# Multitarget Domain Adaptation for Remote Sensing Classification Using Graph Neural Network

Sudipan Saha, *Member, IEEE*, Shan Zhao, and Xiao Xiang Zhu, *Fellow, IEEE*

***Abstract*— Remote sensing deals with huge variations in geography, acquisition season, and a plethora of sensors. Considering the difficulty of collecting labeled data uniformly representing all scenarios, data-hungry deep learning models are often trained with labeled data in a source domain that is limited in the above-mentioned aspects. Domain adaptation (DA) methods can adapt such model for applying on target domains with different distributions from the source domain. However, most remote sensing DA methods are designed for single-target, thus requiring a separate target classifier to be trained for each target domain. To mitigate this, we propose multitarget DA in which a single classifier is learned for multiple unlabeled target domains. To build a multitarget classifier, it may be beneficial to effectively aggregate features from the labeled source and different unlabeled target domains. Toward this, we exploit coteaching based on the graph neural network that is capable of leveraging unlabeled data. We use a sequential adaptation strategy that first adapts on the easier target domains assuming that the network finds it easier to adapt to the closest target domain. We validate the proposed method on two different datasets, representing geographical and seasonal variation. Code is available at https://gitlab.lrz.de/ai4eo/da-multitarget-gnn/.**

***Index Terms*— Coteaching, domain adaptation (DA), graph neural network (GNN), multimodal learning, multitarget adaptation.**

## I. Introduction

Most deep learning-based methods assume that the training data and test data are drawn from the same distribution. However, such assumption often does not hold in remote sensing as differences are induced by geographic variation, differences in acquisition season, and sensor. There are works related to domain adaptation (DA) [1] that try to align the target distribution with the source distribution. Most DA methods adapt a single unlabeled target from a single labeled source domain. Such methods include those based on generative modeling [2], adversarial training [3], and statistical alignment [4], [5]. Such models are not suitable for practical setting in remote sensing as we may come across many target domains and separate model needs to be trained for each target domain. E.g., if the training data consists of images corresponding to a city, every other city can be considered as a different domain. Recently, some works in the computer vision literature have addressed this issue by designing methods to adapt to multiple target domains simultaneously from a single source domain [6], a setting called as multitarget domain adaptation (MTDA).

In multitarget setting, it is important to learn a classifier that generalizes across multiple target domains. Given the intrinsic nature of this task, we argue that learning robust features in a unified space may be beneficial. Thus, feature aggregation can be a suitable direction for multitarget adaptation. Toward this, graph neural networks (GNNs) have been found effective [7]. This motivated us to design a GNN-based incremental approach for MTDA in the context of remote sensing image classification. A GNN with episodic training is integrated to mitigate underlying domain shifts and the adversarial learning is further adopted to close the gap between the source and target distributions. Inspired by coteaching [8] that can exploit noisy labels, we design a dual-head classifier network that consists of a base feature extractor followed by two classifier heads, a multilayer perceptron (MLP) head and a GNN head. The network is first trained with the labeled source samples. Following this, one target domain at a time is processed to further train the network, ordered by their level of difficulty. Training is accomplished with mini-batches comprising of samples from both source domain and target domains. While the MLP-based classifier focuses on individual samples, the GNN-based classifier aggregates feature from different samples in the minibatch. Similar to coteaching [8], they help each other in an iterative manner to learn more effective target classifier. An incremental training scheme aligns conditional distributions across domains by gradually obtaining pseudolabeled target data. Class-specific representations are further learned by aggregating the source and target features and passing through deep GNNs.

Manuscript received November 16, 2021; revised January 7, 2022; accepted January 16, 2022. Date of publication February 7, 2022; date of current version March 1, 2022. This work was supported in part by the German Federal Ministry of Education and Research (BMBF) in the framework of the international future AI Laboratory "AI4EO—Artificial Intelligence for Earth Observation: Reasoning, Uncertainties, Ethics and Beyond" under Grant 01DD20001, in part by the German Federal Ministry of Economics and Technology in the framework of the "National Center of excellence ML4Earth" under Grant 50EE2201C, in part by the European Research Council (ERC) through the European Union's Horizon 2020 Research and Innovation Programme under Grant [ERC-2016-StG-714087], Acronym: *So2Sat*, in part by the Helmholtz Association through the Framework of Helmholtz AI under Grant ZT-I-PF-5-01—Local Unit "Munich Unit at Aeronautics, Space and Transport (MASTr)," and in part by the Helmholtz Excellent Professorship "Data Science in Earth Observation—Big Data Fusion for Urban Research" under Grant W2-W3-100. *(Corresponding author: Xiao Xiang Zhu.)*

Sudipan Saha and Shan Zhao are with the Department of Aerospace and Geodesy, Data Science in Earth Observation, Technical University of Munich, 85521 Ottobrunn, Germany (e-mail: sudipan.saha@tum.de; shan.zhao@tum.de).

Xiao Xiang Zhu is with the Department of Aerospace and Geodesy, Data Science in Earth Observation, Technical University of Munich, 85521 Ottobrunn, Germany, and also with the Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), 82234 Weßling, Germany (e-mail: xiaoxiang.zhu@dlr.de).

Digital Object Identifier 10.1109/LGRS.2022.3149950

The proposed MLP-GNN classifier is seamlessly equipped with a domain discriminator, which further closes the gap between the source and target distributions.

The contributions of this work are as follows.

1) We propose a GNN-based method for MTDA that starts by learning a classifier on the source domain and incrementally updates it on the target domains.
2) We introduce the coteaching [7], [8] in the context of remote sensing DA.
3) We experimented on two multitarget scenarios, one with geographic variation (multiple cities) and the other with season variation (multiple seasons). The datasets are derived from LCZ42 dataset [9] and Sen12-MS dataset [10], respectively. Our experiments indicate the efficacy of the proposed method.

We present the related works in Section II and the proposed method in Section III. Experimental results are presented in Section IV. Finally, the letter is concluded in Section V.

## II. Related Works

In computer vision, different strategies are adopted for DA, e.g., adversarial training [11], [12] and discrepancies measurement minimization [13]. There are several problem settings according to the alignment of label space of domains, e.g., open-set, close-set, and universal DA [14], or the number of sources and targets, e.g., single-source-single-target, multisource [15], [16], and MTDA [17], [18].

MTDA is to learn a robust predictor for all the target domains given one labeled source dataset and multiple unlabeled target datasets that differ in data distributions [6], [17]. Few single-target DA methods can be applied in the multitarget setting. Multiteacher MTDA (MT-MTDA) proposed by Nguyen-Meidine *et al.* [6] used knowledge distillation to iteratively distill domain knowledge from multiple target domains to a common classifier.

Adapting the source classifier to each target domain sequentially for multitarget adaptation is related to incremental learning. Incremental learning [19] gradually extends the existing model's capacity by digesting sequentially available data for the upcoming new tasks. GNNs have the advantage over other neural networks by allowing to capture the interaction between data. Nodes of graphs can represent objects/images and the relationship between nodes are encoded in edges. GNNs can leverage unlabeled data to improve the performance of supervised learning by label propagation and message passing [7], [20]. Many DA problems have also been addressed by hierarchical [21], active learning [22], and semisupervised [23] methods.

Coteaching is a deep learning paradigm first introduced in [8] that simultaneously trains two deep neural networks and let them teach each other.

## III. Proposed Method

Let us assume, we have a labeled source domain dataset $\mathcal{S}$ containing $n_s$ labeled samples $(\mathbf{x}_{s,i}, y_{s,i})_{i=1}^{n_s}$. Our goal is to exploit this dataset to train a classifier for $N$ target datasets $\mathcal{T} = \{\mathcal{T}_j\}_{j=1}^{N}$ each having $n_{tj}$ unlabeled samples $(\mathbf{x}_{tj,k}, y_{tj,k})_{k=1}^{n_{tj}}$. We assume that source domain has sufficient samples to train an initial model (Section III-B). Furthermore, we do not assume any labeled samples from the target domains, however, we assume that the target domains share the same label space as source domain consisting of $n_c$ classes.

TABLE I

Network Architecture Assuming Batch-Size $\mathcal{B}$. Conv(a, b, c) Denotes a Convolutional Filter With $a$ Input Features, $b$ Output Features, and Kernel Size $c \times c$

| Network component | Architecture | Output |
|---|---|---|
| Feature extractor($F$) | Resnet-18 excluding FC layer | $\mathcal{B} \times 256$ |
| MLP classifier($G_{mlp}$) | FC layer | $\mathcal{B} \times n_c$ |
| Edge network($f_{edge}$) | Conv(256,256,1),<br>Conv(256,128,1),<br>Conv(128,1,1) | $\mathcal{B} \times \mathcal{B}$ |
| Node classifier($f_{node}$) | Conv(512, 2 $\times$ $n_c$,1),<br>Conv(2 $\times$ $n_c$,$n_c$,1) | $\mathcal{B} \times n_c$ |

### A. Model Key Components

Given an input image $x$, a feature extractor base network $F$ is used to extract the features from the input image $\mathbf{f} = F(\mathbf{x})$. Extracted features are fed simultaneously to a MLP classifier $G_{\text{mlp}}$ and a GNN-based network $G_{\text{gnn}}$. The $G_{\text{gnn}}$ consists of an edge-network $f_{\text{edge}}$ and a node classifier $f_{\text{node}}$. In addition to the $G_{\text{mlp}}$ and $G_{\text{gnn}}$, a domain discriminator network $D$ is used. The weights of networks $F$, $D$, $G_{\text{mlp}}$, $f_{\text{edge}}$, and $f_{\text{node}}$ are represented as $\theta$, $\psi$, $\phi$, $\varphi$, and $\varphi'$, respectively.

We use Resnet-18 as feature extractor, however, any other suitable model can be used. $G_{\text{mlp}}$ consist of a fully connected (FC) layer mapping 256-dimensional output from $F$ to $n_c$-dimensional output. The $f_{\text{edge}}$ and $f_{\text{node}}$ consist of series $1 \times 1$ convolutional layers (similar to [20]). Network key components are outlined in Table I.

### B. Pretraining on Source Dataset

The model is pretrained for $K_{\text{source}}$ iterations using only the source (labeled) samples $(\mathbf{x}_{s,i}, y_{s,i})_{i=1}^{n_s}$. Cross-entropy loss $\ell_{\text{ce}}^{\text{mlp}}$ computed on the source samples is used to train $F$ and $G_{\text{mlp}}$ and thus updating $\theta$ and $\phi$, respectively.

### C. Adaptation on Targets and Incremental Learning

The model's generalization ability to the target domains is progressively improved by incorporating more target samples in each step. An incremental learning approach is adopted by processing one target domain at a time, in the order of difficulty levels. The easier target domains are processed first followed by the harder ones to avoid potential negative transfer. The level of domain difficulty can be measured by the dissimilarity to the source domain. In our case, the entropy ($H(\mathcal{T}_j)$ for domain $\mathcal{T}_j$) of the predictions returned by the source-trained model is used to determine the level of difficulty of the target domain [7]. $H(\mathcal{T}_j)$ is computed as mean of cross-entropy for all samples belonging to domain $\mathcal{T}_j$.

The labeled samples are denoted by $\hat{\mathcal{S}}^{\text{total}}$ and initially consists of only samples from the source domain dataset $\mathcal{S}$.

Once one target domain (out of $N$ domains) to be processed is fixed, the adaptation is performed for $K$ iterations on that domain. In each iteration, source samples ($\hat{\mathcal{B}}_s^n$) and target samples ($\hat{\mathcal{B}}_t^n$) are drawn to form a mini-batch. Each mini-batch of images is fed to the feature extractor $F$ to obtain features corresponding to them and then fed to the $G_{\text{mlp}}$ and $G_{\text{gnn}}$. While the $G_{\text{mlp}}$ does not aggregate features from different samples, rather predict based on only the sample of interest. On the other hand, GNN-based classifier $G_{\text{gnn}}$ has a sophisticated structure with an edge network $f_{\text{edge}}$ and a node classifier $f_{\text{node}}$, which aggregates the features of the samples in the batch. The edge network $f_{\text{edge}}$ encodes the relationship between nodes and allows the messages passing along the edges, thus efficiently aggregate the information carried by nodes. The node classifier $f_{\text{node}}$ provides more robust predictions by taking advantage of the context-aware learning paradigm. Instead of merely counting on the current sample $\mathbf{x}$, $G_{\text{gnn}}$ is capable of giving the prediction on a global scale based on the entire mini-batch. GNN and MLP capture different aspects of the information and naively using either of them may lead to noisy features or unreliable predictions regarding the classification performance. This motivated a cooperation of the two classifiers to improve each other. The predictions of the MLP head and the GNN head are defined as

$$\hat{y} \leftarrow \texttt{softmax}\big(G_{\text{mlp}}(F(\mathbf{x}))\big) \tag{1}$$

and

$$\bar{y} \leftarrow \texttt{softmax}\big(G_{\text{gnn}}(F(\mathbf{x}))\big). \tag{2}$$

The cross-entropy $\ell_{\text{ce}}^{\text{mlp}}$ and $\ell_{\text{ce}}^{\text{node}}$ are minimized over all source samples to train the $G_{\text{mlp}}$ classifier and $f_{\text{node}}$ of $G_{\text{gnn}}$.

*1) Coteaching and Pseudolabeling of the MLP and GNN:* Following the concept of coteaching [8], MLP and GNN are trained together to provide feedback to each other.

The first information flow is from MLP to GNN. The aim of $f_{\text{edge}}$ is to build an affinity matrix $\hat{\mathcal{A}}$ that encodes the relationship between nodes, i.e., samples in a mini-batch. The binary representation is a simple but effective choice, where 1 indicates that the $i$th and $j$th sample share the same class label and 0 otherwise. Due to the lack of the labels of target samples, $\hat{\mathcal{A}}$ is a rather sparse matrix that carries little information. To solve this, $G_{\text{mlp}}$ provides pseudolabels of the unlabeled target samples and forms a target matrix $\hat{\mathcal{A}}^{\text{tar}}$ in the similar way. $\hat{\mathcal{A}}$ is learned by minimizing the (binary cross entropy) edge loss $\ell_{\text{bce}}^{\text{edge}}$ between the elements of the current affinity matrix produced by $f_{\text{edge}}$ and the elements of the target matrix $\hat{\mathcal{A}}^{\text{tar}}$ given by $G_{\text{mlp}}$. By far, $G_{\text{gnn}}$ is able to encode the pairwise similarity between every two nodes by studying $\hat{\mathcal{A}}$ under the guidance of $\hat{\mathcal{A}}^{\text{tar}}$.

The other information flow is from GNN to MLP. Generally, confident predictions produce higher softmax value for one class in comparison to the other classes. Based on this, $f_{\text{node}}$ computes the score of each sample in the target domain as the max of softmax value

$$w_j \leftarrow \max_{c \in n_c} p\big(\bar{y}_{t,j} = c|\mathbf{x}_{t,j}\big). \tag{3}$$

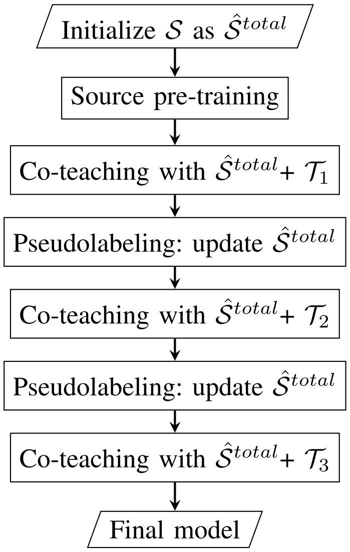
Initialize $\mathcal{S}$ as $\hat{\mathcal{S}}^{total}$
Source pre-training
Co-teaching with $\hat{\mathcal{S}}^{total}$+ $\mathcal{T}_1$
Pseudolabeling: update $\hat{\mathcal{S}}^{total}$
Co-teaching with $\hat{\mathcal{S}}^{total}$+ $\mathcal{T}_2$
Pseudolabeling: update $\hat{\mathcal{S}}^{total}$
Co-teaching with $\hat{\mathcal{S}}^{total}$+ $\mathcal{T}_3$
Final model

Fig. 1. Proposed multitarget approach with source domain $\mathcal{S}$ and assuming three target domains $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3$ (in that order of closeness to the source domain).

Higher value of $w_j$ indicates that the pseudolabel is more reliable to be accepted as an element in the source domain. Thus $w_j$ is compared to a threshold $\tau$, and if higher, the target sample, together with its pseudolabel, is regarded as an additional source sample, and the elements in the labeled samples $\hat{\mathcal{S}}^{\text{total}}$ are updated. By doing this, the $G_{\text{gnn}}$ creates a series of context-aware pseudosamples after processing a target domain, thus increasing the ability of the $G_{\text{mlp}}$ to learn the subsequent target domains.

Subsequently, the target adaption is performed on the next target domain, however using the updated $\hat{\mathcal{S}}^{\text{total}}$. Considering the next target is more different from the source and thus less likely to contain samples that can be included in pseudosource, $\tau$ is increased by $\tau'$.

*2) Domain Discriminator D:* To further close the gap between the source domain and the target domains, a domain discriminator $D$ to predict the domain of the samples is trained using an adversarial loss $\ell_{\text{adv}}$, following [11]. The prediction of $D$ is obtained by $\hat{d}$ computed as $\texttt{sigmoid}(D(F(\mathbf{x})))$.

Fig. 1 shows the proposed multitarget approach assuming three target domains. Furthermore, the training procedure is detailed in Algorithm 1.

### D. Using Trained Model for Inference

Once the proposed MLP-GNN dual head model is trained on all target domains, the GNN head can be used during inference to determine the class of a target test samples. Unlike [7], we do not use any final fine-tuning step.

## IV. Experimental Validation

### A. Datasets

We conducted experiments on the following datasets.

1) **Multicity** dataset containing three domains (Moscow, Mumbai, and Sydney) consisting of six classes each. There are approximately 800 images per class per city and are sampled from LCZ42 dataset [9] Sentinel-2 images with 10 m/pixel resolution.

**Algorithm 1** Training Procedure of the Proposed MTDA Method

```
require: source dataset $\mathcal{S}$, number of classes $n_c$
require: target datasets $\mathcal{T} = \{\mathcal{T}_j\}_{j=1}^N$
require: networks $F$, $D$, $G_{mlp}$, $f_{edge}$, $f_{node}$ with parameters
         $\theta$, $\psi$, $\phi$, $\varphi$, $\varphi'$, respectively. The $f_{edge}$ and $f_{node}$ form
         the $G_{gnn}$.
require: hyperparameters $B$, $\tau$, $\tau'$ $K_{source}$, $K$. We use $B = 32$,
         $K_{source} = 1000$, and $K = 5000$.
Step 1: Pretraining on the source dataset
for $k$ in $(1 : K_{source})$ do
    Sample mini-batch $(\mathbf{x}_{s,i}, y_{s,i})_{i=1}^B \sim \mathcal{S}$
    Compute loss $\ell_{ce}^{mlp}$ and update $\theta$, $\phi$
end
Step 2: Multitarget adaptation
$\hat{\mathcal{S}}^{total} \leftarrow \mathcal{S}$
$\hat{\mathcal{T}}^0 \leftarrow \{\mathcal{T}_j\}_{j=1}^N$
for $n$ in $(0 : N-1)$ do
    $\mathcal{H} \leftarrow \{\}$                                   ▷ Empty list
    Stage 1: Domain selection stage
    for $\mathcal{T}_j$ in $\hat{\mathcal{T}}^n$ do
        compute $H(\mathcal{T}_j)$
        $\mathcal{H} \leftarrow \mathcal{H} \mid\mid H(\mathcal{T}_j)$                        ▷ Append
    end
    $\mathbb{D}^n \leftarrow \mathrm{argmin}_j\, \mathcal{H}$                    ▷ Chosen domain
    Stage 2: Adaptation on single target
    for $k$ in $(1 : K)$ do
        $\hat{\mathcal{B}}_s^n \leftarrow (\mathbf{x}_{s,i}, y_{s,i})$ pairs $\sim \hat{\mathcal{S}}^{total}$
        $\hat{\mathcal{B}}_t^n \leftarrow (\mathbf{x}_{t,i})_{i=1}^B \sim \mathcal{T}_{\mathbb{D}^n}$
        $\hat{y} \leftarrow \mathtt{softmax}(G_{mlp}(F(\mathbf{x})))$
        $\bar{y} \leftarrow \mathtt{softmax}(G_{gnn}(F(\mathbf{x})))$
        $\hat{d} \leftarrow \mathtt{sigmoid}(D(F(\mathbf{x})))$
        update $\psi$ by using $\ell_{adv}$
        update $\theta$, $\phi$ by using $\ell_{ce}^{mlp}$ and $\ell_{adv}$
        update $\theta$, $\varphi$, $\varphi'$ by using $\ell_{bce}^{edge}$ and $\ell_{ce}^{node}$
    end
    Stage 3: Pseudo-labeling stage
    $\mathcal{D}_t^{\mathbb{D}^n} \leftarrow \{\}$                                 ▷ Empty list
    for $\mathbf{x}_{t,j} \in \mathcal{T}_{\mathbb{D}^n}$ do
        $w_j \leftarrow \max_{c \in n_c} p(\bar{y}_{t,j} = c | \mathbf{x}_{t,j})$
        if $w_j > \tau$ then
            $\mathcal{D}_t^{\mathbb{D}^n} \leftarrow \mathcal{D}_t^{\mathbb{D}^n} \mid\mid \{(\mathbf{x}_{t,j}, \mathrm{argmax}_{c \in n_c}\, p(\bar{y}_{t,j} = c | \mathbf{x}_{t,j}))\}$
            ▷ Append
        end
    end
    $\hat{\mathcal{S}}^{total} \leftarrow \hat{\mathcal{S}}^{total} \cup \mathcal{D}_t^{\mathbb{D}^n}$                  ▷ Pseudo-source
    $\hat{\mathcal{T}}^{n+1} = \hat{\mathcal{T}}^n \setminus \mathcal{T}_{\mathbb{D}^n}$
    $\tau \leftarrow \tau + \tau'$       ▷ Increase threshold for next
    target
end
```

2) **Multiseason** dataset containing images from four domains (summer, fall, winter, and spring) corresponding to seven different classes. There are 1000 images per class per season and are sampled from Sen12-MS dataset [10] Sentinel-2 images with 10 m/pixel resolution RGB bands.

### B. Evaluation Protocol and Settings

We use the classification accuracy to evaluate the performance. The performance for a given source is given by setting the remaining domains as target domains. We use the Resnet-18 network as the feature extractor. We show quantitative comparisons to conditional adversarial domain adaptation (CDAN) [11], batch-instance normalization (BIN) [13], coteaching [7], and MT-MTDA [6]. While CDAN is based on adversarial training, BIN is based on statistical alignment, and thus together, they cover both major approaches in DA, as discussed in Section II. CDAN is modified to exploit information from all target domains by combining them together while adapting. Adaptation of BIN is batch-specific and thus it takes into account one target domain at a time. The other two approaches are designed for MTDA and so no such modifications are required.

TABLE II

MULTICITY: PERFORMANCE COMPARISON WITH SYDNEY AS THE SOURCE DOMAIN

| Method | Moscow | Mumbai | Average |
|---|---|---|---|
| Source training | 49.57 | 31.72 | 40.64 |
| CDAN [11] | 57.17 | 36.98 | 47.07 |
| BIN [13] | 50.69 | 31.20 | 40.95 |
| Co-teaching [7] | 54.30 | 43.99 | 49.14 |
| MT-MTDA [6] | 56.19 | 40.90 | 48.55 |
| Proposed | **61.64** | **45.18** | **53.42** |

TABLE III

KL DIVERGENCE VALUE BEFORE AND AFTER ADAPTATION: WITH SYDNEY AS THE SOURCE DOMAIN

| Phase | Moscow | Mumbai |
|---|---|---|
| Before adaptation | 0.0032 | 0.0037 |
| After adaptation | 0.0018 | 0.0026 |

TABLE IV

MULTICITY: PERFORMANCE COMPARISON WITH MUMBAI AS THE SOURCE DOMAIN

| Method | Moscow | Sydney | Average |
|---|---|---|---|
| Source training | 50.91 | 33.23 | 42.07 |
| CDAN [11] | 65.08 | **62.26** | 63.67 |
| BIN [13] | 53.27 | 28.74 | 41.01 |
| Co-teaching [7] | 89.20 | 45.46 | 67.33 |
| MT-MTDA [6] | 45.08 | 49.40 | 47.24 |
| Proposed | **89.69** | 47.06 | **68.37** |

### C. Results

*1) Multicity:* Table II shows the quantitative result by using Sydney as the labeled source domain and Moscow and Mumbai as the unlabeled target domains. Model simply trained on source domain (Sydney) obtains an average accuracy of 40.64% on the two target cities. While adversarial training-based CDAN significantly improves the result (47.07%), statistical alignment-based BIN does not significantly improve the result. This shows that mere statistical alignment is not sufficient for mitigating distribution gaps between multiple cities. Proposed method outperforms all compared paradigms and it obtains an accuracy of 61.64% for Moscow, 45.18% for Mumbai, and thus an average accuracy of 53.42% over two target cities. We also observe reduction in Kullback–Leibler (KL) divergence after adaptation (Table III).

Table IV shows the quantitative result by using Mumbai as the source domain. Source-trained model obtains an average accuracy of 42.07%. Proposed method obtains an average

TABLE V

MULTISEASON: AVERAGE PERFORMANCE FOR EACH SEASON IS SHOWN, TAKING THE CONSIDERED SEASON AS SOURCE AND ALL OTHER SEASONS AS TARGETS

| Method | Spring | Summer | Fall | Winter |
|---|---|---|---|---|
| Source training | 47.16 | 46.03 | 52.51 | 45.06 |
| CDAN | 49.94 | 51.45 | **53.15** | 52.53 |
| BIN | 45.56 | 49.21 | 43.80 | 39.40 |
| Co-teaching | **52.41** | 54.12 | 51.38 | 57.85 |
| MT-MTDA | 42.44 | 49.93 | 44.41 | 47.34 |
| Proposed | 52.38 | **54.53** | 51.83 | **60.22** |

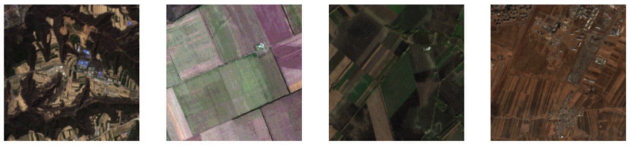

Fig. 2. Visualization of season variation on class cropland. From left to right: spring, summer, fall, and winter.

accuracy of 68.37% and outperforms the source trained model and all compared paradigms. The proposed MTDA improves classification accuracy for the targets by more than 26% over source-trained model.

*2) Multiseason:* Average accuracies, considering each season as source and other three seasons as targets, are shown in Table V. For spring, summer, and winter seasons (as source), proposed method outperforms source-trained model and both CDAN and BIN. However, for spring and fall as source, proposed method is slightly outperformed by coteaching and CDAN, respectively. Gain of the proposed method over the source-trained models are less (approx. 5%, 8%, and 15% for spring, summer, and winter, respectively) in comparison to multicity dataset. A visualization for season variation is shown in Fig. 2.

## V. CONCLUSION

We proposed a GNN-based method for remote sensing MTDA. The proposed method incrementally adapts a source-trained classifier for multiple targets. We validated the proposed method on two different types of domain shifts, namely geographic shift and seasonal shift. Experimental results clearly indicate the potentials of the proposed method. MTDA is a comparatively new area in remote sensing and there is still scope of improving this paradigm. In future, we plan to devise a method to identify irrelevant source samples, discarding which may potentially benefit the adaptation process. Furthermore, we plan to extend the method for multisource multitarget adaptation and mixed target domain settings.

## REFERENCES

[1] N. Bengana and J. Heikkila, "Improving land cover segmentation across satellites using domain adaptation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 1399–1410, 2021.

[2] W. Hong, Z. Wang, M. Yang, and J. Yuan, "Conditional generative adversarial network for structured domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1335–1344.

[3] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 7167–7176.

[4] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1410–1417.

[5] Y. Li, N. Wang, J. Shi, X. Hou, and J. Liu, "Adaptive batch normalization for practical domain adaptation," *Pattern Recognit.*, vol. 80, pp. 109–117, Aug. 2018.

[6] L. T. Nguyen-Meidine, A. Belal, M. Kiran, J. Dolz, L.-A. Blais-Morin, and E. Granger, "Unsupervised multi-target domain adaptation through knowledge distillation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, Jan. 2021, pp. 1339–1347.

[7] S. Roy, E. Krivosheev, Z. Zhong, N. Sebe, and E. Ricci, "Curriculum graph co-teaching for multi-target domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5351–5360.

[8] B. Han *et al.*, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Proc. NIPS*, 2018, pp. 1–11.

[9] X. X. Zhu *et al.*, "So2Sat LCZ42: A benchmark data set for the classification of global local climate zones [software and data sets]," *IEEE Geosci. Remote Sens. Mag.*, vol. 8, no. 3, pp. 76–89, Sep. 2020.

[10] M. Schmitt and Y.-L. Wu, "Remote sensing image classification with the SEN12MS dataset," 2021, *arXiv:2104.00704*.

[11] 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.

[12] O. Tasar, S. L. Happy, Y. Tarabalka, and P. Alliez, "SEMI2I: Semantically consistent Image-to-Image translation for domain adaptation of remote sensing data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Sep. 2020, pp. 1837–1840.

[13] H. Nam and H.-E. Kim, "Batch-instance normalization for adaptively style-invariant neural networks," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 2563–2572.

[14] 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.

[15] S. Ji, D. Wang, and M. Luo, "Generative adversarial network-based full-space domain adaptation for land cover classification from multiple-source remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 3816–3828, May 2021.

[16] 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.

[17] T. Isobe *et al.*, "Multi-target domain adaptation with collaborative consistency learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 8187–8196.

[18] J. Zheng *et al.*, "A two-stage adaptation network (TSAN) for remote sensing scene classification in single-source-mixed-multiple-target domain adaptation ($S^2M^2T$ DA) scenarios," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–13, 2022.

[19] A. Rosenfeld and J. K. Tsotsos, "Incremental learning through deep adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 651–663, Mar. 2020.

[20] Y. Luo, Z. Wang, Z. Huang, and M. Baktashmotlagh, "Progressive graph learning for open-set domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 6468–6478.

[21] C. Paris and L. Bruzzone, "A sensor-driven hierarchical method for domain adaptation in classification of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1308–1324, Mar. 2018.

[22] C. Persello, "Interactive domain adaptation for the classification of remote sensing images using active learning," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 4, pp. 736–740, Jul. 2012.

[23] T. Lasloum, H. Alhichri, Y. Bazi, and N. Alajlan, "SSDAN: Multi-source semi-supervised domain adaptation network for remote sensing scene classification," *Remote Sens.*, vol. 13, no. 19, p. 3861, Sep. 2021.