Adapting Object Detectors via Selective Cross-Domain Alignment

Xinge Zhu[†] Jiangmiao Pang[§] Ceyuan Yang[†] Jianping Shi[‡] Dahua Lin[†]

[†]The Chinese University of Hong Kong [§]Zhejiang University [‡]SenseTime Research

{zx018,cyyang,dhlin}@ie.cuhk.edu.hk

pjm@zju.edu.cn, shijianping@sensetime.com

Abstract

State-of-the-art object detectors are usually trained on public datasets. They often face substantial difficulties when applied to a different domain, where the imaging condition differs significantly and the corresponding annotated data are unavailable (or expensive to acquire). A natural remedy is to adapt the model by aligning the image representations on both domains. This can be achieved, for example, by adversarial learning, and has been shown to be effective in tasks like image classification. However, we found that in object detection, the improvement obtained in this way is quite limited. An important reason is that conventional domain adaptation methods strive to align images as a whole, while object detection, by nature, focuses on local regions that may contain objects of interest. Motivated by this, we propose a novel approach to domain adaption for object detection to handle the issues in "where to look" and "how to align". Our key idea is to mine the discriminative regions, namely those that are directly pertinent to object detection, and focus on aligning them across both domains. Experiments show that the proposed method performs remarkably better than existing methods with about 4% $\sim$ 6% improvement under various domain-shift scenarios while keeping good scalability.

1. Introduction

Over the past several years, the advances in deep learning has significantly pushed forward the state of the art in various tasks in computer vision, such as object detection [3, 7, 14, 15, 18, 37, 34] and semantic segmentation [27, 47]. Yet, it should be noted that such significant progress relies, to a large extent, on large-scale annotated training data. Whereas several public benchmarks [6, 11, 30] have already existed, they can only cover a very limited range of scenarios. In real-world deployment, the changes in environmental conditions, *e.g.* imaging sensors, weather, and illumination, can cause significant domain shift, and thus substantial performance degradation, as shown in Fig 1.

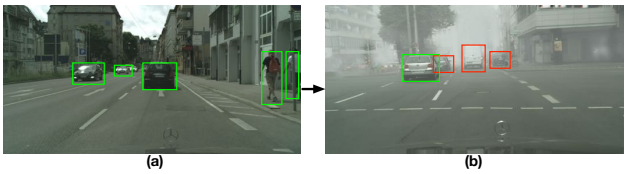

Figure 1. We show two examples from the Cityscapes (a) and Foggy-Cityscapes (b) dataset, respectively. It can be found that there exists large domain gap between two images in appearance distributions. The model trained on the Cityscapes dataset is directly applied on the Foggy-Cityscapes (from a $\rightarrow$ b), and it can be observed that performance drop occurs (boxes with green color mean correct results and red color denotes missing object).

A natural idea to tackle this issue is to obtain new training data as the domain shifts. Unfortunately, this approach is not always feasible in practice, due to the huge cost needed for large-scale annotation. The cost is especially high for object detection or instance segmentation, as it requires detailed annotation, *e.g.* bounding boxes or masks on individual objects. An appealing alternative is *unsupervised domain adaptation*, namely adapting a model trained on standard datasets to a new domain (often referred to as the *target domain*), but without annotating the target data. A variety of methods have been developed along this line, which have shown encouraging results on image classification [1, 12, 13, 16, 28, 43] and semantic segmentation [20, 39, 42, 46, 49]. However, how to effectively adapt an object detector remains a widely open question.

In our initial attempts, we directly applied existing domain adaptation methods [10] to object detection (with necessary technical adjustment), only to find very limited performance gain (more details are showed in the experiments). Our investigation into this issue reveals a key difference between image classification and object detection: in the context of object detection, we usually see an image of a complex scene, where the objects of interest only occupy a small region thereof. Hence, attention to such local

regions is crucial to the success of object detection. To the contrary, conventional domain adaptation methods typically consider the input image as a whole when trying to bridge the domain gap, while neglecting the local nature of object detection. Consequently, their efforts to minimize the domain gap at the image level are met with fundamental difficulties (due to the significant variations in both structures and appearance); yet on the other hand, the local objects are not sufficiently attended.

Motivated by these findings, we propose a new approach to adapting object detectors. The basic idea is to reposition the focus of the adaptation process, from global to local. Specifically, we develop a new framework that consists of two key components, *region mining* and *region-level alignment*, which respectively address the questions of "where to look" and "how to align". Here, *region mining* resorts to a grouping strategy to identify the most important local regions, so as to enhance the robustness against outliers (which often arise in practice); while *region-level alignment* first leverages the region proposal in the source domain to reweigh the target region proposals, thus overcoming the difficulty caused by the lack of target annotations, and then performs the region-level domain alignment in an adversarial manner. Overall, the cooperation of these two components leads to an adaptation process that focuses on the regions of interest, thus improving the effectiveness.

We tested the proposed method under various domain-shift settings, including *normal-to-foggy* (Cityscapes to Cityscapes-foggy), *synthetic-to-real* (Sim10k to Cityscapes), and *cross-camera* (Kitti to Cityscapes). On these experiments, the proposed method yields considerable improvement over existing methods, about $\mathbf{4}\%$ to $\mathbf{6}\%$ in mAP. We also extend the method to instance segmentation, obtaining notable performance gain, which further demonstrates its strong generalization capability and scalability.

The contributions of this work mainly lie in three aspects: (1) Our studies reveal a crucial aspect to the success of object detector adaptation, namely, the focus to local regions when bridging domain gaps. (2) We develop a new domain adaptation framework for object detection, which repositions the focus of the adaptation process, through effective region mining and region-based domain alignment. (3) We conduct extensive experiments to compare the proposed methods with others on various settings, where it yields notable performance gains not only in object detection but also in instance segmentation.

2. Related Work

Object Detection. Object detection has been a central topic of computer vision. Following the lead of R-CNN [15], a number of object detection frameworks based on convolutional networks have been developed in recent years, which significantly push forward the state of the art. Whereas single-stage detectors have emerged as a popular paradigm [26, 36], many top-performing frameworks still adopt the proven two-stage pipeline, *e.g*. Fast R-CNN [14], Faster R-CNN [37], and Mask R-CNN [18], etc.

However, even top-notched object detectors still face signficant challenges when used in real-world settings. The difficulties usually arise from the changes in environmental conditions. For example, a state-of-the-art detector trained on a public dataset often finds it difficult to work reliably in an autonomous driving system where the weather and imaging conditions can vary significantly – existing datasets [6, 11, 22] can only provide limited coverage of such cases. Whereas collecting more data under various conditions can help, it is prohibitively expensive and labor-demanding.

Domain Adaptation. Domain adaptation [2, 32], namely the techniques to adapt a model to a new domain without re-training from scratch, has received increasing attention in recent years. It is often considered as a promising remedy to tackle the difficulties caused by the lack of domain-specific training data. For domain adaptation, a typical approach is to estimate the domain gap formalized in certain ways and minimize it [12, 16, 28]. Some recent methods go further along this line, using more effective ways to reduce the domain gap, *e.g*. incorporating a domain classifier with gradient reversal [9], or directly reverting the distribution distances [33]. Other representative methods include subspace alignment [8], asymmetric kernel transforms [24], tensor-based adaptation [29], and shared encoding for classification and reconstruction [13]. It is noteworthy that the works mentioned above mainly devised on the task of image classification and semantic segmentation. They simply consider the entire image as a whole, while focusing on the design of losses or metrics.

Domain Adaptation for Detection. Whereas there have been extensive studies on the domain adaptation methods for image classification and semantic segmentation, the study on adapting object detectors is still at a relatively earlier stage. That being said, several ideas have been explored in existing attempts. Raj *et al*. [35] tried to adapt class-specific R-CNN detectors by subspace alignment. Inoue *et al*. [21] proposed a weakly-supervised object detection framework, where domain transforms and the pseudo-label technique are employed. Chen *et al*. [4] incorporated a gradient reversal layer [9] into a Faster R-CNN framework in order to reduce the domain gap.

It is worth noting that while the aforementioned efforts have shown encouraging results, the improvement remains limited. As mentioned, a common issue of these works is that they mainly focus on bridging the whole-image representations. This may not be very effective, considering the local nature of object detection. Compared to these works,

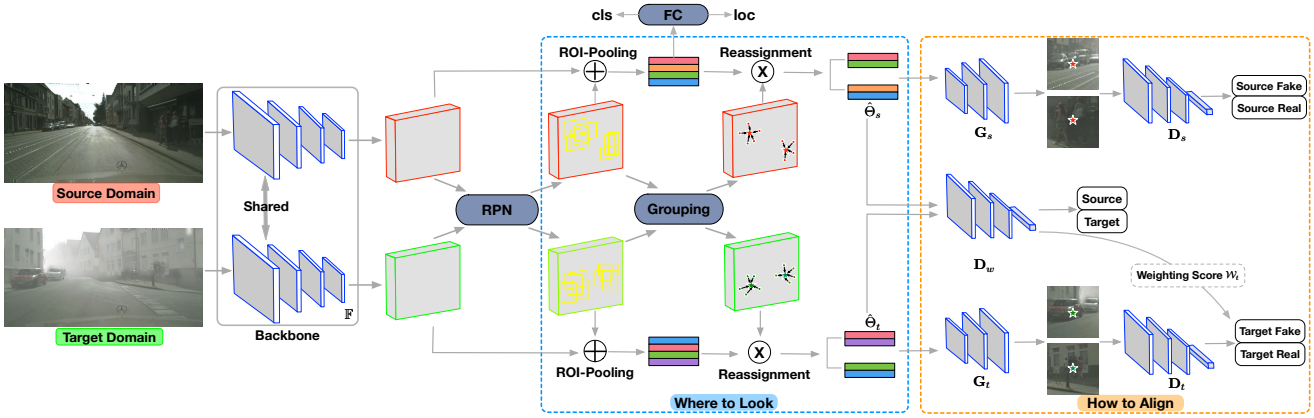

Figure 2. The pipeline of our framework. Two major components, *i.e.* "Where to Look" and "How to Align" are illustrated with two dashed rectangles. For the first component, an ROI-based grouping strategy is designed to mine the discriminative regions for two domains. We display the grouping procedure with cluster number = 2 (Note that ★ and ★ denote the centroids of clusters). For the second one, our model performs the adjusted region-level alignment using generators (G_s and G_t), discriminators (D_s and D_t) and weighting estimator (D_w). We use Faster R-CNN as the detection model ($\mathbb{F}$) which consists of the backbone, RPN and head part. (**Best viewed in color**)

our proposed method differs essentially in that it reshapes the focus of domain alignment, from global to local.

Domain Adaptation through Adversarial Learning. Generative Adversarial Network (GAN) [17] has received great attention in recent years. Inspired to this, some recent works extend this new learning paradigm to domain adaptation. In [1, 43], adversarial discriminators are incorporated to mitigate the impact of domain gap. In [19, 25, 48, 45], unsupervised style transforms are learned to close the gap in appearance between the source and target domains. Other efforts [5, 9, 20, 49, 41], instead focus on the learning of domain-invariant features. In our work, we adopt adversarial learning as the basic machinery for learning region-level alignment. It is important to note that our framework does not require additional annotations and the entire network can be trained in an end-to-end fashion.

3. Methodology

3.1. Framework Overview

We consider a problem that involves two domains, a *source domain* where annotated training data are available and a *target domain* where we only have access to the images. Our task is to train a detector that can generalize well to the target domain, utilizing the data in both domains. Specifically, we desire to obtain a *domain-invariant* feature representation that works equally well in both domains.

To this end, we propose a selective adaptation framework based on region patches. The basic idea is to introduce an additional module to reconstruct the image patches from the features, and then align the reconstructed patches in both the source and target domains. During the training this module can guide the learning of features via back-propagation, reducing the gap between domains. After training, the alignment module is no longer needed. Only the detection part will be used for effective inference, while benefiting from the learned domain-invariant features.

As mentioned, aligning the entire image is difficult, due to large variations in background appearance and scene structures. Our framework, instead, focuses on aligning those local regions that contain objects of interest. As shown in Figure 2, the framework consists of two key components: (1) a *region mining* component to address the question of "*where to look*", selecting those important regions by grouping the object proposals; and (2) a *region-level alignment* component to address the question of "*how to align*", which learns to align the image patches reconstructed from the features of the selected regions, via adversarial learning. Particularly, for this component, two generators G_s and G_t are respectively used to generate image patches based on the features extracted from the source and target regions, and then a set of discriminators are introduced to bridge the gap between them. In what follows, we will present these components in detail.

3.2. Region Mining

The region mining component first identifies important regions that cover the objects of interest by grouping, and then derives representations for these regions by reassigning the RoI features accordingly.

Grouping. We desire to find those regions that cover objects of interest. A natural idea is to utilize the region proposals derived from the RPN. However, we are still facing two challenges: (1) We wish to obtain regions of fixed size, in order to make it convenient for further processing, *e.g.* feeding them to the region alignment module. But the region proposals can distribute arbitrarily over the image.

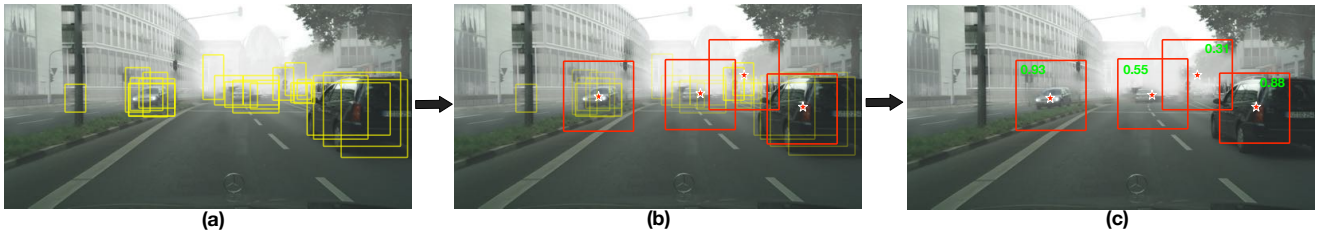

Figure 3. We show the workflow of the grouping scheme and weighting estimator. Image **(a)** $\rightarrow$ **(b)** is the grouping operation, in which yellow rectangles are the region proposals and red squares denote the discriminative regions. ★ is the centroid of cluster (denoted by Ψ) and also the center of discriminative region. **(b)** $\rightarrow$ **(c)** is the process of weighting estimation. The green numbers denote the scores of discriminative regions from target domain. (**Best viewed in color**)

(2) The region proposals from the RPN are often very noisy. It is unnecessary to cover them all.

We tackle this problem by a simple centroid-based grouping scheme. Specifically, after the RPN, we get N_{reg} region proposals in the form of $\{c_x, c_y, w, h\}$, where c_x and c_y are the center coordinates, w the width, and h the height. By applying K-means clustering to the center coordinates, we can obtain K clusters, whose means can be considered as the *centroids* of the grouped regions. Note that in our framework, we fix the size of each region. Hence, given the centroids (from K-means), we automatically have the regions. Figure 3(**(a)**$\rightarrow$**(b)**) shows an example of region grouping with $K = 4$. We can find that the grouping scheme above can identify those regions that cover significant objects, while being resilient to the presence of false proposals.

Feature Reassignment. Given the selected regions, we derive the feature representations thereof by reassigning the RoI features according to the grouping results. Specifically, each region is associated with a subset of region proposals assigned to the corresponding K-means cluster. By stacking the corresponding RoI features, we can obtain a matrix $\Theta_k \in \mathbb{R}^{m_k \times d}$ to represent the k-th region, where m_k is the number of region proposals assigned to the k-th cluster, and d is the feature dimension.

This representation is inconvenient to work with, as the number m_k can vary. It is desirable to fix the number of features. For this purpose, we adopt a simple select-or-copy scheme. Given a pre-defined number m, if m_k is greater than m, we retain only the top-m features; if m_k is less than m, we simply make copies of the assigned features until we get enough. In this way, we can derive a fixed number of features $\hat{\Theta}_k \in \mathbb{R}^{m \times d}$ to represent each region.

3.3. Adjusted Region-level Alignment

After deriving the important regions and their feature representations, we align target regions to the source distributions in an adversarial manner. We first describe the region-level adversarial alignment scheme, then introduce the weighting estimator to perform the adjusted region-level alignment, in order to achieve robust adaptation. Finally, we integrate different components to form a total objective.

Region-Level Adversarial Alignment. Following the conventional practice of adversarial domain adaptation [10, 25, 39], we attach two generators G_s and G_t to reconstruct the mined K regions based on the cluster-wise feature representations $\hat{\Theta}$. We also introduce two discriminators D_s and D_t to distinguish the real and fake inputs and impose the domain alignment constraints. The standard joint objective $\mathcal{L}_{adv}$ is shown as following, which combines both *within-domain* and *cross-domain* losses, as

$$\begin{aligned}\mathcal{L}_{adv}(\mathbb{F}, G, D) = \; & \mathcal{L}_{adv,D_s} + \mathcal{L}_{adv,D_t} + \mathcal{L}_{adv,G_s} \\ & + \mathcal{L}_{adv,G_t} + \mathcal{L}_{adv,\mathbb{F}}.\end{aligned} \tag{1}$$

Each term follows a standard adversarial formulation as:

$$\mathcal{L}_{adv}(\hat{\Theta}, \mathcal{P}; G, D) = \mathbb{E}[\log D(\mathcal{P})] + \mathbb{E}[\log(1 - D(G(\hat{\Theta})))], \tag{2}$$

where $\mathcal{P}$ denotes the real image regions obtained based on the clustering centers Ψ.

Specifically, for discriminators D_s and D_t, within-domain losses, $\mathcal{L}_{adv,D_s}$ and $\mathcal{L}_{adv,D_t}$, aim to classify the real input as real (or fake input as fake). For generators G_s and G_t, $\mathcal{L}_{adv,G_s}$ and $\mathcal{L}_{adv,G_t}$ classify the fake input as real within one domain. For detection model $\mathbb{F}$, $\mathcal{L}^s_{adv,\mathbb{F}}$ performs the cross-domain function by classifying the *fake source* input as *real target* (*i.e.*, feeding the target discriminator with fake source, and similar operation for $\mathcal{L}^t_{adv,\mathbb{F}}$), which imposes the alignment constraint on detection model. By reshaping the focus of adaptation, from global to local, it is more effective to achieve the domain alignment without the difficulties in global statistics, such as significant differences in structures and field of view.

Weighting Estimator D_w. Since there is no ground truth bounding box on the target domain, the region proposals extracted from RPN on target image often fail to cover the objects of interest, especially in the early stage. For example, the recall on target domain is only 35% $\sim$ 45% for the

first 10 epoch. Hence, it is important to emphasize those target regions that are truly relevant to the region-level alignment. To this end, we can resort to the ground-truth bounding boxes in the source domain, which can provide useful references to guide the focus in the target domain.

Based on these findings, we introduce the estimator to weigh the target regions according to how closely they match the source ones. We train the estimator to discriminate representations between source region proposals (labeled as 1) and target proposals (labeled as 0), and the binary cross-entropy loss is used as the objective function.

$$\begin{aligned}\mathcal{L}_w(\hat{\Theta}_s, \hat{\Theta}_t; D_w) = \ & \mathbb{E}[\log(D_w(\hat{\Theta}_s))] + \\ & \mathbb{E}[\log(1 - D_w(\hat{\Theta}_t))],\end{aligned} \tag{3}$$

where $\hat{\Theta}_s$ and $\hat{\Theta}_t$ denote the clustered region representations of source domain and target domain after reassignment, respectively. Here, the scores from D_w are good indicators on how well a target region match the source. We turn these scores into weights for target regions, by applying the sigmoid activation function to the output of estimator $D_w(\hat{\Theta}_t) \in \mathbb{R}^{K\times m}$ (all notations are followed in Section 3.2) and taking the average, thus obtaining the weighting score $\mathcal{W}_t \in \mathbb{R}^K$. We show the workflow of similarity estimation in Figure 3(**(b)**→**(c)**). It can be observed that the higher score indicates that the target region is more likely to contain objects of interest and more similar to the distribution of the source patches. Since $\mathcal{W}_t$ assign weights for target regions, it applies to the terms involving target domain only:

$$\begin{aligned}\mathcal{W}_t \cdot \mathcal{L}_{adv}(\mathbb{F}, G, D) = \ & \mathcal{L}_{adv,D_s} + \mathcal{W}_t\mathcal{L}_{adv,D_t} + \mathcal{L}_{adv,G_s} \\ & + \mathcal{W}_t\mathcal{L}_{adv,G_t} + \mathcal{L}^s_{adv,\mathbb{F}} + \mathcal{W}_t\mathcal{L}^t_{adv,\mathbb{F}}.\end{aligned} \tag{4}$$

Here, we the notation $\mathcal{W}_t \cdot \mathcal{L}_{adv}$ to provide a simple reference to the expression on the right hand side.

Total Objective Function. By incorporating the weighting score $\mathcal{W}_t$, the total optimization of adjusted adversarial alignment can be formulated as:

$$\min_{\mathbb{F},G,D_w} \max_{D} \ \mathcal{L}_{dec}(\mathbb{F}) + \mathcal{W}_t \cdot \mathcal{L}_{adv}(\mathbb{F}, G, D) + \mathcal{L}_w(D_w), \tag{5}$$

where $\mathcal{L}_{dec}$ is the loss for detection task, *i.e.*, $\mathcal{L}_{dec} = \mathcal{L}_{cls} + \mathcal{L}_{loc}$. $\mathcal{L}_{cls}$ is the cross-entropy loss and $\mathcal{L}_{loc}$ denotes the smooth L1 loss. With the constraint of adjusted region-level adversarial alignment, the training process will encourage domain-invariant features through back-propagation.

3.4. Network Optimization

Our full objective is to update the four components, including the detection model $\mathbb{F}$, the generators G_s and G_t, the discriminators D_s and D_t, and the estimator D_w. Inheriting the standard procedure of GAN [17], we alternate the optimization between four steps: 1) update D_s and D_t; 2) update D_w; 3) update G_s and G_t; 4) update $\mathbb{F}$.

	K	m	region-size
Type1	2	256	512×512
Type2	4	128	256×256
Type3	8	64	128×128

Table 1. Three sets of grouping parameters.

Update Discriminators D_s and D_t. We train these discriminators to distinguish between the real regions and the reconstructed ones. For the target image, the score $\mathcal{W}^t$ is used to weigh target regions. The loss for this step is $L_D = \mathcal{L}_{adv,D_s} + \mathcal{W}_t\mathcal{L}_{adv,D_t}$.

Update the Weighting Estimator D_w. We train the weighting estimator D_w to measure the contributions of different target regions. The loss is given by Eq.(3).

Update Generators G_s and G_t. The goal of this step is to encourage realistic output from the generators. The loss for this step is $L_G = \mathcal{L}_{adv,G_s} + \mathcal{W}_t\mathcal{L}_{adv,G_t}$.

Update the Detection part $\mathbb{F}$. We aim to close the gap between the distributions in the source and target domains, while maintaining the detection performance. Hence, the overall loss is the combination of two parts, *i.e.* detection loss and adversarial loss. Note that this adversarial loss is to perform the cross-domain function. The overall loss is formulated as $L_{\mathbb{F}} = \mathcal{L}^s_{dec,\mathbb{F}} + \lambda(\mathcal{L}^s_{adv,\mathbb{F}} + \mathcal{W}_t\mathcal{L}^t_{adv,\mathbb{F}})$.

3.5. Implementation Details

Detection Model. We follow the detection model used in [4] that adopts Faster R-CNN [37] with the VGG16 [40] architecture.

Grouping Strategy and Centroid-based Reassignment. For this part, there are several pre-defined grouping parameters to determine the clustering. In our implementation, three sets of parameters are designed to verify the effect of grouping strategy, which are chosen by cross-validation and shown in Table 1. For example, Type1 denotes that the number of cluster (K) is 2; the number of proposals in one cluster (m) is 256; the reconstructed two region patches have size of 512×512 (all notations are followed in Section 3.2).

Generators, Discriminators and Estimator. For the generators, it consists of bilinear upsampling and convolutional layers, following with instance normalization [44] and leaky ReLU activation [31]. The layer number varies with the size of reconstructed patch image. Similarly, we also use the convolutional layer with kernel size 3×3 and stride 2, with leaky ReLU activation to form the discriminators. For the weighting estimator, we follow the components in the discriminator. To train the generators, we use the Adam optimizer [23] with learning rate of 1e-4 and momentums set as 0.9 and 0.99. For learning the discriminators, we use the SGD with the learning rate of 1e-4 and

Methods	person	rider	car	truck	bus	train	motorbike	bicycle	mAP
Source-only	29.7	32.2	44.6	16.2	27.0	9.1	20.7	29.7	26.2
Chen *et al*. [4]	25.0	31.0	40.5	22.1	35.3	20.2	20.0	27.1	27.6
Ours-Type1(K=2;m=256;512×512)	33.6	37.5	47.8	23.1	39.2	15.2	**29.3**	**34.7**	32.6
Ours-Type2(K=4;m=128;256×256)	**33.9**	**39.7**	**49.7**	21.3	**39.4**	21.9	27.6	34.6	33.5
Ours-Type3(K=8;m=64;128×128)	33.5	38	48.5	**26.5**	39	**23.3**	28	33.6	**33.8**

Table 2. Results of domain adaptation for object detection from Cityscapes to Foggy-Cityscapes (normal → foggy).

Methods	*car* AP
Source-only	33.96
Chen *et al*. [4]	38.97
Ours-Type1(K=2;m=256;512×512)	41.97
Ours-Type2(K=4;m=128;256×256)	42.70
Ours-Type3(K=8;m=64;128×128)	**43.02**

Table 3. Results of detection adaptation from synthetic data to real-world data.

Methods	*car* AP
Source-only	37.4
Chen *et al*. [4]	38.5
Ours-Type2(K=4;m=128;256×256)	41.9
Ours-Type3(K=8;m=64;128×128)	**42.5**

Table 4. Results of cross camera adaptation from Kitti dataset to Cityscapes dataset.

the weight decay of 5×1e-4.

4. Experiments

In this section, we first provide the detailed experimental setup, then evaluate our adaptation framework on object detection under various settings, including normal-to-foggy, synthetic-to-real and cross-camera adaptation. Furthermore, extensive ablation studies are conducted to validate each component. In the end, we extend our method to the instance segmentation task to verify its scalability among region based tasks.

4.1. Experimental Setup

Our experimental setting on detection adaptation follows the setup in [4]. Several datasets are used in our experiments, including Cityscapes dataset [6], Foggy-Cityscapes dataset [38], KITTI [11] and SIM-10k dataset [22]. During training, images and annotations (bounding boxes and object categories) are provided for source domain, and only images are available for target domain. We resize the image to the shorter side of 512 pixels, and use the batch size = 1 (*i.e*., one source image and one target image) to fit the GPU memory. We set the total training epoch as 25 and $\lambda = 0.1$ and warmup strategy is used during training. For all the experiments, we report mean average precisions (mAP) with a threshold of 0.5 to evaluate different methods.

4.2. Domain Adaptation for Detection

In this section, we compare our method with the state-of-the-art domain adaptation for detection [4]. Note that other related works, are either with weakly supervised settings [21] or with class-specific settings [35], and cannot be directly compared.

Normal to Foggy. In this experiment, we use the Cityscapes dataset as our source domain, and all images and annotations in Cityscapes are used. Foggy-Cityscapes dataset is compatible with the Cityscapes among the annotations and data split. We use the training set of Foggy-Cityscapes as the target domain (only images), and report the results on the validation set of Foggy-Cityscapes. Note that we follow [4] to utilize the rectangle of instance mask in Cityscapes as the ground truth bounding boxes.

The results are reported in Table 2. Eight categories are used in the evaluation. Source-only denotes that the method is trained only using source images without domain adaptation. We report three variants of our framework described in Table 1, in which we apply three different sets of grouping hyper-parameters. From the table, it can be observed that all our methods outperform the existing method [4] with a large margin (*i.e*., about **6**% performance gain), which demonstrates the region-level adaptation could improve the detection performance under different weather conditions.

Synthetic to Real. Another domain-shift scenario is from synthetic data to real-world image. Due to the huge cost of human labeling, synthetic data offers an alternative. However, the model trained on synthetic data often suffers a significant drop on real-world data. Hence, domain adaptation from synthetic to real-world is desired and we conduct adaptation experiment under this setting to investigate the effectiveness of our method. In our implementation, SIM-10k dataset is used as the synthetic dataset, in which bounding boxes of category *Car* are provided and total 10,000 images are used in the training stage. The target domain is Cityscapes and we use its validation set for evaluation. Note that because only category *car* is used in the training, the evaluation of Cityscapes also performs on the *car*.

The results of different methods are reported in Table 3. It can be observed that all three variants of the proposed

Variants	mAP
Source-only	26.2
Ours-Type2 w/o Generators	32.8
Ours w/o Discriminators	27.9
Ours w/o Estimator	32.3
Ours-Type2(K=4;m=128;256×256)	33.5
Ours-Type3(K=8;m=64;128×128)	33.8

Table 5. Ablation studies for different components.

method achieve the much better performance than the existing methods, which consistently validates that our region-level adaptation framework does reduce the domain shift. More specifically, our method with Type3 (8 clusters) obtains **+4**% performance gain compared with the state-of-the art model [4], and **+9**% gain compared with our baseline method (*i.e.*, Source-only).

Cross Camera Adaptation. Different camera setups widely exist in autonomous driving dataset. We thus perform the cross camera adaptation from Kitti dataset to Cityscapes dataset. Since the image size in Kitti dataset is 1250 × 375, inconsistent with Cityscapes dataset, we resize the image from Cityscapes so that the shorter length is 375 pixels long. Due to the shorter size 375, Ours-Type1 variant (region-size of 512 × 512 is larger than 375) is not conducted on this adaptation experiment. For the evaluation, we only report the common category, *car*, among two datasets. The results are reported in Table 4. It can be observed that our proposed methods consistently achieve better performance over other baselines. Specifically, our methods obtain a performance gain of about 4% than [4].

4.3. Ablation Studies

In this section, we perform the thorough ablation experiments to investigate the effect of different components in our method, the effect of region-level alignment and the effect of the designed ROI-based grouping strategy. These experiments demonstrate different contributions of components and provide more insights of our proposed method.

Effect of Different Components. In this part, we design several variants of our model to validate the contributions of different components. These variants are shown as follows:

- Ours w/o Generators; we remove generators and perform adversarial alignment in region feature level.
- Ours w/o Discriminators; we perform the adaptation by jointly training the detectors and reconstructing region-level RGB images.
- Ours w/o Weighting Estimator: Weighting estimator is removed, thus all target regions are treated equally.

The results are reported in Table 5. All experiments are conducted on the adaptation from Cityscapes to Foggy-Cityscapes. We use the Type2 to setup the grouping strategy. It can be observed that removing the discriminators (Ours w/o Discriminators) gets much worse compared with our full model, which indicates the discriminators do help align the distributions. The performance of Ours-Type2 w/o Generators (mAP=32.8%) demonstrates that the feature-level alignment of patches also achieves the adaptation, while reconstruction of the regions further makes it learn the structural information of image patches and aids the alignment. The designed Weighting estimator also yields an improvement, which validates our conjecture that biased proposals might introduce incorrect guidance and the estimator could alleviate this impact.

Methods	mAP
Source-only	26.2
Ours-Global(1024×512)	29.8
Ours-Type1(K=2;m=256;512×512)	32.6
Ours-Type2(K=4;m=128;256×256)	33.5
Ours-Type3(K=8;m=64;128×128)	33.8

Table 6. Ablation studies for image-level *v.s.* region-level adaptation. Ours-Global denotes the image-level adaptation.

Figure 4. We show two examples from the fixed-location strategy and the ROI-based grouping strategy, respectively. We display two types of strategies with cluster number $K = 2$ (top two figures) and $K = 4$ (bottom two figures).

Image-level *v.s.* Region-level. To verify the effectiveness of the region-level adaptation, we compare our region-level alignment to the whole-image alignment. In this experiment, we design the variant of Ours-Global, which utilizes the features extracted from the backbone network to reconstruct the whole image and perform the domain adversarial alignment on the image level. The ablation is conducted on the experiment from Cityscapes to Foggy-Cityscapes. We report the results in Table 6. It can be found that our region-level adaptation methods (Type1, Type2 and Type3) do achieve the better performance and yield an improvement of about 4% over the image-level adaptation method. The results demonstrate that our region-level alignment better matches the nature of detection task and is more effective than global statistics alignment for the detection adaptation.

Methods	K	m	size	mAP
Source-only	-	-	-	26.2
Ours-Fixed1	2	256	512 × 512	31.5
Ours-Type1	2	256	512 × 512	32.6
Ours-Fixed2	4	128	512 × 256	30.4
Ours-Type2	4	128	256 × 256	33.5

Table 7. Experimental results of different strategies. Ours-Fixed1 and Fixed2 are the methods using the fixed location strategy.

Fixed Location Strategy *v.s.* ROI-based Grouping Strategy. In this experiment, we perform an ablation study for investigating the effect of the designed ROI-based grouping strategy. Since our grouping strategy aims to dynamically mine the regions which are more likely to contain the objects of interest, to verify its effectiveness, we design a fixed location strategy, which extracts the regions from fixed locations without considering the guidance of the region proposals, as the competitor.

We show examples of the fixed location strategy on the left part of Figure 4, and the right part is the example of the ROI-based grouping strategy. More specifically, the regions used in the fixed location strategy are uniformly divided. For the feature representations of region proposals (*i.e.*, features before ROI-Pooling.), we also split them into several parts based on the center coordinates of region proposals, *i.e.*, if the center is located on the left region, then its feature will be applied to the reconstruction of left part. The detailed setting of fixed location strategy one (named Ours-Fixed1) is K=2, m=256 and region-size of 512 × 512, and Ours-Fixed2 is K=4, m=128 and region-size of 512 × 256 (Note that we follow the notations in Table 1). As shown in Figure 4, in the fixed location strategies, all patches make up the whole image in order to cover all possible objects.

The results are reported in Table 7. It can be observed that under the same setting of cluster (Our-Fixed1 *v.s.* Our-Type1 and Our-Fixed2 *v.s.* Our-Type2), our grouping strategy achieves better performance compared to the fixed location strategy. Specifically, our ROI-based grouping strategy yields an improvement of 3% using Ours-Type2 compared with Ours-Fixed2, which demonstrates that the fixed location strategy may introduce erroneous bias during adaptation because it already contains significantly different field of view, while our grouping strategy is able to focus on the desired regions under the guidance of region proposals.

4.4. Domain Adaptation for Instance Segmentation

In this section, we conduct the proposed domain adaptation framework on instance segmentation task to verify the scalability of our model. Since the instance segmentation is a region based task (the ground-truth bounding box is given during training), it is also a good choice to investigate the effectiveness of the region-level adaptation.

In our implementation, we extend the proposed adaptation framework to the instance segmentation task by adding the mask branch and ROI-Align described in Mask R-CNN [18]. For the adaptation part, we inherit the setting in Table 1 for the grouping strategy. The adaptation experiment from Cityscapes to Foggy-Cityscapes is conducted because these datasets contain both bounding boxes and mask annotations for the instance segmentation task. We evaluate the performance on the validation set of Foggy-Cityscapes and report the mAP with a threshold of 0.5.

Methods	K	mAP (Box)	mAP (Mask)
Faster R-CNN [37]	-	26.2	-
Source-only	-	32.8	26.6
Ours-Type1	2	37.1	30.8
Ours-Type2	4	**38.4**	**31.4**
Ours-Type3	8	37.5	30.9

Table 8. Experimental results of the domain adaptation for instance segmentation. The adaptation is from Cityscapes to Foggy-Cityscapes. Faster R-CNN is trained on source domain with the ground truth bounding box only, thus we just report the mAP of bounding box.

We report the results in Table 8. By comparing the Faster R-CNN with Source-only, it can be found that introducing the mask branch improves the mAP of bounding box. In addition, our adaptation framework further yields an improvement of **5.6**% on mAP (Box) and **4.8**% on mAP (Mask) over the Source-only method. Moreover, all three variants of our method consistently achieve much better results, which indicates the effectiveness of the region-level adaptation in the region based task and its good scalability.

5. Conclusion

In this paper, we have proposed a region-level adaptation framework for object detection. We follow the local nature of detection to reposition the focus of adaptation process, from global to local. Two key components, *region mining* and *adjusted region-level alignment*, are designed to address the questions of "where to look" and "how to align", effectively and robustly. We conduct extensive experiments and ablation studies, which demonstrate our model achieves state-of-the-art on various domain-shift settings and keeps good scalability on other region-based task, such as instance segmentation.

Acknowledgement

This work is partially supported by the Collaborative Research grant from SenseTime Group (CUHK Agreement No. TS1712093), the Early Career Scheme (ECS) of Hong Kong (No. 24204215), and the General Research Fund (GRF) of Hong Kong (No. 14236516 & No. 14203518).

References

[1] Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, and Mario Marchand. Domain-adversarial neural networks. *arXiv preprint arXiv:1412.4446*, 2014.

[2] 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.

[3] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, and Dahua Lin. Hybrid task cascade for instance segmentation. *arXiv preprint arXiv:1901.07518*, 2019.

[4] 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*, pages 3339–3348, 2018.

[5] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *CVPR*, pages 7892–7901, 2018.

[6] 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*, pages 3213–3223, 2016.

[7] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, pages 379–387, 2016.

[8] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *ICCV*, pages 2960–2967, 2013.

[9] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189, 2015.

[10] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

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

[12] Bo Geng, Dacheng Tao, and Chao Xu. Daml: Domain adaptation metric learning. *IEEE Transactions on Image Processing*, 20(10):2980–2989, 2011.

[13] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *ECCV*, pages 597–613. Springer, 2016.

[14] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[16] Boqing Gong, Yuan Shi, Fei Sha, and Kristen Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *CVPR*, pages 2066–2073. IEEE, 2012.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2980–2988. IEEE, 2017.

[19] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017.

[20] 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.

[21] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. *arXiv preprint arXiv:1803.11365*, 2018.

[22] Matthew 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? *arXiv preprint arXiv:1610.01983*, 2016.

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

[24] Brian Kulis, Kate Saenko, and Trevor Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *CVPR*, pages 1785–1792, 2011.

[25] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *NIPS*, pages 469–477, 2016.

[26] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016.

[27] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[28] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *ICML*, pages 97–105, 2015.

[29] Hao Lu, Lei Zhang, Zhiguo Cao, Wei Wei, Ke Xian, Chunhua Shen, and Anton van den Hengel. When unsupervised domain adaptation meets tensor representations. In *ICCV*, volume 2, 2017.

[30] Yuexin Ma, Xinge Zhu, Sibo Zhang, Ruigang Yang, Wenping Wang, and Dinesh Manocha. Trafficpredict: Trajectory prediction for heterogeneous traffic-agents. In *AAAI*, 2019.

[31] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.

[32] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

[33] Saeid Motiian, Marco Piccirilli, Donald A Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *ICCV*, volume 2, 2017.

[34] Jiangmiao Pang, Kai Chen, Jianping Shi, Huajun Feng, Wanli Ouyang, and Dahua Lin. Libra r-cnn: Towards balanced learning for object detection. In *CVPR*, 2019.

[35] Anant Raj, Vinay P Namboodiri, and Tinne Tuytelaars. Subspace alignment based domain adaptation for rcnn detector. *arXiv preprint arXiv:1507.05578*, 2015.

[36] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016.

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[38] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, pages 1–20, 2018.

[39] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *CVPR*, 2018.

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

[41] Ruoqi Sun, Xinge Zhu, Chongruo Wu, Chen Huang, Jianping Shi, and Lizhuang Ma. Not all areas are equal: Transfer learning for semantic segmentation via hierarchical region selection. In *CVPR*, 2019.

[42] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018.

[43] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, volume 1, page 4, 2017.

[44] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[45] Ceyuan Yang, Zhe Wang, Xinge Zhu, Cheng Huang, Jianping Shi, and Dahua Lin. Pose guided human video generation. In *ECCV*, 2018.

[46] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *ICCV*, volume 2, page 6, 2017.

[47] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017.

[48] Xinge Zhu, Zhichao Yin, Jianping Shi, Hongsheng Li, and Dahua Lin. Generative adversarial frontal view to bird view synthesis. *2018 International Conference on 3D Vision (3DV)*, pages 454–463, 2018.

[49] Xinge Zhu, Hui Zhou, Ceyuan Yang, Jianping Shi, and Dahua Lin. Penalizing top performers: Conservative loss for semantic segmentation adaptation. In *ECCV*, pages 587–603. Springer, 2018.