Multi-Source Unsupervised Domain Adaptation via Pseudo Target Domain

Chuan-Xian Ren, Yong-Hui Liu, Xi-Wen Zhang, Ke-Kun Huang

Abstract—Multi-source domain adaptation (MDA) aims to transfer knowledge from multiple source domains to an unlabeled target domain. MDA is a challenging task due to the severe domain shift, which not only exists between target and source but also exists among diverse sources. Prior studies on MDA either estimate a mixed distribution of source domains or combine multiple single-source models, but few of them delve into the relevant information among diverse source domains. For this reason, we propose a novel MDA approach, termed Pseudo Target for MDA (PTMDA). Specifically, PTMDA maps each group of source and target domains into a group-specific subspace using adversarial learning with a metric constraint, and constructs a series of pseudo target domains correspondingly. Then we align the remainder source domains with the pseudo target domain in the subspace efficiently, which allows to exploit additional structured source information through the training on pseudo target domain and improves the performance on the real target domain. Besides, to improve the transferability of deep neural networks (DNNs), we replace the traditional batch normalization layer with an effective matching normalization layer, which enforces alignments in latent layers of DNNs and thus gains further promotion. We give theoretical analysis showing that PTMDA as a whole can reduce the target error bound and leads to a better approximation of the target risk in MDA settings. Extensive experiments demonstrate PTMDA’s effectiveness on MDA tasks, as it outperforms state-of-the-art methods in most experimental settings.

Index Terms—Unsupervised Domain Adaptation, Pseudo Target Domain, Feature Extraction, Batch Normalization, Matching Normalization Layer.

I. INTRODUCTION

Deep neural networks (DNNs) are powerful at extracting features from structured data [1], [2] such as image, speech, and video, and it significantly outperforms traditional machine learning algorithms in image processing and classification. However, these remarkable gains often rely on the availability of large amounts of labeled training samples [3], which limits the utility of DNNs in situations where sample labeling is prohibitively expensive. One natural idea to address the issue is to generalize the model from a labeled dataset to the unlabeled dataset [4], [5]. However, if the distribution of the labeled dataset differs from that of the unlabeled dataset, the model cannot be generalized well. Unsupervised domain adaptation (UDA) [6]–[8] attempts to address this domain shift issue, i.e. the fact that there is a distribution bias between the training and test datasets in many practical applications.

Great efforts [7], [9]–[12] have been devoted to the UDA literatures. While existing works mainly focus on tasks with single source domain, it is common in practice that labeled samples are obtained from multiple sources with diverse distributions. The task with such multiple source domains is known as multi-source unsupervised domain adaptation (MDA) [13], [14]. For instance, we want to predict the category of some photos, which can be taken as the target domain. We can search from the Internet for labeled images such as paintings, cliparts, or sketches. Those images can be taken as multiple sources, which exhibit significant difference in texture or visual style. A straightforward strategy is to combine these source domains into a single domain, but it leads to a sub-optimal model since gaps among multiple source domains are omitted. This practical issue motivates the study on MDA, which aims at learning a prediction model from multiple sources and generalizing it to a different yet related target domain.

Recently, based on the distribution weighting rule of Mansour et al. [15], mixed distributions [16]–[18] from multiple sources are designed and theoretical results [19]–[21] are derived. Therefore, combination-based MDA methods have gained popularity. These methods first train source-specific models corresponding to each pair of source and target domains by minimizing the distribution discrepancy [17], [22],
latent layers of the feature extraction network. Then, a weight vector is derived to combine the predictions for the target samples. While previous methods have already achieved competitive results on MDA, few of them delve into the relevance among diverse source domains with abundant discriminative information. Moreover, it is still important to improve the transferability of the latent layers of DNNs. Although related works [25]–[27] using variants of the Batch Normalization (BN) layers [28] have been raised, these methods achieve only limited effectiveness.

In this paper, we propose a new concept of Pseudo Target Domain to deal with MDA problems, and thus call our method PTMDA for short. The main difference between PTMDA and traditional MDA methods is shown in Fig. 1. Traditional methods (left) focus on aligning each source domain with the real target domain. Notice that while the aligning process mapping target domain and a source domain into a shared space, distribution gap between domains is reduced and the two domains are drew closer in the mapping space. It motivates us to combine the two mapped domains after the alignment (right). The pseudo target domain, which incorporates labeled samples into the unlabeled target, can be served as a new informative target to the other source domains. With the construction of pseudo target domain, PTMDA not only reduces domain shift between source and target domains, and, more importantly, it is capable to efficiently utilize discriminative information from source domain to promote downstream alignments with other source domains. Besides, we design a simple yet effective matching normalization (MN) layer to further improve the generalization ability of DNNs. We propose to update the affine parameters with gradient information only from the target domain in the training stage.

This special layer aligns the distributions of source domain and target domain in the latent layers of the feature extraction network. Our main contributions are summarized as follows.

1) We construct a series of pseudo target domains by performing the adversarial training strategy on the real target domain and each source domain. This approach conveys useful information from each source domain to the target domain, and thus is helpful to the alignment of the pseudo target domain with the remainder source domains. The method takes advantage of multiple source knowledge in MDA tasks and promotes the generalization performance on the real target domain.

2) We introduce the MN layer, which aligns the distributions of different domains in the latent network layers to improve the transfer ability of DNNs. The proposed MN layer is generic and can be embedded into many deep domain adaptation methods.

3) PTMDA achieves state-of-the-art classification performance for MDA tasks on five benchmark datasets. In particular, the average accuracy of MDA tasks increases by 1.2% on Office-Caltech10 and increases by 0.8% on ImageCLEF-DA.

The remainder of this paper is organized as follows. In Section II, we briefly review some related works. Details of the PTMDA method are given in Section III. Section IV presents theoretical analysis for the effectiveness of PTMDA. Section V provides comprehensive experimental results for validating the effectiveness of PTMDA. Section VI concludes the paper.

II. RELATED WORK

In this section, we briefly review some recently proposed and related works of UDA and MDA.

A. Unsupervised Domain Adaptation

Most recent UDA methods are motivated by the results of Ben-David et al. [29], which exploited the $\mathcal{A}$-distance to estimate the discrepancy of distributions between source and target domains. Blitzer et al. [30] deduced a uniform convergence learning bound, which minimizes the convex combination of empirical risks among diverse domains. By virtue of these foundation works, minimizing the domain discrepancy has been extensively used for UDA. Hu et al. [5] proposed deep transfer metric learning and constrain the local manifold to enhance the discrimination ability of representations. Saito et al. [31] proposed to cluster the neighboring target data using self-supervision information to learn discriminative features. Maximum mean discrepancy (MMD) is an important statistic used to reduce distribution shift in various methods [11]. Ren et al. [8] exploited low-rank representation to alleviate the distribution shift and stress the group compactness of features. Some other methods [32], [33] used the idea of optimal transportation to measure the domain discrepancy. Luo et al. [34] proposed a discriminative manifold propagation (DMP) framework to improve the model’s generalization ability on the target domain. Pandey et al. [35] proposed to find the ‘closest-clone’, which is a source image that arbitrarily close to the test image from the target data, and train the domain-adaptive classifier using the clones.

While many methods discussed above were proposed based on distribution discrepancy, there are several other works using the technique of adversarial training. Ganin et al. [6] introduced domain adversarial neural network (DANN) to align the feature distributions. Tzeng et al. [36] used separate feature networks for diverse domains and train the target domain adversarially. Carlucci et al. [37] and Kurmi et al. [38] used multi-class discriminator to improve the performance of adversarial domain adaptation. Satio et al. [7] proposed the maximum classifier discrepancy (MCD), which utilizes two classifiers to simulate a domain discriminator. Zhang et al. [9] designed a symmetric architecture to perform domain confusion both on category-level and domain-level. Pei et al. [39] used specialized domain discriminators for each class to enable fine-grained adaptation among various domains.

The method of pseudo labeling was also employed in some works since it is believed that the construction of pseudo labels can improve the discriminative ability of the unlabeled data. Saito et al. [40] used three asymmetric classifiers to assign robust pseudo labels to unlabeled data. Zhang et al. [41] proposed a collaborative and adversarial network (CAN) to iteratively refine the quality of pseudo labels and then extend the network with a self-paced learning scheme [42]. Deng et
al. [43] used a similarity guided constraint to progressively select pseudo labels. In our work, we will use a confidence threshold to eliminate the unreliable pseudo labels.

B. Multi-source Unsupervised Domain Adaptation

Several works have been devoted to the theoretical study for MDA. Mansour et al. [15] provided theoretical analysis under the assumption that the target domain is a convex combination of source domains. Hoffman et al. [20] considered an extension of the theory of Mansour to derive the mixture parameter. Zhao et al. [19] introduced new error bounds for both regression and classification tasks. Redko et al. [21] used a Wasserstein distance-based error function to reformulate the joint hypothesis estimation of MDA task. Wen et al. [44] derived a finite-sample error bound based on the theory of Mansour et al. [15].

There are other works on specific algorithms. Hoffman et al. [22] introduced a domain transform framework, which uses a cluster approach to discover the latent domains. Xu et al. [16] proposed the deep cocktail network (DCTN) using multi-way domain adversarial learning. Peng et al. [23] defined a cross-moment divergence to enforce alignment between each pair of domains. Zhao et al. [18] designed a novel weighting strategy to combine diverse classifiers. The recently proposed MFSAN [17] first aligns the domain-specific distribution and then matches domain-specific classifiers. These works either estimate a mixed distribution [16]–[18] or combine multiple single-source models [20], [22]–[24]. Carlucci et al. [37] used a Hallucinator block to remove the domain-specific style across various source domains.

In this paper, we propose a novel method to deal with MDA problems. Our PTMDA method is different from existing MDA algorithms in the sense that, PTMDA exploits structured and relevant information among source domains in addition to improve generalization on the target domain, and improves the transferability of the intermediate layers of DNNs to mitigate the domain shift among multiple domains.

III. LEARNING WITH PSEUDO TARGET DOMAIN

In this section, we first present general MDA problems, and then describe the construction of pseudo target domain and the MN layers. Finally, we present the PTMDA method in details.

A. Problem Formulation

Without loss of generality, we consider a C-class problem. Let $\mathcal{X}$ denote the input space and $\mathcal{Y}$ denote the output space, where $\mathcal{Y} = \{1, \ldots , C\}$. We define a domain $D$ enclosed with a distribution $P$ and a labeling function $f : \mathcal{X} \rightarrow \mathcal{Y}$. We consider the adaptation problem with $N$ source domains $\{D_s^i\}_{i=1}^N$ and one target domain $D_t$. In the $i$-th source domain $D_s^i$, $X_s^i = \{x_{s_i}^k\}_{k=1}^{N_{s_i}}$ and $Y_s^i = \{y_{s_i}^k\}_{k=1}^{N_{s_i}}$ denote the observed data and corresponding labels sampled from the source distribution $P_s^i$, i.e., $(x_{s_i}^k, y_{s_i}^k) \sim P_s^i$. Let $X_t = \{x_t^k\}_{k=1}^N$ be the observed target data sampled from the target distribution $P_t$, and $Y_t = \{y_t^k\}_{k=1}^N$ be the unknown target labels, i.e., $(x_t^k, y_t^k) \sim P_t$. In the setting of MDA, there are two basic assumptions: (1) samples from different domains share the same output space, i.e., $y_{s_i} \in \mathcal{Y}$, $y_{t} \in \mathcal{Y}$, and (2) domains have different distributions due to domain shift, i.e., $P_{s_i} \neq P_{s_j}$, $P_{s_i} \neq P_{t}$, $\forall i, j \in \{1, 2, \cdots , N\}$. The goal of MDA is to predict the labels of $X_t$ by exploiting the information of $\{(X_{s_i}, Y_{s_i})\}_{i=1}^N$ and $X_t$.

B. Pseudo Target Domain: Construction and Alignment

Our method not only aims at reducing the domain shift between each group of source and target domains, but also seeks to extract discriminative knowledge from multiple sources to enrich the target domain. Without loss of generality, we consider $D_{s_i}, D_{s_j},$ and $D_{t}$ as an example, where $i, j \in \{1, \ldots , N\}$ and $i \neq j$. The basic motivation of pseudo target domain construction has been shown in Fig. 2. Generally, in domain adaptation, one can regularize the knowledge transfer model to the unlabeled target samples with available labeled source samples. It motivates us to mimic a new target domain, which is called pseudo target domain in this work, to include both labeled data from $D_{s_i}$ and unlabeled data from $D_{t}$. In this way, the labeled data in the pseudo target domain can provide reliable supervision for network feature learning. The reasonability of pseudo target domain construction can be summarized as follows.

- Domain shift exists not only between $D_{t}$ and $D_{s_i}$ but also across $D_{s_i}$ and $D_{s_j}$, thus, it is natural to treat $D_{s_i}$ as a general target of $D_{s_j}$, and $D_{s_j}$ can be used to mimic a new and enlarged (with $D_{t}$) target domain w.r.t. $D_{s_i}$.
- In the alignment of target and source domains, samples from two domains are mapped to a shared space and drew closer. Consequently, it is rational to combining $D_{s_j}$ and $D_{t}$ after the alignment, which leads to the construction of the new pseudo target domain.

In summary, the PTMDA method is consisted of two stages. 1) For each pair of source domain $D_{s_i}$ and target domain $D_{t}$, we initialize a deep network using the adversarial training strategy with a metric constraint to reduce domain shift, and then combine the two mapped domains into a pseudo target domain $D_{s_j,t}$. Samples in $D_{t}$ are equipped with pseudo-labels in this stage. 2) To utilize discriminative and structured information across multiple source domains, each pseudo target
domain generated in the first stage is aligned to each of the
remainder source domains separately.

In these process, each pair of source and target domains
is chosen in turn to construct the pseudo target domain and
then fed into the second stage. Thus, while domain difference
between source and the target in the first stage has been
minimized and has little effect on the second stage relatively,
the second stage can take turns to align among source domains
and utilize diverse distributed information from different sources.

1) Stage 1: Pseudo Target Domain Construction. In this
stage, the adversarial training manner is used to align the
distributions of each pair of source and target domains as
much as possible. Then, the pair of domains are combined
as a pseudo target domain, consisting of samples from both
labeled source domain and pseudo-labeled target domain.

The model consists of a feature extraction network \( G \) with
parameters \( \theta_G \), \( N \) category classifiers \( \{C_i\}_{i=1}^N \) with
parameters \( \theta_{C_i} \), and \( N \) domain discriminators \( \{D_i\}_{i=1}^N \) with
parameters \( \theta_{D_i} \). Each group of \( G, C_i, D_i \) addresses a
batch of samples drawn from \( \{X_{si}, Y_{si}\}, X_{tj}\). 

As illustrated in Fig. 3, each group of source domain
and target domain, e.g., \( \{D_{si}, D_p\} \) or \( \{D_{sj}, D_t\} \), is used to
construct a pseudo target domain. Specifically, for the
pair of \( \{D_{si}, D_t\} \), the feature extractor \( G \) maps \( D_{si} \) and
\( D_t \) into a shared latent space. It learns the feature mappings
from different domains with a single generator network. The
classifier \( C_i \) predicts the category for the input samples from
\( \{D_{si}, D_p\} \), and the domain discriminator \( D_i \) supervises the
learning process of \( G \) towards the direction which learns
domain invariant features with respect to \( \{D_{si}, D_t\} \).

In the training of feature extractor \( G \) and discriminators \( D_i \),
the feature distributions of \( \{(X_{si}, Y_{si})\}_{i=1}^N \) and \( X_t \) are aligned
using the following adversarial loss, i.e.,
\[
L_{adv,G} = -\mathbb{E}_{x_{si}, y_{si}} \log(1 - D_j(\varphi(G(x_{si}), y_{si}))) + \\
\mathbb{E}_{x_t} \log(D_j(\varphi(G(x_t), y_t))).
\]

Here \( y_{si} \) and \( y_t \) represent the category prediction probability
for samples \( x_{si} \) and \( x_t \), respectively. \( \varphi \) is a conditioning
operator, which adds constraints to features \( G(x_{si}) \) and \( G(x_t) \)
via \( y_{si} \) and \( y_t \), respectively. Let \( d_f \) and \( d_p \) be the dimensions
of the input vectors \( f \) and \( p \), respectively. \( d_0 \) is a threshold.
\( \Pi_{\varphi} \) is the outer product of multiple vectors, and \( \Theta_{\varphi} \) is the
explicit randomized multi-linear map. \( \varphi \) is defined as
\[
\varphi(f, p) = \begin{cases} 
\Pi_{\varphi}(f, p) & \text{if } d_f \times d_p \leq d_0 \\
\Pi_{\varphi}(f, p) & \text{otherwise}.
\end{cases}
\]

To keep the computation efficiency of conditioning, the outer
product on \( \Pi_{\varphi} \) is approximated by the inner-product on \( \Pi_{\varphi} \)
when the dimension of joint variable is larger than \( d_0 \). In [12],
[45], \( d_0 \) is set to 4096, which is the largest feature dimension
in typical DNNs, e.g., AlexNet. We use the same setting for
simplicity and fairness. Details of \( \varphi \) are depicted in [45].

We also train the category classifier \( C_j \) with the cross-
entropy loss to preserve the essential discriminative capacity
of the features, i.e.,
\[
L_{cls,j}(\theta_G; \theta_{C_j}) = -\mathbb{E}_{(x_{sj}, y_{sj}) \sim P_{ss}} \log(C_j(G(x_{sj}))).
\]

Note that the equilibrium challenge often appears in the
adversarial training manner [12], [45]–[47]. It means that
adversarial training can potentially deteriorate discriminative
structure of the input data. In other words, the feature distributions
may not be well aligned even if the domain discriminator
is fully confused. To alleviate this challenge, Pandey et al.
[48] used a metric transformation to keep the source samples
clustering near the corresponding categories in the feature
space. Kim et al. [49] used a self-supervised contrastive
loss to make the representations of the positive pair samples
close. However, these interesting works are designed for the
task of domain generalization, in which the target domain
is unaccessible during training, thus, they rarely consider
transferring discriminative information for the target domain.

In this work, we introduce a novel metric constraint (MC)
loss into the adversarial learning process. Inspired by the
Fisher Linear Discriminant Analysis [50], which aims to
maximize separation among distinct classes and minimize
within-class variance, we add an additional constraint on
the adversarial training process using the available category
information. To be specific, assume that \( G(x_{nj}) \) is the output
of the last full connection layer of the feature extractor \( G \)
for sample \( x_{nj} \). \( y_{nj} \) is the corresponding category label. A
normalization factor is first computed as
\[
T_j = \frac{1}{B_j} \sum_{m \neq n \in \{1, \ldots, B_j\}} \|G(x_{mj}) - G(x_{nj})\|^2_2.
\]

Then, the MC loss is formulated as
\[
L_{mc,j}(\theta_G) = \\
\frac{1}{B_j} \sum_{m \neq n \in \{1, \ldots, B_j\}} \exp(-\|G(x_{mj}) - G(x_{nj})\|^2_2/T_j).
\]

\[
L_{mc,j}(\theta_{C_j}) = \\
\frac{1}{B_j} \sum_{m \neq n \in \{1, \ldots, B_j\}} \exp(-\|G(x_{mj}) - G(x_{nj})\|^2_2/T_j),
\]

where \( m, n \in \{1, \ldots, B_j\} \) and \( m \neq n \). With this formulation,
the MC loss preserves structured information from original
data through the adversarial training manner, which ensures
the discriminability of representation and thus is capable to
improve the performance of adversarial learning.

The loss terms in Eqs. (1)-(3) enforce each source distribu-
tion \( P_{ss} \), to align with the target distribution \( P_t \). For each
pair of source domain \( D_{sj} \) and target domain \( D_t \), the feature
extractor \( G \) maps their features into a shared subspace, in
which we get the pseudo target domain containing mapped
samples of \( D_{sj} \) and \( D_t \). To guarantee that the relationships
among categories are preserved across source and target, we assign pseudo labels $\hat{Y}_t$ for the target samples using $G$ and $\{C_j\}_{j=1}^{N}$. The pseudo labels are obtained by the average of predictions from $N$ classifiers $\{C_j\}_{j=1}^{N}$. We denote the pseudo target domain as $D_{s,t}$ with data $\{(X_{s,t}, Y_{s,t})\}$, which contains data $\{(X_{s}, Y_{s})\}$ and $\{(X_{t}, Y_{t})\}$, and the corresponding distribution is $\hat{P}_{s,t}$. It is worth noting that not all the pseudo labels are correct. To ensure the credibility of predictions from $\{D_{s} \}$, we select the target samples with a confidence threshold $\kappa$. The selection rule is formulated as $\{x \in X_t | \max(\frac{1}{N} \sum_{i=1}^{N} C_i(x)) > \kappa\}$.

We align each pair of source domain and target domain in turn, and get a number of pseudo target domains $\{\hat{D}_{s1,t}, \cdots, \hat{D}_{sn,t}\}$, which corresponds to distributions $\{\hat{P}_{s1,t}, \cdots, \hat{P}_{sn,t}\}$.

2) Stage 2: Aligning the Remainder Source Domains with the Pseudo Target. In this stage, we treat the MDA task as a set of single-source domain adaptation tasks. A similar adversarial training strategy is adopted to align the remaindersource domains with the pseudo target domain in each subspace. The training process for stage 2 is illustrated in Fig. 4. We also take the $i$-th remainder source domain $D_{s_i}$ and the $j$-th pseudo target domain $\hat{D}_{s_j,t}$, as an example, in which $D_{s_i}$ needs to be aligned with $\hat{D}_{s_j,t}$. The architecture and parameters of network in stage 1 are shared in stage 2. It is worth noting that, this strategy not only mitigates domain shift between $\{D_t, \hat{D}_{s_i}\}$ and $\{\hat{D}_{s_j}, D_{s_i}\}$, respectively, but also further improve alignment between $D_t$ and $\hat{D}_{s_j}$ with the incorporation of $\hat{D}_{s_j,t}$.

For each pair of the source domain $D_{s_i}$ and the pseudo target domain $\hat{D}_{s_j,t}$, the classification loss is written as

$$L'_{cls}(\theta_G, \theta_{C_i}) = -\mathbb{E}_{(x_{s_i}, y_{s_i}) \sim P_{s_i}} [y_{s_i}] \log (C_i(F(x_{s_i})))$$

$$-\mathbb{E}_{(x_{s_i}, y_{s_i}) \sim P_{s_i}} [y_{s_i}] \log (C_j(F(x_{s_i}))).$$ (4)

We use the following adversarial loss to align the distribution of $D_{s_i}$ and $\hat{D}_{s_j,t}$, i.e.,

$$L'_{adv}(\theta_G, \theta_{D_j}) = -\mathbb{E}_{x_{s_j,t}} [\log (1 - D_j(\varphi(G(x_{s_j,t}), y_{s_j,t})))$$

$$-\mathbb{E}_{x_{s_j,t}} [\log D_j(\varphi(G(x_{s_j,t}), y_{s_j,t})))].$$ (5)

where $x_{s_j,t}$ represents sample from the $j$-th pseudo target domain $\hat{D}_{s_j,t}$, and $y_{s_j,t}$ denotes the corresponding category prediction probability. The MC loss is

$$L'_{mc}(\theta_G) =$$

$$\mathbb{E}_{(x_{s_j,t}, y_{s_j,t}) \sim \hat{P}_{s_j,t}} \log \sum_{y_{s_j,t} \in Y_{s_j,t}} \exp (-||G(x_{s_j,t}) - G(x_{s_j,t}^m)||^2_2/T_j)$$

$$\mathbb{E}_{(x_{s_j,t}, y_{s_j,t}) \sim \hat{P}_{s_j,t}} \log \sum_{y_{s_j,t} \in Y_{s_j,t}} \exp (-||G(x_{s_j,t}) - G(x_{s_j,t}^m)||^2_2/T_j)$$

$$+ \mathbb{E}_{(x_{s_j,t}, y_{s_j,t}) \sim \hat{P}_{s_j,t}} \log \sum_{y_{s_j,t} \in Y_{s_j,t}} \exp (-||G(x_{s_j,t}) - G(x_{s_j,t}^m)||^2_2/T_{s_j,t})$$

$$+ \mathbb{E}_{(x_{s_j,t}, y_{s_j,t}) \sim \hat{P}_{s_j,t}} \log \sum_{y_{s_j,t} \in Y_{s_j,t}} \exp (-||G(x_{s_j,t}) - G(x_{s_j,t}^m)||^2_2/T_{s_j,t}).$$ (6)

Each pair of source domain $D_{s_i}$ and the pseudo target domain $\hat{D}_{s_j,t}$ is used in turn to supervise model training. This alternate learning manner will enhance the robustness towards domain shift and benefit the prediction for the target domain.
in target domain to update the affine parameters $\gamma_t$ and $\beta_t$, i.e.,

$$z^j_t \triangleq \text{MN}_{\gamma_t, \beta_t}(h^j_t) = \gamma_t h^j_t + \beta_t,$$

$$z^j_s \triangleq \text{MN}_{\gamma_l, \beta_l}(h^j_s) = \gamma_l h^j_s + \beta_l.$$

In light of the fact that discriminative information is usually dominated by the labeled source samples, the above training process can relieve over-fitting by sharing the affine parameters $\gamma_l$ and $\beta_l$. Besides, this strategy can align the distributions of the whitened source features with that of target features, and thus improve the transferability of the whole network.

![Diagram](a) BN module

![Diagram](b) MN module

Fig. 5. Architectures of BN (a) and MN (b). $h_s$ and $h_t$ represent the activation of latent layers used in source and target domain, respectively.

Main differences between MN and BN are shown in Fig. 5, and they can be summarized as follows.

1) Each MN layer consists of two branches. One is for the source activations, and the other is for the target ones.
2) Each activation output in MN is normalized by the domain-specific statistics, which are estimated from those activation outputs in each domain.
3) The affine parameters are updated with the gradient information from the target domain, in comparison to BN which uses all data in the same batch.

Complexity Analysis. We consider a 4D tensor with dimension of $B \times C \times H \times W$, where $B$, $C$, $H$, and $W$ indicate batch-size, number of input channels, height and width of the input feature maps in one channel, respectively. For a batch of $B$ samples, MN estimates the statistics for each of the $H \times W$ pixels in the feature maps within $C$ channels respectively, and the computational complexity of MN is $O(BCHW)$, which is comparable with BN. In other words, there is no increase in the parameter scale for MN.

The proposed MN layer is a generic component and can be plugged into many domain adaptation methods that use DNNs as the feature extractor. In this work, we replace each BN layer in feature extractor $G$ with the MN layer.

D. Model Training

The PTMDA algorithm is optimized as follows. For each group of $D_s$ and $D_t$,

$$\max_{\theta_G, \theta_{D_s}, \theta_{D_t}} \min_{\theta_{D_s}} \lambda (L_{adv}\left(\theta_G, \theta_{D_t}\right) - L_{mc}\left(\theta_G\right)) - L_{cls}\left(\theta_G, \theta_{C_s}\right),$$

then, for each pair of $D_s$ and $D_{s,i}$,

$$\max_{\theta_G, \theta_{C_s}, \theta_{D_{s,i}}} \min_{\theta_{D_{s,i}}} \lambda' \left(L_{adv}\left(\theta_G, \theta_{D_{s,i}}\right) - L_{mc}\left(\theta_G\right) - L_{cls}\left(\theta_G, \theta_{C_s}\right)ight),$$

where $\lambda$ and $\lambda'$ are non-negative trade-off weights.

We use the mini-batch stochastic gradient descent (Mini-batch SGD) [53] to perform standard back propagation optimization and solve the objective functions in Eq. (7)-(8).

The domain discriminators $\{D_i\}_{i=1}^N$, and category classifiers $\{C_i\}_{i=1}^N$ are initialized with xavier [54]. The detailed training procedure is shown in Algorithm 1.

**Algorithm 1** Pseudo-code for PTMDA method.

**Input:** Source sets $\{(X_{s_i}, Y_{s_i})\}_{i=1}^N$, target set $X_t$, parameters $\lambda$ and $\lambda'$, confidence threshold $\kappa$.

**Output:** Feature extractor $G$, domain discriminators $\{D_i\}_{i=1}^N$, category classifiers $\{C_i\}_{i=1}^N$.

1. Initialize $G$ with the model pretrained on ImageNet, and replace its BN layers with the MN layers. Initialize $\{D_i\}_{i=1}^N$ and $\{C_i\}_{i=1}^N$ with xavier.
2. for $i = 1$ to $N$
3. Sample a mini-batch from $(X_{s_i}, Y_{s_i})$ and $X_t$.
4. Update $D_i$, $G$ and $C_i$ via Eq. (7);
5. end for
6. Predict labels $\hat{Y}_t$ and select confidential samples for $X_t$.
7. for $i = 1$ to $N$
8. for $j = 1$ to $N$ and $j \neq i$
9. Sample a mini-batch from $(X_{s_i}, Y_{s_i}), (X_{s_j}, Y_{s_j})$, and $(X_t, \hat{Y}_t)$;
10. Update $D_i$, $G$ and $C_i$ via Eq. (8);
11. end for
12. end for

IV. THEORETICAL ANALYSIS

In this section, we present some theoretical analysis for the effectiveness of PTMDA.

Ben-David et al. [55] have proposed the following generalization bound under the MDA settings. Let $\mathcal{H}$ be a hypothesis space of VC dimension $d$. Given a total of $m$ labeled samples from all source domains. For each $j \in \{1, \ldots, N\}$, let $S_j$ be a labeled sample set of size $\beta_j$, with $\sum_{j=1}^{N} \beta_j = 1$. Samples in $S_j$ are drawn from $\mathcal{D}_{s_j}$ and labeled via the labeling function $f_j$. $\epsilon_j(h)$ and $\epsilon_T(h)$ represent the errors of the hypothesis $h \in \mathcal{H}$ in source domain $\mathcal{D}_{s_j}$ and target domain $\mathcal{D}_t$, respectively. $\hat{\epsilon}_j(h)$ and $\hat{\epsilon}_T(h)$ are empirical errors. For any weight vector $\alpha \in \mathbb{R}_+^N$ with $\sum_{j=1}^{N} \alpha_j = 1$, let $\hat{\epsilon}_\alpha(h)$ be the weighted error of some fixed hypothesis $h \in \mathcal{H}$ and $\hat{\epsilon}_\alpha(h) = \sum_{j=1}^{N} \alpha_j \hat{\epsilon}_j(h)$. If $\hat{h} \in \mathcal{H}$ is the empirical minimizer of $\hat{\epsilon}_\alpha(h)$ and $h^* = \min_{h \in \mathcal{H}} \epsilon_T(h)$ is the target error minimizer, then for any $\delta \in (0, 1)$, with probability at least $1 - \delta$,

$$\epsilon_T(h) \leq \epsilon_T(h^*_T) + \sum_{j=1}^{N} \alpha_j d_{\mathcal{H}}(\mathcal{D}_{s_j}, \mathcal{D}_t) + 2 \sqrt{\frac{d \log(2m) - \log \delta}{2m}} \sum_{j=1}^{N} \alpha_j^2 \beta_j \delta.$$
where $\lambda_j = \min_{h \in \mathcal{H}} \{\epsilon_T(h) + \epsilon_s(h)\}$.

Using the triangle inequality, we have

$$\epsilon_T(h) \leq \epsilon_T(h^*_T) + \sum_{j=1}^{N} \alpha_j (d_{HΔH}(D_{s_j}, D_{t_j}^*) + d_H(D_{s_j}^*, D_t))$$

$$+ 2 \sum_{j=1}^{N} \alpha_j \lambda_j + 2 \sqrt{d \log(2m) - \log \delta} \sum_{j=1}^{N} \frac{\alpha_j^2}{\beta_j},$$

where $D_{s_j}^*$ is another source domain. Once we fix the hypothesis class $\mathcal{H}$, the last two terms in Eq. (9) and Eq. (10) will be constant. We set $\alpha_j = 1/N$ without loss of generality. Ganin et al. [56] explained that the optimal domain discriminator, which was explored in the adversarial training strategy, gives an upper bound for $d_{HΔH}(D_{s_j}, D_t)$. Therefore, optimization of the adversarial loss $L_{adv_j}(\theta_G; \theta_{D_j})$ in Eq. (1) actually minimizes an upper bound for $\sum_{j=1}^{N} \alpha_j d_{HΔH}(D_{s_j}^*, D_t)$ when constructing the pseudo target domain, and optimization of the adversarial loss $L'_{adv_j}(\theta_G; \theta_{D_j})$ in Eq. (5) minimizes an upper bound for $\sum_{j=1}^{N} \alpha_j d_{HΔH}(D_{s_j}, D_{s_j}^*)$ when aligning the remainder source domains with the pseudo target domain. The first term in the generalization bound is approximately minimized by the classification loss $L_{cls_j}(\theta_G; \theta_{C_j})$ in Eq. (2) and $L'_{cls_j}(\theta_G; \theta_{C_j})$ in Eq. (4).

Due to the analysis above, PTMDA can reduce the target error bound in Eq. (10) via minimizing the domain discrepancy not only between each pair of the source domains but also between each pair of the source and target domains. Therefore, PTMDA leads to a better approximation of the target risk, and works well in MDA settings.

**V. Experimental Results**

In this section, we first introduce the datasets and some experimental details. Then, we compare the proposed MN with BN on the single-source UDA tasks, and evaluate PTMDA on image classification tasks under MDA settings. At last, feature visualization, ablation study and other analysis are also presented.

**A. Datasets and Experimental Details**

We evaluate the PTMDA method on five benchmarks.

**Office31.** It is a collection of images from three different domains, i.e., Amazon, DSLR and Webcam [58]. Each domain consists of 31 categories which are commonly encountered in family and office, such as bike, desk, and phone.

**Office-Caltech10.** It consists of a subset of the Office31 dataset with three domains and an additional Caltech domain, including 10 classes common to the four domains.

**ImageCLEF-DA.** It consists of subsets from Caltech-256 (C), ImageNet ILSVRC 2012 (I), and Pascal VOC 2012 (P). Each domain is comprised of 12 common categories with 50 images in each category.

**DomainNet.** It consists of six different domains, namely Clipart, Infograph, Painting, Quickdraw, Real, and Sketch [23]. Each domain contains 345 classes of objects. Following the protocol of the VisDA2019 Challenge, we use the training and test splits of the given data for each domain.

**Digits-five.** It consists of five datasets, i.e., MNIST, MNIST-M, Synthetic Digits [6], SVHN, and USPS. Each dataset contains 10 categories of images. We use $ml$, $mm$, $sy$, $sv$ and $up$ to represent these domains for short, respectively.

Unless otherwise stated, the experiments are conducted on the MDA settings\(^\dagger\), and the proposed MN layers are used in PTMDA. We perform each task for five runs and report the average accuracy and standard deviation.

In this work, we follow the same experimental settings as M3SDA [23]. For each benchmark dataset, each domain is selected as the unlabeled target domain in turn, and the rest are used as labeled source domains. Taking the Digits-Five dataset as an example, we randomly sample 25,000 images from the training split and 9,000 images from the test split for MNIST, MNIST-M, Synthetic Digits, and SVHN. Since USPS only contains 9,298 images, we use the entire dataset as a separate domain. $ml$, $mm$, $sy$, $sv$ $\rightarrow$ $up$ means that the training splits of $ml$, $mm$, $sy$ and $sv$ domains are used as the source domains, and the training split of $up$ is used as the target domain. Finally, we evaluate the PTMDA method on the test split of domain $up$. In the test phase of PTMDA, test data is fed into the feature extractor $G$ and then predicted with each classifier $C_i$, and final prediction is made by the average of predictions from $N$ classifiers.

We employ three fully-connected layers for each discriminator, and use a single fully-connected layer as the classifier. In the Digits-Five based experiments, we use three convo-

---

\(^\dagger\)In section V-B, which aims to show the superiority of MN over BN, we just perform single-source UDA experiments due to the time afford.
two standards: existing MDA works [16], [23], we introduce the following
is the total size of epochs.

\[ \lambda = \lambda^t = \frac{2}{1 + \exp(-10 \cdot \frac{\text{epoch}}{\#\{\text{epoch}\}})} - 1, \]

where \( \text{epoch} \) represents the current epoch index and \( \#\{\text{epoch}\} \) is the total size of epochs.

Following the evaluation protocol extensively employed in existing MDA works [16], [23], we introduce the following two standards:

- **Source-combine**: We combine images from all source domains as a single source domain and conduct single-source unsupervised domain adaptation task.
- **Multi-source**: We conduct comparisons with existing MDA methods. We also compare PTMDA with several related single-source UDA algorithms, where single-source UDA models are trained on each pair of source and target domains, and then predictions on test data are combined.

From the experiments in source-combine scenario, we verify the necessity of developing MDA models. Comparing the performance of PTMDA with that of other MDA methods, we validate that PTMDA can aggregate information from multiple source domains effectively. The setting of source-only is used as a baseline, where all images from source domains are used to train a classifier without considering the target domain.

**B. Experiments for MN**

We design our MN layer in a generic way and make it a plug-and-play alternate for the BN layer without additional modification to the network architecture. To demonstrate the efficiency and effectiveness of the MN layer, we first perform single-source UDA experiments on three datasets with MN, and compare the results with those of BN. CDAN [45] is a popular method recently proposed for UDA, and its network architecture contains BN layers. So we choose CDAN as the baseline. For fair comparisons, we follow the experimental settings by Long et al. [45], while replacing the BN layers with MN layers in the CDAN framework. We use ResNet-50 [1] as the backbone network. The batch size is set to 36. We implement all the experiments in the PyTorch library and use an NVIDIA GeForce TITAN Xp GPU.

We conduct experiments on the Office-31 and ImageCLEF-DA datasets, and show the results in Table I and Table II, respectively. MN achieves the best results on both datasets with clear margins. Especially for the Office-31, MN surpasses BN with an improvement of 1.8%. This confirms that MN can improve the transferability of DNNs and significantly boost the performance of existing domain adaptation methods. Although DSBN [26], DA-layer [25] and MultiDIAL [27] are recent UDA-aware batch normalization methods, MN outperforms all of them. These results confirm that MN is a potential alternative to the original BN and the domain-specific BN.

To further verify that the proposed MN layer can be plugged into other domain adaptation methods, we also combine MN with two recent domain adaptation methods, i.e., BSP [57] and SymNets [9]. The experimental results are also summarized in Table I and II. Compared with original results using BN, both the performance of BSP and SymNets are improved when using the proposed MN layer. This confirms that MN is a generic component and can boost the performance of other domain adaptation methods.

**C. MDA Results on Office-Caltech10**

Comparisons on classification performance between PTMDA and the state-of-the-art approaches on Office-Caltech10 datasets are show in Table III. We can see that most methods have very high accuracy when \( D \) or \( W \) is used as the target domain, while PTMDA yields superior performance in all cases due to its ability to leverage structured information among source domains to promote generalization. As compared with the Source-combine scenario, PTMDA exceeds the accuracy of the Source-only model and DAN by 5.5% and 2.8%,

---

**TABLE II**

| Method          | I→P | P→I | I→C | C→I | C→P | P→C | Avg |
|-----------------|-----|-----|-----|-----|-----|-----|-----|
| CDAN+BN         | 77.7±0.3 | 90.7±0.2 | 97.7±0.3 | 91.3±0.3 | 74.2±0.2 | 94.3±0.3 | 87.7 |
| CDAN+DSBN [26]  | 79.0±0.2 | 92.3±0.3 | 96.5±0.2 | 86.3±0.3 | 75.3±0.2 | 94.1±0.1 | 87.3±0.1 |
| CDAN+DA-layer [25] | 79.2±0.2 | 91.8±0.2 | 96.7±0.3 | 92.0±0.1 | 77.5±0.2 | 94.0±0.3 | 88.5±0.1 |
| CDAN+MultiDIAL [27] | 78.3±0.2 | 92.3±0.2 | 97.8±0.3 | 92.8±0.2 | 78.5±0.4 | 94.7±0.4 | 89.1±0.1 |
| CDAN+MN         | 80.0±0.3 | 92.7±0.3 | 97.3±0.1 | 93.0±0.2 | 78.2±0.1 | 95.2±0.1 | 89.4±0.1 |
| BSP+BN [57]     | 79.6±0.2 | 91.8±0.2 | 95.8±0.2 | 92.8±0.4 | 77.0±0.2 | 94.5±0.3 | 88.9±0.1 |
| BSP+MN          | 81.5±0.4 | 92.1±0.2 | 96.5±0.2 | 93.8±0.4 | 79.1±0.2 | 95.4±0.3 | 89.7±0.1 |
| SymNets+BN [9]  | 80.2±0.3 | 93.6±0.2 | 97.0±0.3 | 93.4±0.3 | 78.7±0.3 | 96.4±0.1 | 89.9 |
| SymNets+MN      | 81.8±0.3 | 92.7±0.3 | 96.6±0.2 | 94.3±0.3 | 79.4±0.4 | 96.8±0.2 | 90.3±0.1 |
state-of-the-art approaches. These experimental results indicate that PTMDA can extract 3\% and 2\% comparable accuracy on I, C → P, and exceeds all the compared methods in other cases. The average accuracy of PTMDA on all the three transfer tasks is 90.2\%, which increases by 0.8\% against the base competitor MFSAN. The performance of DCTN is inferior to that of the Source-combine setting, and it seems that the ensemble of adversarial-based classifiers tends to unstable when the target domain displays large domain shift among source domains. In addition, our proposed PTMDA outperforms DANN and D-CORAL which use the ensemble of multiple classifiers in the multi-source setting by a large margin on all adaptation tasks, which verifies the effectiveness of our method.

E. MDA Results on Office31

We report the experimental results on Office31 based on ResNet-50 in Table V. It shows that the performance of DSBN [26] is slightly better than our PTMDA. The good performance of DSBN could probably due to the reason that it uses a semantic matching loss to align the centroids of the same classes across domains and achieves semantic transfer among diverse

### TABLE III

**ACCURACY(\%) COMPARISON AMONG RECENT MDA METHODS ON OFFICE-CALTECH10 WITH RESNET-50.**

| Method       | A,C,D→W | A,C,W→D | A,D,W→C | C,D,W→A | Avg    |
|--------------|---------|---------|---------|---------|--------|
| Source-only  | 99.0    | 98.3    | 87.8    | 86.1    | 92.8   |
| DAN [11]     | 99.3    | 98.2    | 89.7    | 94.8    | 95.5   |
| DANN [6]     | 96.5    | 99.1    | 89.2    | 94.7    | 94.8   |
| DSBN [26]    | 98.8±0.2| 99.9±0.1| 94.6±0.1| 92.5±0.1| 96.4±0.1|
| DSAN [59]    | 99.6±0.3| 99.2±0.1| 91.3±0.2| 92.7±0.4| 95.7±0.2|
| PTMDA        | Source-only | 99.1    | 98.2    | 89.7    | 94.8    | 95.5   |
|              | DAN [11]  | 99.5    | 99.1    | 89.2    | 91.6    | 94.8   |
|              | DANN [6]  | 99.4±0.2| 96.5±0.5| 91.2±0.3| 93.2±0.1| 95.1±0.2|
|              | D-CORAL [60] | 99.3±0.1| 98.9±0.1| 91.0±0.2| 93.2±0.1| 95.6±0.1|
|              | JAN [61]  | 99.4    | 99.4    | 91.2    | 91.8    | 95.5   |
|              | MEDA [62] | 99.3    | 99.2    | 91.4    | 92.9    | 95.7   |
|              | MCD [7]   | 99.5    | 99.1    | 91.5    | 92.1    | 95.6   |
|              | DCTN [16] | 99.4    | 99.0    | 90.2    | 92.7    | 95.3   |
|              | M3SDA [23] | 99.5    | 99.2    | 92.2    | 94.5    | 96.4   |
|              | MFSAN [17] | 99.7    | 99.4    | 93.8    | 95.4    | 97.1   |
|              | PTMDA     | 100.0±0.0| 100.0±0.0| 96.5±0.2| 96.7±0.4| 98.3±0.1|

### TABLE IV

**ACCURACY(\%) COMPARISON AMONG RECENT MDA METHODS ON IMAGECLEF-DA WITH RESNET-50.**

| Method       | LC → P | LP → C | PC → I | Avg    |
|--------------|--------|--------|--------|--------|
| Source-only  | 77.2   | 92.3   | 88.1   | 85.8   |
| DAN [11]     | 77.6   | 93.3   | 92.2   | 87.7   |
| ADDA [36]    | 76.5   | 94.0   | 93.2   | 87.0   |
| DANN [6]     | 77.9   | 93.7   | 91.8   | 87.8   |
| D-CORAL [60] | 77.1   | 93.6   | 91.7   | 87.5   |
| DSBN [26]    | 77.7±0.2| 94.1±0.3| 91.9±0.1| 87.9±0.1|
| DSAN [59]    | 77.6±0.2| 95.1±0.1| 91.4±0.6| 88.1±0.3|
| DANN [6]     | 74.5±0.4| 93.7±0.5| 87.8±0.3| 85.4±0.2|
| D-CORAL [60] | 77.7±0.1| 93.5±0.1| 91.5±0.2| 87.6±0.1|
| DCTN [16]    | 75.0   | 95.7   | 90.3   | 87.0   |
| M3SDA [23]   | 79.1   | 95.4   | 93.6   | 89.4   |
| MFSAN [17]   | PTMDA   | 79.1±0.2| 97.3±0.3| 94.1±0.3| 90.2±0.1|

### TABLE V

**ACCURACY(\%) COMPARISON AMONG RECENT MDA METHODS ON OFFICE31 WITH RESNET-50.**

| Method       | A,W→D | A,D→W | D,W→A | Avg    |
|--------------|--------|--------|--------|--------|
| Source-combine |        |        |        |        |
| DANN [6]     | 99.7   | 98.1   | 67.6   | 88.5   |
| DAN [11]     | 99.6   | 97.8   | 67.6   | 88.3   |
| D-CORAL [60] | 99.3   | 98.0   | 67.1   | 88.1   |
| DSBN [26]    | 99.0±0.2| 98.8±0.2| 70.1±0.3| 89.3±0.1|
| DSAN [59]    | 99.1±0.1| 98.6±0.1| 72.4±0.2| 90.0±0.1|
| DANN [6]     | 99.1±0.1| 98.3±0.2| 73.3±0.3| 90.2±0.2|
| D-CORAL [60] | 99.2±0.1| 98.9±0.2| 69.2±0.1| 89.1±0.1|
| DCTN [16]    | 99.3   | 98.2   | 64.2   | 87.2   |
| MADAN [63]   | 99.4   | 98.4   | 63.9   | 87.2   |
| Adv-Ensemble [64] | 99.3   | 97.3   | 68.1   | 88.3   |
| MFSAN [17]   | 99.5   | 98.5   | 72.7   | 90.2   |
| DSBN [26]    | 100.0  | 99.9   | 75.6   | 91.8   |
| PTMDA        | 100.0±0.0| 99.6±0.2| 75.4±0.4| 91.7±0.1|

respectively. The results verify that it is effective to develop algorithm for MDA task rather than simple combination of diverse source domains. In the multi-source scenario, we compare PTMDA with several state-of-the-art approaches. For single-source UDA methods, model is trained on different source domains and the predictions are combined for the testing data. For DAN, DANN and JAN, which use the adversarial training strategy to match the joint feature distribution cross different domains, PTMDA outperforms them by 3.5\%, 3.2\% and 2.8\%, respectively. In regard to those MDA algorithms, the average performance of PTMDA exceeds the baseline DCTN, M3SDA, and MFSAN, by 3\%, 1.9\%, and 1.2\%, respectively. These experimental results indicate that PTMDA can extract more transferable features from various source domains than state-of-the-art approaches.

### D. MDA Results on ImageCLEF-DA

We also evaluate PTMDA on the more challenging ImageCLEF-DA datasets and summarize the classification accuracies in Table IV. PTMDA outperforms several state-of-the-art methods on most tasks. Specifically, PTMDA achieves comparable accuracy on I, C → P, and exceeds all the compared methods in other cases. The average accuracy of PTMDA on all the three transfer tasks is 90.2\%, which increases by 0.8\% against the base competitor MFSAN. The performance of DCTN is inferior to that of the Source-combine setting, and it seems that the ensemble of adversarial-based classifiers tends to unstable when the target domain displays large domain shift among source domains. In addition, our proposed PTMDA outperforms DANN and D-CORAL which use the ensemble of multiple classifiers in the multi-source setting by a large margin on all adaptation tasks, which verifies the effectiveness of our method.
domains. PTMDA outperforms other compared methods on all tasks with an average classification accuracy of 91.7%. Transfer task on target domain A is more challenging due to the large variations on resolution of the images between A and other domains. With respect to this harder task, both DCTN and MADAN [63] are inferior to those single-source UDA algorithms in the Source-combine scenario, while PTMDA still exceeds most methods and attains an absolute accuracy improvement of 2.7% against the latest MFSAN [17]. Both DCTN and MADAN are based on adversarial learning, this result further testifies that it is necessary to add the metric constraint into the adversarial learning process.

| Method           | A→W→D A,D→W D,W→A | Avg       |
|------------------|--------------------|-----------|
| Source-only      | 98.1 93.2 50.2     | 80.5      |
| DAN [11]         | 98.8 95.2 53.4     | 82.5      |
| DANN [6]         | 98.8 96.2 54.6     | 83.2      |
| D-CORAL [60]     | 98.8 94.4 53.5     | 82.3      |
| DSBN [26]        | 99.0±0.0 95.1±0.0 51.3±0.2 81.8±0.1 |
| DSAN [59]        | 99.1±0.1 93.6±0.1 50.4±0.2 81.0±0.1 |
| Source-only      | 98.2 92.7 51.6     | 80.8      |
| DANN [6]         | 98.1±0.1 93.4±0.3 52.5±0.2 81.4±0.2 |
| D-CORAL [60]     | 98.6±0.3 94.7±0.2 53.3±0.2 82.1±0.2 |
| DA-layer [23]    | 94.8 95.8 62.9     | 84.5      |
| LtC-MSDA [65]    | 99.6 97.2 56.9     | 84.6      |
| MultiDIAL [27]   | 97.2 95.3 62.7     | 85.1      |
| PTMDA            | 99.4±0.1 97.3±0.2 53.3±0.2 83.4±0.1 |

There are a number of existing MDA approaches using AlexNet as the backbone, as it has been used for a long time in this field. We also use the AlexNet as the backbone and compare our PTMDA with these methods on the Office31 dataset. Table VI shows that DA-layers [25], LtC-MSDA [65] and MultiDIAL [27] outperform our PTMDA. DA-layers can discover latent domains and exploit this latent structure to learn a robust target classifier. LtC-MSDA constructs a knowledge graph on the prototypes of various domains to transfer information. MultiDIAL is designed not only to align the feature distributions among various domains but also to automatically decide the degree of alignment at different levels of the deep network. PTMDA outperforms DA-layers and MultiDIAL on the first two tasks. We also note that DA-layers and MultiDIAL outperform PTMDA on the third task which uses Amazon as the target domain. It is worth noting that the transfer task on target domain Amazon is more challenging than others, due to the large variations on resolution of the images between domain Amazon and other domains. In this task, the amount of samples in the source domains is relatively small (i.e., 498 images in the DSLR domain and 795 images in the Webcam domain), while the target domain has 2,817 images. PTMDA performs alignment between DSLR and Amazon, as well as alignment between Webcam and Amazon during the training procedure. This setting may lead to over-fitting among the source domains which have few training samples.

In addition, PTMDA outperforms all the Source-combine-based methods in the average sense. It validates again the effectiveness of PTMDA in transferring knowledge from multiple source domains to target domain.

**F. MDA Results on DomainNet**

The results on the DomainNet benchmark are shown in Table VII. We can see that PTMDA achieves comparable accuracy compared with these methods. Specifically, PTMDA outperforms all the compared methods when Clipart, Painting or Sketch is selected as the target domain, which verifies the effectiveness of PTMDA. PTMDA also obtains 0.7%, 3.0% and 4.0% absolute improvements compared with the recent MDA methods CMSS [66], SHOT [67] and MDDA [18]. This is because CMSS, SHOT and MDDA focus only on aligning multiple source domains with the target, while PTMDA aims at reducing the domain shift which exists not only between source and target domains but also among diverse source domains. Surprisingly, most of the MDA methods achieve lower accuracy than the results of Source-combine scenario in the Quickdraw task. This can be explained by the fact that there are large domain gaps between Quickdraw and the other domains. While DCTN and MDDA also performs multi-way adversarial learning to address the shift between each source and target domain, our PTMDA achieves 9.0% and 4.0% performance improvements over them, which validates the effectiveness of the proposed MC loss term in adversarial learning. Moreover, PTMDA outperforms Source-combine methods which use the mixed data from multiple source domains. It indicates that exploiting structured information among diverse source domains could benefit the adaptation performance. In addition, our proposed PTMDA outperforms DANN and D-CORAL which use the ensemble of classifiers among diverse source domains. It indicates that exploiting structured information among diverse source domains could benefit the adaptation performance. In addition, our proposed PTMDA outperforms DANN and D-CORAL which use the ensemble of classifiers among diverse source domains. It indicates that exploiting structured information among diverse source domains could benefit the adaptation performance.

**G. MDA Results on Digits-five**

Table VIII shows the experimental results of MDA on the Digits-Five benchmark. For fair comparison, all experiments are performed on the same feature extractor architecture. ADAGE [37] slightly outperforms our PTMDA in overall average accuracy. The reason may be that ADAGE uses an elaborate Hallucinator block to remove the domain-specific style of the input images and achieves better adaptation performance. Nevertheless, our PTMDA obtains the best accuracy when using the mt as the target domain. Compared with the LtC-MSDA method [65], our PTMDA still achieves comparable performance. PTMDA also significantly outperforms
is empirically superior for this task. Whereas the performance shows that PTMDA can sufficiently extract discriminative and M3SDA use well-designed loss term in the multi-source [18] on each task, respectively. Notice that even though DCTN and MDDA [18]) on most of the adaptation tasks, which confirms Multi-source LASSIFICATION ACCURACY 73% ± 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 ± 0.5
DAN [11] 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 ± 0.6
JAN [61] 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 ± 0.6
DANN [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 ± 0.7
ADDA [36] 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 ± 0.6
MCD [7] 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 ± 0.6
DSBN [26] 45.5 ± 0.5 19.3 ± 0.1 45.5 ± 0.1 6.7 ± 0.2 54.6 ± 0.4 36.6 ± 0.2 34.7 ± 0.1
DSAN [59] 53.4 ± 0.3 20.1 ± 0.5 40.5 ± 0.2 14.6 ± 0.2 57.4 ± 0.4 45.2 ± 0.1 38.5 ± 0.2
PTMDA 66.0 ± 0.3 28.5 ± 0.2 58.4 ± 0.4 13.0 ± 0.5 63.0 ± 0.2 54.1 ± 0.3 47.2 ± 0.1

TABLE VIII ACCURACY(%) COMPARISON AMONG RECENT MDA METHODS ON DIGITS-FIVE UNDER FULL PROTOCOL [23].

| Method | mt, up, sv → sy | mm, mt, sv, sy → mt | mm, mt, sv, sy → up | mm, mt, up, sv → sy | mm, mt, ap, sv → sy | Avg |
|--------|----------------|----------------------|--------------------|--------------------|---------------------|-----|
| Source-combine | 63.7 ± 0.8 | 92.3 ± 0.9 | 90.7 ± 0.5 | 71.5 ± 0.8 | 83.4 ± 0.8 | 80.3 ± 0.8 |
| Source-only | 67.9 ± 0.8 | 97.5 ± 0.6 | 93.5 ± 0.9 | 67.8 ± 0.8 | 86.9 ± 0.9 | 82.7 ± 0.8 |
| DAN [11] | 70.8 ± 0.9 | 97.9 ± 0.8 | 93.5 ± 0.8 | 68.5 ± 0.9 | 87.4 ± 0.7 | 83.6 ± 0.8 |
| DNN [6] | 68.6 ± 0.1 | 96.3 ± 0.2 | 93.5 ± 0.2 | 75.4 ± 0.1 | 86.5 ± 0.1 | 84.0 ± 0.1 |
| MDDA | 78.1 ± 0.3 | 96.4 ± 0.3 | 92.3 ± 0.2 | 76.4 ± 0.2 | 87.8 ± 0.2 | 86.2 ± 0.2 |
| DSAN [59] | 73.4 ± 0.7 | 90.5 ± 0.8 | 88.7 ± 0.9 | 63.5 ± 0.9 | 82.4 ± 0.7 | 77.7 ± 0.8 |
| D-CORAL [60] | 60.5 ± 0.7 | 95.3 ± 0.5 | 94.2 ± 0.9 | 62.5 ± 0.7 | 85.4 ± 0.8 | 80.4 ± 0.7 |
| JAN [61] | 65.9 ± 0.9 | 97.2 ± 0.7 | 95.4 ± 0.8 | 75.3 ± 0.7 | 86.6 ± 0.6 | 84.1 ± 0.7 |
| ADDA [36] | 71.6 ± 0.5 | 97.9 ± 0.8 | 92.8 ± 0.7 | 75.5 ± 0.5 | 86.5 ± 0.6 | 84.8 ± 0.6 |
| MEDA [62] | 71.3 ± 0.8 | 96.5 ± 0.8 | 97.0 ± 0.8 | 78.5 ± 0.8 | 84.6 ± 0.8 | 85.6 ± 0.8 |
| MCD [7] | 72.5 ± 0.7 | 96.2 ± 0.8 | 95.3 ± 0.7 | 78.9 ± 0.8 | 87.5 ± 0.7 | 81.6 ± 0.7 |
| DCTN [16] | 70.5 ± 1.2 | 96.2 ± 0.8 | 92.8 ± 0.3 | 77.6 ± 0.4 | 86.8 ± 0.8 | 84.8 ± 0.7 |
| M3SDA [23] | 72.8 ± 1.1 | 98.4 ± 0.7 | 96.1 ± 0.8 | 81.3 ± 0.9 | 89.6 ± 0.6 | 87.7 ± 0.8 |
| ADAGE [37] | 85.3 ± 0.2 | 98.3 ± 0.3 | 97.1 ± 0.3 | 85.3 ± 0.2 | 96.2 ± 0.1 | 92.4 |
| MDDA [18] | 78.6 | 98.8 | 93.9 | 79.3 | 89.7 | 88.1 |
| SHOT [67] | 80.2 ± 0.4 | 98.2 ± 0.4 | 97.1 ± 0.3 | 84.5 ± 0.3 | 91.1 ± 0.2 | 90.2 |
| CMSS [60] | 75.3 ± 0.6 | 99.0 ± 0.1 | 97.7 ± 0.1 | 88.4 ± 0.5 | 93.7 ± 0.2 | 90.8 ± 0.3 |
| LtC-MSDA [65] | 85.6 ± 0.8 | 99.0 ± 0.4 | 98.3 ± 0.4 | 83.2 ± 0.6 | 93.0 ± 0.5 | 91.8 |
| PTMDA | 85.2 ± 0.4 | 99.3 ± 0.1 | 97.6 ± 0.3 | 82.5 ± 0.5 | 93.4 ± 0.3 | 91.6 ± 0.2 |

other recent MDA methods (e.g., CMSS [66], SHOT [67] and MDDA [18]) on most of the adaptation tasks, which confirms the improvement when using pseudo target domain for MDA task. Specifically, we achieved 6.64%, 0.52%, 3.71%, 3.17%, and 3.73% accuracy improvements with respect to MDDA [18] on each task, respectively. Notice that even though DCTN and M3SDA use well-designed loss term in the multi-source scenario, their performance is inferior to our PTMDA. This shows that PTMDA can sufficiently extract discriminative features for target and get a more generalized classifier, which is empirically superior for this task. Whereas the performance of DCTN is inferior to those of MCD [7] and MEDA [62], it reveals that only reducing shifts among various domains by adversarial learning may lead to suboptimal results due to equilibrium challenge issue. Compared with DANN [6] in the Source-combine setting, even the most difficult task mt, up, sv, sy → mm, PTMDA yields a significant improvement of 14.43%, which demonstrates that PTMDA can improve the performance of MDA by attenuating domain shifts across multiple domains.
We use DCTN [16], M3SDA [23], and MDDA [18] as the baseline methods. The method denoted by PT+BN implements the combination of pseudo target domain with the BN layers, but excludes the MN layers and the MC loss, where “PT” is short for “Pseudo Target”. The method denoted by PT+MN implements the combination of pseudo target domain with the MN layers. The method denoted by PT+MC implements the combination of pseudo target domain with the MC loss and the BN layers, but excludes the MN layers. Table IX shows the results of ablation study.

Comparing with DCTN [16], M3SDA [23], and MDDA [18], PT+BN performs better in the average sense, which demonstrates the effectiveness of pseudo target domain. Notice that PT+MN obtains an improvement of 0.8% over PT+BN. It indicates that the distribution alignments between the whitened source domains and the target domain enhances the generalization ability of DNNs. PT+MC achieves 6.4%, 3.1%, and 1.3% improvements against DCTN, MDDA, and PT+BN, respectively. Since they are adversarial based methods, these results verify that the MC loss can moderate the equilibrium challenge and improve the performance of adversarial learning. PT+MC has a larger improvement than PT+MN, which indicates that the MC loss makes greater contribution to the MDA tasks. Comparing PT+MC with PTMDA, the average accuracy is improved from 91.2% to 91.6%. The result validates the effectiveness of our proposed MN layers. From the results shown above, we confirm that each component in PTMDA has its specific contribution. By combining pseudo target domain with MN layers and the MC loss, PTMDA leads to further improvement. All these results indicate their complementarity and superiority of the PTMDA method.

We also provide the experimental results obtained by only using the real target domain with pseudo labels (i.e., no source domain in the pseudo target domain) in Table IX, and it is denoted as MN+MC-PT. We can see that PTMDA outperforms MN+MC-PT by a large margin (91.6% vs. 87.6%). It means that the pseudo target domain plays an important role in improving the performance of the MDA tasks.

### J. Other analysis

We conduct experiments with different numbers (i.e., two or three) of source domains on Office-Caltech10. The results are shown in Table X. In terms of the overall performance, the average accuracy which is obtained by using three source domains is slightly better than those of using two source domains. It seems that the number of source domains has little influence on the performance of PTMDA.

To study the impact of different orders of source domains, we conduct additional experiments. For the Office-Caltech10 dataset, we use different orders of source domains for $A, D, W \rightarrow C$ and $C, D, W \rightarrow A$ with different numbers (i.e., two or three) of source domains, and the results are shown in Table X. For the Digits-five dataset, among the twenty-four permutations of four source domains, we just randomly choose six permutations for the task $mt, up, sv, sy \rightarrow mm$ due to the time afford. The results are shown in Table XI. We run

---

**Fig. 7. Visualization on task $A, D, W \rightarrow C$ for both the actual target domain and the pseudo target domains. The actual target domain $C$ is colored in red, and other domains in the pseudo target domains are colored in blue. Features corresponding to the same category are annotated by the same marker.**

**H. Feature Visualization**

To visualize the PTMDA features before and after adaptation, as well as the features obtained by M3SDA, we conduct experiments by using t-SNE [68] on task $A, D, W \rightarrow C$ on Office-Caltech10. Domains are presented in different colors for clarity. As we can see from Fig. 6, compared with the features of the Source-only, all the features of PTMDA show good adaptation patterns. It reveals that PTMDA can successfully learn transferable features from multiple source domains. Besides, the target features in red are easier to be classified than others. It shows that the target features learned by PTMDA achieve desirable discrimination ability. Compared with the features of M3SDA, categories in PTMDA are aligned better, and different domains show lower discrepancy. This leads to a more discriminative target feature space.

We also show the learned features for both the actual target domain $C$ and the pseudo target domains (i.e., $AC$, $DC$, and $WC$) in Fig. 7. Features corresponding to the same category are visualized with the same marker. As we can see, the actual target features do separate clearly. Features of other domains (i.e., $A$, $D$, and $W$) in the pseudo target domains also show class-based discrimination, and they cluster around the corresponding actual target features. It verifies that the domain difference between the source domain and the actual target domain, inside each pseudo target domain, has been minimized actually.

**I. Ablation study**

In order to find which component of PTMDA plays an important role in learning domain adaptation features, we perform ablation study over various combinations on the Digits-Five adaptation task.

---

We use DCTN [16], M3SDA [23], and MDDA [18] as the baseline methods. The method denoted by PT+BN implements the combination of pseudo target domain with the BN layers, but excludes the MN layers and the MC loss, where “PT” is short for “Pseudo Target”. The method denoted by PT+MN implements the combination of pseudo target domain with the MN layers. The method denoted by PT+MC implements the combination of pseudo target domain with the MC loss and the BN layers, but excludes the MN layers. Table IX shows the results of ablation study.

Comparing with DCTN [16], M3SDA [23], and MDDA [18], PT+BN performs better in the average sense, which demonstrates the effectiveness of pseudo target domain. Notice that PT+MN obtains an improvement of 0.8% over PT+BN. It indicates that the distribution alignments between the whitened source domains and the target domain enhances the generalization ability of DNNs. PT+MC achieves 6.4%, 3.1%, and 1.3% improvements against DCTN, MDDA, and PT+BN, respectively. Since they are adversarial based methods, these results verify that the MC loss can moderate the equilibrium challenge and improve the performance of adversarial learning. PT+MC has a larger improvement than PT+MN, which indicates that the MC loss makes greater contribution to the MDA tasks. Comparing PT+MC with PTMDA, the average accuracy is improved from 91.2% to 91.6%. The result validates the effectiveness of our proposed MN layers. From the results shown above, we confirm that each component in PTMDA has its specific contribution. By combining pseudo target domain with MN layers and the MC loss, PTMDA leads to further improvement. All these results indicate their complementarity and superiority of the PTMDA method.

We also provide the experimental results obtained by only using the real target domain with pseudo labels (i.e., no source domain in the pseudo target domain) in Table IX, and it is denoted as MN+MC-PT. We can see that PTMDA outperforms MN+MC-PT by a large margin (91.6% vs. 87.6%). It means that the pseudo target domain plays an important role in improving the performance of the MDA tasks.
each experiment five times, and find that different orders of source domains of each task get identical average accuracy. It is consistent with our intuition. Actually, each group of the source domain and target domain is used in turn to construct a pseudo target domain, and samples are randomly sampled from each source domain. Thus, the order of the source-target adaption task does not affect the final performance.

**VI. Conclusion**

In this paper, we propose Pseudo Target for MDA (PTMDA), in which we construct a pseudo target domain to mimic a new domain in a group-specific subspace and align the remainder source domains with the pseudo target domain. PTMDA can sufficiently extract structural and relevant information from multiple sources to promote transfer efficiency, and improve the performance of classifier on the real target domain. To further enhance the transferability of deep neural networks, we design a matching normalization layer to align the feature distributions of different domains in the intermediate layers of the feature extractor. Extensive experiments on several benchmarks validate that PTMDA can outperform or compete state-of-the-art methods. Ablation study shows that each component in PTMDA has specific contribution to the MDA task. In the future, we plan to extend this method to the scenario with label shift.

**References**

[1] K. M. He, X. Y. Zhang, S. Q. Ren et al., “Deep residual learning for image recognition,” in CVPR, 2016, pp. 770–778.

[2] S. S. Wang, L. Zhang, W. M. Zuo et al., “Class-specific reconstruction transfer learning for visual recognition across domains,” IEEE Trans. Image Process., vol. 29, pp. 2424–2438, 2020.

[3] S. T. Chen, M. Harandi, X. N. Jin et al., “Domain adaptation by joint distribution invariant projections,” IEEE Trans. Image Process., vol. 29, pp. 8264–8277, 2020.

[4] C.-X. Ren, D.-Q. Dai, K.-K. Huang et al., “Transfer learning of structured representation for face recognition,” IEEE Trans. Image Process., vol. 23, no. 12, pp. 5440–5454, 2014.

[5] J. L. Hu, J. W. Lu, Y. P. Tan et al., “Deep transfer metric learning,” IEEE Trans. Image Process., vol. 25, no. 12, pp. 5576–5588, 2016.

[6] Y. Ganin, E. Ustinova, H. Ajakan et al., “Domain-adversarial training of neural networks,” JMLR, vol. 17, no. 59, pp. 1–35, 2016.

[7] K. Saito, K. Watanabe, Y. Ushiku et al., “Maximum classifier discrepancy for unsupervised domain adaptation,” in CVPR, 2018, pp. 3723–3732.

[8] C.-X. Ren, X.-L. Xu, and H. Yan, “Generalized conditional domain adaptation: A causal perspective with low-rank translators,” IEEE Trans. Cyber., vol. 50, no. 2, pp. 821–834, 2020.

[9] Y. B. Zhang, H. Tang, K. Jia et al., “Domain-symmetric networks for adversarial domain adaptation,” in CVPR, 2019, pp. 5026–5035.

[10] C.-X. Ren, J. S. Feng, D.-Q. Dai, and S. Yan, “Heterogenous domain adaptation via covariance structured feature translators,” IEEE Trans. Cyber., vol. 51, no. 4, pp. 2166–2177, 2021.

[11] M. S. Long, Y. Cao, Z. J. Cao et al., “Transferable representation learning with deep adaptation networks,” IEEE Trans. Pattern Anal. and Mach. Intell., vol. 41, no. 12, pp. 3071–3085, 2019.

[12] J. J. Li, E. P. Chen, Z. M. Ding et al., “Maximum density divergence for domain adaptation,” IEEE Trans. Pattern Anal. and Mach. Intell., vol. 43, no. 11, pp. 3918–3930, 2021.

[13] L. X. Duan, D. Xu, and J. W.-H. Tsang, “Domain adaptation from multiple sources: A domain-dependent regularization approach,” IEEE Trans. Neural Netw. and Learning Syst., vol. 23, no. 3, pp. 504–518, 2012.

[14] S. C. Zhao, B. Li, X. Y. Yue et al., “Multi-source domain adaptation for semantic segmentation,” in NeurIPS, 2019, pp. 7285–7298.

[15] Y. Mansour, M. Mohri, and A. Rostamizadeh, “Domain adaptation with multiple sources,” in NeurIPS, 2009, pp. 1041–1048.

[16] R. J. Xu, Z. L. Chen, W. M. Zuo et al., “Deep cocktail network: Multi-source unsupervised domain adaptation with category shift,” in CVPR, 2018, pp. 3964–3973.

[17] Y. C. Zhu, F. Z. Zhuang, and D. Q. Wang, “Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources,” in AAAI, 2019.

[18] S. C. Zhao, G. Z. Wang, S. H. Zhang et al., “Multi-source distilling domain adaptation,” in AAAI, 2020.

[19] H. Zhao, S. H. Zhang, G. H. Wu et al., “Adversarial multiple source domain adaptation,” in NeurIPS, 2018, pp. 8559–8570.
