electronics

MDPI

Article

AdvMix: Adversarial Mixing Strategy for Unsupervised Domain Adaptive Object Detection

Ruimin Chen [1,2,3], Dailin Lv [1,2], Li Dai [1,2], Liming Jin [1,2,*] and Zhiyu Xiang [3]

[1] Zhejiang Geely Holding Group Co., Ltd., Hangzhou 310051, China; chenruimin@mail.sitp.ac.cn (R.C.); dl381@sussex.ac.uk (D.L.); li.dai4@geely.com (L.D.)
[2] Zhejiang Green Intelligent Vehicle and Spare Parts Technology Innovation Center, Ningbo 315336, China
[3] College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China; xiangzy@zju.edu.cn
* Correspondence: liming.jin@geely.com

Abstract: Recent object detection networks suffer from performance degradation when training data and test data are distinct in image styles and content distributions. In this paper, we propose a domain adaptive method, Adversarial Mixing (AdvMix), where the label-rich source domain and unlabeled target domain are jointly trained by the adversarial feature alignment and a self-training strategy. To diminish the style gap, we design the Adversarial Gradient Reversal Layer (AdvGRL), containing a global-level domain discriminator to align the domain features by gradient reversal, and an adversarial weight mapping function to enhance the stability of domain-invariant features by hard example mining. To eliminate the content gap, we introduce a region mixing self-supervised training strategy where a region of the target image with the highest confidence is selected to merge with the source image, and the synthesis image is self-supervised by the consistency loss. To improve the reliability of self-training, we propose a strict confidence metric combining both object and bounding box uncertainty. Extensive experiments conducted on three benchmarks demonstrate that AdvMix achieves prominent performance in terms of detection accuracy, surpassing existing domain adaptive methods by nearly 5% mAP.

Keywords: object detection; domain adaption; adversarial learning; self-training

Citation: Chen, R.; Lv, D.; Dai, L.; Jin, L.; Xiang, Z. AdvMix: Adversarial Mixing Strategy for Unsupervised Domain Adaptive Object Detection. *Electronics* **2024**, *13*, 685. https://doi.org/10.3390/electronics13040685

Academic Editor: Ying Tan

Received: 29 December 2023
Revised: 5 February 2024
Accepted: 5 February 2024
Published: 7 February 2024

1. Introduction

Object detection [1–3] aims to locate and classify the targets in the given image, which has received significant attention in computer vision recently. With the emergence of deep feed-forward architectures [4–6], modern data-driven detection methods [1–3,7–10] lead to considerable improvements in many applications, including security surveillance, autonomous driving, and so on. However, those achievements are obtained only when test data and training data maintain the same distribution. Therefore, the severe performance degradation is inevitable once the domain shift [11–14] occurs due to diverse layouts, illuminations, viewpoints, and weather conditions. One feasible solution is re-training the detector with large-scale labeled data to improve the detection accuracy in new scenes. Although this solution is effective, annotating images is a time-consuming and expensive project [15], which limits the practicality of this method.

In view of above problems, recent researchers have focused their efforts on Unsupervised Domain Adaption (UDA) methods [16–22], which leverages unsupervised transfer learning to alleviate the domain gaps. UDA methods transfer knowledge from the label-rich source domain to the target domain without tedious manual annotations. With joint training of both source and target domain data, the goal of UDA is to generate a detector that performs well on the target domain.

The majority of UDA methods handle domain adaptation in an adversarial manner. The domain classifier is exploited to identify whether the image is from source or target domain

and the gradient reversal layer [23] is introduced in the classifier to maximum the domain discrimination loss. Recent works in UDA have shown that the domain-invariant features can be learned by the detector when employing multi-granularity adversarial alignment [14,24–27] including instance-level and pixel-level feature consistency. One limitation of these methods is that the domain gaps affect performance seriously. For example, these methods obtain remarkable accuracy when trained on paired images with different styles, but suffer from degradation on cross-camera adaptation tasks. To this end, some approaches bridge the domain difference through image translation. A series of image style transfer technologies [14,26,28,29] are implemented to convert images from the source domain to target domain. However, extra complex parameters in the transfer structure hinder the convergence of detectors and make the training process more difficult.

Different from the aforementioned UDA methods, an alternative solution is to finetune the network with pseudo labels generated by the source-trained model. To some extent, the quality of pseudo labels are tightly related to the detection precision. To avoid the noise of the pseudo labels, many novel self-training optimizations [13,19,20,22,30–32] are proposed, including knowledge distillation strategy [19], the progressive confidence restriction [13], imbalanced mini-batch sampling strategy [20], and graph representation [22,32]. Although self-training strategy is an efficient way to boost performance, one shortcoming of these methods is that classification confidences are mostly used as the prediction box selection criteria. Such criteria omits the uncertainty of bounding boxes and the pseudo labels fails to represent precise localization. Therefore, how to select reliable detection boxes is a critical problem to be invested in. With regards to reliability of pseudo labels, most of the self-training methods introduce novel confidence metrics, such as uncertainty-based pseudo labeling [20] and sample mixing technique [13]. However, the detector pre-trained on source domain plays an essential role during self-training process and the above tricks hardly works when the detector is overfitted on the source domain.

Generally speaking, existing UDA approaches are designed for diminishing domain gaps from two perspectives: style gap and content gap [20]. The style gap demonstrates the difference of image styles, such as color, brightness, and overall layout. On the other hand, the content gap contains more instance information, such as distinct distributions, densities, and sizes of objects. We observe that adversarial methods are the expert in the style gap while self-training strategies show outstanding potential in the context gap.

Inspired by previous works, we address domain adaptive object detection from a comprehensive perspective. Our approach, named Adversarial Mixing (AdvMix), is a solution to reduce both style gap and content gap simultaneously in the training phase. With regards to the style gap, the source and target distributions are aligned in feature spaces by the Adversarial Gradient Reversal Layer (AdvGRL). For the content gap, AdvMix introduces a sample synthesis strategy by artificially mixing the region of the paired source image and target image. As illustrated in Figure 1, our network consists of three parts: detector, AdvGRL and region mixing module.

To be specific, we apply one-stage detection architecture, YOLOv5 [33], as the detector to meet the requirement of real-time processing. The AdvGRL is designed as a domain discriminator to generate a compact domain descriptor. The gradients of the discriminator are reversed during the back propagation to reduce domain style shifts. Different from pixel-level gradient reversal layers in the literature [14,27,34], our proposed AdvGRL concentrates on global styles from multi-scale features. Furthermore, most methods omit the diversity of training samples and all samples are treated as evenly, which hinders the model learning on challenging scenarios. To address this problem, our AdvGRL designs an adversarial weight mapping function to mine hard examples and enhance the stability of domain-invariant features. For the region mixing module, the mixed image is synthesized from the source image and a region of the target image with reliable pseudo predictions. We also feed the synthesis image into the detector and its predictions are self-supervised by the consistency loss. To select trustworthy pseudo detections, a strict confidence metric including both object and bounding box confidence is exploited in this paper.

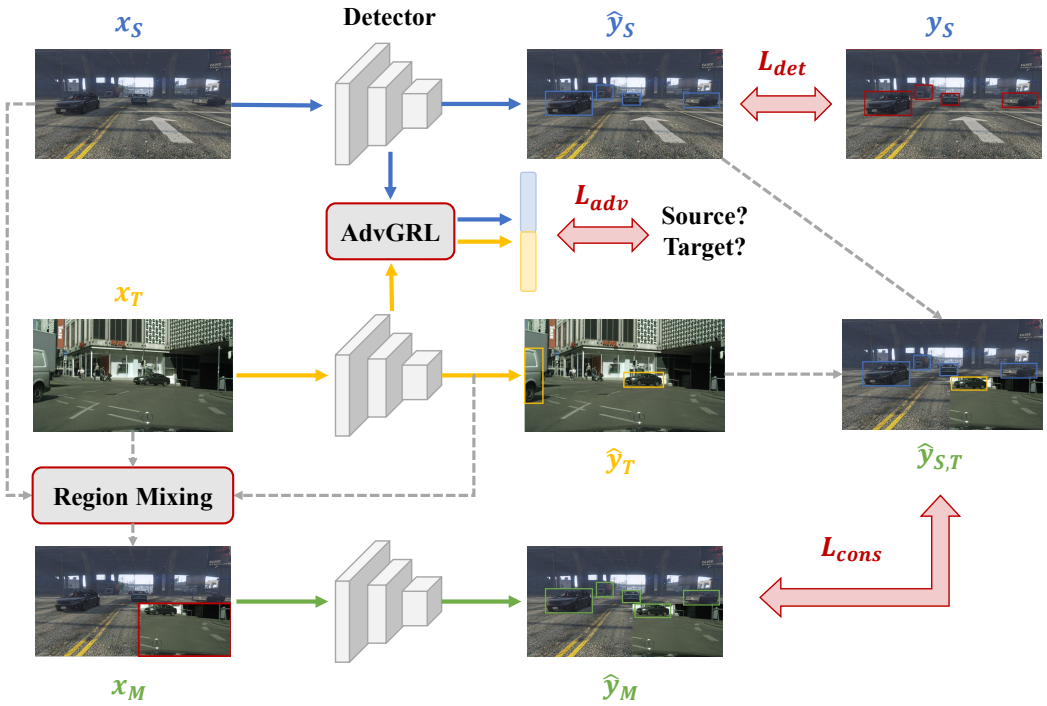


Figure 1. The overall framework of AdvMix, which mainly consists of three parts: the detector to predict detection results, the AdvMix to maximum the domain discrimination loss, and the region mixing module to address the context gap of different domains.

To validate the performance of AdvMix, we conduct extensive experiments on three domain adaptation scenarios, namely cross-weather (Cityscapes [35] → FoggyCityscapes [36]), cross-camera (KITTI [37] → Cityscapes [35]), and synthetic-to-real (Sim10K [15] → Cityscapes [36]). Experimental results show that AdvMix boosts detection accuracy in comparison with other state-of-the-art UDA methods. In addition, we discuss how our proposed AdvMix eliminates the style gap and content gap by visualization analysis, which explains why AdvMix works well in the domain adaption.

Overall, the main contributions of this paper are summarized as follows:

- We propose an unsupervised cross-domain adaptive method, AdvMix, which is a joint adversarial feature alignment and region mixing self-training strategy to reduce style and content gaps simultaneously;
- To address the style gap, a novel AdvGRL is introduced to align global image styles from multi-scale feature maps and enhance the stability of domain-invariant features by hard examples mining;
- To diminish the content gap, we employ a self-supervised training strategy based on region mixing and design a strict confidence metric to improve the reliability of self-training.

The structure of this paper is as follows. In Section 2, we briefly present related works, including object detection and domain adaption. The details of the proposed method AdvMix are described in Section 3. Extensive experiments are conducted in Section 4 and ablation studies are drawn in Section 5. We discuss multi-class domain adaption and the failure cases in Section 6. Finally, Section 7 presents the summary of this paper.

2. Related Work

In this section, we firstly describe mainstream object detection frameworks including both two-stage and one-stage detectors. Besides, we describe recent UDA strategies, which are closely related to our method.

2.1. Object Detection

Object detection, as a core problem in computer vision, predicts both classification labels and bounding box coordinates at the same time. Recent detection frameworks [1–3,7,8,10] have obtained outstanding improvement with the advance of deep learning. According to design principles of frameworks, most detectors are roughly divided into two-stage and one-stage approaches. Two-stage detectors first generate the region proposals and then refine the object detection results by the region selection such as Region Proposal Network (RPN). A typical example of two-stage frameworks is Faster-RCNN [1] and it achieves impressive performance in terms of detection accuracy. However, one limitation of these two-stage methods is the inference speed. To lessen the computation burden, a series of one-stage frameworks are proposed, such as YOLO [2], SSD [7], and FCOS [8]. Instead of excessive proposal generation, one-stage detectors directly produce object labels and regress locations by leveraging pre-defined bounding box candidates (anchors). The focal loss [38] is proposed to address the imbalance between foregrounds and backgrounds, which is beneficial for detection precision. To reduce duplicated locations for the same instance, an additional Non-maximum Suppression (NMS) [39] is applied to filter exhaustive predictions generated by the detector. Furthermore, some one-stage algorithms such as FCOS [8], CenterNet [3], CornorNet [40], and ExtremeNet [10] regard the detection problem as a key-point estimation and omit pre-defined anchors to further improve efficiency. Despite their success in real-time detection, these methods are hardly employed in some high-precision application scenarios due to excessive missed detections. In this paper, we select YOLOv5 [33] as our main detection framework to seek a balance between speed and precision.

2.2. Unsupervised Domain Adaptation

The changing environments lead to domain shift, where the distributions of training data and test data are distinct. As a result, deep neural networks suffer from performance drop and a series of UDA methods [16–22] are proposed to solve this problem. UDA methods leverage both the labeled source data and unlabeled target data to reduce the discrepancy between source and target. Early works minimizes the domain gap by manual alignment, such as Maximum Mean Discrepancy (MMD) minimization [41] and subspace alignment [42]. With the development of end-to-end training process, adversarial UDA approaches [23,43,44] attract the interest of researchers and the gradient reverse layer [23] is introduced to extract domain-invariant feature. Although these methods achieve remarkable improvement on classification adaption, the domain adaption for object detection is still a challenging issue because the object detection is a comprehensive task involving bounding box regression.

Predominant domain adaption for object detection consists of adversarial and self-training methods. The former utilizes adversarial training to learn the domain-invariant features by fooling the domain classifier. Chen et al. [24] first employ both image-level and instance-level gradient reverse in Faster-RCNN framework. Following that, more multi-granularity feature alignment strategies [14,17,25–27] are proposed to reduce the domain discrepancy, such as selective region-level alignment [25] to focus on objects of interest and category-level adaption [27] to learn category-wise representations. To enhance the robustness of adversarial approaches, some image translation modules [14,26,28,29] are designed. For example, Hsu et al. [26] and Li et al. [14] introduce an intermediate domain and a weighted distance loss is added during the process of adversarial training. However, Yu et al. [20] notice that the adversarial alignments mainly account for image style gaps and overlook the domain shift from object density distribution. Some researchers tackle this problem by self-training the detector with robust pseudo detection results [13,19,20,30]. To address the noisy pseudo labels, some methods employ some extra confidence metrics, such as uncertainty-based fusion [20] or self-entropy descent [30]. Furthermore, a novel training strategy is selected for reliable pseudo predictions, including student-teacher framework based knowledge distillation [19,31] and forming mixed samples combined by both source and target images [13].

3. Method

In this section, we present our proposed domain adaptive method AdvMix in detail. First, the overall framework of AdvMix is introduced. Then, we describe two core components (AdvGRL and region mixing module), respectively. Finally, we show the training details of AdvMix, including the loss functions and how to train the whole network.

3.1. Overall Framework

The overall framework of AdvMix is illustrated in Figure 1 and the network input contains three different images, including a labeled source image $x_S \in R^{H\times W\times C}$, an unlabeled target images $x_T \in R^{H\times W\times C}$, and a synthetic image $x_M \in R^{H\times W\times C}$ mixed by the local regions of x_S and x_T. The detector of AdvMix is a detection network with trainable parameters and we adopt lightweight one-stage detector YOLOv5 [33] in this paper. The overall process can be divided into three steps, as follows.

First, x_S and x_T are fed into the detector simultaneously and the detection results are named as $\hat{y}_S$, $\hat{y}_T$. The source prediction $\hat{y}_S$ is supervised by the source label y_S while the target prediction $\hat{y}_T$ is viewed as the pseudo detection label. At the same time, the features of two domains are aligned by AdvGRL, which is a domain discriminator to maximize the domain discrimination loss and perform adversarial hard example mining. Second, we evenly separate the target image x_T into four regions. A region of x_T with the highest pseudo detection confidence is merged with the source image x_S to form the mixed image x_M. The pseudo label of x_M, named as $\hat{y}_{S,T}$, is synthesized by the source detection $\hat{y}_S$ and target pseudo detection label $\hat{y}_T$ according to the image clipping strategy. Finally, x_M passes through the detector to produce a prediction $\hat{y}_M$. The self-supervision of the mixed image is achieved by the consistency loss between its prediction $\hat{y}_M$ and its pseudo label $\hat{y}_{S,T}$.

3.2. AdvGRL

The AdvGRL in this paper is constructed by a domain discriminator to classify whether the input sample belongs to source domain or target domain. To be specific, it produces the domain prediction during forward propagation and the gradients of AdvGRL are reversed to the base detection network in the progress of back propagation. The reversal gradients confuse the domain discriminator and thus domain-invariant features are obtained by the base detection network. Most methods related to gradient reverse [14,27,34] adopt stacked convolutions as fusion layers to generate pixel-level domain features, as shown in Figure 2b. Those methods are always combined with instance-level discriminators in two-stage detection frameworks such as Faster-RCNN to achieve the domain adaption. For one-stage detector, we proposed a novel global-level discriminator with both convolution and Full Connection (FC) layers, as presented in Figure 2a. Compared to full conventional structures, the compact domain descriptor generated from FC reflects the global style. Comparative experiments conducted in Section 5 show the effect of our proposed structure.

In the process of forward propagation, we employ binary cross loss for the domain discriminator D. The domain discrimination loss L_D is computed as Equation (1).

$$L_D = -\sum_{i=1}^{N}\Big[y_i \log D(F_i^S) + (1-y_i)\log(1-D(F_i^T))\Big] \tag{1}$$

where $i \in \{1,\ldots,N\}$ denotes the i-th image. F_i^S and F_i^T are the features extracted from the i-th image in the source domain and target domain, respectively. The domain y_i is 1 if the feature is from the source domain and 0 otherwise.

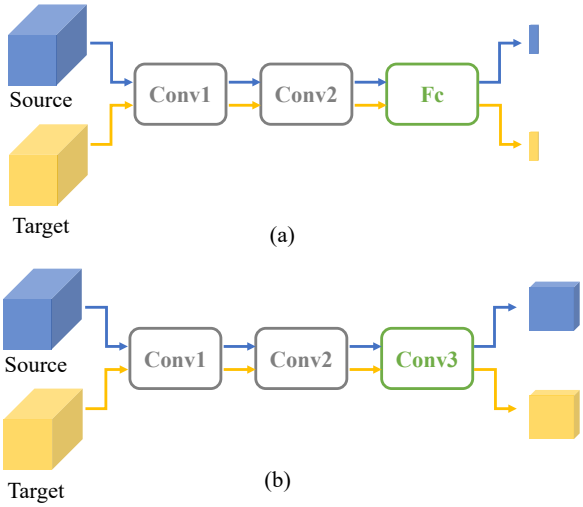


Figure 2. The network structures of different domain discriminators. (**a**) Global-level discriminator proposed in this paper (AdvGRL); (**b**) pixel-level discriminator in the literature.

Furthermore, an adversarial weight mapping function is proposed to address the diversity of training samples. The core of the mapping function is to weigh the different samples according to the domain discrimination loss L_D. For the samples with low L_D, the discriminator can identify them easily while their domain-invariant features are hardly collected by the base detection network. Therefore, they are viewed as hard examples in domain adaption and assigned high loss weights during the training. To be specific, the adversarial weight λ_{adv} is written as follows:

$$\lambda_{adv} = max\left\{\lambda_0, 2 - \frac{2}{1 + e^{-\beta \cdot L_D}}\right\} \tag{2}$$

where λ_0 is a lower-bound of the weight and we set $\lambda_0 = 0.01$ in this paper. In addition, β denotes the scaling threshold of the mapping function. Figure 3 visualizes the relation between adversarial weight λ_{adv} and the loss L_D, where $\beta = 10$.

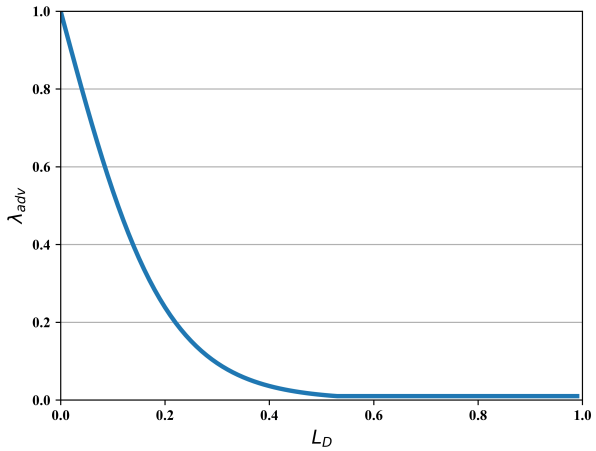


Figure 3. The visualization of the adversarial weight mapping function.

Apart from the structure of AdvGRL, where to insert the AdvGRL is another issue that should be considered. As shown in Figure 4, one-stage detector YOLOv5 [33] is composed of three main blocks, including a backbone for feature extraction, a neck for multi-scale feature fusion and a head to generate detection predictions. Furthermore, the bounding boxes with three scales are predicted separately to prevent missed detections. Following the design of YOLOv5, we add three AdvGRLs in the neck of the base detection network,

as presented in Figure 4. As a result, the style gap between source and target domain can be diminished from the different levels.

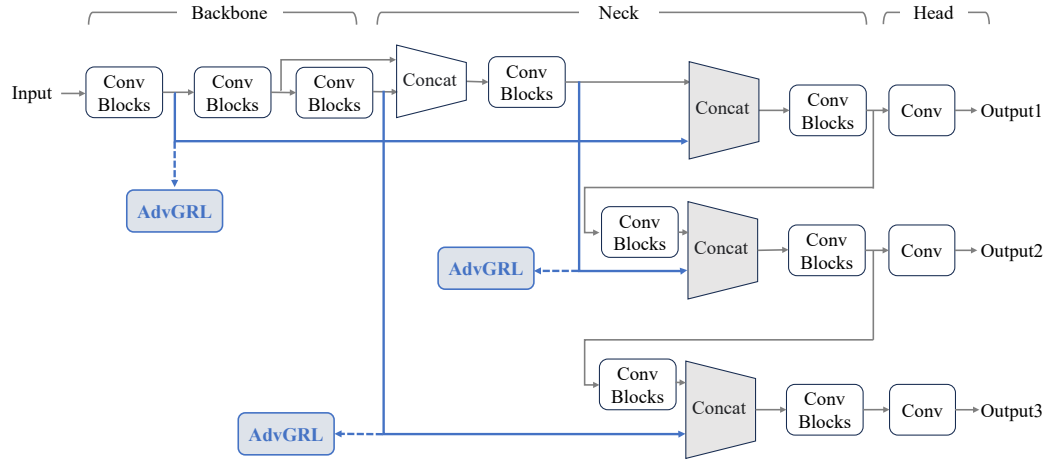


Figure 4. Insertion positions of AdvGRL in YOLOv5.

In summary, the overall adversarial loss L_{adv} conducted by AdvGRL consists of three levels $j \in \{1, 2, 3\}$, as illustrated in Equation (3).

$$L_{adv} = \sum_{j} \lambda_{adv}^{j} \cdot L_{D}^{j} \tag{3}$$

where L_{D}^{j} denotes domain discrimination loss of the j-th discriminator and its weight λ_{adv}^{j} is calculated according to the Equation (2).

3.3. Region Mixing Module

The region mixing module is designed for generating the synthetic image x_M and its pseudo label $\hat{y}_{S,T}$. The target image x_T is divided into four equal regions and a region with the highest average detection confidence is selected, named as R. x_M can be formed by the target image in the selected region x_T^R and the source image out of the selected region x_S^{R-}, as shown in Equation (4).

$$x_M = \left\{ x_T^R, x_S^{R-} \right\} \tag{4}$$

Similarly, the pseudo label $\hat{y}_{S,T}$ consists of two parts: the target prediction in the selected region $\hat{y}_T^R$ and the source detection out of the selected region $\hat{y}_S^{R-}$, as presented in Equation (5). Note that all bounding boxes are clipped by their corresponding region boundaries before being added into $\hat{y}_{S,T}$.

$$\hat{y}_{S,T} = \left\{ \hat{y}_T^R, \hat{y}_S^{R-} \right\} \tag{5}$$

The core problem of region mixing module is how to select trustworthy pseudo predictions, which is significant for the stability of self-training. However, the original confidence score C_{obj} in YOLOv5 [33] reflects whether it contains the object and omits localization uncertainty. Inspired by Gaussian-based detectors [13,45], we introduce the Gaussian-based bounding box $\mathbf{b} = \left[\mathbf{b}_\mu, \mathbf{b}_\Sigma \right]$, where the former denotes the means $\mathbf{b}_\mu = \left[\mu_{bx}, \mu_{by}, \mu_{bh}, \mu_{bw} \right]$ and the later denotes the variance $\mathbf{b}_\Sigma = \left[\Sigma_{bx}, \Sigma_{by}, \Sigma_{bh}, \Sigma_{bw} \right]$. The bounding box confidence is calculated as follows:

$$C_{bbx} = 1 - \text{mean}(\mathbf{b}_\Sigma) \tag{6}$$

The combined confidence C_{comb} accounts for both object confidence C_{obj} and bounding box confidence C_{bbx}, as presented in Equation (7). It is notable that the values of both C_{obj} and C_{bbx} are ranged from 0 to 1. Only the pseudo predictions with $C_{comb} > 0.25$ are accounted as valid pseudo labels.

$$C_{comb} = C_{obj} \cdot C_{bbx} \tag{7}$$

3.4. Training Algorithm

During the forward propagation, the total loss L_{total} of AdvMix is a combination of three terms: the detection loss with ground-truth supervision L_{det}, the adversarial loss L_{adv} in Equation (3), and self-supervised consistency loss L_{cons}.

$$L_{total} = L_{det} + L_{adv} + L_{cons} \tag{8}$$

Specifically, L_{det} is designed for penalizing the difference between source prediction $\hat{y}_S$ and source label y_S. We adopt the standard loss of Gaussian-based YOLOv5 [45] as L_{det}, which contains object score loss L_{obj}, classification loss L_{cls}, and bounding box regression loss L_{bbx}.

For the synthetic image x_M, the similarity between its prediction $\hat{y}_M$ and its label $\hat{y}_{S,T}$ is computed as the consistency loss L_{cons}. In this paper, L_{cons} shares the same loss function with L_{det}, which reflects the prediction precision for the input image. However, the supervision in L_{cons} is the pseudo label $\hat{y}_{S,T}$ instead of the ground-truth label y_S in L_{det}.

To improve the reliability of the self-supervised training, we employ a variable weight λ_{cons} for L_{cons}. The λ_{cons} shows the ratio of $\hat{y}_{S,T}$ with combined confidence C_{comb} greater than the predefined threshold value C_{th}, as presented in Equation (9).

$$\lambda_{cons} = \frac{|\hat{y}^k_{S,T} : C^k_{comb} > C_{th}|}{|\hat{y}_{S,T}|} \tag{9}$$

where k is the k-th prediction of $\hat{y}_{S,T}$ and $|\cdot|$ denotes the cardinality of a set.

The training algorithm of AdvMix can be concluded as three steps. First, the detection network is initialized with COCO [46] pretrained parameters and other modules are randomly initialized. The second step is the forward propagation, where the paired source and target images are fed into the network and the total loss L_{total} is calculated according to Equation (3). Finally, all parameters of the network are updated in the back propagation. It is notable that the proposed AdvGRL and region mixing module only works in the training process and we just employ the detection network in the inference.

4. Experiments

In this section, we first introduce experimental data and implementation details Then, extensive experiments are conducted on different domain adaptation scenarios and the detection results are analyzed in this section. Finally, we compare our proposed AdvMix with some state-of-the-art UDA methods to evaluate the performance of AdvMix.

4.1. Dataset and Experimental Setup

In this section, we adopt four different datasets, including Cityscapes [35], FoggyCityscapes [36], KITTI [37], and Sim10K [15].

Cityscapes [35] is a diverse image set recorded in real-world urban scenarios, including eight categories: person, car, train, rider, truck, motorcycle, bicycle, and bus. To capture the complexity of inner-city traffic scenes, all images are manually selected from 50 cities with a different foreground, background, and layout, as shown in Figure 5. The annotation images are split into 2 parts: 2975 images for training and 500 images for testing.

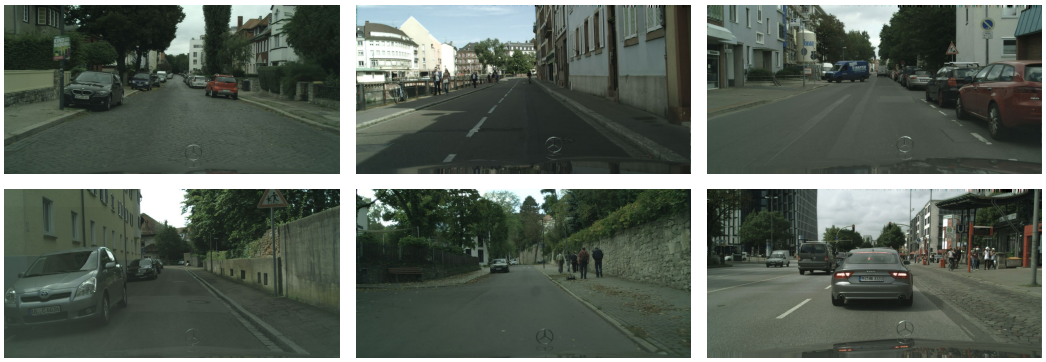


Figure 5. Samples of the Cityscapes dataset.

FoggyCityscapes [36] is a synthetic dataset for foggy scene understanding. The images on Cityscapes are transferred from clear-weather scenes to foggy counterparts according to the fog simulation pipeline. FoggyCityscapes, as an extension of Cityscapes, shares the same object categories and locations with Cityscapes. In addition, the partition of training and testing set in FoggyCityscapes is consistent with Cityscapes. Figure 6 presents some samples of FoggyCityscapes, which is a challenging dataset for object detection under adverse weather conditions.

Figure 6. Samples of the FoggyCityscapes dataset.

KITTI [37] collects 7481 realistic images with accurate annotations provided for 8 categories, including car, pedestrian, person sitting, cyclist, van, truck, tram, and misc. All images in KITTI are acquired via high-resolution cameras mounted on the vehicle and rectified to the resolution of 1240 × 376. Figure 7 shows some samples of the KITTI dataset, whose collection scenes involves both urban streets and highways.

Figure 7. Samples of the KITTI dataset.

Sim10K [15] contains 10,000 photo-realistic computer images from the sophisticated simulation engine, where only the "car" category is annotated. A range of images in different weather and lighting conditions are collected in this dataset, as illustrated in Figure 8. Al-

though Sim10K captures the diversity of real appearance, the difference between simulation and real-world data can be witnessed.

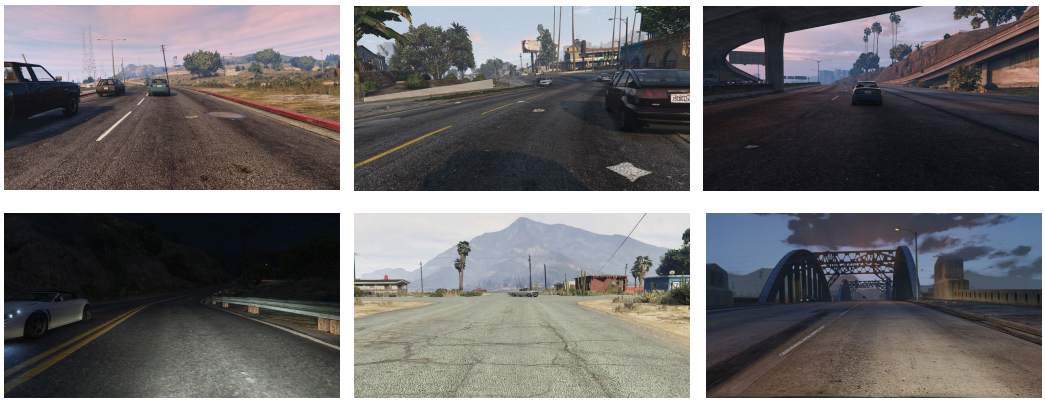


Figure 8. Samples of the Sim10K dataset.

In this section, we conduct experiments on three domain adaptation benchmarks proposed in Refs. [13,27,47], namely cross-weather (Cityscapes → FoggyCityscapes), cross-camera (KITTI → Cityscapes), and synthetic-to-real (Sim10K → Cityscapes). For the cross-weather adaptation, the training set contains both labeled Cityscapes training images (source domain) and unlabeled FoggyCityscapes training images (target domain). Furthermore, all objects with 8 categories are evaluated on FoggyCityscapes testing set. For the cross-camera adaption, we use labeled KITTI data combined with unlabeled Cityscapes training data as the training set and detection performance is validated on Cityscapes testing set. It is notable that only the "car" class is considered in this benchmark due to different categories between Sim10K and Cityscapes. The synthetic-to-real follows the same setting as the cross-camera task, except we adopt all Sim10K images as its source domain.

All experiments are implemented on a PC with Intel(R) Xeon(R) Gold 6230 CPU and a signal NVIDIA Geforce GTX 3090 GPU. The detection framework of AdvMix is the lightweight architecture YOLOv5s [33] with CSP-DarkNet53 as the backbone. The input resolution is set to 1024×1024 and the training batch is set to 2 (paired source and target images). We train the network for 50 epochs with COCO [46] pretrained parameters as initialization. For the computation of loss weight, we set scaling threshold $\beta = 10$ and confidence threshold $C_{th} = 0.5$. During the back propagation, the Stochastic Gradient Descent (SGD) with momentum is used to update the network parameters, where the learning rate is set as 0.01 and the momentum weight is 0.937. Other training setting and hyper-parameters are followed by YOLOv5s. During the inference, we evaluate the performance with the mean Average Precision (mAP) with IoU threshold of 0.5.

4.2. Detection Results on Domain Adaptation

To prove the effectiveness of AdvMix, we first show the detection precision for quantitative analysis. Then, the qualitative analysis is presented according to the visualization results. Experiments are conducted on cross-weather, cross-camera, and synthetic-to-real adaption benchmarks. The main challenge of the cross-weather adaption can be concluded as the style gap, where the overall scenes and weather conditions are different between source and target domains. In contrast, the content gap (including instance distribution, size, and density) needs to be diminished in the cross-camera and synthetic-to-real adaption.

The detection results of three adaption tasks are presented in Table 1. Apart from our proposed AdvMix, the baseline and the oracle are included in this subsection. Three methods share the same detection framework and the main difference between them is the training dataset. Specially, the baseline denotes that the detector is trained only on the labeled source domain, which shows the lower-bound of the network. The oracle means its training data contains the labeled target domain data, serving as the performance upper-bound.

For the cross-weather adaption, the accuracy of the baseline method is lower than 35%, which reflects a huge gap between source and target domain. Meanwhile, 51.8% mAP in the oracle indicates that the lightweight detection structure limits the performance upper-bound. AdvMix achieves competitive performance (50.1%), 15% higher than its baseline. The significant improvement demonstrates that AdvMix addresses the domain gap to some extent. For the cross-camera adaption, AdvMix obtains 57.1% mAP, outperforming the baseline by nearly 11%. The similar phenomenon can be witnessed in the synthetic-to-real task: the accuracy of AdvMix rises to 65.3% while the baseline maintains 59.4% in mAP. The above results prove that AdvMix boosts the detection precision and it yields an effective strategy for domain adaption.

Table 1. Detection results on different domain adaption benchmarks.

Domain Adaption	Method	mAP (%)
Cross-weather	Baseline	34.7
	AdvMix	50.1
	Oracle	51.8
Cross-camera	Baseline	46.6
	AdvMix	57.1
	Oracle	78.2
Synthetic-to-real	Baseline	59.4
	AdvMix	65.3
	Oracle	78.2

Furthermore, how AdvMix diminishes the domain shift is a significant issue that should be analyzed. We first discuss the style gap in the cross-weather adaption. As shown in Figure 9, some image features randomly drawn from the source (blue point) and target domain (red point) are visualized by the dimensionality reduction algorithm T-SNE [48], which maps high-dimensional data to points in the two-dimensional coordinate plane. For the baseline method, blue points cluster together while red points hardly gather into the blue cluster. It indicates that the baseline converges on a local optimum in the source domain and thus hardly overcomes the domain gap between images with different styles. When employing our proposed AdvMix, features from the source and target domain are merged into one cluster, showing that the style gap is addressed in AdvMix.

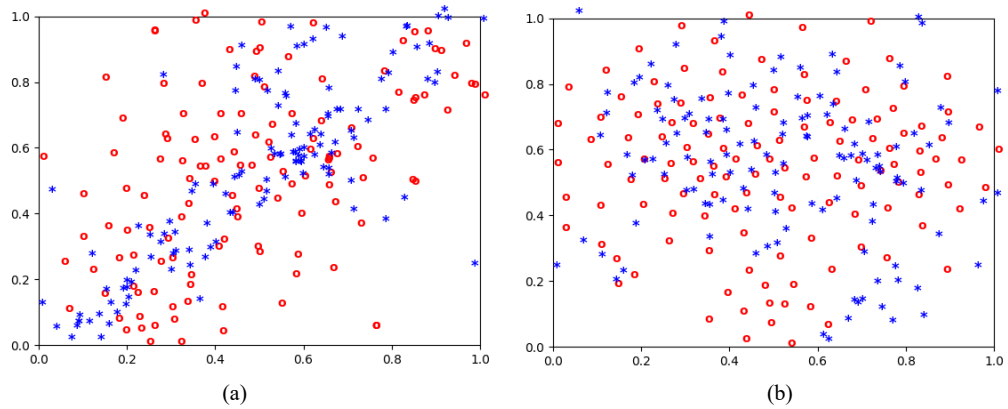
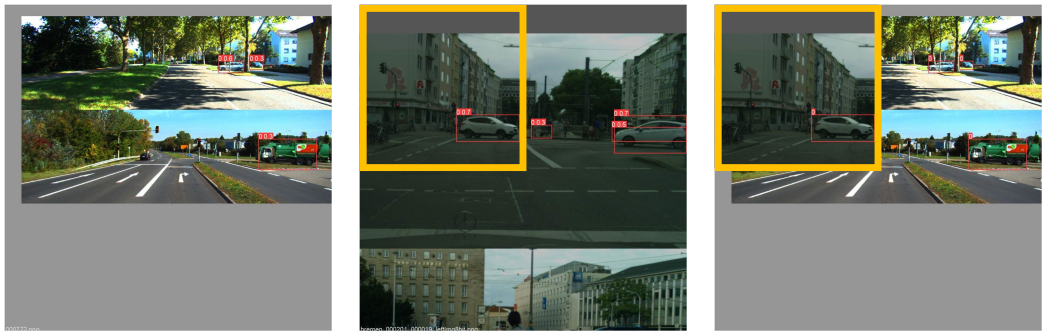


Figure 9. The distributions of features drawn from the source domain (blue) and the target domain (red) on the cross-weather adaption. (**a**) Baseline; (**b**) AdvMix.

Apart from the style gap, the instance differences (such as distinct object sizes, densities, and distributions) between two domains causes the content gap in the cross-camera and synthetic-to-real adaption tasks. AdvMix solves this problem by introducing the region

mixing strategy, where a target region with the highest confidence is combined with the source image to form the synthetic image x^M. Figures 10 and 11 show the forming processes of x_M in cross-camera and synthetic-to-real tasks, respectively. Note that all images in Figures 10 and 11 are drawn during training and they are augmented by Mosaic algorithm [49]. As presented in Figure 10b, AdvMix chooses the left-top region instead of the right-top due to the fact that the right-top obtains lower average confidence. This reflects that the selection strategy in AdvMix is effective at filtering false prediction. On the other hand, x^M with reliable pseudo label contains abundant instances from two domains, which is beneficial for content gap alignment.



(a) (b) (c)

Figure 10. The forming process of a synthetic image x_M on the cross-camera adaption. The yellow rectangle denotes the selective target region and the red rectangle shows instance location. (**a**) Source predictions $\hat{y}_S$; (**b**) target predictions $\hat{y}_S$; (**c**) x_M and its pseudo label $\hat{y}_{S,T}$.

(a) (b) (c)

Figure 11. The forming process of a synthetic image x_M on the synthetic-to-real adaption. The yellow rectangle denotes the selective target region and the red rectangle shows instance location. (**a**) Source predictions $\hat{y}_S$; (**b**) target predictions $\hat{y}_S$; (**c**) x_M and its pseudo label $\hat{y}_{S,T}$.

4.3. Comparison with State-of-the-Art Methods

Our proposed AdvMix is compared with recent advanced UDA methods on three domain adaption benchmarks. The compared methods are divided into two categories: adversarial and self-training strategies. The former contains EPM [26], SSOD [11], MGA [27], DA-AD [14], CDN [50], MeGA [17], SAPN [51], RPN-PR [18], UADAN [16], and SCAN [12]. The latter includes ConfMix [13], IRG [19], SC-UDA [20], FL-UDA [30], CTRP [21], GIPA [32], and SIGMA [22].

We first present the qualitative results on the cross-weather adaption benchmark, as shown in Table 2. Generally speaking, the adversarial methods obtain higher mAP than self-training strategies because the main challenge in cross-weather adaption is the style gap, which is successfully addressed by the gradient reversal layer in adversarial methods. However, SIGMA [22] and our proposed AdvMix are special cases of self-training methods. Different from the graph matching adaptor employed in SIGMA [22], AdvMix introduces a domain discriminator in the self-supervised training process. The detection results show that AdvMix with 50.1% mAP is effective on the style gap in comparison

with SIGMA (44.2%mAP). Furthermore, the APs of "person" and "truck" in AdvMix exceed the sub-optimal approaches by nearly 10%. We attribute this improvement to the hard example mining strategy, an efficient strategy for the precision balance of different categories. Despite the slightly lower AP of train (49.5%) compared to SSOD [11], the average accuracy of AdvMix is 6.8% higher than SSOD. In short, AdvMix outperforms advanced UDA detectors in accuracy, indicating that it is skilled at style gap elimination.

Table 2. Detection results of different UDA methods on the cross-weather adaption benchmark. The bold and the underline represent the highest and the second highest accuracy respectively.

Category	Method	AP (%)								mAP (%)
		Person	Car	Train	Rider	Truck	Bicycle	Bus	Motorcycle	
Adversarial Method	CDN [50]	35.8	50.9	29.8	45.7	30.1	36.5	42.5	30.8	36.3
	EPM [26]	44.0	57.1	39.7	43.6	29.4	36.1	44.9	29.0	40.2
	RPN-PR [18]	33.6	49.6	46.0	43.8	32.9	36.8	45.5	35.7	40.5
	SAPN [51]	40.8	59.8	37.5	46.7	24.3	40.7	46.8	30.4	40.9
	UADAN [16]	36.5	53.6	42.7	46.1	28.9	38.9	49.4	32.3	41.1
	MeGA [17]	37.7	52.4	46.9	49.0	25.4	39.0	49.2	34.5	41.8
	SCAN [12]	41.7	57.3	48.7	43.9	28.7	37.3	48.6	31.0	42.1
	DA-AD [14]	36.5	54.3	48.7	46.7	30.3	39.1	51.2	31.6	42.3
	SSOD [11]	38.8	57.2	**51.9**	45.9	29.9	40.9	50.2	31.9	43.3
	MGA [27]	43.9	60.6	39.0	49.6	29.6	42.8	50.7	38.3	44.3
Self-training Method	CTRP [21]	32.7	50.1	25.4	44.4	21.7	36.8	45.6	30.1	35.9
	SC-UDA [20]	38.5	56.0	29.7	43.7	27.1	39.5	43.8	31.2	38.7
	FL-UDA [30]	34.1	51.9	25.7	44.4	30.4	37.2	41.8	30.3	37.0
	IRG [19]	37.4	51.9	25.2	45.2	24.4	41.6	39.6	31.5	37.1
	GIPA [32]	32.9	54.1	41.1	46.7	24.7	38.7	45.7	32.4	39.5
	ConfMix [13]	45.0	62.6	40.0	43.4	27.3	33.5	45.8	28.6	40.8
	SIGMA [22]	44.0	60.3	51.5	43.9	31.6	40.6	50.4	31.7	44.2
	AdvMix	**54.0**	**68.9**	49.5	**51.5**	**39.5**	**44.3**	**53.5**	**39.3**	**50.1**

We also compared AdvMix with state-of-the-art methods on the cross-camera and synthetic-to-real adaption benchmarks, as illustrated in Table 3. Note that only the "car" class is evaluated on these tasks and thus we report its AP as mAP. It is remarkable that AdvMix achieves outstanding performance on two tasks. AdvMix obtains 65.3% mAP on synthetic-to-real adaption, surpassing most UDA approaches in detection precision. In comparison with Confmix [13], a popular UDA method proposed recently, AdvMix gains 5% and 9% mAPs in cross-camera and synthetic-to-real adaptions, respectively. In addition, we observe that adversarial methods show poor performance on two benchmarks, where the content gap is hardly diminished by feature alignments. However, the self-training detectors solve this problem by the self-supervision strategy. Apart from that, AdvMix proposes an extra region mixing module with a strict confidence metric, which is the reason of significant precision improvement.

In conclusion, the outstanding performance of AdvMix can be witnessed on different benchmarks. To be specific, MGA [27] obtains 44.3% mAP on the cross-weather adaption while its accuracy on the synthetic-to-real task is 54.6%, more than 10 % lower than AdvMix. Although ConfMix [13] diminishes the content gap to some extent, its "motorcycle" AP on the cross-weather benchmark is 28.6% (that of AdvMix is 39.3%). The above results show AdvMix is an efficient UDA method compared to recent advanced UDA detection networks.

Table 3. Detection results of different UDA methods on the cross-camera and synthetic-to-real adaption benchmarks. The bold and the underline represent the highest and the second highest accuracy respectively.

Category	Method	mAP (Car%)	
		Cross-Camera	Synthetic-to-Real
Adversarial Method	MeGA [17]	43.0	44.8
	SAPN [51]	43.4	44.9
	CDN [50]	44.9	49.3
	EPM [26]	45.0	51.2
	SCAN [12]	45.5	52.6
	SSOD [11]	47.6	49.3
	MGA [27]	48.5	54.6
Self-training Method	CTRP [21]	43.6	44.5
	FL-UDA [30]	44.6	43.1
	IRG [19]	45.7	43.2
	SIGMA [22]	45.8	53.7
	SC-UDA [20]	46.4	52.4
	GIPA [32]	47.9	47.6
	ConfMix [13]	<u>52.2</u>	<u>56.3</u>
	AdvMix	**57.1**	**65.3**

5. Ablation Studies

In this section, we conduct ablation studies to understand the effect of adversarial structure, mixing strategy, and image resolution. All detection networks are evaluated on the synthetic-to-real adaption benchmark.

5.1. Effect of Adversarial Structure

In AdvMix, we introduce a novel global-level domain discriminator in AdvGRL, as shown in Figure 2a. Besides, Figure 2b presents the pixel-level discriminator used in most domain adaptive methods [14,27,34]. Table 4 shows the detection results of AdvMix when the adversarial structure is changed from global-level to pixel-label discriminator. For the sake of fairness, we set $\beta = 0$ in the adversarial weight mapping function (Equation (9)) and the dynamic weight λ_{adv} for the domain discrimination loss L_D is fixed as $\lambda_0 = 0.01$ for two structures. We also present the detection accuracy of the method without the discriminator as the baseline in Table 4. Compared to the slight improvement (0.1%) of the pixel-level discriminator, a significant rise in mAP (4%) is witnessed when employing the global-level structure. We attribute it to the fact that the global-level discriminator produces a compact domain descriptor, which is more powerful than the sparse descriptor generated by the pixel-level one.

Table 4. Detection results of different adversarial structures.

Adversarial Structure	mAP (%)
Baseline (w/o discriminator)	60.2
Pixel-level discriminator	60.3
Global-level discriminator	64.2

In addition, we weigh the domain discrimination loss L_D by the adversarial weight mapping function, which enhances the weights of hard examples by an adversarial strategy. The weight λ_{adv} is calculated by Equation (9) and its value is controlled by the scaling threshold β. We compare the detection results with constant weight (β is set as 0) and dynamic weights (vary β from 8 to 12 in the step of 2). As illustrated in Table 5, dynamic weight with $\beta = 10$ achieves higher performance, with an improvement of 1.1% mAP in

comparison with the constant weight. Therefore, we conclude that the dynamic weight contributes to domain adaption by extracting more features of hard examples and increasing the stability of domain-invariant features.

Table 5. Detection results with different loss weights λ_{adv}.

Scaling Thresholds β	mAP (%)
0 (constant weight)	64.2
8	64.3
10	65.3
12	64.7

5.2. Effect of Mixing Strategy

In this subsection, the mixing strategy is varied across four different strategies, including vertical, horizontal, 4-division, and 6-division, as shown in Figure 12. The detection results of these strategies are shown in Table 6 and the baseline method without any mixing strategy is also included. We notice that mixing approaches except for horizontal mix outperform the baseline. It reflects that the cutting direction is a factor to influence the domain adaption. Compared to other mixing options, horizontal cutting contains more background semantic information (such as road) and less objects with high confidence, which impacts the self-training negatively. We also observe that mixing more or less target regions vertically may promote performance to some extent and 4-division is the most suitable strategy with over 5% rise in mAP.

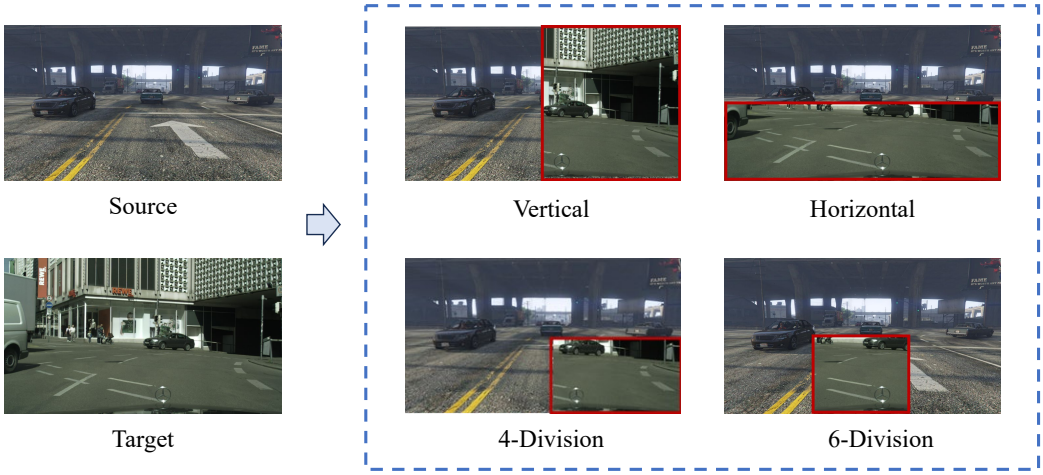


Figure 12. Illustration of the different mixing strategies.

Table 6. Detection results of different mixing strategies.

Mixing Strategy	mAP (%)
Baseline (w/o mix)	59.6
Vertical	64.5
Horizontal	57.1
4-Division	65.3
6-Division	64.6

Regarding the confidence metric in the region mixing module, we use the combined confidence C_{comb} instead of the original object confidence C_{obj} in YOLOv5 [33] to select trustworthy pseudo labels. We present the detection results with two confidence metrics in Table 7. The method with C_{comb} surpasses C_{obj} nearly 4% in terms of accuracy, indicating that C_{comb} filters out most unreliable bounding boxes and is beneficial to self-supervision learning.

Table 7. Detection results with different confidence metrics.

Confidence Metric	mAP (%)
C_{obj}	61.4
C_{comb}	65.3

5.3. Effect of Image Resolution

In this subsection, we analyze the effect of image resolution on different domain adaptive methods, as shown in Table 8. The compared methods consist of baseline (no adaption), Adv-only (adaption with adversarial structure) , Mixing-only (adaption with mixing strategy), and AdvMix (adaption with both adversarial structure and mixing strategy). It is obvious that the accuracy of all methods improves as the image resolution increases from 608 × 608 to 1024 × 1024. At different resolutions, methods with adaption obtain higher mAP than the baseline. We notice that Mixing-only exceeds the baseline by over 5% at 608 × 608 resolution and less than 1% at 1024 × 1024 resolution. However, our proposed AdvMix achieves more than 5.9% promotion at different resolutions, showing that AdvMix is a resolution-agnostic domain adaption approach.

Table 8. Detection results of different domain adaptive methods at different image resolutions.

Image Resolution	Domain Adaptive Method			mAP (%)
	Name	Adversarial Structure	Mixing Strategy	
608 × 608	Baseline			49.3
	Adv-only	✓		51.3
	Mixing-only		✓	54.6
	AdvMix	✓	✓	56.7
1024 × 1024	Baseline			59.4
	Adv-only	✓		59.6
	Mixing-only		✓	60.2
	AdvMix	✓	✓	65.3

6. Discussions

In this section, we discuss the domain adaption for multi-class detection. Then, we give some false examples of our proposed AdvMix and analyze how to improve in the future.

6.1. Domain Adaption for Multi-Class Detection

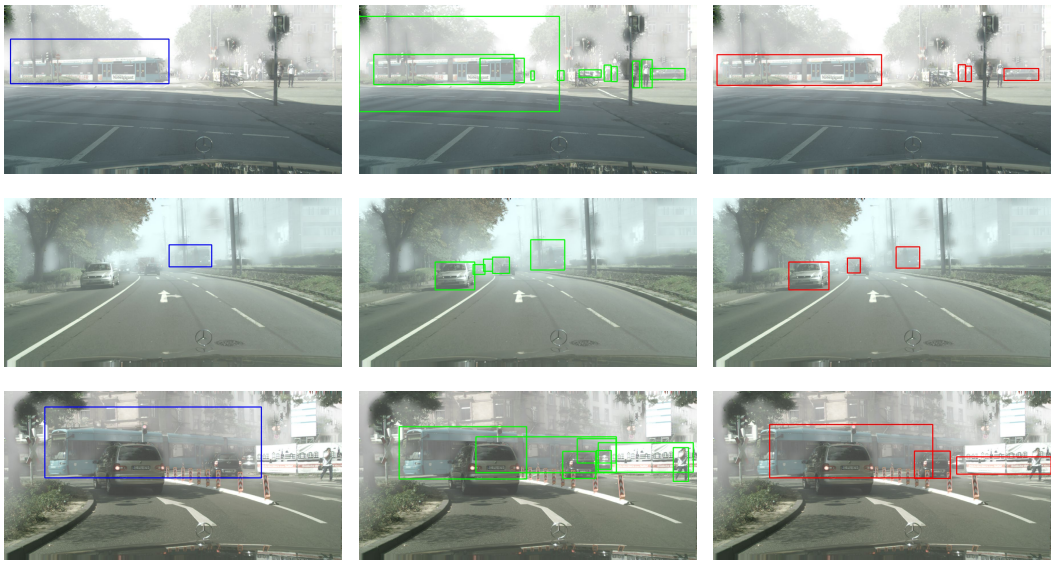
Regarding multi-class detection, the primary obstacle stems from the class imbalance. Taking the cross-weather adaption as example, we present the instance numbers of different classes and their corresponding APs (including baseline, AdvMix, oracle three methods AP) in Table 9. For some classes with insufficient instances, such as “train”, “truck”, “bus”, and “motorcycle”, their accuracy in AdvMix exceeds the baseline considerably and is on par with the oracle at the same time. It shows that AdvMix is effective at mitigating the domain shift even if the class distribution is imbalance. However, the performance gap between different classes is also witnessed. For example, “car” obtains 68.9AP, nearly 30% higher than the accuracy of “truck” (39.5AP) in AdvMix. We attribute this to the fact that the original detector (YOLOv5) overlooks the class-imbalanced issue. In the future, a class-balanced sampling strategy is necessary to introduce in a detection framework to enhance the recognition for rare classes.

Table 9. Instance number and detection result on the cross-weather adaption benchmark.

Class	Instance Number	Baseline AP (%)	AdvMix AP (%)	Oracle AP (%)
Person	3171	45.3	54.0	57.5
Car	4224	55.5	68.9	72.6
Train	22	4.6	49.5	48.4
Rider	481	43.6	51.5	53.3
Truck	88	24.3	39.5	41.2
Bicycle	996	38.1	44.3	45.7
Bus	86	39.0	53.5	53.7
Motorcycle	135	27.5	39.3	41.5

6.2. False Detections

We study false detections by visualizing the prediction bounding boxes of the "train" class in Figure 13. The "train" class with 22 instances is denoted as a rare class. Compared to the baseline method, AdvMix filters out more false positives and improves detection precision. However, some cars located far from the camera are recognized as the "train". Those objects are obscured by fog and the detector hardly distinguishes them. This indicates that the accuracy of rare class detection is still a challenging task although AdvMix copes with the domain shift in object detection.



(a) (b) (c)

Figure 13. Some examples of false detections. We take the "train" detection on the cross-weather adaption as example. (**a**) Ground truths (blue rectangular boxes); (**b**) the baseline predictions (green rectangular boxes); (**c**) AdvMix predictions (red rectangular boxes).

7. Conclusions

In this paper, an unsupervised cross-domain adaption approach AdvMix is proposed to diminish the discrepancy between the source and target domain. In AdvMix, the global descriptors of each domains are extracted by the domain discriminator and aligned by the gradient reversal layer with an adversarial hard examples mining strategy. Furthermore, the detector is self-trained by synthetic images from source and target domain to alleviate the content gap. We conduct extensive experiments and analytical studies on cross-weather, cross-camera and synthetic-to-real domain adaption scenarios. In comparison with advanced UDA methods, AdvMix outperforms them in terms of accuracy, demonstrating the prominent performance of AdvMix. However, there are some challenges that we need to solve in the future, including how to deal with diverse source domains, how to extend

our method to other detection frameworks, and how to confront the issue of inter-class imbalance in domain adaption.

Author Contributions: All of the authors contributed to this study. Conceptualization, R.C.; methodology, R.C. and D.L.; software, R.C and L.D.; data curation, D.L. and L.D.; writing—original draft preparation, R.C.; writing—review and editing, D.L., L.D. and Z.X.; funding acquisition, L.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Zhejiang Province Pioneer Research and Development Project "Research on Multi-modal Traffic Accident Holographic Restoration and Scene Database Construction Based on Vehicle-cloud Intersection" (Grant No. 2024C01017).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Authors Ruimin Chen, Dailin Lv, Li Dai and Liming Jin were employed by the company Zhejiang Geely Holding Group Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
2. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788. [CrossRef]
3. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as points. *arXiv* **2019**, arXiv:1904.07850. https://doi.org/10.48550/arXiv.1904.07850.
4. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
5. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
6. Alom, M.Z.; Taha, T.M.; Yakopcic, C.; Westberg, S.; Sidike, P.; Nasrin, M.S.; Hasan, M.; Van Essen, B.C.; Awwal, A.A.S.; Asari, V.K. A State-of-the-Art Survey on Deep Learning Theory and Architectures. *Electronics* **2019**, *8*, 292. [CrossRef]
7. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Proceedings of the Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer: Cham, Switzerlands, 2016; pp. 21–37. [CrossRef]
8. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully Convolutional One-Stage Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9626–9635. [CrossRef]
9. Yang, S.Y.; Cheng, H.Y.; Yu, C.C. Real-Time Object Detection and Tracking for Unmanned Aerial Vehicles Based on Convolutional Neural Networks. *Electronics* **2023**, *12*, 4928. [CrossRef]
10. Zhou, X.; Zhuo, J.; Krähenbühl, P. Bottom-Up Object Detection by Grouping Extreme and Center Points. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 850–859. [CrossRef]
11. Rezaeianaran, F.; Shetty, R.; Aljundi, R.; Reino, D.O.; Zhang, S.; Schiele, B. Seeking Similarities over Differences: Similarity-based Domain Alignment for Adaptive Object Detection. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 9184–9193. [CrossRef]
12. Li, W.; Liu, X.; Yao, X.; Yuan, Y. SCAN: Cross Domain Object Detection with Semantic Conditioned Adaptation. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 22 February –1 March 2022; Volume 36, pp. 1421–1428. [CrossRef]
13. Mattolin, G.; Zanella, L.; Ricci, E.; Wang, Y. ConfMix: Unsupervised Domain Adaptation for Object Detection via Confidence-based Mixing. In Proceedings of the 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2–7 January 2023; pp. 423–433. [CrossRef]
14. Li, J.; Xu, R.; Ma, J.; Zou, Q.; Ma, J.; Yu, H. Domain Adaptive Object Detection for Autonomous Driving under Foggy Weather. In Proceedings of the 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2–7 January 2023; pp. 612–622. [CrossRef]
15. Johnson-Roberson, M.; Barto, C.; Mehta, R.; Sridhar, S.N.; Rosaen, K.; Vasudevan, R. Driving in the Matrix: Can virtual worlds replace human-generated annotations for real world tasks? In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 746–753. [CrossRef]
16. Guan, D.; Huang, J.; Xiao, A.; Lu, S.; Cao, Y. Uncertainty-Aware Unsupervised Domain Adaptation in Object Detection. *IEEE Trans. Multimed.* **2022**, *24*, 2502–2514. [CrossRef]

17. VS, V.; Gupta, V.; Oza, P.; Sindagi, V.A.; Patel, V.M. MeGA-CDA: Memory Guided Attention for Category-Aware Unsupervised Domain Adaptive Object Detection. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 4514–4524. [CrossRef]
18. Zhang, Y.; Wang, Z.; Mao, Y. RPN Prototype Alignment For Domain Adaptive Object Detector. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 12420–12429. [CrossRef]
19. VS, V.; Oza, P.; Patel, V.M. Instance Relation Graph Guided Source-Free Domain Adaptive Object Detection. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 3520–3530. [CrossRef]
20. Yu, F.; Wang, D.; Chen, Y.; Karianakis, N.; Shen, T.; Yu, P.; Lymberopoulos, D.; Lu, S.; Shi, W.; Chen, X. SC-UDA: Style and Content Gaps aware Unsupervised Domain Adaptation for Object Detection. In Proceedings of the 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2022; pp. 1061–1070. [CrossRef]
21. Zhao, G.; Li, G.; Xu, R.; Lin, L. Collaborative Training Between Region Proposal Localization and Classification for Domain Adaptive Object Detection. In *Proceedings of the Computer Vision—ECCV 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer: Cham, Switzerland, 2020; pp. 86–102. [CrossRef]
22. Li, W.; Liu, X.; Yuan, Y. SIGMA: Semantic-complete Graph Matching for Domain Adaptive Object Detection. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 5281–5290. [CrossRef]
23. Ganin, Y.; Lempitsky, V. Unsupervised Domain Adaptation by Backpropagation. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 7–9 July 2015; Volume 37, pp. 1180–1189.
24. Chen, Y.; Li, W.; Sakaridis, C.; Dai, D.; Van Gool, L. Domain Adaptive Faster R-CNN for Object Detection in the Wild. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3339–3348. [CrossRef]
25. Zhu, X.; Pang, J.; Yang, C.; Shi, J.; Lin, D. Adapting Object Detectors via Selective Cross-Domain Alignment. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 687–696. [CrossRef]
26. Hsu, H.K.; Yao, C.H.; Tsai, Y.H.; Hung, W.C.; Tseng, H.Y.; Singh, M.; Yang, M.H. Progressive Domain Adaptation for Object Detection. In Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass, CO, USA, 1–5 March 2020; pp. 738–746. [CrossRef]
27. Zhou, W.; Du, D.; Zhang, L.; Luo, T.; Wu, Y. Multi-Granularity Alignment Domain Adaptation for Object Detection. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 9571–9580. [CrossRef]
28. Shan, Y.; Lu, W.F.; Chew, C.M. Pixel and feature level based domain adaptation for object detection in autonomous driving. *Neurocomputing* **2019**, *367*, 31–38. [CrossRef]
29. Kim, T.; Jeong, M.; Kim, S.; Choi, S.; Kim, C. Diversify and Match: A Domain Adaptive Representation Learning Paradigm for Object Detection. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 12448–12457. [CrossRef]
30. Li, X.; Chen, W.; Xie, D.; Yang, S.; Yuan, P.; Pu, S.; Zhuang, Y. A Free Lunch for Unsupervised Domain Adaptive Object Detection without Source Data. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35, 8474–8481. [CrossRef]
31. Cai, X.; Luo, F.; Qi, W.; Liu, H. A Semi-Supervised Object Detection Algorithm Based on Teacher-Student Models with Strong-Weak Heads. *Electronics* **2022**, *11*, 3849. [CrossRef]
32. Xu, M.; Wang, H.; Ni, B.; Tian, Q.; Zhang, W. Cross-Domain Detection via Graph-Induced Prototype Alignment. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 12352–12361. [CrossRef]
33. Jocher, G.; Stoken, A.; Borovec, J.; Changyu, L.; Hogan, A.; Diaconu, L.; Poznanski, J.; Yu, L.; Rai, P.; Ferriday, R.; et al. ultralytics/yolov5: v3. 0. Available online: https://github.com/ultralytics/yolov5/tree/v3.0 (accessed on 20 December 2023).
34. Luo, Q.; Wang, Y.; Li, W.; Xiong, R. Joint Feature-level and Pixel-level Domain Adaption for Object Detection in the Wild. In Proceedings of the 2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), Suzhou, China, 29 July–2 August 2019; pp. 559–565. [CrossRef]
35. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 3213–3223. [CrossRef]
36. Sakaridis, C.; Dai, D.; Van Gool, L. Semantic foggy scene understanding with synthetic data. *Int. J. Comput. Vis.* **2018**, *126*, 973–992. [CrossRef]
37. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361. [CrossRef]

38. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017. [CrossRef]
39. Hosang, J.; Benenson, R.; Schiele, B. Learning Non-maximum Suppression. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6469–6477. [CrossRef]
40. Law, H.; Deng, J. CornerNet: Detecting Objects as Paired Keypoints. In *Proceedings of the Computer Vision—ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer: Cham, Switzerland, 2018; pp. 765–781. [CrossRef]
41. Gretton, A.; Borgwardt, K.M.; Rasch, M.J.; Schölkopf, B.; Smola, A. A kernel two-sample test. *J. Mach. Learn. Res.* **2012**, *13*, 723–773.
42. Fernando, B.; Habrard, A.; Sebban, M.; Tuytelaars, T. Unsupervised Visual Domain Adaptation Using Subspace Alignment. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 1–8 December 2013; pp. 2960–2967. [CrossRef]
43. Ganin, Y.; Ustinova, E.; Ajakan, H.; Germain, P.; Larochelle, H.; Laviolette, F.; Marchand, M.; Lempitsky, V., Domain-Adversarial Training of Neural Networks. In *Domain Adaptation in Computer Vision Applications*; Springer International Publishing: Cham, Switzerland, 2017; pp. 189–209. [CrossRef]
44. Tzeng, E.; Hoffman, J.; Saenko, K.; Darrell, T. Adversarial Discriminative Domain Adaptation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2962–2971. [CrossRef]
45. Choi, J.; Chun, D.; Kim, H.; Lee, H.J. Gaussian YOLOv3: An Accurate and Fast Object Detector Using Localization Uncertainty for Autonomous Driving. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 502–511. [CrossRef]
46. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Proceedings of the Computer Vision–ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B.,Tuytelaars, T., Eds.; Springer: Cham, Switzerland, 2014; pp. 740–755. [CrossRef]
47. Hsu, C.C.; Tsai, Y.H.; Lin, Y.Y.; Yang, M.H. Every Pixel Matters: Center-Aware Feature Alignment for Domain Adaptive Object Detector. In *Proceedings of the Computer Vision—ECCV 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer: Cham, Switzerland, 2020; pp. 733–748. [CrossRef]
48. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
49. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934. https://doi.org/10.48550/arXiv.2004.10934.
50. Su, P.; Wang, K.; Zeng, X.; Tang, S.; Chen, D.; Qiu, D.; Wang, X. Adapting Object Detectors with Conditional Domain Normalization. In *Proceedings of the Computer Vision—ECCV 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer: Cham, Switzerland, 2020; pp. 403–419. [CrossRef]
51. Li, C.; Du, D.; Zhang, L.; Wen, L.; Luo, T.; Wu, Y.; Zhu, P. Spatial Attention Pyramid Network for Unsupervised Domain Adaptation. In *Proceedings of the Computer Vision—ECCV 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer: Cham, Switzerland, 2020; pp. 481–497. [CrossRef]