HOMEOMORPHISM ALIGNMENT IN TWO SPACES FOR UNSUPERVISED DOMAIN ADAPTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The existing unsupervised domain adaptation methods rely on aligning the features from the source and target domains explicitly or implicitly in a common space (i.e., the domain invariant space). Explicit distribution matching ignores the discriminability of the learned features, while implicit counterpart such as self-supervised learning suffers from pseudo-label noises. With distribution alignment, it is challenging to acquire a common space which maintains fully the discriminative structure of both source and target domains. We propose a novel *HomeomorphisM Alignment* (HMA) approach characterized by aligning the source and target data in two separate spaces. Specifically, an invertible neural network based homeomorphism is constructed. Distribution matching is then used as a sewing up tool for connecting this homeomorphism mapping between the source and target feature spaces. Theoretically, we show that this mapping can preserve data topological structure (*e.g.*, the cluster/group structure). This property allows us to adapt the model by leveraging simply the original and transformed features of source data in a supervised manner (*e.g.*, cross entropy loss), and those of target domain in an unsupervised manner (*e.g.*, prediction consistency loss). Extensive experiments demonstrate that our method can achieve the state-of-the-art results.

1 INTRODUCTION

Deep learning has revolutionized the progress of machine learning and computer vision (*e.g.*, object recognition (He et al., 2016)). However, this advance relies heavily on a large quantity of manually labeled data, which could be prohibitively expensive or even impossible to collect in many scenarios. To mitigate this issue, there is a strong motivation to exploit pre-existing labeled data (*i.e.*, the source domain) for a target domain without any label annotation. Due to the domain shift challenge (Pan & Yang, 2009), a model pretrained on a source domain often suffers from drastic performance degradation when directly applied on a target domain. This gives rise to the research attention of Unsupervised Domain Adaptation (UDA).

Existing UDA methods can be roughly divided into two categories. One is based on distribution alignment (Long et al., 2015; Kang et al., 2019; Ganin et al., 2016; Long et al., 2018) which minimizes domain discrepancy by aligning the distributions between two domains. They usually match two different distributions to a single distribution. Doing so could distort the original structural information, potentially hurting the final model generalization (Chen et al., 2019; Ge et al., 2022; Tang et al., 2020). Another is self-supervised learning (French et al., 2018; Liang et al., 2021a; Sun et al., 2019) which also learns *a common space* by using pseudo labels or other supervision information. The self-supervised learning faces the same limitation. Since the label noise is inevitable, it is difficult to obtain such a common feature space while keeping discriminative structure. As shown in Fig. 1(a), adapting a model in a common space cannot guarantee better classification performance.

To address the aforementioned problem, a natural strategy is to align the target and source data in two spaces. There are a limited works on this line. For example, CyCADA (Hoffman et al., 2018) uses two different networks to transform the images of the source domain to the target domain and vice versa. However, its learned two networks are not strictly inverse mappings, making the transformed images not necessarily semantically consistent through the transformation. As a result, data topological structure can not be well preserved. Homeomorphism is a concept from Topology (Munkres, 2000). If a bijection satisfies the definition of homeomorphism, i.e., one-to-one correspondence and

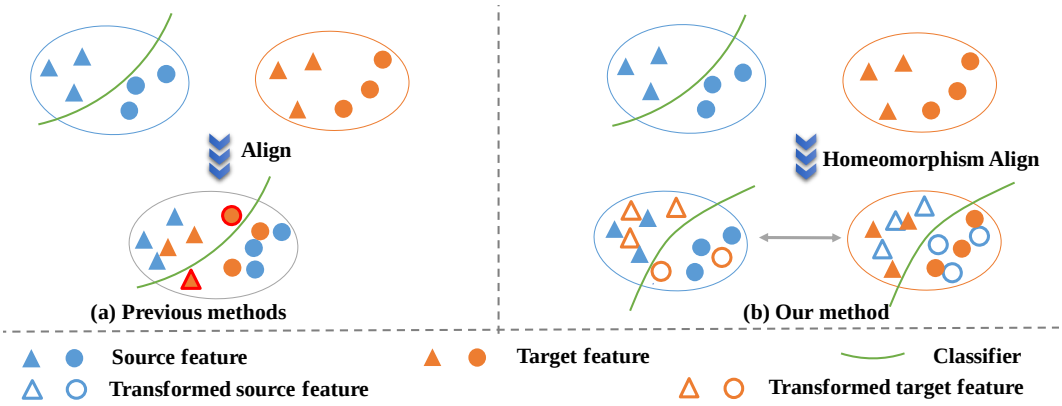


Figure 1: Comparison between previous unsupervised domain adaptation methods and our homeomorphism alignment. (a) Previous methods align the distributions between two domains in a common space, with data discriminative structure is not well preserved. (b) Our method uses a homeomorphism mapping to align the training data in the source and target feature spaces. Since the homeomorphism mapping can preserve data topological structure, the adapted model achieves better performance in both spaces.

continuous, theoretically, we prove that the data topological structure is well preserved in the projected space, *i.e.*, the samples in the same cluster is still in the same projected cluster. As shown in Fig.1(b), by homeomorphism mapping, the adapted model works well in both source and target feature spaces.

Based on the above analysis, we propose a novel unsupervised domain adaptation method, called ***HomeomorphisM Alignment*** (HMA). Our method consists of three parts. The first is the construction of homeomorphism mapping to connect the source and target feature spaces. Fortunately, the recently proposed Invertible Neural Network (INN) (Kingma & Dhariwal, 2018) can naturally help us find a pair of mutually invertible functions through the forward process and invertible process of the network, which also greatly saves the memory space. We therefore adopt INN for realizing a homeomorphism. Then, distribution matching method is used as a sewing up tool. At both ends of the homeomorphism mapping, we require that the transformed features be aligned with the features of the corresponding domain. Intuitively, if we can stitch them by category semantically, the homeomorphic mapping will be able to better implement the transformation between two spaces while keeping the topological structure. Subsequently, the model is further trained in the source and target feature spaces concurrently by the preserved topological structure; The optimization can be facilitated by the constraints that for the source domain the transformed samples share the same labels as the original ones, and for the target domain the original and transformed samples could share the same predictions.

Our *contributions* can be summarized as follows: (1) We theoretically prove that homeomorphism mapping can guarantee the topological structure of the mapped data; This is an important property yet ignored in the existing researches. We also show that INN implements a homeomorphism. (2) We propose a novel UDA method with homeomorphism, the first attempt to consider the UDA problem from the viewpoint of topology and conduct the domain alignment in two spaces. This is in contrast to the previous methods relying on learning a common space to align the source and target features. (3) Extensive experiments demonstrate the superiority over the existing state-of-the-art alternatives, along with in-depth ablation studies.

2 Related Work

Unsupervised Domain Adaptation. UDA aims to improve the generalization ability of a model on an unlabeled target domain by leveraging the labeled source domain. Existing methods can be roughly divided into two categories.

The first category adopts the idea of *distribution alignment* that trains the model by minimizing the source error and the discrepancy between source and target domains concurrently. There exist two main strategies: *statistic moment matching* and *adversarial learning*. The methods based on *statistic moment matching* minimize the statistic discrepancy to align the distributions between two domains. DAN (Long et al., 2015) proposes multiple kernels Maximum Mean Discrepancy (MK-MMD) for adapting marginal distribution between two domains. CORAL (Sun et al., 2017) minimizes the domain shift by aligning the second-order statistics of the source and target distributions. CAN (Kang et al., 2019) proposes Constrastive Domain Discrepancy (CDD) to minimize the intra-class discrepancy and maxmize the inter-class discrepancy, which aligns the conditional distributions between two domains. The methods based on *adversarial learning* are inspired by GAN (Goodfellow et al., 2014), which plays a minimax game between feature extractor and discriminator to learn domain invariant features. DANN (Ganin et al., 2016) directly uses the features of the source and target domains from the same feature extractor as the input of the discriminator for domain classification. ADDA (Tzeng et al., 2017) uses two different feature extractors for the source and target domains respectively and uses a discriminator to identify the domain labels of the features. Both of discriminators used in DANN and ADDA only focus on the domain information of features and ignore the category information, which only achieves the marginal distribution alignment. Therefore, CDAN (Long et al., 2018) takes the prediction of classifier and features as the input of the discriminator, which conduct conditional distribution alignment between two domains.

The second category of methods regards domain adaptation as a *self-supervised learning* problem. The key is to obtain more accurate pseudo-labels or supervision information to tune the source model. In fact, in the training process, such methods are also trying to find a common space to implicitly align the source and target features such that the source and target domain features projected by the feature extractor have better discriminability. For example, SE (French et al., 2018) uses the mean teacher framework with a student network and a teacher network. For the update of the student network, it uses the cross-entropy of the source samples and the consistency constraints of the target samples. While the teacher network is updated by exponential moving average of student network. ATDOC (Liang et al., 2021a) assigns a pseudo-label for each target sample by employing a memory mechanism. ssUDA (Sun et al., 2019) performs self-supervised tasks (*e.g.*, rotation, flip and patch location predictions) to improve the model generalization.

3 Analysis of Distribution Alignment

Problem statements. In the UDA setting, there is a source domain $D_s = \{(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s)\}_{i=1}^{n_s}$ consisting of n_s labeled samples and a target domain $D_t = \{(\boldsymbol{x}_i^t)\}_{i=1}^{n_t}$ consisting of n_t unlabeled samples. The source domain and target domain share the same label space $\{1, 2, \cdots, K\}$, but with different distributions. Suppose the source model Γ_s, pretrained by labeled source data, is composed of a feature extractor F and a classifier C. The goal of UDA is to adapt the source model such that it works well in the unlabeled target domain.

Ideally, given the ground-truth labels, the trained model can work well in both the source and target domains. To verify this, we conduct experiments on the Office-31 dataset. Based on the real labels in both source and target domains, supervised training is performed. 100% accuracy on the source and target domains can be achieved. Next, we will check the adapted model based on distribution alignment approaches. See Appendix for more algorithm details.

Can previous distribution alignment methods really achieve 100% accuracy given ground-truth labels? As mentioned earlier, current domain adaptation methods usually adapt the source domain model to the target domain through two approaches: explicit distribution alignment based on statistic moment matching or adversarial learning, and implicit alignment based self-supervised learning. Essentially, these alignment methods implement domain adaptation by projecting source and target domain samples into a common space, i.e., the domain invariant space. We conduct an experiment on the top-4 challenging adaptation tasks of Office-31 dataset. We use the ground-truth target domain labels to perform different kinds of distribution alignment to show what will happen. The results are shown in Table 1. The first row shows the results of CAN (Kang et al., 2019) which aligns feature distributions based on the real target labels. The second row in Table 1 shows the performance based on adversarial learning method CDAN (Long et al., 2018). From these two rows, it can be observed that the explicit distribution alignment methods cannot achieve 100% accuracy

Table 1: Up-bound performance probing: Comparing different distribution alignment strategies on *Office-31* using the ground-truth target sample labels.

Component	A→D	A→W	D→A	W→A
Statistic moment matching	99.9±0.1	99.9±0.0	92.6±0.2	93.8±0.1
Adversarial learning	99.2±0.1	99.8±0.1	90.9±0.2	92.3±0.1
Self-supervised learning	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**
Bijection alignment	97.8±0.2	98.9±0.1	90.7±0.1	93.2±0.2
Ours	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**

which means that the discriminative data structure is not well preserved when they align the source and target domain samples in a common space. The third row shows the results of self-supervised learning approach using the real target sample labels. It works very well which means the feature extractor can find the domain invariant space while keeping data discriminative structure. However, in practice the label noisy cannot be eliminated.

Can double mapping achieve 100% accuracy? Since it is difficult to find a feature extractor to obtain a common space while keeping data discriminative structure, naturally whether the source and target domain samples can be aligned in two feature spaces in a way of double mapping, so as to achieve accurate classification for target domain samples. The fourth row in Table 1 shows the performance that uses two different networks to map the source features to the target feature space and vice versa. We use the real target sample labels to train these two networks and align the conditional distributions between the transformed features and original features. This bijection composed of these two networks is *not* a homeomorphism. The data topological structure can not be preserved in the process of bidirectional projections between the source and target domains. Despite the access to all the real labels of the target domain, this method still cannot achieve 100% accuracy.

Homeomorphism alignment can achieve 100% accuracy. A homeomorphism, also known as a continuous transformation, is a one-to-one correspondence mapping between the points in two topological spaces that is continuous in both directions. For more details please refer to (Munkres, 2000). Let M and N be two topological spaces, and $g : M \to N$ be a bijection. If both the function g and its inverse function $g^{-1} : N \to M$ are continuous, then g is called a **homeomorphism**. That is to say, a homeomorphism is a bijective correspondence $g : M \to N$ such that $g(U)$ is open if and only if U is open. By the definition above, it is easy to say that g is a homeomorphism if and only if g^{-1} is a homeomorphism. Based on this definition, the following theorem can be easily derived.

Theorem 1. The set boundary corresponds to the set boundary by homeomorphism. More precisely, let (M, d_M) and (N, d_N) be two metric spaces where d_M and d_N are the metrics on M, N respectively. Suppose there is a homeomorphism $g : M \to N$, and A is an open subset in (M, d_M), we have that its image $B := g(A)$ is an open subset in (N, d_N), and

$$g(\partial A) = \partial B = \partial g(A).$$

where ∂ means the boundary.

As shown in Theorem 1, data topological structure is preserved by homeomorphism mapping, i.e., the samples in the same cluster are still in the same projected cluster. Fortunately, in the community of machine learning, there exists a network satisfying homeomorphism definition which is called Invertible Neural Network (INN) (Kingma & Dhariwal, 2018). It is easy to validate that INN satisfies the following theorem.

Theorem 2. Invertible Neural Network is a homeomorphism.

As shown in the last row of Table 1, we use INN to connect two domains at the feature level. With the real target sample labels, the homeomorphism alignment can achieve 100% accuracy thanks to the topological structure preserving property.

4 Method of Homeomorphism Alignment

Overview: Based on the analysis of distribution alignment, as shown in Fig. 2, the proposed HomeomorphisM Alignment (HMA) method consists of three parts. The first part is about homeomorphism mapping construction based on INN. The second part is sewing up which uses the homeomorphism

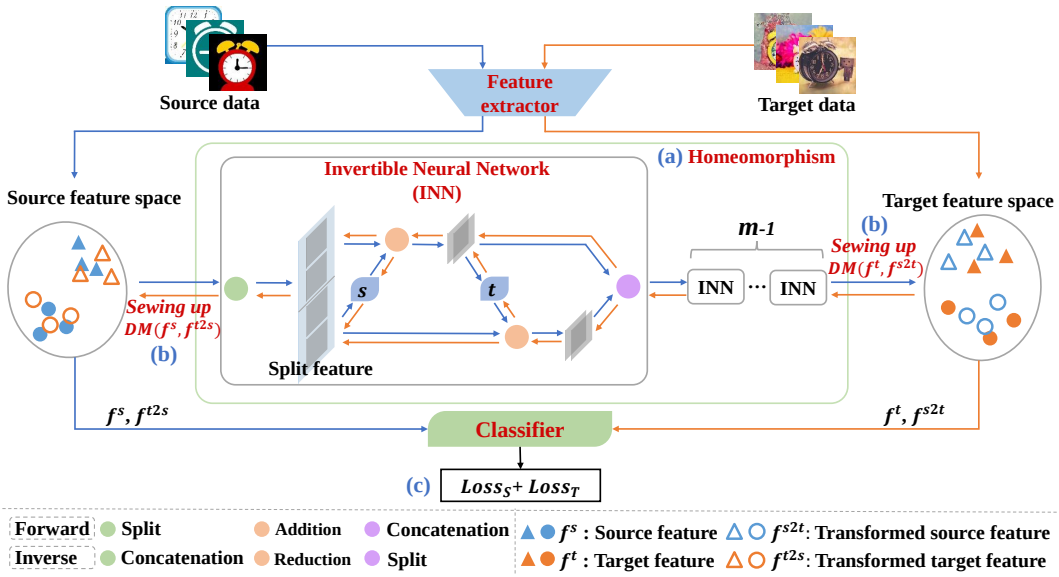


Figure 2: The framework of the proposed HomeomorphisM Alignment (HMA). (a) We cascade m invertible neural networks to implement a homeomorphism. (b) The transformed features are sewed up with the corresponding feature spaces by category. (c) The source model is iteratively trained in the two spaces concurrently.

mapping to connect the source and target feature spaces. The final part is retraining the pretrained source model in the source and target feature spaces by using the property of homeomorphism.

4.1 HOMEOMORPHISM IMPLEMENTED BY INVERTIBLE NEURAL NETWORK (INN)

In each iteration, we randomly sample a batch of source and target samples. We use ResNet (He et al., 2016) as the feature extractor F to map a sample x to the feature space: $\boldsymbol{f}^{s/t} = F(\boldsymbol{x}^{s/t})$ where s/t represents the source and target domain respectively. Due to the distribution discrepancy between the source and target domains, we consider that the source and target features reside on two different spaces (manifolds) respectively.

A homeomorphism g consists of m blocks of INN. m is a hyperparameter discussed in Appendix. In each block, we use an affine network to implement the INN (Dinh et al., 2017), as shown in Fig. 2. For the i-th block, we denote the input $\mu^i_{1:2d}$ with $2d$ dimension. In the *forward* process, we transform from the source feature space $\boldsymbol{f}^s$ to the target feature space $\boldsymbol{f}^t$. Specifically, we split evenly $\mu^i_{1:2d}$ to two parts $[\mu^i_{1:d}, \mu^i_{d+1:2d}]$, and further transform them with two respective linear neural networks $s(\cdot)$, $t(\cdot)$. The output of i-th block $\mu^{i+1}_{1:2d}$ is then obtained with residual as follows:

$$\mu^{i+1}_{1:d} = \mu^i_{1:d} + s(\mu^i_{d+1:2d}), \quad \mu^{i+1}_{d+1:2d} = \mu^i_{d+1:2d} + t(\mu^{i+1}_{1:d}), \quad \mu^{i+1}_{1:2d} = [\mu^{i+1}_{1:d}, \mu^{i+1}_{d+1:2d}]. \tag{1}$$

The output $\mu^{i+1}_{1:2d}$ will be set as the input of the next block.

In the *inverse projection* process, we map $\mu^{i+1}_{1:2d}$ to $\mu^i_{1:2d}$ in the opposite away around. According to equation 1, we can get the following equation:

$$\mu^i_{d+1:2d} = \mu^{i+1}_{d+1:2d} - t(\mu^{i+1}_{1:d}), \quad \mu^i_{1:d} = \mu^{i+1}_{1:d} - s(\mu^i_{d+1:2d}), \quad \mu^i_{1:2d} = [\mu^i_{1:d}, \mu^i_{d+1:2d}]. \tag{2}$$

We similarly split $\mu^{i+1}_{1:2d}$ into two parts $[\mu^{i+1}_{1:d}, \mu^{i+1}_{d+1:2d}]$ and follow equation 2 to get the original input $\mu^i_{1:2d}$. Obviously equation 1 and equation 2 are inverse functions of each other. We denote the forward process of the m INNs as the function g, while the inverse process as g^{-1}. Hence, the function g is a bijection. Since the functions $s(\cdot)$, $t(\cdot)$ are implemented by two linear connected neural networks, they are continuous; This means both g and g^{-1} are continuous, too. According to the definition of homeomorphism, this INN is a homeomorphism.

4.2 SEWING UP

Now we will sew up homeomorphism mapping g to the source and target feature spaces such that the corresponding classes are aligned. Suppose the transformed feature $\boldsymbol{f}^{s2t} = g(\boldsymbol{f}^s)$ according to the source feature $\boldsymbol{f}^s$ and the transformed feature $\boldsymbol{f}^{t2s} = g^{-1}(\boldsymbol{f}^t)$ according to target feature $\boldsymbol{f}^t$. According to the property of homeomorphism mapping g, $\boldsymbol{f}^s = g^{-1}(\boldsymbol{f}^{s2t})$ and $\boldsymbol{f}^t = g(\boldsymbol{f}^{t2s})$. To guarantee that the transformed features are the correct places, i.e., they are aligned with the corresponding classes, the distribution matching method is used for sewing up. The loss function is defined as follows,

$$\min_{g} \text{Loss}_{Sew} = DM(\boldsymbol{f}^{s2t}, \boldsymbol{f}^t) + DM(\boldsymbol{f}^{t2s}, \boldsymbol{f}^s), \tag{3}$$

where $DM(\cdot, \cdot)$ refer to the existing distribution matching methods, such as DAN (Long et al., 2015), CAN (Kang et al., 2019), DANN (Ganin et al., 2016) and CDAN (Long et al., 2018), etc.

It should be noted that only using marginal distribution matching method, such as DAN and DANN, cannot achieve satisfactory results, because these methods only focus on overall distribution alignment instead of class-wise alignment. Although our homeomorphism mapping g can ensure that the transformed features retain topological structure, if not stitched correctly according to the corresponding category, then the source domain and target domain can not achieve the discriminate feature lossless transformation. So class conditional distribution matching becomes a better choice. This is also confirmed in the experiment section.

4.3 MODEL TRAINING

A model often suffers performance degradation from the domain shift. To address this problem, we leverage the homeomorphic property. Specifically, after training the homeomorphism mapping, we perform distribution alignment between $\boldsymbol{f}^s$ and $\boldsymbol{f}^{t2s}$, and between $\boldsymbol{f}^t$ and $\boldsymbol{f}^{s2t}$. As proved by Theorem 1, $\boldsymbol{f}^s$ has the same structural information as $\boldsymbol{f}^{s2t}$. Concretely, for a specific labeled source sample x, the corresponding feature $\boldsymbol{f}^s$ and $\boldsymbol{f}^{s2t}$ share the same label. The following loss function is applied to the supervised training of feature extractor F and classifier C,

$$\min_{F,C} Loss_S = \mathcal{L}^{ce}(C(\boldsymbol{f}^s), \boldsymbol{y}^s) + \mathcal{L}^{ce}(C(\boldsymbol{f}^{s2t}), \boldsymbol{y}^s), \tag{4}$$

where $\boldsymbol{y}^s$ is the corresponding label of the source sample x^s, and $\mathcal{L}^{ce}(\cdot, \cdot)$ denotes the cross entropy function. In particular, the term $\mathcal{L}^{ce}(C(\boldsymbol{f}^s), \boldsymbol{y}^s)$ focuses on the classification of the source domain, whilst $\mathcal{L}^{ce}(C(\boldsymbol{f}^{s2t}), \boldsymbol{y}^s)$ is concerned with the classification of the target domain since $\boldsymbol{f}^{s2t}$ and $\boldsymbol{f}^t$ have been aligned.

Considering that our homeomorphism preserves the structure across the mapping and no label information in the target domain, unsupervised consistency constraint is a natural strategy for optimization. Formally, for an unlabeled target sample x^t, we formulate the consistency constraint on the predictions between $\boldsymbol{f}^t$ and $\boldsymbol{f}^{t2s}$ as:

$$\min_{F,C} Loss_T = L_C(C(\boldsymbol{f}^t), C(\boldsymbol{f}^{t2s})), \tag{5}$$

where $L_C(\cdot, \cdot)$ is a consistency constraint such as L_1-Norm and L_2-Norm. In practice, we found L_2-Norm suffices. By combining equation 4 and equation 5, the overall loss is defined as follows,

$$\min_{F,C} Loss_S + Loss_T. \tag{6}$$

Summary. At the training phase, in each iteration, we first train an INN based homeomorphism mapping, followed by model training in two spaces concurrently. At the inference phase, both the target features $\boldsymbol{f}^t$ and the transformed target features $\boldsymbol{f}^{t2s}$ can be used to make the prediction. Also, average based ensemble can be used to obtain the final prediction.

Remarks. Our model is trained in the source and target feature spaces concurrently. Compared with the existing alignment based UDA methods in a common space, this design naturally overcomes the intrinsic challenges of projecting the source and target domain samples into a single shared feature space using a feature extraction network while keeping their respective discriminative structures. When the distributions between the two feature spaces are not originally aligned typical in practice (*e.g.*, due to domain-specific characteristics such as different background, viewing conditions, etc.),

Table 2: Comparison with the state-of-the-art methods on *Office-31* dataset. Metric: classification accuracy (%); Backbone: ResNet-50.

Method	Venue	A→D	A→W	D→A	D→W	W→A	W→D	*avg*
ResNet-50	CVPR16	68.9	68.4	62.5	96.7	60.7	99.3	76.1
DAN	ICML15	78.6	80.5	63.6	97.1	62.8	99.6	80.4
CAN	CVPR19	95.0	94.5	78.0	99.1	77.0	99.8	90.6
TSA	CVPR21	92.6	94.8	74.9	99.1	74.4	**100.0**	89.3
DANN	JMLR16	79.7	82.0	68.2	96.9	67.4	99.1	82.2
CDAN	NIPS18	89.8	93.1	70.1	98.2	68.0	**100.0**	86.5
DADA	AAAI20	93.9	92.3	74.4	99.2	74.2	**100.0**	89.0
MDD+IA	ICML20	92.1	90.3	75.3	98.7	74.9	99.8	88.8
BCDM	AAAI21	93.8	95.4	73.1	98.6	73.0	**100.0**	89.0
ILA	CVPR21	93.4	**95.7**	72.1	**99.3**	75.4	**100.0**	89.3
MetaAlign	CVPR21	94.5	93.0	75.0	98.6	73.6	**100.0**	89.2
DWL	CVPR21	91.2	89.2	73.1	99.2	69.8	**100.0**	87.1
DALN	CVPR22	95.4	95.2	76.4	99.1	76.5	**100.0**	90.4
ALDA	AAAI20	94.0	95.6	72.2	97.7	72.5	**100.0**	88.7
ATDOC	CVPR21	94.4	94.5	75.6	98.9	75.2	99.6	89.7
CaCo	CVPR22	91.7	89.7	73.1	98.4	72.8	**100.0**	87.6
SUDA	CVPR22	91.2	90.8	72.2	98.7	71.4	**100.0**	87.4
HMA(DANN)	Ours	83.9±0.1	83.5±0.2	70.5±0.1	98.2±0.1	70.1±0.2	**100.0±0.0**	84.4
HMA(CDAN)	Ours	92.4±0.2	95.1±0.2	73.7±0.1	99.2±0.1	72.8±0.2	**100.0±0.0**	88.9
HMA(DAN)	Ours	85.1±0.2	84.5±0.2	67.9±0.3	98.9±0.2	66.7±0.3	**100.0±0.0**	83.9
HMA(CAN)	Ours	**95.8±0.3**	95.1±0.1	**79.3±0.3**	**99.3±0.1**	**77.6±0.2**	**100.0±0.0**	**91.2**

Table 3: Comparisons with the state-of-the-art methods on *Office-Home* dataset. Metric: classification accuracy (%); Backbone: ResNet-50.

Method	Venue	A→C	A→P	A→R	C→A	C→P	C→R	P→A	P→C	P→R	R→A	R→C	R→P	*avg*
ResNet-50	CVPR16	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	ICML15	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
CAN	CVPR19	58.7	78.1	82.1	67.4	75.7	78.1	67.2	54.2	82.5	73.4	60.9	83.5	71.8
TSA	CVPR21	53.6	75.1	78.3	64.4	73.7	72.5	62.3	49.4	77.5	72.2	58.8	82.1	68.3
DANN	JMLR16	45.6	59.3	70.1	47.0	58.5	60.9	46.1	43.7	68.5	63.2	51.8	76.8	57.6
CDAN	NIPS18	49.0	69.3	74.5	54.4	66.0	68.4	55.6	48.3	75.9	68.4	55.4	80.5	63.8
MDD+IA	ICML20	56.2	77.9	79.2	64.4	73.1	74.4	64.2	54.2	79.9	71.2	58.1	83.1	69.5
MetaAlign	CVPR21	59.3	76.0	80.2	65.7	74.7	75.1	65.7	**56.5**	81.6	74.1	61.1	85.2	71.3
DALN	CVPR22	57.8	**79.9**	82.0	66.3	76.2	77.2	66.7	55.5	81.3	73.5	60.4	85.2	71.8
ALDA	AAAI20	53.7	70.1	76.4	60.2	72.6	71.5	56.8	51.9	77.1	70.2	56.3	82.1	66.6
ATDOC	CVPR21	58.3	78.8	82.3	**69.4**	**78.2**	78.2	67.1	56.0	82.7	72.0	58.2	**85.5**	72.2
HMA(DANN)	Ours	48.2	65.1	75.4	57.0	65.0	68.3	55.6	45.2	73.5	66.6	54.3	78.4	62.7
HMA(CDAN)	Ours	58.7	78.1	81.6	67.4	75.8	78.1	66.8	54.2	82.5	73.4	59.7	83.5	71.7
HMA(DAN)	Ours	46.2	63.5	73.9	58.1	65.3	68.3	55.3	43.9	74.8	67.2	53.4	78.4	62.4
HMA(CAN)	Ours	**60.6**	79.1	**82.9**	68.9	77.5	**79.3**	**69.1**	55.9	**83.5**	**74.6**	**62.3**	84.4	**73.2**

the homeomorphism provides a flexible non-invasive means for cross-domain relating via transforming their individual features from each other *externally*. Critically, this alignment in two spaces allows to fully keep the original per-domain characteristics including some discriminative information. Compared with the self-supervised learning methods suffering the noises of pseudo-labeling, our transformed source features in the target domain can directly use the ground-truth source labels, in addition to additionally exploiting the topological structure of the source domain. Further, our consistency constraint can exploit the unlabeled target training data (*i.e.*, the original and transformed target features with shared topological structure) in an unsupervised manner, without the notorious pseudo-label noise issue.

5 Experiments

5.1 Experimental Setup

Datasets: In our experiments, three standard datasets are used. ***Office-31*** (Saenko et al., 2010) is a popular benchmark. It contains a total of 4110 images of 31 office environment objects from

Table 4: Comparison with the state-of-the-art methods on *Visda-17* dataset. Metric: per-class classification accuracy (%); Backbone: ResNet-101.

Method	Venue	plane	bcycl	bus	car	horse	knife	mcycl	person	plant	sktbrd	train	truck	*avg*
ResNet-101	CVPR16	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
DAN	ICML15	84.8	42.1	75.4	53.0	77.9	62.6	86.6	50.7	59.7	52.9	82.5	26.0	62.9
CAN	CVPR19	97.0	87.2	82.5	74.3	97.8	96.2	90.8	80.7	96.6	96.3	87.5	59.9	87.2
TSA	CVPR21	-	-	-	-	-	-	-	-	-	-	-	-	78.6
DANN	JMLR16	81.9	77.7	82.8	44.3	81.2	29.5	65.1	28.6	51.9	54.6	82.8	7.8	57.4
CDAN	NIPS18	85.2	66.9	83.0	50.8	84.2	74.9	88.1	74.5	83.4	76.0	81.9	38.0	73.9
BCDM	AAAI21	95.1	87.6	81.2	73.2	92.7	95.4	86.9	82.5	95.1	84.8	88.1	39.5	83.4
DWL	CVPR21	90.7	80.2	**86.1**	67.6	92.4	81.5	86.8	78.0	90.6	57.1	85.6	28.7	77.1
CLS	ICCV21	92.6	84.5	73.7	72.7	88.5	83.3	89.1	77.6	89.5	89.2	85.8	**72.7**	81.6
DALN	CVPR22	-	-	-	-	-	-	-	-	-	-	-	-	80.6
ALDA	AAAI20	93.8	74.1	82.4	69.4	90.6	87.2	89.0	67.6	93.4	76.1	87.7	22.2	77.8
ATDOC	CVPR21	93.7	83.0	76.9	58.7	89.7	95.1	84.4	71.4	89.4	80.0	86.7	55.1	80.3
CaCo	CVPR22	90.4	80.7	78.8	57.0	88.9	87.0	81.3	79.4	88.7	88.1	86.8	63.9	80.9
SUDA	CVPR22	88.3	79.3	66.2	64.7	87.4	80.1	85.9	78.3	86.3	87.5	78.8	74.5	79.8
HMA(DANN)	Ours	86.9	79.1	83.5	50.5	86.7	47.3	86.1	55.1	64.6	59.8	84.6	36.2	68.4
HMA(CDAN)	Ours	88.3	71.2	85.1	66.4	86.3	79.3	88.8	**87.6**	83.9	79.3	83.4	46.2	78.8
HMA(DAN)	Ours	87.5	49.2	80.2	53.8	81.8	71.8	87.8	57.6	60.9	57.0	85.3	32.8	67.1
HMA(CAN)	Ours	**97.6**	**88.4**	84.3	**76.0**	**98.4**	**97.1**	**91.3**	81.4	**97.0**	**96.7**	**88.8**	60.7	**88.1**

3 domains: Amazon (A), Webcam (W), Dslr(D). ***Office-Home*** (Venkateswara et al., 2017) is a more challenging dataset which contains 15588 images within 65 classes from 4 domains: Artistic images (A), Clip-Art images (C), Product images (P) and RealWorld images (R). ***Visda-17*** (Peng et al., 2017) is a widely used benchmark for domain adaptation with focus on a 12-class synthesis-to-real object classification task. The source domain contains 152,397 synthetic images and the target domain has 55,388 real object images.

Implementation details: Our experiment is performed in Pytorch. Each task is run 5 times to enhance the robustness of the results. The same backbone network is selected as other compared methods for fair comparison. Specifically, Resnet-50 is selected as the backbone on *Office-31* and *Office-Home*, and Resnet-101 is selected on *Visda-17*. It is worth noting that the output dimension of the classifier in the original backbone is replaced by the number of categories to fit each task. The SGD optimizer is chosen to update the network and the CosineAnnealingLR (Loshchilov & Hutter, 2016) is used to update the learning rate of the SGD optimizer.

Competitors: To verify the effectiveness of our method, we compare it with the following three types of state-of-the-art methods. The first type based on distribution alignment, such as statistic moment matching methods DAN (Long et al., 2015), CAN (Kang et al., 2019), and TSA (Li et al., 2021c). The second type based on adversarial learning including DANN (Ganin et al., 2016), CDAN (Long et al., 2018), MDD+IA (Jiang et al., 2020), DADA (Tang & Jia, 2020), BCDM (Li et al., 2021a), CLS (Liu et al., 2021), ILA (Sharma et al., 2021), MetaAlign (Wei et al., 2021), DWL (Xiao & Zhang, 2021), and DALN (Chen et al., 2022). The third type based on self-supervised learning: ALDA (Chen et al., 2020), ATDOC (Liang et al., 2021a), CaCo (Huang et al., 2022), and SUDA (Zhang et al., 2022).

5.2 Comparisons to State-of-the-Art

The performance comparison with other state-of-the-art methods on Office-31, Office-home and Visda-17 are shown in Table 2, Table 3 and Table 4 respectively. The methods HMA(DANN) and HMA(CDAN) mean the sewing up tool is the distribution matching method based on adversarial learning where DANN and CDAN focus on marginal distribution alignment and conditional distribution alignment respectively. While HMA(DAN) and HMA(CAN) apply the statistic moment matching methods as the sewing up tool, where DAN and CAN focus on marginal distribution alignment and conditional distribution alignment respectively.

It can be observed that HMA(CAN) yields the best average performance on both three datasets. This also confirms our previous analysis. Different sewing up methods will affect the final performance. In general, conditional distribution alignment is better than marginal distribution alignment on the two kinds of methods based on adversarial learning strategy and statistic moment matching strategy because conditional distribution alignment can stitch homeomorphism mapping with two spaces by

Table 5: Homeomorphism mapping vs double mapping on *Office-31*. DoubleMAP(CAN) means double mapping sewed by the distribution matching method CAN.

Component	A→D	A→W	D→A	W→A	Parameters
HMA(CAN)	**95.8±0.3**	**95.1±0.1**	**79.3±0.3**	**77.6±0.2**	20992000
DoubleMAP(CAN)	90.3±0.3	91.7±0.2	76.6±0.1	75.8±0.2	33603584

Table 6: Ablation study on *Office-31*.

Component	A→D	A→W	D→A	W→A
CAN	95.0±0.3	94.5±0.3	78.0±0.3	77.0±0.3
INN(CAN)	94.8±0.3	94.1±0.2	77.3±0.4	76.7±0.2
INN(CAN)+S2T	95.6±0.2	94.9±0.3	78.9±0.2	77.4±0.2
HMA(CAN)	**95.8±0.3**	**95.1±0.1**	**79.3±0.3**	**77.6±0.2**

category. Furthermore, for condition distribution matching method, statistic moment matching strategy is better than adversarial learning strategy. The reason is that statistic moment matching strategy stitches homeomorphism mapping with two feature spaces explicitly by category. Interestingly, the alignment method CAN published in 2019 still achieves SOTA results. It can be seen that alignment by category is very important for extracting domain invariant features. Our method HMA(CAN) further boosts the CAN performance, because we realize the difficulty of extracting a domain invariant space and we do alignment in the two spaces by homeomorphism mapping.

5.3 Ablation Analysis and Discussion

Homeomorphism map is better than double mapping. As mentioned in Section 3, there does not exist many methods which do alignments in two feature spaces based on bijection. In this experiment, we will apply the double mapping method in Section 3 to top-4 challenging tasks on Office-31 dataset, the difference between INN and double mapping method is mainly reflected in topological structure maintenance. The results are shown in Table 5. It is obvious that homeomorphism mapping is superior to the normal double mapping method in accuracy and model parameters. For the model size, the INN based homeomorphism mapping uses almost half of the parameters compared to using two neural networks.

Ablation study. To show the effectiveness of alignment in two spaces, we conduct an experiment on top-4 challenging tasks of Office-31 dataset. The results are shown in Table 6. The method CAN is considered as the baseline. INN(CAN) means just sewing up homeomorphism mapping to the two feature spaces and the source model is retrained based on the source labels. INN(CAN)+S2T means the transformed source features are used to learn the model in the target feature space compared with INN(CAN). HMA(CAN) uses all features in two spaces. From Table 6, we can find that simply using INN can achieve similar performance as the feature distribution alignment method CAN. As shown in the third row in Table 6, by transferring the source features to the target feature space, the performance is greatly improved. For this case, the data topological structure and label information from source domain can be correctly transformed to the target domain, which allows that the learned model works well in the target domain. The last row in Table 6 shows that the performance can be further improved if both transformed features are used. Because our homeomorphism mapping keeps the corresponding relationship by category, with the help of supervision information in the source feature spaces, the generalization performance of the model in the two domains is improved.

6 Conclusion

In this paper, we have proposed a new unsupervised domain adaptation method, termed as *HomeomorphisM Alignment in two spaces* (HMA). By analyzing previous alignment based methods, we argue that it is difficult to find a common space or domain invariant space to adapt the pretrained source model. So the alignment is performed in two spaces. The extracted source and target features can be further transformed respectively by a homeomorphism mapping so that they can be aligned semantically. Our method consists of three steps, i.e., constructing an INN based homeomorphism mapping, sewing up by category and retraining iteratively training the model in two spaces. In this way, the source labels can be fully used even in the target feature space for improving the model generalization for the target domain. Extensive experimental results demonstrate the effectiveness of our method.

REFERENCES

Jingyi Cao, Bo Liu, Yunqian Wen, Rong Xie, and Li Song. Personalized and invertible face de-identification by disentangled identity information manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3334–3342, October 2021.

Haibo Chen, Lei Zhao, Huiming Zhang, Zhizhong Wang, Zhiwen Zuo, Ailin Li, Wei Xing, and Dongming Lu. Diverse image style transfer via invertible cross-space mapping. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14880–14889, October 2021.

Lin Chen, Huaian Chen, Zhixiang Wei, Xin Jin, Xiao Tan, Yi Jin, and Enhong Chen. Reusing the task-specific classifier as a discriminator: Discriminator-free adversarial domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7181–7190, 2022.

Minghao Chen, Shuai Zhao, Haifeng Liu, and Deng Cai. Adversarial-learned loss for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3521–3528, 2020.

Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *International conference on machine learning*, pp. 1081–1090. PMLR, 2019.

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*, pp. 3941–3950, 2020.

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)*, pp. 447–463, 2018.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017.

Zhekai Du, Jingjing Li, Hongzu Su, Lei Zhu, and Ke Lu. Cross-domain gradient discrepancy minimization for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3937–3946, 2021.

Geoff French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *International Conference on Learning Representations*, 2018.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

Chunjiang Ge, Rui Huang, Mixue Xie, Zihang Lai, Shiji Song, Shuang Li, and Gao Huang. Domain adaptation via prompt learning. *arXiv preprint arXiv:2202.06687*, 2022.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, 2014.

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

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International conference on machine learning*, pp. 1989–1998. Pmlr, 2018.

Jiaxing Huang, Dayan Guan, Aoran Xiao, Shijian Lu, and Ling Shao. Category contrast for unsupervised domain adaptation in visual tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1203–1214, 2022.

Xiang Jiang, Qicheng Lao, Stan Matwin, and Mohammad Havaei. Implicit class-conditioned domain alignment for unsupervised domain adaptation. In *International Conference on Machine Learning*, pp. 4816–4827. PMLR, 2020.

Junpeng Jing, Xin Deng, Mai Xu, Jianyi Wang, and Zhenyu Guan. Hinet: Deep image hiding by invertible network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4733–4742, October 2021.

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*, pp. 4893–4902, 2019.

Younggeun Kim and Donghee Son. Noise conditional flow model for learning the super-resolution space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 424–432, 2021.

Diederik P Kingma and Prafulla Dhariwal. Glow: generative flow with invertible 1×1 convolutions. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 10236–10245, 2018.

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*, pp. 10285–10295, 2019.

Mengxue Li, Yi-Ming Zhai, You-Wei Luo, Peng-Fei Ge, and Chuan-Xian Ren. Enhanced transport distance for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13936–13944, 2020.

Shuang Li, Fangrui Lv, Binhui Xie, Chi Harold Liu, Jian Liang, and Chen Qin. Bi-classifier determinacy maximization for unsupervised domain adaptation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*, 2021a.

Shuang Li, Fangrui Lv, Binhui Xie, Chi Harold Liu, Jian Liang, and Chen Qin. Bi-classifier determinacy maximization for unsupervised domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8455–8464, 2021b.

Shuang Li, Mixue Xie, Kaixiong Gong, Chi Harold Liu, Yulin Wang, and Wei Li. Transferable semantic augmentation for domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11516–11525, 2021c.

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*, pp. 16632–16642, 2021a.

Jingyun Liang, Andreas Lugmayr, Kai Zhang, Martin Danelljan, Luc Van Gool, and Radu Timofte. Hierarchical conditional flow: A unified framework for image super-resolution and image rescaling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4076–4085, 2021b.

Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 469–477, 2016.

Xiaofeng Liu, Zhenhua Guo, Site Li, Fangxu Xing, Jane You, C.-C. Jay Kuo, Georges El Fakhri, and Jonghye Woo. Adversarial unsupervised domain adaptation with conditional and label shift: Infer, align and iterate. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10367–10376, October 2021.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pp. 97–105. PMLR, 2015.

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

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Zhihe Lu, Yongxin Yang, Xiatian Zhu, Cong Liu, Yi-Zhe Song, and Tao Xiang. Stochastic classifiers for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9111–9120, 2020.

James R. Munkres. Topology (2nd edition). In *Prentice Hall*, 2000.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, pp. 1345–1359, 2009.

Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.visda06924*, 2017.

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*, pp. 1406–1415, 2019.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. European conference on computer vision, pp. 213–226. Springer, 2010.

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*, pp. 3723–3732, 2018.

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*, pp. 618–626, 2017.

Astuti Sharma, Tarun Kalluri, and Manmohan Chandraker. Instance level affinity-based transfer for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5361–5371, 2021.

Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2107–2116, 2017.

Baochen Sun, Jiashi Feng, and Kate Saenko. Correlation alignment for unsupervised domain adaptation. In *Domain Adaptation in Computer Vision Applications*, pp. 153–171. Springer, 2017.

Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019.

Hui Tang and Kui Jia. Discriminative adversarial domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5940–5947, 2020.

Hui Tang, Ke Chen, and Kui Jia. Unsupervised domain adaptation via structurally regularized deep clustering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8725–8735, 2020.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7167–7176, 2017.

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

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5018–5027, 2017.

Guoqiang Wei, Cuiling Lan, Wenjun Zeng, and Zhibo Chen. Metaalign: Coordinating domain alignment and classification for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16643–16653, 2021.

Ni Xiao and Lei Zhang. Dynamic weighted learning for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15242–15251, 2021.

Renjun Xu, Pelen Liu, Liyan Wang, Chao Chen, and Jindong Wang. Reliable weighted optimal transport for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4394–4403, 2020.

Jingyi Zhang, Jiaxing Huang, Zichen Tian, and Shijian Lu. Spectral unsupervised domain adaptation for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9829–9840, 2022.

Shifeng Zhang, Ning Kang, Tom Ryder, and Zhenguo Li. iflow: Numerically invertible flows for efficient lossless compression via a uniform coder. *Advances in Neural Information Processing Systems*, 34:5822–5833, 2021a.

Shifeng Zhang, Chen Zhang, Ning Kang, and Zhenguo Li. ivpf: Numerical invertible volume preserving flow for efficient lossless compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 620–629, 2021b.

Lihua Zhou, Mao Ye, Xiatian Zhu, Shuaifeng Li, and Yiguang Liu. Class discriminative adversarial learning for unsupervised domain adaptation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pp. 4318–4326, 2022.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

A APPENDIX

A.1 MORE RELATED WORKS

Unsupervised Domain Adaptation. In addition to *statistic moment matching* and *adversarial learning* as mentioned in the paper, the methods based on *distribution alignment* also benefit from the following strategies: *adversarial generation framework*, *bi-classifier adversarial learning* and *optimal transport*. Specifically, with the *adversarial generation framework*, existing methods (Liu & Tuzel, 2016; Shrivastava et al., 2017; Zhu et al., 2017)) often combine the domain discriminator and a generator, and generate fake data to align the distributions across domains at the pixel level.

Based on *bi-classifier adversarial learning*, prior methods play a minimax game with a single feature extractor and two distinct classifiers during domain adaptation (Saito et al., 2018; Lee et al., 2019; Li et al., 2021b; Lu et al., 2020; Zhou et al., 2022). Commonly, they maximize the prediction discrepancy when training the classifiers and minimize the prediction discrepancy when training the feature extractor. Specifically, MCD (Saito et al., 2018) uses L_1-Norm to calculate the prediction discrepancy. SWD (Lee et al., 2019) proposes the slide wasserstein distance. BCDM (Li et al., 2021b) proposes the classifier determinacy disparity distance. CDAL (Zhou et al., 2022) proposes an expertise-aware classifier interference strategy to solve the ambiguous samples during domain adaptation. STAR (Lu et al., 2020) integrates an approximately infinite number of classifiers by sampling from a distribution, whilst keeping the model size the same as those with two classifiers.

With the assist of *optimal transport*, previous methods instead learn a transformation between two domains (Damodaran et al., 2018; Xu et al., 2020; Li et al., 2020). Their pipelines generally consist of two steps: the first step is to find a coupling matrix for connecting each source sample and target sample; The second step is to minimize the cost of these pair-wise connections. Specifically, DeepJDOT (Damodaran et al., 2018) minimizes the discrepancy of the features and predictions simultaneously using the Wasserstein distance. RWOT (Xu et al., 2020) exploits spatial prototypical information and intra-domain structure in a precise-pair-wise optimal transport procedure. ETD (Li et al., 2020) builds an attention-aware transport distance, which can be viewed as the prediction feedback of the iteratively learned classifier, to measure the domain discrepancy.

The previous optimal transport based methods are mostly similar to our model in the sense of finding a transformation for cross-domain distribution alignment. However, there are several key conceptual differences. *First*, they use a common space for distribution alignment. Instead, our method keeps per-domain distributions in two separate spaces while preserving their original structures. *Second*, they assume rigidly one-to-one (pairwise) mapping across domains which is not necessarily valid in practice. Favorably, our model does not make such strong assumptions by considering more relaxed coarse class-wise alignment between two distributions during the sewing up process. *Third*, they exploit the optimal transport to compute the transformation by solving a linear programming problem. In contrast, we construct a homeomorphism mapping that could be learned end-to-end more flexibly and scalably (*e.g.*, by using invertible neural networks).

Invertible Neural Network (INN). INN is a flow-based model, which transforms a probability distribution into another distribution by a sequence of invertible and differentiable mappings. It has been applied in image super-resolution, lossless compression, style transfer, privacy protection and so on. For example, HCFlow (Liang et al., 2021b) utilizes the hierarchical conditional flow as a unified framework for image super-resolution and image rescaling. NCSR (Kim & Son, 2021) proposes noise conditional flow model for super-resolution, which increases the visual quality and diversity of images through noise conditional layer. In lossless compression, derived from general volume preserving flows, iVPF (Zhang et al., 2021b) achieves an exact bijective mapping without any numerical error and then proposes a lossless compression algorithm. iFlow (Zhang et al., 2021a) achieves efficient lossless compression by a modular scale transform combining numerically invertible flow transformations. DIST (Chen et al., 2021) designs a diverse image style transfer framework by enforcing an invertible cross-space mapping. Additionally, invertible networks play an important role in protecting privacy, such as invertible de-identification and image hiding (Jing et al., 2021; Cao et al., 2021). Although have been widely and effectively used in image processing, no work has applied INN in domain adaptation.

A.2 PROOF OF THEOREM 1

Theorem 1. Let (M, d_M) and (N, d_N) be two metric spaces with a homeomorphism

$$g : M \to N,$$

and A is an open subset in (M, d_M), we have that its image $B := g(A)$ is an open subset in (N, d_N), and

$$g(\partial A) = \partial B = \partial g(A),$$

where ∂ means the boundary.

Proof. It is sufficient to show that

$$g(\partial A) \subset \partial B. \tag{7}$$

If this is true, then we can implied equation 7 to g^{-1}, and obtain

$$g^{-1}(\partial B) \subset \partial A.$$

Hence we have $\partial B \subset g(\partial A)$. Combing this with equation 7, we have $g(\partial A) = \partial B$.

Now we want to show equation 7, that is, for any $x \in \partial A$, we have $g(x) \in \partial B$. Since $x \in \partial A$, but $x \notin A$, then $g(x) \notin B$, and there is a sequence $\{x_i\} \subset A$ such that $\lim_{i\to\infty} x_i = x$. By the continuity of the function g, we have

$$g(x) = g(\lim_{i\to\infty} x_i) = \lim_{i\to\infty} g(x_i).$$

Noting that $g(x_i) \in B$, we get $g(x) \in \partial B$. This completes the proof. □

A.3 PROOF OF THEOREM 2

Theorem 2. Invertible Neural Network is a homeomorphism.

Proof. The definition of homeomorphism is that a function $g : M \to N$ between two topological spaces is a homeomorphism if it has the following properties: 1. g is a bijection; 2. g is continuous; 3. the inverse function g^{-1} is continuous.

For any invertible neural network, assuming that its forward process is g, then its invertible process can be represent g^{-1}, so g is a bijection. Because the function of each part of the invertible neural network is continuous, such as $s(\cdot)$ and $t(\cdot)$ in our method, so both g and g^{-1} are continuous. To sum up, the invertible network is a homeomorphism. □

A.4 ALGORITHM

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 f^s and
        target features f^t;
        Generate transformed source features f^{t2s} and transformed target features f^{s2t} by INN;
        Select a domain adaptation method and train INN based on equation 3;
        Train the backbone network based on equation 6;
    end for
end for
return Adapted model.
```

Our method is summarized in Algorithm 1. In each iteration, the INN and backbone network, which consists of feature extractor and classifier, are both trained. The loss functions are shown in equation 3 and equation 6 respectively.

A.5 IMPLEMENTATION DETAILS OF METHODS IN TABLE 1

In Table 1, we report the results of several classical domain adaptation strategies given ground-truth labels. Since the real labels are used, so we need to make simple modifications to these algorithms, which are shown below.

For the first line in Table 1, CAN (Kang et al., 2019) is selected to test the statistic moment matching strategy, which is almost the best statistical moment matching method in recent years. It uses the clustering algorithm to pseudo-label the 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, which is shown as follows:

$$\min_{F} Loss_{ALIGN}^{CAN} = \sum_{c=1}^{C} \mathcal{MMD}(\boldsymbol{f}^{s,c}, \boldsymbol{f}^{t,\hat{c}}) - \sum_{c_1=1}^{C} \sum_{c_2 \neq c_1}^{C} \mathcal{MMD}(\boldsymbol{f}^{s,c_1}, \boldsymbol{f}^{t,\hat{c_2}}), \tag{8}$$

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 pesudo label c. When we giving the ground-truth labels to target domain, we do not need to pseudo label target samples, and directly sample all target samples to perform distribution alignment as follows:

$$\min_{F} Loss_{ALIGN}^{CANours} = \sum_{c=1}^{C} \mathcal{MMD}(\boldsymbol{f}^{s,c}, \boldsymbol{f}^{t,c}) - \sum_{c_1=1}^{C} \sum_{c_2 \neq c_1}^{C} \mathcal{MMD}(\boldsymbol{f}^{s,c_1}, \boldsymbol{f}^{t,c_2}), \tag{9}$$

where $\boldsymbol{f}^{t,c}$ represents the target features with true label c. In this case, the feature extractor is retrained by equation 9; the source classifier is retrained by source samples.

For the second line in Table 1, CDAN (Long et al., 2018) is selected to test adversarial learning strategy. CDAN thinks the prediction of the classifier carry the discriminative information which can be used to align 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:

$$\min_{F} \max_{D} Loss_{ALIGN}^{CDAN} = \mathbb{E}_{\boldsymbol{x}_i^s \sim D_s} \log[D(\boldsymbol{f}_i^s \otimes \boldsymbol{p}_i^s)] + \mathbb{E}_{\boldsymbol{x}_i^t \sim D_t} \log[1 - D(\boldsymbol{f}_i^t \otimes \boldsymbol{p}_i^t)], \tag{10}$$

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

$$\min_{F} \max_{D} Loss_{ALIGN}^{CDANours} = \mathbb{E}_{\boldsymbol{x}_i^s \sim D_s} \log[D(\boldsymbol{f}_i^s \otimes \boldsymbol{l}_i^s)] + \mathbb{E}_{\boldsymbol{x}_i^t \sim D_t} \log[1 - D(\boldsymbol{f}_i^t \otimes \boldsymbol{l}_i^t)]. \tag{11}$$

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

For the third line in Table 1, it reports the self-supervised training strategy. Tradition methods based on this strategy (Liang et al., 2021a) usually assign target sample a pseudo-label $\hat{\boldsymbol{y}}_i^t$, and use pseudo-label to train the model as follows:

$$\min_{F,C} Loss^{SELF} = \mathbb{E}_{\boldsymbol{x}_i^s \sim D_s} \mathcal{L}^{ce}(\boldsymbol{p}_i^s, \boldsymbol{y}_i^s) + \mathbb{E}_{\boldsymbol{x}_i^t \sim D_t} \mathcal{L}^{ce}(\boldsymbol{p}_i^t, \hat{\boldsymbol{y}}_i^t), \tag{12}$$

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

$$\min_{F,C} Loss^{SELFours} = \mathbb{E}_{\boldsymbol{x}_i^s \sim D_s} \mathcal{L}^{ce}(\boldsymbol{p}_i^s, \boldsymbol{y}_i^s) + \mathbb{E}_{\boldsymbol{x}_i^t \sim D_t} \mathcal{L}^{ce}(\boldsymbol{p}_i^t, \boldsymbol{y}_i^t), \tag{13}$$

In this case, the feature extractor and source classifier are retrained by equation 13.

For the fourth line in Table 1, which 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

Table 7: 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.
clp	-	14.2	29.6	9.5	43.8	34.3	26.3
inf	21.8	-	23.2	2.3	40.6	20.8	21.7
pnt	24.1	15.0	-	4.6	45.0	29.0	23.5
qdr	12.2	1.5	4.9	-	5.6	5.7	6.0
rel	32.1	17.0	36.7	3.6	-	26.2	23.1
skt	30.4	11.3	27.8	3.4	32.9	-	21.2
Avg.	24.1	11.8	24.4	4.7	33.6	23.2	20.3

MCD	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	15.4	25.5	3.3	44.6	31.2	24.0
inf	24.1	-	24.0	1.6	35.2	19.7	20.9
pnt	31.1	14.8	-	1.7	48.1	22.8	23.7
qdr	8.5	2.1	4.6	-	7.9	7.1	6.0
rel	39.4	17.8	41.2	1.5	-	25.2	25.0
skt	37.3	12.6	27.2	4.1	34.5	-	23.1
Avg.	28.1	12.5	24.5	2.4	34.1	21.2	20.5

BNM	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	12.1	33.1	6.2	50.8	40.2	28.5
inf	26.6	-	28.5	2.4	38.5	18.1	22.8
pnt	39.9	12.2	-	3.4	54.5	36.2	29.2
qdr	17.8	1.0	3.6	-	9.2	8.3	8.0
rel	48.6	13.2	49.7	3.6	-	33.9	29.8
skt	54.9	12.8	42.3	5.4	51.3	-	33.3
Avg.	37.6	10.3	31.4	4.2	40.9	27.3	25.3

SWD	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	14.7	31.9	10.1	45.3	36.5	27.7
inf	22.9	-	24.2	2.5	33.2	21.3	20.0
pnt	33.6	15.3	-	4.4	46.1	30.7	26.0
qdr	15.5	2.2	6.4	-	11.1	10.2	9.1
rel	41.2	18.1	44.2	4.6	-	31.6	27.9
skt	44.2	15.2	37.3	10.3	44.7	-	30.3
Avg.	31.5	13.1	28.8	6.4	36.1	26.1	23.6

CGDM	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	16.9	35.3	10.8	53.5	36.9	30.7
inf	27.8	-	28.2	4.4	48.2	22.5	26.2
pnt	37.7	14.5	-	4.6	59.4	33.5	30.0
qdr	14.9	1.5	6.2	-	10.9	10.2	8.7
rel	49.4	20.8	47.2	4.8	-	38.2	32.0
skt	50.1	16.5	43.7	11.1	55.6	-	35.4
Avg.	36.0	14.0	32.1	7.1	45.5	28.3	27.2

HMA(CAN)	clp	inf	pnt	qdr	rel	skt	Avg.
clp	-	18.9	43.4	9.9	54.7	45.4	34.5
inf	35.9	-	37.2	5.7	54.5	30.8	32.8
pnt	42.6	14.9	-	10.8	61.4	35.1	33.0
qdr	31.0	5.8	15.0	-	15.9	16.2	16.8
rel	53.1	18.8	47.0	4.1	-	43.0	33.2
skt	55.8	18.3	47.3	17.5	59.3	-	39.6
Avg.	43.7	15.3	38.0	9.6	49.2	34.1	**31.7**

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:

$$\begin{aligned}\min_{F_{s2t},F_{t2s}} Loss_{ALIGN}^{Doublemap} = &\sum_{c=1}^{C}\mathcal{MMD}(\boldsymbol{f}^{s,c},\boldsymbol{f}^{t2s,c}) - \sum_{c_1=1}^{C}\sum_{c_2\neq c_1}^{C}\mathcal{MMD}(\boldsymbol{f}^{s,c_1},\boldsymbol{f}^{t2s,c_2}) \\ &+\sum_{c=1}^{C}\mathcal{MMD}(\boldsymbol{f}^{s2t,c},\boldsymbol{f}^{t,c}) - \sum_{c_1=1}^{C}\sum_{c_2\neq c_1}^{C}\mathcal{MMD}(\boldsymbol{f}^{s2t,c_1},\boldsymbol{f}^{t,c_2}).\end{aligned} \tag{14}$$

In this case, the feature extractor is retrained by equation 14; the source classifier is retrained by source samples.

For the fifth 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 label are available. Therefore, We just need to modify our sewing up operation to the following:

$$\begin{aligned}\min_{g} Loss_{ALIGN}^{HMA} = &\sum_{c=1}^{C}\mathcal{MMD}(\boldsymbol{f}^{s,c},\boldsymbol{f}^{t2s,c}) - \sum_{c_1=1}^{C}\sum_{c_2\neq c_1}^{C}\mathcal{MMD}(\boldsymbol{f}^{s,c_1},\boldsymbol{f}^{t2s,c_2}) \\ &+\sum_{c=1}^{C}\mathcal{MMD}(\boldsymbol{f}^{s2t,c},\boldsymbol{f}^{t,c}) - \sum_{c_1=1}^{C}\sum_{c_2\neq c_1}^{C}\mathcal{MMD}(\boldsymbol{f}^{s2t,c_1},\boldsymbol{f}^{t,c_2}).\end{aligned} \tag{15}$$

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

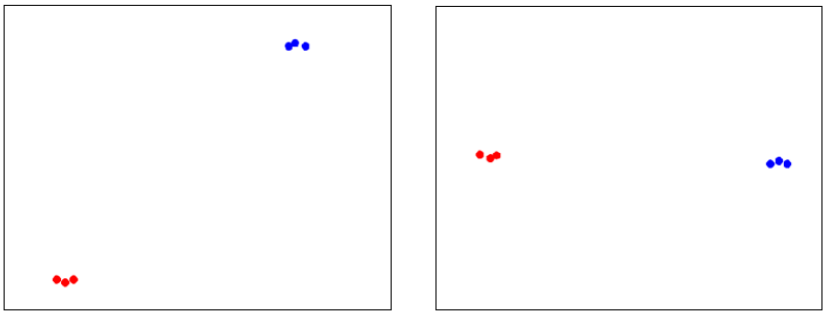
A.6 Comparisons to State-of-the-Art on DomainNet

DomainNet (Peng et al., 2019) 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 (Saito et al., 2018), BNM (Cui et al., 2020), SWD (Lee et al., 2019) and CGDM (Du et al., 2021). ResNet-50 is used as backbone for all methods. As shown in Table 7, our method surpasses all the previous alternatives by a large margin. This verifies the generic advantage of our approach in this more challenging larger-scale benchmark.

A.7 Model Analysis

An empirical visualization of homeomorphism.

For conceptual illustration of homeomorphism, we experiment with hand-designed toy data. Concretely, we first construct 6 2-dimensional feature points from two different clusters, as shown in Figure 3(a) in red and blue. We then transform these points with an INN based homeomorphism



(a) original feature (b) transformed feature

Figure 3: The empirical visualization of homeomorphism.

mapping. As we observed in Figure 3(b), the transformed points still preserve the structural cluster/group information.

Table 8: Different loss functions for consistency constraint on *Office-31*. CE: Cross Entropy; L_2: L_2-Norm.

Loss function	A→D	A→W	D→A	D→W	W→A	W→D
CE	95.8	94.9	79.4	99.1	77.8	100.0
L_2	95.8	95.1	79.3	99.3	77.6	100.0

Loss function for consistency constraint.

In equation 5, 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*. As shown in 8, the performance of our method is marginally affected by the loss function selection, suggesting the stability and flexibility of our model.

Table 9: Block number analysis on *Office-31*. HMA(DAN): Sewing up by DAN; HMA(CAN): Sewing up by CAN.

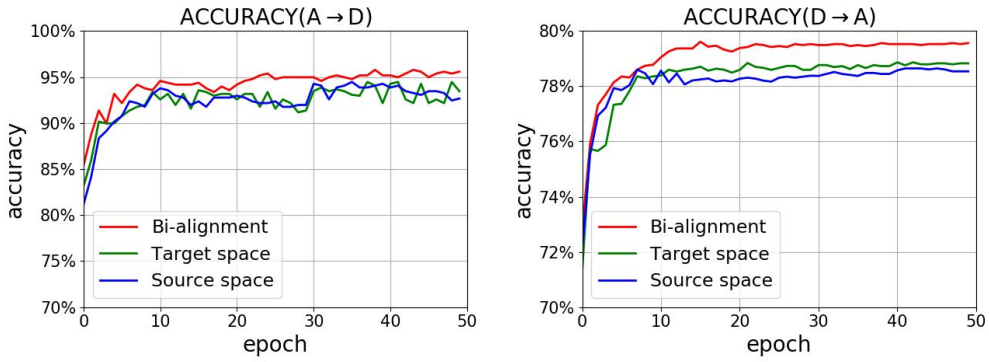
Number	1	2	3	4	5
HMA(DAN)	78.2	82.4	82.9	83.5	83.9
HMA(CAN)	87.6	89.4	90.3	90.7	91.2

How many blocks of INN do we need?

The forward and invertible process of INN for each block are shown in equation 1 and equation 2, so we need to discuss how many INN blocks we need. As shown in Table 9, the average accuracy on Office-31 are reported. It can be found that when the block number is changed from 1 to 2, the performance of both HMA(DAN) and HMA(CAN) has been greatly improved, while when the number of blocks is increased from 2 to 5, the performance increase is relatively slow. 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?

Obviously, 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 equation 3. We compare three strategies: unilateral sewing up:



(a) $A \rightarrow D$ (b) $D \rightarrow A$

Figure 4: 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.

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 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.

Table 10: 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

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 10. 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.

A.8 VISUAL ANALYSIS BY T-SNE

To intuitively understand the proposed HMA, we use t-SNE (Van der Maaten & Hinton, 2008) to visualize the classification results on *Office-31* based on two baselines, DAN and CAN, as shown in Fig. 5 and Fig.6, respectively. For both figures, the first row represents the results on tasks $A \rightarrow D$, and the second row shows the results on tasks $W \rightarrow A$. 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 results of our final proposed method, respectively. From Fig. 5 and Fig.6, 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.

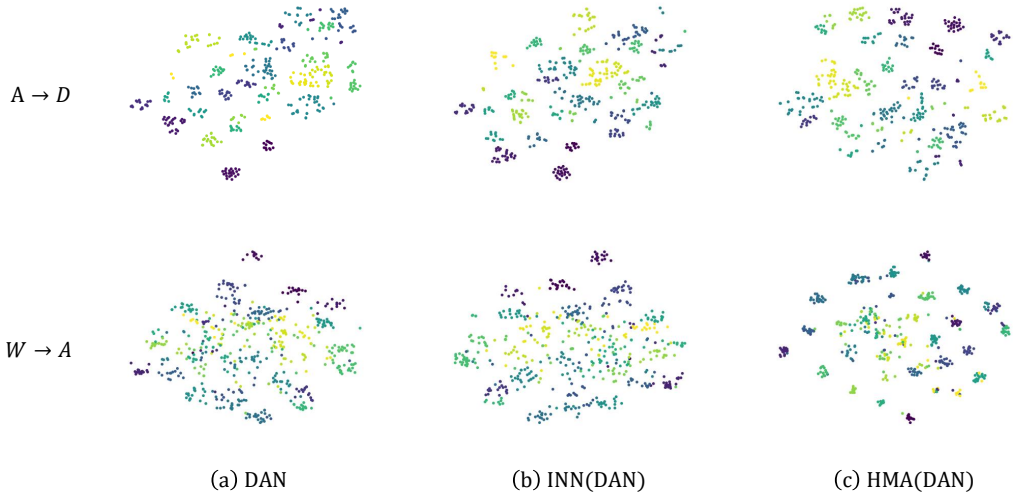


Figure 5: Visualization of ablation study using t-SNE on *Office-31* with DAN as the baseline. The first row is for task $A \rightarrow D$ and the second row represents the task $W \rightarrow A$. **Left:** DAN. **Center:** INN(DAN). **Right:** HMA(DAN).

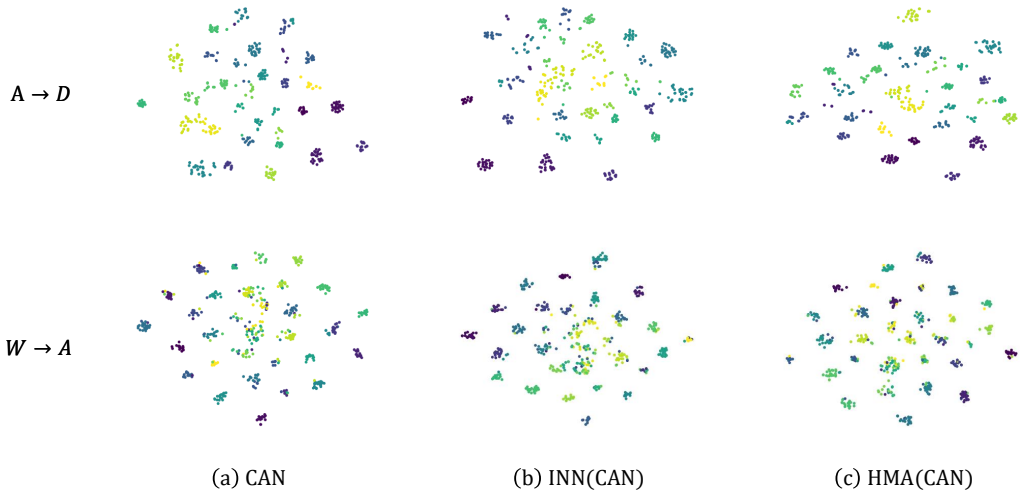


Figure 6: Visualization of ablation study using t-SNE on *Office-31* with CAN as the baseline. The first row represents the task $A \rightarrow D$ and the second row shows the result of task $W \rightarrow A$. **Left:** CAN. **Center:** INN(CAN). **Right:** HMA(CAN).

A.9 Visual analysis by Grad-CAM

We show the visualization of Grad-CAM (Selvaraju et al., 2017) on *Office-31* task $W \rightarrow A$, shown in Fig.7 and Fig.8. We randomly select 8 categories. For each category, one image is showed for activation mapping visualization. For both figures, from top to bottom, the images represent the results of original image, HMA(DAN), doublemap, CAN, HMA(CAN), respectively. From the visualization results, the focus of HMA(DAN) and doublemap is mostly on local points, while ignoring the characteristics of the whole object. Compared with the above two methods, CAN is slightly improved, but it still lacks certain accuracy. Our proposed HMA(CAN) better estimates the attention.

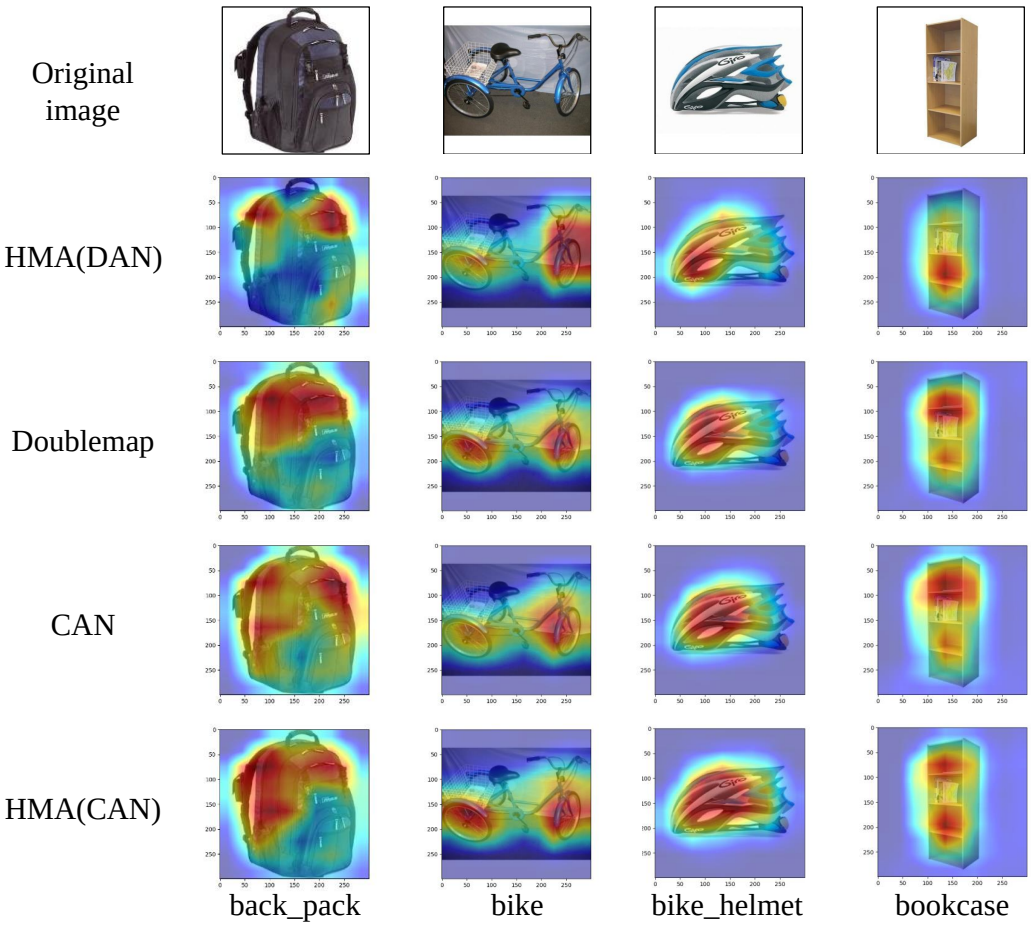


Figure 7: Visualization using CAM on *Office-31* task $W \rightarrow A$.

A.10 NETWORK STRUCTURE

In this section, we will go into detail about the neural network we use. For the feature extractor, Resnet (He et al., 2016) is used, but its original last layer which is 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 shown as follows.

For classifier, a fully connected linear layer is constructed for suit our tasks, which maps features to predictions. The dimension size of predictions are category number which are different in different datasets. Specifically, the dimensions of prediction are 31, 65, 12 in Office-31, Office-home and Visda-17 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 shown in Fig.10(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 same, which consists of four blocks. Each block consists of a fully connected neural networks a batchnorm and a relu function. The specific structure is shown in Fig.10(b).

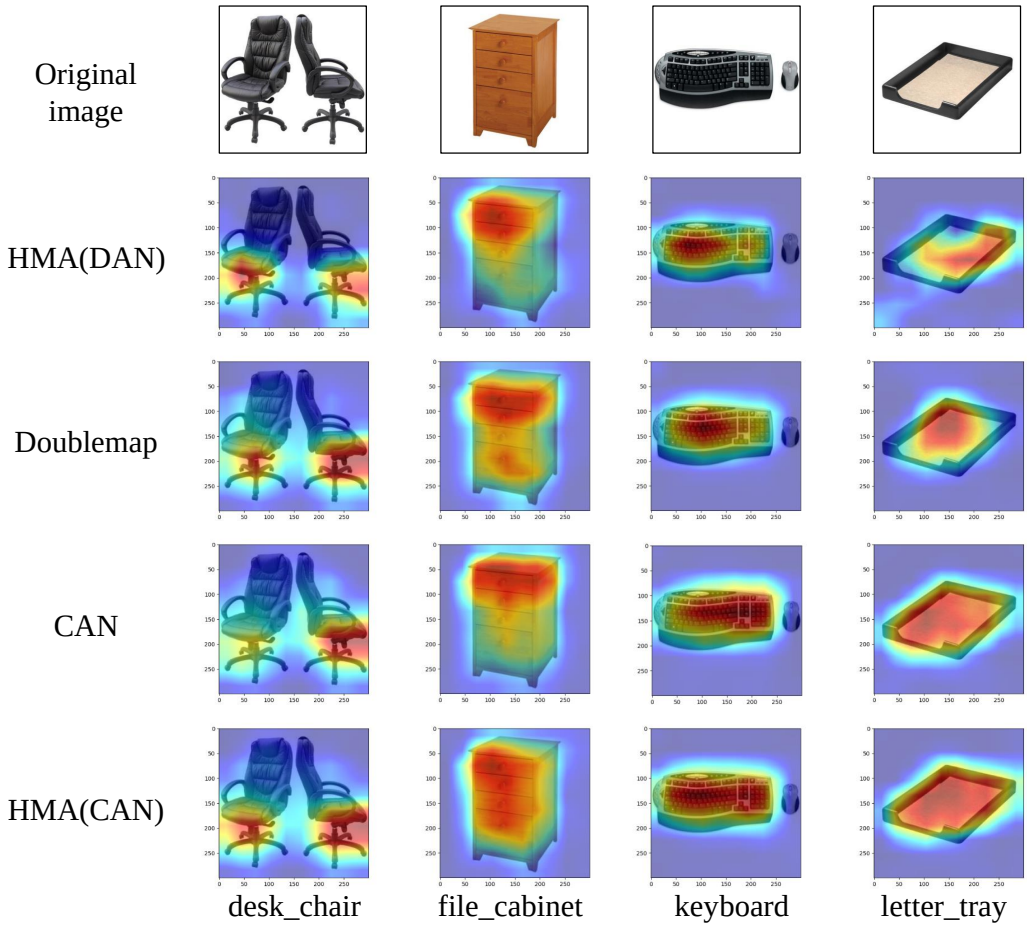


Figure 8: Visualization using CAM on *Office-31* task $W \rightarrow A$.

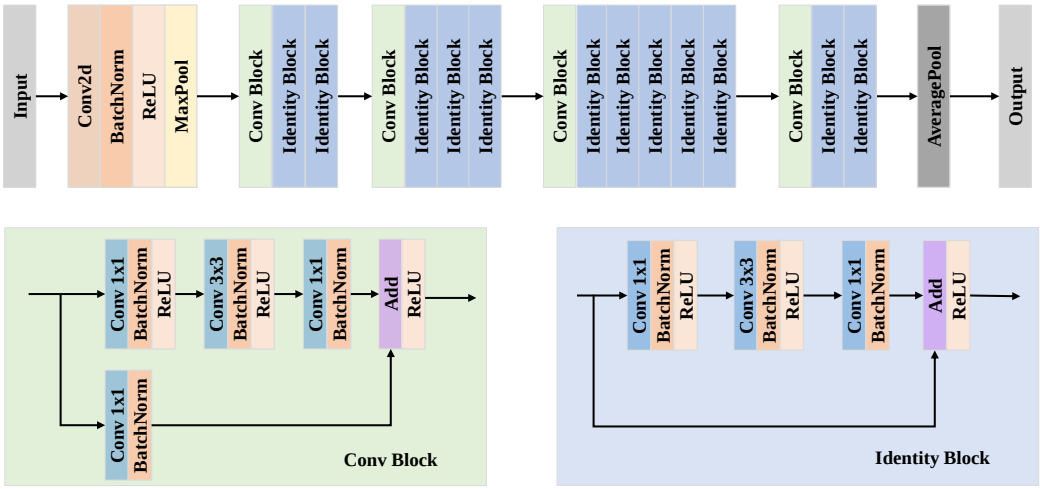


Figure 9: Network structure of Resnet-50.

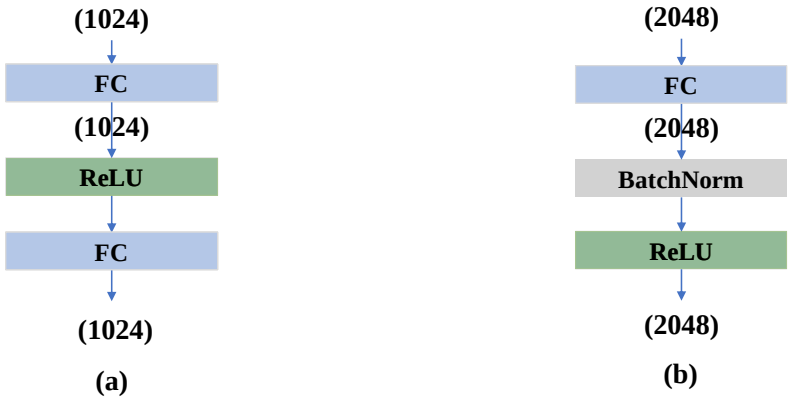


Figure 10: (a) Network structure of $s(\cdot)$ and $t(\cdot)$ in HMA. (b) Network structure of double mapping $F_{s2t}(\cdot)$ and $F_{t2s}(\cdot)$.