

Supplementary materials

I. THE THREE MODULES OF THE PROPOSED METHOD

A. Multi-level Domain Alignment (MDA) Module

The goal of object detection is to learn the posterior probability $P(C, B|I)$, where I , B , and C denote the image, object location, and object category, respectively. The joint distribution of the training samples is $P(C, B, I)$, with source and target domain distributions denoted as $P_s(C, B, I)$ and $P_t(C, B, I)$, respectively. When domain shift occurs, $P_s(C, B, I) \neq P_t(C, B, I)$. Our objective is to determine the object category and location from a given image, represented by $P(C, B|I)$.

According to Bayes' theorem, $P(C, B, I)$ can be expressed as $P(C, B|I)P(I)$. For object detection, the covariate shift assumption requires the conditional probability to be consistent across domains, implying that detection results should be domain-independent. Domain distribution shift arises from differences in $P(I)$, which represents the feature mapping output of the convolutional layers in the *Faster-RCNN* model. To address this, we enforce that image-level distributions of the source and target domains be identical: $P_s(I) = P_t(I)$.

1) *Pixel-Level Alignment*: The pixel-level domain classifier D_1 is designed to focus on low-level features and is a fully convolutional network with a kernel size of one. We split the *VGG-16* feature extraction network into five convolutional blocks, with \mathcal{F}_3 providing a feature map of dimensions $\mathcal{W} \times \mathcal{H}$. This feature map is input to D_1 , as illustrated in Fig.2.

We pass the lower-layer feature maps through the Gradient Reversal Layer (GRL) and use adversarial learning to minimize pixel feature alignment loss during forward propagation. The output is a domain prediction map with dimensions matching the input feature map, thus enhancing robustness.

The domain classifier D_1 is trained using least squares loss to stabilize its training. Ablation experiments in Section 4.6 confirm the module's effectiveness in aligning low-level features. The pixel-level alignment loss functions are defined as:

$$\mathcal{L}_{pixel_s} = \frac{1}{n_s HW} \sum_{i=1}^{n_s} \sum_{w=1}^W \sum_{h=1}^H D_1(F_1(x_i^s))_{wh}^2 \quad (1)$$

$$\mathcal{L}_{pixel_t} = \frac{1}{n_t HW} \sum_{i=1}^{n_t} \sum_{w=1}^W \sum_{h=1}^H (1 - D_1(F_1(x_i^t))_{wh})^2 \quad (2)$$

$$\mathcal{L}_{pixel}(F, D_1) = \frac{1}{2} (\mathcal{L}_{pixel_s} + \mathcal{L}_{pixel_t}) \quad (3)$$

Here, $D_1(F_1(x_i^i))_{wh}^2$ denotes the domain classifier output at each spatial position.

2) *Image-Level Alignment*: As the feature extraction network deepens, it produces increasingly complex feature maps. While pixel-level alignment focuses on low-level features, it may not fully adapt the model to domain shifts. In contrast, image-level features capture the global attributes of an image. We extract these global features from \mathcal{F}_4 and \mathcal{F}_5 and use them as inputs to domain classifiers D_2 and D_3 . These classifiers predict the domain of the input features, aligning global attributes and reducing differences in background, scene, and object layout between the source and target domains, thereby enhancing model generalization.

However, in feature space, target domain images that are easily distinguishable from source domain images are far apart, while less distinct images are closer. This is illustrated in Fig.3. Previous domain classifiers often use softmax or cross-entropy loss, which may overemphasize easily classified samples, leading to overconfidence. To address this, we employ Focal Loss, which introduces an attenuation factor to down-weight easy-to-classify samples, thus focusing more on difficult samples and improving classifier performance.

The loss functions for the image-level domain classifiers are defined as follows, where N_s and N_t represent the number of samples in the source and target domains, respectively, and D_j denotes the j_{th} domain classifier:

$$\mathcal{L}_{img_s} = -\frac{1}{n_s} \sum_{j=2}^3 \sum_{i=1}^{n_s} (1 - D_j(F(x_i^s)))^\gamma \log(D_j(F(x_i^s))) \quad (4)$$

$$\mathcal{L}_{img_t} = -\frac{1}{n_t} \sum_{j=2}^3 \sum_{i=1}^{n_t} (D_j(F(x_i^t)))^\gamma \log(1 - D_j(F(x_i^t))) \quad (5)$$

$$\mathcal{L}_{img}(F, D_j) = \frac{1}{2} (\mathcal{L}_{img_s} + \mathcal{L}_{img_t}) \quad (6)$$



Fig. 1. Comparison of the results of object detection using *Faster-RCNN* in normal and foggy weather.

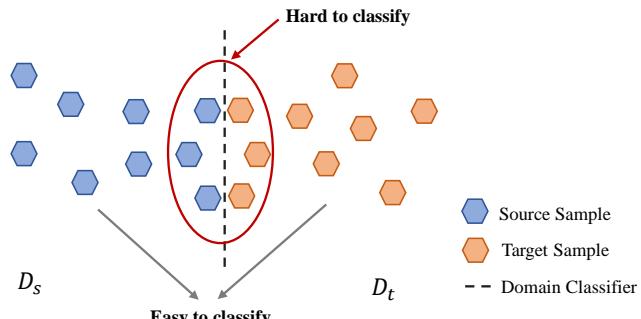


Fig. 3. Distribution alignment using domain classifiers. Blue indicates source domain samples, orange indicates target domain samples, and the dashed line represents the domain classifier.

3) *Object-Level Alignment*: Object-level representation refers to the RoI-based feature vector before it is passed to the final class classifier. Aligning these object-level representations helps mitigate local instance differences such as object appearance and size. To achieve this, we train domain classifiers on these feature vectors to align object-level distributions. Let $P_{i,j}$ denote the output of the object-level domain classifier for the j^{th} region in the i^{th} image. The object-level alignment loss is defined as follows:

$$\mathcal{L}_{obj_s} = -\frac{1}{n_s} \sum_{i=1}^{n_s} (1 - D_4(F(x_i^s))) \log (D_4(F(x_i^s))) \quad (7)$$

$$\mathcal{L}_{obj_t} = -\frac{1}{n_t} \sum_{i=1}^{n_t} D_4(F(x_i^t)) \log (1 - D_4(F(x_i^t))) \quad (8)$$

$$\mathcal{L}_{obj}(F, D_4) = \frac{1}{2} (\mathcal{L}_{obj_s} + \mathcal{L}_{obj_t}) \quad (9)$$

In summary, the total loss for the Multi-Level Domain Alignment (*MDA*) module is:

$$\mathcal{L}_{mda} = \mathcal{L}_{pixel}(F, D_1) + \mathcal{L}_{img}(F, D_2) + \mathcal{L}_{img}(F, D_3) + \mathcal{L}_{obj}(F, D_4) \quad (10)$$

B. Multi-Level Consistency Regularization (*MCR*) Module

The *MCR* module is introduced to stabilize the training process and address potential misalignments that may affect object-level alignment. If feature extraction alignment strategies are ineffective, they can negatively impact object-level alignment. To mitigate this, we apply consistency regularization between the domain classifiers at each feature extraction stage and the object-level domain classifier. This regularization enforces that the feature representations from both domains remain consistent, enhancing the model's generalization and robustness.

The *MCR* loss is defined as:

$$\mathcal{L}_{mcr} = \sum_{i,j} \left\| \frac{1}{|I|} \sum_{u,v} p_i^{(u,v)} - p_{i,j} \right\|_2 \quad (11)$$

Here, $|I|$ denotes the total number of activations in the feature map, and $\|\cdot\|_2$ represents the L_2 norm.

C. Ground Truth-Based Semantic Alignment (GSA) Module

To address discrepancies in data distributions between the source and target domains, we introduce the *GSA* module inspired by knowledge distillation. This module aims to align features or semantic information between the source and target domains using ground truth. By leveraging labeled samples from the source domain, the model is encouraged to learn feature representations that carry similar semantic meaning in both domains. This alignment improves the model's ability to generalize from labeled source data to unlabeled target data.

The *GSA* module minimizes the domain difference and maximizes classification accuracy on labeled source domains. It ensures that the features are not only discriminative for the source domain but also semantically aligned with the target domain. The loss function for *GSA* is given by:

$$\mathcal{L}_{gsa}(x, y) = \begin{cases} 0.5(x - y)^2 & \text{if } |x - y| < 1 \\ |x - y| - 0.5 & \text{otherwise} \end{cases} \quad (12)$$

II. EXPERIMENTAL DETAILS

A. Evaluation Metrics

Precision [8], *IoU* [13], *recall* [8], *AP* [5], *mAP* [6], and *FPS* [7] are commonly used metrics for evaluating the performance of object detection. Below, we will briefly review these formulas.

Intersection Over Union (*IoU*) is used to evaluate the overlap between two bounding boxes. As shown in Equation (13), where *A* is the label of the bbox and *B* is the predicted bbox. For each object, the detection was considered correct when the intersection over union ratio *IoU* was greater than 0.5.

$$IoU = \frac{A \cap B}{A \cup B} \quad (13)$$

Precision, also known as accuracy, is the percentage of true positive predictions (*TP*) of the identified object, as shown in Equation (14). Recall is the ratio of correctly identified objects to the total number of objects, as shown in Equation (15). Where *TP* is a true positive, *FP* is a false positive, and *FN* is a false negative.

$$Precision = \frac{TP}{TP + FP} \quad (14)$$

$$Recall = \frac{TP}{TP + FN} \quad (15)$$

mAP stands for Mean Average Precision, which is an index used to measure the detection accuracy in object detection. The formula is:

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (16)$$

where *N* is the number of categories, and *AP* stands for Average Precision, which represents the average precision, and the formula is:

$$AP = \int_0^1 P(R)dR \quad (17)$$

where *P* is Precision, and *R* is Recall.

FPS represents the number of frames processed per second. In addition to detection accuracy, speed is another crucial evaluation metric for object detection algorithms, especially for achieving real-time detection. Faster detection is indicated by shorter processing times and larger *FPS*. The formula to calculate *FPS* is:

$$FPS = \frac{1}{processing_time} \quad (18)$$

where *processing_time* represents the time it takes to process an image.

B. Datasets

srFog-Vehicle Dataset. Considering the scarcity of publicly available datasets for object detection in adverse weather conditions, we have established a rainy and foggy weather traffic road vehicle detection dataset based on the *Vehicle Color-24* dataset [5] and *Rain Vehicle Color-24* dataset [6]. This dataset is called *srFog-Vehicle*. Specifically, we utilize the well-known atmospheric scattering model [10] to generate foggy weather images $I(x)$, which can be obtained by the following formula:

$$I(x) = J(x)t(x) + A(1 - t(x)) \quad (19)$$

In Equation (19), $I(x)$ represents the hazy image, $J(x)$ denotes the clean image, A stands for the global atmospheric light, and $t(x)$ signifies the medium transmittance. The specific definition of $t(x)$ is as follows:

$$t(x) = e^{-\beta d(x)} \quad (20)$$

where β is the atmospheric scattering coefficient and $d(x)$ is the distance between the object and the camera.

In our experiment, we set the global atmospheric light parameter A to 0.4 while randomly setting the atmospheric scattering parameter β to 0.05 to control the fog level. In addition, considering that *Vehicle Color-24* datasets are vehicle datasets with 24 colors, we modified its label and only detected vehicles, including the category of car, with a total of 10091 images. The example of the dataset is shown in Fig.4.



Fig. 4. Example image from the proposed *srFog-Vehicle Dataset*.

Foggy Cityscapes Dataset. *Foggy Cityscapes* [15] comprises eight categories: person, rider, car, truck, bus, train, motorcycle, and bicycle. Moreover, since *Foggy Cityscapes* is based on *Cityscape* [2], we can utilize the labels from *Cityscape* as the ground truth (or labels) corresponding to *Foggy Cityscapes*.

Foggy Driving Dataset. The *Foggy Driving Dataset* [15] consists of 101 color images depicting real-world foggy driving scenes, designed for evaluating object detection performance in foggy conditions. It comprises 509 vehicle instances (car, truck, bus, train, motorcycle, and bicycle) and 290 human instances (people and rider) for testing generalization.

RTTS Dataset. *RTTS* [11] is a real-world foggy image dataset and a subset of the *RESIDE Dataset*. The *RTTS* dataset is designed to evaluate the performance of traffic surveillance systems under foggy conditions. The images in the dataset capture various traffic scenarios, including car, bicycle, motorcycle, people, and bus, observed on urban roads.

DAWN Dataset. The *DAWN Dataset* [9] consists of over 1000 images captured in real traffic environments. These images cover four weather conditions: fog, snow, rain, and sandstorm, as detailed in Table I. The dataset contains a total of 7845 instances, categorized into five classes: car, bus, truck, motorcycle, and person. It is specifically designed for evaluating object detection performance in adverse weather conditions.

During the experiment, the datasets were divided into two groups: *Cityscapes* and *Foggy Cityscapes*, *Vehicle Color-24*, and *srFog-Vehicle* dataset. To demonstrate the generalization of the proposed algorithm, alignment tests were conducted on synthetic and real datasets separately. Detailed descriptions of the datasets used are listed in Table I.

TABLE I
DESCRIPTION OF SYNTHETIC AND REAL DATASETS. THE VALUES IN THE TABLE REPRESENT THE NUMBER OF IMAGES FOR EACH TRAINING SET AND TEST SET.

Datasets	Weather Type	Training Images	Testing Images
Foggy Cityscapes	Foggy weather	2950	500
Cityscapes	Normal weather	2950	-
srFog-Vehicle	Rainy and Foggy weather	9297	800
Vehicle Color-24	Normal weather	9297	-
Foggy Driving	Foggy weather	-	101
RTTS	Foggy weather	-	100
DAWN	Foggy weather	-	300
	Rain weather	-	200
	Sand weather	-	323
	Snow weather	-	204

C. Implementation Details

Training details. In all experiments, we set the short side of the image to 600 and used *VGG-16* as the feature extraction network. We first trained the network with a learning rate of 1×10^{-3} for 50,000 iterations, then trained the network with a learning rate of 1×10^{-4} for another 50,000 iterations, and reported the final performance. All the models in our experiment were trained using this learning rate decay strategy. In the experiment, we set λ_1 to 0.3 and λ_2 to 0.1. We implemented all of these methods with PyTorch.

Evaluation Settings. To quantitatively evaluate the performance of our proposed *RTRDM* algorithm, we set the Intersection over Union (*IoU*) threshold to 0.5 and calculated the Average Precision (*AP*) for each class and the mean Average Precision (*mAP*) across all classes as evaluation metrics. Then, we compared our proposed method with several recent domain adaptation methods used in experiments, including *DAF* [1], *SWF* [14], *CRDA* [17], *ATF* [3], *MCAR* [19], *PDA* [4], *RPN-PR* [18], *LODS* [12], and *IGR* [16].

III. PARTIAL EXPERIMENTAL RESULTS VISUALIZATION

A. Experiments on Foggy Weather Images

1) *Comparison on Synthetic Dataset:* We quantitatively and qualitatively evaluated our algorithm against other domain-adaptive object detection methods using the synthetic *Foggy Cityscapes Dataset*. Our method achieved the best detection performance, with a *mAP* improvement of 41.2%, outperforming the second-best method, *RPN-PR*, by 2.2%. The proposed approach reduces domain differences across most categories in the *Foggy Cityscapes Dataset*.

Our algorithm performs particularly well for bus (49.3%), motorcycle (35.9%), truck (32.2%), and rider (45.9%), as highlighted in Table II. Specifically, the *AP* for bus is 49.3%, surpassing the second-best method, *CRDA* (45.1%), by 4.2%. While *IGR* excels in bicycle (41.6%), *PDA* in car (54.4%), and *MCAR* in train (43.4%), our method performs comparably well in these categories, with only minor differences. Overall, our approach achieves the highest *mAP* on the *Foggy Cityscapes Dataset* compared to existing domain-adaptive methods. Fig.5 illustrates test examples: in the first row, our method accurately detects both person and rider, while *IGR* and *LODS* struggle. Additionally, our method outperforms *RPN-PR* in identifying distant vehicles. In the second row, while *IGR* misclassifies the motorcycle, both *LODS*, *RPN-PR*, and our approach correctly identify the target, with our method showing higher confidence.



Fig. 5. Comparison of the *RTRDM* algorithm with the latest domain adaptation algorithm on the *Foggy Cityscapes* (zooming in electronically is recommended).

Moreover, Fig.6 provides further detection examples on the *Foggy Cityscapes Dataset*, demonstrating that our method effectively detects objects like person and car, further validating its robustness.

2) *Comparison on Real-world Dataset:* To evaluate the generalization performance of our algorithm in real foggy weather, we conducted tests on the *Foggy Driving*, *RTTS*, and *DAWN* datasets, all featuring foggy conditions. Fig. 7, Fig. 8, and Fig.9 display the results of our algorithm alongside the latest domain-adaptive methods on these datasets. Our algorithm achieves superior localization with higher confidence. These results demonstrate the strong performance of our *RTRDM* algorithm across different real-world foggy traffic datasets, underscoring its ability to learn domain-invariant features for cross-domain object detection.

3) *Comparison on Real-world Dataset:* In order to verify the generalization performance of the algorithm under real Foggy weather, this study conducted tests on *Foggy Driving*, *RTTS* and *DAWN* datasets including foggy weather conditions. Fig.7, Fig.8 and Fig.9 respectively show the test results of our algorithm and the latest domain adaptive method on three real fog weather datasets (*RTTS*, *Foggy Driving* and *DAWN*). For instance, in the first row of Fig.7, all other algorithms suffer from missed detections and false positives, whereas only our algorithm accurately detects both persons and bicycles. In the second



Fig. 6. Detection results on the *Foggy Cityscapes*. We can see that our method could accurately detect the objects existing in the images, e.g., the car, bicycle, rider, and person (zooming in electronically is recommended).

row, the other algorithms incorrectly classify bicycle as car or motorcycle. In the third row, all algorithms correctly identify a vehicle, but our algorithm provides the most accurate localization and the highest confidence level. These observations highlight the significant performance advantage of our *RTRDM* algorithm in object detection tasks.

The comparison results show that our algorithm has very good detection performance in different real-world foggy traffic road datasets, indicating that our algorithm can learn domain invariant features well, so as to achieve cross-domain object detection.

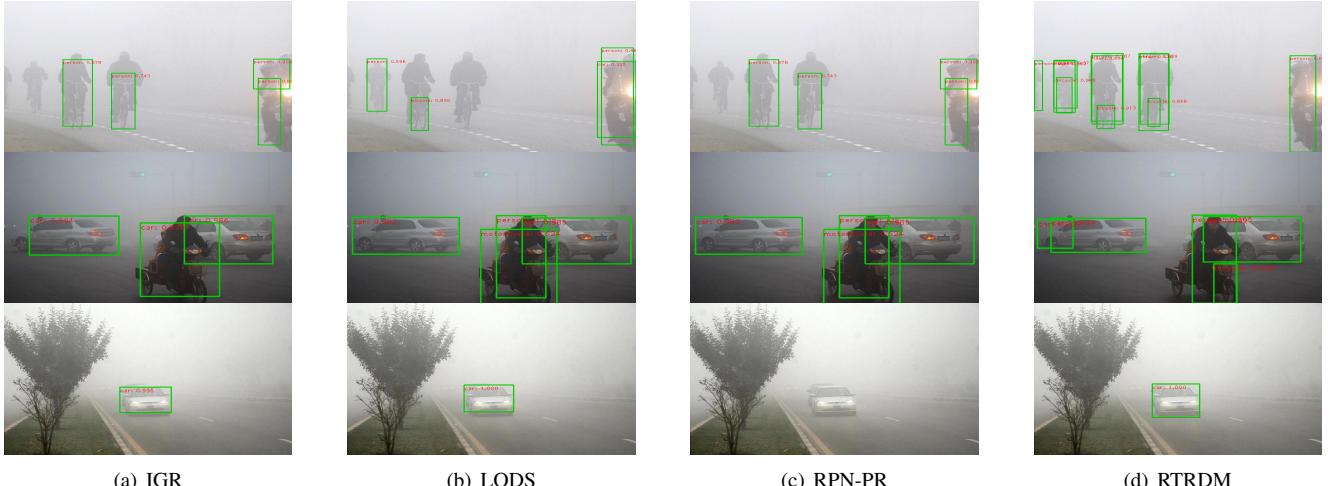


Fig. 7. Comparison of *RTRDM* algorithm with with the latest domain adaptation algorithm on the *RTTS Dataset* (electronic amplification is recommended).

B. Experiments on other Adverse Weather Images

1) *Comparison on Synthetic Dataset*: We evaluated our algorithm against recent domain adaptive object detection methods on the self-built synthetic rainy and foggy *srFog-Vehicle Dataset*. The results are summarized in Fig.10 and Table III . Our method outperforms others, improving *mAP* by 20.6% over *FRCNN*, 13.7% over *LODS*, 6.2% over *IGR*, and 2.7% over *RPN-PR*, demonstrating superior detection accuracy under adverse conditions. Fig.10 further illustrates our algorithm's ability to detect vehicles with the highest confidence.

TABLE III
mAP COMPARISON WITH OTHER METHODS ON THE *Vehicle Color-24 → srFog-Vehicle Dataset* EXPERIMENT (%).

Methods	FRCNN(source)	RPN-PR	LODS	IGR	RTRDM
<i>mAP</i>	65.3	87.2	76.2	83.7	89.9



Fig. 8. Comparison of *RTRDM* algorithm with with the latest domain adaptation algorithm on the *Foggy Driving Dataset* (electronic amplification is recommended).

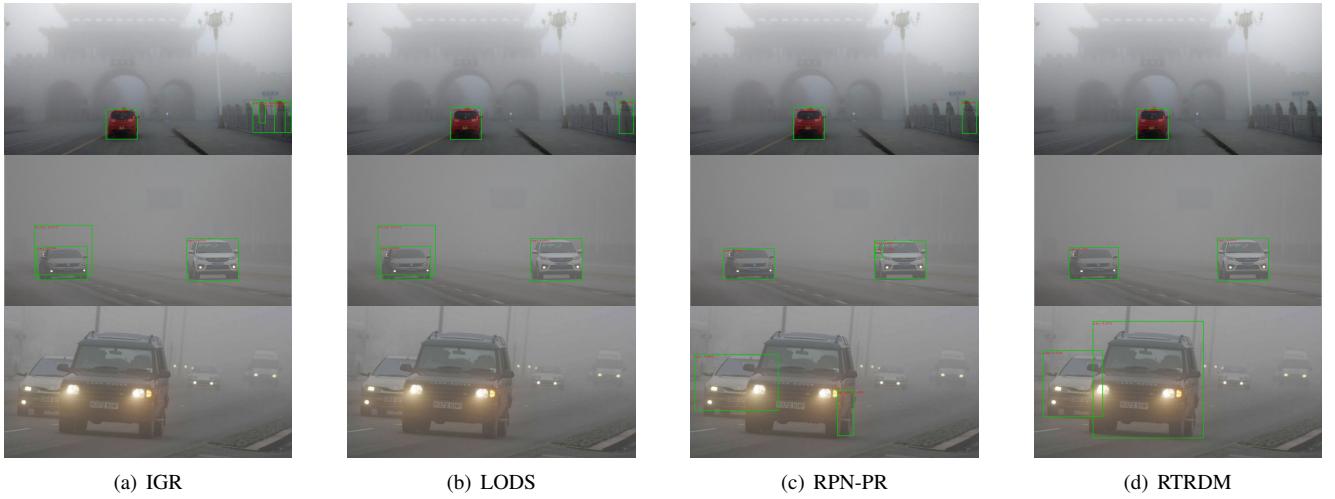


Fig. 9. Comparison of *RTRDM* algorithm with with the latest domain adaptation algorithm on the *DAWN (fog) Dataset* (electronic amplification is recommended).

2) *Comparison on Real-world Dataset*: To assess the generalization ability of our algorithm under real adverse weather, we tested it on the *DAWN* dataset, which includes fog, rain, dust, and snow conditions. The results, shown in Fig.11, Fig.12, and Fig.13, confirm that our network learns domain-invariant features effectively and performs well across different weather scenarios, demonstrating strong generalization.

C. Ablation Study

Effect of the different components on mAP. Compared to classical domain adaptive object detection methods, the proposed network exhibits superior detection performance. To further evaluate the effectiveness of *RTRDM*, we performed ablation studies of each module to analyze the effect of different components on network accuracy. Firstly, we used the original *Faster-RCNN* detector as the baseline for building our base network. Then, we trained this model using the implementation details mentioned earlier and tested it on the *Foggy Cityscapes Dataset*. The performance of these models is shown in Table IV . We can observe from Table IV that each component in *RTRDM* contributes to improving the object detection performance.

Effect of the weights in loss functions. In order to enhance the detection performance of *RTRDM* under adverse weather conditions,we developed an effective uniform loss function that includes losses \mathcal{L}_{det} , \mathcal{L}_{mda} , \mathcal{L}_{mcr} , and \mathcal{L}_{gsa} . Since our task involves domain adaptive object detection, we set the weights of detection loss and domain adaptation loss to 1 and use two loss weights (λ_1 and λ_2) to balance the remaining two modules. To determine the optimal values of λ_1 and λ_2 , we conducted

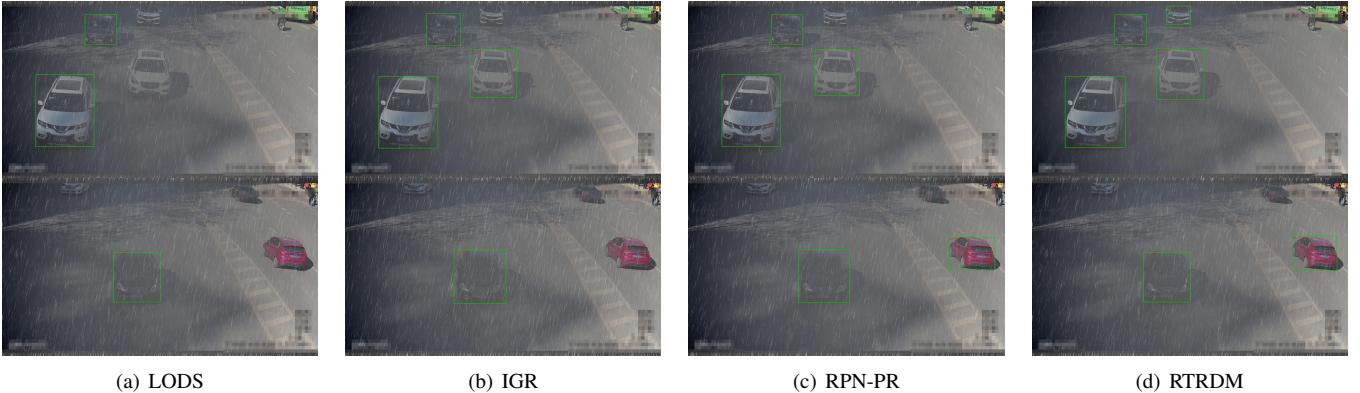


Fig. 10. Comparison of *RTRDM* algorithm with classical domain adaptive algorithm on the *srFog-Vehicle Dataset* (electronic amplification is recommended).



Fig. 11. Comparison of *RTRDM* algorithm with the latest domain adaptation algorithm on the *DAWN (rain) Dataset* (electronic amplification is recommended).

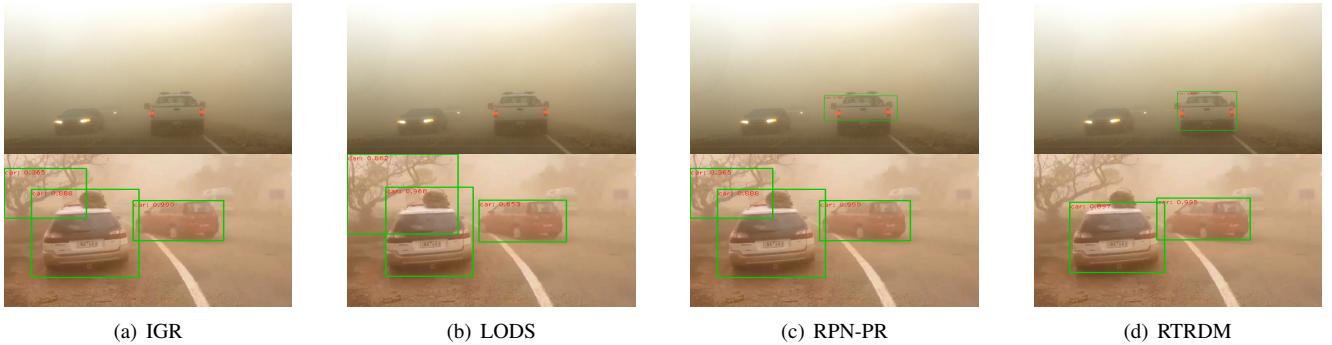


Fig. 12. Comparison of *RTRDM* algorithm with the latest domain adaptation algorithm on the *DAWN (sand) Dataset* (electronic amplification is recommended).

TABLE IV
ABLATION STUDY OF DIFFERENT TRAINING STRATEGIES ON THE *Foggy Cityscapes dataset*.

Model	MDA	MCR	GSA	mAP
Faster RCNN				22.0
RTRDM1	✓			30.5
RTRDM2	✓	✓		32.2
RTRDM3	✓		✓	36.8
RTRDM	✓	✓	✓	41.2

extensive experiments using *Cityscapes* as the source domain and *Foggy Cityscapes* as the target domain, as shown in Table V . Therefore, in our experiments, the best performance of *RTRDM* was achieved when $\lambda_1=0.3$ and $\lambda_2=0.1$.

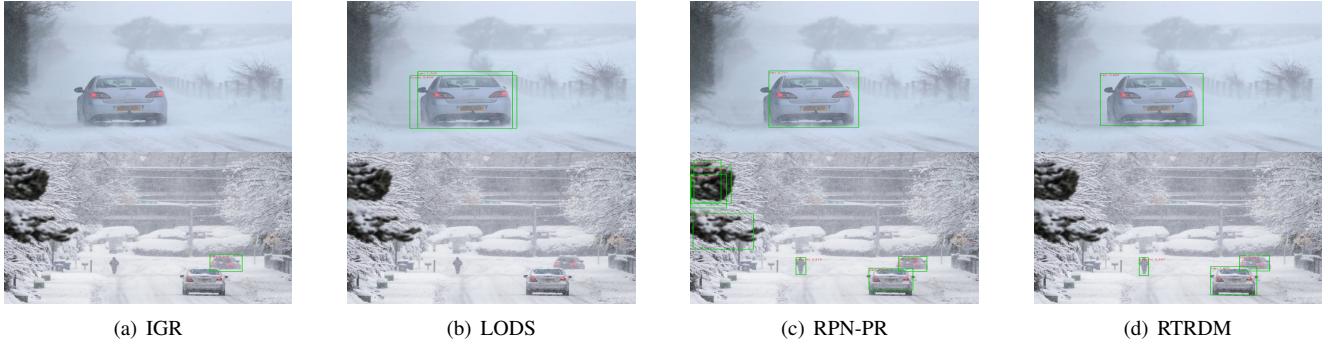


Fig. 13. Comparison of *RTRDM* algorithm with the latest domain adaptation algorithm on the *DAWN (snow) Dataset* (electronic amplification is recommended).

TABLE VI

THE RUNTIME (S) AND FPS COMPARISONS OF DIFFERENT DOMAIN ADAPTIVE DETECTION METHODS ARE TESTED ON 1024×2048 PIXEL IMAGES.

Methods	Platform	Run time	FPS
IGR	PyTorch(GPU)	0.071	14.1
LODS	PyTorch(GPU)	0.073	13.7
RPN-PR	PyTorch(GPU)	0.059	16.9
RTRDM	PyTorch(GPU)	0.056	17.9

TABLE V
ABLATION STUDY OF DIFFERENT TRAINING STRATEGIES ON THE *Foggy Cityscapes dataset*.

$\lambda_1 \& \lambda_2$	1 & 1	0.7 & 0.3	0.5 & 0.5	0.3 & 0.8	0.3 & 0.8	0.3 & 0.1
mAP	32.6	36.7	39.5	40.2	40.2	41.2

REFERENCES

- [1] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018.
- [2] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.
- [3] Zhenwei He and Lei Zhang. Domain adaptive object detection via asymmetric tri-way faster-rcnn. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pages 309–324. Springer, 2020.
- [4] Han-Kai Hsu, Chun-Han Yao, Yi-Hsuan Tsai, Wei-Chih Hung, Hung-Yu Tseng, Maneesh Singh, and Ming-Hsuan Yang. Progressive domain adaptation for object detection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 749–757, 2020.
- [5] Mingdi Hu, Long Bai, Jiulun Fan, Sirui Zhao, and Enhong Chen. Vehicle color recognition based on smooth modulation neural network with multi-scale feature fusion. *Frontiers of Computer Science*, 17(3):173321, 2023.
- [6] Mingdi Hu, Yixuan Li, Jiulun Fan, and Bingyi Jing. Joint semantic deep learning algorithm for object detection under foggy road conditions. *Mathematics*, 10(23):4526, 2022.
- [7] Mingdi Hu, Yi Wu, Yize Yang, Jiulun Fan, and Bingyi Jing. Domain adaptive faster r-cnn for vehicle object detection in rainy and foggy weather conditions. *Displays*, page 102484, 2023.
- [8] Hena Jose, Thangavel Vadivukarasi, and Jyothi Devakumar. Extraction of protein interaction data: a comparative analysis of methods in use. *EURASIP Journal on Bioinformatics and Systems Biology*, 2007(1):53096, 2007.
- [9] Mourad A Kenk and Mahmoud Hassaballah. Dawn: vehicle detection in adverse weather nature dataset. *arXiv preprint arXiv:2008.05402*, 2020.
- [10] Boyi Li, Xiulan Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng. Aod-net: All-in-one dehazing network. In *Proceedings of the IEEE international conference on computer vision*, pages 4770–4778, 2017.
- [11] Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE Transactions on Image Processing*, 28(1):492–505, 2018.
- [12] Shuaifeng Li, Mao Ye, Xiatian Zhu, Lihua Zhou, and Lin Xiong. Source-free object detection by learning to overlook domain style. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8014–8023, 2022.
- [13] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [14] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6956–6965, 2019.
- [15] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126:973–992, 2018.
- [16] Vibashan VS, Poojan Oza, and Vishal M Patel. Instance relation graph guided source-free domain adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3520–3530, 2023.
- [17] Chang-Dong Xu, Xing-Ran Zhao, Xin Jin, and Xiu-Shen Wei. Exploring categorical regularization for domain adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11724–11733, 2020.
- [18] Yixin Zhang, Zilei Wang, and Yushi Mao. Rpn prototype alignment for domain adaptive object detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12425–12434, 2021.
- [19] Zhen Zhao, Yuhong Guo, Haifeng Shen, and Jieping Ye. Adaptive object detection with dual multi-label prediction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16*, pages 54–69. Springer, 2020.