Partial Feature Selection and Alignment for Multi-Source Domain Adaptation

Yangye Fu[1], Ming Zhang[1,2], Xing Xu[1]*, Zuo Cao[2], Chao Ma[2], Yanli Ji[1], Kai Zuo[2], and Huimin Lu[3]
[1]Center for Future Media & School of Computer Science and Engineering
University of Electronic Science and Technology of China, China [2]MeiTuan
[3]Department of Mechanical and Control Engineering, Kyushu Institute of Technology, Japan

Abstract

Multi-Source Domain Adaptation (MSDA), which dedicates to transfer the knowledge learned from multiple source domains to an unlabeled target domain, has drawn increasing attention in the research community. By assuming that the source and target domains share consistent key feature representations and identical label space, existing studies on MSDA typically utilize the entire union set of features from both the source and target domains to obtain the feature map and align the map for each category and domain. However, the default setting of MSDA may neglect the issue of "partialness", i.e., 1) a part of the features contained in the union set of multiple source domains may not present in the target domain; 2) the label space of the target domain may not completely overlap with the multiple source domains. In this paper, we unify the above two cases to a more generalized MSDA task as Multi-Source Partial Domain Adaptation (MSPDA). We propose a novel model termed Partial Feature Selection and Alignment (PFSA) to jointly cope with both MSDA and MSPDA tasks. Specifically, we firstly employ a feature selection vector based on the correlation among the features of multiple sources and target domains. We then design three effective feature alignment losses to jointly align the selected features by preserving the domain information of the data sample clusters in the same category and the discrimination between different classes. Extensive experiments on various benchmark datasets for both MSDA and MSPDA tasks demonstrate that our proposed PFSA approach remarkably outperforms the state-of-the-art MSDA and unimodal PDA methods.

1. Introduction

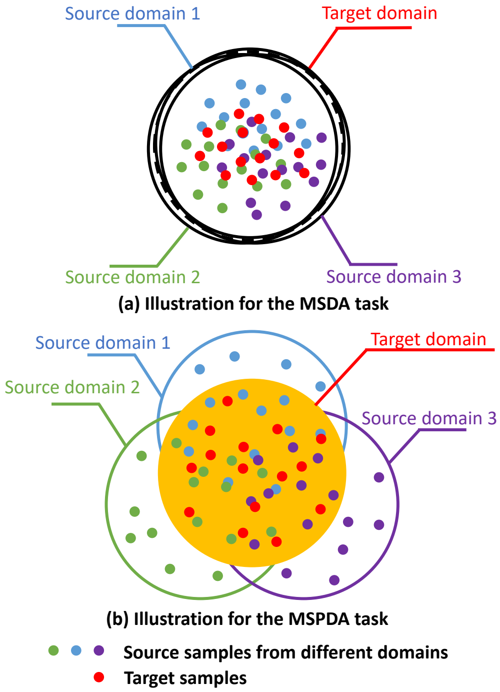
Domain adaptation methods focus on reducing the domain shift [21, 30] between a single labeled source domain and an unlabeled target domain. Recently, a more practical task termed Multi-Source Domain Adaptation (MSDA), which dedicates to transfer the knowledge learned from multiple source domains to an unlabeled target domain, has drawn much attention in the research community.

With MSDA datasets [20, 22], a variety of approaches have been proposed aiming at different application scenarios, *e.g.*, text classification [8], semantic segmentation [33], person re-identification [7], and visual sentiment classification [13]. Recently, different MSDA strategies, such as adversarial learning [24, 32], and source distilling [34], have been proposed to improve the performance on target domain using labeled source domains. However, most of the existing MSDA methods conduct feature alignment on the entire common features, ignoring the fact that some of the source features may not present in the target domain, which may contribute to negative transfer especially when the target domain only shares a part of features with distinct source domains. More practically, the label space of the target domain is unknown, *i.e.*, the label shift between the source and the target domain is ubiquitous.

Although several partial domain adaptation (PDA) researches [2, 3, 4, 31] have reported promising results, very few previous works have paid attention to the situation where multiple source domains are introduced while the target domain does not share the identical label space as the source domains. Thus, we raise a more challenging but practical research topic named *Multi-Source Partial Domain Adaptation (MSPDA)*: Leveraging multiple source domains with distinct label spaces, and perform tasks on the target domain that does not share an identical label space with any specific source domain. As illustrated in Figure 1, we make a comparison between the problem settings of MSDA and MSPDA tasks. In MSDA tasks, all source domains and the target domain share an identical label space, so all circles completely overlap each other. As for MSPDA tasks, any pair of domains do not perfectly share the label space, so some domain-specific labels can be witnessed. In other words, the MSDA task can be considered as a special case of the MSPDA task.

In this paper, we propose a novel end-to-end trainable model termed Partial Feature Selection and Alignment (PFSA) to jointly tackle both MSDA and MSPDA problems. Based on the assumption that all of the target's in-

[*] Corresponding author.



(a) Illustration for the MSDA task

(b) Illustration for the MSPDA task

Figure 1: Illustration of the different settings for the MSDA and MSPDA tasks.

formative features are contained in the union set of multiple source domains, we introduce a selecting vector in PFSA to derive the refined feature map that can reduce the discrepancy between the source and the target features. The selected partial features are highly associated with the target domain and further improve the adaptation performance with three effective feature alignment losses, which are respectively derived from the class-level, domain-level, and discrimination aspects. We conduct extensive experiments on various benchmark datasets to exhibit its superior capability of adapting multiple domains for both MSDA and MSPDA tasks comparing to the state-of-the-art MSDA and PDA approaches.

We summarize our contributions in this paper as follows: (1) We propose a general framework termed Partial Feature Selection and Alignment (PFSA) that is capable of tackling both MSDA and MSPDA problems. (2) We utilize the similarity between the source and the target domain to derive feature selection vectors, aiming at preserving the features that are highly related to the target domain. (3) Three novel feature alignment losses are proposed to further align the selected features, aiming to improve the model's capability of generating discriminative feature representations.

2. Related Work

Multi-Source Domain Adaptation. Due to the lack of variety in single-source domain adaptation and practical demands, multi-source domain adaptation (MSDA) has been raised as a novel research area, which is tougher but more practical and valuable than single-source UDA tasks. With theoretical analysis done by [1, 9, 18], multiple trending strategies have been designed for MSDA tasks, such as adversarial and GAN-based approaches [13, 24, 32, 33]. Latent space learning and domain generation are also applied [17]. Other techniques such as source distilling [34] are proposed to select related source samples as training data. In [8], different distance-based metrics are compared and data samples are chosen dynamically during training according to the correlation between source and target domain. [20] aligns features using high-order moment distance. Class confusion is utilized as a novel metric in [11]. In our work, we dig further into the process of feature alignment and derive partial features that are related to the target domain.

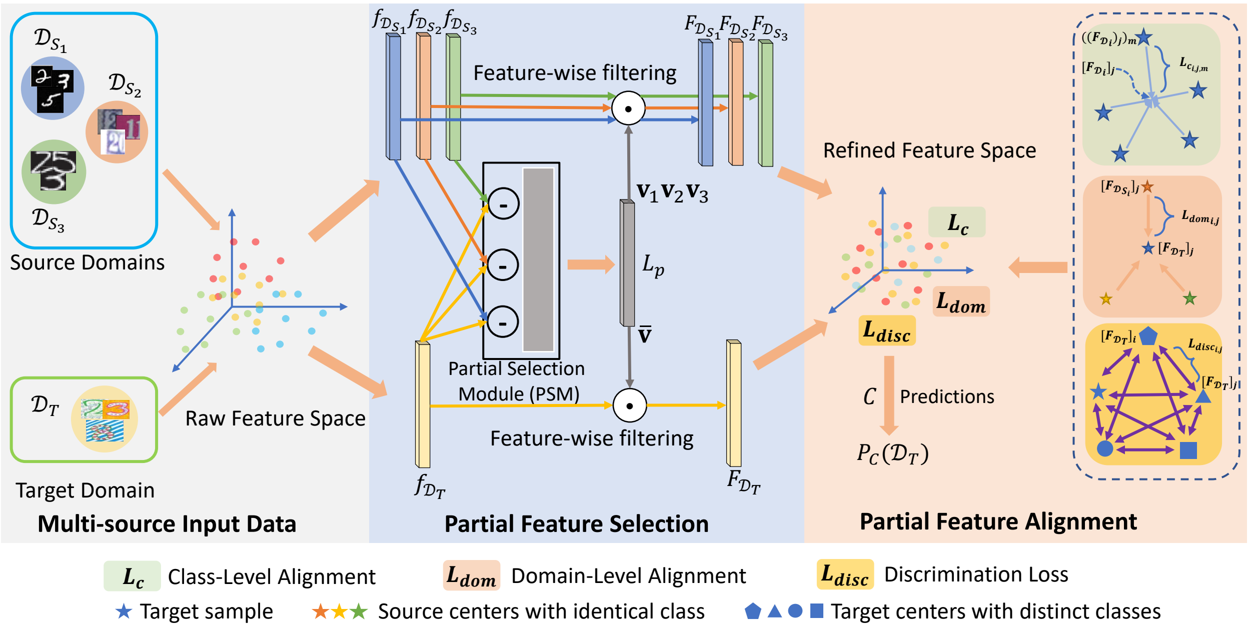
Partial Domain Adaptation. Ordinary domain adaptation stands on the assumption that source and target domains share an identical label space, while practically, the label space of the target domain is unknown, but the abundant source samples can cover the entire label space of the target domain. In recent years, partial domain adaptation (PDA) approaches have been proposed [2, 3, 4, 31], concentrating on tackling situations where the target domain only contains a part of the source labels. Previous works of PDA have been focusing on different aspects, *e.g.*, [2] applies a selective weighting mechanism to multiple adversarial networks and [3] uses one adversarial network and class-level weight to judge source samples. In [31], an auxiliary domain classifier is utilized to derive the possibility that a source sample is contained in the target label space. A domain discriminator is also introduced to qualify the sample transferability and to re-weight source examples [4].

In this paper, we introduce multiple source domains to PDA, aiming at opening a novel topic named Multi-Source Partial Domain Adaptation (MSPDA) to the research community. Note that our MSPDA setting differs from the one proposed in [11], *i.e.*, we don't assume that the source domains share an identical label space.

3. Proposed Method

3.1. Problem Definition

Suppose that there are N source domains $\mathcal{D}_{S_1}$, $\mathcal{D}_{S_2}$, $\mathcal{D}_{S_3}$, $\dots$, $\mathcal{D}_{S_N}$ with label spaces $\mathcal{Y}_{S_1}$, $\mathcal{Y}_{S_2}$, $\dots$, $\mathcal{Y}_{S_N}$ and one target domain $\mathcal{D}_T$ without labels. $f_\mathcal{D}$ represents the feature map of domain $\mathcal{D}$ (extracted by common feature extractors such as ResNet) and $F_\mathcal{D}$ stands for the refined features through our proposed model (note that F is derived through source-target pairs). We use $\mathcal{Y}_T$ to represent the target label space. According to the illustration in Figure 1, the problem settings of the MSDA and MSPDA tasks are as follows: (1) **MSDA**. All domains share an identical label space: $\mathcal{Y}_{S_1} = \mathcal{Y}_{S_2} = \cdots = \mathcal{Y}_{S_N} = \mathcal{Y}_T$. (2) **MSPDA**. Any pair of source domains or source-target pair does not share the label space, but the union set of source domains contains the



$\mathcal{D}_{S_1}$
$\mathcal{D}_{S_2}$
$\mathcal{D}_{S_3}$
Source Domains
$\mathcal{D}_T$
Target Domain
Raw Feature Space
Multi-source Input Data
$f_{\mathcal{D}_{S_1}} f_{\mathcal{D}_{S_2}} f_{\mathcal{D}_{S_3}}$
Feature-wise filtering
$F_{\mathcal{D}_{S_1}} F_{\mathcal{D}_{S_2}} F_{\mathcal{D}_{S_3}}$
$\mathbf{v}_1 \mathbf{v}_2 \mathbf{v}_3$
L_p
$\bar{\mathbf{v}}$
Partial Selection Module (PSM)
Feature-wise filtering
$f_{\mathcal{D}_T}$
$F_{\mathcal{D}_T}$
Partial Feature Selection
Refined Feature Space
L_c
L_{dom}
L_{disc}
C
Predictions
$P_C(\mathcal{D}_T)$
$((F_{\mathcal{D}_i})_j)_m$
$[F_{\mathcal{D}_i}]_j$
$L_{c_{i,j,m}}$
$[F_{\mathcal{D}_{S_i}}]_j$
$L_{dom_{i,j}}$
$[F_{\mathcal{D}_T}]_j$
$[F_{\mathcal{D}_T}]_i$
$L_{disc_{i,j}}$
$[F_{\mathcal{D}_T}]_j$
Partial Feature Alignment
L_c Class-Level Alignment
L_{dom} Domain-Level Alignment
L_{disc} Discrimination Loss
★ Target sample
★★★ Source centers with identical class
⬟▲●■ Target centers with distinct classes

Figure 2: The general framework of the proposed PFSA method, which consists of two parts: partial feature selection and partial feature alignment. A selection vector is firstly derived to refine the extracted feature maps of multiple source domains and target domain samples. Three advanced feature alignment losses are jointly integrated to learn refined feature maps for the adaptation for target domain samples.

entire target label space: $\mathcal{Y}_{S_1} \neq \mathcal{Y}_{S_2} \neq \ldots \neq \mathcal{Y}_{S_N} \neq \mathcal{Y}_T$, $\mathcal{Y}_T \subseteq \bigcup_i^N \mathcal{Y}_{S_i}$.

Our goal is to predict the category labels of the target sample given the extracted features of the labeled samples in the multiple source domains and the unlabeled examples in the target domain.

3.2. Our PFSA Approach

As the overall framework shown in Figure 2, our proposed PFSA model has two major steps: partial feature selection and partial feature alignment. The first step derives filtering vectors for refining the feature map given the extracted features of multiple source domains and the target domain. Then in the second step, three novel feature alignment losses focusing on three different aspects in classification tasks are jointly considered to learn refined feature maps that eliminate the discrepancy of different domains.

3.2.1 Partial Feature Selection

In practice, different pairs of domains share distinct common features. If we compulsively align the features that do not exist in the target domain, the negative transfer will occur. So pair-wise selecting vectors are in demand to refine the features. We use the L_1 distance (to keep the sparsity of output) to represent the similarity between $f_{\mathcal{D}_S}$ and $f_{\mathcal{D}_T}$:

$$\Delta(\mathcal{D}_T, \mathcal{D}_{S_i}) = \left| f_{\mathcal{D}_T} - f_{\mathcal{D}_{S_i}} \right| \in \mathbb{R}^n \tag{1}$$

For convenience, we use Δ_i to represent $\Delta(\mathcal{D}_T, \mathcal{D}_{S_i})$. Ideally, a specific dimension of Δ_i denotes the similarity between $\mathcal{D}_T$ and $\mathcal{D}_{S_i}$ on the feature level, and the smaller the value is, the closer it is to the target. For instance, if the j^{th} dimension of Δ_i, namely $(\Delta_i)_j$, has the minimum value, we infer that the j^{th} dimension of the feature appears to be the most related feature among the source and the target domain. We can select r dimensions with the smallest L_1 distance as the refined features. To extract partial features, a selection vector can be applied to the original feature map (suppose that the dimension of the raw feature space is n):

$$\mathbf{v}_i = [(v_i)_1, (v_i)_2, \ldots, (v_i)_n] \in \mathbb{R}^n, \tag{2}$$

where $\mathbf{v}_i$ is the selecting vector conducted from $\mathcal{D}_T$ and $\mathcal{D}_{S_i}$, $(v_i)_j = 1$ if the j^{th} feature is selected (one of the r dimensions with the smallest values), otherwise, $(v_i)_j = 0$. However, the ability of the trivial version of the method illustrated above is limited, because practically partial features should be conducted through weighted combinations of the raw features, instead of completely sparse selections. So a partial selection module (PSM) that is built on a two-layer fully-connected neural network is utilized to automatically learn and derive the filtering vector of Δ_i, and the refined features of the source domains can be represented as follows:

$$F_{\mathcal{D}_{S_i}} = F(\mathcal{D}_T, \mathcal{D}_{S_i}) = f_{\mathcal{D}_{S_i}} \cdot \mathbf{v}_i, \tag{3}$$

where $\mathbf{v}_i = PSM(\Delta_i)$ is the output of PSM. Note that we take the L_1 distance between the source and the target domain as the input of PSM, which is different from previous approaches, *e.g.*, the domain attention strategy proposed by Wang *et al*. [27] uses the feature map as the input of the global pooling layer. The output of PSM (represented as $\mathbf{v}_i$ for domain $\mathcal{D}_{S_i}$) is regarded as the filtering vector instead of refined features. In the training procedure, we use the average value $\bar{\mathbf{v}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{v}_i$ to filter target domain, while no selecting vector is applied in testing procedure.

To improve the effectiveness of domain adaptation, we use the high-order moment distance proposed in [20] as the loss function of PSM:

$$MD^2(\mathcal{D}_S, \mathcal{D}_T) = \frac{1}{N}\sum_{i=1}^{N}\|\mathbb{E}(F^k_{\mathcal{D}_{S_i}}) - \mathbb{E}(F^k_{\mathcal{D}_T})\|_2 + \binom{N}{2}^{-1}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\|\mathbb{E}(F^k_{\mathcal{D}_{S_i}}) - \mathbb{E}(F^k_{\mathcal{D}_{S_j}})\|_2, \quad (4)$$

where $F_{\mathcal{D}_T} = f_{\mathcal{D}_T}\cdot\bar{\mathbf{v}}$ represents the refined features of the target domain, and $\mathbb{E}(F^k)$ denotes the k-order moment of F. To avoid the zero or infinite trivial solutions, we apply a L_2 regularization to restrict the output of PSM to be close to 1 as:

$$R(\bar{\mathbf{v}}) = \sum_{i=1}^{n}(1-\bar{\mathbf{v}})_i^2. \quad (5)$$

Finally, the partial selection loss can be represented as follows:

$$L_p = MD^2(\mathcal{D}_S, \mathcal{D}_T) + \lambda_{reg}R(\bar{\mathbf{v}}), \quad (6)$$

where λ_{reg} is the trade-off parameter of regularization.

3.2.2 Partial Feature Alignment

To align the features on the refined feature map, we propose three feature alignment loss functions to redistribute the samples of different domains, focusing on three distinct aspects (*i.e.*, domain-level, class-level, and discrimination) in the classification task.

Suppose that n_b represents the batch size, and n_c is the number of samples that belong to class c in a batch. To construct alignment losses, we define the center of class as $[F_{\mathcal{D}_{S_i}}]_c = \frac{1}{n_c}\sum_{j=1}^{n_c}(F_{\mathcal{D}_{S_i}})_j$ (only the refined features of samples that belongs to class c is used in calculating $[F_{\mathcal{D}_{S_i}}]_c$). We use symbol $[F_{\mathcal{D}}]_c$ to denote the center of class c in domain $\mathcal{D}$. Note that the centers are maintained for each category in each domain. For example, if there are 4 categories and 3 domains, 4×3 centers of class will be calculated. For the target domain, we generate pseudo labels to indicate target samples' categories. In order to preserve the information learned from previous batches, the centers are updated through exponential moving average [28]:

$$[F_{\mathcal{D}}]_{c,1} = \frac{1}{n_c}\sum_{j=1}^{n_c}(F_{\mathcal{D}})_{j,1}, \quad (7)$$

$$[F_{\mathcal{D}}]_{c,b} = \beta_c[F_{\mathcal{D}}]_{c,b-1} + (1-\beta_c)\frac{1}{n_c}\sum_{j=1}^{n_c}(F_{\mathcal{D}})_{j,b}, \quad (8)$$

where b denotes the current number of batch and β_c is the trade-off hyper-parameter for formal and novel centers. The centers are maintained every batch.

Considering that the pseudo labels generated from the model are not credible initially, we use self-entropy as a metric to judge the reliability of pseudo labels. For each target sample, we conduct the prediction probabilities of each category (represented as p_i), and compute the self-entropy:

$$H(p) = -\sum_i p_i \log p_i. \quad (9)$$

Then we sort the target samples in a batch according to H and remain only $\alpha\%$ of the target samples in the batch with lower self-entropy (indicating that the model is confident with the prediction of these samples).

Class-Level Alignment Loss. Inspired by the local Relation Alignment Loss (RAL) proposed by [26], we require the samples of the same category to be mapped close to the respective center, and such constraint can be represented in the following form using moment distance:

$$L_c = \sum_{i=1}^{N}\sum_{j=1}^{C}\sum_{m=1}^{n_c}\|\mathbb{E}([F_{\mathcal{D}_{S_i}}]^k_{j,b}) - \mathbb{E}(((F_{\mathcal{D}_{S_i}})_{j,b})^k_m)\| + \sum_{j=1}^{C}\sum_{m=1}^{n_c}\|\mathbb{E}([F_{\mathcal{D}_T}]^k_{j,b}) - \mathbb{E}(((F_{\mathcal{D}_T})_{j,b})^k_m)\|, \quad (10)$$

where $[F_{\mathcal{D}_{S_i}}]_{j,b}$ and $((F_{\mathcal{D}_{S_i}})_{j,b})_m$ represent the center of class, and data sample m, of domain i, class j, at batch b, respectively. $[F_{\mathcal{D}_T}]_{j,b}$ and $((F_{\mathcal{D}_T})_{j,b})_m$ denote the target center of class, and data sample m, of class j at batch b, respectively. $\mathbb{E}(F^k)$ is the k-order moment of F, and N, C, n_c represent the number of source domains, the number of total categories, and the number of samples that belong to class j, respectively. As a consequence, this loss function creates a restriction on data samples that images of the identical category should be aligned correctly to the respective center by the feature extractor.

Domain-Level Alignment Loss. In addition to class-level alignment, it is also expected that the source domains "follow" the route of the target domain, which alleviates the negative influence caused by the misalignment between each pair of the source domain and the target domain. Instead of aligning the entire data samples of the source domains and the target domain, we calculate the L_2 distance between the source centers and the target centers as the domain-level alignment loss:

$$L_{dom} = \sum_{i=1}^{N}\sum_{j=1}^{C}([F_{\mathcal{D}_T}]_{j,b} - [F_{\mathcal{D}_{S_i}}]_{j,b})^2, \tag{11}$$

where $[F_{\mathcal{D}_T}]_{j,b}$ represents the center of class j at batch b in target domain $\mathcal{D}_T$, and $[F_{\mathcal{D}_{S_i}}]_{j,b}$ stands for the center of class j at batch b in source domain $\mathcal{D}_{S_i}$.

Discrimination Loss. Leveraging the centers of the target domain, we construct a metric that describes the degree of dispersion. First, a Euclidean distance is calculated between each pair of the centers for the target domain. Considering that the centers of different categories are required to isolate from each other, we set a hyper-parameter B to control the sparsity of class centroids, *i.e.*, keep the distance between different centers close to B. The discrimination loss can be written as follows:

$$L_{disc} = \sum_{i=1}^{C-1}\sum_{j=i+1}^{C}(B - \|[F_{\mathcal{D}_T}]_{i,b} - [F_{\mathcal{D}_T}]_{j,b}\|_2)^2, \tag{12}$$

where C denotes the total number of categories, $[F_{\mathcal{D}_T}]_{i,b}$ and $[F_{\mathcal{D}_T}]_{j,b}$ are the i^{th} and the j^{th} category's center of the target domain at batch b, respectively. From Eq. 12, the classification error rate can be reduced by isolating target domain samples with different pseudo labels.

3.3. Optimization

For the training procedure, we introduce an auxiliary classifier C_2 to strengthen the robustness of feature extraction and classification by utilizing the adversarial training strategy proposed in [23], *i.e.*, training two classifiers by maximizing the discrepancy of the predictions with a fixed feature extractor, and then updating the parameters of the feature extractor when keeping the two classifiers fixed.

Specifically, the entire PFSA framework introduces three major components, G (general feature extractor), C (C_1 and C_2, the main and the auxiliary classifiers, respectively), and PSM (inferring selecting vector). We first forward the network as illustrated above and calculate the proposed losses $L_p, L_c, L_{dom}, L_{disc}$, note that only the main classifier C_1 is used for gaining pseudo labels of the target domain. The network (G, C and PSM) is updated with the following objective function:

$$\min L_s + \lambda_p L_p + \lambda_c L_c + \lambda_{dom} L_{dom} + \lambda_{disc} L_{disc}, \tag{13}$$

where L_s is the cross-entropy loss of the two classifiers, and $\lambda_p, \lambda_c, \lambda_{dom}, \lambda_{disc}$ are trade-off hyper-parameters of each loss function.

And then the discrepancy between the target predictions of C_1 and C_2, namely $P_{C_1}(\mathcal{D}_T)$ and $P_{C_2}(\mathcal{D}_T)$, is calculated through absolute distance. We expect the two classifiers to enlarge the discrepancy of the predictions and minimize the cross-entropy loss L_s, while the feature extractor reduces the discrepancy during the adversarial training. As the model converges, the discrepancy approaches 0, indicating that the feature extractor successfully derives invariant features with respect to the classifiers. The objective function can be written as follows:

$$\min_C L_s - |P_{C_1}(\mathcal{D}_T) - P_{C_2}(\mathcal{D}_T)|, \tag{14}$$

$$\min_G |P_{C_1}(\mathcal{D}_T) - P_{C_2}(\mathcal{D}_T)|. \tag{15}$$

The entire training procedure of our proposed PFSA method is summarized in Algorithm 1.

Algorithm 1 Training procedure of our proposed PFSA method.

```
Input: data samples $\mathcal{D}_T, \mathcal{D}_{S_1}, \mathcal{D}_{S_2}, \ldots, \mathcal{D}_{S_N}$, hyper-parameters $\lambda_{reg}, \lambda_p, \beta_c, \lambda_c, \lambda_{dom}, \lambda_{disc}, B$.
Output: Model parameters $\theta_G, \theta_{PSM}, \theta_{C_1}, \theta_{C_2}$
repeat
    Generate feature map $f_{\mathcal{D}_T} = G(\mathcal{D}_T)$, $f_{\mathcal{D}_{S_1}} = G(\mathcal{D}_{S_1})$, $f_{\mathcal{D}_{S_2}} = G(\mathcal{D}_{S_2})$, $\ldots$, $f_{\mathcal{D}_{S_N}} = G(\mathcal{D}_{S_N})$.
    Derive refined feature $F_{\mathcal{D}_T}, F_{\mathcal{D}_{S_1}}, F_{\mathcal{D}_{S_2}}, \ldots, F_{\mathcal{D}_{S_N}}$ using Eq. 3.
    Conduct predictions $P_{C_1}(\mathcal{D}_T)$, $P_{C_2}(\mathcal{D}_T)$ and calculate the cross-entropy loss $L_s$ for $C_1$ and $C_2$.
    Predict pseudo labels for the target domain utilizing $P_{C_1}(\mathcal{D}_T)$ and select credible predictions according to Eq. 9.
    Update class centers using Eq. 8.
    Calculate $L_c$, $L_{dom}$, $L_{disc}$, $L_p$ using Eq. 10, 11, 12, 6, respectively.
    Update $\theta_G, \theta_{PSM}, \theta_{C_1}, \theta_{C_2}$ with Eq. 13.
    Regenerate refined feature map $F_{\mathcal{D}_T}$, $F_{\mathcal{D}_{S_1}}$, $F_{\mathcal{D}_{S_2}}$, $\ldots$, $F_{\mathcal{D}_{S_N}}$ with Eq. 3.
    Conduct predictions $P_{C_1}(\mathcal{D}_T)$, $P_{C_2}(\mathcal{D}_T)$ and calculate the cross-entropy loss $L_s$ for $C_1$ and $C_2$.
    Update $\theta_G$, $\theta_{C_1}$, $\theta_{C_2}$ with Eq. 15.
until Reach the maximum iterations or convergence.
```

4. Experiments

4.1. Experimental Setup

Datasets and Features. We perform multiple experiments on three prevailing MSDA datasets to evaluate our model: (1) **Digit-Five** is collected from five different digit classification datasets, including MNIST-M [5], MNIST [12], USPS [10], SVHN [19], and Synthetic Digits [5]. Each domain contains ten classes corresponding to digits ranging from 0 to 9. Specifically, we use the data generated by [20]. In MSPDA scenario, we choose 5 classes out of the entire 10 classes for each source domain and 7 classes for the target domain. (2) **Office-31** [22] is a conventional MSDA dataset containing 4652 images in 31 categories. Images are collected from the office environment and are presented in

Standards	Models	→ **mm**	→ **mt**	→ **up**	→ **sv**	→ **sy**	Avg
Single-Best	DAN (2015) [14]	63.8±0.7	96.3±0.5	94.2±0.9	62.5±0.7	85.4±0.8	80.4
	DANN (2016) [6]	71.3±0.6	97.6±0.8	92.3±0.9	63.5±0.8	85.4±0.8	82.0
	ADDA (2017) [25]	71.6±0.5	97.9±0.8	92.8±0.7	75.5±0.5	86.5±0.6	84.8
Source-Combine	DAN (2015) [14]	67.9±0.8	97.5±0.6	93.5±0.8	67.8±0.6	86.9±0.5	82.7
	DANN (2016) [6]	70.8±0.8	97.9±0.7	93.5±0.8	68.5±0.5	87.4±0.9	83.6
	JAN (2017) [15]	65.9±0.7	97.2±0.7	95.4±0.8	75.3±0.7	86.6±0.6	84.1
	ADDA (2017) [25]	72.3±0.7	97.9±0.6	93.1±0.8	75.0±0.8	86.7±0.6	85.0
	MCD (2018) [23]	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 (2018) [32]	69.5±0.3	98.0±0.9	92.4±0.7	69.2±0.6	87.4±0.5	83.3
	DCTN (2018) [29]	70.5±1.2	96.2±0.8	92.8±0.3	77.6±0.4	86.8±0.8	84.8
	M^{3}SDA(2019) [20]	72.8±1.1	98.4±0.7	96.1±0.8	81.3±0.9	89.6±0.6	87.7
	MDDA (2020) [34]	78.6±0.6	98.8±0.4	93.9±0.5	79.3±0.8	89.7±0.7	88.1
	LtC-MSDA (2020) [26]	85.6±0.8	99.0±0.4	98.3±0.4	83.2±0.6	93.0±0.5	91.8
	PFSA (Ours)	**89.6**±1.2	**99.4**±0.1	**98.6**±0.1	**84.1**±1.1	**95.7**±0.3	**93.5**

Table 1: Experiment results on Digit-Five dataset. **mm**, **mt**, **up**, **sv**, and **sy** represents MNIST-M, MNIST, USPS, SVHN and Synthetic Digits, respectively. The result of maximum accuracy is marked in bold.

three domains: Amazon, Webcam, and DSLR. We choose 21 categories for each source domain and 21 categories for the target domain when performing MSPDA experiments. (3) **DomainNet** [20] is currently the largest and the most challenging dataset in MSDA. It comes with around 0.6 million images in 6 domains: clipart, infograph, painting, quickdraw, real, and sketch. Within each domain, images of 345 categories are collected.

In each experiment, we use symbol "→ **A**" to indicate that the target domain is **A**, while other domains are used as source domains. For fair comparisons, we use the MSDA settings reported in [26]. The choice of categories in MSPDA experiments follows the **MSPDA** setting discussed in Section 3.1.

Implementation Details. As for the parameter settings, we set the four trade-of hyper-parameters $\lambda_p = 0.005, \lambda_c = 0.05, \lambda_{dom} = 0.008, \lambda_{disc} = 1.5 \times 10^{-6}$. β_c is set to 0.5 and $B = 10^3$. Experiments are done under identical settings of hyper-parameters except for the learning rate. In the training procedure of Digit-Five, we set the learning rate to 0.001. As for Office-31 and DomainNet, the learning rate is set to 5×10^{-5} for the pretrained backbone while the rest of the model updates with learning rate 0.001.

Compared Methods and Evaluation Metric. To examine the effectiveness of our model, we take the following MSDA models as our *Multi-Source* baselines: MDAN [32], DCTN [29], M^{3}SDA [20], MDDA [34], and LtC-MSDA [26]. We also conduct *Single-Best* (the best performance of single-source domain adaptation among all source domains) and *Source-Combine* (all source domains are combined as a single source domain) standards to compare our model with typical single-source methods, *e.g.*, DAN [14], DANN [6], JAN [15], ADDA [25], and MCD [23].

As for MSPDA experiments, we migrate MSDA models MDAN [32], DCTN [29], M^{3}SDA [20], LtC-MSDA [26] and PDA models PADA [3], SAN [2], ETN [4] to MSPDA setting as our MSPDA baselines. Specifically, few changes are needed for MSDA models other than the label spaces among source domains and target domain are not shared anymore. While for PDA models, we combine all the source domains as one single domain to adapt to the conventional PDA settings. We do not conduct single-best criterion as MSDA settings, because a single source domain in MSPDA does not contain the entire label space of the target, which doesn't fit the conventional problem setting of PDA.

Standards	Models	→ D	→ W	→ A	Avg
Single-Best	DAN (2015) [14]	99.0	96.0	54.0	83.0
	ADDA (2017) [25]	99.4	95.3	54.6	83.1
Source-Combine	DAN (2015) [14]	98.8	96.2	54.9	83.3
	JAN (2017) [15]	99.4	95.9	54.6	83.3
	ADDA (2017) [25]	99.2	96.0	55.9	83.7
	MCD (2018) [23]	99.5	96.2	54.4	83.4
Multi-Source	MDAN (2018) [32]	99.2	95.4	55.2	83.3
	DCTN (2018) [29]	99.6	96.9	54.9	83.8
	M^{3}SDA (2019) [20]	99.4	96.2	55.4	83.7
	MDDA (2020) [34]	99.2	97.1	56.2	84.2
	LtC-MSDA (2020) [26]	99.6	97.2	56.9	84.6
	PFSA (Ours)	**99.7**	**97.4**	**57.0**	**84.7**

Table 2: Results on Office-31 dataset. A, W, and D stands for domain Amazon, Webcam, and DSLR, respectively. The best results are marked in bold.

4.2. Overall Comparison on MSDA Task

Results on Digit-Five Dataset. Experiment results on Digit-Five is reported in Table 1. According to Table 1, our model achieves an average accuracy of 93.5%, which outperforms current MSDA approaches by a large margin. Especially, a performance gain of 4% is presented in "→ **mm**" task. Notice that we also get some slight but steady performance gain on tasks where accuracies are relatively high, which indicates our model's efficiency in the circumstance where pseudo labels are approximately reliable.

Results on Office-31 Dataset. According to Table 2, the result is much lower in "→ A" task than others, mainly because domain DSLR (D) and Webcam (W) are highly similar but differ from Amazon (A), which may not meet our assumption that the union set of source domain features contains the target's features. In other words, only a part of target (Amazon) features are aligned with source domains (DSLR and Webcam), and some key features of the target domain may be ignored by the model because they do not present in source features.

Results on DomainNet Dataset. In Table 3, we report our results on DomainNet. Generally, a performance gain of

Standards	Models	→ clp	→ inf	→ pnt	→ qdr	→ rel	→ skt	Avg
Single-Best	DAN (2015) [14]	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
	DANN (2016) [6]	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
	JAN (2017) [15]	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
	ADDA (2017) [25]	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 (2018) [23]	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	DAN (2015) [14]	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
	DANN (2016) [6]	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
	JAN (2017) [15]	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
	ADDA (2017) [25]	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 (2018) [23]	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 (2018) [32]	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
	DCTN (2018) [29]	48.6±0.7	23.5±0.6	48.8±0.6	7.2±0.5	53.5±0.6	47.3±0.5	38.2
	M^{3}SDA (2019) [20]	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 (2020) [34]	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
	LtC-MSDA (2020) [26]	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
	PFSA (Ours)	**64.5**±0.8	**29.2**±0.8	**57.6**±0.5	**17.2**±0.6	**67.2**±0.6	**55.1**±0.7	**48.5**

Table 3: Results on DomainNet. clp, inf, pnt, qdr, rel, skt reprecents clipart, infograph, painting, quickdraw, real, sketch, respectively. Results of maximum accuracy scores are marked in bold.

Standards	Models	Digit-Five						Office-31			
		→mm	→mt	→up	→sv	→sy	Avg	→D	→W	→A	Avg
MSDA	MDAN (2018) [32]	55.4	79.8	73.1	35.4	43.1	57.4	74.2	71.9	28.6	58.2
	M3SDA (2019) [20]	69.3	98.0	96.3	47.8	78.3	77.9	78.2	71.0	32.9	60.7
	LtC-MSDA (2020) [26]	60.0	97.6	97.7	43.5	83.7	76.5	83.3	76.2	31.0	63.5
PDA	PADA (2018) [3]	63.9	90.4	93.1	40.5	62.8	70.1	75.6	73.8	38.0	62.5
	SAN (2018) [2]	55.4	96.1	96.5	32.6	55.1	67.1	83.1	77.8	39.8	66.9
	ETN (2019) [4]	48.7	93.6	93.4	36.6	64.6	67.4	83.9	78.5	**41.1**	**67.8**
MSPDA	**PFSA (Ours)**	**69.5**	**98.2**	**99.0**	**68.7**	**86.9**	**84.5**	**84.2**	**79.0**	40.2	**67.8**

Table 4: Overall comparison of our PFSA approach and state-of-the-art MSDA and PDA approaches in MSPDA task.

1.1% is produced by our model. In particular, we achieve an accuracy of 64.5% and 55.1% on task "→ clp" and "→ skt", respectively, outperforming existing approaches by a large margin. According to the results, the dataset is quite challenging, especially on "→ qdr" task. We think the dataset is difficult due to the following reasons: First, DomainNet contains 345 categories of images in each domain, which is much more than any other MSDA dataset. The large number of categories makes it difficult for feature extractors to derive unique features, and it is harder for classifiers to discern certain samples. Second, significant distribution shift presents from domain to domain, especially among quickdraw and others. Such distribution shift increases the difficulty of refining informative features since misalignment can occur when pseudo labels are not credible.

4.3. Overall Comparison on MSPDA Task

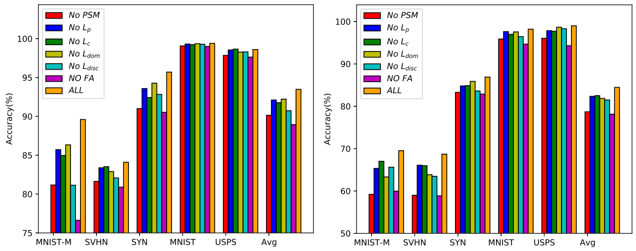
The results with the experimental setting of MSPDA problem on Digit-Five and Office-31 are presented in Table 4. It can be found that our proposed method performs significantly better than all MSDA methods on both datasets. Besides, our approach outperforms the existing PDA models by a large margin on Digit-Five and reaches SOTA on Office-31. Specifically, compared with MSDA baselines, our method produces a performance gain of 6.6% on Digit-Five dataset and a performance gain of 4.3% on Office-31 dataset. In comparison with PDA approaches, a performance gain of 14.4% is presented on Digit-Five. Dramatic drops in performance can be witnessed compared with conventional MSDA or PDA settings, we think this is because other methods only take either *multi-source domain adaption* or *partial domain adaption* into consideration, while our approach is capable of handling both scenarios.

4.4. Further Analysis

Effect of the Key Components. To further interpret the efficiency of each component of partial feature extracting strategy and alignment losses, we conduct ablation study on Digit-Five and the result is presented in Figure 3.

By reducing the loss terms used in the model, different degrees of performance drop can be witnessed according to Figure 3. Particularly, a significant performance drop is presented in situation *No FA*, i.e., all three alignment losses are removed from our model, either in MSDA or MSPDA. It demonstrates that the alignment mechanism makes a great contribution to the performance of our method. Other than the alignment losses, the PSM structure also plays a vital role, especially in MSDA scenario where"→ **mm**" task and "→ **sv**" task suffer a great loss without PSM. We state that the proposed structures contribute in different aspects to the overall task, and the combination of all loss functions reaches the best performance on all five tasks.

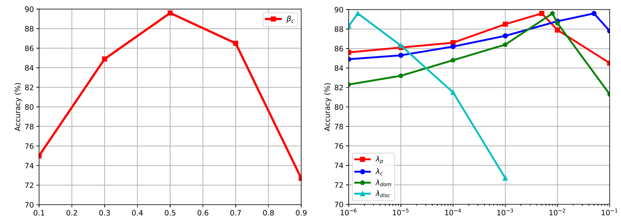
Parameter Sensitiveness Analysis. Furthermore, we investigate the effect of the hyper-parameters β_c in Eq. 8 and λ_p, λ_c, λ_{dom}, λ_{disc} in Eq. 13. In this experiment, we set the numerical range of β_c in the range $[0.1, 0.9]$ and increase it by step. For the hyper-parameters λ_*, we set their range in $[10^{-6}, 10^{-1}]$. We change the value of one specific parameter and fix the others in each experiment. The sensitivity analysis of the five parameters of PFSA on task "→ **mm**" is shown in Figure 4. In particular, the optimal value for β_c is 0.5. As the value gets lower, the accuracy drops, indicating that placing too much reliance on novel centers will bring more instability, which does harm to the performance. If β_c is too high, *e.g.*, 0.9, the performance also drops, since the centers update slowly with high β_c, and the model may ignore some crucial information from novel batches in this case. Furthermore, the optimal values for $\lambda_p, \lambda_c, \lambda_{dom}, \lambda_{disc}$ are $0.005, 0.05, 0.008$, and 1.5×10^{-6}, respectively. Thus it indicates that the loss terms behind the four hyper-parameters have different contributions. When the values of the four hyper-parameters are too large ($e.g.$, lager than 0.1), some performance drops are presented due to the ignorance of the conventional cross-entropy loss for classification tasks. As the four hyper-parameters get too low (less than 10^{-6}), nevertheless, the performance generally falls as the impacts of these loss terms become trivial. In practice, we can efficiently search for suitable settings of hyper-parameters on the validation set for different application scenarios.



(a) Effect of different components in MSDA task.

(b) Effect of different components in MSPDA task.

Figure 3: Ablation study of PFSA in MSDA and MSPDA tasks on Digit-Five dataset.

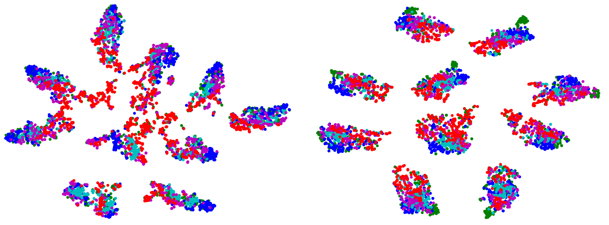


(a) β_c

(b) λ_p, λ_c, λ_{disc} and λ_{dom}

Figure 4: Sensitivity analysis on the five parameters.

Visualization of Selected Features Subspace. To illustrate our method's capability of selecting the features associated with the target domain and aligning the target domain to source domains, we visualize the feature spaces of the source domains and the target domain using t-SNE [16] feature embedding when executing "→ **sv**" task. The visualization result is presented in Figure 5. As Figure 5(a) demonstrates, some outlier target samples can be witnessed, indicating that the model without PFSA can confuse with the domain and the category information. Without PFSA, the misalignment among the target and the source domains occurs, which contributes to the presence of discrete target features. In Figure 5(b), nevertheless, clear clusters are generated through our proposed PFSA approach in the universal feature space and target features are well-aligned with all source samples within the identical category. The centers of different categories are discernable after PFSA refinement and the outlier samples are filtered out, which demonstrates that partial features are correctly extracted and aligned through our PFSA strategy. Besides, the cluster centers of the target domain are also closer to those of the source domains, indicating the effectiveness of our PFSA approach in domain adaptation.



(a) Without PFSA

(b) With PFSA

Figure 5: Feature space t-SNE [16] visualization of the model trained with and without PFSA in "→ **sv**" task. The target domain SVHN is marked in red and the source domains are presented in other colors.

5. Conclusion

In this paper, we proposed a novel Partial Feature Selection and Alignment (PFSA) scheme to refine and align the feature map extracted by conventional feature generators. We regarded the L_1 distance between the source and target features as the similarity among features, and selecting vectors are derived through partial selection module (PSM) for each pair of source and target domain using L_1 distance as the input. The selecting vectors are applied to the original feature map to conduct a more informative feature space. Three alignment losses are calculated on the basis of the novel feature space, concentrating on three different requirements in classification tasks. Extensive experiments on MSDA and MSPDA tasks show that our model is capable of tackling both distribution shift and label shift problems.

6. Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under grants No. 61976049, 61632007, U20B2063; the Fundamental Research Funds for the Central Universities under Project ZYGX2019Z015 and the Sichuan Science and Technology Program 2018GZDZX0032, 2019ZDZX0008 and 2019YFG0003.

References

[1] John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 129–136, 2007. 2

[2] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Michael I. Jordan. Partial transfer learning with selective adversarial networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 2724–2732, 2018. 1, 2, 6, 7

[3] Zhangjie Cao, Lijia Ma, Mingsheng Long, and Jianmin Wang. Partial adversarial domain adaptation. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII*, pages 139–155, 2018. 1, 2, 6, 7

[4] Zhangjie Cao, Kaichao You, Mingsheng Long, Jianmin Wang, and Qiang Yang. Learning to transfer examples for partial domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2985–2994, 2019. 1, 2, 6, 7

[5] Yaroslav Ganin and Victor S. Lempitsky. Unsupervised domain adaptation by backpropagation. In Francis R. Bach and David M. Blei, editors, *International Conference on Machine Learning, ICML*, volume 37, pages 1180–1189, 2015. 5

[6] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17:59:1–59:35, 2016. 6, 7

[7] Yixiao Ge, Dapeng Chen, and Hongsheng Li. Mutual mean-teaching: Pseudo label refinery for unsupervised domain adaptation on person re-identification. In *International Conference on Learning Representations, ICLR*, 2020. 1

[8] Han Guo, Ramakanth Pasunuru, and Mohit Bansal. Multi-source domain adaptation for text classification via distancenet-bandits. In *AAAI Conference on Artificial Intelligence, AAAI*, pages 7830–7838, 2020. 1, 2

[9] Judy Hoffman, Mehryar Mohri, and Ningshan Zhang. Algorithms and theory for multiple-source adaptation. In *Advances in Neural Information Processing Systems*, pages 8256–8266, 2018. 2

[10] Jonathan J. Hull. A database for handwritten text recognition research. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16:550–554, 1994. 5

[11] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum class confusion for versatile domain adaptation. In *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXI*, volume 12366, pages 464–480. Springer, 2020. 2

[12] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2324, 1998. 5

[13] Chuang Lin, Sicheng Zhao, Lei Meng, and Tat-Seng Chua. Multi-source domain adaptation for visual sentiment classification. In *AAAI Conference on Artificial Intelligence, AAAI*, pages 2661–2668, 2020. 1, 2

[14] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning, ICML*, volume 37, pages 97–105, 2015. 6, 7

[15] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *International Conference on Machine Learning, ICML*, volume 70, pages 2208–2217, 2017. 6, 7

[16] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 8

[17] Massimiliano Mancini, Lorenzo Porzi, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. Boosting domain adaptation by discovering latent domains. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 3771–3780, 2018. 2

[18] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems*, pages 1041–1048, 2008. 2

[19] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *Advances in Neural Information Processing Systems Workshops*, 2011. 5

[20] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *IEEE/CVF International Conference on Computer Vision, ICCV*, pages 1406–1415, 2019. 1, 2, 4, 5, 6, 7

[21] Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009. 1

[22] Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision*, pages 213–226, 2010. 1, 5

[23] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 3723–3732, 2018. 5, 6, 7

[24] Alice Schoenauer Sebag, Louise Heinrich, Marc Schoenauer, Michèle Sebag, Lani F. Wu, and Steven J. Altschuler. Multi-domain adversarial learning. In *International Conference on Learning Representations, ICLR*, 2019. 1, 2

[25] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2962–2971, 2017. 6, 7

[26] Hang Wang, Minghao Xu, Bingbing Ni, and Wenjun Zhang. Learning to combine: Knowledge aggregation for multi-source domain adaptation. *CoRR*, abs/2007.08801, 2020. 4, 6, 7

[27] Xudong Wang, Zhaowei Cai, Dashan Gao, and Nuno Vasconcelos. Towards universal object detection by domain attention. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 7289–7298, 2019. 4

[28] Shaoan Xie, Zibin Zheng, Liang Chen, and Chuan Chen. Learning semantic representations for unsupervised domain adaptation. In Jennifer G. Dy and Andreas Krause, editors, *International Conference on Machine Learning, ICML*, volume 80, pages 5419–5428, 2018. 4

[29] Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, and Liang Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 3964–3973, 2018. 6, 7

[30] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, pages 3320–3328, 2014. 1

[31] Jing Zhang, Zewei Ding, Wanqing Li, and Philip Ogunbona. Importance weighted adversarial nets for partial domain adaptation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8156–8164, 2018. 1, 2

[32] Han Zhao, Shanghang Zhang, Guanhang Wu, José M. F. Moura, João Paulo Costeira, and Geoffrey J. Gordon. Adversarial multiple source domain adaptation. In *Advances in Neural Information Processing Systems*, pages 8568–8579, 2018. 1, 2, 6, 7

[33] Sicheng Zhao, Bo Li, Xiangyu Yue, Yang Gu, Pengfei Xu, Runbo Hu, Hua Chai, and Kurt Keutzer. Multi-source domain adaptation for semantic segmentation. In *Advances in Neural Information Processing Systems, NeurIPS*, pages 7285–7298, 2019. 1, 2

[34] 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. 1, 2, 6, 7