Appendix: Homeomorphism Alignment for Unsupervised Domain Adaptation

Lihua Zhou[1], Mao Ye[1,*], Xiatian Zhu[3], Siying Xiao[1], Xu-Qian Fan[2], Ferrante Neri[3]
[1]University of Electronic Science and Technology of China
[2]Jinan University
[3]University of Surrey
lihua.zhou@std.uestc.edu.cn, cvlab.uestc@gmail.com, xiatian.zhu@surrey.ac.uk
2018270101016@std.uestc.edu.cn, txqfan@jnu.edu.cn, f.neri@surrey.ac.uk

1. Algorithm

Our method is summarized in Algorithm 1. In each iteration, the INN g and backbone network, which consists of feature extractor F and classifier C, are both trained. The loss functions are shown as follows:

$$\min_{g} \text{Loss}_{Sew} = DM(\boldsymbol{f}^{s2t}, \boldsymbol{f}^{t}) + DM(\boldsymbol{f}^{t2s}, \boldsymbol{f}^{s}), \quad (1)$$

$$\begin{aligned} \min_{F,C} Loss_{S} + Loss_{T} = \mathcal{L}^{ce}(C(\boldsymbol{f}^{s}), \boldsymbol{y}^{s}) + \\ \mathcal{L}^{ce}(C(\boldsymbol{f}^{s2t}), \boldsymbol{y}^{s}) + \mathcal{L}_{C}(C(\boldsymbol{f}^{t}), C(\boldsymbol{f}^{t2s})), \end{aligned} \quad (2)$$

where $DM(\cdot,\cdot)$ refers to any existing distribution matching method, $\mathcal{L}^{ce}(\cdot,\cdot)$ denotes the cross entropy function, and $\mathcal{L}_{C}(\cdot,\cdot)$ is a consistency constraint.

Algorithm 1 HMA

Input: Source domain $D_s = \{(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s)\}_{i=1}^{n_s}$, target domain $D_t = \{(\boldsymbol{x}_i^t)\}_{i=1}^{n_t}$, the epoch number T, the mini-batch number M.

Output: An adapted model.

Procedure:

```
for $t$ = 1:$T$ do
    for $m$ = 1:$M$ do
        Forward a mini-batch through the feature extractor $F$ and get source features $\boldsymbol{f}^s$ and target features $\boldsymbol{f}^t$;
        Generate transformed source features $\boldsymbol{f}^{t2s}$ and transformed target features $\boldsymbol{f}^{s2t}$ by INN;
        Select a domain adaptation method and train INN based on Eq. 1;
        Train the backbone network based Eq.2;
    end for
end for
return Adapted model.
```

Table 1. Up-bound performance probing: Comparing different distribution alignment strategies on *Office-31* using the ground-truth target sample labels. SMM: Statistic moment matching; AL: Adversarial learning; OP: Optimal transport; SL: Self-supervised learning; BA:Bijection alignment.

	Component	A→D	A→W	D→A	W→A
1	SMM	99.9±0.1	99.9±0.0	92.6±0.2	93.8±0.1
2	AL	99.2±0.1	99.8±0.1	90.9±0.2	92.3±0.1
3	OP	96.2±0.1	98.8±0.1	89.9±0.2	90.9±0.1
4	SL	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**
5	BA	97.8±0.2	98.9±0.1	90.7±0.1	93.2±0.2
6	**Ours**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**

2. Implementation of methods in Table 1

In Table 1, we evaluate classical domain adaptation strategies given ground-truth labels. Since the ground-truth labels are used, we need to modify these methods accordingly as follows.

For the first row in Table 1, CAN [4] is selected to test the statistic moment matching strategy, one of the best statistical moment matching methods. Specifically, it uses the clustering algorithm to pseudo-label all target domain samples, and then uses the CAS strategy to sample target domain samples with high-confidence pseudo-label and source samples; Finally, it minimizes the inter-class cross domain discrepancy and maximizes the intra-class cross domain discrepancy as follows:

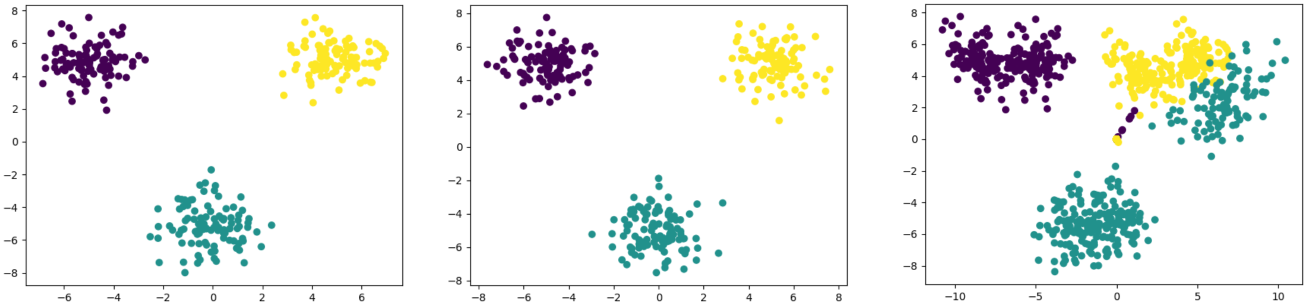
$$\begin{aligned} \min_{F} Loss_{CAN} = \sum_{c=1}^{C} \mathcal{MMD}(\boldsymbol{f}^{s,c}, \boldsymbol{f}^{t,\hat{c}}) \\ - \sum_{c_1=1}^{C} \sum_{c_2 \neq c_1}^{C} \mathcal{MMD}(\boldsymbol{f}^{s,c_1}, \boldsymbol{f}^{t,\hat{c_2}}), \end{aligned} \quad (3)$$

where $\mathcal{MMD}(A, B)$ represents the MMD discrepancy between A and B, $\boldsymbol{f}^{s,c}$ represents the source features with true label c and $\boldsymbol{f}^{t,\hat{c}}$ represents the target features with pseudo

[*]corresponding author.

Table 2. Comparisons with the state-of-the-art methods on *DomainNet* dataset. Metric: classification accuracy (%); Backbone: ResNet-50. For each cross-domain pair, the source/target domains are specified in the corresponding row/column fields.

ResNet	clp	inf	pnt	qdr	rel	skt	Avg.	MCD	clp	inf	pnt	qdr	rel	skt	Avg.	BNM	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	14.2	29.6	9.5	43.8	34.3	26.3	clp	-	15.4	25.5	3.3	44.6	31.2	24.0	clp	-	12.1	33.1	6.2	50.8	40.2	28.5
inf	21.8	-	23.2	2.3	40.6	20.8	21.7	inf	24.1	-	24.0	1.6	35.2	19.7	20.9	inf	26.6	-	28.5	2.4	38.5	18.1	22.8
pnt	24.1	15.0	-	4.6	45.0	29.0	23.5	pnt	31.1	14.8	-	1.7	48.1	22.8	23.7	pnt	39.9	12.2	-	3.4	54.5	36.2	29.2
qdr	12.2	1.5	4.9	-	5.6	5.7	6.0	qdr	8.5	2.1	4.6	-	7.9	7.1	6.0	qdr	17.8	1.0	3.6	-	9.2	8.3	8.0
rel	32.1	17.0	36.7	3.6	-	26.2	23.1	rel	39.4	17.8	41.2	1.5	-	25.2	25.0	rel	48.6	13.2	49.7	3.6	-	33.9	29.8
skt	30.4	11.3	27.8	3.4	32.9	-	21.2	skt	37.3	12.6	27.2	4.1	34.5	-	23.1	skt	54.9	12.8	42.3	5.4	51.3	-	33.3
Avg.	24.1	11.8	24.4	4.7	33.6	23.2	20.3	Avg.	28.1	12.5	24.5	2.4	34.1	21.2	20.5	Avg.	37.6	10.3	31.4	4.2	40.9	27.3	25.3
SWD	clp	inf	pnt	qdr	rel	skt	Avg.	CAN	clp	inf	pnt	qdr	rel	skt	Avg.	HMA(CAN)	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	14.7	31.9	10.1	45.3	36.5	27.7	clp	-	17.3	38.4	8.6	53.2	39.7	31.4	clp	-	18.9	43.4	9.9	54.7	45.4	34.5
inf	22.9	-	24.2	2.5	33.2	21.3	20.0	inf	33.5	-	34.2	4.7	51.2	26.7	30.1	inf	35.9	-	37.2	5.7	54.5	30.8	32.8
pnt	33.6	15.3	-	4.4	46.1	30.7	26.0	pnt	39.9	14.5	-	8.2	59.4	33.7	31.1	pnt	42.6	14.9	-	10.8	61.4	35.1	33.0
qdr	15.5	2.2	6.4	-	11.1	10.2	9.1	qdr	25.9	3.0	10.8	-	13.7	14.9	13.7	qdr	31.0	5.8	15.0	-	15.9	16.2	16.8
rel	41.2	18.1	44.2	4.6	-	31.6	27.9	rel	52.4	16.9	46.3	3.9	-	41.9	32.3	rel	53.1	18.8	47.0	4.1	-	43.0	33.2
skt	44.2	15.2	37.3	10.3	44.7	-	30.3	skt	53.9	17.5	45.9	15.5	57.6	-	38.1	skt	55.8	18.3	47.3	17.5	59.3	-	39.6
Avg.	31.5	13.1	28.8	6.4	36.1	26.1	23.6	Avg.	41.1	13.8	35.1	8.2	47.0	31.4	29.5	Avg.	43.7	15.3	38.0	9.6	49.2	34.1	**31.7**



a) original data

b) data transformed by homeomorphism

c) data transformed by linear network

Figure 1. The empirical visualization of homomorphism and linear network.

label c. Given the ground-truth labels of the target domain, there is no need for the pseudo labels, and we can directly sample all target samples to perform distribution alignment as:

$$\begin{aligned}\min_F Loss^{mod}_{CAN} = &\sum_{c=1}^{C} \mathcal{MMD}(\boldsymbol{f}^{s,c}, \boldsymbol{f}^{t,c}) \\ &- \sum_{c_1=1}^{C} \sum_{c_2 \neq c_1}^{C} \mathcal{MMD}(\boldsymbol{f}^{s,c_1}, \boldsymbol{f}^{t,c_2}),\end{aligned} \tag{4}$$

where $\boldsymbol{f}^{t,c}$ represents the target features with true label c. In this case, the feature extractor is retrained by Eq. 4; the source classifier is retrained by source samples.

For the second row in Table 1, CDAN [7] is selected to test adversarial learning strategy. CDAN considers the prediction of the classifier carry the discriminative information useful for aligning the conditional distribution between two domains. Specifically, it first introduces a domain discriminator D to perform domain classification. The input of the domain discriminator is the outer product of features and predictions and the loss function is defined as follows:

$$\begin{aligned}\min_F \max_D Loss_{CDAN} = &\log[D(\boldsymbol{f}^s_i \otimes \boldsymbol{p}^s_i)] \\ &+ \log[1 - D(\boldsymbol{f}^t_i \otimes \boldsymbol{p}^t_i)],\end{aligned} \tag{5}$$

where $\otimes$ is the outer product, $\boldsymbol{p}^s_i$ is the prediction of i-th source sample and $\boldsymbol{p}^t_i$ is the prediction of i-th target sample. While in our test, ground-truth labels are available during the training, we perform an one-hot operation on the ground-truth labels $\boldsymbol{y}^s_i$ and $\boldsymbol{y}^t_i$ to get $\boldsymbol{l}^s_i$ and $\boldsymbol{l}^t_i$, and train the feature extraction network and the discrimination network in the following way:

$$\begin{aligned}\min_F \max_D Loss^{mod}_{CDAN} = &\log[D(\boldsymbol{f}^s_i \otimes \boldsymbol{l}^s_i)] \\ &+ \log[1 - D(\boldsymbol{f}^t_i \otimes \boldsymbol{l}^t_i)].\end{aligned} \tag{6}$$

In this case, the feature extractor is retrained by Eq. (6); The source classifier is retrained by source samples.

For the third row in Table 1, it reports the optimal transport strategy. With this strategy, DeepJDOT [2] minimizes

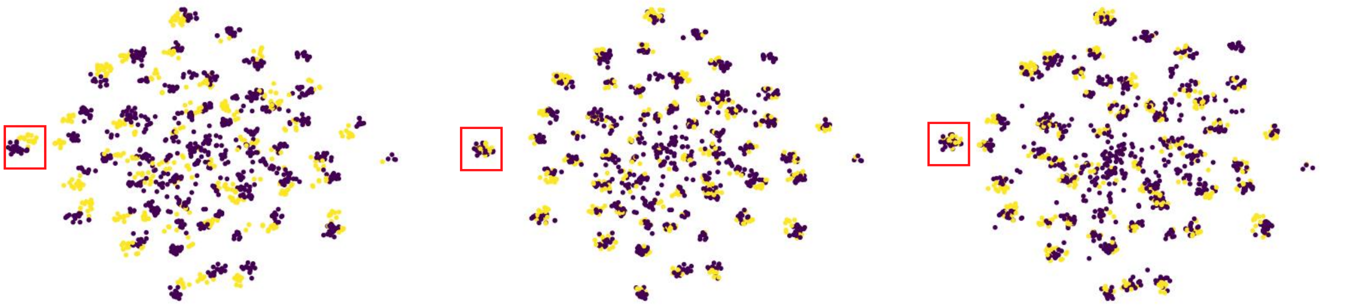


Figure 2. (Left) Distribution of $\boldsymbol{f}^s$ and $\boldsymbol{f}^t$; (Middle) Distribution of $\boldsymbol{f}^{s2t}$ and $\boldsymbol{f}^t$; (Right) Distribution of $\boldsymbol{f}^s$ and $\boldsymbol{f}^{t2s}$. Purple: source features; Yellow: target features. Red box: highlights.

the discrepancy of joint deep feature/labels domain distributions as follows:

$$\min_{F,M} Loss_{OP} = \sum_i \sum_j m_{i,j}(\alpha\|\boldsymbol{f}_i^s - \boldsymbol{f}_j^t\|^2 + \beta L(\boldsymbol{y}_i^s, \hat{\boldsymbol{y}}_j^t)), \tag{7}$$

where M means coupling matrix and $m_{i,j}$ is the element of its row i and column j, α and β are two hyperparameters, $L(\cdot,\cdot)$ is a similarity function, such as hinge or cross-entropy. The optimization by Eq. 7 is generally divided into two steps: first optimizing the coupling matrix M, and then optimizing the feature extractor F. When the true target labels $\boldsymbol{y}_i^t$ are given, it can be trained as follows:

$$\min_{F,M} Loss_{OP}^{mod} = \sum_i \sum_j m_{i,j}(\alpha\|\boldsymbol{f}_i^s - \boldsymbol{f}_j^t\|^2 + \beta L(\boldsymbol{y}_i^s, \boldsymbol{y}_j^t)), \tag{8}$$

In this case, our training is also divided into two steps, which first finds coupling matrix M and optimizes the feature extractor. The source classifier is retrained by source samples.

For the fourth row in Table 1, it reports the self-supervised training strategy. Traditional methods based on this strategy [6] usually assign target sample a pseudo-label $\hat{\boldsymbol{y}}_i^t$, and use pseudo-labels to train the model as follows:

$$\min_{F,C} Loss_{SELF} = \mathcal{L}^{ce}(\boldsymbol{p}_i^s, \boldsymbol{y}_i^s) + \mathcal{L}^{ce}(\boldsymbol{p}_i^t, \hat{\boldsymbol{y}}_i^t), \tag{9}$$

where C means the classifier. When the true target labels $\boldsymbol{y}_i^t$ are given, it can be directly supervised train the model as follows:

$$\min_{F,C} Loss_{SELF}^{mod} = \mathcal{L}^{ce}(\boldsymbol{p}_i^s, \boldsymbol{y}_i^s) + \mathcal{L}^{ce}(\boldsymbol{p}_i^t, \boldsymbol{y}_i^t), \tag{10}$$

In this case, the feature extractor and source classifier are retrained by Eq. 10.

For the fifth row in Table 1, it uses two different networks to learn two transformations, which maps the source features to the target feature space and vice versa. Specifically, two linear networks $F_{s2t}(\cdot)$ and $F_{t2s}(\cdot)$ are introduced, and we have $\boldsymbol{f}^{s2t} = F_{s2t}(\boldsymbol{f}^s)$, $\boldsymbol{f}^{t2s} = F_{t2s}(\boldsymbol{f}^t)$. We hope the transformed features can be aligned to original features in their feature spaces respectively. In this test, we also have true labels from both source domain and target domain and use CAN to align the distributions between transformed features and original features, which is shown as follows:

$$\begin{aligned} &\min_{F_{s2t},F_{t2s}} Loss_{Doublemap} = \\ &\sum_{c=1}^{C} \mathcal{MMD}(\boldsymbol{f}^{s,c}, \boldsymbol{f}^{t2s,c}) - \sum_{c_1=1}^{C} \sum_{c_2\neq c_1}^{C} \mathcal{MMD}(\boldsymbol{f}^{s,c_1}, \boldsymbol{f}^{t2s,c_2}) \\ &+ \sum_{c=1}^{C} \mathcal{MMD}(\boldsymbol{f}^{s2t,c}, \boldsymbol{f}^{t,c}) - \sum_{c_1=1}^{C} \sum_{c_2\neq c_1}^{C} \mathcal{MMD}(\boldsymbol{f}^{s2t,c_1}, \boldsymbol{f}^{t,c_2}). \end{aligned} \tag{11}$$

In this case, the feature extractor is retrained by Eq. 11; The source classifier is retrained by source samples.

For the sixth line in Table 1, which is our method based on ground-truth label, we introduce invertible neural network g to connect two feature spaces. Specifically, the transformed features can be obtained by g as $\boldsymbol{f}^{s2t} = g(\boldsymbol{f}^s)$ and $\boldsymbol{f}^{t2s} = g^{-1}(\boldsymbol{f}^t)$. Due to the ground-truth target labels are available. Therefore, We just need to modify our sewing up operation to the following:

$$\begin{aligned} &\min_{g} Loss_{HMA} = \\ &\sum_{c=1}^{C} \mathcal{MMD}(\boldsymbol{f}^{s,c}, \boldsymbol{f}^{t2s,c}) - \sum_{c_1=1}^{C} \sum_{c_2\neq c_1}^{C} \mathcal{MMD}(\boldsymbol{f}^{s,c_1}, \boldsymbol{f}^{t2s,c_2}) \\ &+ \sum_{c=1}^{C} \mathcal{MMD}(\boldsymbol{f}^{s2t,c}, \boldsymbol{f}^{t,c}) - \sum_{c_1=1}^{C} \sum_{c_2\neq c_1}^{C} \mathcal{MMD}(\boldsymbol{f}^{s2t,c_1}, \boldsymbol{f}^{t,c_2}). \end{aligned} \tag{12}$$

In this case, the feature extractor and source classifier are retrained in two spaces.

Table 3. Different loss functions for consistency constraint on *Office-31* and *Office-home*. CE: Cross Entropy; L_2: L_2-Norm.

Loss	A→D	A→W	D→A	D→W	W→A	W→D	Office-home
CE	95.8	94.9	79.4	99.1	77.8	100.0	72.9
L_2	95.8	95.1	79.3	99.3	77.6	100.0	73.2

Table 4. Block number analysis on *Office-31* (first two rows) and *Office-home* (last two rows) . HMA(DAN): Sewing up by DAN; HMA(CAN): Sewing up by CAN.

Block number	1	2	3	4	5
HMA(DAN)	78.2	82.4	82.9	83.5	83.9
HMA(CAN)	87.6	89.4	90.3	90.7	91.2
HMA(DAN)	56.9	59.2	60.5	61.7	62.4
HMA(CAN)	69.2	70.9	71.8	72.6	73.2

Table 5. Test on *Office-31*. $\boldsymbol{f}^t$: classify $\boldsymbol{f}^t$ directly; $\boldsymbol{f}^{t2s}$: transform $\boldsymbol{f}^t$ to $\boldsymbol{f}^{t2s}$ then classify $\boldsymbol{f}^{t2s}$; $\boldsymbol{f}^t$+$\boldsymbol{f}^{t2s}$: ensemble these two strategies.

Strategy	A→D	A→W	D→A	D→W	W→A	W→D
$\boldsymbol{f}^t$	95.3	94.7	78.5	98.9	77.2	100.0
$\boldsymbol{f}^{t2s}$	95.1	94.7	78.7	99.2	76.9	100.0
$\boldsymbol{f}^t$+$\boldsymbol{f}^{t2s}$	95.8	95.1	79.3	99.3	77.6	100.0

3. Experiments

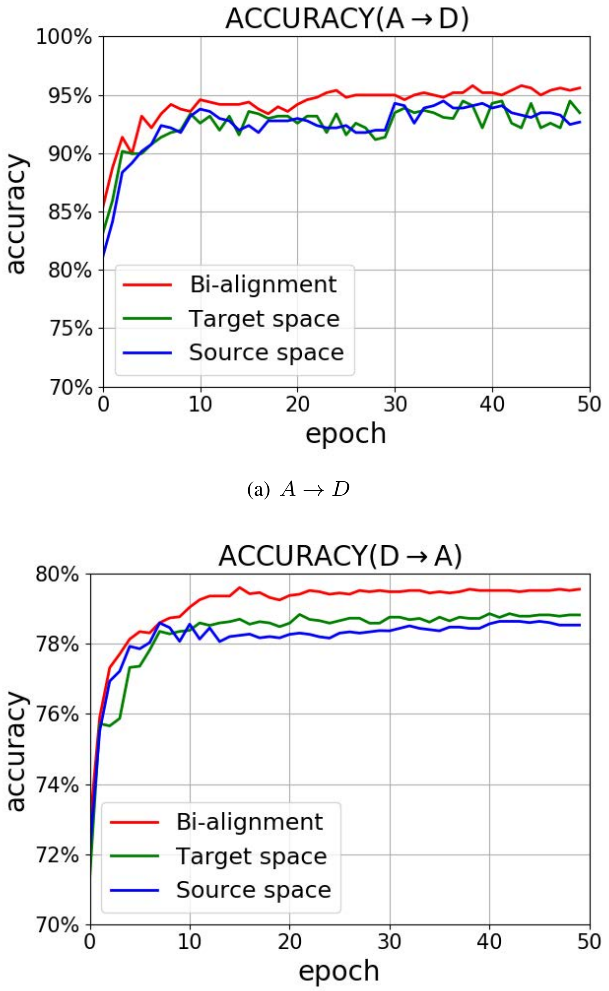
3.1. Comparisons to State-of-the-Art on DomainNet

DomainNet [8] is one of the most challenging datasets in domain adaptation. It contains about 600 thousand images in 345 categories from 6 domains: Clipart (C), Infograph (I), Painting (P), Quickdraw (Q), Real (R) and Sketch (S). We compare our method HMA(CAN) with existing state-of-the-art methods: MCD [9], BNM [1], SWD [5] and CAN [4]. ResNet-50 is used as backbone for all methods. As shown in Table 2, our method HMA(CAN) surpasses all the previous alternatives by a large margin. This verifies the generic advantage of our approach in this more challenging larger-scale benchmark.

3.2. Model analysis

Visualization of homeomorphism. For conceptual illustration of homeomorphism, we experiment with hand-designed toy data. Concretely, we first construct 100 2-dimensional feature points for three different clusters respectively, as shown in Figure 1(a) in purple, yellow and green. We then transform these points with an INN based homeomorphism mapping. As we observe in Figure 1(b), the transformed points still preserve the structural cluster/group information. At the same time, we transform these points with a linear network, as shown in Figure 1(c). The transformed points do not preserve the structural cluster/group information, as seen that the yellow and green points are mixed. Please note both INN and linear network are not trained, but only initialized by Gaussian distribution.

In addation, we further visualize the alignments in each space on the task A→R of Office-Home. As shown in Fig.2(Left), both domains ($\boldsymbol{f}^s$, $\boldsymbol{f}^t$) have similar topological structure, verifying the realization of homeomorphism. The distances of the same category in Fig.2(Middle) and Fig.2(Right) are smaller than that in Fig.2(Left), showing the alignment by homeomorphism.



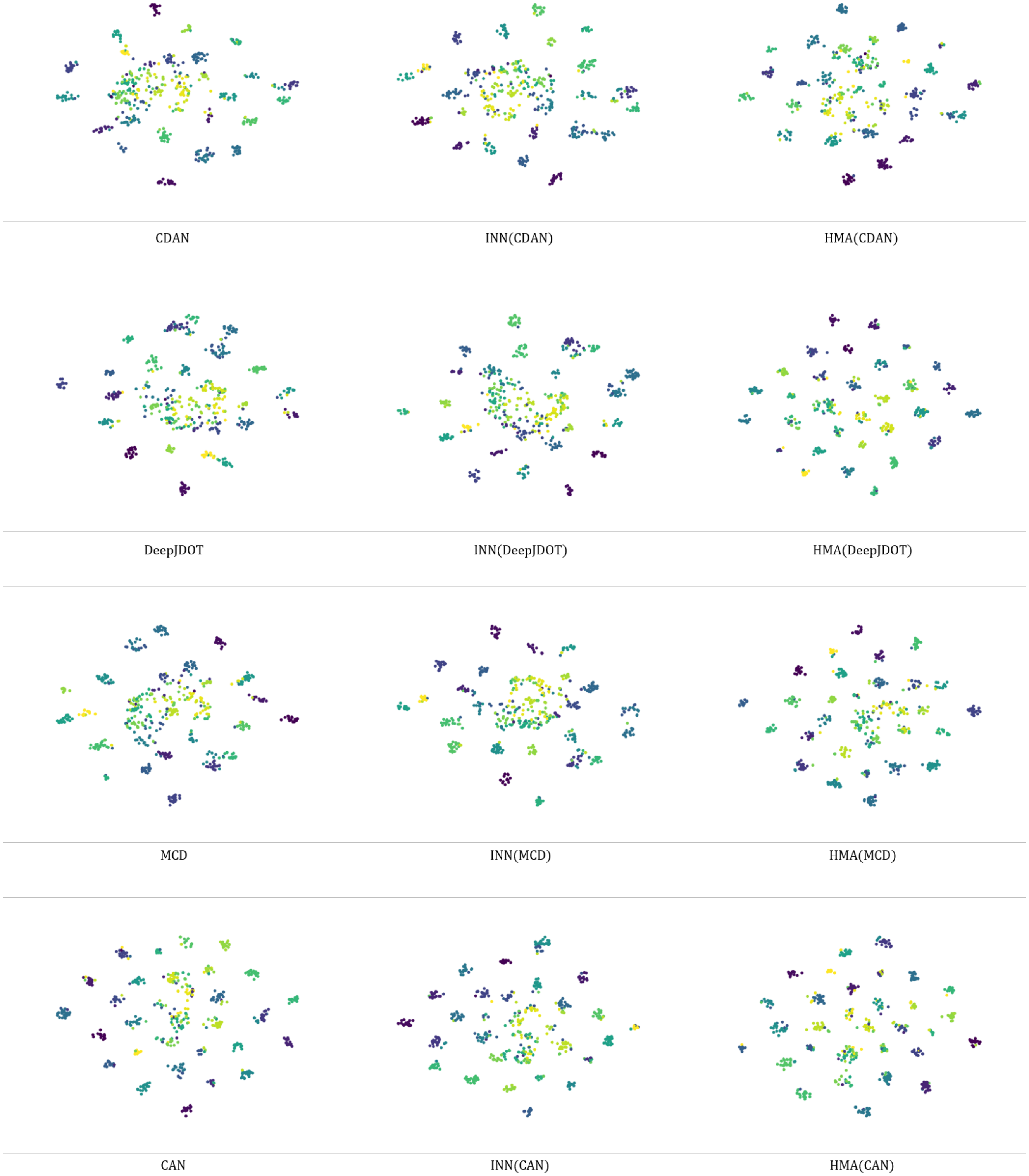
(a) $A \to D$

(b) $D \to A$

Figure 3. The accurary of different sewing up strategies using INN on *Ofiice-31*. The curves named Target space and Source space are unilateral sewing up strategies which are performed in the target feature space and source feature space respectively. The curve named Bi-alignment means the bilateral sewing up strategy.

Loss function for consistency constraint. For $\mathcal{L}_C(\cdot,\cdot)$, we use L_2-Norm to implement the consistency constraint on the unlabeled target features $\boldsymbol{f}^t$ and $\boldsymbol{f}^{t2s}$. To evaluate the effect of this loss function selection, we further test cross entropy on *Office-31* and *Office-home* with HMA(CAN). As shown in Table 3, the performance of our method is marginally affected by the loss function selection, suggesting the stability and flexibility of our model.

How many blocks of INN do we need? The forward and invertible process of INN for each block are shown



CDAN INN(CDAN) HMA(CDAN)

DeepJDOT INN(DeepJDOT) HMA(DeepJDOT)

MCD INN(MCD) HMA(MCD)

CAN INN(CAN) HMA(CAN)

Figure 4. Visualization of ablation study using t-SNE on the task $W \rightarrow A$ of *Office-31*.

in method section, so we need to discuss how many INN blocks we need. As shown in Table 4, the average accu-

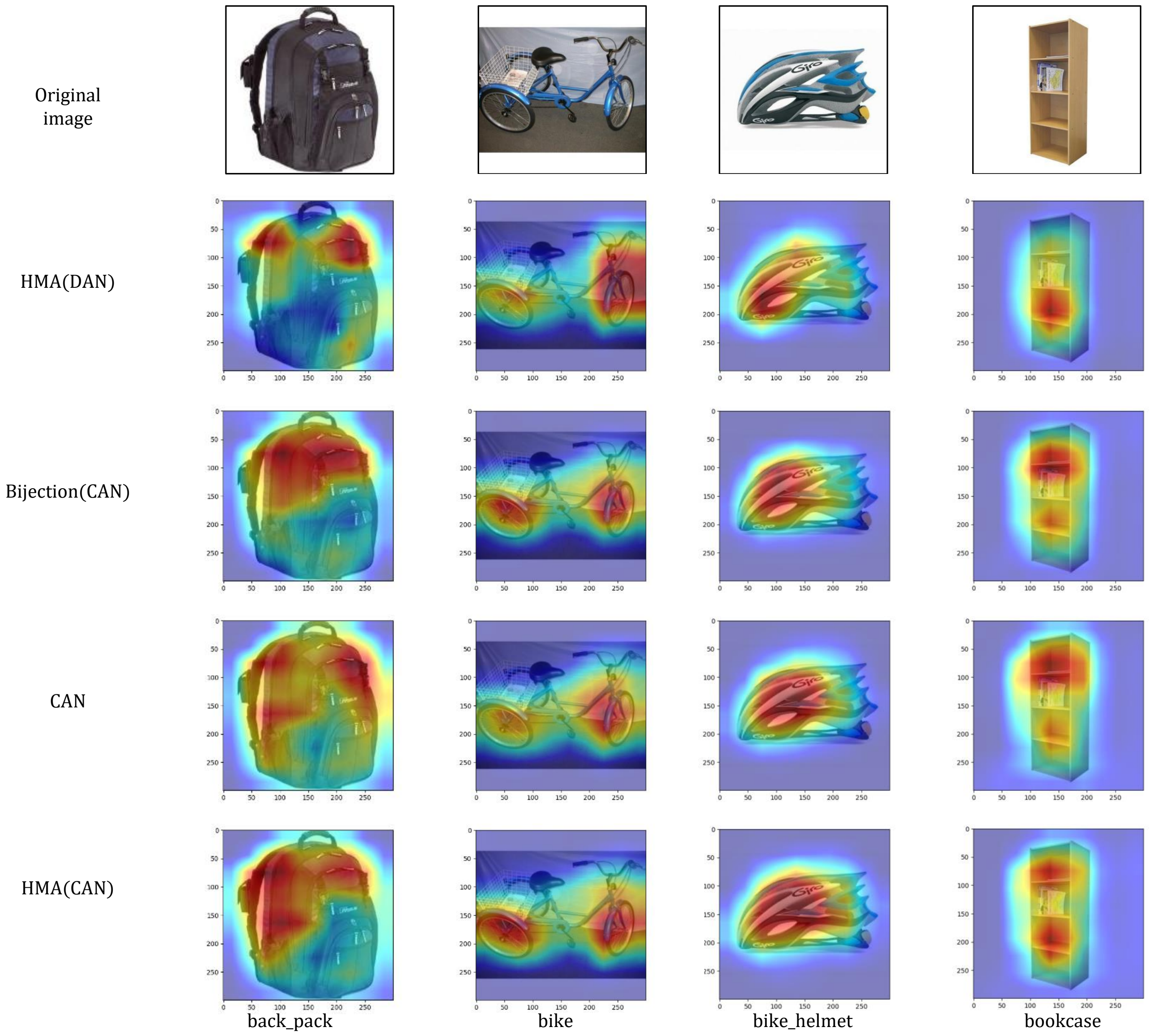


Figure 5. Attention visualization on *Office-31* task $W \rightarrow A$.

racy on *Office-31* and *Office-home* are reported. It can be found that when the block number is increased from 1 to 2, the performance of both HMA(DAN) and HMA(CAN) also greatly improve; While further increased to 5 blocks, the performance increase starts to saturate relatively. This is because when the block number is 1, the $\boldsymbol{y}_1$ in the output of the INN and the $\boldsymbol{x}_1$ in the input are linearly related, *i.e.* $\frac{\partial \boldsymbol{y}_1}{\partial \boldsymbol{x}_1} = I$ where I is the identity matrix. When the block number becomes 2, there is no such linear relationship, which makes the network has more capacity. In addition, as the number of blocks in the network increases, the nonlinearity of the network also becomes stronger, resulting in better results. Of course, before the learning ability is saturated, more blocks will definitely have better learning ability, but considering the computational overhead, we finally chose 5 blocks.

Unilateral sewing up or bilateral sewing up? In our method, when the distributions between $\boldsymbol{f}^t$ and $\boldsymbol{f}^{s2t}$ are aligned, the discrepancy between $\boldsymbol{f}^s$ and $\boldsymbol{f}^{t2s}$ can also be minimized due to the reversibility of the INN, and vice versa. But in our method, we do not use this unilateral sewing up but bilateral sewing up, i.e., $\boldsymbol{f}^s$ and $\boldsymbol{f}^{t2s}$; $\boldsymbol{f}^t$ and $\boldsymbol{f}^{s2t}$ are aligned as shown in Eq. 1. We compare three strategies: unilateral sewing up: only alignment between $\boldsymbol{f}^t$ and $\boldsymbol{f}^{s2t}$ in target feature space or only alignment $\boldsymbol{f}^s$ and $\boldsymbol{f}^{t2s}$ in source feature space; and bilateral sewing up where the above mentioned pairs are all aligned. We select

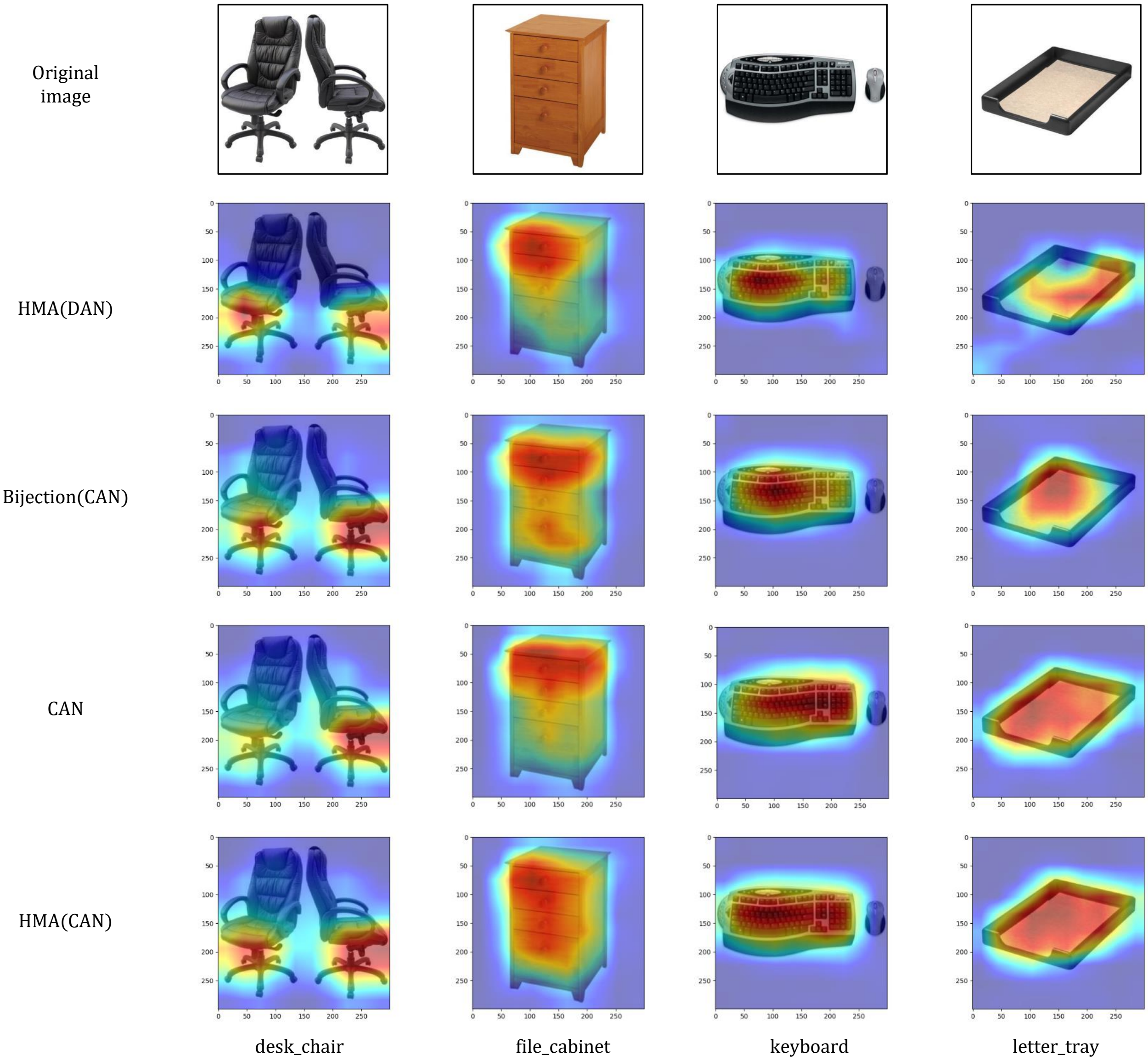


Figure 6. Attention visualization on *Office-31* task $W \rightarrow A$.

HMA(CAN) as the baseline and conduct experiments on A→D and D→A tasks of Office-31. From the experimental results, the bilateral sewing up can make training faster than other two strategies. In addition, we also find that bilateral sewing up can get better performance compared with unilateral sewing up.

How to use our model? Our method do alignment in two spaces, it is natural to ask a problem in which space using our model. There are three strategies: using our model in the target feature space $\boldsymbol{f}^t$, or in the source feature space $\boldsymbol{f}^{t2s}$ or in both source and target feature spaces where the average prediction is considered as the final result. We test these three strategies on Office-31 dataset using HMA(CAN), which is shown in Table 5. From the experimental results, the effect of adopting the ensemble strategy is slightly better than others, so for using our model, we adopt this ensemble strategy.

Visual analysis by t-SNE. To intuitively understand the proposed HMA, we use t-SNE [11] to visualize the classification results on *Office-31* based on four different strategies: *adversarial learning* (CDAN), *optimal transport* (DeepJDOT), *bi-classifier adversarial learning* (MCD) and *statistic moment matching* (CAN). as shown in Fig. 4. From left to right, the visualization images represent the visualization results of the baseline method, the alignment results using INN on the baseline method, and the visualization

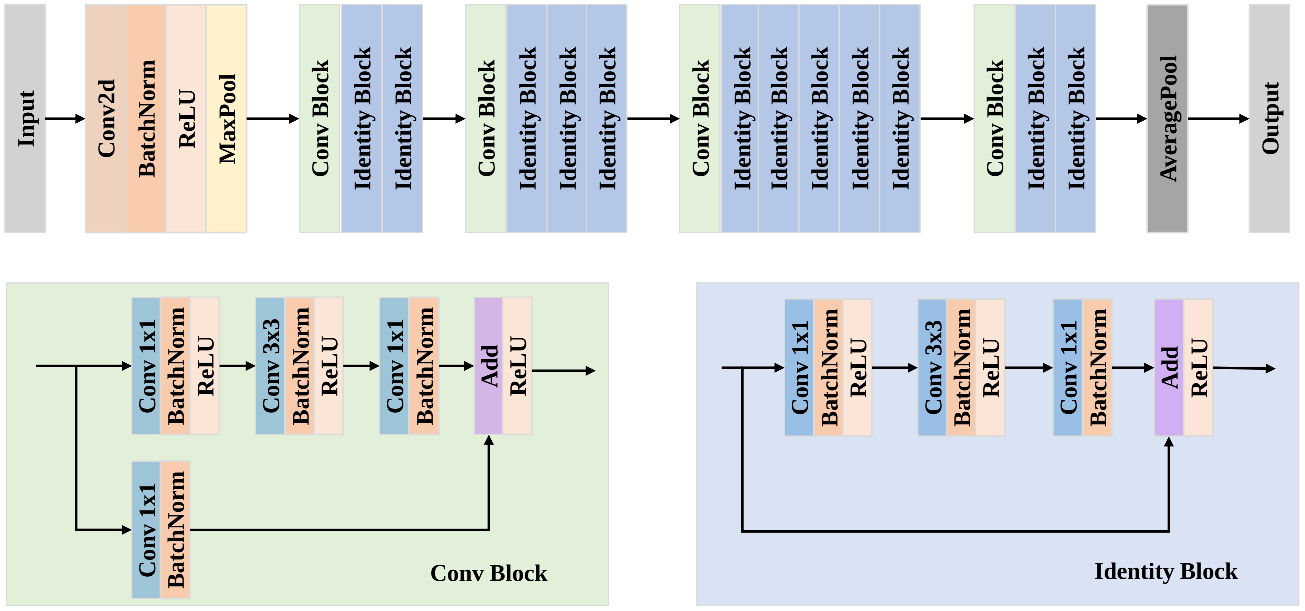


Figure 7. Network structure of Resnet-50.

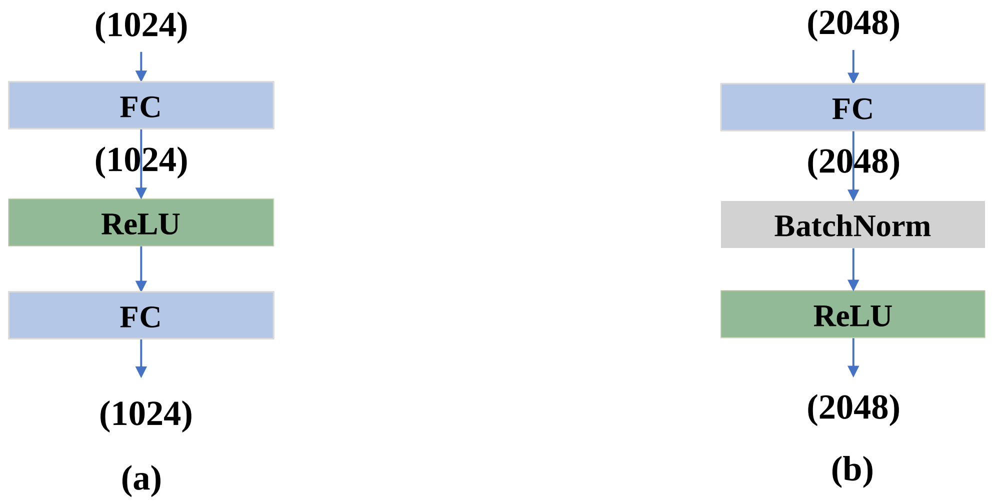


Figure 8. (a) Network structure of $s(\cdot)$ and $t(\cdot)$ in HMA. (b) Network structure of double mapping $F_{s2t}(\cdot)$ and $F_{t2s}(\cdot)$.

results of our final proposed method, respectively. From Fig. 4, it can be seen that only using INN to sew up two domains can achieve similar results with the previous alignment method. HMA shows a huge improvement over other visualization results. This is because in addition to the distribution alignment using INN, our HMA approach further applies the property of INN to train the feature extractor and classifier and yields better performance.

Visual analysis by Grad-CAM. We make visualization analysis by Grad-CAM [10] on *Office-31* task $W \rightarrow A$ in Fig.5 and Fig.6. We randomly select 8 categories. For each category, one image is randomly selected in activation mapping visualization. It is evident that, the attention of HMA(DAN) and Bijection(CAN) is largely not complete. Compared with the above two methods, CAN is slightly better, but still lags behind in the accuracy. Our HMA(CAN) can best estimate the attention overall.

Network structure Here we will detail the neural network we use. For the feature extractor, ResNet [3] is used, but its original last layer, a fully connected linear layer for classification, is removed. It is worth noting that the dimension size of features yielded by feature extractor of both ResNet-50 and ResNet-101 is 2048. The structure is detailed as follows.

For the classifier, a fully connected linear layer is constructed for suit our tasks, which maps features to predictions. The dimension size of predictions is the category number, specific for different datasets. Specifically, the dimensions of prediction are 31, 65, 12, 345 in Office-31, Office-home, Visda-17 and domainnet respectively. For the INN, the affine network is used. Specifically, it consists of two two-layers linear networks $s(\cdot)$ and $t(\cdot)$. The structure of $s(\cdot)$ and $t(\cdot)$ are the same. Specifically, the network $s(\cdot)$ consists of two fully connected neural networks and a ReLU function. The detail is given in Fig.8(a).

We also discuss that using two different linear networks $F_{s2t}(\cdot)$ and $F_{t2s}(\cdot)$ to learn the mappings between two feature spaces. The structures of $F_{s2t}(\cdot)$ and $F_{t2s}(\cdot)$ are identical, composed of four blocks. Each block consists of a fully connected neural networks with a batchnorm and a relu function. The specific structure is given in Fig.8(b).

References

[1] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3941–3950, 2020. 4

[2] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 447–463, 2018. 2

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 8

[4] Guoliang Kang, Lu Jiang, Yi Yang, and Alexander G Hauptmann. Contrastive adaptation network for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4893–4902, 2019. 1, 4

[5] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10285–10295, 2019. 4

[6] Jian Liang, Dapeng Hu, and Jiashi Feng. Domain adaptation with auxiliary target domain-oriented classifier. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16632–16642, 2021. 3

[7] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in neural information processing systems*, 2018. 2

[8] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1406–1415, 2019. 4

[9] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3723–3732, 2018. 4

[10] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 8

[11] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 7