Domain Adaptive Object Detection

Yusu Fang* Xinyi Xiong* Yachun Shan*
School of Electronics Engineering and Computer Science, Peking University

Abstract

Object detection methods are usually applied to scenarios with the same distribution as the source domain. However, in many cases we apply the object detection to the target domain which has a gap with the source domain. In this case, object detection is severely affected by the domain bias, and there is a significant distributional difference between the pre-trained data and the actual application environment.In this work, in order to mitigate the performance degradation caused by the domain bias without the need of additional labeling, we need to perform unsupervised domain-adaptive object detection in the hope that the model trained on a labeled source domain and an unlabeled target domain can be well generalized to the the target domain. We mainly use the city2foggy dataset as a benchmark to take the effect of training using only the source domain data as a reference without domain adaptation. In order to improve the effect of the model on the target domain, we use the Adversarial Feature Alignment method to make the global distribution of the data in the source and target domains close. In addition, we incorporate domain adaptation through a teacher-student model and propose other methods to implement domain-adapted object detectors.

1. Introduction

In scenarios where the training domain is identically distributed with the source domain, the target detection model usually consists of a more desirable performance. However, in many cases, the inference of the target detection model is performed under a target domain that has a gap with the source domain. In this case, there is a significant distributional difference between the pre-training data and the actual application environment, and object detection is severely affected by the domain bias, which leads to a larger decrease in model accuracy. Methods to solve this problem include, but are not limited to, domain adaptation, migration learning, and GAN. we use the method of domain adaptation here to make the original model adapt to the data distribution of the target domain.

In this work, we choose the deformableDETR model as the baseline which is an end-to-end detector[2] without anchor generation, on the basis of which we realize the domain adaptation through two steps: 1) Add the feature extractor and adversarial feature aligner through the Adversarial Feature Alignment method[23]. The two networks compete with each other to reach a balance, enabling the feature extractor to generate more generalized representations.2) Knowledge distillation through the teacher-student model. The predicted results of the teacher model on the training data are used as the training goals of the student model, minimizing the goal difference between the student model and the teacher model[22]. The teacher-student framework reduces the model size and computational overhead and yields a lightweight model with reliable performance.

We compare the performance of the feature-aligned model, the model obtained by knowledge distillation, with the performance of the benchmark of SINGLE DOMAIN TRAIN. As a reference, the model trained on the source domain dataset, foggy_cityscapes (0.02), achieves an AP@50 of 0.3109. In contrast, the feature-aligned model achieves an AP@50 of 0.4431, while the teacher-student model post-distillation achieves an AP@50 of 0.5003. It can be seen that domain adaptation can significantly improve the accuracy of the model on the target domain.

2. Related Work

Object Detection: Object detection is a task to localize the object and its location given an input image. Classical work formulated this task as a sliding window classification problem. With the rise of deep convolutional networks, region-based CNNs and anchor-based approaches[18] have received significant attention due to their effectiveness. Recently, transformer-based models[30][2] are also developed in object detection. DETR [2] was introduced to achieve fully end-to-end detection and eliminate the hand-craft designed components, such as anchor generation and non-maximum

[*] Equal contribution

suppression, which attracted a surge of research interest. Following, Deformable DETR [30] develops a sparse attention module named deformable attention to fasten the convergence speed of DETR. Taking the domain adaptation issue for object detection into consideration, we employ Deformable DETR as the detector due to its simplified one-stage structure and the flexible transfer-learning ability of the transformer architecture.

Domain Adaptation Object Detection: Domain adaptation (DA) aims to learn a model from additional labeled source domain to achieve satisfactory performance on the target domain. Here we focus on the object detection problem. Different approaches of feature alignments[4] were used to align feature representation distributions and eliminate domain distribution mismatch problem. In one-stage detector rather than RCNN, aside from backbone and detector output features, more information could be used[9]. Another direction was to utilize Mean Teacher (MT)[1], which further developed into a more mature teacher-student framework. Different strategies were utilized to overcome the low quality pseudo label issue, [16] applied adversarial alignment and weak-strong augmentation, [26] proposed multi-level feature alignment to improve the pseudo labels.

3. Method

3.1. Deformable DETR

We choose deformableDETR[30] as the baseline. The deformableDETR method can be overviewed in Figure 1. It is a sequence-to-sequence framework to predict the target position by encoding the image with decoded output. We use ResNet50MultiScale as the backbone, embed the position information of the target into the sequence by PositionEncodingSine, and use DeformableTransformer[11] as the decoder to predict the anchor frame position. Attention masking mechanism is also introduced to ensure that one target is predicted for each position to prevent overlapping of targets. The deformable attention mechanism makes the model converge faster and can basically converge after 50 epochs. DeformableDETR is a single-shot detection framework, which performs well on the end-to-end target detection task.

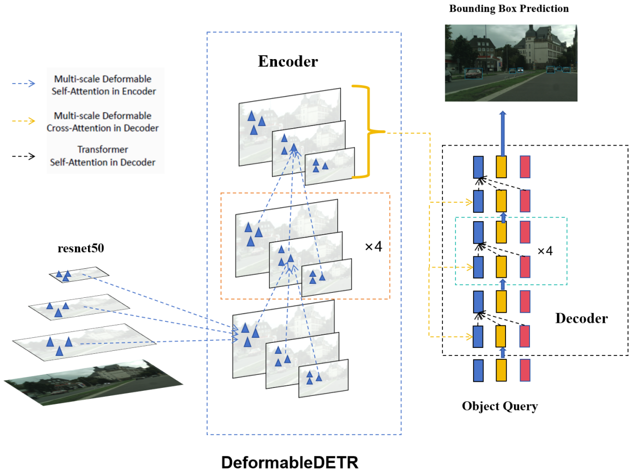
Multi-scale Deformable Self-Attention in Encoder
Multi-scale Deformable Cross-Attention in Decoder
Transformer Self-Attention in Decoder
resnet50
Encoder
×4
Bounding Box Prediction
×4
Decoder
Object Query
DeformableDETR

Figure 1. **Illustration of the proposed Deformable DETR object detector.** In deformableDETR, ResNet is used as a backbone to extract multi-scale feature maps from layer2, layer3, and layer4, and all the feature maps have 256 channels. Deformable Transformer Encoder uses the multi-scale deformable attention module to output multi-scale feature maps with the same resolution. The key values and query values are pixels in the feature map and we add position embedding and scale embedding to recognize the location of the pixel and the feature map. Deformable Transformer Decoder queries the pixels to extract the image features around the reference point by using self-attention and cross-attention modules. The detection head predicts the location of the bounding box making the decoder attention strongly correlated with the bounding box, improving the speed of convergence.

3.2. Adversarial Feature Alignment

A detector trained solely in the source domain may not perform well in the target domain. This discrepancy can be attributed to a domain gap between the two domains, meaning that the features extracted from images in each domain are significantly different. To address this issue, feature alignment can be employed to reduce the domain gap.

We applied adversarial feature alignment proposed in DA-Faster[4]. The architecture is illustrated in Figure 2. We employ two domain classifiers for image and instance levels, which discriminate features belonging to the source or target domain. These two domain adaptation components for the image and instance levels will align feature representation distributions on those two levels, thus boosting our object detection task and solving the domain shift problem.

3.2.1 Image-Level Adaptation

In this model, the image-level representation refers to the feature map outputs of 4 different layers of our model backbone, which could capture features of different scales and thus enable better domain distribution mismatch elimination. We employ a patch-based domain classifier on the multiscale feature map drawn from a ResNet backbone, so it learns to predict the domain label for image patches of different sizes. This approach could rule out distribution difference resulted from global image difference and increases the domain classifier sample size which helps with the training and domain adapting.

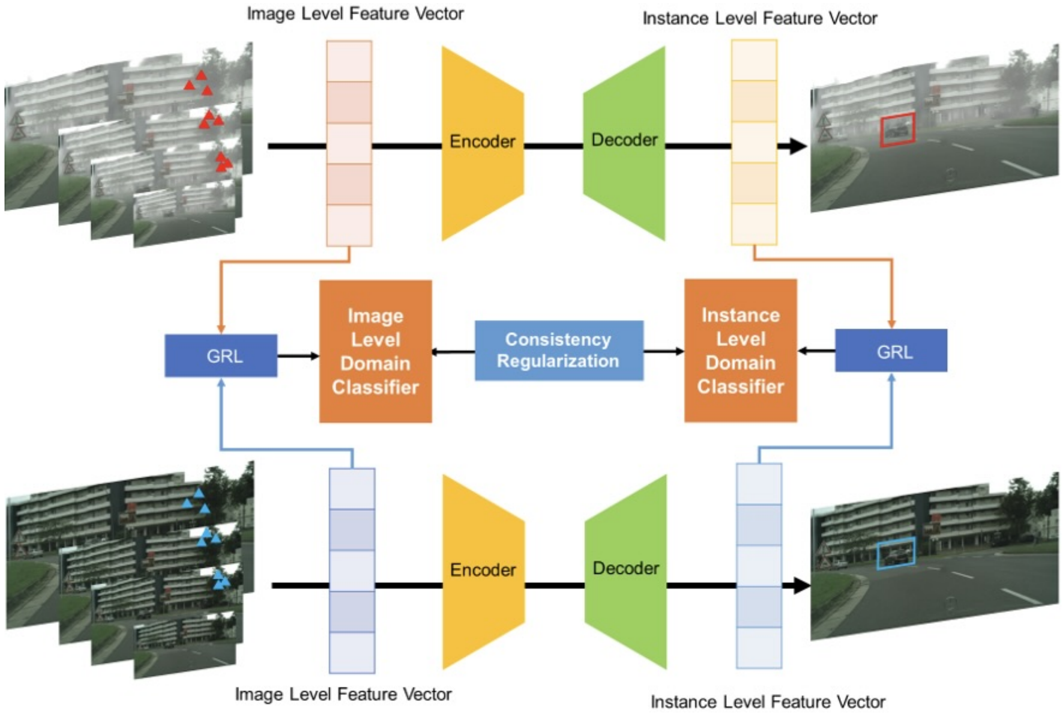
Image Level Feature Vector
Instance Level Feature Vector
Encoder
Decoder
GRL
Image Level Domain Classifier
Consistency Regularization
Instance Level Domain Classifier
GRL
Encoder
Decoder
Image Level Feature Vector
Instance Level Feature Vector

Figure 2. **An overview of our Domain Adaptive Deformable DETR model.** We tackle the domain shift on two levels, the image level and the instance level. We take the model backbone and decoder outputs as image level and instance level feature vectors. Domain classifiers are built and trained in an adversarial training manner. A consistency regularizer is incorporated within these two classifiers.

the image-level adaptation loss can be written as:

$$L_{img} = -\sum_{i,u,v}\left[D_i logp_i^{(u,v)} + (1-D_i)\, log\left(1-p_i^{(u,v)}\right)\right] \tag{1}$$

3.2.2 Instance-Level Adaptation

In this model, the instance-level representation refers to the outputs of transformer decoders, which contains the features of predicted bounding boxes. Instance-level representations aligning helps to reduce differences resulted from object appearance, size, viewpoint and so on.

$$L_{ins} = -\sum_{i,j}\left[D_i logp_{i,j} + (1-D_i)\, log\left(1-p_{i,j}\right)\right] \tag{2}$$

A gradient reverse layer before the domain classifier is necessary because it enables the adversarial training strategy, which means to improve the domain classifiers and align representations in backbone and detectors at the same time.

3.2.3 Consistency Regularization

This regularization means that the image-level domain classifier output averaged over an image should be consistent with the instance-level domain classifier output.

$$L_{cst} = \sum_{i,j} || \frac{1}{|I|}\sum_{u,v} p_i^{(u,v)} - p_{i,j}||_2 \tag{3}$$

This regularization promotes the joint adaptation of both levels and alleviate the bias in bounding box prediction.

3.3. Masked Retraining Teacher Student Framework

Some studies[25, 27] utilize the Teacher-Student framework to achieve Domain Adaptation Object Detection (DAOD). We use it as a baseline and further integrate the masked autoencoder (MAE) branch inspired by the MRT[29] model into the student model. In this approach, a portion of the target image's multi-scale feature maps is randomly masked, then fed into the encoder. Additionally, an auxiliary decoder is introduced to reconstruct the missing features from its context.Our A-T framework consists of two modules: a target-specific teacher model and a cross-domain student model, as illustrated in Figure3.

3.3.1 Teacher-student framework

The teacher model T and student model S share the same backbone, encoder, and decoder structure. The teacher model

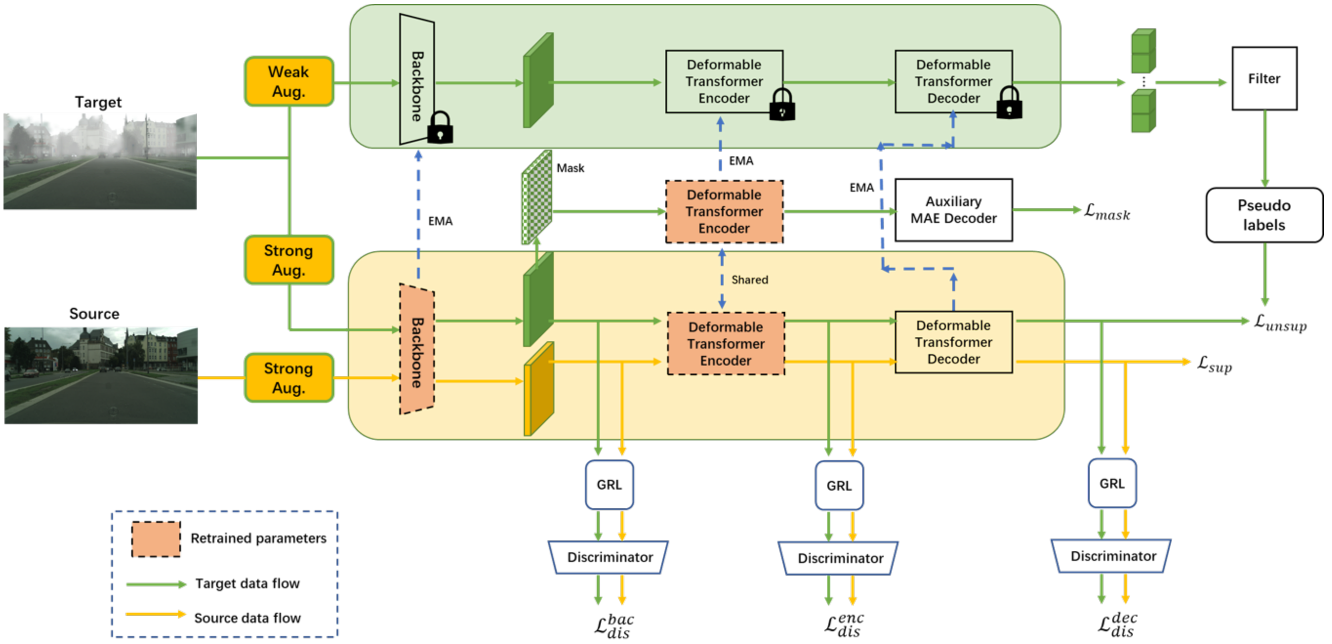

Figure 3. **Overview Masked Retraining Teacher-student framework(MRT).** Our model consists of two modules: 1) target-specific Teacher model for taking weakly-augmented images from target domain and 2) cross-domain Student model for taking strongly-augmented images from both domains. We train our model using two learning streams: Teacher-Student mutual learning and adversarial learning. MAE branch masks feature maps of target images, and and reconstructs the feature by student encoder and an auxiliary decoder. Selective retraining mechanism periodically re-initialize certain parts of the student parameters as highlighted. The Teacher model generates pseudo-labels to train the Student while the Student updates the Teacher model with exponential moving average (EMA). The discriminator with gradient reverse layer is employed to align the distributions across two domains in Student model.

extracts weakly augmented images x_t only from the target domain (D_t) and generates pseudo labels $(\hat{b_t}, \hat{c_t})$ to train the student. The student model extracts strongly augmented images from both the source and target domains (D_s and D_t) and updates the learned knowledge to the teacher using Exponential Moving Average (EMA).

The supervised loss used to train and initialize the student model with annotated source data is defined same as[30]:

$$\mathcal{L}_{\text{sup}} = \mathcal{L}^S_{\text{box}}(x_s, y_s) + \mathcal{L}^S_{\text{giou}}(x_s, y_s) + \mathcal{L}^S_{\text{cls}}(x_s, y_s) \quad (4)$$

Since there are no labels available in the target domain, we employ the method of pseudo labeling $(\hat{b_t}, \hat{c_t})$ on target domain images to generate virtual labels for training the student. Therefore, upon obtaining pseudo labels from the teacher model on target domain images, we can update the student with the loss:

$$\mathcal{L}_{\text{unsup}} = \mathcal{L}^S_{\text{cls}}(x_t, \hat{b}_t, \hat{c}_t) \quad (5)$$

The above is applicable only for the classification task[25]; unsupervised loss is not applied in the regression task.

To obtain strong pseudo labels from target images, we apply Exponential Moving Average (EMA) of the student to update the teacher model (without gradient accumulation) by temporarily copying the weights of the student model. The update formula can be defined as:

$$\theta_t \leftarrow \alpha\theta_t + (1 - \alpha)\theta_s \quad (6)$$

Where θ_t and θ_s are the model parameters of the teacher and student respectively, and α is a hyperparameter.

3.3.2 Discriminators for adversarial alignment

The approach involves setting up discriminators for different components of the detector architecture: the backbone, encoder, and decoder. These discriminators serve to provide alignment at various levels: global, token-wise multi-scale feature, and instance level respectively. The weighted sum of their losses is denoted as $\mathcal{L}_{dis}$.

The adversarial optimization objective is formulated as:

$$\mathcal{L}_{adv} = \max_S \min_D \mathcal{L}_{dis} \quad (7)$$

Where S and D denote the student and discriminators respectively. Gradient Reverse Layers (GRL)[8] are employed for the min-max optimization.

The overall objective of the student model is formulated as:

$$\mathcal{L}_{teach} = \mathcal{L}_{sup} + \mathcal{L}_{adv} + \lambda_{unsup}\mathcal{L}_{unsup} \tag{8}$$

Where λ_{unsup} is a hyper-parameter. The teacher model is updated using Exponential Moving Average (EMA), as discussed previously.

Before the teaching process, the model undergoes training with supervised loss $\mathcal{L}_{sup}$ using annotated source data D_s. Both the teacher and student models are initialized with these parameters.

3.3.3 Masked Autoencoder Branch

Drawing inspiration from MRT, we've employed a customized MAE branch design tailored for the student model. This branch enables the masking and reconstruction of multi-scale feature maps from target images. The masking and reconstruction process is outlined in detail below.

Feature Masking: Unlike ViT[7], which directly processes image patches, Deformable DETR[30] utilizes multi-scale feature maps $\{z_i \in \mathbb{R}^{C_i \times H_i \times W_i}\}_{i=1}^{K}$ derived from the backbone output. These feature maps undergo random masking $\{m_i \in \{0,1\}^{H_i \times W_i}\}_{i=1}^{K}$ to introduce sparsity. Deformable attention exploits the spatial structure of these maps to generate reference points for sparse attention. Similar to MRT[29], a zero-masking approach is employed.

Reconstruction: The masked feature maps are processed using deformable attention. An auxiliary MAE decoder reconstructs the features, focusing only on the masked portion. Mean Squared Error (MSE) is computed between the reconstructed and original features, but only on the masked portion, formulated as:

$$\hat{z_K} = S_m\left(S_e(mask(m, x_t)), q_m\right) \tag{9}$$

$$\mathcal{L}_{mask} = \mathcal{L}_{MSE}(mask(m_K, \hat{z_K}), mask(m_K, z_K)) \tag{10}$$

where S_e, S_m, q_m denotes the encoder of the student model, the auxiliary MAE decoder, and the mask query respectively.

Training Data: MAE is applied only on target images to enhance the student encoder's understanding of the target domain. This strategy avoids sub-optimal performance associated with applying MAE on source images. The teacher model does not utilize MAE to prevent incompatibility between encoder and decoder.

Training Strategy: The MAE branch and detection loss are trained simultaneously instead of following a pretrain-finetune paradigm. Initially, the model is trained with both $\mathcal{L}_{sup}$ in Equation(4) and $\mathcal{L}_{mask}$ in Equation(10) to obtain an enhanced initialization θ_{mask}. During teaching, the student model's objective is:

where $\mathcal{L}_{teach}$ has been formulated in Equation(8), and λ_{mask} is the coefficient of L_{mask} in Equation(10). the $\mathcal{L}_{mask}$ decay as teaching progresses to prevent over-fitting to the reconstruction task.

$$\mathcal{L} = \mathcal{L}_{teach} + \lambda_{mask}\mathcal{L}_{mask} \tag{11}$$

4. Experiments

4.1. Datasets

The benchmark we chose is city2foggycityscapes (11G) dataset[5] which contains 5000 images and is divided into training, testing and validation sets. Foggy_cityscapes (30G) is obtained using simulated fogging processed on each image of cityscapes and is also divided into training, testing and validation set, where 0.02, 0.01, and 0.005 represent the degree of fogging from high to low[20]. Cityscapes dataset is used as source domain, and foggy_cityscapes(0.02) is used as target domain. We perform weak augmentation and strong augmentation on the dataset, both on the basis of normalization respectively RandomHorizontalFlipImgAnno(p=0.5), ResizeImgAnno(size=800, max_size=1333) and RandomApplyImgAnno([ColorJitterImgAnno(0.4, 0.4, 0.4, 0.1)], p=0.8), RandomGrayScaleImgAnno(p=0.2), RandomApplyImgAnno([GaussianBlurImgAnno([0.1, 2.0])], p=0.5) and convert the dataset to coco format.

4.2. Implementation Details

We use DETR[30] as the base detector. In adversarial feature alignment, we set the weights for calculating domain loss at image level, instance level loss, and consistency regularization to 1.0, 1.0, and 0.1 respectively. In the teacher-student model, we set $\lambda_{unsup} = 1.0$ and initial $\lambda_{mask} = 1.0$. In EMA, we set the weight smoothing parameter $\alpha = 0.9996$. For dynamic thresholds, we initialize the threshold for each class as 0.3. For the MAE branch, we use a 2-layer asymmetric decoder with a mask ratio of 0.8. We optimize the network using the Adam optimizer with an initial learning rate of 2×10^{-4} and a batch size of 4. Weak augmentation data augmentation methods include random horizontal flipping, while strong augmentation methods include random color jitter, grayscale, and Gaussian blur.

4.3. Comparison

Comparing with baseline: As is shown in Table 1, compared with training on source domain only, the feature alignment and Masked Retraining Teacher Student Frame- work approaches achieves better performance with less training epochs but the runtime speed is slower. Compared to the baseline (*deformable DETR only*), using the adversarial feature alignment method can effectively improve detection accuracy, with an increase in mAP of 12.1% . Additionally, using the Masked Retraining Teacher-Student Framework compared to the baseline can increase mAP by 19% , and compared to the adversarial feature alignment, it can increase mAP by 7% .

		mAP50	person	car	train	rider	truck	motorcycle	bicycle	bus	training GPU hour
source only		31.1	40.6	49.3	0.8	45.1	14.3	20.0	43.4	35.2	6
feature align	original	43.2	47.1	65.7	28.8	52.6	31.8	36.1	47.2	50.2	11
	image only	44.3	46.3	62.7	35.4	49.7	31.8	34.6	47.6	46.5	9
	instance only	42.8	46.9	64.8	27.2	49.0	31.6	27.0	47.3	48.1	11
MRT		50.0	51.4	67.3	44.2	55.0	33.3	38.6	51.4	58.1	17(6 for S + 11 for T)

Table 1. **Comparison of Deformable DETR trained without domain adaptation and with different levels of feature alignment.**

Method	Detector	person	rider	car	truck	bus	train	mcycle	bicycle	mAP
FasterRCNN[19](Source)	FRCNN	26.9	38.2	35.6	18.3	32.4	9.6	25.8	28.6	26.9
DA-Faster[4]	FRCNN	29.2	40.4	43.4	19.7	38.3	28.5	23.7	32.7	32.0
UMT[6]	FRCNN	33.0	46.7	48.6	34.1	56.5	46.8	30.4	37.3	41.7
TIA[28]	FRCNN	34.8	46.3	49.7	31.1	52.1	48.6	37.7	38.1	42.3
D-adapt[14]	FRCNN	40.8	47.1	57.5	33.5	46.9	41.4	33.6	43.0	43.0
SIGMA[15]	FRCNN	44.0	43.9	60.3	31.6	50.4	**51.5**	31.7	40.6	44.2
AT^{1}[16]	FRCNN	43.7	54.1	62.3	31.9	54.4	49.3	35.2	47.9	47.4
TDD[10]	FRCNN	50.7	53.7	**68.2**	**35.1**	53.0	45.1	**38.9**	49.1	49.2
PT[3]	FRCNN	40.2	48.8	63.4	30.7	51.8	30.6	35.4	44.5	42.7
FCOS[21](Source)	FCOS	36.9	36.3	44.1	18.6	29.3	8.4	20.3	31.9	28.2
EPM[12]	FCOS	44.2	46.6	58.5	24.8	45.2	29.1	28.6	34.6	39.0
SSAL[17]	FCOS	45.1	47.4	59.4	24.5	50.0	25.7	26.0	38.7	39.6
Def DETR[30] (Source)	Def DETR	40.6	45.1	49.3	14.3	35.2	0.8	20.0	43.4	31.1
SFA[24]	Def DETR	46.5	48.6	62.6	25.1	46.2	29.4	28.3	44.0	41.3
MTTrans[26]	Def DETR	47.7	49.9	65.2	25.8	45.9	33.8	32.6	46.5	43.4
O^{2}net[9]	Def DETR	48.7	51.5	63.6	31.1	47.6	47.8	38.0	45.9	46.8
AQT[13]	Def DETR	49.3	52.3	64.4	27.7	53.7	46.5	36.0	46.4	47.1
Feature Align(task2)	Def DETR	46.8	50.8	65.3	31.8	50.2	21.1	32.4	47.1	43.2
MRT(task3)	Def DETR	**51.4**	**55.0**	67.3	33.3	**58.1**	44.2	38.6	**51.4**	**50.0**

Table 2. **Results of Cityscapes to *Foggy Cilyscapes(0.02)*.** ”FRCNN” denotes Faster R-CNN and “DefDETR” denotes Deformable DETR

Comparing with other methods: We compared the feature alignment approach and MRT with other methods on the benchmark mentioned above. It can be observed that MRT outperforms previous methods and achieves significant improvements compared to DETR-based methods. As shown in Table 2, for categories with fewer instances (i.e., ”rider”) in *Cityscapes* to *Foggy Cityscapes*, MRT performs much better, as the data-efficient MAE branch boosts the performance. For confusing categories (i.e., ”bicycle” and ”motorcycle”), MRT shows a significant performance gain with the help of the selective retraining mechanism.

4.4. Ablation Study

Table 1 presents ablations for various design choices of domain adaptation modules: image-level adaptation only, instance-level adaptation only and the original setting with both level adaptation and consistency constraint. Using these feature alignment approaches can effectively improve detection accuracy with 12.1%, 13.2% and 11.7% AP. It is unexpected that image-level feature alignment only can lead to the highest detection accuracy , with 1.1% AP better than the original setting. Using instance level adaptation on decoder outputs does not improve the performance as well as image-level adaptation do. This could be attributed to improper choice of parameters or improper choice of features. Detailed convergence curves are shown in Figure 4. As shown in Table 3, we can see the following: 1) Intro-

Source	Baseline	DT	Retrain	MAE	mAP
✓					31.9
✓	✓				44.9
✓	✓	✓			51.9
✓	✓	✓		✓	51.5
✓	✓	✓	✓	✓	50.0

Table 3. **Ablation studies of proposed modules on Cityscapesto Foggy Cityscapes.** ”Source” denotes the source-only trainedmodel. ”Baseline” denotes the adaptive teacher-student baseline. ”DT”, ”Retrain” and ”MAE” denotes proposed dynamic threshold, selective retraining and masked autoencoder branch, respectively.

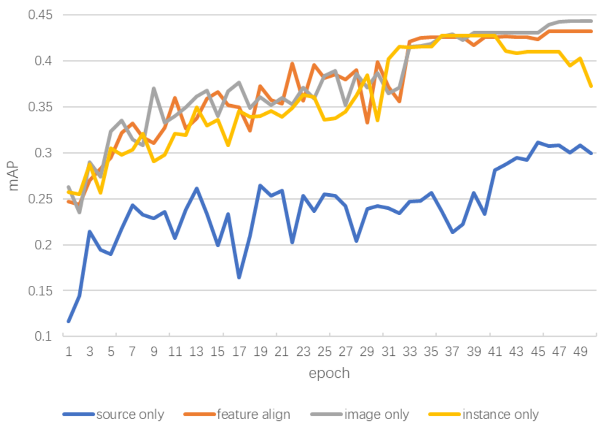

Figure 4. **Convergence curves of Deformable DETR trained without domain adaptation and with different levels of feature alignment.** It should be noted that in the source-only training, we reduced the learning rate at the 40th epoch while in feature alignment training at the 30th to improve performance.

ducing the teacher-student framework (line 2) significantly improves performance compared to models trained only on the source, indicating that the teacher-student framework is an effective semi-supervised learning method that can achieve significant accuracy gains in cross-domain object detection. 2) Performance is greatly improved when adding the MAE branch and dynamic filtering (lines 3-4), demonstrating their effectiveness. 3) When introducing selective retraining (line 7), performance does not improve further. Compared to introducing only DT and introducing both DT and the MAE branch simultaneously, the accuracy decreases, indicating that the reasons behind this phenomenon require further investigation.

5. Conclusion

This paper investigates the implementation of Domain Adaptive Object Detection using three methods: Deformable DETR, Adversarial Feature Alignment, and the Masked Retraining Teacher-Student Framework. It can be seen that feature alignment, by discriminating features belonging to the source or target domain, can effectively improve detection accuracy. Meanwhile, the MAE branch in MRT helps the student model better capture target domain features and acquire knowledge from a limited number of pseudo boxes. Experimental results demonstrate that MRT can effectively enhance the model's domain adaptation ability.

References

[1] Qi Cai, Yingwei Pan, Chong-Wah Ngo, Xinmei Tian, Lingyu Duan, and Ting Yao. Exploring object relation in mean teacher for cross-domain detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11457–11466, 2019. 2

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1

[3] Meilin Chen, Weijie Chen, Shicai Yang, Jie Song, Xinchao Wang, Lei Zhang, Yunfeng Yan, Donglian Qi, Yueting Zhuang, Di Xie, and Shiliang Pu. Learning domain adaptive object detection with probabilistic teacher, 2022. 6

[4] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018. 2, 6

[5] 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. *Proceedings of the IEEE conference on computer vision and pattern recognition*, abs/1604.01685:3213–3223, 2016. 5

[6] Jinhong Deng, Wen Li, Yuhua Chen, and Lixin Duan. Unbiased mean teacher for cross-domain object detection, 2021. 6

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *International Con- ference on Learning Representations*, abs/2010.11929, 2020. 5

[8] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation, 2015. 4

[9] Kaixiong Gong, Shuang Li, Shugang Li, Rui Zhang, Chi Harold Liu, and Qiang Chen. Improving transferability for domain adaptive detection transformers. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1543–1551, 2022. 2, 6

[10] Mengzhe He, Yali Wang, Jiaxi Wu, Yiru Wang, Hanqing Li, Bo Li, Weihao Gan, Wei Wu, and Yu Qiao. Cross domain object detection by target-perceived dual branch distillation, 2022. 6

[11] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019. 2

[12] Cheng-Chun Hsu, Yi-Hsuan Tsai, Yen-Yu Lin, and Ming-Hsuan Yang. Every pixel matters: Center-aware feature alignment for domain adaptive object detector. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 733–748. Springer, 2020. 6

[13] Wei-Jie Huang, Yu-Lin Lu, Shih-Yao Lin, Yusheng Xie, and Yen-Yu Lin. Aqt: Adversarial query transformers for domain adaptive object detection. In *31st International Joint Conference on Artificial Intelligence, IJCAI 2022*, pages 972–979. International Joint Conferences on Artificial Intelligence, 2022. 6

[14] Junguang Jiang, Baixu Chen, Jianmin Wang, and Mingsheng Long. Decoupled adaptation for cross-domain object detection, 2022. 6

[15] Wuyang Li, Xinyu Liu, and Yixuan Yuan. Sigma: Semantic-complete graph matching for domain adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5291–5300, 2022. 6

[16] Yu-Jhe Li, Xiaoliang Dai, Chih-Yao Ma, Yen-Cheng Liu, Kan Chen, Bichen Wu, Zijian He, Kris Kitani, and Peter Vajda. Cross-domain adaptive teacher for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7581–7590, 2022. 2, 6

[17] Muhammad Akhtar Munir, Muhammad Haris Khan, M. Saquib Sarfraz, and Mohsen Ali. Synergizing between self-training and adversarial learning for domain adaptive object detection, 2021. 6

[18] T. Darrell R. Girshick, J. Donahue and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CVPR*, 12(1):221–334, 2014. 1

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. 6

[20] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126:973–992, 2018. 5

[21] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection, 2019. 6

[22] Jiabao Wang, Yuming Chen, Zhaohui Zheng, Xiang Li, Ming-Ming Cheng, and Qibin Hou. Crosskd: Cross-head knowledge distillation for dense object detection. *arXiv preprint arXiv:2306.11369*, 2023. 1

[23] Wen Wang, Yang Cao, Jing Zhang, Fengxiang He, Zheng-Jun Zha, Yonggang Wen, and Dacheng Tao. Exploring sequence feature alignment for domain adaptive detection transformers. In *Proceedings of the 29th ACM International Conference on Multimedia*. ACM, 2021. 1

[24] Wen Wang, Yang Cao, Jing Zhang, Fengxiang He, Zheng-Jun Zha, Yonggang Wen, and Dacheng Tao. Exploring sequence feature alignment for domain adaptive detection transformers. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1730–1738, 2021. 6

[25] Minghao Xu, Hang Wang, Bingbing Ni, Qi Tian, and Wenjun Zhang. Cross-domain detection via graph-induced prototype alignment. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, abs/2003.12849: 12355–12363, 2020. 3, 4

[26] Jinze Yu, Jiaming Liu, Xiaobao Wei, Haoyi Zhou, Yohei Nakata, Denis Gudovskiy, Tomoyuki Okuno, Jianxin Li, Kurt Keutzer, and Shanghang Zhang. Cross-domain object detection with mean-teacher transformer. *arXiv preprint arXiv:2205.01643*, 2022. 2, 6

[27] Jinze Yu, Jiaming Liu, Xiaobao Wei, Haoyi Zhou, Yohei Nakata, Denis Gudovskiy, Tomoyuki Okuno, Jianxin Li, Kurt Keutzer, and Shanghang Zhang. Mttrans: Cross-domain object detection with mean-teacher transformer, 2022. 3

[28] Liang Zhao and Limin Wang. Task-specific inconsistency alignment for domain adaptive object detection, 2022. 6

[29] Zijing Zhao, Sitong Wei, Qingchao Chen, Dehui Li, Yifan Yang, Yuxin Peng, and Yang Liu. Masked retraining teacher-student framework for domain adaptive object detection. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19039–19049, 2023. 3, 5

[30] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 1, 2, 4, 5, 6

6. Contributes

Yusu Fang:

Code: Task 3

milestone: technical part

Final report: In the methods section for Task 3(section 3.3), explain our approach in detail and include a structural diagram. In the experiments section for Task 3(section 4.2-4.4),provide the implementation details, compare our results with other methods and present the findings of the ablation study. Additionally, provide the data results for tabulation and summary the whole work(section 5). Finally, use Latex to finish this report.

Xinyi Xiong:

Code: Task 2

milestone: introduction and data part

Final report: Related works section(section 2), In the methods section for Task 2(section3.2), explain our approach in detail and include a structural diagram. In the experiments section for Task 2(section 4.3, 4.4), compare our results with other methods and present the findings of the ablation study.

Yachun Shan:

Code: Task 1

milestone: result part

Final report: Abstract and Introduction section(section 1), In the methods section for Task 1(section 3.1), explain our approach in detail and include a structural diagram. In experiments section, outline the datasets used in our experiments(section 4.1).

7. Configs

Requirements:

pip install -r requirements.txt

Download the pre-trained model provided by Deformable DETR through the link and put it in the main folder: https://pan.baidu.com/s/14LK_9zRgzMOEL-nGaTlzNw?pwd=hme0

Compiling Deformable DETR CUDA operators: cd ./models/ops sh ./make.sh

Training and Evaluation:

source_only

sh configs/def-detr-base/city2foggy/source_only.sh

feature alignment

sh configs/def-detr-base/city2foggy/feature_align_both.sh

use image level feature alignment only:

sh configs/def-detr-base/city2foggy/feature_align_img_only.sh

use instance level feature alignment only:

sh configs/def-detr-base/city2foggy/feature_align_ins_only.sh

Teacher and student model:

```
sh configs/def-detr-base/city2foggy/source_only.sh
sh configs/def-detr-base/city2foggy/teaching.sh
```