Cross-Domain Object Detection with Missing Classes in Target Domain

Benliu Qiu, Heqian Qiu, Haitao Wen, Zichen Song, Linfeng Xu
School of Information and Communication Engineering
University of Electronic Science and Technology of China
Chengdu, China
{qbenliu, hqqiu, haitaowen}@std.uestc.edu.cn, szc.uestc@gmail.com, lfxu@uestc.edu.cn

2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP) | 978-1-6654-7189-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/MMSP55362.2022.9949546

***Abstract*—Many existing methods focus on detecting either objects from different domains or those of rare classes, but it's difficult for them to tackle the two issues together. However, in the real world, due to the difficulty of collecting samples of special classes, deep learning practitioners have to use simulated images to substitute for them. To deal with this scenario, in this paper, we research a new task: cross-domain object detection with missing classes in target domain, where there are only partial classes have images and annotations in the target domain. We devise a simple but effective play-and-plug method to address this new task, named the three-stage learning approach with domain and class information preservation. In addition, extensive experiments demonstrate our method is effective and can boost the performance when added to existing unsupervised domain adaptation object detectors.**

***Index Terms*—object detection, unsupervised domain adaptation**

I. Introduction

In real-world applications, object detectors suffer from the domain gap between development and deployment, e.g., different viewpoints, illuminations, etc., which greatly drops performance of object detectors. To popularize the application of object detection, how to mitigate the domain gap is an unavoidable problem. [1] [2] have adopted the gradient reverse layer via which models can learn domain-invariant features, to handle unsupervised domain adaptation for object detection (abbr. UDA-OD) as illustrated in Fig. 1a. [3] addresses the class-imbalance problem that hinders the usage of pseudo-labelling technique which is commonly used in UDA-OD. [4] introduces category adaptation and bounding box adaptation based on margin disparity discrepancy theory to handle the challenges in UDA-OD.

Apart from the domain gap, a robot deployed in the real world is likely to encounter rare classes that are not collected in the training data. Many few-shot object detection (FSOD) methods [5] [6] are developed to solve this issue by learning fast from only a few samples including objects of these classes as shown in Fig. 1b, whilst zero-shot object detection (ZSOD) methods [7] [8] achieve this goal by transferring semantic class-level information contained in word phrases or attributes, as shown in Fig. 1c. [6] proposes a soft-attention module to calculate class attentive weights from few-shot samples, and use these weights to remodel R-CNN predictor heads for detecting objects. [7] introduces the transformer into ZSOD and outperforms the state-of-the-art at that time. [8] develops a novel semantics-guided contrastive network that aligns the region features and the related semantic information better.

However, the domain gap and the rare class problems usually appear together, since the real world is diverse, changeable, and complicated. For example, developers plan to deploy a robot in the real world, but they can only collect and annotate a few samples from the real world because of limits on resources and manpower. Furthermore, rare classes may never appear in the collected real-world data, and thus the developers have to generate many simulated data to cover these classes. So we expect a robot to utilize the few real-world samples and abundant simulated samples to solve our above-mentioned task. Several works [9] [10] have considered the domain gap and the rare class problems together, but their specific task settings are distinct from ours. [9] proposes a partially zero-shot domain adaptation task, in which the data from target domain are unlabeled. [10] processes the few-shot domain adaptation and few-shot recognition jointly, but a few target domain samples of rare classes are available for their task. Besides these, the two methods consider only recognition.

In this work, we propose firstly the novel cross-domain object detection with missing classes in target domain, which has some different sides to UDA-OD, FSOD and ZSOD. When dealing with the new task, UDA-OD methods need to alleviate the problem incurred by the missing classes in target domain, whereas FSOD and ZSOD methods are required to adapt into a different domain. Therefore, we devise a play-and-plug method to advance performance, which is established on the foundation of UDA-OD models. We utilize a domain information learning stage to master domain information, a missing class learning stage to obtain class information, and a domain information recalling stage to recall learned domain information and meanwhile retain class information with the help of an information preservation technique. Finally, extensive experiments demonstrate our plugin method is effective and can boost the performance of base models.

II. Task Descriptions

In our task settings as illustrated in Fig. 1d, training data consist of images and corresponding annotations from source domain and target domain. But some images with missing

978-6654-7189-3/22/$31.00 © 2022 IEEE

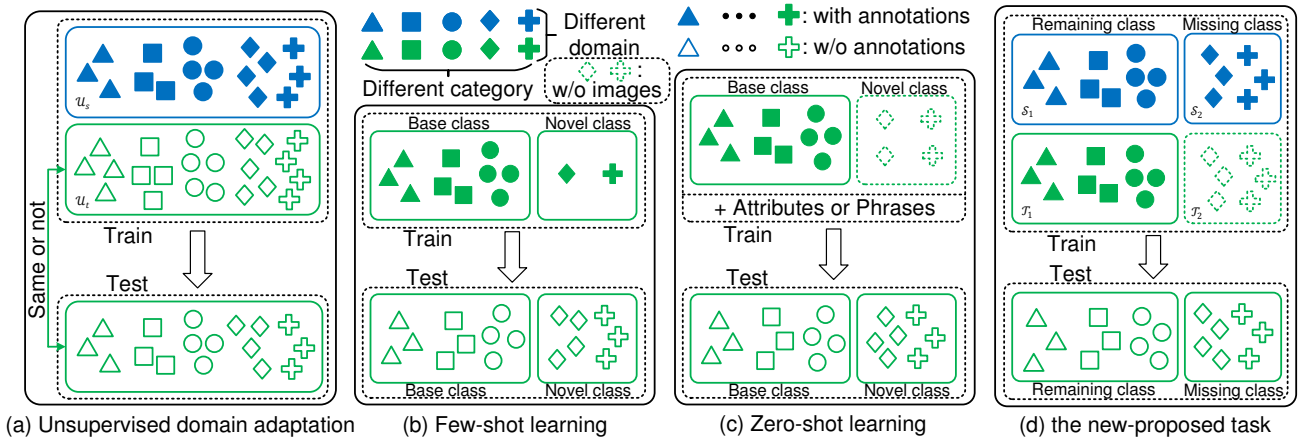


Fig. 1: Comparison between unsupervised domain adaptation, few-shot learning and our new-proposed task. Different colors and shapes denote samples of different domains and categories, respectively. Hollow shapes mean corresponding annotations are unavailable. Dashed shapes mean the corresponding images are unavailable.

classes only appear in source domain and images with annotated objects of these classes are removed in target domain.

On the point of processing images containing missing classes, our setting is different from class-incremental object detection [11] and few-shot object detection [5]. In class-incremental object detection, all images with objects of seen classes in old episodes can be still utilized for learning in current episode, where only annotations of seen classes are inaccessible. Analogously, "novel" does not mean that objects of novel classes are never seen when we train a model using data of base classes for few-shot object detection. We think this setting is unreasonable because the classes we hope the model has never seen have been actually seen. Therefore, to be consistent with the realistic applications, we choose to pick out all images including objects of the missing classes.

A. Notations

We denote the whole training data in source domain as $\mathcal{U}_s$. The source training dataset of remaining classes is $\mathcal{S}_1$ and the source training dataset of missing classes is $\mathcal{S}_2$. Analogously, the target training dataset of remaining classes and missing classes are denoted as $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively. The whole training data in target domain $\mathcal{U}_t$ should have contained $\mathcal{T}_1$ and $\mathcal{T}_2$ for supervised object detection, i.e., $\mathcal{U}_t = \mathcal{T}_1 \cup \mathcal{T}_2$, but for the new task, the training data only include $\mathcal{T}_1$. There are some natural properties for our new-proposed task:

$$\mathcal{S}_1 \cap \mathcal{S}_2 = \emptyset, \mathcal{S}_1 \cup \mathcal{S}_2 = \mathcal{U}_s,$$
$$\mathcal{T}_1 \cap \mathcal{T}_2 = \emptyset, \mathcal{T}_1 \cup \mathcal{T}_2 = \mathcal{U}_t.$$

The label set of all annotated objects is $\mathcal{Y}$. We represent the label set of $\mathcal{S}_1$ and $\mathcal{S}_2$ as $\mathcal{Y}_1^s$ and $\mathcal{Y}_2^s$, respectively. Similarly, for the datasets $\mathcal{T}_1$ and $\mathcal{T}_2$, their corresponding label sets are $\mathcal{Y}_1^t$ and $\mathcal{Y}_2^t$. When diversity and quantity of images are adequate, some interesting relationships possibly exist:

$$\mathcal{Y}_1^s = \mathcal{Y}_1^t, \quad \mathcal{Y}_2^s = \mathcal{Y}_2^t,$$
$$\emptyset \neq \mathcal{Y}_1^s \cap \mathcal{Y}_2^s, \quad \mathcal{Y} = \mathcal{Y}_1^s \cup \mathcal{Y}_2^s$$
$$\emptyset \neq \mathcal{Y}_1^t \cap \mathcal{Y}_2^t, \quad \mathcal{Y} = \mathcal{Y}_1^t \cup \mathcal{Y}_2^t.$$

Notice that equations in the first row do not definitely hold, because some classes maybe coexist with missing classes in one domain but the property may not share across domains.

For object detection, annotations of all objects in an image are composed of three parts: an image $\mathbf{x}_i$, a bounding box list $\mathbf{b}_i$ and a category label list y_i, which are denoted by a triple $(\mathbf{x}_i, \mathbf{b}_i, y_i)$. Based on this notation, we denote the source training dataset of remaining classes as $\mathcal{S}_1 = \{(\mathbf{x}_i^s, \mathbf{b}_i^s, y_i^s)\}_{i=1}^{M}$, and the one of missing classes as $\mathcal{S}_2 = \{(\mathbf{x}_i^s, \mathbf{b}_i^s, y_i^s)\}_{i=M+1}^{M+N}$, where M and N are both the number of images of corresponding datasets. In an akin way, $\mathcal{T}_1 = \{(\mathbf{x}_i^t, \mathbf{b}_i^t, y_i^t)\}_{i=1}^{P}$ and $\mathcal{T}_2 = \{(\mathbf{x}_i^t, \mathbf{b}_i^t, y_i^t)\}_{i=P+1}^{P+Q}$, where P and Q are the corresponding number of images.

B. Task Differences

As illustrated in Fig. 1d, the whole training data of the new-proposed task only contain source dataset $\mathcal{U}_s$ and target dataset $\mathcal{T}_1$. In contrast, the task a UDA-OD model can tackle have only access to the dataset $\mathcal{U}_s$ and images $\{\mathbf{x}_i^t\}_{i=1}^{P+Q}$. Compared with the new-proposed task, UDA-OD has three key distinctions. UDA-OD can be a transductive learning [12] task while our task is of inductive learning. This is because UDA-OD can utilize images of test dataset in the training phase. Thus, models can touch information comprised in test images in advance, which assists models to have a great performance. But test data including images and annotations are not accessible in our new task. Another difference is UDA-OD has no bounding boxes and category labels for training data in target domain. The last but critical difference is data in target domain for our new task miss some classes and any image has no possibility to appear in training phase as long as it comprises objects of missing classes, whereas the covered classes in target domain are identical with those in source domain for UDA-OD. This difference exists as well between the new-proposed task and FSOD. Besides this, FSOD has no cross-domain requirement and the training data of novel classes are scarce. Compared to ZSOD, the new task cannot

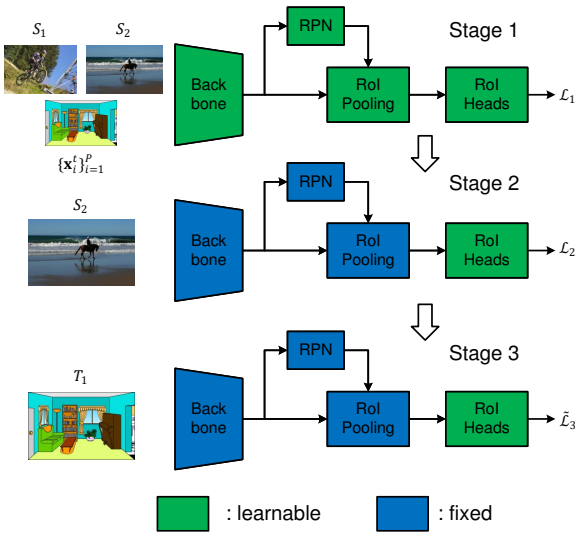


Fig. 2: Overview of the three-stage learning approach.

use attributes or phrases but can access the annotated samples in the source domain.

III. Algorithms for the New Task

A. Overview

Compared with UDA-OD, our new task cannot use data with objects of missing classes, i.e., dataset $\mathcal{T}_2$ in training phase, but could take advantages of annotations in dataset $\mathcal{T}_1$, which provides information on objects of missing classes in target domain.

In order to solve this new task, a model should have the capacity to capture general domain-invariant information of missing classes from dataset $\mathcal{S}_2$. On the other hand, the model need to consolidate the obtained domain information about objects of remaining classes, with the help of annotated instances in dataset $\mathcal{T}_1$. Therefore, we devise our three-stage learning method based on these two considerations.

B. Three-stage learning approach

In this subsection, we introduce a three-stage learning approach for the new task, as shown in Fig. 2. We choose a typical anchor-based object detector Faster R-CNN [13] as our base detection model. Faster R-CNN includes mainly a backbone, a region proposal network (RPN) and a pair of region-of-interest heads (RoI heads). The RoI heads are composed of a category classifier and a bounding box regressor, to classify correctly and localize accurately instances, respectively. The joint loss for object detection is commonly denoted as

$$\mathcal{L}^{\text{det}} = \mathcal{L}^{\text{rpn}}_{\text{cls}} + \mathcal{L}^{\text{rpn}}_{\text{reg}} + \mathcal{L}^{\text{roi}}_{\text{cls}} + \mathcal{L}^{\text{roi}}_{\text{reg}}. \tag{1}$$

In this loss formula, $\mathcal{L}^{\text{rpn}}_{\text{cls}}$ represents the box-classification loss of discriminating foreground from background, and $\mathcal{L}^{\text{rpn}}_{\text{reg}}$ denotes the regression loss of localization for RPN. In a similar way, $\mathcal{L}^{\text{roi}}_{\text{cls}}$ and $\mathcal{L}^{\text{roi}}_{\text{reg}}$ represent the multi-classification loss and localization loss for every instance, respectively.

Domain information learning. UDA-OD methods usually have a competitive capacity to mitigate domain gap after learning on datasets $\mathcal{S}_1$, $\mathcal{S}_2$ and $\{\mathbf{x}_i^t\}_{i=1}^P$, and thus we train in the first stage a UDA-OD model on datasets $\mathcal{S}_1$, $\mathcal{S}_2$ and $\{\mathbf{x}_i^t\}_{i=1}^P$. Based on this point, our method is extensible to any UDA-OD method which does not depend on target domain data of missing classes, i.e, $\mathcal{T}_2$. We simplify the optimization objectives of base methods as

$$\mathcal{L}_1(\boldsymbol{\phi}, \boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x}_i, \mathbf{b}_i, y_i) \in \mathcal{D}} \mathcal{L}^{\text{det}}, \tag{2}$$

where $\mathcal{D} = \mathcal{S}_1 \cup \mathcal{S}_2 \cup \{\mathbf{x}_i\}_{i=1}^P$, and $\boldsymbol{\phi}, \boldsymbol{\theta}$ denote the corresponding learnable parameters of feature extractor and the box predictor, respectively.

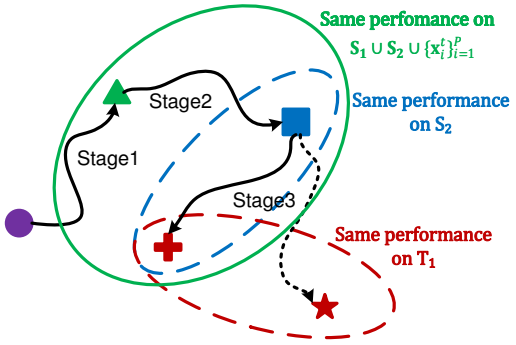


Fig. 3: Diagram of optimization trajectories in three stages. Solid shapes denote model parameters. The purple round represents initial parameters. Dashed arrow is the parameter path without domain and class information preservation. Parameters on an ellipse of the same color have equivalent performance.

Class information learning. In the second stage, we fine-tune the model on dataset $\mathcal{S}_2$. This is because this dataset is the only dataset containing information on objects of missing classes among all three datasets $\mathcal{S}_1$, $\mathcal{S}_2$ and $\mathcal{T}_1$, which are accessible in training phase and we need to obtain adequate class information to remedy the lack of dataset $\mathcal{T}_2$. Wang et.al [14] demonstrates that parameters in RoI heads are plentiful to transfer features from base classes to novel classes in few-shot object detection. So, we only fine-tune parameters of RoI heads. The optimization objective in the second stage is denoted as follows

$$\mathcal{L}_2(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x}_i, \mathbf{b}_i, y_i) \in \mathcal{S}_2} \mathcal{L}^{\text{roi}}_{\text{cls}} + \mathcal{L}^{\text{roi}}_{\text{reg}}. \tag{3}$$

Domain information recalling. Apart from the operations above, in the third stage, we further fine-tune our model on dataset $\mathcal{T}_1$ which is the only dataset implying domain information that we can access in training phase, since our final objective is to make the model have an excellent object detection performance on test dataset of target domain. Although the UDA-OD base model has achieved the domain information in the first stage, the direct fine-tuning on dataset $\mathcal{S}_2$ changes many model parameters that are beneficial for preserving domain information. Hence, the obtained domain information is likely to be interfered, and fine-tuning on dataset $\mathcal{T}_1$ is required to recall domain information. In the third stage, the goal of parameter update is to minimize the following loss:

$$\mathcal{L}_3(\boldsymbol{\theta}) = \mathbb{E}_{(\mathbf{x}_i, \mathbf{b}_i, y_i) \in \mathcal{T}_1} \mathcal{L}^{\text{roi}}_{\text{cls}} + \mathcal{L}^{\text{roi}}_{\text{reg}}. \tag{4}$$

Algorithm 1: 3-stage learning with DCIP.

Input: source domain dataset $\mathcal{S}_1$ and $\mathcal{S}_2$, target domain dataset $\mathcal{T}_1$, training epochs K_1 in second stage and K_2 in third stage.

Output: model parameters $\boldsymbol{\phi}, \boldsymbol{\theta}$

Initialize model parameters $\boldsymbol{\phi}^0, \boldsymbol{\theta}^0$;

// Domain information learning

Train a UDA-OD model on datasets $\mathcal{D} = \mathcal{S}_1 \cup \mathcal{S}_2 \cup \{\mathbf{x}_i^t\}_{i=1}^P$ with the objectives simplified as $\mathcal{L}_1(\boldsymbol{\phi}, \boldsymbol{\theta})$ to achieve parameters $\boldsymbol{\phi}, \boldsymbol{\theta}$;

// Class information learning

for *t = 1 to* $N{\cdot}K_1$ **do**

  Sample a batch data $\mathcal{B} = \{(\mathbf{x}_j, \mathbf{b}_j, y_j)\}_j$ from $\mathcal{S}_2$;

  Calculate the loss of detection on this batch according to $\mathcal{L}_2(\boldsymbol{\theta}^t)$;

  Backward the loss above in the computational graph to get gradients $\nabla_{\theta_i} \sum_j \log p(\mathbf{x}_j|\boldsymbol{\phi}^t, \boldsymbol{\theta}^t), \theta_i \in \boldsymbol{\theta}^t$;

  if *in the last epoch* **then**

    Calculate diagonal elements of FIM iteratively as follows $F_{ii}^t = F_{ii}^{t-1} + \frac{1}{|\mathcal{B}|}\left[\nabla_{\theta_i} \sum_j \log p(\mathbf{x}_j|\boldsymbol{\phi}^t, \boldsymbol{\theta}^t)\right]^2$;

  end

  $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \nabla_{\boldsymbol{\theta}^t} \sum_j \log p(\mathbf{x}_j|\boldsymbol{\phi}^t, \boldsymbol{\theta}^t)$;

end

// Domain information recalling

for *t =* $N{\cdot}K_1{+}1$ *to* $N{\cdot}K_1{+}P{\cdot}K_2$ **do**

  Sample a batch data $\mathcal{B} = \{(\mathbf{x}_j, \mathbf{b}_j, y_j)\}_j$ from $\mathcal{T}_1$;

  Calculate the loss of detection on this batch according to $\tilde{\mathcal{L}}_3(\boldsymbol{\theta}^t)$;

  Backward the above loss and update parameters $\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \nabla_{\boldsymbol{\theta}^t} \tilde{\mathcal{L}}_3(\boldsymbol{\theta}^t)$;

end

C. Domain and class information preservation

The naive three-stage learning approach described in III-B exists a heavy problem that the latter learning stage could destroy the approximate optimal parameters which is achieved in the former learning stage, as shown in Fig. 3 . An analogous phenomenon has been sufficiently researched in continual learning community and is named catastrophic forgetting [15], appearing in task, domain and class incremental scenarios [16] where a model is expected to learn a sequence of subtasks.

To handle the problem, we adopt Fisher information matrix (FIM), whose effectiveness has been validated by elastic weight consolidation [17] in domain-incremental classification and by IncDet [11] in incremental object detection, to measure the importance of parameters for the seen dataset and use the degree of importance to constrain the changes of parameters. Specifically, if a parameter is very important for the learned dataset $\mathcal{S}_2$, then when we are fine-tuning the model in dataset $\mathcal{T}_1$, this parameter should have never changed in an ideal situation and only less important parameters will be varied for improving performance on the target dataset $\mathcal{T}_1$. In a mathematical perspective to formulate this fine-tuning stage, we need to maximize the posterior $p(\boldsymbol{\theta}|\mathcal{S}_2, \mathcal{T}_1)$ where $\boldsymbol{\theta}$ is the parameters of RoI heads that we aim to fine-tune in the third stage. According to Bayes rule,

$$p(\boldsymbol{\theta}|\mathcal{S}_2, \mathcal{T}_1) = \frac{p(\mathcal{S}_2, \mathcal{T}_1|\boldsymbol{\theta}) \cdot p(\boldsymbol{\theta})}{p(\mathcal{S}_2, \mathcal{T}_1)}, \tag{5}$$

$$p(\mathcal{S}_2|\boldsymbol{\theta}) \cdot p(\boldsymbol{\theta}) = p(\mathcal{S}_2) \cdot p(\boldsymbol{\theta}|\mathcal{S}_2). \tag{6}$$

From the rule of multiplication of probability, we have

$$p(\mathcal{S}_2, \mathcal{T}_1|\boldsymbol{\theta}) = p(\mathcal{T}_1|\mathcal{S}_2, \boldsymbol{\theta}) \cdot p(\mathcal{S}_2|\boldsymbol{\theta}), \tag{7}$$

$$p(\mathcal{S}_2, \mathcal{T}_1) = p(\mathcal{T}_1|\mathcal{S}_2) \cdot p(\mathcal{S}_2). \tag{8}$$

Moreover, the prior probabilities of datasets $\mathcal{S}_2$ and $\mathcal{T}_1$ are independent because $\mathcal{S}_2 \cap \mathcal{T}_1 = \emptyset$, and hence we have

$$p(\mathcal{T}_1|\mathcal{S}_2, \boldsymbol{\theta}) = p(\mathcal{T}_1|\boldsymbol{\theta}). \tag{9}$$

Applying Eq. 7, 6, 8 and 9 in order on Eq. 5, we can derive

$$p(\theta|\mathcal{S}_2, \mathcal{T}_1) = \frac{p(\mathcal{T}_1|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{S}_2)}{p(\mathcal{T}_1|\mathcal{S}_2)}. \tag{10}$$

After adopting negative logarithm on both sides of Eq. 10, the final loss in the third stage is formulated as follows:

$$\begin{aligned} \tilde{\mathcal{L}}_3(\boldsymbol{\theta}) &= -\log p(\boldsymbol{\theta}|\mathcal{S}_2, \mathcal{T}_1) \\ &= -\log p(\mathcal{T}_1|\boldsymbol{\theta}) - \log p(\boldsymbol{\theta}|\mathcal{S}_2) + \log p(\mathcal{T}_1|\mathcal{S}_2). \end{aligned} \tag{11}$$

Since the term $\log p(\mathcal{T}_1|\mathcal{S}_2)$ is a constant, we ignore it when minimizing the final loss. The second term in the right side of Eq. 11 is intractable and can be approximated using Fisher information matrix as in [11] [17]. Considering the base model has had a competitive performance on target domain, we replace model parameters after training on $\mathcal{S}_2$ with those of the base model after training on $\mathcal{D}$ as the pivot values. Finally, we simplify the calculation of loss $\tilde{\mathcal{L}}_3(\boldsymbol{\theta})$ as

$$\tilde{\mathcal{L}}_3(\boldsymbol{\theta}) = \mathcal{L}_3(\boldsymbol{\theta}) + \sum_i \lambda F_{ii}(\theta_i - \theta^*_{\mathcal{D},i})^2, \tag{12}$$

where $\mathcal{L}_3(\boldsymbol{\theta}) = -\log p(\mathcal{T}_1|\boldsymbol{\theta})$ whose calculation obeys Eq. 4, $\theta^*_{\mathcal{D},i}$ is the parameter value of the model after training on dataset $\mathcal{D}$, F_{ii} is a diagonal element of FIM for efficient computation and λ is a tradeoff hyper-parameter. For our task settings, the elements of Fisher information matrix can be calculated as follows

$$F_{ij} = \mathbb{E}_{(\mathbf{x},\cdot,\cdot)\in\mathcal{S}_2}\left[\frac{\partial \log p(\mathbf{x}|\boldsymbol{\phi}, \boldsymbol{\theta})}{\partial \theta_i} \cdot \frac{\partial \log p(\mathbf{x}|\boldsymbol{\phi}, \boldsymbol{\theta})}{\partial \theta_j}\right]. \tag{13}$$

We abbreviate the domain and class information preservation to DCIP and our entire method is summarized in Alg. 1.

IV. Experiments

In this section, we introduce datasets in Section IV-A, implementation details in Section IV-B and compare our method against a state-of-the-art method of UDA-OD on plenty of task settings of our new-proposed task in Section IV-C.

TABLE I: Ablative study from PASCAL VOC to Clipart1k when 20% classes are missing. Upper Bound means training data include annotations of objects of all classes. +2-stage means adding class information learning and domain information recalling stages. +DCIP means adding domain and class information preservation. mAP is the averaged AP50 over 20 classes.

Method	+2-stage	+DCIP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
D-adapt [4]			46.9	54.0	29.6	40.3	44.7	55.1	48.6	21.9	43.6	63.4	28.2	17.5	46.7	70.0	65.0	43.7	29.9	33.2	44.2	55.4	44.1
	✓		51.7	56.0	40.3	43.2	40.1	41.4	50.5	22.5	51.4	52.9	54.8	25.9	43.5	67.0	75.3	51.0	23.4	22.9	38.2	32.6	44.2
	✓	✓	46.75	62.3	41.0	45.2	38.7	34.8	56.2	19.0	51.5	64.5	57.5	25.6	48.7	69.7	71.2	55.8	34.3	18.8	35.2	50.7	**46.4**
Unbiased [3]			34.4	69.1	30.1	33.0	48.5	40.8	24.2	3.0	19.0	17.9	46.7	3.8	34.3	77.9	67.1	53.7	16.2	9.1	17.1	20.0	33.3
	✓		33.3	61.4	37.9	31.8	48.8	31.1	51.4	22.3	54.1	26.7	38.9	13.9	46.3	62.7	74.9	47.4	9.3	4.5	24.2	9.1	36.5
	✓	✓	29.9	56.6	33.9	21.9	45.2	59.3	58.9	18.7	50.1	31.9	49.7	8.4	39.8	68.8	72.2	47.2	10.1	4.5	26.6	10.9	**37.2**
Upper Bound	–	–	56.2	64.6	36.5	52.2	48.0	69.0	57.7	25.3	50.3	67.7	53.7	38.4	55.6	80.1	75.2	52.4	42.6	34.3	60.7	57.9	53.9

Fig. 4: Qualitative results on the Clipart1k when 20% classes are missing. "Baseline" is the D-adapt method.

A. Datasets

We conduct our main experiments on the PASCAL VOC 2007 and 2012 detection datasets and Clipart1k dataset, which all include 20 categories of objects with annotations. In PASCAL VOC 2007, there are 5011 images in trainval split and 4952 in test split, while PASCAL VOC 2012 dataset comprises 11540 images in trainval split and its test split is not available in public. For Clipart1k that consists of 1000 clipart images, it is split into train and test splits both with 500 images. Same as the conventional settings, we treat the PASCAL VOC 2007 and 2012 as the source domain and the Clipart1k as the target domain. We view the last percent of 20 categories in alphabetical order as the missing classes. The test split of Clipart1k is reserved for only testing and the train split of remaining classes are accessible in training phase.

B. Implementation Details

We adopt ResNet101 [18] as the backbone network and the implementation details of the first stage are consistent with those in [4]. mAP@0.5 is chosen as the evaluation metric [4] [11]. We set learning rate as 2.5×10^{-5} and use exponential annealing learning-rate schedule with $\gamma = 0.2$. After training in the first stage, the models are fine-tuned 2 epochs in the second stage and 10 epochs in the third stage. λ is set as 0.1.

C. Results Analysis

We first show ablative study from PASCAL VOC to Clipart1k when 20% classes are missing. As shown in Table I, applying our 3-stage training with modified EWC could boost performance on the base of D-adapt [4] and Unbiased [3] and the effectiveness of every module is validated. We show a few qualitative results in Fig. 4 as well. As can be seen from the first column in Fig. 4, our entire method can detect the blue diningtable when compared to the naive 3-stage training approach, and the girl with blond hair and the boy in blue in the center of the image when compared to the baseline. In the second column of Fig. 4, the baseline mistakenly detects a tvmonitor but cannot detect the white horse in the center which is also lost for the 3-stage training but detected via our entire method. The last column of Fig. 4 shows us a failure example.The models are all confused by local characteristics of the aeroplane, whose partial image is indeed akin to a chair even for a human. This failure is thought to be originated from that CNNs focus on local images overmuch and overlook global information.

TABLE III: Effects of λ from PASCAL VOC to Clipart1k when missing 20% classes.

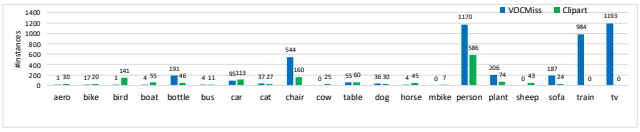
λ	mAP	AP	AP75
0.01	45.1	22.8	18.9
0.1	**46.4**	**23.6**	**20.4**
1	43.7	21.6	18.0
10	41.7	20.6	17.7

Moreover, the effect of λ is shown in Table III and the reported results in Table I and II are both based on $\lambda = 0.1$.

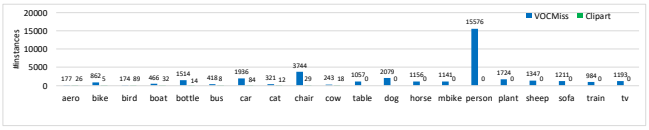
To investigate the effect of the number of missing classes, we further conduct experiments with 10%, 20%, 30% , 40% and 50% missing classes. Compared with the baseline, our method achieves 2.3, 2.0 and 2.2 increases in mAP when 20%, 30% and 40% classes are missing, respectively, as shown in Table II. In the scenario of missing 10% or 50% classes, the performance of our method is equivalent or close to the baseline's, and we think the former is caused by severe class-imbalance of dataset $\mathcal{S}_2$ as shown in Fig. 5a while the latter suffers from the lack of instances in dataset $\mathcal{T}_1$ as shown in Fig. 5b. Along with the increase of missing classes, models' performance have a downward trend, but the decreased performance between scenarios of missing 10% and 20% classes is minor. This is because the number of instances in $\mathcal{T}_1$ scarcely changes when the percentage of missing classes varies from 10% to 20%, which is revealed in Fig. 5c. There is an obvious decrease of mAP between missing 20% and 30% classes, to which the performance decrease of "mbike" contributes the most. We think the reasons are the number of "person" instances is larger than any other classes as shown in Fig. 5, and

TABLE II: Results of missing different percentages from PASCAL VOC to Clipart (ResNet101). The results are evaluated on test split of Clipart1k. The "base" denotes D-adapt method.

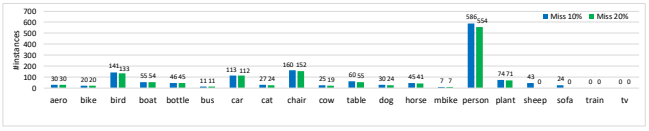
Test on Clipart		aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
miss 10%	base	46.0	56.5	33.8	40.1	42.2	63.5	47.0	18.6	47.1	62.8	38.5	15.2	56.1	71.1	69.5	46.6	21.4	37.1	52.8	51.1	**45.8**
	+ours	45.8	61.3	39.5	45.3	42.9	49.9	52.0	22.6	52.2	61.7	54.6	28.5	53.7	64.3	76.1	54.1	38.9	23.6	20.7	29.3	**45.8**
miss 20%	base	46.9	54.0	29.6	40.3	44.7	55.1	48.6	21.9	43.6	63.4	28.2	17.5	46.7	70.0	65.0	43.7	29.9	33.2	44.2	55.4	44.1
	+ours	46.75	62.3	41.0	45.2	38.7	34.8	56.2	19.0	51.5	64.5	57.5	25.6	48.7	69.7	71.2	55.8	34.3	18.8	35.2	50.7	**46.4**
miss 30%	base	41.8	53.1	28.1	37.3	43.8	76.6	41.4	5.5	42.1	51.3	38.2	9.1	36.7	45.3	53.2	45.6	13.8	27.8	43.1	52.9	39.3
	+ours	42.3	58.6	24.0	39.3	34.9	62.8	49.4	13.4	46.0	75.8	55.0	4.3	37.3	74.8	44.5	39.9	20.4	13.6	42.5	47.7	**41.3**
miss 40%	base	43.4	46.0	31.4	39.6	40.4	52.3	43.2	17.7	42.8	54.7	28.3	10.3	46.4	74.4	58.8	45.4	22.8	29.5	40.1	51.9	41.0
	+ours	47.2	59.6	30.5	42.4	41.9	76.4	49.2	17.1	48.7	60.5	53.2	3.8	43.4	65.0	44.8	41.9	22.5	26.4	42.3	47.6	**43.2**
miss 50%	base	43.7	52.6	26.5	38.6	39.1	73.4	41.0	6.0	41.6	41.4	32.0	6.4	44.9	59.4	49.1	42.2	19.8	29.1	51.0	51.5	39.5
	+ours	49.8	63.7	23.9	39.2	35.5	79.8	55.1	14.4	42.2	46.3	41.3	7.0	41.8	53.0	43.5	42.0	15.9	16.7	31.0	50.0	**39.6**
D-adapt [4]		56.4	63.2	42.3	40.9	45.3	77.0	48.7	25.4	44.3	58.4	31.4	24.5	47.1	75.3	69.3	43.5	27.9	34.1	60.7	64.0	49.0
Upper Bound		56.2	64.6	36.5	52.2	48.0	69.0	57.7	25.3	50.3	67.7	53.7	38.4	55.6	80.1	75.2	52.4	42.6	34.3	60.7	57.9	53.9



(a) Comparison between $\mathcal{S}_2$ and $\mathcal{T}_1$ when 10% classes are missing



(b) Comparison between $\mathcal{S}_2$ and $\mathcal{T}_1$ when 50% classes are missing



(c) Comparison between $\mathcal{T}_1$ when 10% and 20% classes are missing

Fig. 5: Comparisons of the number of instances in datasets.

the spurious correlations [19] between "person" and "mbike". Because models have more opportunities to meet "person" instances in the training, models will pay more attention to "person" instances. When the dataset misses 30% classes, the number of "person" instances in target domain will be zero whereas samples with "mbike" can be still available. Due to a model trained in source domain relies on "person" instances to detect "mbike", the model loses the ability to detect "mbike" without the help of annotations when missing "person" in target domain, causing the large reduction of the mAP for the base model. But our method can utilize annotations of "mbike", and hence mitigates this problem.

V. Conclusion

To the extent of our knowledge, this work is the first in the literature to research a new cross-domain object detection with partially missing classes in target domain. We describe the new task and its relationships with other analogous tasks accurately. Aiming at two requirements of domain adaptation and class generalization, we devise a 3-stage learning approach with DCIP. Experiments validate our approach is effective and can improve performance of the base methods as a plugin. We think models with the ability to capture global information will be a promising avenue to explore, e.g. Vision Transformers.

Acknowledgment

This research was supported in part by the National Natural Science Foundation of China under Grant 62071086, Sichuan Science and Technology Program under Grant 2021YFG0296 and the Science and Technology Innovation (seedling project) Cultivation and Invention Creation project in Sichuan province under Grant 2021015.

References

[1] N. Inoue *et al.*, "Cross-domain weakly-supervised object detection through progressive domain adaptation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[2] Y. Chen *et al.*, "Domain adaptive faster R-CNN for object detection in the wild," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[3] Y.-C. Liu *et al.*, "Unbiased teacher for semi-supervised object detection," in *International Conference on Learning Representation (ICLR)*, 2021.

[4] J. Jiang *et al.*, "Decoupled adaptation for cross-domain object detection," in *International Conference on Learning Representation (ICLR)*, 2022.

[5] S. Antonelli *et al.*, "Few-shot object detection: A survey," *ACM Computing Surveys*, Feb 2022.

[6] X. Yan *et al.*, "Meta R-CNN: Towards general solver for instance-level low-shot learning," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[7] Y. Zhang *et al.*, "Zero-shot object detection with transformers," in *IEEE International Conference on Image Processing (ICIP)*, 2021.

[8] C. Yan *et al.*, "Semantics-guided contrastive network for zero-shot object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[9] M. Ishii *et al.*, "Partially zero-shot domain adaptation from incomplete target data with missing classes," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[10] A. Zhao *et al.*, "Domain-adaptive few-shot learning," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021.

[11] L. Liu *et al.*, "IncDet: In defense of elastic weight consolidation for incremental object detection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 6, pp. 2306–2319, 2021.

[12] M. Bianchini *et al.*, "A comparative study of inductive and transductive learning with feedforward neural networks," in *AIIA 2016 Advances in Artificial Intelligence*, 2016, pp. 283–293.

[13] S. Ren *et al.*, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[14] X. Wang *et al.*, "Frustratingly simple few-shot object detection," in *International Conference on Machine Learning (ICML)*, 2020.

[15] M. McCloskey *et al.*, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychology of Learning and Motivation*, vol. 24, pp. 109–165, 1989.

[16] G. M. van de Ven *et al.*, "Three scenarios for continual learning," in *arXiv preprint arXiv:1904.07734*, 2019.

[17] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 114, no. 13, pp. 3521–3526, 2017.

[18] K. He *et al.*, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[19] V. Agarwal *et al.*, "Towards causal VQA: Revealing and reducing spurious correlations by invariant and covariant semantic editing," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.