analysis: (a) Quantitative distribution discrepancy measured by $\mathcal{A}$-distance after domain adaptation. (b) Convergence on the test errors of different models.

where ϵ is the classification error of a binary domain classifier (*e.g.*, SVM) to discriminate the source and target domain. Therefore, with the increasing discrepancy between domains, the error ϵ becomes smaller. Clearly, a large $\mathcal{A}$-distance denotes a large domain discrepancy. The distribution discrepancy analysis based on $\mathcal{A}$-distance in Office-31 dataset on tasks $A \rightarrow W$ and $W \rightarrow D$ is conducted by using ResNet, DAN and our model, respectively. Fig. 3 (a) shows $\mathcal{A}$-distance on different tasks by using different models. We can observe that the $\mathcal{A}$-distance between domains after using our model is much smaller than that of ResNet and DAN methods, which suggests that our model is more effective in reducing the domain discrepancy gap. By comparing the distribution discrepancy between $A \rightarrow W$ and $W \rightarrow D$, obviously, $W \rightarrow D$ has a much smaller $\mathcal{A}$-distance than $A \rightarrow W$. From the classification accuracy in Table I, the recognition rate of $W \rightarrow D$ is 99.8%, which is higher than $A \rightarrow W$ (92.2%). The reliability of $\mathcal{A}$-distance is demonstrated.

Convergence. We choose the task $A \rightarrow W$ in Office-31 dataset as an example and the test errors (misclassification rate) of different methods with the increasing number of iterations are shown in Fig. 3 (b). We can observe that the proposed model has the lowest test error of 0.078.

Parameter Sensitivity Analysis. There are two trade-off hyper-parameters λ_1 and λ_2 in our objective function. In order to further investigate the properties of the proposed method, the performance with respect to λ_1 and λ_2 is explored. Due to

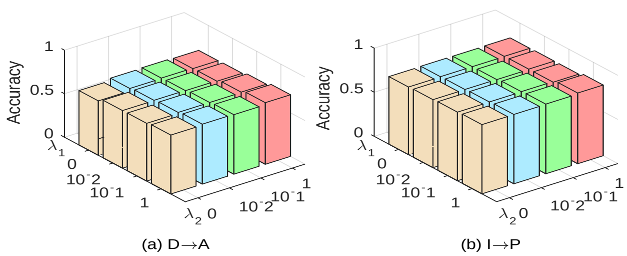

Fig. 4: Parameter sensitivity of our model: (a) Classification accuracy on task $A \rightarrow W$ in Office-31 dataset and (b) Classification accuracy on $I \rightarrow P$ in imageCLEF-DA dataset.

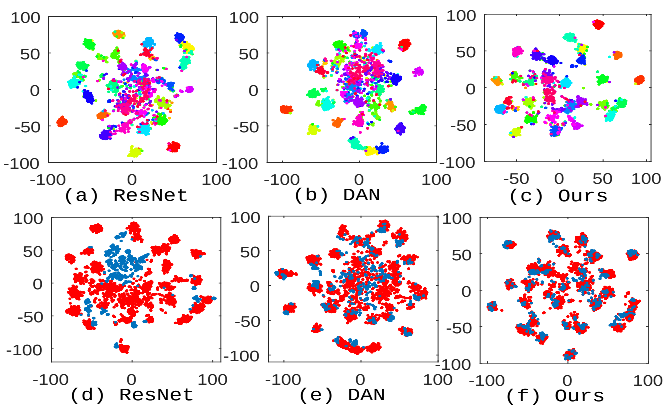

Fig. 5: Feature visualization with t-SNE algorithm. **First Row**: Visualization of *Amazon* source domain feature learned by (a) ResNet, (b) DAN and (c) Ours, respectively. **Second Row**: Visualization of source domain *Amazon* (red) and target domain *Webcam* (blue) learned by (d) ResNet, (e) DAN and (f) Ours, respectively.

the classification loss is dominant, we conduct parameter analysis on the Office-31 and imageCLEF-DA databases, by tuning the value of both parameters in the range of $\{0, 0.01, 0.1, 1\}$. The results are shown in Fig. 4, from which we observe that when λ_1 and λ_2 are set as 1, the best performance is achieved.

Feature Visualization. We employ the t-SNE visualization method [11] on the source domain and target domain in the $A \rightarrow W$ task from Office-31 dataset. The results of feature visualization for ResNet (traditional CNN), DAN (CNN after marginal distribution alignment) and our model are illustrated in Fig. 5, in which (a)-(c) represent the results of source features from 31 classes. We observe that our model can reserve better discrimination than other two baselines.

The features of target domain are visualized in Fig. 5 (d)-(f). We observe the different distribution across source and target features learned by ResNet. By aligning the marginal distribution between domains using DAN, the distribution discrepancy is reduced. Since DAN does not take into account the class conditional distribution, the class discrimination is not good. For ours, the features are well aligned between domains and more class discrimination is also reserved. This evidence confirms the superiority of our model in UDA tasks.

Potential Application to CAS Community. Neural network has been deployed in various applications [2], [12]. Circuits and Systems are the hardware basis of each intelligent system, in which the learning algorithms should be deployed. However, in real system application scenario, general machine learning algorithms embedded in the system have weak adaptation capability to new sensing input because of the unknown noise and uncertainty. Therefore, the proposed deep transfer learning algorithm may be a better candidate for CAS researchers engaging in intelligent systems for recognition.

V. Conclusion

In this paper, we focus on the under-studied CNN model bias problem for unsupervised domain adaptation. For alleviating the learning bias, we first propose to utilize the similarity based kNN classifier to regularize CNN by reducing the discrepancy error between them. To quantify the classifier discrepancy conditioned on target data, we use the ℓ_1 loss based on the Laplacian probabilistic distribution. For self-regularization of the source classifier, a kNN classifier guided cross-entropy loss is designed. Further, the marginal and conditional distribution across domains are aligned. Experiments demonstrate the superiority of our model over others.

References

[1] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.
[2] Y. Choi, D. Bae, J. Sim, S. Choi, M. Kim, L. S. Kim, Y. Choi, D. Bae, J. Sim, and S. Choi. Energy-efficient design of processing element for convolutional neural network. *IEEE Trans.CAS II Express Briefs*, 2017.
[3] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016.
[4] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pages 2066–2073, 2012.
[5] A. Gretton, K. Borgwardt, M. J. Rasch, B. Scholkopf, and A. J. Smola. A kernel method for the two-sample problem. In *NIPS*, 2008.
[6] G. Kang, L. Zheng, Y. Yan, and Y. Yang. Deep adversarial attention alignment for unsupervised domain adaptation: the benefit of target expectation maximization. *arXiv preprint arXiv:1801.10068*, 2018.
[7] J. Li, Y. Wu, and K. Lu. Structured domain adaptation. *IEEE Trans.Circuits and Systems for Video Technology*, 27:1700–1713, 2017.
[8] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105, 2015.
[9] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *NIPS*, 2016.
[10] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, pages 2208–2217, 2017.
[11] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *JMLR*, 9:2579–2605, 2008.
[12] A. Mansoori, M. Eshaghnezhad, and S. Effati. An efficient neural network model for solving the absolute value equations. *IEEE Trans.CAS II Express Briefs*, 2017.
[13] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Trans. Neural Networks*, 2011.
[14] Z. Peng, W. Zhang, N. Han, X. Fang, P. Kang, and L. Teng. Active transfer learning. *IEEE Trans. Circuits and Systems for Video Technology*, 2019. doi: 10.1109/TCSVT.2019.2900467.
[15] P. O. Pinheiro and A. Element. Unsupervised domain adaptation with similarity learning. In *CVPR*, pages 8004–8013, 2018.
[16] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. *ECCV*, pages 213–226, 2010.
[17] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.
[18] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv*, 2014.
[19] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan. Deep hashing network for unsupervised domain adaptation. In *CVPR*, 2017.
[20] L. Zhang, S. Wang, G.-B. Huang, W. Zuo, J. Yang, and D. Zhang. Manifold criterion guided transfer learning via intermediate domain generation. *IEEE Trans. Neural Networks and Learning Systems*, 2018.
[21] L. Zhang and D. Zhang. Robust visual knowledge transfer via extreme learning machine-based domain adpatation. *IEEE Trans. Image Processing*, 25(3):4959–4973, 2016.
[22] W. Zhang, W. Ouyang, W. Li, and D. Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *CVPR*, pages 3801–3809, 2018.