SCIDA: Self-Correction Integrated Domain Adaptation from Single- to Multi-label Aerial Images

Tianze Yu†, *Student Member, IEEE,* Jianzhe Lin†, *Student Member, IEEE,* Lichao Mou, Yuansheng Hua, Xiaoxiang Zhu, *Senior Member, IEEE,* Z. Jane Wang, *Fellow, IEEE*

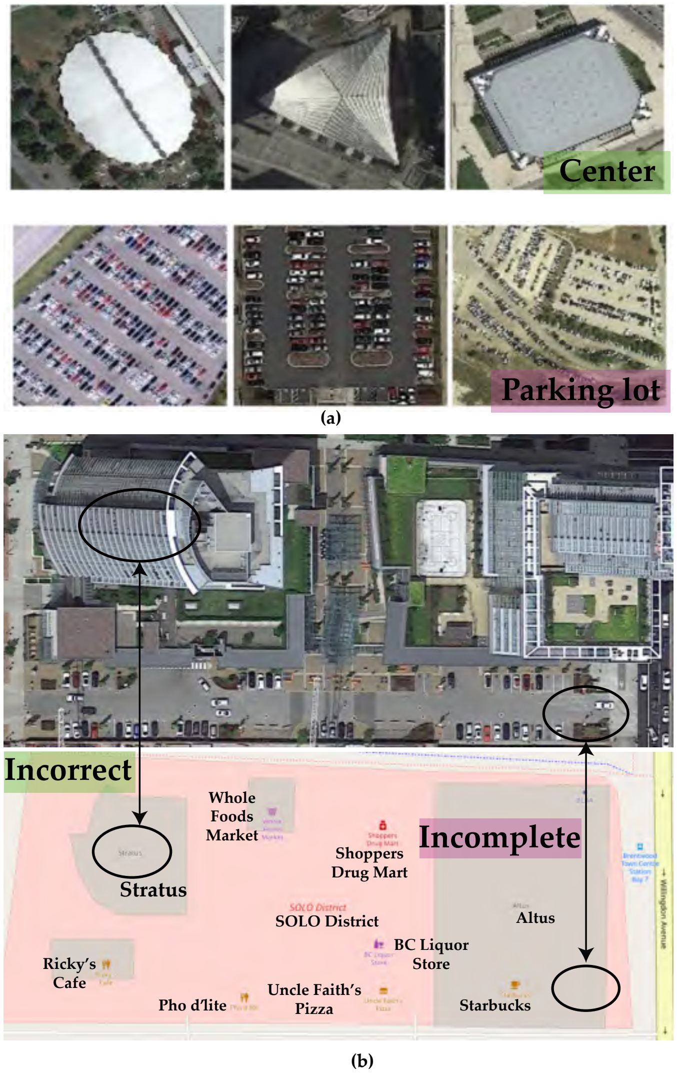
***Abstract*—Most publicly available datasets for image classification are with single labels, while images are inherently multi-labeled in our daily life. Such an annotation gap makes many pre-trained single-label classification models fail in practical scenarios. This annotation issue is more concerned for aerial images: Aerial data collected from sensors naturally cover a relatively large land area with multiple labels, while annotated aerial datasets, which are publicly available (e.g., UCM, AID), are single-labeled. As manually annotating multi-label aerial images would be time/labor-consuming, we propose a novel self-correction integrated domain adaptation (SCIDA) method for automatic multi-label learning. SCIDA is weakly supervised, i.e., automatically learning the multi-label image classification model from using massive, publicly available single-label images. To achieve this goal, we propose a novel Label-Wise self-Correction (LWC) module to better explore underlying label correlations. This module also makes the unsupervised domain adaptation (UDA) from single- to multi-label data possible. For model training, the proposed model only uses single-label information yet requires no prior knowledge of multi-labeled data; and it predicts labels for multi-label aerial images. In our experiments, trained with single-labeled MAI-AID-s and MAI-UCM-s datasets, the proposed model is tested directly on our collected Multi-scene Aerial Image (MAI) dataset. The code and data are available on GitHub(https://github.com/Ryan315/Single2multi-DA).**

***Index Terms*—Unsupervised Domain Adaptation, Aerial Image, GCN, MAI Dataset, Noise, OSM.**

I. Introduction

With easy access to increasing aerial data from satellites/ Unmanned Aerial Vehicles (UAVs), annotating the newly collected aerial data is of great importance. However, obtaining clean multi-label annotations manually for aerial data has long been a challenging task. A recent trend for aerial data annotation is resorting to crowdsourcing data, such as OpenStreetMap (OSM). OSM, an editable map, is built/annotated by volunteers from scratch. However, the quality of such a manually annotated map may not be satisfying. Incompleteness and incorrectness are two primary concerns, as illustrated in the example in Fig. 1.

Instead of resorting to crowdsourcing data, recent advances in machine learning (ML) make automatic annotation of aerial



(a)

(b)

Fig. 1. (a) Single-label aerial image examples from the MAI-AID-s dataset, which serves as the source domain data. (b) The top is a multi-label aerial image from the MAI dataset, which serves as the target domain data. The bottom shows the corresponding noisy annotations from OSM for the top image. As indicated, “Center” is incorrectly annotated as “Stratus”, and “Parking lot” is missed.

† indicate equal contribution. Jianzhe Lin, Tianze Yu, and Z. Jane Wang are with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada. e-mail: jianzhelin, tianzey, zjanewang@ece.ubc.ca.

Lichao Mou, Yuansheng Hua, Xiaoxiang Zhu are with the Technical University of Munich and the German Aerospace Center, Germany.e-mail: lichao.mou,Yuansheng.Hua, Xiaoxiang.Zhu@dlr.de.

images possible. To train a reliable multi-label aerial image classification framework, we need to (1) design an efficient ML model architecture; and (2) learn with massive annotated data. However, collecting such multi-label aerial images with an exhaustive, consistent list of annotations requires significant time and effort. Furthermore, there is almost no publicly available multi-label aerial image dataset, and the online annotations from OSM are not always reliable. Training the model with noisy labels from OSM could lead to poor classification performance. Therefore, direct training of a multi-label aerial image classification model remains challenging.

An alternative way is to train the model with single-labeled aerial image data, since annotating a single-label aerial image is much easier than a multi-label one. Moreover, there are publicly available single-labeled datasets, e.g., AID and UCM datasets, which have covered almost all classes of interest for aerial images. Therefore, they could provide intense supervision for training a multi-label classification model. However, using the model trained by single-label data to predict the labels for a multi-label image is a challenging task. This task is illustrated in Fig. 1, where single-labeled data (the source domain) is shown in Fig. 1(a), and the corresponding multi-label data including the same labels (the target domain) is shown in Fig. 1(b). To realize multi-label aerial image classification using single-labeled data, we formulate the problem as a domain adaptation task and propose a novel framework where prior information from single-label data can be adapted to multi-label aerial images.

To our knowledge, this is the first work to examine the challenging task of learning a multi-label aerial image classifier on large-scale datasets (the target domain) by using publicly available single-label data (the source domain). Our major contributions are as follows:

- We propose a challenging single- to multi-label domain adaptation task. The target domain multi-label data are unannotated, large-scale, unconstrained aerial images in real-world scenarios, while the source domain is the annotated single-label aerial data publicly available (e.g., AID [1], UCM [2]).
- We propose a novel Self-Correction Integrated Domain Adaptation (SCIDA) framework, from single- to multi-label aerial images. Different from existing feature-based unsupervised domain adaptation methods, our framework explores the underlying label correlations by introducing the Label-Wise self-Correction (LWC) module. With GCN as the backbone, the LWC explores label correlations and iteratively corrects the pseudo labels learned by our domain adaptation module (the DWC branch, as in Fig. 2).
- We empirically compare labeling strategies for multi-label datasets to explore the learning potential of using single labels. Given the same annotation budget, our experiments show that the networks trained with single-label images can provide competitive performances as those learned with a fully annotated subset of multi-label images.
- A new single-multi-label aerial image (MAI) dataset with clean labels is collected in our study to make the experiment possible. It will be the first web available dataset for multi-label aerial images.

II. Related Work

A. Partial Multi-Label Learning (PML)

Multi-label learning has been an active research topic of practical importance, as images collected in the wild are always with more than one annotation [3]. Conventional multi-label learning research [4][5][6] mainly relies on the assumption that a small subset of images with full labels are available for training. However, this may be difficult to satisfy in practice, as manually yielded annotations always suffer from incomplete and incorrect annotation problems, as illustrated in Fig. 1. Therefore, partial multi-label learning is emerging, which aims to learn a multi-label classification model from ambiguous data [7], [8], [9]. Traditional methods treat missing labels as negatives and wrong labels as positives during the model training process [10], [11], [12], [13], [14], while it could lead to degradation of the classification performance.

To mitigate the problem of missing or wrong labels, a novel approach for partial label learning is to treat missing labels as a hidden variable via probabilistic models and predict missing labels by posterior inference [15], [16]. The work in [17] models missing labels as negatives, and then corrects the induced error by learning a transformation on the output of the multi-label classifier. However, this approach requires high memory and is hard to optimize. Scaling these models to large datasets would be difficult [18], [19]. Another recent trend for partial label learning can be found in [20], [21], [22], which introduces curriculum learning and bootstrapping to increase the number of annotations. During model training, this approach uses the partially annotated data and the unannotated data whose predicted labels are with the highest confidences [23], [24]. Curriculum learning is further combined with the graph model in [25] to better capture the label association when exploiting the unlabeled data during training. This approach still relies heavily on the unlabeled images, which would taint the training data when they are attached to incorrect labels. This problem is called semantic drift.

Different from the existing models, we propose a novel approach for multi-label learning with the extreme case of partial multi-label data, namely the **single label** data. Compared with commonly used partial multi-label data, single-label data are collected specifically for a single class. Therefore, such data are much easier to acquire online, and do not have the incomplete or incorrect annotation problem. In this paper, for the first time, we introduce domain adaptation to train the multi-label classification model using the annotated single-label data.

B. Domain Adaptation (DA)

Domain adaptation is a method to share knowledge between data from different datasets. DA aims to minimize the data gap between datasets [26], [27], [28]. Here, we formulate the knowledge transfer between the single-label data and multi-label data as a domain adaptation problem in our task. Recent

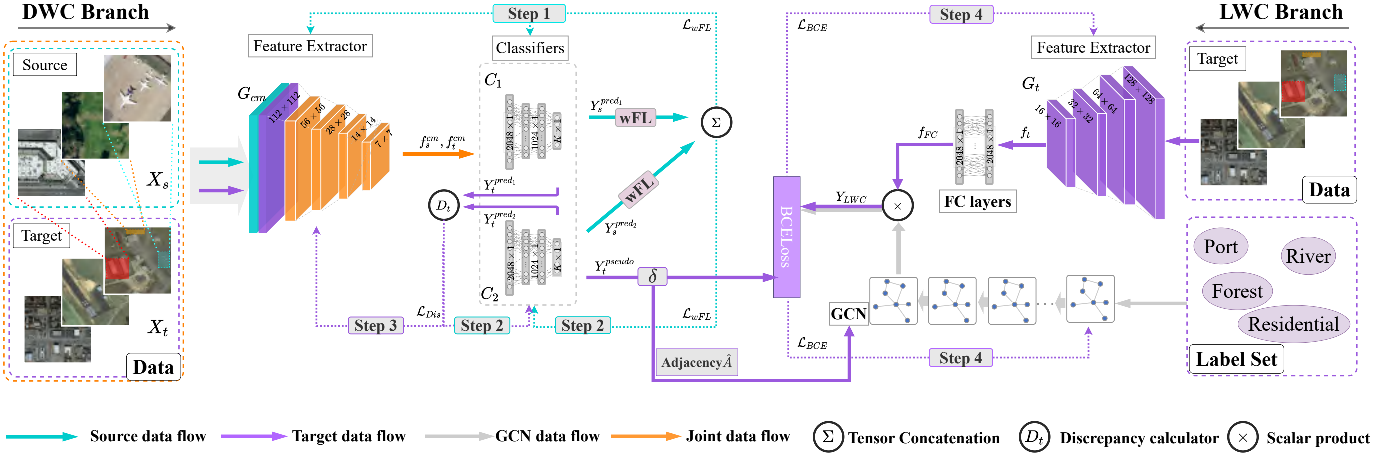


Fig. 2. The flowchart of training the proposed SCIDA with the ResNet backbone. The flow mainly consists of the DWC branch and the LWC branch. G_{cm} and G_t are feature generators; and C_1 and C_2 represent classifiers. "wFL" means "weighted focal loss", and δ controls the learning depth at each training step.

years have witnessed the exploitation of adversarial domain adaptation, which stems from the technique proposed in [29]. The principal idea is to introduce adversarial learning by one feature generator and one domain discriminator [30], [31], [32]. The generated features from the source domain and the target domain are aligned to confuse the domain discriminator until it cannot figure out which domain the features are from [33], [34], [35], [36]. One major limitation of existing adversarial domain adaptation methods is that they are not task-specific. For instance, the generated features in the DA model might not work well for the classifier [37], [38], [39]. This problem could be even more severe for our task, since the classification tasks for the source and target domains are two different types (one is single-label classification, and the other is multi-label classification). A recent advance for task-specific DA is the Maximum Classifier Discrepancy (MCD) method, which is proposed to make the adversarial mechanism task-specific by constructing adversarial learning between *task-specific classifiers* and *feature generator* [40], [41], [42]. However, this method couldn't generalize well when the classifiers from two domains are not the same. To solve this problem, here we propose our new domain adaptation framework.

III. Method

In this section, we present the proposed SCIDA model for single- to multi-label domain adaptation.

A. Overview

In the proposed SCIDA framework, we need to correlate the domain-wise data, and also explore the label-wise correlation among target domain data. This label-wise correlation is used for self-correction in the framework.

For the domain-wise correlation (DWC), we propose using the domain adaptation to correlate the two domains, due to the large domain gap between single-label and multi-label data. For the label-Wise self-Correction (LWC), due to the lack of correlation for the one-hot encoded source domain data, the LWC is learned with self-supervision in the target domain. The Graph convolutional network (GCN) is introduced to model the LWC directly. The LWC is used for self-correction for multi-label classification.

The general flowchart of the proposed model is illustrated in Fig. 2, which is mainly made up of the DWC branch and the LWC branch. The inputs of the proposed framework are introduced as follows. The annotated source domain data is represented with $X_s = \{x_s^i, \mathbf{y}_s^i\}_{i=1}^{N_s}$ (x and y represent the data and the label respectively), while the target domain data is represented with $X_t = \{x_t^i\}_{i=1}^{N_t}$ where N represents the number of images in the dataset.

B. Domain-wise Correlation

For the domain-wise correlation, the goal of this branch is to align the features from source and target domains by utilizing two task-specific classifiers as a discriminator. The output of this branch is the pseudo label for target domain data. This branch is made up of three parts: a common feature extractor G_{cm}, two classifiers C_1 and C_2, and a classifier discriminator D_t. The extracted source domain feature $f_s^{cm} = G_{cm}(X_s; \theta_{G_{cm}})$ and target domain feature $f_t^{cm} = G_{cm}(X_t; \theta_{G_{cm}})$ are the inputs to the two classifiers. We need to detect target samples that largely deviate from the distribution of the source data, and align features from the two domains. As two classifiers are assumed to be effective on source domain samples with full annotations, the classification results from these two classifiers should be the same. While the target samples deviate from the source data distribution, and are likely to be classified differently by the two distinct classifiers. Our goal is to minimize the performance gap between the two classifiers for target domain samples. In our framework, this discrepancy for the two classifiers can be calculated in the target domain by D_t. If the discrepancy is minimized, we assume the data feature from the target domain is aligned with the source domain. And then we can use the source domain annotations to supervise the classification of the target domain data. The general training of this branch can be found in Sec. IV.

However, different from regular task-specific domain adaptation in [40], the major challenge for domain adaptation from single-label to multi-label is the sharing of classifiers. If the single-label classifier is trained to generate multiple labels by setting a threshold for probability, the single-label source data will suffer from the problem of imbalanced training data. The imbalance issue contains two aspects: the imbalance of positive and negative samples in each class, and the imbalance of the samples in different classes in the entire dataset. In this task, there are much fewer positive samples than the negative samples, with a ratio of about $1/K$ (K is the number of classes). To overcome this problem, we propose to use the **weighted focal loss (wFL)** instead of the regular cross-entropy loss for the optimization of the model, which is formulated as below:

$$\begin{aligned}\mathcal{L} = &-\sum_{i=1}^{K} p_\beta(\alpha y^i (1 - p^i(y^i\,|x))^\gamma \log p^i(y^i\,|x) \\ &+ (1-\alpha)(1-y^i)p_i^\gamma(y^i\,|x)\log(1-p^i(y^i\,|x))),\end{aligned} \tag{1}$$

where p_β is the proportion of class-wise samples with respect to all the data in the dataset and fulfills $p_\beta \in (0,1)$ & $\sum_{\beta=1}^{C} p_\beta = 1$. α and γ here are empirically set as 0.25 and 2 respectively [43].

For the multi-label target dataset, the imbalance is eased to some extent. However, as annotations are not available for the target domain data, the classification in the target domain doesn't generate a loss for the supervision of the model. Instead, the discrepancy between the two classifiers in the target domain is regarded as the loss and used to optimize the model.

C. Label-wise Correlation

For the Label-Wise self-Correction (LWC) branch, the goal of this branch is to self-correct the pseudo label generated by the DWC branch. This branch is made up of two components, including a separate target convolution neural network with G_t being the backbone for image classification, and a graph convolution network being the backbone for label correlation learning.

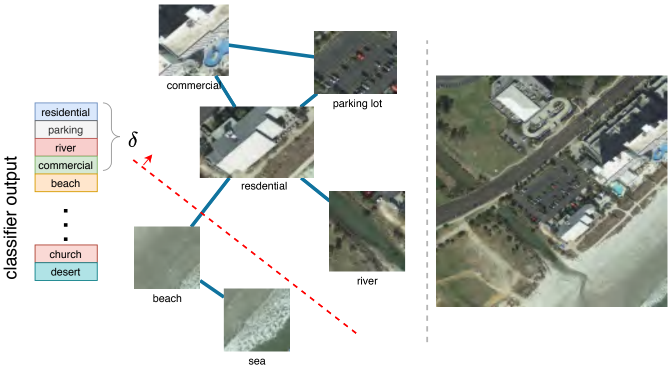


Fig. 3. An example of δ. The right shows the image, and the left shows the predicted labels.

Algorithm 1: Correlation matrix construction.

```
Stage I: Calculate the co-occurrence label
Input: Pseudo ground truth $Y_t^{pseudo}$
Output: Correlation matrix A;
while $epoch \leq max\ epoch$ do
    for $batch \leftarrow 1$ to $N$ do
        for $idx\_i,\ idx\_j \leftarrow 1$ to $num\_categories$ do
            if $Y_t^{pseudo}[idx\_i]\ \&\ Y_t^{pseudo}[idx\_j] \neq 0$ then
                cor_count[idx_i][idx_j] += 1
            end
        end
    end
end
Stage II: Correlation matrix normalization
for $idx \leftarrow 1$ to $num\_categories$ do
    cor_count[idx] = $\frac{cor\_count[idx]}{\sum cor\_count[idx][:]}$
end
return cor_count
```

1) Correlation matrix construction: Label-wise correlation works by propagating label information between nodes based on the correlation matrix. It's a crucial point of how to construct the correlation matrix. As in the proposed unsupervised scenario, there's no pre-defined correlation matrix in the target task. We will construct the correlation matrix in a target-data-driven approach based on the pseudo ground truth labels. The procedure of constructing the correlation matrix is detailed in Alg. 1. For the pseudo ground truth, we introduce a hyper-parameter δ for controlling the number of pseudo ground truth labels for each image, as shown in Fig. 3. As there's no prior label knowledge for the target domain data and the total number of labels for each image varies, we assume that the percentage of pseudo ground truth labels per image is a fixed constant δ (e.g., suppose the dataset has 20 labels in total, with $\delta = 0.2$, the number of pseudo ground truth labels per image is 4). δ is used to determine how many nodes in GCN need to be updated in each iteration. An intuition is that δ is set as the average number of labels per image in the pseudo ground truth. Besides, as LWC branch is supervised by the pseudo labels generated by DWC branch, introducing δ will help to smooth the decision boundary of the classification. In this way, the LWC could converge towards the correct direction, and meanwhile is not restricted to the output of DWC branch. What's more, when calculating the co-occurrence of labels, some rare co-occurrence in the labels will introduce much noise and cause a long-tail distribution problem. Introducing the parameter δ could solve the long-tail distribution problem to a great extent. In the ablation study, we also analyzed the effects of the parameter δ.

2) Label-wise GCN: The inputs of this branch include four parts: the original target domain data, the target domain label set, the pseudo ground truth, as well as the normalized adjacency matrix $\hat{A}$ learned from the DWC branch (which represents the occurrence frequency of each label). We also need to point out that the label in the GCN module is different in a conventional convolution neural network, and is intended to solve the problem under a non-Euclidean topological graph. The computation graph is generated based on the label embedding of each node and its neighbors. We follow a common practice[44] to deploy GCN:

$$H^{(l+1)} = \sigma(\hat{A}H^{(l)}W^{(l)}) \tag{2}$$

Algorithm 2: Training for SCIDA.

```
Input: $X_s, X_t, Y_s, \hat{A}$, labels;
Output: Parameters for $G_{cm}$, $C_1$, $C_2$, GCN, and $G_t$;
while $epoch \leq max\ epoch$ do
    for $batch \leftarrow 1$ to $N$ do
        Pseudo labels generation: Input the normalized source and target
          domain data to optimize DWC branch by minimizing Eq. (3), (4),
          (5), and generate $Y_t^{pseudo}$;
        Self Correction: Update both DWC and LWC by minimizing Eq.
          (6), which indicates the difference between $Y_t^{pseudo}$ and
          $Y_{LWC}$.
    end
end
```

where $\hat{A}$ is the normalized adjacency matrix mentioned above, $H^{(l)}$ denotes the label embedding at the l-th layer in GCN, $W^{(l)}$ is a learnable transformation matrix, and σ acts as a non-linear operation. We employ LeakyReLU to implement this operation.

The general operation routine for this branch is as follows: The feature vector of target domain data is first extracted by $G_t(x_t;\theta_{G_t})$. This feature vector is fed to two fully connected (FC) layers and a 2048×1 feature vector f_{FC} is generated. For the input of the GCN model, GloVe[45] is used to generate the embedding of labels. Then the output of GCN (a $2048 \times K$ matrix) and f_{FC} are fed to a scalar product layer to produce a classification result Y_{LWC} for this branch. The difference between the classification result $Y_t^{pred_2}$ from C_2 (we assume C_1 and C_2 can get a unified result finally) and Y_{LWC} is used to generate a binary cross-entropy loss (BCE loss), which will optimize all components in the LWC branch.

IV. Two-stage Model Training

In the training procedure, we learn the parameters of DWC and LWC branches jointly and iteratively. In the primary stage, we initialize the pseudo label by the DWC branch. In the second stage, both branches are trained to optimize the pseudo label. The iterative training way is introduced as follows, and concluded in Alg. 2.

A. Pseudo labels generation in DWC

Pseudo labels generation is generated by an adversarial domain adaptation way in DWC branch [40]. We first initialize the weights of the feature generator G_{cm} and the classifiers C_1, C_2, and freeze other components. The model in this stage is optimized by the weighted focal loss calculated on the annotated source domain data. This process can be formulated as:

$$\min_{\theta_{G_{cm}},\theta_{C_1},\theta_{C_2}} \mathcal{L}_{wFL}(X_s, \mathbf{Y}_s) \tag{3}$$

here $\mathcal{L}$ is defined in Eq. (1).

We initialize the parameters of G_{cm}, C_1, C_2 and train the framework for classification in the source domain, as well as to achieve an adversarial training ready state. Then we use an adversarial manner to train the two classifiers C_1 and C_2 using the domain discriminator D_t, and optimize the weights of G_{cm}.

To be more specific, we first maximize the discrepancy loss generated by D_t for target domain data. This loss is used for optimizing C_1 and C_2. The purpose is to identify target samples whose extracted features deviate the most from the distribution of the source domain features. At the same time, we need to keep C_1 and C_2 effective for the classification of the source domain data. We formulate this by

$$\min_{\theta_{G_{cm}},\theta_{C_1},\theta_{C_2}} \mathcal{L}_{wFL}(X_s, Y_s) - \mathcal{L}_{dis}(Y_t^{pred_1}, Y_t^{pred_2}) \tag{4}$$

here $Y_t^{pred_1}, Y_t^{pred_2}$ represent multi-label predictions from C_1 and C_2 respectively, and $\mathcal{L}_{dis}$ is computed with a scalar subtraction between these two [40].

Then, adversarial to Eq. (4), we minimize the classifier discrepancy loss by optimizing the weights of G_{cm}, in order to encourage uniformed classification results from the two classifiers. C_1 and C_2 are frozen now, and the objective function is

$$\min_{\theta_{G_{cm}}} \mathcal{L}_{dis}(Y_t^{pred_1}, Y_t^{pred_2}) \tag{5}$$

Finally, we assume Y_t^{pred} from C_2 to be the Y_t^{pseudo}. We need to point out that the Y_t^{pred} from C_1 is equal to Y_t^{pred} from C_2 (Y_t^{pseudo}), when the training converges.

B. Self Correction

Self Correction is achieved unsupervised in the target domain for self-correcting the pseudo label Y_t^{pseudo}. Only by learning the label correlation with GCN (which can be understood as a self-supervision) but without extra supervisions/annotations, the Y_t^{pseudo} is optimized and corrected.

In this stage, we first get Y_{LWC} from the LWC branch for the target domain data. This obtained Y_{LWC} from the scalar product operation (as in Fig. 2) is used for self-correction for the pseudo label Y_t^{pseudo}. Noted that here Y_{LWC} is the label predictions of target domain samples from LWC branch. Y_{LWC} can be represented as $Y_{LWC} = \{\hat{y}^1, \hat{y}^2, ..., \hat{y}^N\}$, supposing there are N target domain samples. As we note that the choice of different losses in this stage doesn't make noticeable difference, we empirically choose the BCE loss as below:

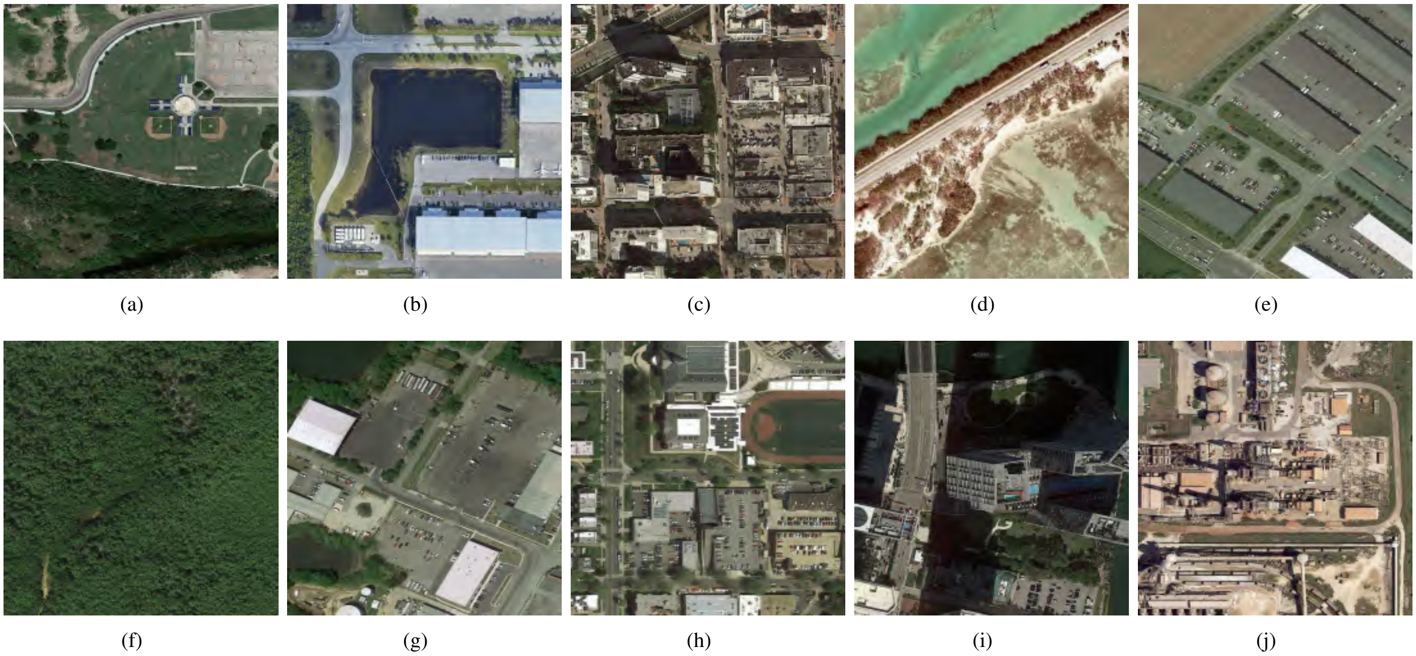
$$\min_{\theta_{G_t},\theta_{GCN},\theta_{G_{cm}},\theta_{C_1},\theta_{C_2}} \mathcal{L}_{BCE}(Y_{LWC}, Y_t^{pseudo}) \tag{6}$$

and $\mathcal{L}_{BCE}$ for sample x_i is further defined as

$$\mathcal{L}_{BCE} = -\frac{1}{K}\sum_{i=1}^{K}(y^i \log(\hat{y}^i) + (1-y^i)\log(1-(\hat{y}^i))) \tag{7}$$

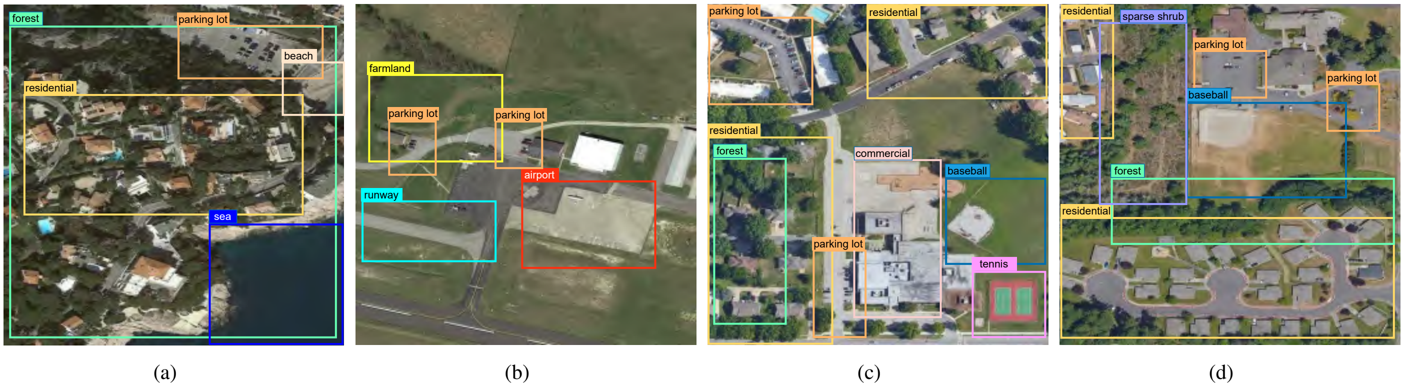
y^i is the pseudo labels from DWC branch. We need to point out that both y^i and $\hat{y}^i$ are variables being optimized, and the final convergence is only realized when the y^i equals to $\hat{y}^i$. This BCE loss will be used to optimize the parameters of the whole network, including both DWC and LWC branches.

These two stages will iterate until convergence in the training process. Finally, Y_t^{pseudo} will stay unchanged, and we assume it as Y_{pred} in the testing phase.


(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Fig. 4. Examples images with multiple labels from MAI-AID-m dataset. The labels are: (a) baseball, parking lot, park, river; (b) airport, forest, lake; (c) commercial, bridge, parking lot, residential; (d) beach, bridge; (e) farmland, commercial, parking lot; (f) forest, river; (g) commercial, lake, parking lot, residential, storage tank; (h) stadium, soccer, residential, parking lot; and (i) river, bridge, commercial.



(a) (b) (c) (d)

Fig. 5. Examples with object level annotations from the MAI-UCM-m dataset. The annotations are: (a) beach, sea, residential, forest, parking lot; (b) airport, farmland, parking lot, runway; (c) commercial, residential, forest, parking lot, baseball, tennis; and (d) sparse shrub, forest, baseball, residential, parking lot.

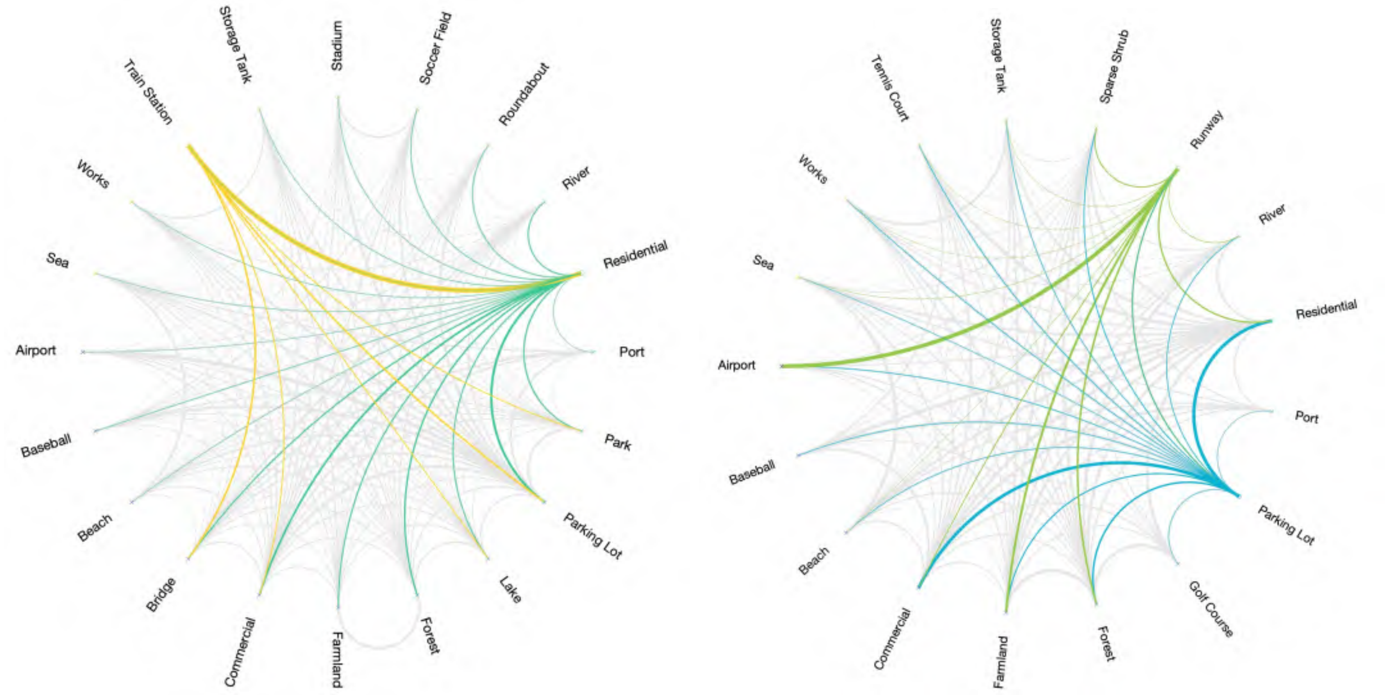
V. Experiments

A. Datasets, Setup, and Evaluation Metrics

1) MAI Dataset: Data is playing an especially critical role in enabling computers to interpret images as compositions of objects. Based on the scenario described above, we created a new dataset named MAI, which contains images and ground-truth annotations for the single- to multi-label domain adaptation task. The proposed MAI dataset contains two subsets, MAI-AID and MAI-UCM. Besides, each subset is in pairs, which contains one single-label dataset and one multi-label dataset. The statistical parameters of the dataset is detailed in Table. I and Fig. 7.

MAI-AID As a large-scale aerial image dataset, AID is introduced in 2017[1], which collects sample images from Google Earth imagery. The original AID dataset is made up of 30 categories, including 10,000 images. 7,050 images from 20 categories are collected from AID dataset and used as the single-label dataset named **MAI-AID-s**. 3,239 images with exactly the same 20 categories are collected from OpenStreetMap(OSM)[**?**] which is a collaborative project to create a free editable geographic database of the world and named **MAI-AID-m**. The difference lies in the fact that the annotation of MAI-AID-s is single-label based and the annotation of MAI-AID-m is multi-label based. All the images in MAI-AID-m are labelled by the specialists in the field of remote sensing image interpretation. Some samples of MAI-AID-m dataset are exhibited in Fig. 4.

MAI-UCM UC Merced Land Use Dataset(UCM)[2] is an aerial image dataset whose images are manually extracted from the USGS National Map Urban Area Imagery collection for various urban areas around the country. The original UCM dataset is made up of 21 categories, including 2,100 images.



(a) Correlation visualization of MAI-AID-m.

(b) Correlation visualization of MAI-UCM-m.

Fig. 6. Correlation visualization of the proposed MAI dataset. Here grey lines show all connections between label pairs; the linewidths represent the correlation values between label pairs; the categories with more and less occurrence are highlighted in different colors for demonstration. I.e., they are 'Residential' and 'Train Station' in MAI-AID-m, and they are 'Parking Lot' and 'Runway' in MAI-UCM-m respectively.

1,700 images from 17 categories are collected from UCM dataset and used as the single-label dataset named **MAI-UCM-s**. 1,799 images with exactly the same 17 categories are collected from OSM and named **MAI-UCM-m**. Similar to MAI-AID-m, the images of the MAI-UCM-m are also annotated with multi labels. Fig. 5 exhibits four examples of MAI-UCM-m dataset with object level annotations.

Meanwhile, for multi-label datasets, there exist inner connections and correlations between different categories. In Fig. 3, the adjacent matrix of the two multi-label datasets are visualized in an intuitional way. The line connecting two labels represents the connection and the width represents the correlation degree between two categories. Among them, for instance, "Residential" and "Parking Lot" have wide correlations with other categories, while "Runway" only have a few and the closest correlation is with "Airport".

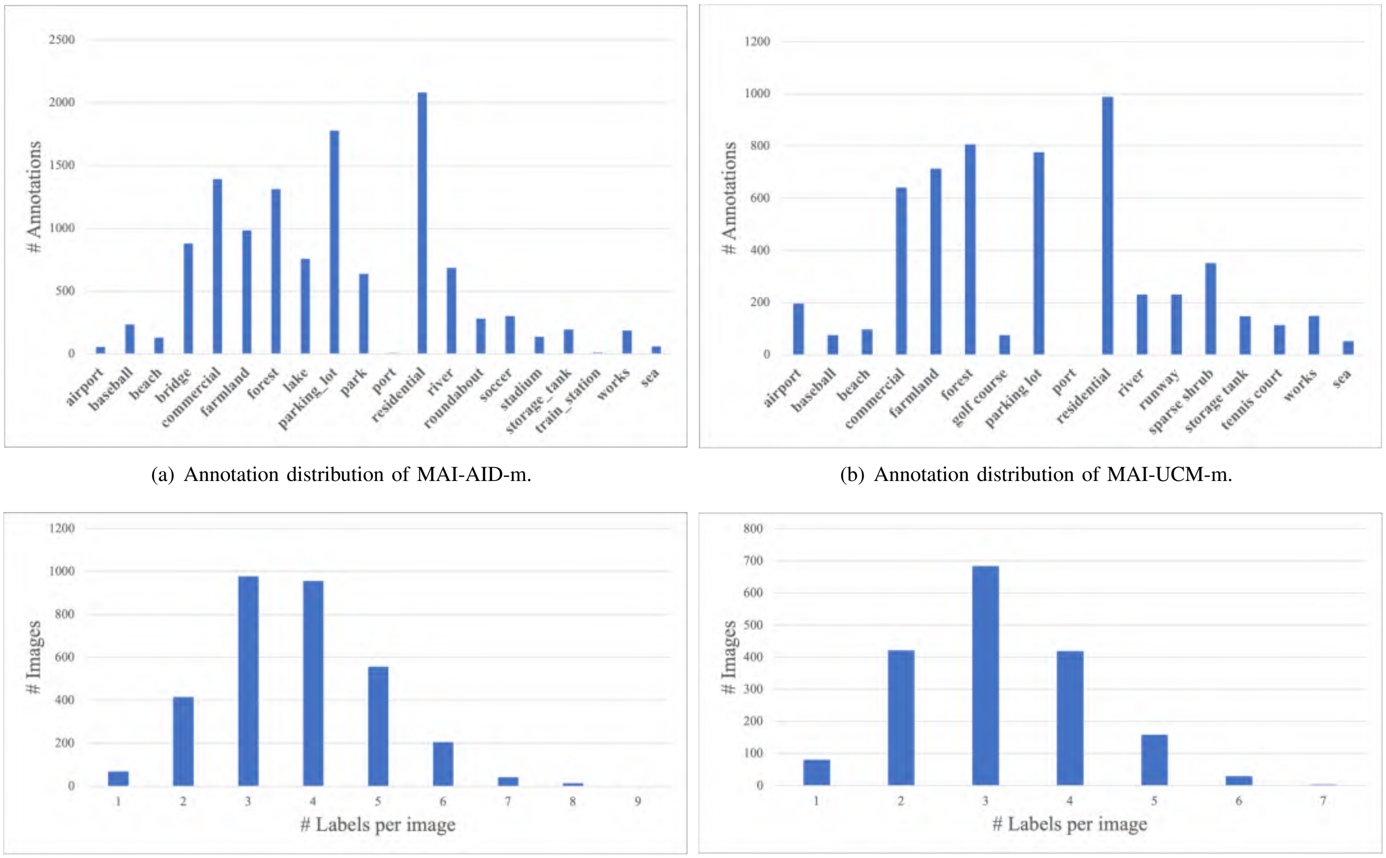
2) Scene: MAI-AID-s to MAI-AID-m adaptation: For this task, we consider 20 classes for single- to multi-label domain adaptation, including Airport, Baseball, Beach, Bridge, Commercial, Farmland, Forest, Lake, Parking Lot, Park, Port, Residential, River, Roundabout, Soccer Field, Stadium, Storage Tank, Train Station, Works, and Sea. 7,050 images from the MAI-AID-s dataset are used as the source domain, while 3,239 images from the MAI-AID-m dataset are used as the target domain. The number of source domain images in each class is imbalanced, ranging from 200 to 700. The input images are re-scaled to 512 × 512. Examples can be found in Fig. 8.

MAI-UCM-s to MAI-UCM-m adaptation: For this task, the datasets both have 17 classes, including Airport, Baseball, Beach, Commercial, Farmland, Forest, Golf Course, Parking Lot, Port, Residential, River, Runway, Sparse Shrub, Storage Tank, Tennis Court, Works, Sea. In total, 1,700 images from MAI-UCM-s are used as the source domain. 1,799 images from MAI-UCM-m are used as the target dataset. The input images are re-scaled to 224 × 224. Examples can be found in Fig. 8.

3) Setup: A pre-trained ResNet-101[46] is used as the backbone of the feature generator. For different datasets, the input images are randomly cropped and resized for data augmentation. SGD is used for network optimization. The

TABLE I
PROPERTIES OF THE PROPOSED MAI DATASET

Dataset	Size	# Images	# Categories	# Avg. Labels	# Max. Labels	# Min. Labels	Resolution
MAI-AID-s	1.7G	7,050	20	Single label annotation			600×600
MAI-AID-m	198.0M	3,239	20	3.73	9	1	512×512
MAI-UCM-s	176.3M	1,700	17	Single label annotation			256×256
MAI-UCM-m	105.5M	1,799	17	3.14	7	1	512×512



(a) Annotation distribution of MAI-AID-m.

(b) Annotation distribution of MAI-UCM-m.

(c) Image distribution of MAI-AID-m.

(d) Image distribution of MAI-UCM-m.

Fig. 7. Distributions of the proposed MAI dataset. (a) and (b) show the category-wised annotation distributions. (c) and (d) show the image distributions, with the number of labels in one image as the variable.

momentum is set to be 0.9, with a decay of 10^{-4}. The batch size is 4. The initial learning rate is 0.001 for the DWC branch, and 0.01 for the LWC branch. Both learning rates decay by a factor of 10 for every 30 epochs and 200 epochs. The framework is implemented in PyTorch and trained with two 2080-TI GPUs.

4) Evaluation Metrics: In experiments, for performance evaluation, we report the average overall precision (OP), recall (OR), F1 (OF1), F2 (OF2); and the top-3 OP, OR, OF1, OF2 [44]. Specifically, the F score (F1 when $\beta = 1$ and F2 when $\beta = 2$) is calculated using:

$$F_\beta = \left(1+\beta^2\right) \cdot \frac{\text{precision } \cdot \text{ recall}}{\beta^2 \cdot \text{ precision } + \text{ recall}} \tag{8}$$

B. Qualitative and Quantitative Comparisons

In this section, the qualitative and quantitative results of the proposed task and method are detailed. To fully explore the capacity of our proposed network, the order and the content of the experiments are designed on purpose. The proposed task is single to multi-label domain adaptation, but there's no method reported in the literature for the proposed task yet. So according to the structure of the proposed framework which is consist of two main modules, DWC branch and LWC branch, we have:

1) comparison with the state-of-the-art methods for multi-label classification(e.g., the best performing methods KSSNet [47] and GCN-ASL [48] and the most representative method ML-GCN [44]) to prove the effectiveness of DWC branch.
2) comparison with the state-of-the-art methods for domain adaptation approaches(e.g, the best performance unsupervised domain adaptation method HAFN/SAFN [49] and the most representative task-specific unsupervised domain adaptation(UDA) method MCD [40]) to prove the effectiveness of LWC branch.
3) comparison with the state-of-the-art methods for partial label learning for multi-label classification(the PRODEN[50] and the DNPL method [51]).

An intuitive comparison can be found in Fig. 9. In this figure, 6 images with different number of labels ranging from 2 to 7 are chosen for demonstration. The multi label method could have multiple prediction. But because of the existence of the domain gap between the source and the target dataset, the results are not satisfying. On the other hand, UDA method shows a higher precision on the prediction than ML, but because of the its own limitation, it cannot learn the correlation between the categories of the target dataset. The proposed method shows a much higher accuracy on predicting the multiple labels in the proposed task.

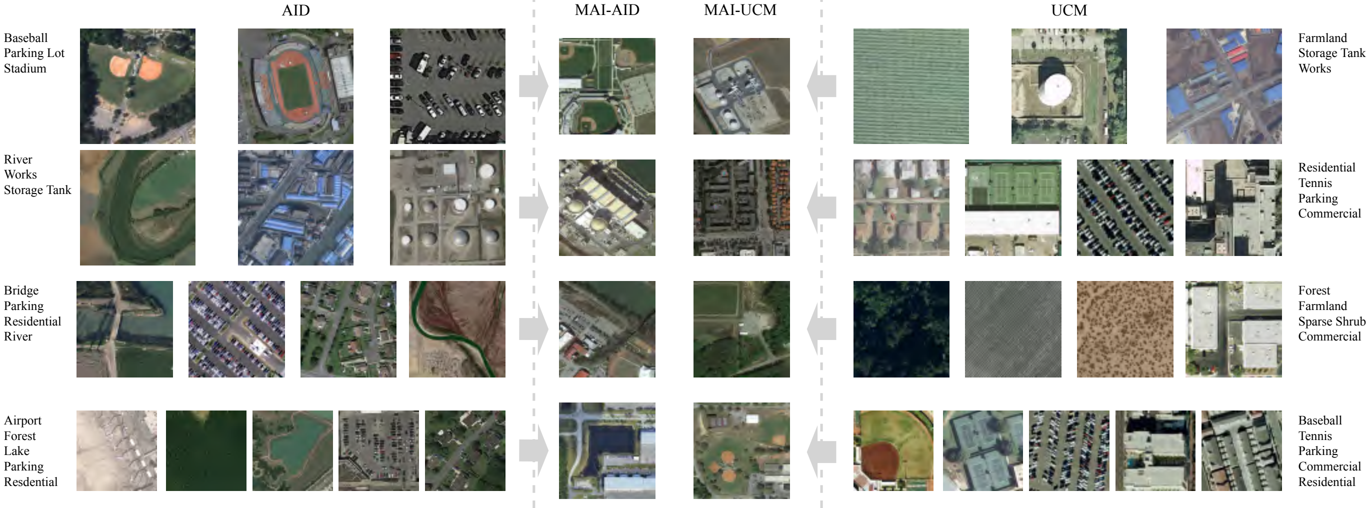


Fig. 8. Image mappings of MAI-AID-s → MAI-AID-m and MAI-UCM-s → MAI-UCM-m in the proposed domain adaptation task.

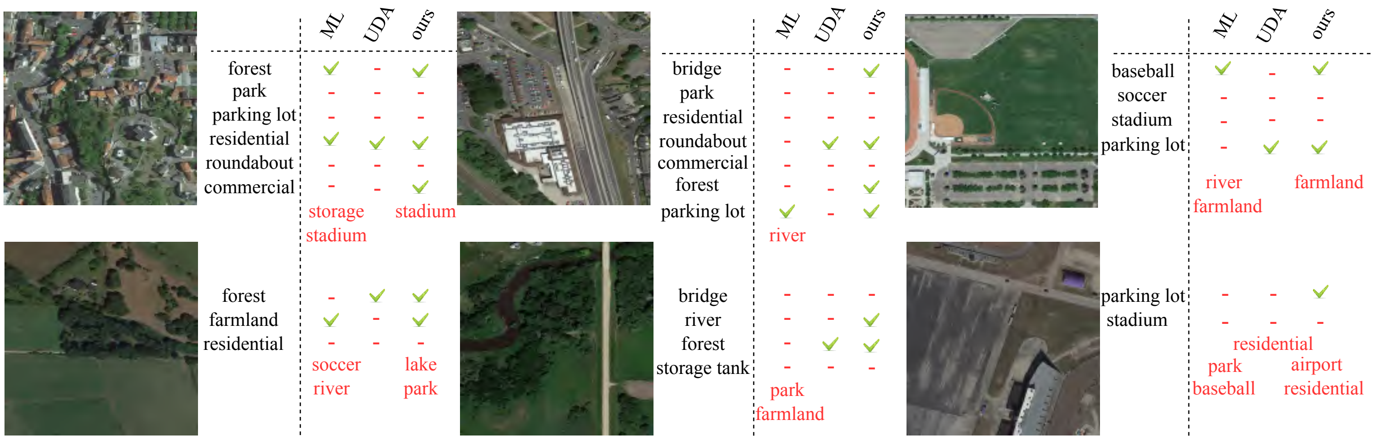


Fig. 9. MAI-AID dataset: Visualization examples of the classification outputs. We use visualized results of ML-GCN [44] and MCD [40] for the results of multi-label classification (ML) and unsupervised domain adaptation (UDA) respectively.

Comparisons with multi-label classification methods. Quantitative results are reported in Table. II and III. The results of SCIDA and SCIDA with optimized-δ are reported separately. All comparison methods are trained on the annotated single label data (MAI-AID-s/MAI-UCM-s) and tested directly on the multi-label data (MAI-AID-m/MAI-UCM-m).

For the MAI-AID-s to MAI-AID-m task, it is evident that the proposed SCIDA method provides superior classification performances under all metrics. For SCIDA with optimized-δ, the OP drops a bit because more related correlation estimation is made, while OR/OF1/OF2 improve significantly. For the MAI-UCM-s to MAI-UCM-m task, our proposed method outperforms other comparison methods under all metrics except OP. With a similar OP accuracy, the OR/OF1/OF2 are improved by about 30%. For both tasks, the optimized SCIDA consistently achieves better performances than the regular SCIDA. Based on the performance comparisons with multi-label classification methods (without domain adaptation) in Table. II and III, we can verify that the correlation between single- and multi-label data learned by domain adaptation is significant.

Comparisons with domain adaptation methods. Since the resolution of the MAI-UCM images is quite low, i.e., only 1/4 of that of MAI-AID images, almost all existing domain adaptation methods fail on the MAI-UCM dataset, except our proposed method with self-correction. Therefore, we only report the performance comparisons for the MAI-AID-s to MAI-AID-m task. As shown in Table. IV, the proposed methods significantly outperform comparison methods under all performance metrics. Especially the super performances of the optimized SCIDA (with opt-δ) clearly demonstrate the effectiveness of our proposed self-correction module (LWC branch).

Comparison with partial label learning methods. Partial-label learning(PLL) is a typical weakly supervised learning problem, where each training instance consists of a data and a set of candidate labels containing a unique ground truth label. The task setting is the same as our source domain dataset. The difference is that our target domain consists of an uncertain number of labels. The candidate labels are set

TABLE II
MAI-AID Dataset: Classification Accuracy Comparisons with Different Multi-label Classification Methods

Method	All				Top 3			
	OP	OR	OF1	OF2	OP	OR	OF1	OF2
KSSNet [47]	0.2907	0.1616	0.2077	0.1774	0.3336	0.1303	0.1874	0.1484
GCN-ASL [48]	0.2026	0.1615	0.1797	0.1683	0.2742	0.0389	0.0681	0.0470
ML-GCN [44]	0.3831	0.0452	0.0809	0.0549	0.3837	0.0447	0.0801	0.0543
SCIDA	**0.5432**	0.2230	0.3162	0.2528	**0.5496**	0.2196	0.3138	0.2496
SCIDA(opt-δ)	0.4474	**0.3242**	**0.3760**	**0.3431**	0.4725	**0.3185**	**0.3805**	**0.3407**

TABLE III
MAI-UCM Dataset: Classification Accuracy Comparisons with Different Multi-label Classification Methods

Method	All				Top 3			
	OP	OR	OF1	OF2	OP	OR	OF1	OF2
KSSNet [47]	0.2817	0.1804	0.2199	0.1944	0.2829	0.1769	0.2177	0.1912
GCN-ASL [48]	0.1844	0.3200	0.2340	0.2790	0.1579	0.0487	0.0745	0.0565
MC-GCN [44]	**0.3585**	0.0404	0.0726	0.0491	**0.4127**	0.0356	0.0656	0.0436
SCIDA	0.3358	0.3105	0.3227	0.3153	0.3412	0.2995	0.3190	0.3070
SCIDA(opt-δ)	0.3371	**0.3219**	**0.3293**	**0.3248**	0.3380	**0.3192**	**0.3284**	**0.3228**

TABLE IV
MAI-AID Dataset: Classification Accuracy Comparisons with Different Domain Adaptation Methods

Method	All				Top 3			
	OP	OR	OF1	OF2	OP	OR	OF1	OF2
SAFN [49]	0.4542	0.1216	0.1918	0.1425	0.4542	0.1216	0.1918	0.1425
HAFN [49]	0.4281	0,1147	0.1809	0.1344	0.4281	0.1147	0.1809	0.1344
MCD [40]	0.3327	0.0891	0.1406	0.1044	0.3327	0.0891	0.1406	0.1044
SCIDA	**0.5432**	0.2230	0.3162	0.2528	**0.5496**	0.2196	0.3138	0.2496
SCIDA(opt-δ)	0.4474	**0.3242**	**0.3760**	**0.3431**	0.4725	**0.3185**	**0.3805**	**0.3407**

TABLE V
MAI-AID Dataset: Classification Accuracy Comparisons with Partial-label Learning Methods

Method	All				Top 3			
	OP	OR	OF1	OF2	OP	OR	OF1	OF2
PRODEN [50]	0.1559	0.0422	0.0664	0.0494	0.1559	0.0422	0.0664	0.049
DNPL [51]	0.1031	0.0402	0.0578	0.0458	0.1031	0.0402	0.0578	0.0458
SCIDA	**0.5432**	0.2230	0.3162	0.2528	**0.5496**	0.2196	0.3138	0.2496
SCIDA(opt-δ)	0.4474	**0.3242**	**0.3760**	**0.3431**	0.4725	**0.3185**	**0.3805**	**0.3407**

to be the number of the classes in the experiments. Besides, all the algorithms have the same training, testing dataset and the same number of annotations. Table. V and VI exhibit the results on MAI-AID and MAI-UCM dataset respectively. We can observe that our model surpasses the two competitors on both datasets. Specifically, compared with the two partial label learning methods, the proposed method improves the F1 and F2 score by more than 0.3.

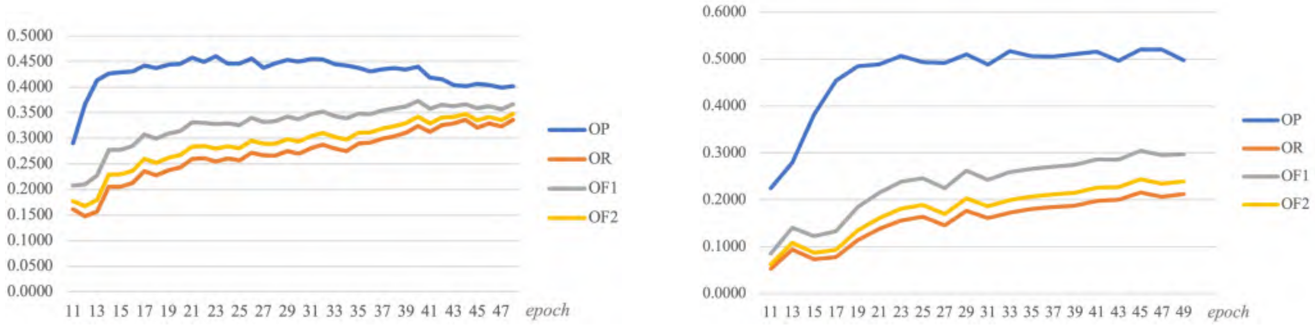
C. *Ablation Studies*

In this section, we perform ablation studies on the other parts of the framework which are not covered in the experiments above. Based on the task of MAI-AID-s to MAI-AID-m adaptation, the ablation studies are separated into the following aspects:

1) the effectiveness of GCN module
2) the effectiveness of LWC branch
3) the comparison of different loss functions
4) the influence of the parameter δ
5) stability of the proposed framework
6) the results under the same annotation budget

GCN module. In this study, we compare the results of the proposed framework SCIDA with and without GCN module to verify the importance of GCN. The results are shown in Table. VII. According to the result, we could verify that, by introducing GCN module, the performance of SCIDA is much improved.

With/without LWC branch. In this part, we generally evaluate the effectiveness of the proposed LWC branch. By deleting the LWC branch directly and only optimizing the



(a) Training curve when $\delta = 0.15$.

(b) Training curve when $\delta = 0.2$.

Fig. 10. Ablation study: Training curves of SCIDA with different values of δ. The evaluation measures OP, OR, OF1 and OF2 of every epoch are logged until the model converges and is stable. To better show the figures, the initial 10 epochs are skipped when plotting.

TABLE VI
MAI-UCM Dataset: Classification Accuracy Comparisons with Partial-label Learning Methods

Method	All				Top 3			
	OP	OR	OF1	OF2	OP	OR	OF1	OF2
PRODEN [50]	0.1421	0.0560	0.0803	0.0637	0.1421	0.0560	0.0803	0.0637
DNPL [51]	0.1143	0.0478	0.0674	0.0541	0.1143	0.0478	0.0674	0.0541
SCIDA	**0.3358**	0.3105	0.3227	0.3153	**0.3412**	0.2995	0.3190	0.3070
SCIDA(opt-δ)	0.3371	**0.3219**	**0.3293**	**0.3248**	0.3380	**0.3192**	**0.3284**	**0.3228**

TABLE VII
Ablation Study: Classification Accuracy Comparisons of the Proposed SCIDA with and without GCN.

Method	All				Top 3			
	OP	OR	OF1	OF2	OP	OR	OF1	OF2
SCIDA(None-GCN)	0.3423	**0.3688**	0.3551	**0.3641**	0.3652	0.2945	0.3260	0.3060
SCIDA	**0.4474**	0.3242	**0.3760**	0.3431	**0.4725**	**0.3185**	**0.3805**	**0.3407**

DWC branch, we can get around 0.14 for OF1 and 0.10 for OF2. In comparison, SCIDA gets 0.38 for OF1 and 0.34 for OF2. This result verifies that the proposed LWC module is quite necessary for our framework.

BCE loss vs weight focal loss (wFL). In this part, we evaluate different loss functions for step-1 training. Specifically, we investigate two loss functions, including the widely used BCE loss and the proposed wFL. To make a fair comparison, hyper-parameters under these two loss functions are tuned specifically to achieve the best performance. The results of the two loss functions are both selected when the model is converged and stable. Table. VIII shows the results using different loss functions on the MAI-AID-s to MAI-AID-m task. We can see that the wFL clearly yields better accuracy under all performance metrics.

TABLE VIII
Ablation Study: Comparisons When Using Different Loss Functions

Method	All			
	OP	OR	OF1	OF2
BCE Loss	0.1747	0.2475	0.2048	0.2285
wFL	**0.4546**	**0.2878**	**0.3524**	**0.3106**
	Top-3			
BCE Loss	0.1480	0.0764	0.1008	0.0846
wFL	**0.4654**	**0.2806**	**0.3501**	**0.3048**

Different values of δ for LWC branch. To explore the effects of δ on classification performance, we consider different values of δ, ranging from 10% to 25%, as depicted in Fig. 11. We can observe that, when δ is set as 20%, the performance is the best. If δ is too small, the correlation learning in the GNN will be slow and the training of the branch will be insufficient; while when δ is set larger than 20%, redundant and useless connections will seriously affect the LWC branch, resulting in a worse overall performance. Therefore, we empirically set δ as 20%.

Model stability. The proposed framework SCIDA is consist of several components. The stability of the framework is quite critical. In fact, the two-stage training manner introduced in Sec. IV could ensure the stability of the model to a great extent. To verify that, the training curve of SCIDA under different δ values are plotted, as shown in Fig. 10. The proposed method is quite stable under different conditions.

Same budget of annotations. In this experiment, we want to demonstrate that with the same number of annotations, when compared with the method directly trained on the

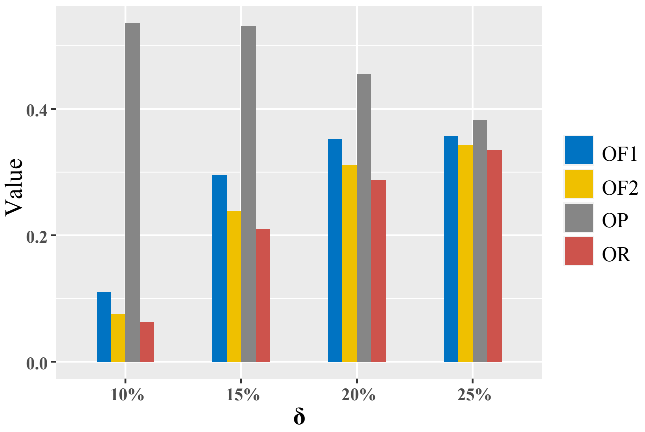


Fig. 11. Ablation study: Performance evaluation when using different δ.

multi-label dataset, the proposed transfer method can yield comparable results. First, the proposed SCIDA uses 5,000 images from the AID dataset for training. In comparison, we train the state-of-the-art multi-label classification method ML-GCN using the target MAI-AID images directly. As the average number of annotations per image for MAI-AID is 3.4, 1,500 images (with $3.4 \times 1500 \approx 5000$ labels) are chosen randomly for training and the remaining are used for testing. With the same number of annotations as described above, compared with the 0.4355 (OF1) and 0.3820 (OF2) of ML-GCN which is trained directly on the target data, SCIDA could achieve comparable results as 0.3673(OF1) and 0.3493 (OF2). It is worth emphasizing that the proposed SCIDA only uses publicly available annotated single-label images. The results suggest that for model training, prior knowledge from publicly available single-label images can be an efficient alternative to manual annotations of multi-label images.

VI. Conclusion

In this paper, we propose a novel framework for single-label to multi-label aerial scene transfer. The proposed SCIDA model integrates self-correction to domain adaptation. This model can be applied to large scale, unlabeled and unconstrained aerial images. The model is trained in a two-stage manner. Our reported multi-label classification results in the target domain demonstrate the effectiveness of the proposed model. A new multi-label aerial image (MAI) dataset is collected and used for experiments. For future work, we will extend the proposed SCIDA model to more challenging image types and applications.

References

[1] G. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu, “Aid: A benchmark data set for performance evaluation of aerial scene classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3965–3981, 2017. 2, 6

[2] Y. Yang and S. Newsam, “Bag-of-visual-words and spatial extensions for land-use classification,” in *SIGSPATIAL GIS*, 2010, pp. 270–279. 2, 6

[3] G. Tsoumakas and I. Katakis, “Multi-label classification: An overview,” *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, 2007. 2

[4] S. Behpour, “Arc: Adversarial robust cuts for semi-supervised and multi-label classification,” in *CVPR Workshops*, 2018, pp. 1905–1907. 2

[5] G. Lyu, S. Feng, and Y. Li, “Partial multi-label learning via probabilistic graph matching mechanism,” in *KDD*, 2020, pp. 105–113. 2

[6] L. Wang, Z. Ding, and Y. Fu, “Adaptive graph guided embedding for multi-label annotation.” in *IJCAI*, 2018, pp. 2798–2804. 2

[7] M. Xie and S. Huang, “Partial multi-label learning with noisy label identification.” in *AAAI*, 2020, pp. 6454–6461. 2

[8] L. Sun, S. Feng, T. Wang, C. Lang, and Y. Jin, “Partial multi-label learning by low-rank and sparse decomposition,” in *AAAI*, 2019, pp. 5016–5023. 2

[9] M. Zhang, F. Yu, and C. Tang, “Disambiguation-free partial label learning,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 10, pp. 2155–2167, 2017. 2

[10] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten, “Exploring the limits of weakly supervised pretraining,” in *ECCV*, 2018, pp. 181–196. 2

[11] S. Bucak, R. Jin, and A. Jain, “Multi-label learning with incomplete class assignments,” in *CVPR*, 2011, pp. 2801–2808. 2

[12] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, “Revisiting unreasonable effectiveness of data in deep learning era,” in *ICCV*, 2017, pp. 843–852. 2

[13] Q. Wang, B. Shen, S. Wang, L. Li, and L. Si, “Binary codes embedding for fast image tagging with incomplete labels,” in *ECCV*, 2014, pp. 425–439. 2

[14] A. Joulin, L. Van Der Maaten, A. Jabri, and N. Vasilache, “Learning visual features from large weakly supervised data,” in *ECCV*, 2016, pp. 67–84. 2

[15] D. Vasisht, A. Damianou, M. Varma, and A. Kapoor, “Active learning for sparse bayesian multilabel classification,” in *KDD*, 2014, pp. 472–481. 2

[16] H. Chu, C. Yeh, and Y. Wang, “Deep generative models for weakly-supervised multi-label classification,” in *ECCV*, 2018, pp. 400–415. 2

[17] I. Misra, C. Lawrence Zitnick, M. Mitchell, and R. Girshick, “Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels,” in *CVPR*, 2016, pp. 2930–2939. 2

[18] J. Deng, O. Russakovsky, J. Krause, M. Bernstein, A. Berg, and F. Li, “Scalable multi-label annotation,” in *CHI*, 2014, pp. 3099–3102. 2

[19] D. Huynh and E. Elhamifar, “Interactive multi-label cnn learning with partial labels,” in *CVPR*, 2020, pp. 9423–9432. 2

[20] H. Dong, Y. Li, and Z. Zhou, “Learning from semi-supervised weak-label data.” in *AAAI*, 2018, pp. 2926–2933. 2

[21] Y. Liu, R. Jin, and L. Yang, “Semi-supervised multi-label learning by constrained non-negative matrix factorization,” in *AAAI*, 2006, pp. 421–426. 2

[22] L. Feng and B. An, “Leveraging latent label distributions for partial label learning.” in *IJCAI*, 2018, pp. 2107–2113. 2

[23] H. Wang, W. Liu, Y. Zhao, C. Zhang, T. Hu, and G. Chen, “Discriminative and correlative partial multi-label learning.” in *IJCAI*, 2019, pp. 3691–3697. 2

[24] M. Zhang and J. Fang, “Partial multi-label learning via credible label elicitation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[25] T. Durand, N. Mehrasa, and G. Mori, “Learning a deep convnet for multi-label classification with partial labels,” in *CVPR*, 2019, pp. 647–657. 2

[26] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, “Revisiting batch normalization for practical domain adaptation,” *arXiv preprint arXiv:1603.04779*, 2016. 2

[27] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulo, “Autodial: Automatic domain alignment layers,” in *CVPR*, 2017, pp. 5067–5075. 2

[28] Y. Li, N. Wang, J. Shi, X. Hou, and J. Liu, “Adaptive batch normalization for practical domain adaptation,” *Pattern Recognition*, vol. 80, pp. 109–117, 2018. 2

[29] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in *ICML*, 2015, pp. 1180–1189. 3

[30] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, “Domain separation networks,” in *NIPS*, 2016, pp. 343–351. 3

[31] B. Sun, J. Feng, and K. Saenko, “Return of frustratingly easy domain adaptation,” 2016. 3

[32] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in *CVPR*, 2017, pp. 7167—-7176. 3

[33] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell., “Cycada: Cycle-consistent adversarial domain adaptation,” in *ICML*, 2018, pp. 1989–1998. 3

[34] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, “Generate to adapt: Aligning domains using generative adversarial networks,” in *CVPR*, 2018, pp. 8503–8512. 3

[35] Y. Tamaazousti, H. Le Borgne, C. Hudelot, M. Seddik, and M. Tamaazousti, “Learning more universal representations for transfer-learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 3

[36] Y. Zhu, F. Zhuang, and D. Wang, “Aligning domain-specific distribution and classifier for cross-domain classification from multiple sources,” in *AAAI*, 2019, pp. 5989–5996. 3

[37] Z. Pei, Z. Cao, M. Long, J. Wang, and J. Wang, “Multi-adversarial domain adaptation,” in *AAAI*, 2018. 3

[38] Z. Cao, M. Long, J. Wang, and M. I. Jordan, “Partial transfer learning with selective adversarial networks,” in *CVPR*, 2018, pp. 2724–2732. 3

[39] M. Long, Z. Cao, J. Wang, and M. I.Jordan, “Conditional adversarial domain adaptation,” in *NIPS*, 2018, pp. 1640–1650. 3

[40] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, “Maximum classifier discrepancy for unsupervised domain adaptation,” *CVPR*, pp. 3723–3732, 2018. 3, 4, 5, 8, 9, 10

[41] C. Lee, T. Batra, M. Baig, and D. Ulbricht, “Sliced wasserstein discrepancy for unsupervised domain adaptation,” in *CVPR*, 2019, pp. 10 285–10 295. 3

[42] S. Kuroki, N. Charoenphakdee, H. Bao, J. Honda, I. Sato, and M. Sugiyama, “Unsupervised domain adaptation based on source-guided discrepancy,” in *AAAI*, 2019, pp. 4122–4129. 3

[43] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *ICCV*, 2017, pp. 2980–2988. 4

[44] Z. Chen, X. Wei, P. Wang, and Y. Guo, “Multi-label image recognition with graph convolutional networks,” in *CVPR*, 2019, pp. 5177–5186. 4, 8, 9, 10

[45] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *EMNLP*, 2014, pp. 1532–1543. 5

[46] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *CVPR*, pp. 770–778, 2016. 7

[47] Y. Wang, D. He, F. Li, X. Long, Z. Zhou, J. Ma, and S. Wen, “Multi-label classification with label graph superimposing,” *arXiv preprint arXiv:1911.09243*, 2019. 8, 10

[48] E. Ben-Baruch, T. Ridnik, N. Zamir, A. Noy, I. Friedman, M. Protter, and L. Zelnik-Manor, “Asymmetric loss for multi-label classification,” *arXiv preprint arXiv:2009.14119*, 2020. 8, 10

[49] R. Xu, G. Li, J. Yang, and L. Lin, “Larger norm more transferable: An adaptive feature norm approach for unsupervised domain adaptation,” in *CVPR*, 2019, pp. 1426–1435. 8, 10

[50] J. Lv, M. Xu, L. Feng, G. Niu, X. Geng, and M. Sugiyama, “Progressive identification of true labels for partial-label learning,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 6500–6510. 8, 10, 11

[51] J. Seo and J. S. Huh, “On the power of deep but naive partial label learning,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3820–3824. 8, 10, 11