A Two-Stage Adaptation Network (TSAN) for Remote Sensing Scene Classification in Single-Source-Mixed-Multiple-Target Domain Adaptation (S^2M^2T DA) Scenarios

Juepeng Zheng, Wenzhao Wu, Shuai Yuan, Yi Zhao, Weijia Li,
Lixian Zhang, *Graduate Student Member, IEEE*, Runmin Dong, *Student Member, IEEE*,
and Haohuan Fu, *Member, IEEE*

***Abstract*—Over the past decade, domain adaptation (DA) algorithms have been proposed to address domain gap problems as they do not need any interpretation in the target domain. However, most existing efforts focus on scenarios with only one source domain and one target domain. In this article, we explore the scenario with one source domain and mixed multiple target domains for remote sensing applications and propose a new algorithm, named the two-stage adaptation network (TSAN). First, we utilize the adversarial learning approach to confuse the classifier to discriminate between the source domain and the whole mixed-multiple-target domain. Second, we adopt self-supervised learning to divide the mixed-multiple-target domain with automated generation of "pseudo"-domain labels, which guides our network to learn intrinsic features of multiple target domains. Finally, these two steps are combined as an iterative procedure. We integrate a test dataset that includes five remote sensing datasets and ten classes. Our method achieves an average accuracy of 63.25% and 73.68% with two typical backbones, considerably outperforming other DA methods with an average accuracy improvement of 4.84%–20.19% and 9.06%–17.04%, respectively. Furthermore, we identify the negative transfer effect in existing mainstream DA methods in remote sensing image classification with multiple different domains.**

***Index Terms*—Adversarial learning, deep learning, domain adaptation (DA), mixed-multiple-target domain, remote sensing image classification, self-supervised learning.**

Manuscript received February 23, 2021; revised June 28, 2021; accepted August 12, 2021. Date of publication August 24, 2021; date of current version January 31, 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0604500 and in part by the National Natural Science Foundation of China under Grant U1839206. *(Corresponding author: Haohuan Fu.)*

Juepeng Zheng, Lixian Zhang, and Runmin Dong are with the Ministry of Education Key Laboratory for Earth System Modeling, Tsinghua University, Beijing 100084, China, and also with the Department of Earth System Science, Tsinghua University, Beijing 100084, China (e-mail: zjp19@mails.tsinghua.edu.cn; zhanglx18@mails.tsinghua.edu.cn; drm20@mails.tsinghua.edu.cn).

Wenzhao Wu is with the National Supercomputing Center in Wuxi, Wuxi 214072, China (e-mail: wumz13@tsinghua.org.cn).

Shuai Yuan is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: s-yuan16@tsinghua.org.cn).

Yi Zhao and Haohuan Fu are with the Ministry of Education Key Laboratory for Earth System Modeling, Tsinghua University, Beijing 100084, China, also with the Department of Earth System Science, Tsinghua University, Beijing 100084, China, and also with the National Supercomputing Center in Wuxi, Wuxi 214072, China (e-mail: y-zhao19@mails.tsinghua.edu.cn; haohuan@tsinghua.edu.cn).

Weijia Li is with the CUHK-SenseTime Joint Lab, The Chinese University of Hong Kong, Hong Kong (e-mail: weijiali@cuhk.edu.hk).

Digital Object Identifier 10.1109/TGRS.2021.3105302

I. Introduction

ALTHOUGH deep learning has been successfully exploited in remote sensing image classification tasks [1], [2], it requires sufficient annotations in a particular region (named the source domain) to extract efficient features. When it turns to a new environment (named the target domain), the accuracy of the trained classifier may drop dramatically. More specifically, due to different sensors, illumination, reflectance conditions, and topographic features, applying deep learning to large-scale and multitemporal studies becomes a challenging problem, which needs time-consuming annotations for images from different domains to obviate the deterioration of model accuracy. For example, as illustrated in Fig. 1, remote sensing images with the same annotations but derived from different datasets (i.e., UC Merced and NWPU-RESISC45) exhibit significant differences in spectral distribution via gray histograms. If we directly adopt the model trained from the source domain, the classifier may make incorrect decisions for the target domain.

Due to domain adaptation (DA), we can address the scarcity of annotations for target domains by leveraging the available labeled images in the source domain more effectively [3]. The proposed DA methods adapt a model to a new data source by minimizing the distribution gap between source and target domains. However, most existing DA methods work with an assumption of "single-source-single-target (S^3T)." Only with one source domain and one target domain, S^3T assumes that the target domain dataset conforms with one specific distribution. S^3T is an ideal situation for DA. In reality, there are many circumstances where we only have one source domain, while we have to face multiple target domains. For example, as shown Fig. 2(a), when we are supposed to address three different datasets (i.e., A, B, and C), while we only possess one dataset (i.e., A) with annotations, we have to train two different S^3T DA models (i.e., $A \rightarrow B$ and $A \rightarrow C$) to deal with two target domains whose annotations are unavailable. When extending to many different target domains, such a solution becomes impractical. To this end, some researchers dive into the "single-source-multiple-target (S^2MT)" DA issue [4], [5], which is similar to Fig. 2(b). S^2MT DA methods often focus on tackling the ineffectiveness of mainstream feature

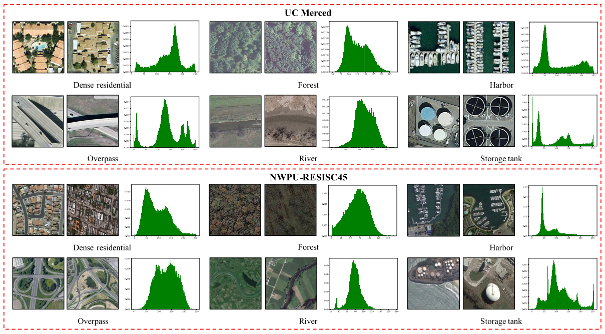


Fig. 1. Spectral difference between two different remote sensing image datasets. The green histograms represent the mean of all histograms in one specific category. The x-axis denotes the pixel values ranging from 0∼255, and the y-axis represents the statistics of percentage for each pixel value.

alignment methods when dealing with multiple target domains and adapting the representative model from source to multiple target domains simultaneously.

In practical applications, the real-world scenario contains adequate ranges of environments, continuous generation of satellite images from different sources, and various changes in the underlying technologies [6]. Specifically, for large-scale applications, it is challenging to adopt one single source dataset to different target datasets simultaneously from different locations, sensors, or times. Take land cover mapping as an example: we only have a city's land cover annotations (e.g., Shanghai) and intend to make a land cover mapping in other different cities (e.g., Beijing, New York, and London), where we do not have any annotations, or we only have land cover annotations in the year of 2020 and intend to acquire land cover maps in other different years (e.g., 1990, 2000, and 2010) without annotations. It would be time-consuming that we have to train different adaptive models that only transfer from one domain to another domain. Furthermore, in many cases, fusing multisensor and multitemporal images is necessary for cross-regional and large-scale remote sensing applications. As a consequence, it is quite common that we have to face various satellite images (without any metainformation) from multisensor and multitemporal over a large heterogeneous region at the same time. On the other hand, most existing remote sensing datasets do not provide the exact metainformation for each satellite image, such as DOTA [7] and RSD46-WHU [8]. The scenarios mentioned above have a common characteristic: the satellite images are composited from different sensors or acquired from different dates. Therefore, when we conduct a large-scale or long-time-series application, it is difficult to distinguish and separate the origins of these composited satellite images for domain transferring. Hence, it is essential to exploit the potential for a more advanced DA scenario, i.e., "single-source-mixed-multiple-target (S^2M^2T)" DA, which learns knowledge from one source domain and adapts the model to the mixed multiple target domains simultaneously. Fig. 2(c) indicates that target domains B and C are mixed and encrypted. That is, in S^2M^2T DA, target datasets B and C are not explicitly separated as S^2MT DA scenario, and we only possess the annotations in source dataset A. Limited discussions have been made on the S^2M^2T problem [9], [10]. If we consider the mixed-multiple-target domain as the single target domain [see Fig. 2(c)], the performance of $A \rightarrow B, C$ may fail to employ the representations shared across multiple domains at the same time under the same methods as S^3T DA scenario, resulting in inevitable negative transfer effect. We will elaborate on the negative transfer effect on our collected dataset in the result and analysis section.

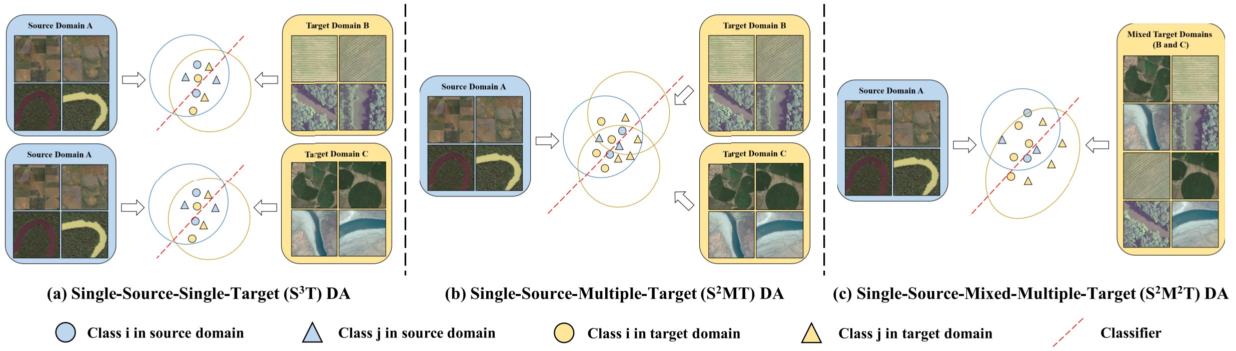
In this article, we emphasize on the S^2M^2T DA problem, that is, we only utilize one source domain dataset with annotations to adapt to the mixed-multiple-target domain dataset, whose annotations and domain labels are both unobservable. To this end, our contributions in this context can be highlighted as three aspects.

1) We propose a two-stage adaptation network (TSAN) to provide DA solutions for the S^2M^2T DA scenario in remote sensing classification problems. To the best of our knowledge, this work is among the first attempts on the S^2M^2T DA issue in the remote sensing community.
2) We employ a two-stage adaptation that integrates the subtarget adaptation and the source–target adaptation. The subtarget adaptation adopts self-supervised learning to distinguish mixed multiple target domains and confuse the discriminator among "pseudo"-subtarget domains. The source–target adaptation adopts adversarial learning to confuse the discriminator between the source domain and the mixed multitarget domain. In addition, these two major adaptation methods are combined as an iterative procedure.
3) We identify and analyze the negative transfer effect of existing mainstream DA methods in S^2M^2T DA. A major reason might be the domain gaps and category misalignments among mixed target domains, indicating the importance and the challenges of the S^2M^2T DA issue.

II. Related Work

A. Domain Adaptation

DA approaches aim at aligning the model to new data distributions without utilizing a large number of time-consuming annotations and recently have been paid much attention in the machine learning domain [11]. A rich line of DA methods can help to diminish the discrepancy between the source domain and the target domain through adapting the representations in hidden feature map [12]–[15] or input space [3], [16]. Some recent algorithms concentrate on aligning feature distributions through minimizing discrepancy across domains [17] or incorporating a classifier in the source domain with gradient reversal to fool a domain discriminator [12], [15]. At present, the latter one, also named adversarial-based DA, has become a mainstream avenue for DA issues. The adversarial-based DA assigns a discriminator (i.e., a binary classifier) to identify if an input image comes from the source or target domain and tries to fool the discriminator not to well recognize different domains. However, the aforementioned DA methods generally concentrate on the S^3T DA scenario, which could not meet


Source Domain A
Target Domain B
Source Domain A
Target Domain C
Source Domain A
Target Domain B
Target Domain C
Source Domain A
Mixed Target Domains (B and C)
○ Class i in source domain
△ Class j in source domain
○ Class i in target domain
△ Class j in target domain
Classifier

(a) Single-Source-Single-Target (S³T) DA (b) Single-Source-Multiple-Target (S²MT) DA (c) Single-Source-Mixed-Multiple-Target (S²M²T) DA

Fig. 2. Different DA scenarios: (a) S^3T DA, (b) S^2MT DA, and (c) S^2M^2T DA.

the real-world demands in many cases. Unfortunately, rare attention has been paid to S^2M^2T DA, which is a more challenging transfer task than S^2MT DA [9], [10]. To date, S^2MT and S^2M^2T are still at an embryonic stage and await more exploration.

B. Domain Adaptation in Remote Sensing

Despite the promises of remote sensing to address such ambitious issues for remote sensing image classification, two major difficulties hinder this technology from achieving a broader array of applications [6]: 1) labeled data are not always adequate at each scenario and 2) the models need to be enough general to address data acquired with different sensors and probably under different environments. Therefore, DA has been applied in the remote sensing field to deal with large-scale and long-time-series applications using multisource and multitemporal remote sensing images, in which differences in ground environment and photographed instrument may readily impact the model's transferable capacity [18]. Nowadays, DA effectively minimizes the distribution gaps between images due to different sensors and conditions, and emerges in all kinds of remote sensing applications ranging from classification [19]–[23], semantic segmentation [24]–[26], and object detection [27]–[29]. In the remote sensing community, the off-the-shelf DA methods mainly concentrate on the S^3T DA issue. The issues of S^2MT and S^2M^2T have never been studied in the remote sensing field until now. To the best of our knowledge, our work is the first attempt on the S^2M^2T DA issue in the remote sensing community.

III. Methodology

A. Preliminary and Overview

In the S^3T DA scenario, the source domain dataset ($\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}$) has annotations, and we can access the label-free target domain dataset ($\mathcal{D}_t = \{(\mathbf{x}_i^t)\}_{i=1}^{n_t}$), in which s and t denote the source domain and the target domain, respectively. n_s and n_t are the number of images in the source domain dataset and the target domain dataset. Here, $\mathbf{x}_i^s$ is the ith sample in $\mathcal{D}_s$, and y_i^s denotes the corresponding annotation; $\mathbf{x}_i^t$ is the ith sample in $\mathcal{D}_t$ without a known label. We assume that the distributions of the source domain ($\mathcal{P}_s(\mathbf{x}^s, y^s)$) and the target domain ($\mathcal{P}_t(\mathbf{x}^t)$) are usually different. S^3T DA aims at training a classifier and a domain-invariant feature extractor that work for both $\mathcal{D}_s$ and $\mathcal{D}_t$.

To smooth the presentation of the S^2M^2T DA, we first describe the S^2MT DA scenario. In S^2MT DA, we have multiple target domains $\mathcal{D}_T = \{\mathcal{D}_t^j\}_{j=1}^k = \{\{(\mathbf{x}_i^{t,j})\}_{i=1}^{n_t^j}\}_{j=1}^k$, where k is the quantity of target domains and n_t^j is the quantity of satellite images in the jth target domain. The overall target domain distribution is formulated as $\mathcal{P}_T = \mathcal{P}_t(\mathbf{x}^t) = \{\mathcal{P}_t^j(\mathbf{x}^{t,j})\}_{j=1}^k = \sum_{j=1}^k \omega^j \mathcal{P}_t^j(\mathbf{x}^{t,j})$, in which, $\forall j \in [k]$, $\omega^j \in [0, 1]$ and $\sum_{j=1}^k \omega^j = 1$. S^2MT DA attempts to adapt the model training from $\mathcal{D}_s$ to k target domains $\{\mathcal{D}_t^j\}_{j=1}^k$ simultaneously. Since the distribution of the multiple-target-domain dataset is available in S^2MT DA, the jth target domain $\mathcal{D}_t^j$ can be explicitly drawn from the posterior $\omega^j \mathcal{P}_t^j(\mathbf{x}^{t,j}) / \sum_{j'=1}^k \omega^{j'} \mathcal{P}_t^{j'}(\mathbf{x}^{t,j'})$. Therefore, existing S^3T DA approaches can deal with the problem through training k target-specific adaptation models or by other S^2MT DA algorithms [4], [5].

In contrast to S^2MT DA, S^2M^2T DA is established on a mixture of target domains $\mathcal{D}_T = \{\mathcal{D}_t^j\}_{j=1}^k = \{(\mathbf{x}_i^t)\}_{i=1}^{n_T}$, where $n_T = \sum_{j=1}^k n_t^j$ denotes the total number of images in the target domain. Unlike the S^2MT DA scenario, the proportions of different target domains in the mixed datasets $\{\omega^j\}_{j=1}^k$ are unknown. Hence, we cannot leverage S^2MT DA methods for S^2M^2T DA. If we directly adopt existing S^3T DA algorithms and consider the mixed target domains as one target domain in a brute-force way, the training objective will facilitate domain-invariant representations to align the whole mixed-multiple-target domain $\mathcal{D}_T$ rather than k target domains $\{\mathcal{D}_t^j\}_{j=1}^k$. Because of the discrepancy among subsubjects from k distributions, adaptation for the whole mixed-multiple-target domain may possibly lead to severe category misalignments and drastic negative transfer effects. Compared to S^2MT DA, it is a key message in the S^2M^2T DA scenario that we are unavailable to any information about the domain label of each image in the target dataset in advance. Therefore, the challenges of the S^2M^2T DA scenario are twofold. On the one hand, as the number of target domains is more than one and all of them are organized as a mixture distribution, it is easy to hamper the accuracy of mainstream S^3T DA methods because of the domain gaps among the multiple target domains. On the other hand, the category misalignments among these mixed multiple target domains will ruin the performance of existing DA approaches, which leads to an unavoidable negative transfer effect.

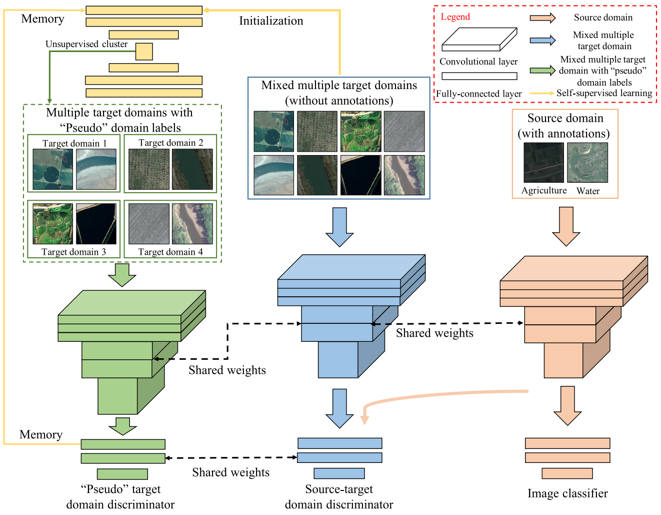

Memory
Initialization
Unsupervised cluster
Legend
Convolutional layer
Fully-connected layer
Source domain
Mixed multiple target domain
Mixed multiple target domain with "pseudo" domain labels
Self-supervised learning
Mixed multiple target domains (without annotations)
Source domain (with annotations)
Agriculture
Water
Multiple target domains with "Pseudo" domain labels
Target domain 1
Target domain 2
Target domain 3
Target domain 4
Shared weights
Shared weights
Memory
Shared weights
"Pseudo" target domain discriminator
Source-target domain discriminator
Image classifier

Fig. 3. Flowchart of our proposed method, i.e., TSAN.

Therefore, we propose a TSAN to resolve the domain gaps among the multiple targets and the category misalignments among these mixed multiple target domains in the scenario of S^2M^2T DA. Fig. 3 shows the framework of our TSAN, including three major parts, i.e., a source–target adaptation, a subtarget adaptation, and a dynamic memory-aware collaboration.

1) *Source–Target Adaptation:* We adopt an adversarial learning algorithm to confuse the discriminator between the source and the whole mixed multitarget dataset.
2) *Subtarget Adaptation:* We first adopt a self-supervised learning approach for preliminarily dividing the mixed multitarget dataset so that each target image will have a "pseudo"-domain class label. Following that, we employ another adversarial learning model to confuse the discriminator among "pseudo"-subtarget domains.
3) A dynamic memory-aware collaboration between the self-supervised learning and the two adversarial learning-based adaptations, so as to train domain-invariant representations and reduce the negative transfer effect from the mixed-multiple-target domain.

B. Source–Target Adaptation Between a Source Domain and a Mixed-Multiple-Target Domain

Adversarial learning methods have been widely and successfully exploited in previous DA studies [12], [14], [15], [30]. They align feature distributions by incorporating a classifier in the source domain with gradient reversal to fool a domain discriminator. For example, the objective of the well-established DANN [12] can be formulated as

$$\begin{aligned} C_{\text{DANN}}(\theta_f, \theta_y, \theta_d) = &\frac{1}{n_s} \sum_{\mathbf{x}_i \in \mathcal{D}_s} L_y(G_y(G_f(\mathbf{x}_i)), y_i) \\ &- \frac{\lambda}{n} \sum_{\mathbf{x}_i \in \mathcal{D}_s \cup \mathcal{D}_t} L_d(G_d(G_f(\mathbf{x}_i)), d_i) \quad (1) \end{aligned}$$

where $n = n_s + n_t$ and λ denote a hyperparameter that balances the domain loss (L_d) and the classification loss (L_y), which is calculated by the classifier of the source domain (G_y). Similar to DANN, we can effectively confuse our source–target domain discriminator G_{st} between source domain images and the whole mixed multiple-target-domain images. To this end, the objective of the source–target domain discriminator is formulated as

$$C_{\text{st}}(\theta_f, \theta_{\text{st}}) = \frac{1}{n} \sum_{\mathbf{x}_i \in \mathcal{D}_s \cup \mathcal{D}_t} L_{\text{st}}(G_{\text{st}}(G_f(\mathbf{x}_i)), d_i) \quad (2)$$

where L_{st} represents the training loss for the source–target domain discriminator G_{st}. d_i is equivalent to one for the source domain images and zero for the mixed multiple-target-domain images. More specifically, the result of $G_{\text{st}}(G_f(\mathbf{x}_i))$ describes the probability of the hidden representation for input data $\mathbf{x}_i$ belonging to the source domain. When the source–target domain discriminator G_{st} cannot distinguish the differences between the source domain ($\mathcal{D}_s$) and the target domain ($\mathcal{D}_t$), the generator G_f wins the min–max game.

C. Subtarget Adaptation Among the Mixed Multiple Target Domains

As the target domain is a mixed dataset, we do not have the metainformation about the location at the source sensor of each image. If we put all the target images from different datasets into a union target dataset, we would usually expect negative transfer effects because of dataset discrepancy and class misalignment among $\{\mathcal{D}_t^j\}_{j=1}^k$. To deal with this problem, we adopt a self-supervised learning network and an unsupervised cluster approach. In the following, Section III-C1 introduces the self-supervised approach, and Section III-C2 presents the unsupervised cluster and the adaptation among the mixed multiple target domains with "pseudo"-labels.

1) Self-Supervised Approach: Self-supervised learning algorithms are designed to learn visual representations from a large amount of data in the absence of any human interpretation [31]. In this context, our multiple domain images are mixed without known domain labels. We adopt self-supervised learning to extract representative features and preliminarily acquire the "pseudo"-domain labels.

Our self-supervised learning approach includes two parts, i.e., a deep autoencoder with reconstruction loss and an auxiliary target distribution for minimizing the Kullback–Leibler (KL) divergence. First, we use autoencoder [32] ($\mathcal{A}$), which contains an encoder ($\mathcal{A}_1$) and decoder ($\mathcal{A}_2$). For initialization, the input of $\mathcal{A}_1$ is the vector of original images $\{(\mathbf{x}_i^t)\}_{i=1}^{n_T}$, while, for following training steps, the input of $\mathcal{A}_1$ is the output of generator $\{G_f(\mathbf{x}_i^t)\}_{i=1}^{n_T}$, where $G_f(\cdot)$ denotes a feature extractor, as presented in Section III-B. To learn the representative features without any annotations, we adopt the self-reconstruction loss (L_{rec}) to train the autoencoder

$$C_{\text{rec}}(\theta_f, \theta_{\mathcal{A}}) = \begin{cases} \dfrac{1}{n_T} \displaystyle\sum_{\mathbf{x}_i \in \mathcal{D}_T} L_{\text{rec}}(\mathcal{A}(\mathbf{x}_i), \mathbf{x}_i), & \text{initial} \\ \dfrac{1}{n_T} \displaystyle\sum_{\mathbf{x}_i \in \mathcal{D}_T} L_{\text{rec}}(\mathcal{A}(G_f(\mathbf{x}_i)), \mathbf{x}_i), & \text{others} \end{cases} \quad (3)$$

in which n_T is the total quantity of input images of mixed multitarget domain dataset in the autoencoder. We use a target feedback pair $(\mathcal{A}(G_f(\mathbf{x}_i)), G_f(\mathbf{x}_i))$ (or $(\mathcal{A}(\mathbf{x}_i), \mathbf{x}_i)$ for initialization) to calculate a l_2 self-reconstruction. This is a self-supervised learning approach that learns visual representations from mixed multiple domain datasets without any domain labels.

On the other hand, to categorize different images into different domains, we have to simultaneously deal with both hidden representation and cluster assignment [33]. Therefore, we add a KL divergence loss between the auxiliary distribution p_{ij} and the soft assignment q_{ij} as follows:

$$L_{\mathrm{KL}} = \mathrm{KL}(P||Q) = \frac{1}{n_T \times k}\sum_{i=1}^{n_T}\sum_{j=1}^{k} p_{ij} \times \log\frac{p_{ij}}{q_{ij}} \tag{4}$$

where the soft assignment q_{ij} denotes the probability of allocating sample $\mathbf{x}_i^t$ to the cluster μ_j, which can be calculated as (5). $\left\{\mu_j\right\}_{j=1}^k$ denotes the cluster centroids that describe the k centers for mixed multitarget domain feature representations. Equation (5) measures the similarity between the hidden features $\mathcal{A}_1(\mathbf{x}_i)$ and the centroid μ_j [34]. The auxiliary distribution p_{ij} pays more attention on data points allocated with higher confidence. p_{ij} can be calculated as (6)

$$q_{ij} = \begin{cases} \dfrac{\left(1+\|\mathcal{A}_1(\mathbf{x}_i)-\mu_j\|^2\right)^{-1}}{\sum_{j'=1}^{k}\left(1+\|\mathcal{A}_1(\mathbf{x}_i)-\mu_{j'}\|^2\right)^{-1}}, & \text{initial,} \\ \dfrac{\left(1+\|\mathcal{A}_1(G_f(\mathbf{x}_i))-\mu_j\|^2\right)^{-1}}{\sum_{j'=1}^{k}\left(1+\|\mathcal{A}_1(G_f(\mathbf{x}_i))-\mu_{j'}\|^2\right)^{-1}}, & \text{others.} \end{cases} \tag{5}$$

$$p_{ij} = \frac{q_{ij}/f_j}{\sum_{j'=1}^{k} q_{ij'}/f_{j'}}, \quad f_j = \sum_{i=1}^{n_T} q_{ij}. \tag{6}$$

Algorithm 1 presents the detailed procedures on our self-supervised learning approach for dividing mixed multiple-target-domain datasets. The objective of self-supervised learning is defined as

$$C_{\mathrm{ss}}\left(\theta_{\mathcal{A}}, \{\mu_j\}_{j=1}^k\right) = C_{\mathrm{rec}}(\theta_f, \theta_{\mathcal{A}}) + \gamma\,\mathrm{KL} \tag{7}$$

where γ is a tradeoff parameter that controls the distorting degree of the embedded space [35], $\theta_{\mathcal{A}}$ are trained by minimizing the loss of L_{rec}, and $\{\mu_j\}_{j=1}^k$ denotes k cluster centroid that updates in each self-supervised learning iteration. After the converges of self-supervised learning, we assign an unsupervised cluster (i.e., K-Means) to assign images a "pseudo"-domain label by dividing $\mathcal{D}_T$ into k subtarget domains. Specifically, $\mathbf{x}_i^t$ in a mixed-multiple-target domain dataset $\mathcal{D}_T = \{(\mathbf{x}_i^t)\}_{i=1}^{n_T}$ is classified into subtarget domain $\{\hat{\mathcal{D}}_t^j\}_{j=1}^k$ if q_{ij} is the maximum in $\{q_{ij'}\}_{j'=1}^k$

$$\forall j \in [k], \hat{\mathcal{D}}_t^j = \left\{\mathbf{x}_i \in \mathcal{D}_T \,\&\, j = \arg\max_{j'} q_{ij'}\right\}. \tag{8}$$

2) Subtarget Adaptation Among the Mixed Multiple Target Domains Using "Pseudo"-Labels: Taking account for "pseudo"-domain labels generated from Section III-C1, and recalling the principle of adaptation between a source domain and a mixed multitarget domain introduced in Section III-B, TSAN employs a k-subtarget domain loss of L_{mt} to maximally confuse the discriminator (G_{mt}) through gradient reversal layer as the same as DANN [12]. It should be emphasized that the parameters are sharable between the softmax classifier of G_{mt} and the binary classifier of G_{st}. Thereby, the objective of the k-subtarget domain discriminator can be formulated as follows:

Algorithm 1 Self-Supervised Learning Approach for Dividing Mixed Multitarget Domain Datasets

```
Input: Mixed multi-target domain dataset $\mathcal{D}_T$. $k$ is the number of sub-target domains that we preliminarily expected. Feature extractor $G_f$ and autoencoder $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2\}$.
Output: Well-trained $\mathcal{A}* = \left\{\mathcal{A}_1^*, \mathcal{A}_2^*\right\}$. "Pseudo" domain labels for mixed multi-target dataset $\hat{\mathcal{D}}_T = \left\{\hat{\mathcal{D}}_t^j\right\}_{j=1}^k$.
Initiate the $k$ unsupervised cluster centroids $\left\{\mu_j\right\}_{j=1}^k$ and the autoencoder $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2\}$.
while not converged do
    Sample a mini-batch $\mathbf{X}_i^t$ from $\mathcal{D}_T$;
    if initialization then
        Acquire hidden feature $\mathcal{A}_1\left(\mathbf{X}_i^t\right)$;
    else
        Acquire hidden feature $\mathcal{A}_1\left(G_f\left(\mathbf{X}_i^t\right)\right)$;
    end if
    Calculate the soft assignment $\left\{q_{ij}\right\}_{j=1}^k$ and the auxiliary distribution $\left\{p_{ij}\right\}_{j=1}^k$ according to Equation (5) and (6), respectively for each $\mathbf{x}_i^t$ in the mini-batch $\mathbf{X}_i^t$;
    Update parameters of $\mathcal{A}$ via minimizing Equation (3), (4) and (7);
end while
return $\mathcal{A}^* = \left\{\mathcal{A}_1^*, \mathcal{A}_2^*\right\}$ and "pseudo" domain labels for mixed multi-target dataset $\mathcal{D}_T = \left\{\mathcal{D}_t^j\right\}_{j=1}^k$ according to Equation (8).
```

$$C_{\mathrm{mt}}(\theta_f, \theta_{\mathrm{mt}}) = \frac{1}{n_T}\sum_{\mathbf{x}_i \in \mathcal{D}_T} L_{\mathrm{mt}}(G_{\mathrm{mt}}(G_f(\mathbf{x}_i)), \hat{d}_i) \tag{9}$$

where $\hat{d}_i \in [1, k]$ is the "pseudo"-domain label for the mixed-multiple-target-domain dataset, derived from self-supervised learning in Section III-C1. θ_{mt} is trained through maximizing the loss L_{mt} so as to confuse the domain discriminator G_{mt}. More specifically, the subtarget adaptation among the mixed multiple target domains facilitates G_f to make multitarget domain more ambiguous so that G_{mt} could not distinguish which exact domain label an example from "pseudo"-subtarget belongs to.

D. Dynamic Collaboration Between Unsupervised Cluster and an Adversarial Learning-Based Adaptation

Finally, in the light of Sections III-B and III-C, our final learning objective is formulated as follows:

$$\begin{aligned} C_{\mathrm{TSAN}}(\theta_f, \theta_y, \theta_{\mathrm{st}}, \theta_{\mathrm{mt}}) &= C_y - \alpha C_{\mathrm{st}} - \beta C_{\mathrm{mt}} \\ &= \frac{1}{n_s}\sum_{\mathbf{x}_i \in \mathcal{D}_s} L_y(G_y(G_f(\mathbf{x}_i)), y_i) \\ &\quad - \frac{\alpha}{n}\sum_{\mathbf{x}_i \in \mathcal{D}_s \cup \mathcal{D}_t} L_{\mathrm{st}}(G_d(G_f(\mathbf{x}_i)), d_i) \\ &\quad - \frac{\beta}{n_T}\sum_{\mathbf{x}_i \in \mathcal{D}_T} L_{\mathrm{mt}}(G_{\mathrm{mt}}(G_f(\mathbf{x}_i)), \hat{d}_i) \end{aligned} \tag{10}$$

TABLE I

DETAILED INFORMATION OF FIVE OPEN-SOURCE REMOTE SENSING DATASETS

Index	UC Merced	AID	NWPU-RESISC45	RSD46-WHU	PatternNet
Year	2010	2017	2017	2017	2018
Classes	21	31	45	46	38
Images per class	100	220-420	700	500-3,000	800
Images	2,100	10,000	31,500	117,000	30,400
Resolution (m)	0.3	0.5-8	0.2-30	0.5-2	0.062-4.693
Size	256 × 256	600 × 600	256 × 256	256 × 256	256 × 256
Source	USGS	Google Earth	Google Earth	Google Earth, Tianditu	Google Earth, GoogleMap

where C_y is the objective for classification of the labeled source domain dataset. λ and β represent the hyperparameters, and d_i represents the domain label of source or multitarget dataset, while $\hat{d}_i$ represents the exact "pseudo"-domain label of subtarget dataset. Given that the numbers of samples in different domains are imbalanced, and the numbers of samples in different classes in a certain domain are also imbalanced for our collected dataset (which will be introduced in Section IV), we adopt the focal loss [36] in L_y, L_{st} and L_{mt} to alleviate the problem of sample imbalance. Furthermore, we will present the elaborate explanation and the ablation study in Section V-C. To this end, the minimax optimization problem is to jointly satisfy the network parameters $\hat{\theta}_f$, $\hat{\theta}_y$, $\hat{\theta}_{\text{st}}$, and $\hat{\theta}_{\text{mt}}$

$$(\hat{\theta}_f, \hat{\theta}_y, \hat{\theta}_{\text{st}}, \hat{\theta}_{\text{mt}}) = \arg\min_{\theta_f,\theta_y}\max_{\theta_{\text{st}},\theta_{\text{mt}}} C_{\text{TSAN}}(\theta_f, \theta_y, \theta_{\text{st}}, \theta_{\text{mt}}). \quad (11)$$

Notably, our self-supervised approach is retrained to update the "pseudo"-domain labels of the mixed multitarget images per M iteration, which is a memory-ware training process. For initialization, we first generate "pseudo"-domain labels using original input images. After that, we generate "pseudo"-domain labels using extracted features from deep learning. In a summary, our self-supervised-based unsupervised cluster and adversarial learning-based adaptation are two collaborative and iterative procedures. However, according to the off-the-shelf alternating adversarial manner [12], we iteratively update $(\hat{\theta}_f, \hat{\theta}_y)$ and $(\hat{\theta}_{\text{st}}, \hat{\theta}_{\text{mt}})$ through switching the optimization objectives between (12) and (13). Algorithm 2 presents the detailed procedures for our proposed method to tackle the S^2M^2T DA issue

$$(\hat{\theta}_f, \hat{\theta}_y) = \arg\min_{\theta_f,\theta_y} C_{\text{TSAN}}(\theta_f, \theta_y) \quad (12)$$

$$(\hat{\theta}_{\text{st}}, \hat{\theta}_{\text{mt}}) = \arg\min_{\theta_{\text{st}},\theta_{\text{mt}}} C_{\text{TSAN}}(\theta_{\text{st}}, \theta_{\text{mt}}). \quad (13)$$

Algorithm 2 TSAN

```
Input: Source domain dataset $\mathcal{D}_s$ and the mixed-multiple-target-domain dataset $\mathcal{D}_T$. Feature extractor $G_f$, classifier $G_y$, source-target domain discriminator $G_{\text{st}}$ and sub-target domain discriminator $G_{\text{mt}}$. $k$ is the number of sub-target domain label and $M$ denotes that after every $M$ steps, the "pseudo" sub-target domain label will be updated through Algorithm 1.
Output: Well-trained $G_f^*$, $G_y^*$, $G_{\text{st}}^*$, $G_{\text{mt}}^*$.
while not converged do
  Employ self supervised approach according to Algorithm 1 and divide $D_T$ into $k$ sub-target domains $\hat{\mathcal{D}}_T = \left\{\hat{\mathcal{D}}_t^j\right\}_{j=1}^k$ with "pseudo" domain labels.
  for 1 : M do
    Sample a mini-batch $X_s$ from $\mathcal{D}_s$ and $k$ mini-batches $\left\{\mathbf{X}_t^j\right\}_{j=1}^k$ from $\hat{\mathcal{D}}_T = \left\{\hat{\mathcal{D}}_t^j\right\}_{j=1}^k$, respectively.
    if iteratively update then
      Update $G_f$, $G_y$ according to Equation (12).
      Update $G_{\text{st}}$ and $G_{\text{mt}}$ according to Equation (13).
    else
      Update $G_f$, $G_y$, $G_{\text{st}}$ and $G_{\text{mt}}$ according to Equation (11).
    end if
  end for
end while
return $G_f^* = G_f$, $G_y^* = G_y$, $G_{\text{st}}^* = G_{\text{st}}$, $G_{\text{mt}}^* = G_{\text{mt}}$.
```

IV. DATASETS

We collect a remote sensing dataset to validate the performance of our proposed method. The dataset is on the basis of five different open-source remote sensing datasets, i.e., UC Merced [37], AID [38], NWPU-RESISC45 [39], RSD46-WHU [8], and PatternNet [19]. Table I elaborately lists the information of five open-source remote sensing datasets. They are derived from different platforms and regions with different resolutions and acquisition dates, which is suitable for validating DA approaches [20], [40], [41].

Our dataset contains ten classes, including agriculture, forest, water, residential, parking, sports court, airfield, overpass, port, and storage tank. We merge some of the common classes into one class for each public dataset. For example, dense residential, medium residential, and sparse residential are collected as residential for AID, NWPU-RESISC45, and UC Merced. Airport and Airplane are merged as Airfield for NWPU-RESIS45 and RSD46-WHU. In addition, we consider similar classes in the same category in our collected dataset. For example, we consider port, harbor, and dock as the same class of port. Overpass and viaduct constitute the same class of overpass. The information on our dataset is listed in Table II elaborately. Fig. 4 illustrates the examples of our collected dataset. Our collected dataset has two major characteristics.

1) Our collected dataset has high within-class diversity. For example, dense residual, medium residual, and sparse residual have tremendous diversity, and they are separated into different classes for the most common datasets. However, we unify these three kinds of residential as one class, making our collected dataset more challenging for classification performance.
2) Our collected dataset has imbalanced samples, especially among different domains. As shown in Fig. 5

TABLE II
DETAILED INFORMATION OF OUR COLLECTED DATASET

Index	UC Merced	AID	NWPU-RESISC45	RSD46-WHU	PatternNet
Agriculture	Farmland	Circular farmland Rectangular farmland	Christmas tree farm	Irregular farmland Regular farmland	Agriculture
Forest	Forest	Forest	Forest	Natural dense forest land Natural sparse forest land	Forest
Water	River Pond	River Lake	River	Fish pond Water	River
Residential	Dense residential Medium residential Sparse residential	Dense residential Medium residential Sparse residential	Dense residential Sparse residential	Sparse residential area	Dense residential Medium residential Sparse residential
Parking	Parking	Parking lot	Parking lot	Parking lot	Parking lot
Sports court	Baseball diamond Playground	Baseball diamond Ground track field	Football field Baseball field	Playground	Baseball diamond
Airfield	Airport	Airport Airplane	Airplane	Airport Airplane	Airplane
Overpass	Viaduct	Overpass	Overpass	Overpass	Overpass
Port	Port	Harbor	Harbor	Dock	Harbor
Storage tank	Storage tank	Storage tank	Storage tank	Oil tank	Storage tank

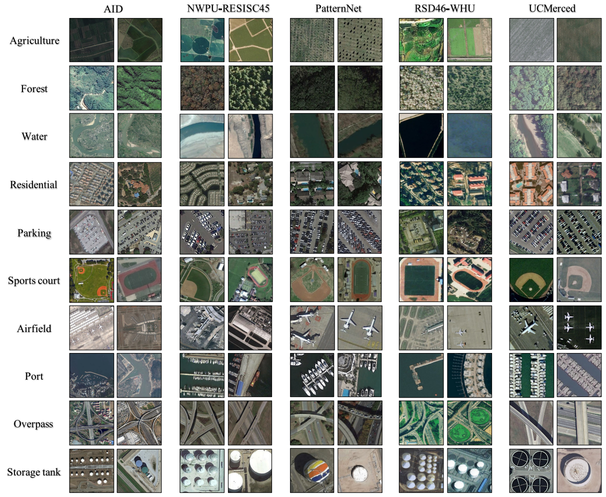


Fig. 4. Examples of our collected dataset. The dataset contains ten classes (agriculture, forest, water, residential, parking, sports court, airfield, overpass, port, and storage tank) from five open-source remote sensing dataset (AID, NWPU-RESISC45, PatternNet, RSD46-WHU, and UC Merced).

and Table II, RSD46-WHU holds the most samples for each class, and UC Merced has the smallest number of samples for each category. Furthermore, the numbers of samples between different classes in a particular domain are also imbalanced. Both of these two imbalanced aspects make our transfer task more challenging and difficult.

V. EXPERIMENTAL RESULTS

A. Experimental Setup

In our experiments, we follow the standard protocols of unsupervised DA in all our experiments [12], [14], [42]. We use all labeled source samples as the training dataset and all unlabeled multidomain target samples as the test dataset to compare the average classification accuracy. We set tradeoff hyperparameters as $\alpha = 1.0$ and $\beta = 0.1$. The backbone architectures in our experiments are AlexNet [43] and ResNet-50 [44] pretrained on the ImageNet dataset. The learning rate is 0.001, and we adopt SGD as our optimizer, following the batch size of 36. Our method TSAN is validated by our collected datasets described in Section IV. We evaluate the accuracies for five types of transfer tasks for S^2M^2T DA and their average accuracy. Although the preset of existing DA methods is S^3T, we directly adopt S^3T DA methods in the S^2M^2T DA scenario in comparison to other DA approaches. We release our codes and collected datasets on https://github.com/rs-dl/TSAN.

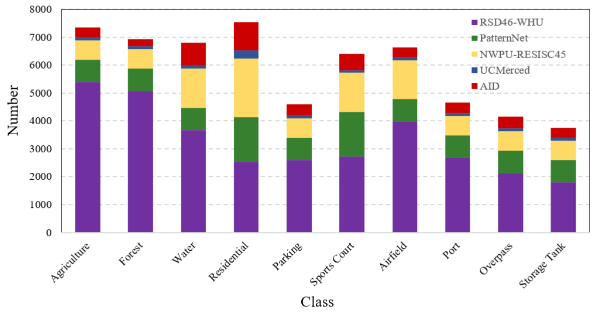


Fig. 5. Distribution of image numbers for ten classes and five domains in our collected dataset. It is observed that our collected dataset has imbalance samples among different domains, and the number of samples between different classes in a certain domain is also imbalanced.

B. Comparisons Between TSAN and Other State-of-the-Art DA Methods

We compare TSAN with the other seven state-of-the-art DA approaches. DDC [17] introduces an adaptation layer and an additional domain confusion loss through maximum mean discrepancy (MMD). DAN [11] embeds all task-specific layers in a reproducing kernel Hilbert space measured by a multikernel selection method. DANN [12] adopts a gradient reversal layer to facilitate adaptation so that the model does not perform well in discriminating between the source domain and the target domain. Deep Coral [13] learns a transfer network through a linear transformation to align the second-order statistics of the source and target distributions. JAN [42] jointly aligns the information of multiple domain-specific layers based on MMD. CDAN [14] is designed with two conditioning strategies, i.e., multilinear conditioning and entropy conditioning. TADA [15] focuses the adaptation model on more transferable regions and images. Notably, we follow

TABLE III

TWO KINDS OF ACCURACIES (%) (I.E., $\mathrm{ACC}_{S^2\ MT}$ AND $\mathrm{ACC}_{S^2\ M^2\ T}$) OF DIFFERENT DA METHODS FOR OUR COLLECTED DATASET (AlexNet). THE BASELINE METHOD DENOTES A STRAIGHTFORWARD AlexNet MODEL

Index	A → {N, P, R, U}		N → {A, P, R, U}		P → {A, N, R, U}		R → {A, N, P, U}		U → {A, N, P, R}		Avg	
	ACC_{S^2MT}	$\mathrm{ACC}_{S^2M^2T}$	ACC_{S^2MT}	$\mathrm{ACC}_{S^2M^2T}$	ACC_{S^2MT}	$\mathrm{ACC}_{S^2M^2T}$	ACC_{S^2MT}	$\mathrm{ACC}_{S^2M^2T}$	ACC_{S^2MT}	$\mathrm{ACC}_{S^2M^2T}$	ACC_{S^2MT}	$\mathrm{ACC}_{S^2M^2T}$
Baseline	63.87	66.53	56.20	63.47	49.41	44.91	56.19	55.21	54.53	54.90	56.04	57.00
DDC	61.47	58.49	61.99	55.49	44.82	43.32	54.07	50.01	46.59	53.34	53.79	52.13
DAN	64.31	62.47	63.29	58.33	51.86	37.50	55.87	44.33	54.66	49.36	58.00	50.40
DANN	65.81	62.00	67.47	58.62	51.76	35.69	59.81	51.58	55.81	50.66	60.13	51.71
Deep Coral	65.98	66.13	63.19	64.49	44.09	44.27	54.84	54.62	55.52	55.44	56.72	56.99
JAN	63.47	52.12	62.23	50.56	55.97	31.62	56.57	39.58	55.57	41.44	58.76	43.06
CDAN	**75.57**	69.18	**76.00**	68.10	**56.15**	49.59	**70.47**	46.61	**56.84**	**58.58**	**67.07**	58.41
TADA	70.44	63.06	70.77	62.86	54.07	37.65	65.79	58.83	55.52	50.97	63.32	54.67
TSAN	–	**70.59**	–	**69.22**	–	**53.86**	–	**64.39**	–	58.20	–	**63.25**

TABLE IV

TWO KINDS OF ACCURACIES (%) (I.E., $\mathrm{ACC}_{S^2\ MT}$ AND $\mathrm{ACC}_{S^2\ M^2\ T}$) OF DIFFERENT DA METHODS FOR OUR COLLECTED DATASET (ResNet-50). THE BASELINE METHOD DENOTES A STRAIGHTFORWARD ResNet-50 MODEL

Index	A → {N, P, R, U}		N → {A, P, R, U}		P → {A, N, R, U}		R → {A, N, P, U}		U → {A, N, P, R}		Avg	
	ACC_{S^2MT}	$\mathrm{ACC}_{S^2M^2T}$	ACC_{S^2MT}	$\mathrm{ACC}_{S^2M^2T}$	ACC_{S^2MT}	$\mathrm{ACC}_{S^2M^2T}$	ACC_{S^2MT}	$\mathrm{ACC}_{S^2M^2T}$	ACC_{S^2MT}	$\mathrm{ACC}_{S^2M^2T}$	ACC_{S^2MT}	$\mathrm{ACC}_{S^2M^2T}$
Baseline	75.98	76.86	70.79	72.28	45.33	46.60	61.12	59.77	58.07	62.05	62.26	63.51
DDC	79.23	74.27	74.45	71.43	51.06	50.53	66.70	60.42	61.97	60.03	66.68	63.34
DAN	71.55	68.40	70.13	71.98	59.44	47.21	65.58	54.45	58.89	63.92	65.12	61.19
DANN	77.34	71.18	73.50	70.20	58.41	47.34	**68.85**	55.64	61.89	57.37	**68.00**	60.35
Deep Coral	77.94	74.22	**74.47**	71.44	51.91	54.15	67.49	60.14	59.97	62.92	66.36	64.57
JAN	72.03	62.95	66.95	67.86	**59.93**	45.09	62.37	53.52	62.90	55.82	64.84	57.05
CDAN	80.25	71.39	72.87	69.39	52.68	45.75	67.80	64.14	**63.30**	58.67	67.38	61.87
TADA	**81.26**	67.09	72.32	69.95	55.03	45.43	63.98	46.97	51.79	53.51	64.88	56.59
TSAN	–	**84.25**	–	**81.87**	–	**60.52**	–	**73.40**	–	**68.10**	–	**73.63**

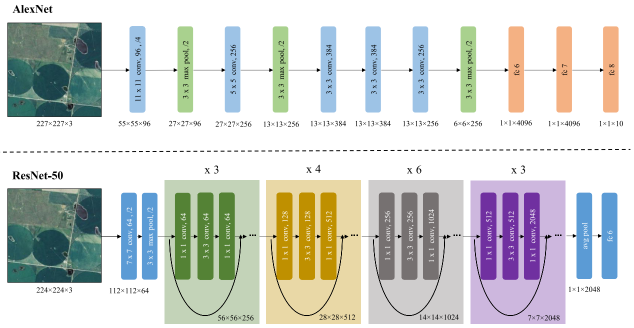


Fig. 6. Network architecture of (Top) AlexNet and (Bottom) ResNet-50.

the same hyperparameters as the above literature. We also list the baseline that leverages only classification loss without any DA approaches in the following tables. We conduct our experiments on two popular architectures, i.e., AlexNet [43] and ResNet-50 [44].AlexNet has only seven CNN layers, while ResNet-50 is far deeper with 50 CNN layers (see Fig. 6).

Tables III and IV list the accuracy of different DA methods for our collected dataset in the S^2MT DA scenario and S^2M^2T DA scenario using two classical architectures (AlexNet and ResNet-50). There two kinds of evaluation approaches, i.e., $\mathrm{ACC}_{S^2\ MT}$ and $\mathrm{ACC}_{S^2\ M^2\ T}$. $\mathrm{ACC}_{S^2\ MT}$ denotes that we view the whole multiple target domains as a mixed target-domain dataset, and we are unknown about the distribution for each target domain. $\mathrm{ACC}_{S^2\ M^2\ T}$ represents that we are known about the distribution for each target domain ($\mathcal{D}_t^j \forall j \in [k]$), which can be indicated by $w^j \in [0, 1]$ and $\sum_{j=1}^{k} w^j = 1$. To this end, we calculate $\mathrm{ACC}_{S^2\ MT}$ by averaging each accuracy of the S^3T transfer task ($\mathrm{ACC}_{S^3\ T}$) according to w^j, as described in (14), which denotes the proportion of the subtarget domain $\mathcal{D}_t^j$ in the mixed-multiple-target dataset $\mathcal{D}_T$. For instance, if we want to calculate the $\mathrm{ACC}_{S^2\ MT}$ of **A** → {**N**, **P**, **R**, **U**}, we average the accuracy of four types of S^3T DA tasks according to their weights, i.e., **A** → **N**, **A** → **P**, **A** → **R**, and **A** → **N**

$$\mathrm{ACC}_{S^2MT} = \sum_{j=1}^{k} w^j \mathrm{ACC}^j_{S^3T}. \tag{14}$$

We compare our proposed TSAN with other DA approaches under two kinds of DA scenarios. First, we compare our TSAN with other DA methods with respect to $\mathrm{ACC}_{S^2\ M^2\ T}$, where we view the whole multiple target domains as a mixed target-domain dataset, and we are unknown about the distribution for each target domain. It is corroborated that our TSAN reaches the highest average accuracy for five transfer tasks compared to other DA methods under S^2M^2T DA scenario, no matter using shallow neural networks or deep CNN architectures, with 63.25% for AlexNet and 73.68% for ResNet-50. Except for the transfer task of **U** → {**A**, **N**, **P**, **R**} in AlexNet, each transfer task delivers superior classification accuracy. TSAN improves the performance by 6.25% and 10.17% with respect to average accuracy compared to the baseline method (without any DA approach) for AlexNet and ResNet-50, respectively. In addition, our proposed TSAN performs 4.84%–20.19% and 9.11%–17.09% better than other DA approaches for $\mathrm{ACC}_{S^2\ M^2\ T}$ under two different architectures.

On the other hand, we compare our TSAN with different DA approaches with respect to $\mathrm{ACC}_{S^2\ MT}$. Although $\mathrm{ACC}_{S^2\ MT}$ is calculated by averaging each accuracy of the S^3T transfer task ($\mathrm{ACC}_{S^3\ T}$), our TSAN performs better than most of the existing DA methods under AlexNet, except for CDAN (−3.76%) and TADA (−0.07%). More surprisingly, our TSAN achieves better average accuracy than other cutting-edge DA approaches under ResNet-50, with 5.63-8.79% improvement. In most cases, the accuracy of $\mathrm{ACC}_{S^2\ MT}$ is higher than $\mathrm{ACC}_{S^2\ M^2\ T}$. The main reason could be that the distribution information of multiple target domains is known for the S^2MT DA scenario, while that is unknown for the S^2M^2T DA scenario. We will present the negative transfer effect to disclose this phenomenon in Section VI-A.

TABLE V

ACCURACIES (%) OF S^2M^2T DA SCENARIOS ($\text{ACC}_{S^2M^2T}$) FOR BASELINE AND TSAN WITH/WITHOUT AUGMENTATION STRATEGY OR SEMISUPERVISED LEARNING (WITH 15% LABELED TARGET DATA IN TRAINING PHASE). THE BASELINE METHOD DENOTES A STRAIGHTFORWARD ResNet-50 MODEL

Index	A → {N, P, R, U}	N → {A, P, R, U}	P → {A, N, R, U}	R → {A, N, P, U}	U → {A, N, P, R}	Avg
Baseline	76.86	72.28	46.60	59.77	62.05	63.51
Baseline + Augmentation	79.07	75.62	50.20	65.40	63.64	66.79
Baseline + Semi-supervised	78.57	78.78	60.39	70.28	65.71	70.75
TSAN	84.25	81.87	60.52	73.40	68.10	73.63
TSAN + Augmentation	82.84	82.02	60.30	73.74	67.94	73.37
TSAN + Semi-supervised	**85.82**	**82.72**	**64.30**	**75.88**	**68.83**	**75.51**

TABLE VI

EFFICIENCY OF DIFFERENT DA METHODS (AlexNet)

Methods	Number of Parameters (M)	GFLOPs	Inference time (ms per image)
Baseline	56.82	0.43	1.54
DDC	56.82	0.43	1.76
DAN	56.82	0.43	2.04
DANN	67.31	0.45	1.80
Deep Coral	56.82	0.43	1.72
JAN	56.82	0.43	1.97
CDAN	67.31	0.45	1.84
TADA	445.27	1.17	1.96
TSAN	382.54	1.03	1.88

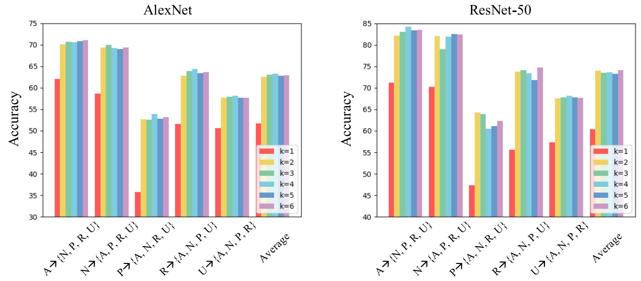


Fig. 7. Accuracy (%) of TSAN for our collected dataset in S^2M^2T DA scenario under different values of k. k denotes the number of subtarget domains in the unsupervised cluster.

Also, we evaluate the impact of data augmentation (horizontal flipping, vertical flipping, and brightness transformation) and semisupervised learning (with 15% labeled target data in the training phase) on the baseline (ResNet-50) and TSAN methods in Table V. Experimental results show that the data augmentation strategy improves the detection accuracy of the baseline method by 3.28%, with little impact on the results of TSAN. Adding limited labeled target data in the training phase can considerably improve the classification accuracy, with +7.24% and +1.88% gains for the baseline method and our proposed TSAN. Furthermore, we list the efficiency of different DA methods in Table VI. Although our method has a larger number of parameters and giga floating point of operations (GFLOPs) compared to other DA approaches except for TADA, the inference time (ms per image) is comparable with other DA methods.

C. Ablation Studies for Our Proposed TSAN

1) Effect of the Unsupervised Domain Cluster: We compare the effectiveness of k for TSAN, and it should be emphasized that $k = 1$ means that the mixed multiple target domains are not separated so that L_{mt} has not been assigned in the procedures. Fig. 7 illustrates the accuracy of TSAN for our collected dataset under different values of k. It is observed that the performance of $k = 1$ is considerably lower than others, probably due to the absence of L_{mt}, indicating that our self-supervised method and unsupervised domain clustering are benefits to S^2M^2T DA problems. $k = 4$ achieves the highest average accuracy for AlexNet with 63.25%, yet $k = 6$ reaches the best result for ResNet-50 with 74.10%. For clarity, the performance of our proposed TSAN is obtained with $k = 4$, equal to the number of multiple target domains.

Toward one probable concern that whether $\hat{\mathcal{D}}_T = \{\hat{\mathcal{D}}_t^j\}_{j=1}^k$ and $\hat{\mathcal{D}}_T = \{\mathcal{D}_t^j\}_{j=1}^k$ are similar, we think that the answer is not. On one hand, the technical difficulty in S2M2T DA arises from the category misalignment instead of the hidden subtarget domains, as introduced in Section III-A. As long as a DA algorithm performs appropriate category alignment in a mixed-multiple-target domain, it is unnecessary to explicitly discover these hidden subtarget domains. On the other hand, our collected dataset is derived from five open-source remote sensing datasets and downloaded from USGS, Google Earth, Tianditu, and so on (as shown in Table I). For example, all images from AID and NWPU-RESIS45 and some images from RSD46-WHU and PatternNet are downloaded from Google Earth. In fact, the real domain label may not be the most suitable for the hidden subtarget domains. Therefore, we do not consider the accuracy of pseudo labels generated from the unsupervised domain cluster.

2) Effect of the Focal Loss: Since our collected dataset has imbalanced samples among different domains and may unavoidably contaminate the performance, as introduced in Section IV, we validate the effectiveness of focal loss for our TSAN. We embed focal loss to the binary cross-entropy loss (L_{st}) and cross-entropy loss (L_y and L_{mt}). Table VII lists the performance of different focal loss strategies. It is observed that focal loss achieves $0.44 \sim 3.86\%$ improvement compared to "None" (without a focal loss for any loss function) for AlexNet. We consider that adopting focal loss in all three kinds of loss functions gains the best result. To this end, we employ focal loss for all loss functions in our proposed TSAN instead of binary cross-entropy loss and cross-entropy loss.

3) Sensitive Analysis: Fig. 8 illustrates the results of TSAN for our collected dataset under two hyperparameters of α and β for AlexNet. When $\beta = 0.1$, we evaluate different values of α ranging from 0.1 to 2.0. We observe that $\alpha = 1.0$ performs better than others with a slight improvement. When $\alpha = 0.1$, we evaluate different values of β ranging from 0.01 to 2.0.

TABLE VII

ACCURACY (%) OF TSAN FOR OUR COLLECTED DATASET IN S^2M^2T DA SCENARIO BY DIFFERENT FOCAL LOSS STRATEGIES (AlexNet)

Index	**A → {N, P, R, U}**	**N → {A, P, R, U}**	**P → {A, N, R, U}**	**R → {A, N, P, U}**	**U → {A, N, P, R}**	**Avg**
None	67.15	67.33	48.77	57.55	56.13	59.39
F_L_y	70.20	68.36	46.13	62.54	57.82	61.01
F_L_{st}	69.82	68.05	45.96	61.92	57.57	60.66
F_L_{mt}	70.33	67.46	43.86	61.58	55.92	59.83
$F\_L_y + F\_L_{st}$	70.49	**69.41**	50.31	62.65	55.22	61.62
$F\_L_y + F\_L_{mt}$	69.32	69.27	48.83	62.47	55.84	61.15
$F\_L_{st} + F\_L_{mt}$	70.54	68.25	44.22	61.24	56.49	60.15
$F\_L_y + F\_L_{st} + F\_L_{mt}$	**70.59**	69.22	**53.86**	**64.39**	**58.20**	**63.25**

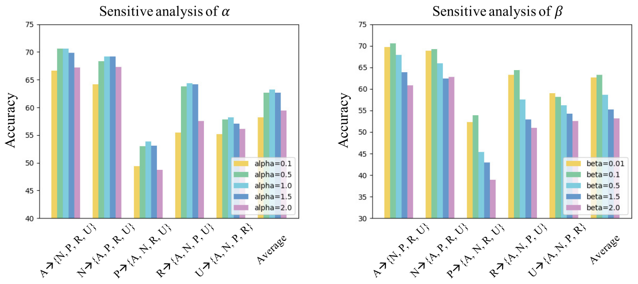


Fig. 8. Accuracy (%) of TSAN for our collected dataset in S^2M^2T DA scenario under different hyperparameters of α and β (AlexNet).

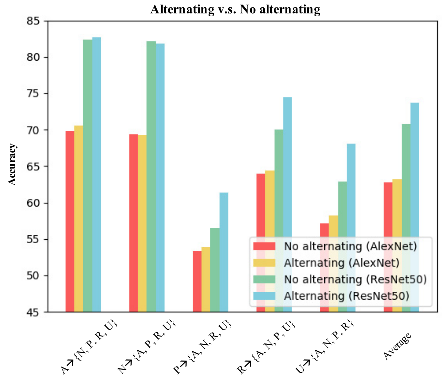


Fig. 9. Accuracy (%) of TSAN for our collected dataset in S^2M^2T DA scenario with/without alternatively updating parameters.

It is evident that, when the $\beta > 0.1$, the accuracies drop dramatically. To this end, we set the hyperparameters α and β as 1.0 and 0.1 in all our experiments. Furthermore, we compare two different adversarial training manners: alternating versus no alternating. As shown in Fig. 9, we observe that there is only a slight improvement in switching the optimization parameters between $(\hat{\theta}_f, \hat{\theta}_y)$ and $(\hat{\theta}_{\mathrm{st}}, \hat{\theta}_{\mathrm{mt}})$ under AlexNet. However, as for ResNet-50, alternating adversarial manner attains approximately 3% gains compared to simultaneously update all parameters. Therefore, we select the alternating training manner in our proposed TSAN.

VI. DISCUSSION

A. Negative Transfer Effect

In this part, we propose two types of negative transfer effect proxies, i.e., the relative negative transfer effect (RNTE) and the absolute negative transfer effect (ANTE), and deep analysis of the negative transfer effect for the abovementioned DA methods in the S^2M^2T DA problem. ANTE describes the transferability of the S^2M^2T model compared to the baseline method (i.e., AlexNet and ResNet), which only uses the source dataset without any DA approach. To this end, the ANTE can be calculated as

$$\mathrm{ANTE} = \mathrm{ACC}_{S^2M^2T} - \mathrm{ACC}_{\mathrm{Baseline}}. \tag{15}$$

If ANTE > 0, it means that the DA method performs a positive transfer effect in the S^2M^2T DA scenario. Conversely, ANTE < 0 denotes that the DA method exhibits a negative transfer effect in the S^2M^2T DA scenario without any improvement. Tables VIII and IX list the performance of ANTE for all DA methods above. It is confirmed that our proposed method TSAN effectively obviates the negative transfer effect, with 6.25% (AlexNet) and 10.17% (ResNet-50) improvement compared to baseline. Unfortunately, although employing DA algorithms, most DA methods have serious negative transfer effects (except for CDAN in AlexNet and Deep Coral in ResNet-50). To this end, directly adopting S^3T DA approaches (view the whole mixed-multiple-target domain as a target domain) will lead to relatively severe deficiency in the S^2M^2T DA scenario. The results also prove the necessity for our strategy of preliminarily dividing the mixed multitarget dataset using the self-supervised learning method and confusing the mixed multitarget discriminator using the adversarial learning method.

RNTE describes the comparison between S^2MT DA and S^2M^2T DA scenarios, using the same DA methods. In S^2MT DA, we are known about the distribution for each target domain ($\mathcal{D}_t^j \forall j \in [k]$), which can be indicated by $w^j \in [0, 1]$ and $\sum_{j=1}^{k} w^j = 1$. Therefore, we report the accuracy of S^2MT by averaging each transfer task according to w^j, which can be followed as in 14. Therefore, the RNTE can be formulated as

$$\mathrm{RNTE} = \mathrm{ACC}_{S^2M^2T} - \mathrm{ACC}_{S^2MT}. \tag{16}$$

The performance of RNTE is tabulated on Tables VIII and IX under AlexNet and ResNet-50. We can observe that the baseline retains the accuracy between S^2MT and S^2M^2T, even with a slight improvement of 0.96% and 1.25% for AlexNet and ResNet-50, respectively. As for other DA algorithms, it is evident that the performance of S^2M^2T dramatically drops by 1.78%–8.29% for ResNet-50 and 1.66%–15.70% for AlexNet (except for Deep Coral). In addition, we find that Deep Coral [13] is less susceptible to negative transfer effects compared to other DA methods. As a consequence, we can conclude that directly adopting common S^3T DA methods in the S^2M^2T DA problem leads to seriously degraded performance.

TABLE VIII

ANTE AND RNTE (%) OF DIFFERENT DA METHODS FOR OUR COLLECTED DATASET IN S^2M^2T DA SCENARIO (AlexNet). THE RED NUMBERS DENOTE THE ANTE < 0 OR RNTE < 0

Index	A → {N, P, R, U}		N → {A, P, R, U}		P → {A, N, R, U}		R → {A, N, P, U}		U → {A, N, P, R}		Avg	
	ANTE	RNTE	ANTE	RNTE	ANTE	RNTE	ANTE	RNTE	ANTE	RNTE	ANTE	RNTE
Baseline	–	+2.66	–	+7.27	–	-4.50	–	-0.98	–	+0.37	–	+0.96
DDC	-8.04	-2.98	-7.98	-6.50	-1.59	-1.50	-5.20	-4.06	-1.56	+6.75	-4.87	-1.66
DAN	-4.06	-1.84	-5.14	-4.96	-7.41	-14.36	-10.88	-11.54	-5.54	-5.30	-6.61	-7.60
DANN	-4.53	-3.81	-4.85	-8.85	-9.22	-16.07	-3.63	-8.23	-4.24	-5.15	-5.29	-8.42
Deep Coral	-0.40	+0.15	+1.02	+1.30	-0.64	+0.18	-0.59	-0.22	+0.54	-0.08	-0.01	+0.27
JAN	-14.41	-11.35	-12.91	-11.67	-13.29	-24.35	-15.63	-16.99	-13.46	-14.13	-13.94	-15.70
CDAN	+2.65	-11.07	+4.63	-4.77	+4.68	-3.09	-8.60	-21.19	+3.68	-4.72	+1.41	-8.97
TADA	-3.47	-7.38	-0.61	-7.91	-7.26	-16.42	+3.62	-6.96	-3.93	-4.55	-2.33	-8.64
TSAN	+4.06	–	+5.75	–	+8.95	–	+9.18	–	+3.30	–	+6.25	–

TABLE IX

ANTE AND RNTE (%) OF DIFFERENT DA METHODS FOR OUR COLLECTED DATASET IN S^2M^2T DA SCENARIO (ResNet-50). THE RED NUMBERS DENOTE THE ANTE < 0 OR RNTE < 0

Index	A → {N, P, R, U}		N → {A, P, R, U}		P → {A, N, R, U}		R → {A, N, P, U}		U → {A, N, P, R}		Avg	
	ANTE	RNTE	ANTE	RNTE	ANTE	RNTE	ANTE	RNTE	ANTE	RNTE	ANTE	RNTE
Baseline	–	+0.88	–	+1.49	–	+1.27	–	-1.35	–	+3.98	–	+1.25
DDC	-2.59	-4.96	-0.85	-3.02	+3.93	-0.53	-0.65	-6.28	-2.02	-1.94	-0.18	-3.35
DAN	-8.46	-3.15	-0.30	+1.85	+0.61	-12.23	-5.32	-11.13	+1.87	+5.03	-2.32	-3.93
DANN	-5.68	-6.16	-2.08	-3.30	+0.74	-11.07	-4.13	-13.21	-4.68	-4.52	-3.17	-7.65
Deep Coral	-2.64	-3.72	-0.84	-3.03	+7.55	+2.24	+0.37	-7.35	+0.87	+2.95	+1.06	-1.78
JAN	-13.91	-9.08	-4.42	+0.91	-1.51	-14.84	-6.25	-8.85	-6.23	-7.08	-6.46	-7.79
CDAN	-5.47	-4.18	-2.90	-6.62	-0.85	-10.40	+4.37	-6.33	-3.38	+1.83	-1.65	-5.14
TADA	-9.77	-14.17	-2.33	-2.37	-1.17	-9.60	-12.80	-17.01	-8.54	+1.72	-6.92	-8.29
TSAN	+5.82	–	+9.58	–	+14.73	–	+14.67	–	+6.05	–	+10.17	–

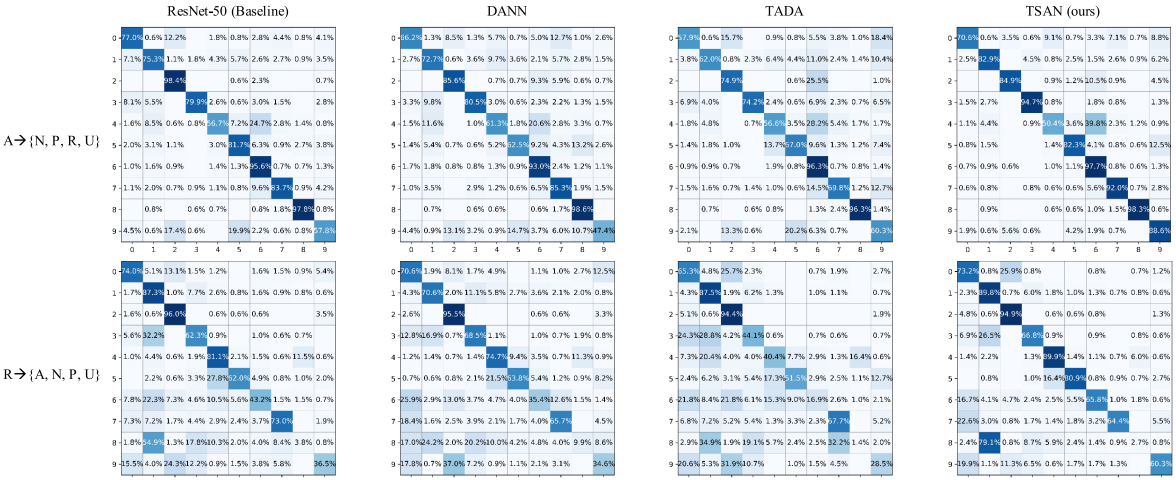


Fig. 10. Confusion matrices of different methods for our collected dataset in S^2M^2T DA scenarios (ResNet-50). We display four methods, including ResNet-50 (Baseline), DANN, CDAN, TADA, and TSAN (ours). The deeper the color is, the higher the percentage is. The numbers 0–9 denote ten different classes in our collected dataset (i.e., agriculture, airfield, forest, overpass, parking, port, residential, sports court, storage tank, and water).

B. Confusion Matrices and Feature Visualization

Fig. 10 shows the confusion matrices of different methods for our collected dataset in two S^2M^2T DA scenarios under ResNet-50. The deeper the color is, the higher the percentage is. Our TSAN achieves considerably less confusion than other state-of-the-art approaches. However, our proposed TSAN is not always better than other DA approaches and has worse performance in some cases. We can also observe that there are severe confusions that exist among different classes. For example, parking and residential are usually misclassified in all transfer tasks except for **R** → {**A, N, P, U**} [see Fig. 11(a)]. There are two possible reasons. The one is that the parking area usually includes some buildings, which is very similar to the sparse residential or medium residential. The other is that the shapes of buildings in dense residential images are similar to the shapes of vehicles at a low spatial resolution. Besides that, pond and water [see Fig. 11(b)], sports court, and agriculture [see Fig. 11(c)] are two pairs that are really confusing to identify for deep learning models.

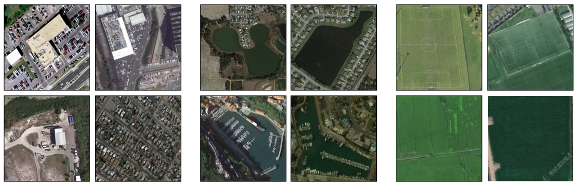

(a) The confusion between parking (top) and residential (bottom)
(b) The confusion between pond (top) and water (bottom)
(c) The confusion between sports court (top) and agriculture (bottom)

Fig. 11. Confusion examples between different classes. (a) (Top) Parking and (Bottom) residential. (b) (Top) Pond and (Bottom) water. (c) (Top) Sports court and (Bottom) agriculture.

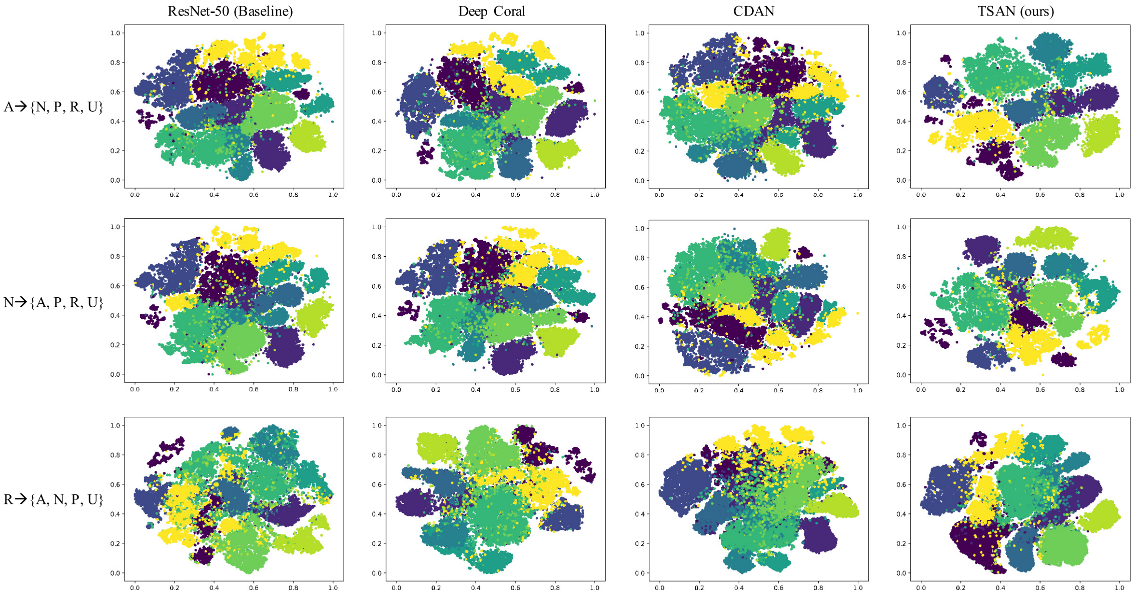


Fig. 12. t-SNE visualization of features for three transfer tasks (from Top to Bottom: **A** → {**N**, **P**, **R**, **U**}, **N** → {**A**, **P**, **R**, **U**}, and **R** → {**A**, **N**, **P**, **U**}) learned by the Baseline (ResNet-50), Deep Coral, CDAN, and TSAN (ours) (from Left to Right).

To display the feature transferability, we visualize the network representations of the last convolutional layer from three transfer tasks in Fig. 12 using t-SNE visualization [34]. We can observe that the features for the target domain by our proposed TSAN are the most distinguishable. The better visualization results of our method indicate that our strategies are able to learn more transferable features and eliminate the negative transfer effect for S^2M^2T DA scenarios.

VII. Conclusion

In this context, we define the "single-source-mixed-multiple-target DA (S^2M^2T DA)" issue for remote sensing image classification and propose a novel algorithm named TSAN. First, we separate the multitarget domain images via a self-supervised learning approach. Following that, we adopt adversarial learning to confuse the discriminator to distinguish among "pseudo"-subtarget domains, facilitating the model to learn the intrinsic features for remote sensing images. Second, we adopt another adversarial learning model to confuse the classifier to discriminate between the source domain images and the whole target domain images. Notably, the self-supervised-based classification and the adversarial learning are dynamically iterative processes. We collect five different open-source remote sensing image classification datasets and select ten classes to demonstrate the effectiveness of our approach. TSAN achieves an average accuracy of 63.25% and 73.68% under AlexNet and ResNet-50, respectively. Our TSAN outperforms other DA approaches by a considerable margin, with 4.84%–20.19% improvements when using the AlexNet backbone and 9.06%–17.04% improvements when using the ResNet-50 backbone. We also identify and analyze that other state-of-the-art DA algorithms, having serious negative transfer effects when we adopt S^3T DA algorithms to address the S^2M^2T DA problem. Experimental results indicate our method is promising for large-scale, multiregional, and multitemporal remote sensing applications.

References

[1] X. X. Zhu *et al.*, "Deep learning in remote sensing: A comprehensive review and list of resources," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 8–36, Dec. 2017.

[2] J. Zheng *et al.*, "Growing status observation for oil palm trees using unmanned aerial vehicle (UAV) images," *ISPRS J. Photogramm. Remote Sens.*, vol. 173, pp. 95–121, Mar. 2021.

[3] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[4] X. Peng, Z. Huang, X. Sun, and K. Saenko, "Domain agnostic learning with disentangled representations," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 5102–5112.

[5] B. Gholami, P. Sahu, O. Rudovic, K. Bousmalis, and V. Pavlovic, "Unsupervised multi-target domain adaptation: An information theoretic approach," *IEEE Trans. Image Process.*, vol. 29, pp. 3993–4002, 2020.

[6] D. Tuia, D. Marcos, and G. Camps-Valls, "Multi-temporal and multi-source remote sensing image classification by nonlinear relative normalization," *ISPRS J. Photogramm. Remote Sens.*, vol. 120, pp. 1–12, Oct. 2016.

[7] G.-S. Xia *et al.*, "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3974–3983.

[8] Z. Xiao, Y. Long, D. Li, C. Wei, G. Tang, and J. Liu, "High-resolution remote sensing image retrieval based on CNNs from a dimensional perspective," *Remote Sens.*, vol. 9, no. 7, p. 725, Jul. 2017.

[9] Z. Chen, J. Zhuang, X. Liang, and L. Lin, "Blending-target domain adaptation by adversarial meta-adaptation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2248–2257.

[10] Z. Liu *et al.*, "Open compound domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12406–12415.

[11] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 97–105.

[12] Y. Ganin *et al.*, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, May 2015.

[13] B. Sun and K. Saenko, "Deep CORAL: Correlation alignment for deep domain adaptation," in *Computer Vision—ECCV 2016 Workshops*. Cham, Switzerland: Springer, 2016, pp. 443–450.

[14] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 1647–1657.

[15] X. Wang, L. Li, W. Ye, M. Long, and J. Wang, "Transferable attention for domain adaptation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 5345–5352.

[16] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A DASVM classification technique and a circular validation strategy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 770–787, May 2010.

[17] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," 2014, *arXiv:1412.3474*. [Online]. Available: http://arxiv.org/abs/1412.3474

[18] D. Tuia, C. Persello, and L. Bruzzone, "Domain adaptation for the classification of remote sensing data: An overview of recent advances," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 41–57, Jun. 2016.

[19] X. Zhou and S. Prasad, "Deep feature alignment neural networks for domain adaptation of hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 10, pp. 5863–5872, Oct. 2018.

[20] R. Zhu, L. Yan, N. Mo, and Y. Liu, "Semi-supervised center-based discriminative adversarial learning for cross-domain scene-level land-cover classification of aerial images," *ISPRS J. Photogramm. Remote Sens.*, vol. 155, pp. 72–89, Sep. 2019.

[21] J. Lin, T. Yu, L. Mou, X. Zhu, R. K. Ward, and Z. J. Wang, "Unifying top–down views by task-specific domain adaptation," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 4689–4702, Jun. 2021.

[22] X. Ma, X. Mou, J. Wang, X. Liu, J. Geng, and H. Wang, "Cross-dataset hyperspectral image classification based on adversarial domain adaptation," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4179–4190, May 2021.

[23] J. Zheng *et al.*, "Unsupervised mixed multi-target domain adaptation for remote sensing images classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Sep. 2020, pp. 1381–1384.

[24] W. Liu and F. Su, "Unsupervised adversarial domain adaptation network for semantic segmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 11, pp. 1978–1982, Nov. 2020.

[25] G. Mateo-García, V. Laparra, D. López-Puigdollers, and L. Gómez-Chova, "Transferring deep learning models for cloud detection between Landsat-8 and Proba-V," *ISPRS J. Photogramm. Remote Sens.*, vol. 160, pp. 1–17, Feb. 2020.

[26] P. Shamsolmoali, M. Zareapoor, H. Zhou, R. Wang, and J. Yang, "Road segmentation for remote sensing images using adversarial spatial pyramid networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 4673–4688, Jun. 2021.

[27] Y. Koga, H. Miyazaki, and R. Shibasaki, "A method for vehicle detection in high-resolution satellite images that uses a region-based object detector and unsupervised domain adaptation," *Remote Sens.*, vol. 12, no. 3, p. 575, Feb. 2020.

[28] J. Zheng *et al.*, "Cross-regional oil palm tree counting and detection via a multi-level attention domain adaptation network," *ISPRS J. Photogramm. Remote Sens.*, vol. 167, pp. 154–177, Sep. 2020.

[29] J. Zheng, W. Wu, S. Yuan, H. Fu, W. Li, and L. Yu, "Multisource-domain generalization-based oil palm tree detection using very-high-resolution (VHR) satellite images," *IEEE Geosci. Remote Sens. Lett.*, early access, Mar. 9, 2021, doi: 10.1109/LGRS.2021.3061726.

[30] W. Wu, J. Zheng, H. Fu, W. Li, and L. Yu, "Cross-regional oil palm tree detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 56–57.

[31] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, May 4, 2020, doi: 10.1109/TPAMI.2020.2992393.

[32] P. Vincent *et al.*, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 1–38, 2010.

[33] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 478–487.

[34] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[35] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1753–1759.

[36] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[37] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst. (GIS)*, 2010, pp. 270–279.

[38] G.-S. Xia *et al.*, "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Jul. 2017.

[39] G. Cheng, J. Han, and X. Lu, "Remote sensing image scene classification: Benchmark and state of the art," *Proc. IEEE*, vol. 105, no. 10, pp. 1865–1883, Oct. 2017.

[40] X. Lu, T. Gong, and X. Zheng, "Multisource compensation network for remote sensing cross-domain scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 2504–2515, Apr. 2020.

[41] R. Adayel, Y. Bazi, H. Alhichri, and N. Alajlan, "Deep open-set domain adaptation for cross-scene classification based on adversarial learning and Pareto ranking," *Remote Sens.*, vol. 12, no. 11, p. 1716, May 2020.

[42] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2208–2217.

[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25. Stateline, NV, USA, Dec. 2012, pp. 1097–1105.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.