Sequential Instance Refinement for Cross-Domain Object Detection in Images

Jin Chen, Xinxiao Wu, *Member, IEEE*, Lixin Duan, and Lin Chen

***Abstract*— Cross-domain object detection in images has attracted increasing attention in the past few years, which aims at adapting the detection model learned from existing labeled images (source domain) to newly collected unlabeled ones (target domain). Existing methods usually deal with the cross-domain object detection problem through direct feature alignment between the source and target domains at the image level, the instance level (*i.e.,* region proposals) or both. However, we have observed that directly aligning features of all object instances from the two domains often results in the problem of negative transfer, due to the existence of (1) *outlier target instances* that contain confusing objects not belonging to any category of the source domain and thus are hard to be captured by detectors and (2) *low-relevance source instances* that are considerably statistically different from target instances although their contained objects are from the same category. With this in mind, we propose a reinforcement learning based method, coined as sequential instance refinement, where two agents are learned to progressively refine both source and target instances by taking sequential actions to remove both outlier target instances and low-relevance source instances step by step. Extensive experiments on several benchmark datasets demonstrate the superior performance of our method over existing state-of-the-art baselines for cross-domain object detection.**

***Index Terms*— Cross-domain object detection, negative transfer, reinforcement learning.**

I. Introduction

OBJECT detection is one of the most fundamental and challenging task in computer vision and has been an active research area for several decades [1]. The goal of object detection is to simultaneously localize and recognize all object instances belonging to the pre-defined categories in an image. It supports many applications, such as autonomous driving [2], [3], intelligent video surveillance [4], [5] and so on. Recently, with the development of deep learning [6], object detection has made remarkable breakthroughs [7]–[17], and achieved superior performances on large benchmark datasets [18], [19].

Manuscript received March 1, 2020; revised October 9, 2020 and January 25, 2021; accepted March 12, 2021. Date of publication March 26, 2021; date of current version April 1, 2021. This work was supported in part by the Natural Science Foundation of China (NSFC) under Grant 62072041 and Grant 61673062. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Soma Biswas. *(Corresponding author: Xinxiao Wu.)*

Jin Chen and Xinxiao Wu are with the Beijing Laboratory of Intelligent Information Technology, Beijing Institute of Technology, Beijing 100081, China, and also with the School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: wuxinxiao@bit.edu.cn).

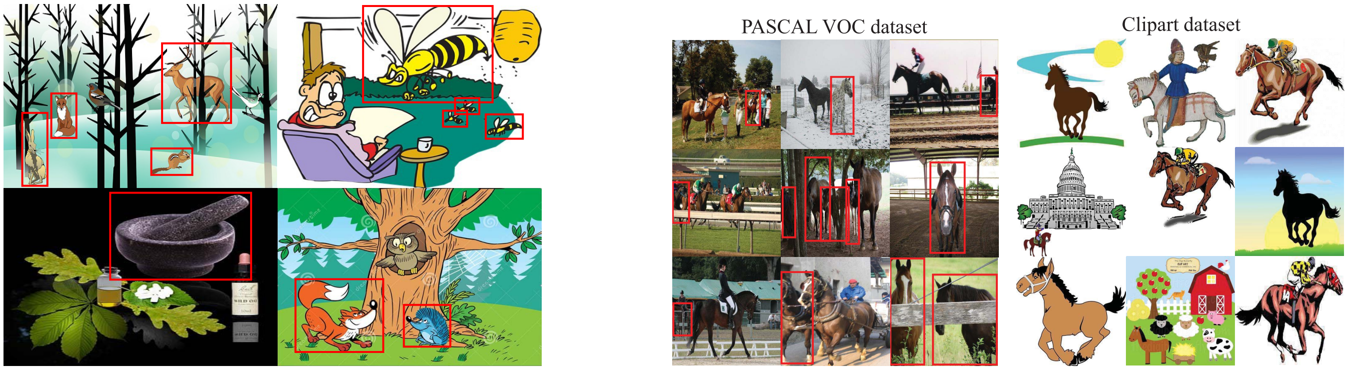
Lixin Duan is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China.

Lin Chen is with Wyze Labs, Kirkland, WA 98034 USA.

Digital Object Identifier 10.1109/TIP.2021.3066904

Despite its recent success, object detection still faces some problems in real-world applications. The deep learning based methods depend heavily on abundant manually labeled data. But for a new task, images are often unlabeled and it is time-consuming and labor-intensive to annotate them. To address this problem, cross-domain object detection is proposed, which aims at improving the detection performance on the unlabeled images (target domain) by leveraging an adaptive detector learned on a fully annotated, different but related domain (source domain). Since there exists a considerable domain shift between the source domain and the target domain due to the statistical differences caused by backgrounds, viewpoints, illumination and object appearances, directly applying object detection model trained on the source domain to the target domain would generally lead to performance degradation [20]. Thus several methods of cross-domain object detection are proposed to reduce the domain shift by learning domain-invariant features [21]–[24]. Among them, [21] is the first work to address the cross-domain object detection task and performs both image-level and instance-level feature alignment to reduce the domain mismatch. Several later works [22]–[24] focus on aligning instance features between the source and target domains to learn an adaptive object detector.

Nevertheless, directly aligning instance features between different domains is often prone to negative transfer, due to the existence of *outlier target instances* and *low-relevance source instances*. The outlier target instances refer to target instances containing objects that do not belong to any category of the source domain. The low-relevance source instances refer to source instances that are quite dissimilar to the target ones although they contain objects from the same categories as the target domain. For better understanding, we present several examples of outlier target instances and low-relevance source instances as illustrated in Fig. 1, where the PASCAL VOC dataset is considered as the source domain and the Clipart dataset as the target domain. The outlier target instances and low-relevance source instances are marked with red boxes in Fig. 1(a) and Fig. 1(b), respectively. For example, the bees with red boxes in the upper right corner of Fig. 1(a) are outlier target instances because the "bee" category is not in the category space of the source domain. In the left part of Fig. 1(b), horses with red boxes represent low-relevance source instances that are less similar to the target domain (shown in the right part) due to different appearances, viewpoints, *etc.*



(a) Examples of outlier target instances in the Clipart dataset.

(b) Examples of low-relevance source instances in the PASCAL VOC dataset.

Fig. 1. Examples of outlier target instances and low-relevance source instances on the PASCAL VOC→Clipart setting, which are marked with red boxes. (a) Outlier target instances contain objects that do not belong to any category of the source domain. There are 20 categories in the source domain, including "aeroplane", "bicycle", "bird", "boat", "bus", "car", "cat", "chair", "cow", "table", "dog", "horse", "motorbike", "person", "plant", "sleep", "sofa", "train", and "tv". Hence, target instances, containing "fox", "bee", *etc.*, are outlier target instances. (b) For the "horse" category, the low-relevance source instances from the PASCAL VOC dataset are shown in the left part, and the target instances from the Clipart dataset are presented in the right part for comparison. The low-relevance source instances (in the red boxes) are less similar to the target ones due to different appearances, viewpoints, *etc*.

To tackle these outlier target instances and low-relevance source instances in cross-domain object detection, we propose a reinforcement learning based method, called sequential instance refinement (SIR), under the framework of Faster R-CNN [8]. Our SIR trains two agents (defined as *T-agent* and *S-agent*) to progressively refine the source and target instances by taking sequential actions to remove the outlier target instances from the target domain and the low-relevance source instances from the source domain.

Specifically, T-agent is responsible for selecting out outlier target instances from the target domain to enhance the positive transfer. At each selection, T-agent takes an action to remove one target instance according to its Q-value from the target domain and the reward of taking this action is fed back to T-agent to update the selection policy. If the selected target instance is an outlier target instance, the mismatch of data distributions between the source and target domains will be reduced and thus the performance can be improved. It is worth noting that the relevance of outlier target instances to the source domain is low as the label space of source domain is different to that of outlier target instances. So we utilize the relevance of the target instances to the source domain as the reward of T-agent. Similarly, S-agent aims at removing the low-relevance source instances from the source domain to reduce the negative transfer caused by irrelevant source instances. Hence, the reward function of S-agent (*resp.,* T-agent) is defined by a domain classifier that measures the relevance of the source instances to the target domain (*resp.,* the target instances to the source domain.

By means of sequential actions taken by S-agent and T-agent, both target and source instances are refined and then used to train the domain classifier in an adversarial manner. In return, the rewards of the two agents computed by the domain classifier are further calibrated to further improve selection policies. Owing to such a progressive procedure, our method can simultaneously refine the source and target instances by removing unrelated ones, which helps alleviate negative transfer in cross-domain object detection.

We summarize our contributions as follows:

- To the best of our knowledge, we make the first attempt to explicitly address the negative transfer problem in cross-domain object detection through refining both source and target instances.
- We propose sequential instance refinement (SIR) based on reinforcement learning to progressively remove outlier target instances and low-relevance source instances by taking sequential actions of two agents.
- Evaluations on several benchmark datasets demonstrate that our SIR achieves superior performance over the existing state-of-the-art methods.

II. Related Work

A. Cross-Domain Object Detection

Many existing cross-domain object detection methods resort to matching distributions of the image or instance features between the source and target domains [21]–[24], [26]. Chen *et al.* [21] are the first to tackle the cross-domain object detection and construct two domain classifiers on both image and instance levels to reduce the domain mismatch. Zhu *et al.* [24] first mine the discriminative regions that are directly pertinent to object detection and then align those regions in the source and target domains to reduce the domain shift. Saito *et al.* [22] propose weak alignment to focus on the similar part of images and strong alignment to focus on the local fields of the feature map. In [23], the object relationship is integrated into the mean teacher paradigm and the relation graphs between the source and target domains are matched to reduce the domain shift. Hsu *et al.* [26] first utilize CycleGAN [27] to generate the intermediate domain via translating source images to the target domain and then align the data distributions of the intermediate domain and the target domain via adversarial training at the feature level. Several methods focus on generating pseudo-labeled target data to fine-tune the source detector [28]–[30]. [28] utilizes the tracking information to label the target data and refines the label of target data with the pseudo label predicted by the source detector. In [29], an object classifier is trained with bounding boxes of the source domain and then is used to label target instances detected by the detector trained on the source domain. Kim *et al.* [30] introduce a weak self-training method to diminish the effects of inaccurate pseudo-labels and propose an adversarial background score regularization to extract discriminative features for target backgrounds.

Rather than directly matching the data distributions of the source and target domains, we train two agents to select out the outlier target instances and low-relevance source instances, respectively, thus alleviating the negative transfer and improving the detection performance.

B. Domain Adaptation

Domain adaptation leverages the knowledge of the existing labeled domain to enhance the classification performance on the unlabeled but interested domain. Traditional domain adaptation methods can be roughly divided into three categories: instance-based [31], [32], feature-based [33]–[37] and parameter-based domain adaptation methods [38]–[41]. Recently, deep neural networks have made great process due to its strong power in feature learning. Several deep domain adaptation methods are proposed to learn domain-invariant features by minimizing the Maximum Mean Discrepancy (MMD) [42]–[44] or regularizing features by Batch Normalization (BN) layer [45]–[48]. Other methods [25], [49]–[55] introduce the generative adversarial learning [56] for domain adaptation, which aim at making the feature representations between the source and target domains as non-discriminative as possible.

All the aforementioned works focus on domain adaptation in image classification whereas our method focus on domain adaptation in object detection that simultaneously localizes and classifies multiple objects in an image.

C. Reinforcement Learning for Object Detection

Reinforcement learning has been widely used in computer vision recently [57]–[60]. Some approaches [61]–[63] apply reinforcement learning to object detection to reduce the cost of proposal generation and generating more correct proposals. [64] utilizes a class-specific agent to deform a bounding box via a sequence of transformation actions until localizing the object, where the reward is defined as the difference in Intersection-over-Union (IoU) after taking the action. [61] applies reinforcement learning to sample more image regions for better accuracy and stop the region search when the agent is sufficiently confident about the location of the object. It is more powerful than exhaustive spatial hypothesis search such as sliding windows. [62] designs a reinforcement learning based region proposal network to generate proposals by automatically deciding when to stop the search process, where the parameters of the policy network and the detector are jointly learned.

All those methods adopt the reinforcement learning to generate object proposals. In contrast, our method utilizes the reinforcement learning to make selections on proposals for effectively handling the negative transfer in cross-domain object detection.

III. Proposed Method

In unsupervised cross-domain object detection, we are given a labeled source domain $\mathcal{D}_s = \{(x_i^s, \boldsymbol{y}_i^s)|_{i=1}^{N_s}\}$ with N_s images and an unlabeled target domain $\mathcal{D}_t = \{x_i^t|_{i=1}^{N_t}\}$ with N_t images, where x_i^s and x_i^t represent the i-th images in $\mathcal{D}_s$ and $\mathcal{D}_t$, respectively. $\boldsymbol{y}_i^s$ is the set of annotations of objects in x_i^s, *i.e.,* $\boldsymbol{y}_i^s = \{\boldsymbol{y}_{i,1}^s, \boldsymbol{y}_{i,2}^s, \cdots, \boldsymbol{y}_{i,L}^s\}$, where L is the number of contained objects and $\boldsymbol{y}_{i,l}^s$ is the corresponding annotation of the l-th object in x_i^s. Each annotation $\boldsymbol{y}_{i,l}^s$ is formulated as a 5-tuple, *i.e,* $\boldsymbol{y}_{i,l}^s \in \mathbb{R}^{5\times 1}$, consisting of the category label of the l-th object, and the coordinates of the upper left corner, height and width of its corresponding bounding box.

In this work, we employ Faster R-CNN [8] as our base detector due to its robustness and flexibility. The region proposals (*i.e.,* object instances) are generated via a region proposal network. Let $\mathcal{P}_i^s = \{p_{i,j}^s|_{j=1}^{N_s^p}\}$ represent a set of region proposals in x_i^s, where $p_{i,j}^s$ is the feature map of the j-th region proposal, and N_s^p is the number of source proposals. Let $\mathcal{P}_i^t = \{p_{i,j}^t|_{j=1}^{N_t^p}\}$ represent a set of region proposals in x_i^t, where N_t^p is the number of target proposals.

In order to improve the detection performance on the target domain, we aim to select out both *outlier target region proposals* that contain objects not belonging to any category of the source domain and *low-relevance source region proposals* that have low relevance to the target domain. We define two agents to make selections via a sequence of actions under a deep reinforcement learning framework. Specifically, T-agent learns to select out outlier target region proposals from $\mathcal{P}_i^t$ for generating an updated set $\hat{\mathcal{P}}_i^t$. S-agent learns to select out low-relevance source region proposals from $\mathcal{P}_i^s$ for generating an updated set $\hat{\mathcal{P}}_i^s$. Then, we align $\hat{\mathcal{P}}_i^t$ and $\hat{\mathcal{P}}_i^s$ to learn domain-invariant features for reducing the domain shift.

The architecture of our SIR is shown in Fig. 2, which consists of a Faster R-CNN as the base detector and a sequential instance refinement module to refine target and source instances by T-agent and S-agent, respectively.

A. Faster R-CNN

Faster R-CNN [8] is a two-stage detector and consists of three major components: shared convolutional layers, a region proposal network (RPN) and a region of interest (RoI) based classifier. The shared convolutional layers firstly extract a feature map of an input image. Then RPN generates a set of region proposals with pre-defined anchor boxes. Finally, the RoI-based classifier predicts categories of those region proposals. The loss of Faster R-CNN is summarized as

$$\mathcal{L}_{det} = \mathcal{L}_{rpn} + \mathcal{L}_{roi}, \tag{1}$$

where $\mathcal{L}_{rpn}$ and $\mathcal{L}_{roi}$ are the training losses of RPN and the RoI-based classifier, respectively. $\mathcal{L}_{rpn}$ and $\mathcal{L}_{roi}$ both have two loss terms: a cross-entropy loss about mis-classification error and a regression loss about localization error. The detection and localization in RPN take no account of object categories

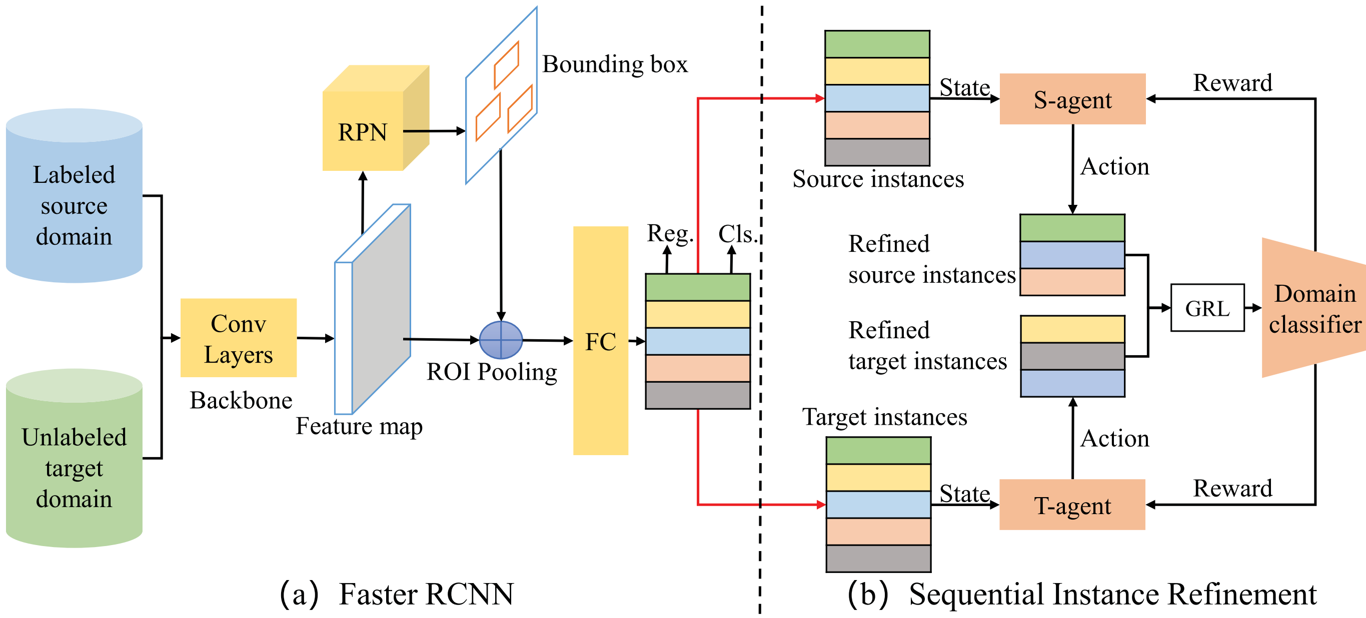


Fig. 2. An overview of our method. Faster R-CNN is used as the base detector shown in the left part, and the right part is a sequential instance refinement module. T-agent and S-agent are built on the target and source domains, respectively, to refine instances. A domain classifier with the gradient reverse layer (GRL) [25] is constructed to provide rewards for the two agents, which is trained with refined source and target proposals.

and the RoI-based classifier is trained to predict the object categories.

B. Sequential Instance Refinement

Due to the large size of proposal set, we randomly split the proposal set $\mathcal{P}_i^t$ of each target image x_i^t into N_c^t candidate sets, denoted as $\mathcal{P}_i^t = \bigcup_{n=1}^{N_c^t} \mathcal{C}_{i,n}^t$ with $\mathcal{C}_{i,n}^t = \{p_{i,n,k}^t|_{k=1}^{N_c}\}$, where $p_{i,n,k}^t$ represents the k-th region proposal in the n-th candidate set of x_i^t and is extracted by a RoI pooling layer of Faster R-CNN, and N_c is the size of candidate set. Similarly, the proposal set $\mathcal{P}_i^s$ of each source image x_i^s is also randomly split into N_c^s candidate sets, denoted as $\mathcal{P}_i^s = \bigcup_{n=1}^{N_c^s} \mathcal{C}_{i,n}^s$ with $\mathcal{C}_{i,n}^s = \{p_{i,n,k}^s|_{k=1}^{N_c}\}$, where $p_{i,n,k}^s$ represents the k-th proposal in the n-th candidate set of x_i^s.

Each episode observed by the agent consists of a sequence of selections for one candidate set. In the selection process for each candidate set $\mathcal{C}_{i,n}^t$, at time e, T-agent first observes the current state s_e^t and takes an action a_e^t to remove a region proposal from $\mathcal{C}_{i,n}^t$. Then, T-agent receives the next state s_{e+1}^t and a reward r_e^t of taking a_e^t. When this episode terminates at time E, T-agent completes the selections of $\mathcal{C}_{i,n}^t$, and the current candidate set can be treated as an optimal candidate set, denoted as $\hat{\mathcal{C}}_{i,n}^t$. When T-agent finishes selections for all the candidate sets of $\mathcal{P}_i^t$, we can obtain the optimal target proposal set $\hat{\mathcal{P}}_i^t = \bigcup_{n=1}^{N_c^t} \hat{\mathcal{C}}_{i,n}^t$ for the target image x_i^t. In a similar way, we can obtain the optimal source proposal set $\hat{\mathcal{P}}_i^s = \bigcup_{n=1}^{N_c^s} \hat{\mathcal{C}}_{i,n}^s$ for the source image x_i^s.

1) State: Since each agent makes selections on a single candidate set, the state of each agent is defined by the region proposals in the corresponding candidate set. Specifically, the state of T-agent is represented by the feature vectors of region proposals in a candidate set $\mathcal{C}_{i,n}^t$, denoted as $\boldsymbol{s}^t = [\boldsymbol{f}_{i,n,1}^t, \cdots, \boldsymbol{f}_{i,n,N_c}^t] \in \mathbb{R}^{d \times N_c}$ where $\boldsymbol{f}_{i,n,k}^t$ is a d-dimensional feature vector of the region proposal $p_{i,n,k}^t$. When T-agent removes the k-th region proposal $p_{i,n,k}^t$ from the candidate set $\mathcal{C}_{i,n}^t$, the corresponding feature vector $\boldsymbol{f}_{i,n,k}^s$ in $\boldsymbol{s}^t$ is replaced with a zero-valued feature vector to keep a constant size of $\boldsymbol{s}^t$. Similarly, the state of S-agent is represented by $\boldsymbol{s}^s = [\boldsymbol{f}_{i,n,1}^s, \cdots, \boldsymbol{f}_{i,n,N_c}^s] \in \mathbb{R}^{d \times N_c}$ where $\boldsymbol{f}_{i,n,k}^s$ is a d-dimensional feature vector of the region proposal $p_{i,n,k}^s$.

2) Action: The actions of T-agent are denoted by $A^t = \{1, \cdots, N_c\}$ where the action k means selecting out the k-th proposal $p_{i,n,k}^t$ from $\mathcal{C}_{i,n}^t$. The actions of S-agent are similarly denoted by $A^s = \{1, \cdots, N_c\}$, where the action k means selecting out the k-th proposal $p_{i,n,k}^s$ from $\mathcal{C}_{i,n}^s$.

Since S-agent and T-agent make decisions in a similar way, we use $s_e \in \{s_e^t, s_e^s\}$ to denote the state of T-agent or S-agent, $a_e \in \{a_e^t, a_e^s\}$ to denote the action of T-agent or S-agent at time e, and $A \in \{A^t, A^s\}$ to denote the action set of T-agent or S-agent. The agent takes an optimal action a_e to maximize the accumulated rewards $R_e = \sum_{e'=e}^{E} \gamma^{e'-e} r_{e'}$, where $r_{e'}$ is the immediate reward of taking action $a_{e'}$ under state $s_{e'}$, E is the terminal time of an episode, and γ is a discount factor. We apply a deep Q-learning network (DQN) to estimate the accumulated rewards by learning the action-value function $Q(s_e, a_e)$. The agent takes an action a_e^* from A to get the maximum accumulated rewards via a policy defined by

$$a_e^* = \max_{a_e} Q(\boldsymbol{s}_e, a_e). \tag{2}$$

3) Reward: For T-agent, the reward of taking action a_e^t is determined by the relevance of the selected target proposal p_{i,n,a_e^t} to the source domain since the outlier target instances are less relevant to the source domain than the other target ones. Similarly, for S-agent, the reward of taking action a_e^s is determined by the relevance of the selected source proposal p_{i,n,a_e^s} to the target domain.

So we propose a domain classifier whose classification score can be used to measure the relevance of the target proposal p_{i,n,a_e^t} to the source domain or the relevance of the source proposal p_{i,n,a_e^s} to the target domain. Specifically, we adopt a patch-based domain classifier D that predicts multiple domain labels for each pixel of a region proposal. Let W and H denote the width and height of a region proposal, respectively. The output of D is a domain prediction map with the size of $W \times H$, where $D(p_{i,n,k}^t)_{(w,h)}$ and $D(p_{i,n,k}^s)_{(w,h)}$ denote domain predictions of the pixel (w, h) of region proposals $p_{i,n,k}^t$ and $p_{i,n,k}^s$, respectively. Given the optimal region proposal sets of $\hat{\mathcal{P}}_i^t$ and $\hat{\mathcal{P}}_i^s$, we utilize a least-squares loss to train the domain classifier by following [65], which is formulated as

$$\mathcal{L}_{adv} = \sum_{i,n,k,w,h} D^2(p_{i,n,k}^s)_{(w,h)} + \sum_{i,n,k,w,h} \left(1 - D(p_{i,n,k}^t)_{(w,h)}\right)^2 \tag{3}$$

When $D(p_{i,n,k}^t)_{(w,h)} = 1$, the pixel (w, h) of $p_{i,n,k}^t$ is predicted as coming from the target domain. When $D(p_{i,n,k}^t)_{(w,h)} = 0$, the pixel (w, h) of $p_{i,n,k}^t$ is predicted as coming from the source domain. We employ a gradient reverse layer (GRL) [66] to conduct the adversarial training between D and the backbone network of Faster R-CNN. Specifically, D is trained by the ordinary gradient descent to minimize $\mathcal{L}_{adv}$ and the backbone network of Faster R-CNN is updated with the gradient whose sign is reversed through the GRL layer to maximize $\mathcal{L}_{adv}$.

With the output of D, the relevance measure function $\varphi(p)$ is formulated as

$$\varphi(p) = \begin{cases} \frac{1}{W \times H} \sum_{(w,h)} D(p)_{(w,h)}, & p \in \hat{\mathcal{P}}_i^s \\ 1 - \frac{1}{W \times H} \sum_{(w,h)} D(p)_{(w,h)}, & p \in \hat{\mathcal{P}}_i^t. \end{cases} \tag{4}$$

where $\frac{1}{W \times H} \sum_{(w,h)} D(p)_{(w,h)}$ is the average domain predictions of all pixels of region proposal p. The larger value of $\varphi(p)$ means that the region proposal p is more similar to the opposite domain.

With the relevance measure function $\varphi(p)$, the reward of action a_e^t is

$$r_e^t = \begin{cases} 1, & \varphi(p_{i,n,a_e^t}^t) < \tau \\ -1, & \text{otherwise}, \end{cases} \tag{5}$$

where the action a_e^t corresponds to removing the target region proposal $p_{i,n,a_e^t}^t$ from $\mathcal{C}_{i,n}^t$, and τ is a threshold. As $\varphi(p_{i,n,a_e^t}^t) < \tau$ means that the selected target region proposal $p_{i,n,a_e^t}^t$ is less relevant to the source domain, a positive reward is given to T-agent. The reward of action a_e^s is

$$r_e^s = \begin{cases} 1, & \varphi(p_{i,n,a_e^s}^s) < \tau \\ -1, & \text{otherwise}, \end{cases} \tag{6}$$

where action a_e^s corresponds to removing source region proposal $p_{i,n,a_e^t}^t$ from $\mathcal{C}_{i,n}^s$. When S-agent takes an action of selecting out a region proposal with lower relevance to the target domain, *i.e.*, $\varphi(p_{i,n,a_e^s}^s) < \tau$, a positive reward is given to S-agent. For both S-agent and T-agent, we quantify the reward to 1 and -1 to help the agent clearly distinguish good or bad actions.

4) Loss Function Based on Q-Values: The DQN is trained with the temporal difference error, formulated as

$$\mathcal{L}_q = \mathbb{E}_{\boldsymbol{s}_e, a_e}\left[\left(V(\boldsymbol{s}_e) - Q(\boldsymbol{s}_e, a_e)\right)^2\right], \tag{7}$$

where $Q(\boldsymbol{s}_e, a_e)$ is the output Q-value of action a_e under the current state s_e. $V(\boldsymbol{s}_e)$ is the target value of $Q(\boldsymbol{s}_e, a_e)$, given by

$$V(\boldsymbol{s}_e) = \mathbb{E}_{\boldsymbol{s}_{e+1}}\left[r_e + \gamma \max_{a_{e+1}} Q\left(\boldsymbol{s}_{e+1}, a_{e+1} | \boldsymbol{s}_e, a_e\right)\right], \tag{8}$$

where the first term r_e is the immediate reward of taking the action a_e at time e, the second term is the future reward estimated by the current deep Q-learning network with the next state $\boldsymbol{s}_{e+1}$ as input at time $e + 1$, and γ is a discount factor of reward.

C. Training

With the detection loss $\mathcal{L}_{det}$ in Eq. (1), adversarial loss $\mathcal{L}_{adv}$ in Eq. (3) and deep Q-learning loss $\mathcal{L}_q^s$ and $\mathcal{L}_q^t$ for the two agents from both domains as in Eq. (7), the overall objective function is given by

$$\mathcal{L} = \mathcal{L}_{det} + \mathcal{L}_{adv} + \mathcal{L}_q^s + \mathcal{L}_q^t. \tag{9}$$

We use the ϵ-greedy strategy [67] and the experience replay strategy [68] to train S-agent and T-agent. Specifically, the ϵ-greedy strategy is used to balance the exploration and exploitation of an agent, which refers to that the agent has a certain probability to perform random actions. The selection policy of the agent (defined in Eq. (2)) is then rewritten by

$$a_e^* = \begin{cases} \max_{a_e} Q(\boldsymbol{s}_e, a_e), & \text{if } \lambda \geq \epsilon \\ a_e', & \text{otherwise} \end{cases} \tag{10}$$

where ϵ represents the probability of the agent to perform exploration and λ is a random variable drawn from $[0, 1]$. When $\lambda \geq \epsilon$, the agent takes actions by Eq. (2). Otherwise, the agent performs exploration by taking a randomly action a' from the action set A, which can expand the solution space and avoid falling into a local optimal solution. The training of DQN requires data to be independent and identically distributed (i.i.d.) while the data obtained in the training process is strongly correlated sequentially. Hence, the experience replay strategy [68] is exploited to break the correlation between samples, which stores experiences in the experience pool and samples from the experience pool when updating the model. The whole training process of our SIR is summarized in Algorithm 1.

D. Discussion

In this paper, we investigate alleviating the negative transfer in cross-domain object detection via selecting out the outlier target instances and the low-relevance source instances. A reinforcement learning paradigm is applied to automatically learn policies for selecting out the two types instances. We remark the advantages of reinforcement learning as follows. First, the policies learned by reinforcement learning is optimized in a sequential decision process with the guidance of the

Algorithm 1: Sequential Instance Refinement

Input: Labeled source domain $D_s = \{(x_i^s, \boldsymbol{y}_i^s)\}_{i=1}^{N_s}$;
Unlabeled target domain $D_t = \{(x_i^t)\}_{i=1}^{N_t}$;
Tradeoff parameters: γ, τ, ϵ;
The size of candidate set: N_c;
Terminal time of one episode: E.
Output: Adaptive Faster R-CNN.

Initialize the experience pools of T-agent and S-agent $M^t = \emptyset$,$M^s = \emptyset$;
while *not converge* **do**
  Generate the proposal set $\mathcal{P}_i^t$;
  Split $\mathcal{P}_i^t$ to N_c^t candidate sets $\mathcal{C}_{i,n}^t$;
  for $n \leftarrow 1$ *to* N_c^t **do**
    Generate the initial state $\boldsymbol{s}_e^t$ with $\mathcal{C}_{i,n}^t$;
    repeat
      Take an action a_e^t by Eq.(10);
      Remove p_{i,n,a_e^t} from $\mathcal{C}_{i,n}^t$;
      Generate the next state $\boldsymbol{s}_{e+1}^t$;
      Compute the reward r_e^t by Eq.(5);
      Insert $(\boldsymbol{s}_e^t, a_e^t, \boldsymbol{s}_{e+1}^t, r_e^t)$ into M^t;
    until *Terminal time* E;
    Obtain the updated candidate set $\hat{\mathcal{C}}_{i,n}^t$;
  end
  Obtain the updated proposal set $\hat{\mathcal{P}}_i^t$;
  Generate the proposal set $\mathcal{P}_i^s$;
  Split $\mathcal{P}_i^s$ to N_c^s candidate sets $\mathcal{C}_{i,n}^s$;
  for $n \leftarrow 1$ *to* N_c^s **do**
    Generate the initial state $\boldsymbol{s}_e^s$ with $\mathcal{C}_{i,n}^s$;
    repeat
      Take an action a_e^s by Eq.(10);
      Remove p_{i,n,a_e^s} from $\mathcal{C}_{i,n}^s$;
      Generate the next state $\boldsymbol{s}_{e+1}^s$;
      Compute the reward r_e^s by Eq.(5);
      Insert $(\boldsymbol{s}_e^s, a_e^s, \boldsymbol{s}_{e+1}^s, r_e^s)$ into M^s;
    until *Terminal time* E;
    Obtain the updated candidate set $\hat{\mathcal{C}}_{i,n}^s$;
  end
  Obtain the updated proposal set $\hat{\mathcal{P}}_i^s$;
end
Compute $\mathcal{L}_{det}$ by Eq.(1) with $\mathcal{P}_i^t$ and $\mathcal{P}_i^s$;
Compute $\mathcal{L}_{adv}$ by Eq.(3) with $\hat{\mathcal{P}}_i^t$ and $\hat{\mathcal{P}}_i^s$;
Compute $\mathcal{L}_q^t$ and $\mathcal{L}_q^s$ by Eq.(7) with sampling experiences from M^t and M^s, respectively;
Update model by Eq.(9);

accumulated rewards since the output of DQN represents both the immediate and future rewards. In this way, the agent learns to make selection at the set level, which is more correct compared with making selection based on the relevance measure function at the instance level. Second, reinforcement learning does not only make use of the learned relevance information but also explores in a wider space to find better solutions since the agent has ϵ probability to take actions of small Q-values for conducting the exploration. Therefore, the agent can accumulate more rich experience and has ability to jump out the local optimum. For example, although some source instances have high relevance to the target domain, the high relevance is caused by similar scenes not by similar objects, and these instances should be removed to avoid the negative transfer. In this case, SIR can search such instances in a wide space and select out them to avoid the negative transfer while the method of selecting based on the learned relevance information cannot select out them due to the high relevance to the target domain.

IV. Experiments

A. Datasets

To evaluate the effectiveness of our method, we conduct experiments using five image datasets as follows:

- The PASCAL VOC dataset [19] is a standardised image dataset for object detection, which has been created and maintained for many years (from 2005-2012). This dataset contains 20 object categories including "aeroplane", "bicycle", "bird", "boat", "bus", "car", "cat", "chair", "cow", "table", "dog", "horse", "motorbike", "person", "plant", "sleep", "sofa", "train", and "tv". PASCAL VOC 2007 has $24,640$ annotated instances in $9,963$ images, and PASCAL VOC 2012 has $27,450$ annotated instances in $11,530$.
- The Clipart dataset [69] is a comical image dataset, which is collected from the CMPlaces dataset [70] and two image search engines (Openclipart[1] and Pixabay[2]). The Clipart dataset consists of $1,000$ images in total with the same 20 object categories as the PASCAL VOC dataset.
- The Watercolor dataset [69] is an artistic dataset, which is from the BAM! [71] dataset and includes six object categories: "bicycle", "bird", "cat", "car", "dog", and "person". The watercolor dataset contains $2,000$ images with $3,315$ annotated instances.
- The SIM 10K dataset [72] is a simulated dataset, which is rendered from the computer game Grand Theft Auto V. Images in this dataset are captured by a dash-cam under car driving scenes. There are $10,000$ synthetic images with $58,701$ annotated car instances.
- The Cityscapes dataset [73] is a benchmark dataset for instance segmentation with pixel-level annotations, which is captured by a dash-cam in urban street scenes. It has $2,975$ images in the training set and 500 images in the validation set, covering eight object categories. Following [21], we use the tightest rectangles of each instance segmentation mask to generate the bounding box annotations.
- The Foggy Cityscapes dataset [74] is a collection of synthetic foggy images, which simulates fog on real scenes and is generated from Cityscapes by adding fog noise.
- The KITTI dataset [75] is a real dataset, which contains $7,481$ images. In this dataset, images have original resolution of 1250×375.

[1] https://openclipart.org/
[2] https://pixabay.com/

TABLE I
FIVE SETTINGS FOR THE CROSS-DOMAIN OBJECT DETECTION TASK

Setting	Training data		Evaluation data
	Source domain	Target domain	Target domain
PASCAL VOC→Clipart (P→C)	Train and validation splits (15,000 images)	All 1,000 images	All 1,000 images
PASCAL VOC→Watercolor (P→W)	Train and validation splits (15,000 images)	Train split (1,000 images)	Test split (1,000 images)
SIM 10K→Cityscapes (S→Ci)	Train split (10,000 images)	Train split (2,975 images)	Validation split (500 images)
Cityscapes→Foggy Cityscapes (Ci→F)	Train split (2,975 images)	Train split (2,975 images)	Validation split (500 images)
KITTI→Cityscapes (K→Ci)	Train split (7,481 images)	Train split (2,975 images)	Validation split (500 images)

Five settings for cross-domain object detection are constructed in our experiments as follows:

- **PASCAL VOC→Clipart (P→C)**: The training and validation splits in the PASCAL VOC 2007 and PASCAL VOC 2012 datasets have totally $15,000$ images that are used as the source domain. The Clipart dataset is used as the target domain. Since the number of source images is much larger than that of target images, all the images in the Clipart dataset are used for training (without labels) and evaluation, following [22], [30].
- **PASCAL VOC→Watercolor (P→W)**: We use the training and validation splits of the PASCAL VOC 2007 and PASCAL VOC 2012 datasets as the source domain and the Watercolor dataset as the target domain, where the six common categories between the source and target domains are used. The training images of the Watercolor dataset ($1,000$ images) are used during training (without labels), and we evaluate our model on the test split of the Watercolor dataset, following [22], [30].
- **SIM 10K→Cityscapes (S→Ci)**: We use the SIM 10K dataset as the source domain and the Cityscapes dataset as the target domain. Both the training images in the source and target domains are used for training and the validation split of the Cityscapes dataset is used for evaluating our model. Since the SIM 10K dataset only has annotations for cars, we evaluate the detection performance on the "car" category following [21], [22].
- **Cityscapes→Foggy Cityscapes (Ci→F)**: We use the Cityscapes dataset as the source domain and the Foggy Cityscapes dataset as the target domain. The training sets of the two datasets are used for training and the validation set of the Foggy Cityscapes dataset is used for evaluation following [76].
- **KITTI→Cityscapes (K→Ci)**: We use the KITTI dataset as the source domain and the Cityscapes dataset as the target domain. The training sets of the two datasets are used for training and the validation set of the Cityscapes dataset is used for evaluation. We report the results on the common "car" category following [21], [76].

Note that our work focuses on the unsupervised cross-domain object detection [21], where only the annotations of source images are provided when training, and the annotations of target images are only used for evaluation. The detailed settings are summarized in TABLE I.

B. Implementation Details

The domain classifier D is constructed by three convolution layers ($512 \rightarrow 128 \rightarrow 1$) with the kernel size of 1. The first two layers are activated by the LeakyReLU function, and the last layer is activated by the Sigmoid function. Both architectures of S-agent and T-agent are built with three full-connected layers ($1024 \rightarrow 512 \rightarrow 16$) by using ReLU as the activation function. For both T-agent and S-agent, the size of each candidate is set to $N_c = 16$, and the number of proposals to be selected out from each candidate set (*i.e.,* the terminal time E of each episode) is set to 3. The discount factor γ in Eq.(8) is set to 0.9 followed by [63], [77], and the probability of exploration ϵ in Eq.(10) is decayed from 0.9 to 0.01 during training. Moreover, we set the threshold τ in the reward function as 0.5. Following Faster R-CNN [8], we resize the shorter side of each image to 600 by preserving its aspect ratio. We train the overall network with a learning rate of 0.001 for the first $50,000$ iterations and reduce the learning rate to 0.0001 for the rest $50,000$ iterations. Each batch consists of one source image and one target image. Following [66], the learning rate ratio of domain classifier to the backbone network of Faster R-CNN is set as $10:1$, *i.e.,* setting the parameter of GRL layer as 0.1.

For the P→C and P→W settings, we adopt the ImageNet pre-trained ResNet101 [78] as the backbone of Faster R-CNN by following [22]. For the S→Ci setting, we adopt both VGG16 [79] and ResNet50 [78] as the backbones of Faster R-CNN. For the Ci→F and K→Ci settings, ResNet50 [78] is used as the backbone of Faster R-CNN by following [76]. For evaluation, both per-category and mean average precisions (mAP) with a threshold of 0.5 are reported for all the settings.

C. Results

We compare our SIR with three existing methods for cross-domain object detection on all the settings with the same backbones: (1) Faster R-CNN [8] is trained on the source domain and directly applied to the target domain without any adaptation. (2) [21] is the first work for cross-domain object detection and reduces the domain shift both on image-level and instance-level. (3) [22] makes strong alignment on the local features of the source and target domains and performs weak alignment on the global features of the two domains. [21] and [22] both utilize Faster R-CNN as the base detection network. We report the results of [21] and [22] from [22]. With ResNet50 as the backbone network, we directly copy the results of [21], [22], [76] from [76].

TABLE II, TABLE III, TABLE IV, TABLE V and TABLE VI show results on the P→C, P→W, S→Ci, Ci→F and K→Ci settings, respectively. From the results, we have the following observations:

TABLE II

AVERAGE PRECISIONS (%) FROM THE PASCAL VOC DATASET TO THE CLIPART DATASET (P→C). WE REPORT THE RESULTS ON THE CLIPART DATASET

Method	aero	bcycl	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	hrs	mcycl	prsn	plnt	sheep	sofa	train	tv	mAP
Faster R-CNN	**35.6**	52.5	24.3	23.0	20.0	43.9	32.8	10.7	30.6	11.7	13.8	6.0	**36.8**	45.9	48.7	41.9	**16.5**	7.3	22.9	32.0	27.8
Chen *et al.* [21]	15.0	34.6	12.4	11.9	19.8	21.1	23.2	3.1	22.1	26.3	10.6	10.0	19.6	39.4	34.6	29.3	1.0	17.1	19.7	24.8	19.8
Saito *et al.* [22]	26.2	48.5	32.6	33.7	38.5	54.3	37.1	**18.6**	34.8	58.3	17.0	12.5	33.8	65.5	61.6	**52.0**	9.3	24.9	54.1	49.1	38.1
SIR	**35.6**	**58.5**	**35.6**	**33.3**	**41.9**	**66.6**	**42.6**	16.1	**37.5**	**59.3**	**27.1**	**21.3**	35.2	**77.0**	**62.3**	48.6	16.1	**30.3**	**54.5**	**52.3**	**42.6**

TABLE III

AVERAGE PRECISIONS (%) FROM THE PASCAL VOC DATASET TO THE WATERCOLOR DATASET (P→W). WE REPORT THE RESULTS ON THE TEST SPLIT OF THE WATERCOLOR DATASET

Method	mcycle	bird	car	cat	dog	prsn	mAP
Faster R-CNN [8]	68.8	46.8	37.2	32.7	21.3	60.7	44.6
Chen *et al.* [21]	75.2	40.6	48.0	31.5	20.6	60.0	46.0
Saito *et al.* [22]	82.3	**55.9**	46.5	32.7	35.5	**66.7**	53.3
SIR	**87.3**	51.8	**47.3**	**37.2**	**43.5**	64.1	**55.2**

TABLE IV

AVERAGE PRECISIONS (%) FROM THE SIM 10K DATASET TO THE CITYSCAPES DATASET (S→CI). WE REPORT THE RESULTS ON THE VALIDATION SPLIT OF THE CITYSCAPES DATASET

Method	AP on car (VGG16)	AP on car (ResNet50)
Faster R-CNN [8]	34.6	39.4
Chen *et al.* [21]	38.97	41.9
Saito *et al.* [22]	40.1	44.6
Xu *et al.* [76]	-	**47.6**
SIR w/o S&T-agents	-	43.6
SIR	**40.3**	46.0

- SIR outperforms all the compared methods on the five settings, clearly demonstrating the effectiveness of sequential instance refinement for cross-domain object detection. In particular, SIR substantially promotes the performance on difficult settings, *e.g.,* improving Faster R-CNN with a gain of 14.8% on the P→C setting, where the domain shift is serious due to the large variances in object styles, backgrounds, and viewpoints.
- SIR performs much better than [21] and [22] on both P→C and P→W settings. This is probably because our SIR can successfully refine both source and target instances by removing outlier target instances and low-relevance source instances, relieving the negative transfer and thereby enhancing the detection performance.
- The improvement of SIR on the S→Ci setting is lower than that on other settings. The possible reason is that object categories in the SIM 10K dataset are similar to the Cityscapes dataset and there is a small variance in the appearance of cars in different domains. Hence, there are few outlier target instances and low-relevance source instances. In this situation, our SIR can still outperform the state-of-the-art methods, which shows the stability of our SIR in different situations. Moreover, with ResNet-50 as the backbone, it is noteworthy that SIR achieves comparable results compared with the current state-of-the-art method [76] and outperforms "SIR w/o S&T-agents", clearly showing the effectiveness of reinforcement learning in handling the negative transfer.
- From the results shown in TABLE V, SIR outperforms "SIR w/o S&T-agents" and the compared method on the Ci→F setting, where the domain shift is caused by different weather conditions. The improved performance demonstrates that SIR relieves the negative transfer and better adapts the model from the normal weather to the foggy weather. Moreover, in this setting, the negative transfer is more serious than the K→Ci and S→Ci settings since there are more low-relevance source instances. In this case, SIR selects more low-relevance source instances and performs better on handling the serious negative transfer.
- On the K→Ci setting, both the source and target domains are real datasets and the images in the two domains are captured by different cameras. From the results shown in TABLE VI, we can find that SIR outperforms "SIR w/o S&T-agents" with a gain of 3%, clearly showing that SIR alleviates the negative transfer when performing adaptation between different real datasets. [76] performs adaptation much better than the adaptation module in SIR ("SIR w/o S&T-agents") probably due to that [76] aligns the conditional distributions between domains and constructs the relation graph between region proposals while "SIR w/o S&T-agents" only matches the marginal distributions between domains via the adversarial training. Thanks to the ability of reinforcement learning in handling the negative transfer, SIR also achieves comparable results based on the simply adaptation module compared with [76].

D. Ablation Study

To analyze our method in depth, ablation study is conducted for empirically evaluating the importance of each individual component. We compare SIR with five variants summarized in TABLE VII: without S-agent (denoted as "SIR w/o S-agent"), without T-agent (denoted as "SIR w/o T-agent"), without S-agent and T-agent (denoted as "SIR w/o S&T-agents"), replacing the patch-based domain classifier with a standard domain classifier (denoted as "SIR-stanD"), and selecting instances directly based on the relevance measure function as defined in Eq. (4) (denoted as "SIR-relevance"). In "SIR w/o S&T-agents", all the source and target instances are utilized for the adversarial training between the backbone network of Faster R-CNN and the domain classifier by minimizing $\mathcal{L}_{adv}$ defined in Eq. (3). The results of ablation study on the P→C setting are shown in TABLE VIII.

1) Effect of S-Agent and T-Agent: From the results shown in TABLE VIII, "SIR w/o S-agent" and "SIR w/o T-agent" work worse than SIR with a drop of 2% and 0.8% in terms of mAP, respectively, which validates that both S-agent and T-agent can contribute to instance refinement for improving the cross-domain object detection performance.

TABLE V

AVERAGE PRECISIONS (%) FROM THE CITYSCAPES DATASET TO THE FOGGY CITYSCAPES DATASET (CI→F). WE REPORT THE RESULTS ON THE VALIDATION SPLIT OF THE FOGGY CITYSCAPES DATASET

Method	person	rider	car	truck	bus	train	motorcycle	bicycle	mAP
Faster-RCNN [8]	26.9	38.2	35.6	18.3	32.4	9.6	25.8	28.6	26.9
Chen *et al.* [21]	29.2	40.4	43.4	19.7	38.3	28.5	23.7	32.7	32.0
Saito *et al.* [22]	31.8	44.3	48.9	21.0	43.8	28.0	28.9	35.8	35.3
Xu *et al.* [76]	32.9	**46.7**	**54.1**	24.7	45.7	**41.1**	32.4	**38.7**	39.5
SIR w/o S&T-agents	25.5	44.1	45.0	25.1	47.1	38.6	24.8	33.9	35.5
SIR	**33.0**	45.6	52.3	**30.5**	**50.8**	40.9	**32.6**	35.9	**40.2**

TABLE VI

AVERAGE PRECISIONS (%) FROM THE KITTI DATASET TO THE CITYSCAPES DATASET (K→CI). WE REPORT THE RESULTS ON THE VALIDATION SPLIT OF THE CITYSCAPES DATASET

Method	AP on car
Faster R-CNN [8]	37.6
Chen *et al.* [21]	41.8
Saito *et al.* [22]	43.2
Xu *et al.* [76]	**47.9**
SIR w/o S&T-agents	43.5
SIR	46.5

"SIR w/o S-agent" and "SIR w/o T-agent" work better than "SIR w/o S&T-agents" by gains of 1.9% and 3.1% in terms of mAP, respectively, which shows that both source and target instances need refinement to alleviate the negative transfer.

2) Effect of the Patch-Based Domain Classifier: To evaluate the patch-based domain classifier, we compare SIR with SIR-stanD in TABLE VIII. SIR outperforms SIR-stanD with a gain of 7.1%, possibly due to the fact that the patch-based domain classifier provides a robust relevance measure by averaging the classification scores of all the pixels in the region proposal. Moreover, from the results in TABLE VIII and TABLE II, "SIR w/o S&T-agents" achieves 10.9% improvement over the source only method ("Faster R-CNN" in TABLE II), clearly demonstrating the effectiveness of perfoming adaptation between different domains. Moreover, "SIR w/o S&T-agents" outperforms [21], [22], showing that it is beneficial to perform fine-grained alignment at the instance-level via the patch-based domain classifier.

3) Effect of Reinforcement Learning: To evaluate the effect of reinforcement learning, we compare SIR with "SIR w/o S&T-agents" and SIR-relevance. The difference between "SIR w/o S&T-agents" and SIR is whether handling the negative transfer by selecting out the outlier target instances and the low-relevance source instances. From the results in TABLE VIII, SIR outperforms "SIR w/o S&T-agents" with an improvement of 3.9%, which demonstrates the benefit of allievating the negative transfer by selecting out the outlier target instances and the low-relevance source instances. As shown in TABLE VIII, SIR outperforms SIR-relevance by 2.8% since the instance selection in SIR is optimized based on a sequential decision procedure, where both the immediate and future rewards are considered. That is to say, the decisions of SIR are made according to the accumulated rewards, while the decisions of SIR-relevance only consider the immediate rewards. For example, in the leftmost column of Fig. 3, SIR-relevance wrongly selects out the source instance (in red box) as low-relevance source instance according to the low immediate reward, ignoring that this instance contains a horse similar to the target horses. In contrast, SIR does not select out this source instance by taking into account future rewards.

E. Statistical and Divergence Analysis

To illustrate the importance of sloving the negative transfer in cross-domain object detection task, we conduct statistics on outlier target instances and low-relevance source instances on all settings to quantify the severity of negative transfer. The statistical results are shown in TABLE IX. From the results, it is noteworthy that outlier target instances and low-relevance source instances exist in each experiment, clearly confirming the significance of addressing the negative transfer. For the P→C and P→W settings, there are many outlier target instances due to the large domain gap between the target domain and the source domain. For the Ci→F setting, when adding foggy noise on images, the appearances of objects are changed, leading to more low-relevance source instances.

To further evaluate the effectiveness of SIR in handling negative transfer, we make the statistical analysis of the number of selected outlier target instances and selected low-relevance source instances on the P→C settting. The statistic results are shown in TABLE X. From the results, it is noteworthy that SIR can select out more outlier target instances and more low-relevance source instances than SIR-relevance, which shows the effectiveness of reinforcement learning in instance refinement. Moreover, SIR achieves better selection results than SIR-relevance, especially in some categories, such as "bird", "bottle", "cow", "horse" and so on. The reason is as that objects of those categories with different viewpoints have large variances in appearance representation and the immediate rewards may not be sufficient to take actions right. In those cases, SIR works better by considering accumulated rewards than SIR-relevance that independently considers the relevance of instance. For some other categories, *e.g.,* "pottedplant (plnt)" and "train", the detection precisons are high in TABLE II, showing that the variance in appearances is relatively small. So the immediate reward may be sufficient to take actions right, and SIR-relevance works well.

Moreover, we also conduct the divergence analysis on the P→C settting. Specifically, we compute the Jensen-Shannon divergence (JS divergence) between the source and target domains, including the JS divergence between all source and target instances (denoted as JS_{all}), the JS divergence between the refined source instances and all the target instances

(a) Low-relevance source instances selected by SIR-relevance

(b) Low-relevance source instances selected by S-agent in SIR

Fig. 3. Examples of selection results on the source domain of the P→C setting. Bounding boxes denote the instances that are selected to be removed from the source domain. Yellow boxes indicate correctly selected source instances, and red boxes indicate wrongly selected source instances. Low-relevance source instances selected by SIR-relevance and S-agent in SIR are shown in (a) and (b), respectively.

(a) Outlier target instances selected by SIR-relevance

(b) Outlier target instances selected by T-agent in SIR

Fig. 4. Examples of selection results on the target domain of the P→C setting. Bounding boxes denote the instances that are selected to be removed from the target domain. Yellow boxes indicate correctly selected target instances, and red boxes indicate wrongly selected target instances. Outlier target instances selected by SIR-relevance and T-agent in SIR are shown in (a) and (b), respectively.

TABLE VII

FIVE VARIANTS OF SIR

SIR w/o S-agent	Without selecting low-relevance source instances
SIR w/o T-agent	Without selecting outlier target instances
SIR w/o S&T-agents	Without making selection and directly performing adaptation
SIR-stanD	Replacing the patch-based domain classifier with a standard domain classifier
SIR-relevance	Making selections according to the relevance measure function defined in Eq.(4)

TABLE VIII

ABLATION STUDY OF SIR FROM THE PASCAL VOC DATASET TO THE CITYSCAPES DATASET (P→C)

Method	mAP (%)
SIR w/o S-agent	40.6
SIR w/o T-agent	41.8
SIR w/o S&T-agents	38.7
SIR-stanD	35.5
SIR-relevance	39.8
SIR	**42.6**

TABLE IX

THE NUMBER OF OUTLIER TARGET INSTANCES AND LOW-RELEVANCE SOURCE INSTANCES IN DIFFERENT SETTINGS. "TOTAL" DENOTES THE NUMBER OF INSTANCES, AND "PER IMAGE" DENOTES THE AVERAGE NUMBER OF INSTANCES IN EACH IMAGE

Setting	#Outlier		#Low-relevance	
	total	per Image	total	per Image
P→C	2,009	2.01	45,143	3.01
P→W	3,609	3.61	26,261	1.75
S→Ci	1,364	0.46	15,514	1.55
Ci→F	313	0.11	17,517	5.89
K→Ci	3,244	1.09	1,806	0.24

(denoted as JS^s_{refine}), the JS divergence between all the source instances and the refined target instances (denoted as JS^t_{refine}), and the JS divergence between the refined source and target instances (denoted as JS_{refine}). The results are shown in TABLE XI. From the results, we can have the following observations. First, $JS^s_{refine} < JS_{all}$ inidicates that the refined source instances are closer to the target domain, demonstrating that s-agent can select out low-relevance source instances to avoid the negative transfer. Second, $JS^t_{refine} < JS_{all}$ shows that after the selection conducted by t-agent, the outlier target instances are removed from the target domain, demonstrating the effectiveness of the t-agent. Third, $JS_{refine} < JS_{all}$, which shows that the refined source and target instances are closer to each other and the instance refinements in both the source and target domains failicate the positive transfer.

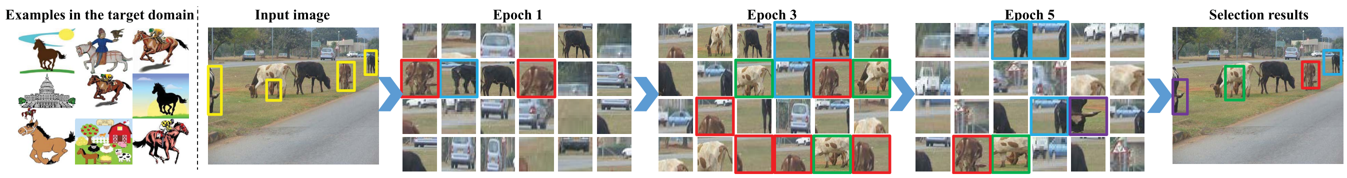
F. Qualitative Evaluation

Fig. 3 and Fig. 4 show the selection comparisons between SIR and SIR-relevance on the $P \rightarrow C$ setting. As shown

TABLE X

THE STATISTICS OF THE SELECTED OUTLIER TARGET INSTANCES AND LOW-RELEVANCE SOURCE INSTANCES ON THE P → C SETTING. "#OUTLIER" AND "#LOW-RELEVANCE" DENOTE THE NUMBER OF THE SELECTED OUTLIER TARGET INSTANCES AND THE NUMBER OF THE SELECTED LOW-RELEVANCE SOURCE INSTANCES, RESPECTIVELY

Method	#Outlier	#Low-relevance																			
		aero	bcycl	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	hrs	mcycl	prsn	plnt	sheep	sofa	train	tv
SIR-relevance	1542	1014	1057	2304	2016	876	676	3625	**2117**	**3799**	1665	1256	2836	1213	**736**	**8596**	**1290**	1784	1462	**1236**	**636**
SIR	**1724**	**1220**	**1104**	**2731**	**2045**	**1012**	**756**	**3799**	2072	3788	**2048**	**1324**	**2916**	**1311**	708	8432	1099	**2018**	**2089**	1129	604



(a)

(b)

Fig. 5. Two example of the selection processes of S-agent in SIR for the "horse" category on the P→C setting. The leftmost column represents several horses in the target domain (the Clipart dataset) for comparison. In the second column, the low-relevance source instances of the input image are marked with yellow boxes, which are less relevant to the target domain and should be removed. In the following three columns, we present 24 proposals selected out by S-agent at epoch 1, 3 and 5, and mark all proposals of the same object with the same color boxes. The rightmost column represents the final selection results of the input image at epoch 10.

TABLE XI

THE JENSEN-SHANNON DIVERGENCE BETWEEN DOMAINS ON THE P → C SETTING

JS_{all}	JS^s_{refine}	JS^t_{refine}	JS_{refine}
0.2635	0.2628	0.2619	0.2618

in Fig. 3(b), we find that S-agent removes more low-relevance source instances than SIR-relevance. In addition, some instances containing similar objects to the target ones are wrongly selected by SIR-relevance such as the horse in the rightmost column of Fig. 3(a). The reason is probably that the person on the horse has a different pose to the target ones and the immediate reward is small. In contrast, SIR does not select out this horse by considering accumulated rewards, which validates that the agent can make more correct decisions than independently considering the relevance of instance.

The selection results of T-agent are shown in Fig. 4. We can find that T-agent correctly removes more outlier target instances from the target domain than SIR-relevance, such as fox in the rightmost column of Fig. 4(b). Since the appearance of fox is similar to that of cat in the source domain, resulting in the low immediate reward, SIR-relevance does not filter out this fox. In contrast, SIR successfully selects out the fox with the exploration ability of reinforcement learning.

To go deeper with the effectiveness of sequential instance refinement, we visualize the selection processes of S-agent and T-agent on the P→C setting. In Fig. 5, we show two examples of the selection processes of S-agent for the "horse" category. Since the goal of S-agent is selecting out low-relevance source instances, we present some object examples (horses) from the target domain (the Clipart dataset) in the leftmost column for comparison. The second column shows the input image, where the low-relevance source instances are marked with yellow boxes. The following three columns show 24 proposals that are selected from 128 proposals of the input image by S-agent at epoch 1, 3 and 5. We use the same color box to denote the same object. The rightmost column demonstrates the selection results at epoch 10. From the results, it is worth noting that S-agent selects out increasing low-relevance source instances with the increasing epoch. Specifically, in Fig. 5(a), at epoch 1, S-agent selects out two horses with different viewpoints to the target ones. At epoch 3 and 5, with the accumulated experience, S-agent selects out more horses that are heavily occluded.

In Fig. 6, we visualize two examples of the selection processes of T-agent at epoch 1, 3 and 5 in the target domain (the Clipart dataset). The leftmost column shows the object categories of the source domain (the PASCAL VOC dataset). The second column shows the input image, where instances with yellow boxes are outlier target instances. The following three columns show the selected proposals by T-agent at epoch 1, 3 and 5, where we denote all the proposals of the same object with the same color boxes. For example, all the proposals of the bigger dolphin in the input image of Fig. 6(b) are marked with red boxes. From the results, T-agent progressively selects out all the outlier target instances with the

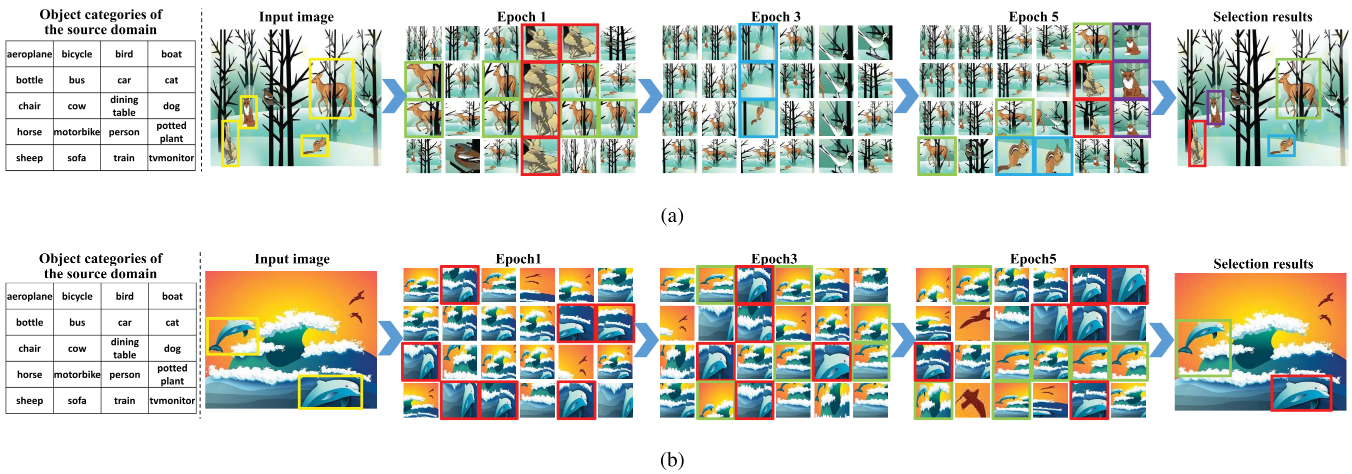


Fig. 6. Two example of the selection processes of T-agent in SIR on the P→C setting. (a) The selection process of T-agent on an image of multiple animals. (b) The selection process of T-agent on an image of a dolphin. The leftmost column shows the object categories of the source domain (the PASCAL VOC dataset). In the second column, the outlier target instances of the input image are marked with yellow boxes, which does not belong to any category of the source domain and should be removed. In the following three columns, we present 24 proposals selected out by T-agent at epoch 1, 3 and 5, and mark all proposals of the same object with the same color boxes. The rightmost column shows the final selection results of the input image at epoch 10.

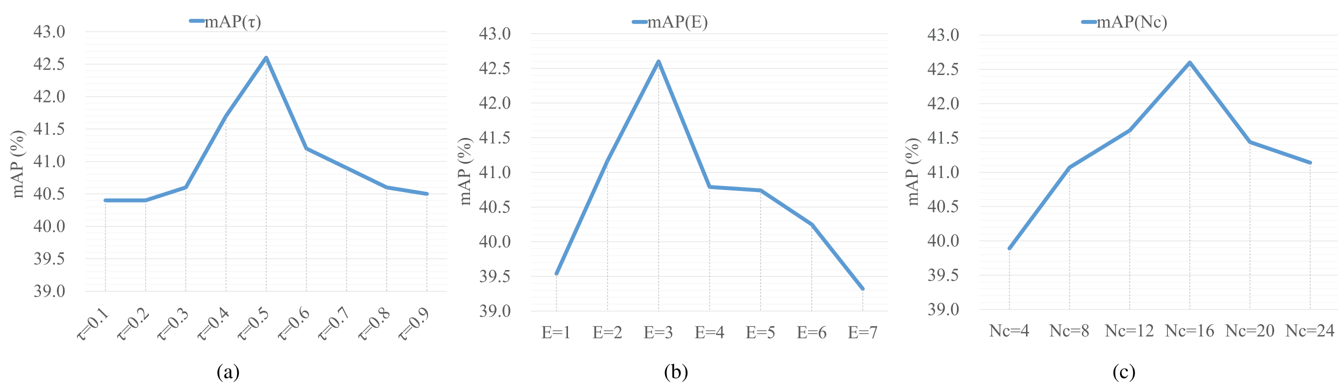


Fig. 7. Hyperparameter analysis on the P→C setting. (a) Results of different threshold τ. (b) Results of different terminal time E (the number of selected instances). (c) Results of different size N_c of the candidate set.

increasing epoch. Concretely, as shown in Fig. 6(a), at epoch 1, two deers are selected out by T-agent. Since the fox is more similar to the dog in the source domain and the squirrel is more similar to the cat in the source domain, instances of fox and squirrel are not selected out by T-agent at first. It is interesting to observe that at epoch 5, instances containing the fox and the squirrel are selected out by T-agent owing to better Q-value estimated by DQN. In Fig. 6(b), T-agent only selects out the bigger dolphin at epoch 1. With the epoch increases, T-agent selects out the smaller dolphin at epoch 3 and selects out more proposals of the two dolphins at epoch 5, which clearly demonstrates the effectiveness of T-agent in refining target proposals.

G. Hyperparameter Analysis

In this section, we conduct experiments on the P→C setting to evaluate the effect of the threshold τ, the terminal time E, the size N_c of the candidate set, and the probability of exploration ϵ. The results of different τ, E and N_c are shown in Fig. 7, and the results of different ϵ are shown in TABLE XII. From the results, it is noteworthy that the

TABLE XII

AVERAGE PRECISIONS (%) ON THE P → C SETTING WITH DIFFERENT VALUES OF ϵ

Method	Initial value	1			0.9		
	Final value	0	0.01	0.1	0	0.01	0.1
SIR		41.9	40.1	40.7	40.5	**42.6**	40.0

performances of SIR with different hyperparameters are all over 39.0% and are better than "SIR w/o S&T-agents", clearly demonstrating the effectiveness of sequential instance refinement for cross-domain object detection. The followings are the detailed hyperparameter analysis.

1) Discussion of the Threshold τ: We select τ in the range of $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ and show the mAP-threshold curves in Fig. 7 (a), where the horizontal axis represents the value of τ in Eq. (5) and Eq. (6), and the vertical axis represents the mAP. From the results, we can find that the mAP first increases and then decreases as the threshold τ increases. Specifically, when the threshold τ is small, few outlier target instances and low-relevance source instances are selected out, and some outlier target instances and low-relevance source instances still remain in the candidate set. In this case, the negative transfer is mildly relieved and the mAP is low. When τ becomes large, some target instances in the shared classes are wrongly selected out, leading to the decreasing mAP. Based on the experimental results, we set $\tau = 0.5$ to achieve the best performance.

2) Discussion of the Terminal Time E: We tune E in the range of $\{1, 2, 3, 4, 5, 6, 7\}$ and show the mAP-E curves in Fig. 7 (b), where the horizontal axis represents the value of E, and the vertical axis represents the mAP. From the results, the highest mAP on the target domain is achieved when E is set to 3, which means that selecting 3 instances from each candidate set most properly relieves the negative transfer. When E is large, the agents continue selection although all the outlier target instances and low-relevance soruce instances have been selected out, leading to wrong selections of instances. When E is small, the agents stop selection before all the outlier target instances and low-relevance source instances are selected out, resulting in mildly relieving the negative transfer.

3) Discussion of the Size N_c of the Candidate Set: Fig. 7 (c) shows the perfromance of our method trained with different values of N_c. From the results, the mAP first increases, achieves the highest value when $N_c = 16$, and then decreases. The possible reason is that a large N_c leads to fewer candidate sets and fewer instances are selected out from the proposal set. A small N_c means more candidate sets and more instances are selected out from the proposal set, which are prone to wrong selections of instances. We set $N_c = 16$ for the best performance in our experiments.

4) Discussion of the Probability of Exploration ϵ: TABLE XII shows the performance of our method trained with different decay schemes of ϵ, where ϵ is decayed from the initial value to the final value during training. The larger ϵ is, the more the agent explores. The larger ϵ is, the more the agent explores. For convenience, we denote the intial ϵ as ϵ_s and the final ϵ as ϵ_f. When $\epsilon_s = 1$, the agent explores more in the initial stage and requires more exploitation in the final stage, so $\epsilon_f = 0$ achieves better result than $\epsilon_f = 0.01$ and $\epsilon_f = 0.1$. When $\epsilon_s = 0.9$, the agent explores less in the initial stage and requires more exploration in the final stage. So $\epsilon_f = 0.01$ achieves better result than $\epsilon_f = 0$. In other words, the best performance is achieved when ϵ is decayed from 0.9 to 0.01, which shows that under this setting, the agent can better balance the exploration and the exploitation during learning the policies of instance refinement.

V. Conclusion

We have presented a reinforcement learning based method, namely sequential instance refinement (SIR), to address the negative transfer problem in cross-domain object detection. In our SIR, S-agent and T-agent learn to remove the low-relevance source instances and outlier target instances, respectively. Via the sequential actions in the reinforcement learning process, the two agents can progressively refine both source and target instances and thus successfully alleviate negative transfer. Extensive experiments conducted on several benchmark datasets clearly demonstrate that our SIR outperforms the existing state-of-the-art methods for the cross-domain object detection task. As we believe that SIR is a general solution for tackling the negative transfer problem in object detection and can be readily incorporated by existing cross-domain methods to improve the overall performance.

References

[1] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Trans. Comput.*, vol. C-22, no. 1, pp. 67–92, Jan. 1973.

[2] B. Li, W. Ouyang, L. Sheng, X. Zeng, and X. Wang, "GS3D: An efficient 3D object detection framework for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1019–1028.

[3] P. Li, X. Chen, and S. Shen, "Stereo R-CNN based 3D object detection for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7644–7652.

[4] J. Tao, M. Turjo, M.-F. Wong, M. Wang, and Y.-P. Tan, "Fall incidents detection for intelligent video surveillance," in *Proc. 5th Int. Conf. Inf. Commun. Signal Process.*, 2005, pp. 1590–1594.

[5] M. Ahmadi, W. Ouarda, and A. M. Alimi, "Efficient and fast objects detection technique for intelligent video surveillance using transfer learning and fine-tuning," *Arabian J. Sci. Eng.*, vol. 45, no. 1, pp. 1421–1433, 2019.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[7] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.

[10] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 21–37.

[11] B. Singh, H. Li, A. Sharma, and L. S. Davis, "R-FCN-3000 at 30fps: Decoupling detection and classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1081–1090.

[12] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

[13] Y. Li, Y. Chen, N. Wang, and Z.-X. Zhang, "Scale-aware trident networks for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6053–6062.

[14] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6569–6578.

[15] F. Sun, T. Kong, W. Huang, C. Tan, B. Fang, and H. Liu, "Feature pyramid reconfiguration with consistent loss for object detection," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 5041–5051, May 2019.

[16] S. Jiang, S. Liang, C. Chen, Y. Zhu, and X. Li, "Class agnostic image common object detection," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2836–2846, Jun. 2019.

[17] Y. Zhu, C. Zhao, H. Guo, J. Wang, X. Zhao, and H. Lu, "Attention CoupleNet: Fully convolutional attention coupling network for object detection," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 113–126, Jan. 2019.

[18] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2014, pp. 740–755.

[19] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[20] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 999–1006.

[21] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster R-CNN for object detection in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3339–3348.

[22] K. Saito, Y. Ushiku, T. Harada, and K. Saenko, "Strong-weak distribution alignment for adaptive object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6956–6965.

[23] Q. Cai, Y. Pan, C.-W. Ngo, X. Tian, L. Duan, and T. Yao, "Exploring object relation in mean teacher for cross-domain detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11457–11466.

[24] X. Zhu, J. Pang, C. Yang, J. Shi, and D. Lin, "Adapting object detectors via selective cross-domain alignment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 687–696.

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

[26] H.-K. Hsu *et al.*, "Progressive domain adaptation for object detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 749–757.

[27] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.

[28] A. RoyChowdhury *et al.*, "Automatic adaptation of object detectors to new domains using self-training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 780–790.

[29] M. Khodabandeh, A. Vahdat, M. Ranjbar, and W. Macready, "A robust learning approach to domain adaptive object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 480–490.

[30] S. Kim, J. Choi, T. Kim, and C. Kim, "Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6092–6101.

[31] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Scholkopf, "Correcting sample selection bias by unlabeled data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 601–608.

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

[33] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2066–2073.

[34] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2200–2207.

[35] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and B. Schölkopf, "Domain adaptation with conditional transferable components," in *Proc. Int. Conf. Multimodal Interact. (ICML)*, 2016, pp. 2839–2848.

[36] J. Wang, Y. Chen, S. Hao, W. Feng, and Z. Shen, "Balanced distribution adaptation for transfer learning," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Nov. 2017, pp. 1129–1134.

[37] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1859–1867.

[38] Z. Xu, W. Li, L. Niu, and D. Xu, "Exploiting low-rank structure from latent domains for domain generalization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 628–643.

[39] L. Niu, W. Li, D. Xu, and J. Cai, "An exemplar-based multi-view domain generalization framework for visual recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 2, pp. 259–272, Aug. 2018.

[40] A. Rozantsev, M. Salzmann, and P. Fua, "Residual parameter transfer for deep domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4339–4348.

[41] W. Li, Z. Xu, D. Xu, D. Dai, and L. Van Gool, "Domain generalization and adaptation using low rank exemplar SVMs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1114–1127, May 2018.

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

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

[44] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 2208–2217.

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

[46] F. M. Carlucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulo, "AutoDIAL: Automatic domain alignment layers," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5067–5075.

[47] L. F. Alvarenga e Silva and J. Almeida, "MS-DIAL: Multi-source domain alignment layers for unsupervised domain adaptation," in *Proc. Anais Workshop Vis. Comput. (WVC)*, Oct. 2020, pp. 357–369.

[48] M. Mancini, L. Porzi, S. R. Bulo, B. Caputo, and E. Ricci, "Boosting domain adaptation by discovering latent domains," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3771–3780.

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

[50] W. Zhang, W. Ouyang, W. Li, and D. Xu, "Collaborative and adversarial network for unsupervised domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3801–3809.

[51] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1640–1650.

[52] Y. Zhang, H. Tang, K. Jia, and M. Tan, "Domain-symmetric networks for adversarial domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5031–5040.

[53] A. Chadha and Y. Andreopoulos, "Improved techniques for adversarial discriminative domain adaptation," *IEEE Trans. Image Process.*, vol. 29, pp. 2622–2637, 2020.

[54] F. Yu, X. Wu, J. Chen, and L. Duan, "Exploiting images for video recognition: Heterogeneous feature augmentation via symmetric adversarial learning," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5308–5321, Nov. 2019.

[55] J. Chen, X. Wu, L. Duan, and S. Gao, "Domain adversarial reinforcement learning for partial domain adaptation," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Oct. 16, 2020, doi: 10.1109/TNNLS.2020.3028078.

[56] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[57] L. Ren, J. Lu, Z. Wang, Q. Tian, and J. Zhou, "Collaborative deep reinforcement learning for multi-object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 586–602.

[58] X. Dong, J. Shen, W. Wang, Y. Liu, L. Shao, and F. Porikli, "Hyperparameter optimization for tracking with continuous deep Q-learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 518–527.

[59] X. Wang, W. Chen, J. Wu, Y.-F. Wang, and W. Y. Wang, "Video captioning via hierarchical reinforcement learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4213–4222.

[60] N. Li, Z. Chen, and S. Liu, "Meta learning for image captioning," in *Proc. Assoc. Adv. Artif. Intell. (AAAI)*, 2019, pp. 8626–8633.

[61] S. Mathe, A. Pirinen, and C. Sminchisescu, "Reinforcement learning for visual object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2894–2902.

[62] A. Pirinen and C. Sminchisescu, "Deep reinforcement learning of region proposal networks for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6945–6954.

[63] Z. Jie, X. Liang, J. Feng, X. Jin, W. Lu, and S. Yan, "Tree-structured reinforcement learning for sequential object localization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 127–135.

[64] J. C. Caicedo and S. Lazebnik, "Active object localization with deep reinforcement learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2488–2496.

[65] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2794–2802.

[66] Y. Ganin and V. S. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2015, pp. 1180–1189.

[67] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[68] L.-J. Lin, "Self-improving reactive agents based on reinforcement learning, planning and teaching," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 293–321, May 1992.

[69] N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa, "Cross-domain weakly-supervised object detection through progressive domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5001–5009.

[70] L. Castrejon, Y. Aytar, C. Vondrick, H. Pirsiavash, and A. Torralba, "Learning aligned cross-modal representations from weakly aligned data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2940–2949.

[71] M. J. Wilber, C. Fang, H. Jin, A. Hertzmann, J. Collomosse, and S. Belongie, "Bam! The behance artistic media dataset for recognition beyond photography," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1202–1211.

[72] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan, "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?" in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 1–8.

[73] M. Cordts *et al.*, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

[74] C. Sakaridis, D. Dai, and L. Van Gool, "Semantic foggy scene understanding with synthetic data," *Int. J. Comput. Vis.*, vol. 126, no. 9, pp. 973–992, 2018.

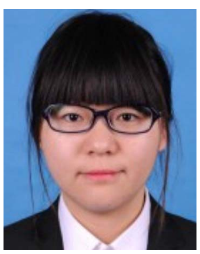
[75] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[76] M. Xu, H. Wang, B. Ni, Q. Tian, and W. Zhang, "Cross-domain detection via graph-induced prototype alignment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, p. 12.

[77] K. M. Lee, H. Myeong, and G. Song, "SeedNet: Automatic seed generation with deep reinforcement learning for robust interactive segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1760–1768.

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

[79] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556



Jin Chen received the B.S. degree in computer science from the Beijing Institute of Technology, Beijing, China, in 2017, where she is currently pursuing the Ph.D. degree in computer science with the Beijing Laboratory of Intelligent Information Technology.

Her current research interests include domain adaptation, reinforcement learning, and machine learning.

Xinxiao Wu (Member, IEEE) received the B.S. degree in computer science from the Nanjing University of Information Science and Technology, Nanjing, China, in 2005, and the Ph.D. degree in computer science from the Beijing Institute of Technology (BIT), Beijing, China, in 2010. From 2010 to 2011, she was a Postdoctoral Research Fellow with Nanyang Technological University, Singapore. She is currently an Associate Professor with the School of Computer Science, BIT. Her research interests include machine learning, computer vision, and video analysis and understanding.

Lixin Duan received the B.Eng. degree from the University of Science and Technology of China in 2008 and the Ph.D. degree from the Nanyang Technological University in 2012.

He is currently a Full Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC). His research interests include machine learning algorithms especially in transfer learning and domain adaptation, and their applications in computer vision. He was a recipient of the Microsoft Research Asia Fellowship in 2009 and the Best Student Paper Award at the IEEE Conference on Computer Vision and Pattern Recognition 2010.

Lin Chen received the B.E. degree from the University of Science and Technology of China in 2009 and the Ph.D. degree from Nanyang Technological University, Singapore, in 2014.

He is currently a Senior Principal Scientist and the Head of AI with Wyze Labs, Kirkland, WA, USA. His current research interests include machine learning, such as transfer learning, domain adaptation, weakly-supervised learning, few-shot-learning, and their applications in computer vision, such as object detection, face recognition, and video understanding. He was a recipient of IEEE TRANSACTIONS ON MULTIMEDIA Prize Paper Award in 2014.