RPN Prototype Alignment For Domain Adaptive Object Detector

Yixin Zhang Zilei Wang* Yushi Mao
Department of Automation, University of Science and Technology of China
zhyx12@mail.ustc.edu.cn zlwang@ustc.edu.cn mys@mail.ustc.edu.cn

Abstract

Recent years have witnessed great progress of object detection. However, due to the domain shift problem, applying the knowledge of an object detector learned from one specific domain to another one often suffers severe performance degradation. Most existing methods adopt feature alignment either on the backbone network or instance classifier to increase the transferability of object detector. Differently, we propose to perform feature alignment in the RPN stage such that the foreground and background RPN proposals in target domain can be effectively distinguished. Specifically, we first construct one set of learnable RPN prototpyes, and then enforce the RPN features to align with the prototypes for both source and target domains. It essentially cooperates the learning of RPN prototypes and features to align the source and target RPN features. Particularly, we propose a simple yet effective method suitable for RPN feature alignment to generate high-quality pseudo label of proposals in target domain, i.e., using the filtered detection results with IoU. Furthermore, we adopt Grad CAM to find the discriminative region within a foreground proposal and use it to increase the discriminability of RPN features for alignment. We conduct extensive experiments on multiple cross-domain detection scenarios, and the results show the effectiveness of our proposed method against previous state-of-the-art methods.

1. Introduction

Object detection is a fundamental task in computer vision, which aims to identify and localize objects of interest in an image. In the past decade, a great progress has been witnessed for object detection due to the advance of large-scale benchmarks and modern CNN-based detection frameworks, such as Fast/Faster R-CNN [15, 41]. Currently, state-of-the-art detectors generally require massive training samples with annotations of bounding boxes and semantic labels. Particularly, the detection model learned from the data in a domain (*i.e.*, source domain) would incur severe performance degradation when facing some new environment (*i.e.*, target domain) where object appearance, background, or weather condition make a difference [49]. At the same time, accurately annotating all new samples usually involve heavy labor and high cost. To address this challenge, unsupervised domain adaptation (UDA) [37] is developed to adapt the model learned from the annotated source samples to the target samples, *i.e.*, enabling the model work well in target domain by incorporating unlabeled samples.

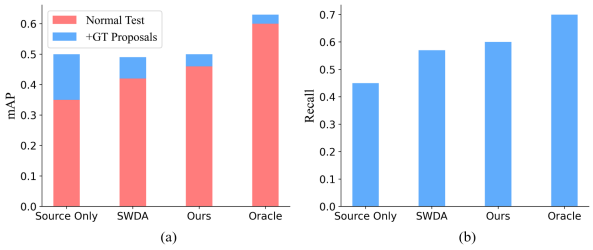

Figure 1. (a) Comparison of detection performance among "Source only", "SWDA" [42], "Ours", and "Oracle", where the normal case and adding ground truth boxes are tested. (b) Comparison of recall among different methods with the IoU threshold of 0.5. Here Sim10k [24]→Cityscapes [7] is adopted as the benchmark. Best viewed in color.

Evidently, UDA need learn the knowledge useful for target domain from the data of source domain. A common practice is to build invariant feature representation across domains, which essentially enforces to align the feature distributions of two domains. To this end, the measure of domain shift is usually minimized, *e.g.*, correlation distances [32, 39, 34, 62]. Recently, domain adversarial learning is widely adopted and achieves promising performance by conducting a min-max game between object detector and domain discriminator [13, 33, 21, 50]. Along previous works [6, 42], we particularly focus on the domain adaptation problem of two-stage object detectors in this paper.

Regarding cross-domain object detection, several works have attempted to incorporate adversarial learning into

*Corresponding author

mainstream detection frameworks, *e.g*., Faster R-CNN. Typically, a two-stage object detector can be split into three main modules, *i.e*., backbone network, region proposal network (RPN), and region proposal classifier (RPC). Due to locality nature of the object detection task, current methods usually minimize the domain disparity at multiple levels via adversarial feature adaptation, such as image and instance alignment [6], strong-local and weak-global alignment [42], and multi-level feature alignment [19, 58]. Most of these works conduct feature alignment in either backbone network or RPC, and hold a common belief that in domain adversarial learning, the foreground regions should be given more attentions to increase the transferability of interested objects and meanwhile alleviate the negative effect of background noises. Different from previous methods, we focus on the transerability of RPN across domains in this work.

Actually, RPN plays an important role in domain adaption of object detectors. To intuitively show its effect, here we particularly conduct an analysis experiment to investigate the upper bound of RPN. To be specific, we add the ground-truth bounding boxes into the RPN proposals in the test phase, and then use them to infer the final detection results as usual. Figure 1(a) give the performance comparison of before and after adding ground truth, where different methods are used, including *source only*, SWDA [42], our method, and *Oracle*. Comparing *source only* and *Oracle*, we can observe that the quality of RPN proposals is especially important for cross-domain object detection. In addition, SWDA can reduce the performance gap although it only performs feature alignment in backbone network. This is because SWDA actually can produce higher quality proposals with larger RPN recall than *source only*, as shown in Figure 1(b). However, only aligning features in backbone network is not enough to generate high-quality RPN proposals since domain adversarial learning on backbone network ignores to distinguish the discriminability of foreground and background.

In object detection, an RPN proposal commonly contains the foreground and background contents, and their ratio varies a lot for different proposals. Such a characteristic makes RPN feature alignment very challenging since the foreground features are inevitably contaminated by various background noises. Moreover, the interested objects usually involve many semantic categories, but they would be unified into foreground in RPN. In RPN feature alignment, therefore, we need carefully balance different object classes in order to make the final detection work well.

In this paper, we propose a novel RPN prototype alignment method to separately align foreground and background RPN features. Specifically, we first construct a set of learnable RPN prototpyes, and then enforce the RPN features in both source and target domains to align with the corresponding prototypes. Through cooperating with RPN prototype learning, the RPN features in source and target domains can be effectively aligned. In this paradigm, the pseudo label of proposals in target domain need be first generated, and we propose a simple yet effective method suitable for RPN feature alignment. To be specific, we first filter the detection results after RPC to only reserve the high-confidence ones, where a class-agnostic filter ratio is adopted to balance different classes. Then we use the filtered detection results to generate the pseudo label of RPN proposals, in which IoU is used to assign proposal labels rather than the predicted scores in previous methods. Furthermore, to increase the discriminability of RPN features for alignment, we propose to use Grad CAM to find the discriminative regions of a proposal and then adjust RPN features by spatially weighting. Consequently, foreground and background RPN features are better aligned, and more accurate proposals can be obtained.

The main contributions of this work are summarized as follows:

- We propose a novel RPN prototype alignment method which can significantly improve the transferability of RPN and further generate high-quality RPN proposals for target domain.
- We propose a simple yet effective pseudo label generation method suitable for feature alignment of RPN proposals, which can effectively guide the learning of RPN prototypes and features.
- We propose a discriminability-aware prototype alignment module, which can improve the quality of RPN features for alignment by paying more attention on discriminative regions within a proposal.
- We conduct extensive experiments on multiple benchmark scenarios, and the results demonstrate the effectiveness of our proposed method against previous state-of-the-art methods.

2. Related Work

Object Detection. Object detection is an essential task of computer vision, which has been studied for many years [29]. Most of traditional methods [54, 8, 11] rely on handcrafted features and sophisticated pipelines. In the era of deep learning, object detection can be roughly categorized into two classes: one-stage detectors [40, 31, 28, 30] and two-stage ones [16, 15, 41, 27]. Although one-stage detectors have high efficiency and have become popular paradigms, two-stage detectors are still widely adopted for pursuing much higher performance. In particular, Faster R-CNN [41] is a classical two-stage object detector and is widely adopted for domain adaptive object detection due to its robustness and scalability. Following previous works, we choose Faster R-CNN as the baseline detector in this paper.

Unsupervised Domain Adaptation. UDA [2, 1] aims to generalize the model learned from labeled source domain to another unlabeled target domain. It has been investigated for different computer vision tasks [57, 10, 65, 35, 5, 12, 63], *e.g*., image classification, semantic segmentation, and object detection. Considering the powerful capacity of deep learning, many solutions attempt to reduce domain shift by learning domain-invariant features. Early domain adaptive models minimized the estimated disparity between different domains, such as maximum mean discrepancy (MMD) [52, 32, 48]. Recently, domain adversarial learning is adopt to improve the performance [13, 51, 3, 38]. In this work, we particularly focus on domain adaptation of object detection.

Cross-domain Object Detection. Traditional studies [55, 53, 36, 60] mainly adapt some specific model (*e.g*., for pedestrian or vehicle detection) across domains, while recently domain adaptive object detection has been raised for unconstrained scenes. Chen *et al*. [6] first propose two alignment practices, *i.e*., image-level and instance-level alignments by imposing adversarial learning at image and instance scales. Following this work, many works manage to reduce the feature discrepancy in backbone network. [19, 58] apply this idea to multi-layer feature adaptation. [42] proposes strong-weak alignment components to incorporate strong matching in local features and weak matching in global features. [69] mines discriminative regions that contain objects of interest and aligns their features. [4] uses the output of domain discriminator to get the discriminative and transferrable regions for local and global alignments. [22, 26] introduce object centerness and spatial attention into cross-domain feature adversarial learning to avoid the influence of background. [59, 66] add an image-level multi-label classifier upon backbone network to align crucial regions and preserve the discriminability of features.

Apart from aligning backbone features, some other works consider to align RPC features. [61, 67] propose to align the prototypes of RPC. To be specific, [61] proposes to first perform a graph-based information propagation to obtain more precise instance-level features, then construct prototypes within a mini-batch for source and target domains, and finally conduct contrastive learning among these prototypes. Although GPA uses RPN prototype alignment, they attach an extra network after backbone to perform backbone feature alignment, which serves as regularization and does not affect the training of RPN. Our method is directly to align the RPN features for producing more accurate proposals. [67] constructs the global prototypes for source and target domains, and then updates the prototypes using mini-batch samples and meanwhile minimizes the distance between the source and target prototypes.

Different from previous methods that align the features in backbone network and RPC, we propose to align RPN features to produce high-quality proposals. A related work is CoT [64] which proposes to perform collaborative training between RPN and RPC. Specifically, the high-confidence outputs are leveraged as mutual guidance to train each other, and as in MCD [43] the low-confidence ones are used for discrepancy calculation between RPN and RPC and minimax optimization. Although CoT considers the RPN, it is essentially different from our proposed method. First, they have different learning modes for RPN. CoT adopts the predicted probabilities of PRC to cooperate the learning of RPN with self-training. Our method adopts the learnable prototypes to align the intermediate RPN features. Second, they adopt different techniques to transfer knowledge from RPC to RPN. CoT uses the online RPC probabilities as soft pseudo labels to weight RPN self-training. We use the filtered detection results of RPC to generate the pseudo label, which is periodically updated, and then used to select the RPN proposals by IoU threshold. In addition, we introduce the Grad CAM from RPC to more accurately extract RPN features for foreground proposals.

3. Method

In this work, we focus on the unsupervised domain adaptation problem in object detection. Formally, we are given a source domain $\mathcal{D}_s = \{(\boldsymbol{x}_i^s, \boldsymbol{y}_i^s)\}_{i=1}^{n_s}$ of n_s labeled samples and a target domain $\mathcal{D}_t = \{\boldsymbol{x}_j^t\}_{j=1}^{n_t}$ of n_t unlabeled samples, which are from the joint distributions $P(\boldsymbol{x}_s, \boldsymbol{y}_s)$ and $Q(\boldsymbol{x}_t, \boldsymbol{y}_t)$ with $P \neq Q$, respectively. Then our work aims to learn an object detector that can reduce the shifts in the joint distribution across domains and further generalize well to the target domain. In what follows, we will first review the widely adopted feature alignment in backbone network, which servers as our baseline model. Then we deeply explore the proposed RPN prototype alignment method, and elaborate on how it can improve the transferability of object detection network.

3.1. Baseline Model

Following previous works [42, 19], we use Faster R-CNN with the VGG16 [46] backbone as our basic object detector. In particular, we take the mainstream backbone feature alignment method as our baseline, which actually conducts domain adversarial learning for middle-layer and high-layer features. Here the adversarial loss acts as a min-max game [17], and the training procedure contains two opposite optimization objectives with the loss function

$$\begin{aligned}\mathcal{L}_{ADV} = \sum_{l} \min_{\theta_{G_l}} \max_{\theta_{D_l}} \mathbb{E}_{x_s \sim \mathcal{D}_S} \log D_l(G_l(x_s)) \\ + \mathbb{E}_{x_t \sim \mathcal{D}_T} \log(1 - D_l(G_l(x_t))),\end{aligned} \tag{1}$$

where $l \in \{3, 4, 5\}$ represents the l-th convolutional block of VGG16, G and D denote the backbone network and domain discriminator, respectively. θ_{G_l} and θ_{D_l} correspond to

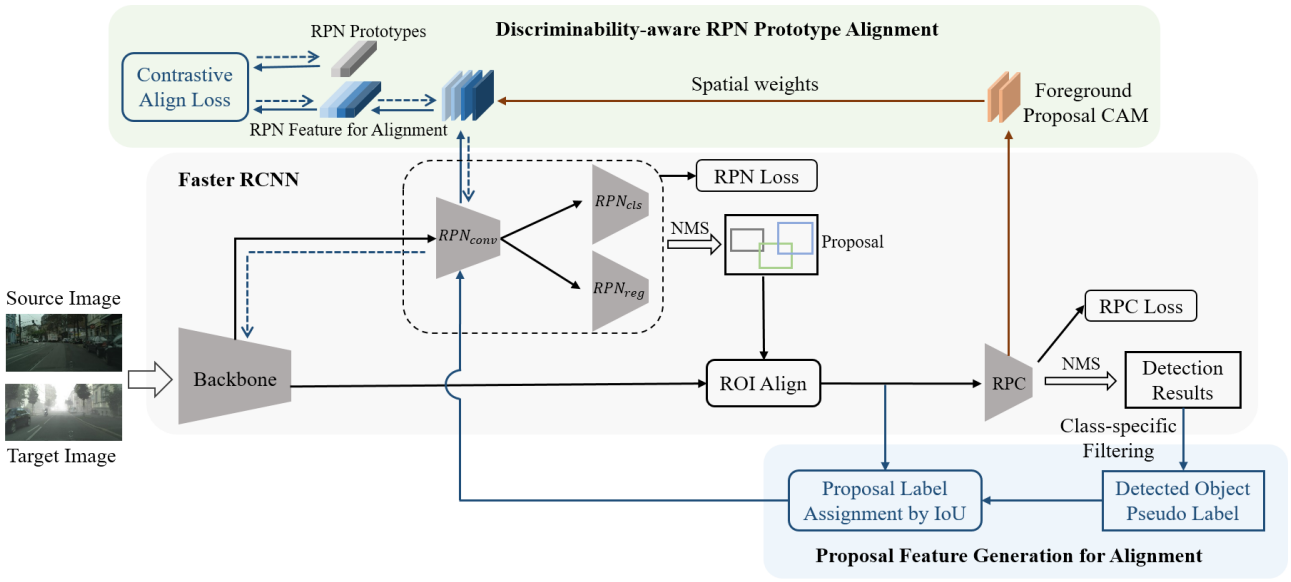

Figure 2. The framework of our proposed method, where Faster R-CNN is adopted as the basic detector. Considering RPN characteristics, we particularly propose pseudo label generation to get proposals for alignment in target domain, and discriminability-aware RPN prototype alignment to improve the feature alignment of foreground proposals. Here the opposing dotted arrow represents the gradient flow of contrastive align loss. Best viewed in color.

the parameters of G_l and D_l. In practice, G and D are connected by the Gradient Reverse Layer (GRL) [13], which reverses the gradients that flow through G.

For this baseline method, the optimization objective integrates two major losses, *i.e.*, detection loss and domain adversarial loss. The former is applied to the labeled data in the source domain, and the latter is applied to both the source and target domains. As for the detection loss, each stage of Faster R-CNN contains a classification loss and a localization loss, and the total detection loss is defined by

$$\mathcal{L}_{det} = \mathcal{L}_{cls}^{RPN} + \mathcal{L}_{loc}^{RPN} + \mathcal{L}_{cls}^{RPC} + \mathcal{L}_{loc}^{RPC}. \quad (2)$$

Then the overall objective can be presented as follows:

$$\mathcal{L} = \mathcal{L}_{det} + \lambda_1 \mathcal{L}_{ADV}, \quad (3)$$

where λ_1 is the trade-off parameter.

Inspired by previous methods [47, 56], we further adopt AdaIN [23] to align the low-level features (*e.g.* features after conv1 and conv2). Specifically, we adjusts the mean and variance of source features by those of target features during training, and directly use target features for testing.

3.2. RPN Prototype Alignment

To better understand the proposed RPN prototype alignment, we first review the training and test procedures of Faster R-CNN, whose structure is shown in the middle part of Figure 2. Here we split the RPN into three modules, *i.e.*, $\texttt{RPN}_{conv}, \texttt{RPN}_{cls}, \texttt{RPN}_{reg}$, each of which consists of a convolutional layer. In the training phase, the backbone network G takes an image as input and produces the corresponding global feature $\boldsymbol{f}_g \in \mathbb{R}^{C\times H\times W}$. Then RPN takes $\boldsymbol{f}_g$ into $\texttt{RPN}_{conv}$ to produce the global RPN features $\boldsymbol{f}_{rpn}$. The feature $\boldsymbol{f}_{rpn}$ is taken by $\texttt{RPN}_{cls}$ and $\texttt{RPN}_{reg}$ to perform the classification and box regression, where the pre-defined anchors are used. To assign each anchor box a label, two IoU thresholds are usually adopted, *i.e.*, the foreground threshold (*e.g.* 0.7) and background threshold (*e.g.* 0.3). Finally, the top-N (*e.g.* $N = 2000$) boxes after NMS with high foreground probabilities are sent to RPC as the training samples of different objects. In the test phase, RPN provides the top-M (*e.g.* $M = 300$) foreground boxes after NMS to feed RPC, and the final results are produced by RPC from these boxes.

According to the structure of Faster R-CNN and its training procedure, an intuitive idea to implement domain adaptation in the RPN stage is to generate the pseudo labels (*e.g.*, probability) of all RPN proposals in the target domain and then apply them to the training procedure. In such a way, however, the number of foreground proposals are more than enough, and at the same time the obtained locations are imprecise. Consequently, the training of RPN may be misled.

In this paper, we propose an RPN prototype based feature alignment method. The core idea is to first explicitly construct two learnable prototypes $\boldsymbol{p}_i \in \mathbb{R}^C, i \in \{0, 1\}$, and then enforce the RPN features in both source and target domains to align with the corresponding prototypes. To be specific, the foreground RPN features are enforced to

align with $\boldsymbol{p}_1$, and the background RPN features are with $\boldsymbol{p}_0$. With the prototype intermediary, the RPN features in the source and target domains can be aligned automatically, and the learned $\texttt{RPN}_{conv}$ would achieve better transferability across domains. Further, high-quality proposals can be produced for the target domain.

Figure 2 illustrates the framework of our proposed domain adaptation method, where the top part shows the prototype based feature alignment, *i.e.*, *RPN prototype alignment*, and the bottom part shows the proposal feature generation for alignment. In general, we use the detection results of RPC to construct the pseudo labels of objects for the target domain (the ground truth is directly used for the source domain). Then we use them to generate the foreground and background proposals, in which the spatial information (*i.e.*, IoU) is particularly adopted to assign labels rather than the predicted probability by RPC. Finally, the RPN proposal features pass through $\texttt{RPN}_{conv}$ to get the RPN alignment features, which are used to align with the prototypes.

RPN Feature Generation for Alignment. As shown in Figure 2, in order to get the RPN feature for alignment in the top part, we need first generate proposal feature in the bottom part. The proposal feature generation involves two main procedures, *i.e.*, pseudo label generation for target domain objects, and proposal label assignment for both domains. Here we elaborate on their details.

As for the pseudo label generation, previous work [67, 61] often uses soft probabilities (or one-hot format) produced by RPC to label the proposals, where the spatial information is ignored. Differently, we propose to use the box information of different classes of detection results after NMS as the pseudo labels of foreground objects. Specifically, we first set a probability threshold (*i.e.*, 0.05) to filter the low-confidence boxes. Then we adopt a class-agnostic ratio ρ to only keep the top high-confidence boxes for each class separately. Through this way, different classes of foreground objects can be more balanced to be reserved. This is especially important for the low-confidence classes with scarce training samples to perform domain adaptation since they cannot be readily aligned in practice. In the training phase, we generate the above pseudo labels every T (*e.g.*, $T = 3000$) iterations, which are used to guide the learning of next T iterations.

Given the ground truth (or pseudo label), we first use them to label the proposals, then we select a portion from all proposals, and finally the selected proposals would be passed through $\texttt{RPN}_{conv}$ to generate the RPN features for alignment. Specifically, we determine if the top-N proposals belong to the foreground or background by their IoU with ground truth (or pseudo label), where the threshold strategy is used. Then we take all foreground proposals and randomly selected background proposals of the same amount to construct the training samples for RPN feature alignment. Formally, given the global backbone feature $\boldsymbol{f}_g$ and an RPN proposal box B_i, the proposal feature $\boldsymbol{f}_{bk}^i$ and RPN feature for alignment $\boldsymbol{f}_{rpn}^i$ is generated as follows

$$\boldsymbol{f}_{bk}^i = \texttt{RoIAlign}(\boldsymbol{f}_g, B_i), \tag{4}$$

$$\boldsymbol{f}_{rpn}^i = \frac{1}{HW}\sum_{h,w}(\texttt{RPN}_{conv}(\boldsymbol{f}_{bk}^i))(h,w), \tag{5}$$

where $\texttt{RoIAlign}$ proposed in [18] is used to generate the proposal feature from the global feature, $\boldsymbol{f}_{bk}^i \in \mathbb{R}^{C\times H\times W}$ is the proposal feature corresponding to B_i, and $\boldsymbol{f}_{rpn}^i \in \mathbb{R}^C$ is its RPN feature for alignment.

Contrastive Alignment Loss. In our framework, the RPN features for alignment are expected to align with the prototypes for both source and target domains, where the foreground and background are processed separately. Specifically, for an RPN feature $\boldsymbol{f}_{rpn}^i$ and its label $y_i \in \{0, 1\}$ (0 for background and 1 for foreground), we enforce $\boldsymbol{f}_{rpn}^i$ close to $\boldsymbol{p}_{y_i}$ and at the same time far away from another prototype $\boldsymbol{p}_{1-y_i}$. To this end, we choose the contrastive loss to train the network, in which the cosine distance is adopted to measure the similarity between the RPN alignment features and prototypes. Then the loss can be presented as follows:

$$\begin{aligned}
\mathcal{L}_{pos}^i &= 1 - \cos(\boldsymbol{f}_{rpn}^i, \boldsymbol{p}_{y_i}),\\
\mathcal{L}_{neg}^i &= \max(0, \cos(\boldsymbol{f}_{rpn}^i, \boldsymbol{p}_{1-y_i}) - m),\\
\mathcal{L}_{Align} &= \sum_{i=1}^{N}\frac{1}{N}(\mathcal{L}_{pos}^i + \mathcal{L}_{neg}^i),
\end{aligned} \tag{6}$$

where $\cos(x_1, x_2) = \frac{x_1^\intercal \cdot x_2}{\|x_1\|\|x_2\|}$ is the cosine similarity, N is the number of selected proposals, and m is the margin which is set to 0 in our experiments.

Combined with the baseline model, the overall optimization objective becomes

$$\mathcal{L} = \mathcal{L}_{det} + \lambda_1\mathcal{L}_{ADV} + \lambda_2\mathcal{L}_{Align}, \tag{7}$$

where λ_1 and λ_2 are the control parameters.

3.3. Discriminability-aware Alignment

In the above process, all spatial locations within an RPN proposal are treated equally. But the foreground proposals often contain some background pixels, which would bring interference into the RPN alignment features. To further increase the discriminability of foreground RPN features, we propose a discriminability-aware alignment method, which allows alignment to mainly focus on the discriminative object regions. Specifically, we use the $\texttt{Grad CAM}$ [45] on RPC to find the discriminative regions with respect to the ground truth (or pseudo label) class, and then use the map

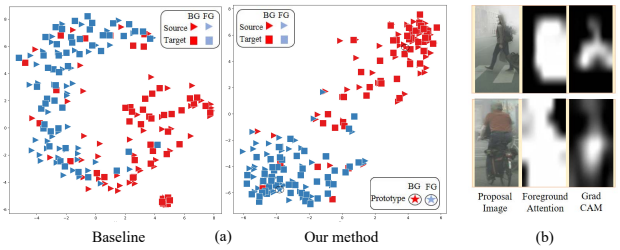

Figure 3. (a) TSNE of RPN features produced by different methods. The feature points are obtained by spherical k-means. Experiments are conducted on SIM10k → Cityscapes scenario. (b) Visual comparison between Grad CAM and foreground attention.

to weight the alignment features. That is, the foreground RPN alignment feature is generated by

$$\begin{aligned} \boldsymbol{f}_{bk}^{i} &= \texttt{RoIAlign}(\boldsymbol{f}_g, B_i), \\ P(h,w) &= (1 + \texttt{cam}_i(h,w)), \\ \boldsymbol{f}_{rpn}^{i} &= \frac{1}{HW}\sum_{h,w} P(h,w)(\texttt{RPN}_{conv}(\boldsymbol{f}_{bk}^{i}))(h,w), \end{aligned} \tag{8}$$

where $\texttt{cam}_i$ is the attention map for the proposal B_i produced by the original CAM.

To intuitively show the effect of our method, we first visualize the features and prototypes as shown in Figure 3 (a). We can see that the prototypes can represent the features well, and the foreground and background are better separated. We further visualize the CAM attention and foreground attention in Figure 3 (b). Here the foreground attention is generated by calculating the normalized cosine similarity between the RPN features and prototypes. It can be seen that the CAM attention can focus on the discriminative part of objects (*e.g.*, the body of a person), and the foreground attention can almost cover the whole object.

4. Experiments

4.1. Datasets and Scenarios

Following [67], we evaluate different methods under the three adaptation scenarios.

Normal-to-Foggy. Cityscapes [7] is a street scene dataset for driving, whose images are collected in the clear weather. It consists of $2,975$ images for training and 500 images for validation. The Foggy Cityscapes [44] dataset is synthesized from Cityscapes for the foggy weather. In the training phase, we use the training set of Cityscapes and Foggy Cityscapes as the source and target domains, respectively. The results on the validation set of Foggy Cityscapes are reported.

Synthetic-to-Real. Sim10k [24] is a collection of synthesized images, which consists of $10,000$ images and corresponding bounding box annotations. To adapt the synthetic scenes to the real ones, we utilize the entire SIM10k dataset as the source domain and the training set of Cityscapes as the target domain. Since only *Car* is annotated in both domains, we report the performance of *Car* on the validation set of Cityscapes.

Cross-Camera. KITTI [14] is a similar scene dataset to Cityscapes except that KITTI has different camera setup. It consists of $7,481$ labeled images for training. To simulate the cross-camera adaptation, we use the training set of *Cityscapes* as the source domain and the training set of *KITTI* as the target domain. Here we follow [6, 58] to classify $\{$*Car*, *Van*$\}$ as *Car*, $\{$*Pedestrian*, *Person sitting*$\}$ as *Person*, *Tram* as *Train*, *Cyclist* as *Rider* for matching Cityscapes and KITTI. The results in the training set of KITTI are reported, as in [6, 58].

4.2. Implementation Details

Here we adopt the Faster R-CNN with VGG16 [46] as the backbone that is pre-trained on ImageNet [9]. We resize the shorter sides of all images to 600 pixels. The batch size is set to 2, *i.e.*, one image per domain. The detector is trained with SGD for $50k$ iterations with the learning rate of 10^{-3}, and it is then dropped to 10^{-4} for another $30k$ iterations. The prototypes are also trained with SGD but the learning rate is 10 times of that of the detector. The domain discriminators are trained by the Adam optimizer [25] with the learning rate of 10^{-4}. The factor $\lambda_1 = 1.0$ and $\lambda_2 = 0.1$ are set. The class-agnostic ratio ρ is set to 0.6 for all scenarios. During training, we add the RPN alignment loss from the iteration of $25k$ and the update interval of pseudo label is set to $3k$ iterations. We report mAP with an IoU threshold of 0.5 for evaluation.

4.3. Comparison with State-of-the-Art

Here we compare our proposed method with recently published state-of-the-art methods. In particular, "Source Only" denotes the model that is directly trained using labeled source data without involving the target data. "Baseline" represents the model that only performs feature alignment in backbone network except for *Normal-to-Foggy* scenario, in which we also adopt pixel-level style transformation from the source domain to target domain with CycleGAN [68]. "RPA" represents the model using the basic RPN feature alignment in Sec. 3.2, and "Final" represents our final model combined with discriminability-aware alignment in Sec. 3.3. In addition, we provide the "Oracle" results, in which the model is trained using the labeled data in target domain as in supervised learning.

Normal-to-Foggy. Table 1 gives the results of different methods for this scenario, and we have the following observations. First, compared with the baseline, our RPA approach can boost the performance by 1.5%, and further increase by 0.5% when combined with discriminability-aware alignment, which show the effectiveness of our proposed

Table 1. Detection performance (%) on *Normal-to-Foggy* cross-domain adaptation task, Cityscapes → Foggy Cityscapes.

Method	Bus	Bicycle	Car	Motor	Person	Rider	Train	Truck	**mAP**
DAF (CVPR'18) [6]	35.3	27.1	40.5	20.0	25.0	31.0	20.2	22.1	27.6
SCDA (CVPR'19) [69]	39.0	33.6	48.5	28.0	33.5	38.0	23.3	26.5	33.8
SWDA (CVPR'19) [42]	36.2	35.3	43.5	30.0	29.9	42.3	32.6	24.5	34.3
CR (CVPR'20)[59]	45.1	34.6	49.2	30.3	32.9	43.8	36.4	27.2	37.4
C2F (CVPR'20) [67]	43.2	37.4	52.1	34.7	34.0	46.9	29.9	30.8	38.6
HTCN (CVPR'20) [4]	47.4	37.1	47.9	32.3	33.2	47.5	40.9	31.6	39.8
CoT (ECCV'20)[64]	45.6	36.8	50.1	30.1	32.7	44.4	25.4	21.7	35.9
CDN (ECCV'20) [47]	42.5	36.5	50.9	30.8	35.8	45.7	29.8	30.1	36.6
DMLP (ECCV'20) [66]	44.1	36.6	43.9	37.4	32.0	42.1	43.4	31.3	38.8
Tri-way (ECCV'20) [20]	43.3	38.8	50.0	33.4	34.6	47.0	38.7	23.7	38.7
SAPNet (ECCV'20) [26]	46.8	40.7	59.8	30.4	40.8	46.7	37.5	24.3	40.9
Source Only	25.0	26.8	30.6	15.5	24.1	29.4	4.6	10.6	20.8
Baseline	44.0	34.9	49.0	31.0	32.7	44.0	33.8	26.5	37.0
Ours (RPA)	44.8	36.3	50.1	29.9	33.4	44.3	39.1	29.9	38.5
Ours (Final)	43.6	36.8	50.5	29.7	33.3	45.6	42.0	30.4	39.0
HTCN + our proposals	45.5	36.8	49.6	35.7	33.6	43.8	46.0	32.9	40.5
Oracle	47.7	37.1	52.3	35.6	33.8	45.0	46.7	34.6	41.6

Table 2. Detection performance (%) on *Synthetic-to-Real* cross-domain adaptation task, SIM10k → Cityscapes.

Methods	***car* AP**
DAF (CVPR'18) [6]	39.0
SWDA (CVPR'19) [42]	42.3
SCDA (CVPR'19) [69]	43.0
HTCN (CVPR'20) [4]	42.5
C2F (ECCV'20) [67]	43.8
CoT (ECCV'20) [64]	44.5
Source-only	34.6
Baseline	42.3
Ours (RPA)	45.3
Ours (Final)	45.7
Oracle	60.0

Table 3. Detection performance (%) on *Cross Camera* cross-domain adaptation task, Cityscapes → KITTI.

Method	Person	Rider	Car	Truck	Train	**mAP**
DAF (CVPR'18) [6]	40.9	16.1	70.3	23.6	21.2	34.4
MDA (ICCV'19) [58]	53.3	24.5	72.2	28.7	25.3	40.7
C2F (CVPR'20) [67]	50.4	29.7	73.6	29.7	21.6	41.0
Source-only	49.9	17.2	73.7	16.4	13.0	34.0
Baseline	56.0	27.3	75.1	25.8	23.6	41.6
Ours (RPA)	56.5	27.3	75.0	41.1	21.0	44.2
Ours (Final)	56.2	30.62	75.1	39.4	23.0	44.8
Oracle	72.4	86.0	89.2	90.6	89.8	85.6

methods. Second, our method greatly outperforms the CoT [64] which also performs the RPN learning across domains (39.0% *vs*. 35.9%). Third, our method can bring significant improvement for the classes with relatively scarce instances, *e.g*., *Train*, *Truck*, and *Bus*. Finally, to further investigate the effect of our proposed method, we feed our produced RPN proposals (box information) into the HTCN [4] model to perform inference. It can be seen that the mAP of HTCN is boosted from 39.8% to 40.5%, which shows the quality of our produced proposals.

Synthetic-to-Real. Table 2 reports the results on the *car* category. It can be seen that both the RPA and final versions of our method outperform the existing methods. Compared with the similar method CoT [64], our method has an mAP gain of 1.2%.

Cross-Camera. Table 3 gives the results on the five categories. We can see that that our final model outperforms the existing works a lot, which verifies the effectiveness of our model to alleviate the domain shift caused by cross-camera.

4.4. Analysis and Discussion

In this section, we conduct several experiments to analyze our model from the design of RPA, parameter sensitivity, and visualization. For the sake of simplicity, we use C, F, S, and K to denote Cityscapes, Foggy Cityscapes, Sim10k, and KITTI, respectively.

Design Choice of RPA. We particularly consider the designs of pseudo label generation and pseudo label utilization in our RPA framework. For pseudo label generation, we can also use the online soft (or hard) probabilities generated by PRC to label each proposal, instead of our assignment by IoU. For pseudo label usage, we can also apply pseudo labels to directly guide the RPN training loss, instead of our RPA. Here the "cls loss" means the pseudo labels are used only for RPN classification and "cls+box loss" means the

(a) Source Only (b) Baseline (c) Ours (d) Foreground Attention

Figure 4. Visualization of detection results on the *Synthetic-to-Real* scenario for different methods. Here the foreground attention is additionally visualized which represents the normalized cosine similarity between the global RPN features and RPN prototypes.

Table 4. Design choices of pseudo label generation and usage.

#	Choices	Methods	C→F	S→C	C→K
1		**Baseline**	37.0	42.3	41.6
2	Label generation	hard label	37.5	42.8	42.8
3		soft label	37.8	42.8	42.9
4	Label usage	cls loss	37.3	42.5	41.9
5		cls+box loss	34.9	39.2	38.5
6		**Ours**	39.0	45.7	44.8

pseudo labels are used for both RPN classification and box regression. Table 4 gives the results on different scenarios.

From the results of pseudo label generation, it can be seen that both soft and hard pseudo labels from RPC can achieve better performance than the baseline, but they are inferior to our proposed method with a large margin. In practice, we observe that the soft (or hard) label method would generate much more foreground samples than ours, which will introduce more inaccurate proposals. From the results of pseudo label usage, it can be seen that directly applying the pseudo labels to the training of RPN classification can boost the performance compared with the baseline. But additionally applying the pseudo labels to RPN box regression would lead to the performance degradation. Evidently, the pseudo labels are not accurate enough to directly guide the training of RPN.

Parameter sensitivity. Here we investigate the influence of class-agnostic ratio ρ used in pseudo label generation, which essentially controls the ratio of samples reserved for each class. Table 5 gives the results about sensitivity of ρ on different scenarios. It can be seen that our method is robust for a wide range of ρ, and $\rho = 0.6$ is particularly chosen for all cross-domain experiments.

Table 5. Performance for different class-agnostic ratio ρ.

	0.3	0.4	0.5	0.6	0.7	0.8
C→F	37.8	38.2	38.7	39.0	38.9	38.5
S→C	44.7	45.2	45.6	45.7	45.3	44.9
C→K	43.9	44.0	44.4	44.8	44.7	44.5

Visualization. Here we visualize some detection results of different methods in Figure 4, along with our foreground attentions representing the normalized cosine similarity between the global RPN features and RPN prototypes. It can be seen that our method can generate more accurate and clean detection results than the baseline, and the learned prototypes can precisely locate the foreground objects.

5. Conclusion

In this paper, we present a novel RPN prototype alignment method for cross-domain object detection, which enforces the RPN features in both domains to align with the learnable prototypes of foreground and background, respectively. In particular, we propose a simple yet effective pseudo label generation method to guide the learning of RPN features in target domain. Furthermore, to increase the discriminability of foreground RPN features, we propose to generate the attention maps from RPC to spatially modulate the RPN features. Comprehensive experiments on different scenarios validate the effectiveness of our proposed method.

6. Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grant 61673362 and 61836008, Youth Innovation Promotion Association CAS (2017496). We acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

References

[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010.

[2] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. In *NeurIPS*, 2007.

[3] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, 2017.

[4] Chaoqi Chen, Zebiao Zheng, Xinghao Ding, Yue Huang, and Qi Dou. Harmonizing transferability and discriminability for adapting object detectors. In *CVPR*, 2020.

[5] Min-Hung Chen, Zsolt Kira, Ghassan AlRegib, Jaekwon Yoo, Ruxin Chen, and Jian Zheng. Temporal attentive alignment for large-scale video domain adaptation. In *ICCV*, 2019.

[6] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *CVPR*, 2018.

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[8] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[10] Zhengming Ding, Sheng Li, Ming Shao, and Yun Fu. Graph adaptive knowledge transfer for unsupervised domain adaptation. In *ECCV*, 2018.

[11] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[12] Yang Fu, Yunchao Wei, Guanshuo Wang, Yuqian Zhou, Honghui Shi, and Thomas S. Huang. Self-similarity grouping: A simple unsupervised cross domain adaptation approach for person re-identification. In *ICCV*, 2019.

[13] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

[15] Ross Girshick. Fast r-cnn. In *ICCV*, 2015.

[16] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[19] Zhenwei He and Lei Zhang. Multi-adversarial faster-rcnn for unrestricted object detection. In *ICCV*, 2019.

[20] Zhenwei He and Lei Zhang. Domain adaptive object detection via asymmetric tri-way faster-rcnn. In *ECCV*, 2020.

[21] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016.

[22] Cheng-Chun Hsu, Yi-Hsuan Tsai, Yen-Yu Lin, and Ming-Hsuan Yang. Every pixel matters: Center-aware feature alignment for domain adaptive object detector. In *ECCV*, 2020.

[23] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[24] M. Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In *ICRA*, 2017.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[26] Congcong Li, Dawei Du, Libo Zhang, Longyin Wen, Tiejian Luo, Yanjun Wu, and Pengfei Zhu. Spatial attention pyramid network for unsupervised domain adaptation. In *ECCV*, 2020.

[27] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[28] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[29] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *IJCV*, 2020.

[30] Songtao Liu, Di Huang, and Yunhong Wang. Receptive field block net for accurate and fast object detection. In *ECCV*, 2018.

[31] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.

[32] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.

[33] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *NeurIPS*, 2018.

[34] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, 2017.

[35] Yawei Luo, Ping Liu, Tao Guan, Junqing Yu, and Yi Yang. Significance-aware information bottleneck for domain adaptive semantic segmentation. In *ICCV*, 2019.

[36] Fatemeh Mirrashed, Vlad I Morariu, Behjat Siddiquie, Rogerio S Feris, and Larry S Davis. Domain adaptive object detection. In *WACV*, 2013.

[37] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, pages 1345–1359, 2009.

[38] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *AAAI*, 2018.

[39] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, 2019.

[40] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

[41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *T-PAMI*, 39(6):1137–1149, 2017.

[42] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *CVPR*, 2019.

[43] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, 2018.

[44] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *IJCV*, 126(9):973–992, 2018.

[45] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

[46] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint*, arXiv:1409.1556, 2014.

[47] Peng Su, Kun Wang, Xingyu Zeng, Shixiang Tang, Dapeng Chen, Di Qiu, and Xiaogang Wang. Adapting object detectors with conditional domain normalization. In *ECCV*, 2020.

[48] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV*, 2016.

[49] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011.

[50] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018.

[51] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

[52] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[53] David Vázquez, Antonio M López, and Daniel Ponsa. Unsupervised domain adaptation of virtual and real worlds for pedestrian detection. In *ICPR*, 2012.

[54] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.

[55] Meng Wang and Xiaogang Wang. Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In *CVPR*, 2011.

[56] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gokhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S Davis. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 518–534, 2018.

[57] Yu Xia, Di Huang, and Yunhong Wang. Detecting smiles of young children via deep transfer learning. In *ICCV Workshops*, 2017.

[58] Rongchang Xie, Fei Yu, Jiachao Wang, Yizhou Wang, and Li Zhang. Multi-level domain adaptive learning for cross-domain detection. In *ICCV Workshops*, 2019.

[59] Chang-Dong Xu, Xing-Ran Zhao, Xin Jin, and Xiu-Shen Wei. Exploring categorical regularization for domain adaptive object detection. In *CVPR*, 2020.

[60] Jiaolong Xu, Sebastian Ramos, David Vázquez, and Antonio M López. Domain adaptation of deformable part-based models. *T-PAMI*, 36(12):2367–2380, 2014.

[61] Minghao Xu, Hang Wang, Bingbing Ni, Qi Tian, and Wenjun Zhang. Cross-domain detection via graph-induced prototype alignment. In *CVPR*, 2020.

[62] Werner Zellinger, Thomas Grubinger, Edwin Lughofer, Thomas Natschläger, and Susanne Saminger-Platz. Central moment discrepancy (cmd) for domain-invariant representation learning. In *ICLR*, 2017.

[63] Yixin Zhang and Zilei Wang. Joint adversarial learning for domain adaptation in semantic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6877–6884, 2020.

[64] Ganlong Zhao, Guanbin Li, Ruijia Xu, and Liang Lin. Collaborative training between region proposal localization and classification for domain adaptive object detection. In *ECCV*, 2020.

[65] Shanshan Zhao, Huan Fu, Mingming Gong, and Dacheng Tao. Geometry-aware symmetric domain adaptation for monocular depth estimation. In *CVPR*, 2019.

[66] Zhen Zhao, Yuhong Guo, Haifeng Shen, and Jieping Ye. Adaptive object detection with dual multi-label prediction. In *ECCV*, 2020.

[67] Yangtao Zheng, Di Huang, Songtao Liu, and Yunhong Wang. Cross-domain object detection through coarse-to-fine feature adaptation. In *CVPR*, 2020.

[68] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

[69] Xinge Zhu, Jiangmiao Pang, Ceyuan Yang, Jianping Shi, and Dahua Lin. Adapting object detectors via selective cross-domain alignment. In *CVPR*, 2019.