Mutual Learning of Joint and Separate Domain Alignments for Multi-Source Domain Adaptation

Yuanyuan Xu[1,2] Meina Kan[1,2] Shiguang Shan[1,2,3] Xilin Chen[1,2]
[1] Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Peng Cheng Laboratory, Shenzhen, China
yuanyuan.xu@vipl.ict.ac.cn {kanmeina,sgshan,xlchen}@ict.ac.cn

Abstract

Multi-Source Domain Adaptation (MSDA) aims at transferring knowledge from multiple labeled source domains to benefit the task in an unlabeled target domain. The challenges of MSDA lie in mitigating domain gaps and combining information from diverse source domains. In most existing methods, the multiple source domains can be jointly or separately aligned to the target domain. In this work, we consider that these two types of methods, i.e. joint and separate domain alignments, are complementary and propose a mutual learning based alignment network (MLAN) to combine their advantages. Specifically, our proposed method is composed of three components, i.e. a joint alignment branch, a separate alignment branch, and a mutual learning objective between them. In the joint alignment branch, the samples from all source domains and the target domain are aligned together, with a single domain alignment goal, while in the separate alignment branch, each source domain is individually aligned to the target domain. Finally, by taking advantage of the complementarity of joint and separate domain alignment mechanisms, mutual learning is used to make the two branches learn collaboratively. Compared with other existing methods, our proposed MLAN integrates information of different domain alignment mechanisms and thus can mine rich knowledge from multiple domains for better performance. The experiments on DomainNet, Office-31, and Digits-five datasets demonstrate the effectiveness of our method.

1. Introduction

The conventional machine learning approaches assume that the training data and testing data share the same distribution, so it can be ensured that the model trained with a large amount of data performs well on the testing data. [36, 22, 7, 15]. However, in practical applications, the testing data are often distributed differently from the training one (*e.g.*, comes from different scenes or devices), which causes significant performance degradation. To get a good model, one can relabel large quantities of data for each new scenario, which however is with high cost. Unsupervised Domain Adaptation (UDA) is an alternative yet low-cost way to optimize the model for a target domain by transferring knowledge from the sophisticated labeled source domain. In this way, no labeling is needed for a new target domain, which is low-cost and time-efficient.

Based on the number of source domains, UDA can be divided into two categories: Single-Source UDA methods and Multi-Source UDA methods. In this paper, we focus on Multi-Source Domain Adaptation (MSDA), which aims at transferring knowledge from multiple labeled source domains to benefit the task in the unlabeled target domain. This is challenging for two reasons: 1) Domain gaps exist between any two domains, so the model needs to deal with diverse inter-domain discrepancies. 2) There are no labels in the target domain, leading to insufficient supervision on how to extract complementary information from different source domains and how to combine them properly.

To address these challenges, recent works mainly attempt to align the distributions of source and target domains to reduce the domain discrepancies. These methods either jointly align all domains or separately align each source domain and the target domain. The joint domain alignment methods [53, 48, 17, 50] minimize the discrepancies of all domains jointly, by using a unique objective to reduce domain gaps between the target domain and the combined source domains. This induces one common classifier that produces category predictions for all domains. Methods of this type look for information that is shared across all domains, thus a consensus prediction result for a target sample can be obtained via the common classifier. However, it is hard to align multiple different distributions, especially

when domain gaps are notable. Besides, the common information decreases when the number of domains increases. In one word, the joint domain alignment methods take advantage of feature interaction before decision-making but suffer from much information loss.

The separate domain alignment methods [49, 32, 54, 33] instead align the feature distribution of the target domain with each source domain pair-wise and produce multiple source-specific classification results for a target sample. At last, all classification predictions are fused to get a final decision. Methods based on pair-wise alignment strategies simplify the alignment difficulty and make full use of shared information between the target and each source domain. But at the same time, each source domain interacts with the target domain separately, without exploring the assistance of other domains when transferring. Therefore, they can utilize more available information, but each domain makes decisions rather independently, which cannot utilize the complementary information between multiple source domains.

The above analysis shows that the joint and separate domain alignment methods are complementary in terms of their advantages. So in this paper, we present a mutual learning based alignment network (MLAN) to combine the two types of approaches for Multi-Source Domain Adaptation. Our proposed MLAN contains three modules: a joint alignment branch, a separate alignment branch, and a mutual learning objective between them. The joint and the separate alignment branches mitigate domain discrepancies jointly and separately, and the mutual learning module is designed to make the two branches utilize their complementarity for better results. In the mutual learning module, categorical and logits mutual learning objectives are proposed to make the joint and separate alignment branches fully interact. Specifically, categorical mutual learning is designed to enable the two branches to communicate their highly-confident predictions, while logits mutual learning is designed for communicating about those lowly-confident predictions.

Our contributions are summarized as follows:

- We consider the complementarity of the joint and separate domain alignment methods and design a mutual learning network to utilize their complementarity for MSDA. Specifically, a mutual learning module consisting of categorical and logits mutual learning objectives is designed to guide the collaborative learning between joint and separate domain alignment branches.

- Our method is evaluated on DomainNet, Office-31, and Digits-five datasets, and achieves state-of-the-art results.

2. Related Work

In addition to Multi-Source Domain Adaptation, our work is also related to Single-Source Unsupervised Domain Adaptation in terms of minimizing domain discrepancies. Besides, our mutual learning module is related to model distillation. All these related works are detailed below.

Single-Source Domain Adaptation. By merging all data from source domains and regarding them as a larger source domain, Single-Source UDA methods are also applicable for the MSDA problem. In recent years, deep Single-Source UDA methods are mainly instance-based and feature-based. The instance-based methods [21, 9, 1] aim to align distributions of the source and target domains at the image level by using generative adversarial networks (GANs) [5]. The key idea of the feature-based methods [23, 26, 3, 31, 24, 10, 13, 43, 6, 11] is to map data from two domains into a common space and align their distributions at the feature level. Among the feature-based methods, the gradient inversion layer is cleverly designed by DANN [3] to extract the domain-invariant features. Furthermore, CDAN [24] and MADA [31] apply the category information in the adversarial process to align feature distributions at both the domain and class levels. GSDA [11] further models the synchronization relationship among the local distribution pieces and global distribution. Besides, SRDC [43] and RSDA [6] maintain the intrinsic discrimination of target data when aligning feature distributions.

Multi-Source Domain Adaptation. MSDA methods mainly contain joint domain alignment ones[53, 48, 18] and separate domain alignment ones [49, 32, 54, 33]. For the joint domain alignment methods, interlinked with Single-Source UDA approaches, MDAN [53] aims to learn feature representations that are invariant to the multiple domain shifts while still being discriminative for the learning task. LtC-MSDA [48] utilizes class prototypes of different domains to construct a knowledge graph and combines information from different domains by using Graph Convolutional Networks (GCN). DRT [18] adapts the model's parameters for each sample to simplify the joint alignment between source domains and target domain. The separate domain alignment methods are mainly inspired by the distribution weighted combining rule [27]. DCTN [49] minimizes the discrepancy between the target and each source domain pair-wise. It produces multiple classification results for a target sample by different source-specific classifiers, and these results are then fused to get a final prediction. M^3SDA [32] argues that there are also distribution differences between source domains, so the constraint of aligning different source distributions is also taken into account. MDDA [54] considers distances on both the domain and instance levels, so source samples similar to the target are selected and used to finetune the source-specific classifiers. Despite considerable progress, existing works only consider

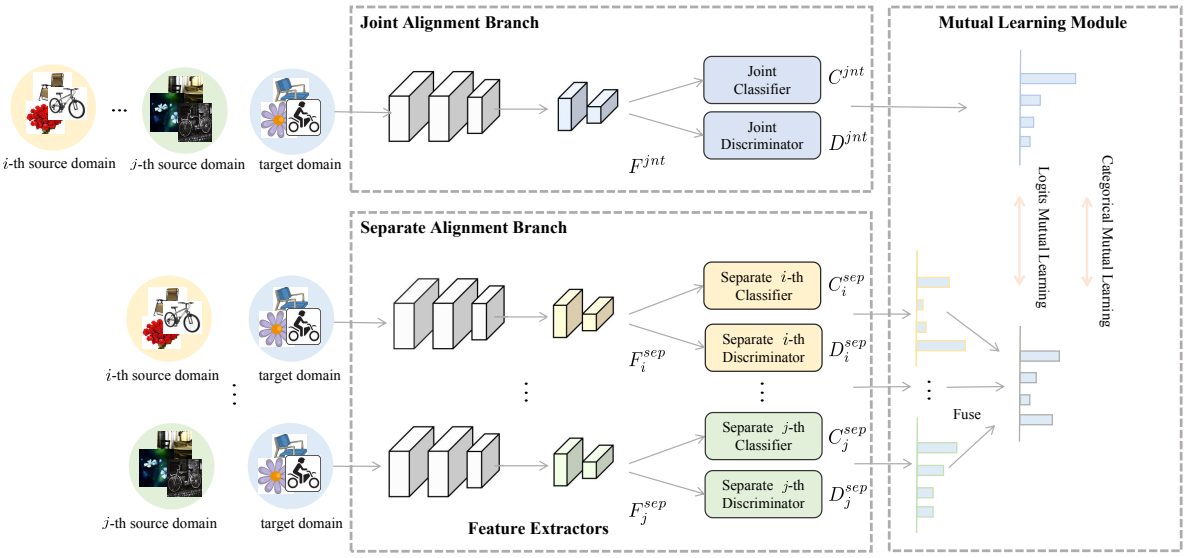


Figure 1: The architecture of our proposed MLAN. Our network is composed of three modules: a joint alignment branch, a separate alignment branch, and a mutual learning objective between them. The joint alignment branch aligns the target domain and all source domains jointly, while the separate alignment branch aligns the target domain and each source domain separately. Based on the complementarity of the two branches, the categorical mutual learning objective is designed for collaborating between those highly-confident samples from two branches, while logits mutual learning objective is designed for collaborating between those lowly-confident samples from two branches.

one type of alignment mechanism, and ML-MSDA [20] further explores the complementarity between different alignment strategies leading to better domain adaptation performance. In this work, we use mutual learning to combine the two types of approaches, which shares a similar idea as ML-MSDA [20] but largely differs from it in the design of the mutual learning module.

Model Distillation. The mutual learning module in our method is associated with model distillation. Model distillation [8] is an effective and widely used technique to transfer knowledge from a teacher to a student network in a variety of tasks [51, 2, 42, 14, 19, 35, 40]. Usually, the well-trained large model is used as the teacher, and the small model is used as the student. Besides, with a collection of student networks, DML [52] is proposed to make them learn collaboratively and teach each other throughout the training process. Inspired by DML [52], mutual distillation is exploited to design our logits mutual learning to make the joint and separate alignment branches learn collaboratively. In particular, the logits mutual learning loss in our work is unidirectional for a single sample to accommodate the unsupervised task, which is slightly different from the bidirectional one used in DML [52].

3. Method

3.1. Overview

In unsupervised multi-source domain adaptation, there are N labeled source domains and one unlabeled target domain, of which each draws from different distributions. The labeled images from the j^{th} source domain are written as (X_{s_j}, Y_{s_j}), where $X_{s_j} = \{x_{s_j}^i\}_{i=1}^{|X_{s_j}|}$ denotes the images and $Y_{s_j} = \{y_{s_j}^i\}_{i=1}^{|Y_{s_j}|}$ denotes the category labels. Similarly, the target image set is denoted as $X_t = \{x_t^i\}_{i=1}^{|X_t|}$. All domains share the same category set, and M is the total number of classes. The goal of MSDA is to design a classifier that works for the target domain, by utilizing all the labeled source data and unlabeled target data.

Aiming for Multi-Source Domain Adaptation (MSDA), we propose MLAN to utilize the complementarity of joint domain alignment and separate domain alignment. The overall framework is shown in Fig.1. Our proposed MLAN consists of three modules, including a joint alignment branch, a separate alignment branch, and a mutual learning objective between them. In this section, we will first introduce the two branches and then present the mutual learning method that allows them to learn collaboratively.

Algorithm 1 Algorithm for MLAN

Input: labeled images from N source domains $\{X_{s_j}, Y_{s_j}\}_{j=1}^{N}$; unlabeled images of the target domain X_t; feature extractors $F = \{F^{jnt}, F^{sep}\}$; classifiers $C = \{C^{jnt}, C^{sep}\}$; discriminators $D = \{D^{jnt}, D^{sep}\}$; confidence threshold γ; loss weight α; iteration number IterA, IterB.

Output: well-trained feature extractors F^*, classifiers C^*, and discriminators D^*.

Pre-train C and F using source data to get a warm-up.
while not converged **do**
  // Adversarially Training Each Branch
  for 1:IterA **do**
    Calculate the adversarial and classification loss of the joint alignment branch by Eq.(1)(2)(3);
    Calculate the adversarial and classification loss of each sub-branch of the separate alignment branch by Eq.(4)(5)(6);
    Backward to update D and F.
  end for
  // Categorical and Logits Mutual Learning
  Calculate pseudo labels for the target samples to get (X_t^p, Y_t^p) by source-guided K-means clustering.
  for 1:IterB **do**
    Calculate the categorical mutual learning loss by Eq.(12) with highly-confident samples.
    Calculate the logits mutual learning loss by Eq.(15) with lowly-confident samples.
    Calculate the classification loss of the joint and separate alignment branches by Eq.(3)(6).
    Backward to update C and F.
  end for
end while

3.2. The Joint Alignment Branch

The joint alignment branch minimizes the discrepancies of all domains jointly, by aligning the target domain with the combined source domains. It contains a feature extractor F^{jnt}, a common classifier C^{jnt} for the category classification of all domains, and a domain discriminator D^{jnt}.

Specifically, given an image x, F^{jnt} encodes x as $f^{jnt} = F^{jnt}(x)$. On one hand, the discrepancies among all domains should be minimized, so the samples from the target domain can be classified by the common classifier C^{jnt} based on the domain-agnostic feature f^{jnt}. To reduce the domain discrepancies, an adversarial learning strategy is exploited, i.e., a domain discriminator D^{jnt} is designed to distinguish the target domain and source domains, while the feature extractor F^{jnt} tries to confuse the domain discriminator to get domain invariant feature. For the discriminator D^{jnt}, the discriminating loss is:

$$\begin{aligned}\mathcal{L}_{adt}^{jnt}(D^{jnt}) = \mathbb{E}_{x\in X_s}\left[D^{jnt}(f^{jnt}, p^{jnt}) - 0\right]^2 \\ + \mathbb{E}_{x\in X_t}\left[D^{jnt}(f^{jnt}, p^{jnt}) - 1\right]^2.\end{aligned} \tag{1}$$

where X_s is the set of samples from all source domains, and $p^{jnt} \in \mathbb{R}^{|M|}$ is the softmax output by feeding f^{jnt} into classifier C^{jnt}. The loss is a combination of LSGANs [28] and CDAN [24]. Specifically, taking a feature and its corresponding category probabilities as inputs, D^{jnt} classifies domains by outputting 0 and 1 when they come from the source domains and the target domain, respectively.

Then, for the feature extractor F^{jnt}, the confusing loss is:

$$\begin{aligned}\mathcal{L}_{adt}^{jnt}(F^{jnt}) = \mathbb{E}_{x\in X_s}\left[D^{jnt}(f^{jnt}, p^{jnt}) - \frac{1}{2}\right]^2 \\ + \mathbb{E}_{x\in X_t}\left[D^{jnt}(f^{jnt}, p^{jnt}) - \frac{1}{2}\right]^2.\end{aligned} \tag{2}$$

F^{jnt} tries to confuse D^{jnt} by forcing it to output $\frac{1}{2}$ whenever the feature and the category probabilities come from the target domain or not, to generate domain-agnostic features that can't be distinguished by D^{jnt}.

On the other hand, the extracted feature should be discriminative for the classification task, so a classification loss is imposed on the softmax output of labeled source domain samples based on the classifier C^{jnt}:

$$\mathcal{L}_{cls}^{jnt}(C^{jnt}, F^{jnt}) = \frac{1}{N}\sum_{j=1}^{N}\mathbb{E}_{(x,y)\in(X_{s_j}, Y_{s_j})}\left[\mathcal{L}_{ce}(p^{jnt}, y)\right], \tag{3}$$

where $\mathcal{L}_{ce}$ is the cross-entropy loss function between the network prediction p^{jnt} and the ground truth label y.

3.3. The Separate Alignment Branch

The separate alignment branch aligns the target domain and each source domain in a pair-wise manner. This branch consists of N sub-branches, and each sub-branch is responsible for domain alignment between the target domain and one source domain.

Specifically, for the j^{th} sub-branch, the feature f_j^{sep} from the feature extractor F_j^{sep} is enforced to be domain invariant as well as discriminative. It is constrained by classification loss together with adaptation loss similar as that in Eq.(2) and (3) of the joint alignment branch. Specifically, the domain classification loss for D_j^{sep} is:

$$\begin{aligned}\mathcal{L}_{adt}^{sep,j}(D_j^{sep}) = \mathbb{E}_{x\in X_{s_j}}\left[D_j^{sep}(f_j^{sep}, p_j^{sep}) - 0\right]^2 \\ + \mathbb{E}_{x\in X_t}\left[D_j^{sep}(f_j^{sep}, p_j^{sep}) - 1\right]^2,\end{aligned} \tag{4}$$

and the confusing loss for F_j^{sep} is:

$$\begin{aligned}\mathcal{L}_{adt}^{sep,j}(F_j^{sep}) = \mathbb{E}_{x\in X_{s_j}}\left[D_j^{sep}(f_j^{sep}, p_j^{sep}) - \frac{1}{2}\right]^2 \\ + \mathbb{E}_{x\in X_t}\left[D_j^{sep}(f_j^{sep}, p_j^{sep}) - \frac{1}{2}\right]^2.\end{aligned} \tag{5}$$

The classification loss of the j^{th} branch is:

$$\mathcal{L}_{cls}^{sep,j}(C_j^{sep}, F_j^{sep}) = \mathbb{E}_{(x,y)\in(X_{s_j}, Y_{s_j})}\left[\mathcal{L}_{ce}(p_j^{sep}, y)\right], \tag{6}$$

where $p_j^{sep} \in \mathbb{R}^{|M|}$ is the softmax output from C_j^{sep}.

As there are multiple classifiers, one for each source-target pair, the final prediction of a given target sample is obtained by combing the predictions from all classifiers. Different from the straightforward average of multiple prediction results, in our method, the final result is obtained as the weighted linear combination of multiple prediction results from N sub-branches:

$$p^{sep} = \sum_{j=1}^{N} w_j p_j^{sep}, \tag{7}$$

where $p_j^{sep} \in \mathbb{R}^{|M|}$ is the predicted class probabilities of the j^{th} sub-branch, and w_j is the weight for prediction from the j^{th} sub-branch which considers the similarity between the j^{th} source domain and target domain as well as the prediction certainty for each sample. Specifically, w_j is calculated as below:

$$w_j = \frac{w_j^{sim} w_j^{cer}}{\sum_{i=1}^{N} w_i^{sim} w_i^{cer}}. \tag{8}$$

Here, w_j^{sim} reflects the similarity between the j^{th} source domain and the target domain, and it should be large if the two domains are similar. So, it is calculated as that in [49] i.e. $w_j^{sim} = \frac{\mathcal{L}_{adt}^{sep,j}(D_j^{sep})}{\sum_{k=1}^{N} \mathcal{L}_{adt}^{sep,k}(D_k^{sep})}$. w_j^{cer} denotes the prediction certainty, which is calculated as $w_j^{cer} = e^{-H(p_j^{sep})}$, where $H(\cdot)$ is the entropy of a probability distribution. Note that w_j^{cer} varies depending on different samples, and the sample subscript is omitted for simplicity.

3.4. Mutual Learning

The mutual learning module is designed to make the two branches complement each other for better results. The two branches not only can authenticate each other on those highly-confident target domain samples but also can communicate about those uncertain ones. So, the mutual learning module is equipped with two kinds of objectives aiming for dealing with highly-confident and lowly-confident target domain samples respectively, i.e. categorical mutual learning objective and logits mutual learning objective. The overall loss of mutual learning is:

$$\mathcal{L}_{mut}(C, F) = \mathcal{L}_{cat}(C, F) + \alpha \mathcal{L}_{log}(C, F), \tag{9}$$

where $\mathcal{L}_{cat}(C, F)$ is the loss for categorical mutual learning, $\mathcal{L}_{log}(C, F)$ is for logits mutual learning, and α is a hyperparameter that balances two losses.

Categorical Mutual Learning. When the joint and separate alignment branches give consistent predictions to a target sample and at least one branch with high confidence, we can safely think that the classification result for this sample is reliable. Under this circumstance, the pseudo label is directly assigned to the target sample for training the two branches. To generate more robust pseudo labels, the pseudo label of a target sample is assigned by considering both the prediction p^{jnt}/p^{sep} from source classifiers and the clustering result of the target domain detailed as below.

Step 1: categorical prediction from the joint alignment branch. Since the domain discrepancies between source and target domains are reduced, so the labeled source domains and unlabeled target domain are both used for calculating categorical prediction of target domain samples. K-means clustering is firstly applied on target domain features, with the cluster centers initialized by M source domain centers (*e.g.* the k^{th} source center is the average of all features of class k in the source domains). This type of clustering is denoted as source-guided K-means clustering. After clustering, the updated category centers are denoted as $\{c_k^{jnt}\}_{k=1}^{M}$. Then, each target sample is assigned with the class label according to its nearest cluster center as follows:

$$q_k^{jnt} = \frac{e^{s \cdot cos(\theta_k^{jnt})}}{\sum_{i=1}^{M} e^{s \cdot cos(\theta_i^{jnt})}}, \tag{10}$$

where $cos(\theta_k^{jnt}) = \frac{<f^{jnt}, c_k^{jnt}>}{\|f^{jnt}\| \|c_k^{jnt}\|}$ denotes the cosine similarity between the target feature f^{jnt} and the k^{th} cluster center c_k^{jnt}, and s is a hyperparameter which is set to 20 following the works for face recognition [34, 47]. Thus $q^{jnt} \in \mathbb{R}^{|M|}$ represents the predicted category probabilities for the target sample, with the k^{th} dimension corresponds to the probability it belongs to category k.

Step 2: categorical prediction from the separate alignment branch. Similar as that in step 1, each sub-branch firstly does the source-guided K-means clustering individually with q_j^{sep} denoting the predicted category probabilities of the j^{th} sub-branch. Then, the final prediction by considering all the separate alignment branches is calculated similarly to that in Eq.(7):

$$q^{sep} = \sum_{j=1}^{N} w_j q_j^{sep}, \tag{11}$$

where w_j is calculated similarly to that in Eq.(8).

Step 3: pseudo category labels for those confident target domain samples. Now, for the target sample, two predicted category labels $\hat{y}^{jnt}$ and $\hat{y}^{sep}$ are obtained from the two branches, with q^{jnt} and q^{sep} denoting corresponding predicted category probabilities. If the two branches give consistent predictions (i.e. $\hat{y}^{jnt} = \hat{y}^{sep}$) and the maximum category probability of either branch exceeds γ, the pseudo label of this target domain sample is assigned as $\hat{y}^{jnt}$.

For these target domain samples assigned with category pseudo labels (X_t^p, Y_t^p), their classification loss is calculated as:

$$\mathcal{L}_{cat}(C, F) = \mathcal{L}_{cat}^{jnt}(C^{jnt}, F^{jnt}) + \mathcal{L}_{cat}^{sep}(C^{sep}, F^{sep}), \tag{12}$$

where

$$\mathcal{L}_{cat}^{jnt}(C^{jnt}, F^{jnt}) = \mathbb{E}_{(x,\hat{y}) \in (X_t^p, Y_t^p)} \left[\mathcal{L}_{ce}(p^{jnt}, \hat{y}) \right], \tag{13}$$

and

$$\mathcal{L}_{cat}^{sep}(C^{sep}, F^{sep}) = \sum_{j=1}^{N} \mathbb{E}_{(x,\hat{y}) \in (X_t^p, Y_t^p)} \left[\mathcal{L}_{ce}(p_j^{sep}, \hat{y}) \right]. \tag{14}$$

Logits Mutual Learning. When the two branches make different decisions, or they make consensus predictions but both with low certainty, the pseudo labels are no longer reliable. In this circumstance, i.e. when target samples have not been assigned with pseudo labels, mutual distillation is introduced to make them learn collaboratively.

To encourage each branch to learn the advantage of the other, the branch with higher confidence prediction supervises the other. Therefore, the distillation is unidirectional for a given target sample, while bidirectional for the whole distillation process. Here, the entropy of predicted category probabilities is used to measure the confidence, with lower entropy meaning higher confidence. For example, for a given target sample, if the joint alignment branch gets a lower entropy prediction, then its classifying output is used to supervise the separate alignment branch, and vice versa.

Specifically, Kullback Leibler (KL) divergence loss is used to constrain the logits of two branches here. The logits mutual learning loss is:

$$\begin{aligned}\mathcal{L}_{log}(C,F) = &\mathbb{E}_{x\in\{x|H(p^{sep})<H(p^{jnt})\}} KL(p^{sep}||p^{jnt}) \\ &+ \mathbb{E}_{x\in\{x|H(p^{jnt})<H(p^{sep})\}} KL(p^{jnt}||p^{sep}),\end{aligned} \tag{15}$$

where p^{jnt} and p^{sep} are the predicted category probabilities from the two branches. $KL(\cdot)$ is the KL Divergence between two probability distributions, and $H(\cdot)$ is the entropy of a probability distribution. The first term of Eq.(15) is the logits mutual learning loss when the separate alignment branch is more confident than the joint alignment branch, while the second term is the logits mutual learning loss when the joint alignment branch is more confident than the separate alignment branch.

With Eq.(12) and Eq.(15), we can get the overall loss of the mutual learning module, which is shown in Eq.(9). The whole model C, F, D can be trained by optimizing the mutual learning module and the two branches alternately, as in Alg.1. The mutual learning module is only used for training, and during testing, the final prediction of a target domain sample is calculated as the average of p^{jnt} and p^{sep} from the joint and separate alignment branches:

$$p = 0.5 * (p^{jnt} + p^{sep}). \tag{16}$$

4. Experiments

We conduct experiments on three commonly used datasets including DomainNet, Office-31, and Digits-five to compare the proposed method with existing ones. Besides, we carry out the ablation study on DomainNet to evaluate the effectiveness of our method design.

4.1. Datasets and Settings

Datasets. DomainNet [32] is a large scale dataset, with six domains, i.e. clipart (clp), infograph (info), painting (pnt), quickdraw (qdr), real (rel) and sketch (skt). There are 345 categories, and $\sim$0.6 million images in this dataset. **Office-31** [37] is a classical and widely used dataset with 4652 images and 31 categories. It consists of objects from 3 different domains, i.e. Amazon (A), Webcam (W) and Dslr (D). **Digits-five** contains five digit sub-datasets: MNIST (mt) [16], MNIST-M (mm) [4], SVHN (sv) [29], USPS (up) [12], and Synthetic Digits (syn) [4]. Each sub-dataset is considered as one domain, containing images of numbers ranging from 0 to 9. We follow the same setting in LtC-MSDA [48] to sample the data for Digits-five and use the default train/test setup for other datasets. All testing data are used for evaluations as default.

Implementation Details. The proposed algorithm is implemented in PyTorch. For model architecture, LeNet, ResNet-50, and ResNet-101 are exploited as the backbones for Digits-five, Office-31, and DomainNet respectively, which are the same as the compared methods. Each feature extractor includes a shared backbone and several branch-specific layers as shown in Fig.1. Note that the model size (number of parameters) of our proposed MLAN is only about 1.07 times that of DCTN [49] when using ResNet-101 as the backbone, i.e. similar efficiency. For the training process, LeNet is trained from scratch. ResNet-50 and ResNet-101 are initialized by ImageNet pre-trained weights. All models are firstly trained with only source data to get a warm-up. Then, the mutual learning module and the two branches are optimized alternately as in Alg.1. Adam optimizer is used to update LeNet with the learning rate of 2×10^{-4}, and to update ResNet-50 and ResNet-101 with a small learning rate of 10^{-5}. The threshold γ for determining pseudo labels is set to 0.9 as in [49, 38], and the weight α for balancing the categorical and logits mutual learning losses is set to 1.0 as discussed in [8]. As for training time, it only takes a few hours to train a transferring task after the warm-up for a small dataset (*e.g*. Digits-five or Office-31), and when the dataset is larger, the training time becomes longer (*e.g*. an average of 30 hours for DomainNet).

4.2. Comparison with State-of-the-arts

Experiments on DomainNet. We first compare the proposed method to the existing ones on the most challenging dataset DomainNet. Three categories of DA approaches are compared including Single-Best methods, Source-Combine methods, and Multi-Source methods. The comparison results are shown in Tab.1. As can be seen, Source-Combine methods are better than Single-Best methods, and Multi-Source methods perform the best. This indicates that the data of multiple source domains provides richer information for the task of the target domain, and elaborately designed algorithms can make better use of multiple source domains.

Among Multi-Source methods in Tab.1, MDAN [53], LtC-MSDA [48] and DRT+ST [18] are based on joint domain alignment, and other methods such as DCTN [49],

Table 1: Comparison with existing methods on DomainNet dataset in terms of top-1 classification accuracy (mean±std%)

Standards	Methods	→ clp	→ inf	→ pnt	→ qdr	→ rel	→ skt	Avg
Single Best	Source-only	39.6 ± 0.6	8.2 ± 0.8	33.9 ± 0.6	11.8 ± 0.7	41.6 ± 0.8	23.1 ± 0.7	26.4
	DAN [23]	39.1 ± 0.5	11.4 ± 0.8	33.3 ± 0.6	16.2 ± 0.4	42.1 ± 0.7	29.7 ± 0.9	28.6
	JAN [26]	35.3 ± 0.7	9.1 ± 0.6	32.5 ± 0.7	14.3 ± 0.6	43.1 ± 0.8	25.7 ± 0.6	26.7
	DANN [3]	37.9 ± 0.7	11.4 ± 0.9	33.9 ± 0.6	13.7 ± 0.6	41.5 ± 0.7	28.6 ± 0.6	27.8
	ADDA [44]	39.5 ± 0.8	14.5 ± 0.7	29.1 ± 0.8	14.9 ± 0.5	41.9 ± 0.8	30.7 ± 0.7	28.4
	MCD [39]	42.6 ± 0.3	19.6 ± 0.8	42.6 ± 1.0	3.8 ± 0.6	50.5 ± 0.4	33.8 ± 0.9	32.2
Source Combine	Source-only	47.6 ± 0.5	13.0 ± 0.4	38.1 ± 0.5	13.3 ± 0.4	51.9 ± 0.9	33.7 ± 0.5	32.9
	DAN [23]	45.4 ± 0.5	12.8 ± 0.9	36.2 ± 0.6	15.3 ± 0.4	48.6 ± 0.7	34.0 ± 0.5	32.1
	JAN [26]	40.9 ± 0.4	11.1 ± 0.6	35.4 ± 0.5	12.1 ± 0.7	45.8 ± 0.6	32.3 ± 0.6	29.6
	DANN [3]	45.5 ± 0.6	13.1 ± 0.7	37.0 ± 0.7	13.2 ± 0.8	48.9 ± 0.7	31.8 ± 0.6	32.6
	ADDA [44]	47.5 ± 0.8	11.4 ± 0.7	36.7 ± 0.5	14.7 ± 0.5	49.1 ± 0.8	33.5 ± 0.5	32.2
	MCD [39]	54.3 ± 0.6	22.1 ± 0.7	45.7 ± 0.6	7.6 ± 0.5	58.4± 0.7	43.5 ± 0.6	38.5
Multi-Source	MDAN [53]	52.4 ± 0.6	21.3 ± 0.8	46.9 ± 0.4	8.6 ± 0.6	54.9 ± 0.6	46.5 ± 0.7	38.4
	M^3SDA [32]	58.6 ± 0.5	26.0 ± 0.9	52.3 ± 0.6	6.3 ± 0.6	62.7 ± 0.5	49.5 ± 0.8	42.6
	MDDA [54]	59.4 ± 0.6	23.8 ± 0.8	53.2 ± 0.6	12.5 ± 0.6	61.8 ± 0.5	48.6 ± 0.8	43.2
	ML-MSDA [20]	61.4 ± 0.8	26.2 ± 0.4	51.9 ± 0.2	19.1 ± 0.3	57.0 ± 1.0	50.3 ± 0.7	44.3
	LtC-MSDA [48]	63.1 ± 0.5	28.7 ± 0.7	56.1 ± 0.5	16.3 ± 0.5	66.1 ± 0.6	53.8 ± 0.6	47.4
	DCTN [49]	69.6 ± 0.7	27.5 ± 0.6	57.3 ± 0.6	17.8 ± 0.5	72.5 ± 0.6	55.3 ± 0.5	49.8
	DRT+ST[18]	71.0 ± 0.2	**31.6** ± 0.4	**61.0** ± 0.3	12.3 ± 0.4	71.4 ± 0.2	**60.7** ± 0.3	51.3
	MLAN (ours)	**71.4** ± 0.2	29.3 ± 0.3	59.5 ± 0.2	**28.4** ± 0.5	**73.9** ± 0.1	58.7 ± 0.4	**53.5**

Table 2: Classification accuracy (%) on Office-31 dataset

Standards	Methods	→ D	→ W	→ A	Avg
Single Best	Source-only	99.3	96.7	62.5	86.2
	DDC [45]	98.2	95.0	67.4	86.9
	DAN [23]	99.5	96.8	66.7	87.7
	CORAL [41]	**99.7**	98.0	65.3	87.7
	DANN [3]	99.1	96.9	68.2	88.1
	RTN [25]	99.4	96.8	66.2	87.5
Source Combine	DAN [23]	99.6	97.8	67.6	88.3
	CORAL [41]	99.3	98.0	67.1	88.1
	DANN [3]	**99.7**	98.1	67.6	88.5
Multi-Source	DCTN [49]	99.3	98.2	64.2	87.2
	M^3SDA [32]	99.3	98.0	67.2	88.2
	LtC-MSDA [48]	99.4	97.7	68.6	88.6
	M^3SDA-β [32]	99.6	**99.3**	69.4	89.5
	MFSAN [55]	99.5	98.5	72.7	90.2
	MLAN (ours)	99.6	98.8	**75.7**	**91.4**

M^3SDA [32], and MDDA [54] are based on separate domain alignment. ML-MSDA [20] is a method that also combines two domain alignment mechanisms. We re-implement DCTN [49] on DomainNet, achieving an average accuracy of 49.8% with a long time (30 epochs) warm-up, and the results of other compared methods are directly copied from the works [48, 18].

Our method MLAN achieves an average accuracy of 53.5% on 6 transfer tasks of DomainNet, which exceeds existing methods by a large margin. MLAN outperforms the state-of-the-art method of joint domain alignment i.e. DRT+ST [18] by 2.2% and state-of-the-art method of separate domain alignment i.e. DCTN [49] by 3.7%, which shows that combining the two domain alignment mechanisms is useful. Also, our method is 9.2% better than ML-MSDA [20], demonstrating the superiority of the mutual learning module in our MLAN. In addition, there is a large difference between MLAN and other approaches for the '→ qdr' task. Based on some initial investigation, we think this improvement comes from two aspects, mutual learning and pseudo labeling on the target domain. The effectiveness of mutual learning has been analyzed in Tab.4. Besides, when there is a large gap between the 'qdr' target domain and other source domains, our source-guided K-means clustering can utilize guidance from the source domains and the structure information from the target domain, to effectively improve the quality of the pseudo labels.

Experiments on Office-31. As shown in Tab.2, on Office-31, we compare the proposed method to recent methods based on the ResNet-50 backbone. Among Muti-Source methods, DCTN [49], M^3SDA(-β) [32], and MFSAN [55] utilize the separate domain alignment, and the experimental results are directly copied from the works [55, 30, 46]. LtC-MSDA [48] applies the joint domain alignment, and its results are obtained by re-implementing the method on Office-31 dataset based on the official open-source code. On the whole, the conclusion on Office-31 is similar to that on DomainNet, but with smaller performance gaps between different methods as Office-31 is easier than DomainNet. Specifically, our proposed MLAN achieves an average classification accuracy of 91.4%, which outperforms other existing methods. In particular, our model achieves 3% accuracy improvement than the state-of-the-art method MFSAN [55] on the '→ A' task. The performance of MLAN on Office-31 also shows the superiority of combining two domain alignment methods.

Experiments on Digits-five. The experimental results are shown in Tab.3. Digits-five only contains 10 classes of numbers, which is much easier than the object datasets. For some tasks such as '→ mt' and '→ up' , the performance is almost saturated. So, most methods perform well on this dataset. Even so, our method still suppresses the other four Multi-Source methods and achieves comparable results as DRT+ST [18], showing the advantage of our method.

Table 3: Comparison with existing methods on Digits-five dataset in terms of top-1 classification accuracy (mean±std%)

Standards	Methods	→ mm	→ mt	→ up	→ sv	→ syn	Avg
Single Best	Source-only	59.2 ± 0.6	97.2 ± 0.6	84.7 ± 0.8	77.7 ± 0.8	85.2 ± 0.6	80.8
	DAN [23]	63.8 ± 0.7	96.3 ± 0.5	94.2 ± 0.9	62.5 ± 0.7	85.4 ± 0.8	80.4
	CORAL [41]	62.5 ± 0.7	97.2 ± 0.8	93.5 ± 0.8	64.4 ± 0.7	82.8 ± 0.7	80.1
	DANN [3]	71.3 ± 0.6	97.6 ± 0.8	92.3 ± 0.9	63.5 ± 0.8	85.4 ± 0.8	82.0
	ADDA [44]	71.6 ± 0.5	97.9 ± 0.8	92.8 ± 0.7	75.5 ± 0.5	86.5 ± 0.6	84.8
Source Combine	Source-only	63.4 ± 0.7	90.5 ± 0.8	88.7 ± 0.9	63.5 ± 0.9	82.4 ± 0.6	77.7
	DAN [23]	67.9 ± 0.8	97.5 ± 0.6	93.5 ± 0.8	67.8 ± 0.6	86.9 ± 0.5	82.7
	DANN [3]	70.8 ± 0.8	97.9 ± 0.7	93.5 ± 0.8	68.5 ± 0.5	87.4 ± 0.9	83.6
	JAN [26]	65.9 ± 0.7	97.2 ± 0.7	95.4 ± 0.8	75.3 ± 0.7	86.6 ± 0.6	84.1
	ADDA [44]	72.3 ± 0.7	97.9 ± 0.6	93.1 ± 0.8	75.0 ± 0.8	86.7 ± 0.6	85.0
	MCD [39]	72.5 ± 0.7	96.2 ± 0.8	95.3 ± 0.7	78.9 ± 0.8	87.5 ± 0.7	86.1
Multi-Source	MDAN [53]	69.5 ± 0.3	98.0 ± 0.9	92.4 ± 0.7	69.2 ± 0.6	87.4 ± 0.5	83.3
	DCTN [49]	70.5 ± 1.2	96.2 ± 0.8	92.8 ± 0.3	77.6 ± 0.4	86.8 ± 0.8	84.8
	M^3SDA [32]	72.8 ± 1.1	98.4 ± 0.7	96.1 ± 0.8	81.3 ± 0.9	89.6 ± 0.6	87.7
	MDDA [54]	78.6 ± 0.6	98.8 ± 0.4	93.9 ± 0.5	79.3 ± 0.8	89.7 ± 0.7	88.1
	LtC-MSDA [48]	85.6 ± 0.8	99.0 ± 0.4	98.3 ± 0.4	83.2 ± 0.6	93.0 ± 0.5	91.8
	DRT+ST[18]	81.0 ± 0.3	**99.3** ± 0.1	**98.4** ± 0.1	**86.7** ± 0.4	**93.9** ± 0.3	**91.9**
	MLAN (ours)	**86.3** ± 0.3	98.6± 0.0	97.5± 0.2	82.8 ± 0.1	93.0 ± 0.3	91.6

Table 4: Ablation study for the mutual learning module

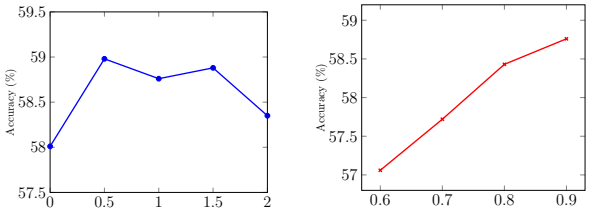
Methods	→ clp	→ inf	→ pnt	→ qdr	→ rel	→ skt	Avg
joint	70.4	28.2	58.1	22.2	72.6	55.9	51.3
separate	70.0	28.5	58.6	25.5	72.5	56.7	51.9
w. categorical	71.1	28.8	58.9	26.4	73.7	58.0	52.8
MLAN	**71.4**	**29.3**	**59.5**	**28.4**	**73.9**	**58.7**	**53.5**

4.3. Ablation Study and Sensitivity Analysis

Effectiveness of Categorical and Logits Mutual Learning. Tab.4 investigates each part of our MLAN method on DomainNet. 'joint' and 'separate' mean the joint alignment branch and the separate alignment branch, respectively. To compare more fairly, we train them with their own pseudo labels generated by source-guided K-means clustering. 'w. categorical' combines the two branches with only the categorical mutual learning objective, and MLAN further adds the logits mutual learning objective.

As shown in Tab.4, MLAN performs 2.2% better than 'joint' and 1.6% better than 'separate'. Also, 'w. categorical' is worse than MLAN but better than both of the two branches. These results show that our mutual learning module is effective. Besides, both the categorical and the logits mutual learning objectives improve the performance, demonstrating that the two branches are complementary on highly-confident samples as well as lowly-confident samples. More analyses of the complementarity between the two branches are shown in the supplementary material.

Sensitivity of Hyperparameters. Fig.2a and Fig.2b show the sensitivity of two hyperparameters α in Eq. (9) and the threshold γ for pseudo labeling. As can be seen, MLAN is not sensitive to α when it varies around 1. The performance decreases when it is 0, illustrating the logits mutual learning for those lowly-confident samples is beneficial. For γ, there is a clear upward trend when γ becomes larger, which means that it is better to select fewer samples for pseudo labeling to avoid wrong labeling.



(a) α (b) γ

Figure 2: Sensitivity analysis experiments of α and γ (All results are reported on the '→ skt' task of DomainNet).

5. Conclusion and Future Work

This work presents a new mutual learning method for MSDA to utilize the complementarity of the joint and separate domain alignment mechanisms. Categorical and logits mutual learning objectives are designed to make the two types of methods learn collaboratively. Extensive experiments demonstrate the superiority of our method. In the future, we will explore the complementarity from more challenging domains with larger domain discrepancies.

Acknowledgement

This work is partially supported the National Key Research and Development Program of China (No. 2017YFA0700800), the Natural Science Foundation of China (Nos. 61772496 and 62122074), and the Beijing Nova Program (Z191100001119123).

References

[1] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 95–104, 2017.

[2] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 742–751, 2017.

[3] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning (ICML)*, pages 1180–1189, 2015.

[4] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, Franois Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research (JMLR)*, 17(1):2096–2030, 2017.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 2672–2680, 2014.

[6] Xiang Gu, Jian Sun, and Zongben Xu. Spherical space domain adaptation with robust pseudo-label loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9101–9110, 2020.

[7] Kaiming He, Xiangyu Zhang, Shaoqi ng Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[8] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[9] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *International Conference on Machine Learning (ICML)*, pages 1989–1998, 2017.

[10] Lanqing Hu, Meina Kan, Shiguang Shan, and Xilin Chen. Duplex generative adversarial network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1498–1507, 2018.

[11] Lanqing Hu, Meina Kan, Shiguang Shan, and Xilin Chen. Unsupervised domain adaptation with hierarchical gradient synchronization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4043–4052, 2020.

[12] J. J. Hull. A database for handwritten text recognition research. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 16(5):550–554, 2002.

[13] 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 (CVPR)*, pages 4893–4902, 2019.

[14] Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1317–1327, 2016.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, volume 25, pages 1097–1105, 2012.

[16] Y. Lecun and L. Bottou. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[17] Da Li and Timothy M. Hospedales. Online meta-learning for multi-source and semi-supervised domain adaptation. In *European Conference on Computer Vision (ECCV)*, pages 382–403, 2020.

[18] Yunsheng Li, Lu Yuan, Yinpeng Chen, Pei Wang, and Nuno Vasconcelos. Dynamic transfer for multi-source domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10998–11007, 2021.

[19] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(12):2935–2947, 2018.

[20] Zhenpeng Li, Zhen Zhao, Yuhong Guo, Haifeng Shen, and Jieping Ye. Mutual learning network for multi-source domain adaptation. *arXiv preprint arXiv:2003.12944*, 2020.

[21] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 469–477, 2016.

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[23] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning (ICML)*, pages 97–105, 2015.

[24] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan. Conditional adversarial domain adaptation. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 1640–1650, 2018.

[25] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 136–144, 2016.

[26] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *International Conference on Machine Learning (ICML)*, pages 2208–2217, 2017.

[27] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 1041–1048, 2008.

[28] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821, 2017.

[29] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. *Workshop*

on Neural Information Processing Systems (NeurIPS Workshop), 2011.

[30] Geon Yeong Park and Sang Wan Lee. Information-theoretic regularization for multi-source domain adaptation. *arXiv preprint arXiv:2104.01568*, 2021.

[31] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 3934–3941, 2018.

[32] 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 (ICCV)*, pages 1406–1415, 2019.

[33] Xingchao Peng, Yichen Li, and Kate Saenko. Domain2vec: Domain embedding for unsupervised domain adaptation. In *European Conference on Computer Vision (ECCV)*, pages 756–774, 2020.

[34] Rajeev Ranjan, Carlos Castillo, and Ramalingam Chellappa. L2 constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*, 2018.

[35] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2017, pages 5533–5542, 2017.

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 91–99, 2015.

[37] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*, pages 213–226, 2010.

[38] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, pages 2988–2997, 2017.

[39] 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 (CVPR)*, pages 3723–3732, 2018.

[40] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3420–3429, 2017.

[41] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European Conference on Computer Vision (ECCV)*, pages 443–450, 2016.

[42] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. In *Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4322–4331, 2019.

[43] 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 (CVPR)*, pages 8725–8735, 2020.

[44] 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 (CVPR)*, pages 2962–2971, 2017.

[45] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[46] Naveen Venkat, Jogendra Nath Kundu, Durgesh Singh, Ambareesh Revanur, and Venkatesh Babu R. Your classifier can secretly suffice multi-source domain adaptation. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 4647–4659, 2020.

[47] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *ACM International Conference on Multimedia (ACM MM)*, pages 1041–1049, 2017.

[48] Hang Wang, Minghao Xu, Bingbing Ni, and Wenjun Zhang. Learning to combine: Knowledge aggregation for multi-source domain adaptation. In *European Conference on Computer Vision (ECCV)*, pages 727–744, 2020.

[49] Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, and Liang Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3964–3973, 2018.

[50] Luyu Yang, Yogesh Balaji, Ser-Nam Lim, and Abhinav Shrivastava. Curriculum manager for source selection in multi-source domain adaptation. *European Conference on Computer Vision (ECCV)*, pages 608–624, 2020.

[51] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7130–7138, 2017.

[52] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4320–4328, 2018.

[53] Han Zhao, Shanghang Zhang, Guanhang Wu, José M. F. Moura, João Paulo Costeira, and Geoffrey J. Gordon. Adversarial multiple source domain adaptation. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 8559–8570, 2018.

[54] Sicheng Zhao, Guangzhi Wang, Shanghang Zhang, Yang Gu, Yaxian Li, Zhichao Song, Pengfei Xu, Runbo Hu, Hua Chai, and Kurt Keutzer. Multi-source distilling domain adaptation. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 12975–12983, 2020.

[55] Y. Zhu, F. Zhuang, and D. Wang. Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources. *AAAI Conference on Artificial Intelligence (AAAI)*, pages 5989–5996, 2019.