MS2A: Memory Storage-to-Adaptation for Cross-domain Few-annotation Industrial Detection

Anonymous submission

Abstract

Cross-domain few-annotation object detection (CFOD) aims to transfer a detector trained on the source domain to the target domain with less annotated data, which is a common problem that needs to be solved urgently in the industrial fields. The main challenge lies in that it is difficult to maintain the performance of the source domain when the annotation of target domain data is severely insufficient and its environment is very different from the source domain. Current methods primarily concentrate on addressing the data imbalance problem by using source data to augment the target data or aligning the target domain features with the source data. However, their gains are constrained as they solely focus on the limited labeled target data, which cannot fully represent the entire target domain. Moreover, they fail to consider the valuable prior information present in the extensive unlabeled target data. In this paper, we propose a novel **m**emory **s**torage-**t**o-**a**daptation (MS2A) mechanism to effectively mine prior knowledge as memory and transfer the memory to both domains adaptively. Specifically, our approach involves two key components. On the one hand, we introduce a prior memory storage module, which comprehensively captures the prior knowledge from both source and unlabeled target data and stores it as memory for later domain adaptation. On the other hand, we present an efficient memory adaptation module that integrates the stored memory into the feature alignment of target domain data, resulting in more discriminative features and improving the detection performances significantly. To evaluate the effectiveness of our proposed method, we have curated a new CFOD dataset specifically designed for industrial scenes. Our method is assessed on both the constructed dataset and publicly available datasets and the experimental results demonstrate that we achieve new state-of-the-art performances on both sets of datasets.

1 Introduction

Domain adaptation poses a significant challenge in industrial object detection, as the performance of a trained detector often drops noticeably when applied to a new domain. Due to the substantial variances of object appearance and backgrounds (Liu et al. 2022a,b; Mirza et al. 2022; Wu and Deng 2022; Xu et al. 2022; Yoo, Chung, and Kwak 2022; Gao et al. 2022) in the industrial environments, it is almost impossible to maintain the performance of the model without fine-tuning. On the other hand, collecting the labeled data from the target domain is time-consuming and expensive, therefore, it is important to design a versatile cross-domain few-annotation object detection (CFOD) method for quickly deploying models on new domains.

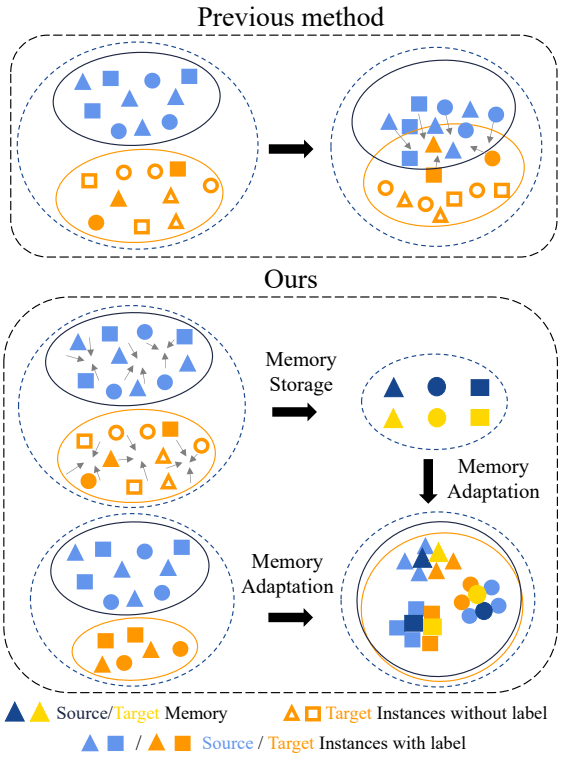

Figure 1: Illustration of the proposed MS2A. In general, previous methods (Long et al. 2015, 2016; Wang et al. 2019; Sugiyama et al. 2007) align the source domain data with the limited labeled target data, however, the limited labeled target data can not represent the whole target domain. Differently, this paper presents a novel memory storage-to-adaptation mechanism that first learns prior memory from the source data and massive unlabeled target data, and then utilize the memory to align the source data with the target data adaptively.

There are two main strategies to address the domain adaptation problem in the few-annotation setting. One line of work (Chen et al. 2020, 2018; He and Zhang 2019; Kim et al. 2019; Saito et al. 2019) attempts to minimize the domain disparity (Long et al. 2015, 2016, 2017; Sugiyama et al. 2007) and reach appealing transfer-ability (Gretton et al. 2012; Jin et al. 2020; Long et al. 2015, 2016) via well-designed modules, aligning the features from source domain and target domain. On the other hand, some studies (Bochkovskiy, Wang, and Liao 2020; Yun et al. 2019; Zhang et al. 2017; Dwibedi, Misra, and Hebert 2017) focus on utilizing the existing data to augment the data of the target domain. To generate new target domain data, they can mix the data of different domains in image-level (Bochkovskiy, Wang, and Liao 2020; Yun et al. 2019; Zhang et al. 2017) and box-level (Dwibedi, Misra, and Hebert 2017), or emerge generative models to transfer the source domain data to the target domain (Zhu et al. 2017). In short, existing methods aims to align the distribution and features of the source data with the target data.

However, existing CFOD methods are insufficient to bring further improvements in industrial scenes because of the following reasons. First and foremost, current methods focus on aligning the source data with the limited labeled target data, yet the limited target data can not represent the whole domain as shown in Fig. 1. At the same time, they neglect to mine important prior knowledge from the readily available unlabeled target data. Secondly, compared with conventional scenes, the object and background changes in industrial scenes are more complex, which makes it difficult to transfer the previous methods into cross-domain few-annotation industrial detection. Moreover, most of current benchmarks (Cordts et al. 2016; Sakaridis, Dai, and Van Gool 2018; Johnson-Roberson et al. 2016; Geiger, Lenz, and Urtasun 2012) are built upon urban scenes, which limits the application of CFOD in industrial scenes. It is unsuitable to transfer the detector trained on previous datasets to industrial scenarios.

To meet these challenges, we propose a pioneering memory storage-to-adaptation mechanism as depicted in Fig. 1 to mine the prior knowledge comprehensively and transfer it to the target environment adaptively. In the memory storage stage, we introduce a prior memory module that utilizes both the source domain data and unlabeled target data to learn the prior knowledge and store it as memory. On the other hand, to adapt the acquired prior knowledge memory to the target domain, we propose an additional memory adaptation module that can efficiently align the source data with the target one and extract discriminative features for the limited annotated target data. Since we use the prior knowledge, which almost represents the whole target domain, to guide the feature alignment adaptively, our method significantly improves detection performance on the target domain. Furthermore, to bridge the gap of CFOD in industrial settings, we construct a new industrial dataset with complex foreground and background variations.

Specifically, we first utilize the abundant source domain data to learn a prior base detector that is used to extract prior knowledge for both the source domain and target domain. Then, the prior knowledge is refined through clustering and momentum updates and stored as a final memory. In the subsequent memory adaptation stage, we introduce an efficient window cross-attention mechanism that transfers the prior memory into feature alignment adaptively. Through adaptive transfer, we can extract more discriminative features that are tailored to the specific characteristics of each domain. In summary, the main contributions of this work are as follows:

- We propose a novel memory storage-to-adaptation mechanism to learn the prior knowledge and transfer it to feature alignment adaptively. To the best of our knowledge, this is the first work to extract the prior knowledge of unlabeled target data to address the CFOD task.
- We construct a new challenging benchmark to bridge the gap of CFOD in industrial scenarios.
- Experiments show that our method achieves new state-of-the-art performances on both public datasets and the proposed industrial dataset.

2 Related Work

Cross-domain Object detection. Cross-domain object detection aims to maintain the detection performance on the target domain in which the detector is trained on the source domain, which mainly includes data-driven and algorithm-driven approaches. The key idea of data-driven methods (Gretton et al. 2012; Long et al. 2015; Jin et al. 2020) is utilizing the source domain data to augment the target domain data, minimizing the gap of distributions of the two domains. On the other hand, the algorithm-driven approaches adopt adversarial feature alignment and minimize the cross-domain discrepancy to bridge the domain gap. Early works (Saito et al. 2019; Zheng et al. 2020) align the global features with diverse mechanisms such as the spatial attention (Zheng et al. 2020) and the strong-weak alignment (Saito et al. 2019). Recent studies (Kingma and Welling 2013; Van den Oord et al. 2016) introduce the category-level adaptation in class-conditional distributions to align the features in a more fine-grained way. In this paper, we propose a novel memory storage-to-adaptation mechanism that utilizes the prior knowledge memory to extract more discriminative features for the target domain.

Few-shot object detection. Few-shot object detection involves detecting novel objects with only a few annotated instances after pretraining on abundant publicly available data. In this context, three widely adopted techniques are used for few-shot object detection. Methods (Yoo et al. 2018; Kozerawski and Turk 2018; Keshari et al. 2018) based on transfer learning involve re-using network weights pre-trained on a baseline dataset to improve generalization capabilities on a new domain with limited data; the metric learning-based approaches (Koch et al. 2015; Vinyals et al. 2016; Snell, Swersky, and Zemel 2017; Sung et al. 2018; Yan et al. 2019) aim to learn an embedding space in which inputs with similar content are encoded as features that are close to each other while dissimilar inputs are supposed to be far apart; and the meta-learning approaches (Finn, Abbeel, and Levine 2017; Ren et al. 2018; Lee et al. 2019; Li and Li 2021; Hu et al. 2021; Han et al. 2022) focus on learning how to learn, enabling generalization for new tasks or new data. To trans-

fer the prior knowledge to the target domain, we incorporate meta-learning into cross-domain few-annotation detection to let the network learn the memory adaptation process.

Cross-domain Few-annotation Object Detection. Related studies (Wang et al. 2019; Zhong et al. 2022; Gao et al. 2022; Zhao, Meng, and Xu 2022; Gao et al. 2023) on cross-domain few-annotation object detection usually follow the techniques of cross-domain detection and few-shot detection. PICA (Zhong et al. 2022) proposes to exploit point-wise alignment over instances to tackle the problem of scarcity of labeled target instances. OA-FSUI2IT (Zhao, Meng, and Xu 2022) proposes an unsupervised image-to-image translation to synthesize the target domain data. Recent AsyFOD (Gao et al. 2023) aligns feature distributions in an asymmetric way, which leverages the source and target instances from different perspectives. However, they do not consider the prior knowledge contained by the massive unlabeled target data. In this work, we propose to mine the prior knowledge as the memory and utilize it to guide the target domain detection with few annotations. Besides, different from previous methods that focus on urban scenes, we also address the more challenging industrial environments.

3 Proposed Method

The overall pipeline of MS2A consists of two key components that are illustrated in Fig. 2. We first recap the problem formulation in Sec. 3.1. Then, we present the proposed MS2A framework in Sec. 3.2, including the memory storage module, memory adaptation module, and training objective.

3.1 Problem Formulation

In general CFOD setting, we have two separate sets: one is the source domain data with lots of annotations $\mathcal{D}^s = \{(x_i^s, y_i^s)\}_{i=1}^{N^s}$ and the other is the target domain data with limited annotations $\mathcal{D}^t = \{(x_i^t, y_i^t)\}_{i=1}^{N^t}$. Here, x_i^s, x_i^t represents the i-th image in the source domain and target domain respectively, $y_i^{s,t} = (b_i^{s,t}, c_i^{s,t})$ is the corresponding labels where $b_i^{s,t}$ and $c_i^{s,t}$ are the coordinates of bounding box and associated category. The goal of CFOD is to achieve satisfactory performance on the target domain when there are sufficient annotations in the source domain but not the target domain. In actual situation, we can easily acquire adequate unlabeled target data $\{\hat{x}_i^t\}_{i=1}^{N_u^t}, N_u^t \gg N^t$ that contains rich prior information of the whole target domain. However, previous methods mainly align the source data with the very limited target one, ignoring the prior knowledge present in the unlabeled target data.

3.2 Framework of MS2A

Different from previous methods, we propose to mine prior knowledge from the massive source data and unlabeled target data as depicted in Fig. 2. Specifically, our framework consists of two key components. In the memory storage module, we learn the prior memory from lots of source data and unlabeled target data. Then, the memory adaptation module uses the memory as guidance and align the features of different domains adaptively.

Memory Storage The goal of the memory storage module is to learn prior knowledge from the source data and the unlabeled target data and store the prior knowledge as memory. We intend to let the memory be informative and not resource-intensive, therefore, we select to extract memory in feature space rather than the original image space. Given the source data $\mathcal{D}^s$ with annotations, we first conduct a prior learning process in that we train a base detector using the source data. The prior learning can obtain a simple yet effective feature extractor, which absorbs the prior knowledge from the source domain and possesses the base ability to detect similar objects. We then use the base detector to extract features for both source and unlabeled target data and call the features prior knowledge. In practice, we use the YOLOX (Ge et al. 2021) as the base detector and extract the multi-scale features after the neck to obtain prior knowledge from different scales. The process can be formulated by:

$$\{K_{ij}^d\}_{j=1}^3 = \mathcal{B}(x_i^d), d = \{s, t\}, K_{ij}^d \in \mathbb{R}^{\frac{H}{2^{j-1}} \times \frac{W}{2^{j-1}} \times C} \tag{1}$$

where $\mathcal{B}$ is the backbone+neck of the base detector, x_i^d represents the image, $\{K_{ij}^d\}$ represents the multi-scale features extracted by $\mathcal{B}$, j and d index to scale and domain respectively, $j = 1, 2, 3$ since YOLOX outputs three multi-scale features and the default (H, W, C) is $(80, 80, 256)$. The source prior knowledge K_{ij}^s is naturally similar to the target prior knowledge K_{ij}^t as they are extracted by the same base detector, which means they are expectant guides to align the two different domains.

Though we obtain the prior knowledge in feature space, they are so heavy that there are $N^s \times \{K_{ij}^s\}_{j=1}^3$ in the source domain and $N_u^t \times \{K_{ij}^t\}_{j=1}^3$ in the target domain. To acquire a refined representation, we consider clustering the prior knowledge since the previous studies (Ding et al. 2022; Li et al. 2022) have shown the cluster center is an efficient and lightweight representation to stand for a set of data. In particular, we concatenate $\{K_{ij}^d\}_{j=1}^3$ from the same domain together, resulting in $\{\hat{K}_j^s\}_{j=1}^3 \in \mathbb{R}^{N^s \times \frac{H}{2^{j-1}} \times \frac{W}{2^{j-1}} \times C}$ and $\{\hat{K}_j^t\}_{j=1}^3 \in \mathbb{R}^{N_u^t \times \frac{H}{2^{j-1}} \times \frac{W}{2^{j-1}} \times C}$. We expect the refined memory to remain in the spatial dimensions $(H \times W)$ so that it can contain the prior object and background information at the same time. In practice, we reshape $\{\hat{K}_j^s\}_{j=1}^3$ and $\{\hat{K}_j^t\}_{j=1}^3$ to have the size of $N^s C \times \frac{H}{2^{j-1}} \frac{W}{2^{j-1}}$ and $N_u^t C \times \frac{H}{2^{j-1}} \frac{W}{2^{j-1}}$, which means there are $N^s C$ and $N_u^t C$ samples in the source domain and target domain for each scale level, and each sample of the j-th scale is a vector with size $1 \times \frac{H}{2^{j-1}} \frac{W}{2^{j-1}}$. Afterward, we cluster these samples into k classes by the KNN algorithm (Cover and Hart 1967) and use the cluster centers as the memory prototypes, which are reshaped back to be spatial feature maps. The process can be formulated by:

$$\hat{M}_j^d = \mathbf{Reshape}(\mathbf{KNN}(\hat{K}_j^d)), \hat{M}_j^d \in \mathbb{R}^{\frac{H}{2^{j-1}} \times \frac{W}{2^{j-1}} \times k}. \tag{2}$$

Now we have extracted two memory prototypes that assemble the spacial information including the foreground and background of the whole source and target domains. However, it is still inadequate as the memory to guide the subse-

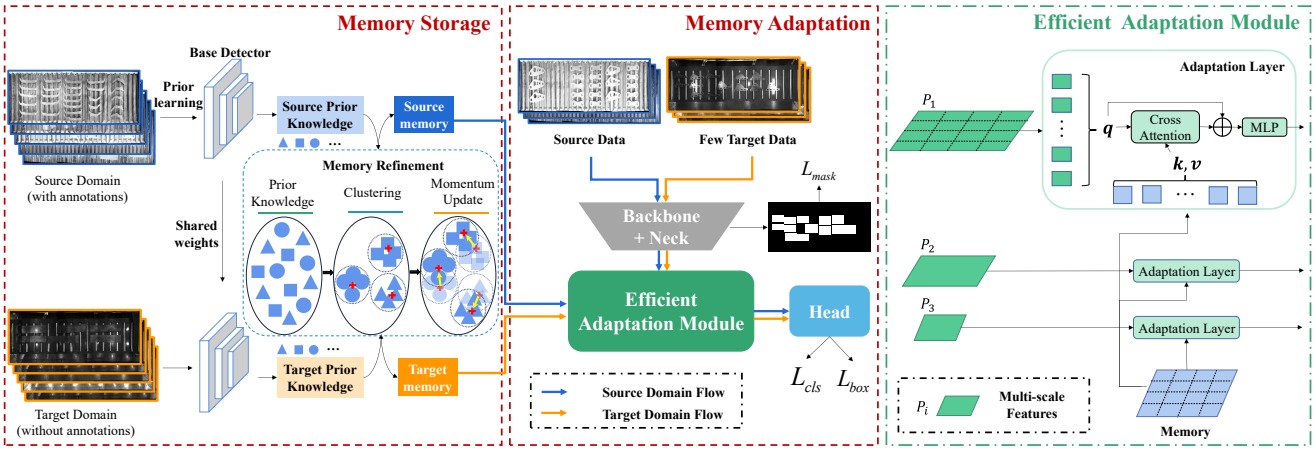

Figure 2: Overall pipeline. The proposed MS2A consists of two core parts. In the memory storage module, we first use the source data to train a base detector (we use YOLOX (Ge et al. 2021) in practice and name this process prior learning), and use the base detector to extract prior knowledge for both source data and unlabeled target data. The prior knowledge is refined through clustering and momentum updates and stored as memory. On the other hand, in the memory adaptation module, we introduce an efficient adaptation module, which utilizes the memory to align the source data with the target data adaptively.

quent adaptation process. Since the memory prototypes are fixed after clustering, they may become ineffective gradually as more and more new images are added to the target domain. To address the problem, we propose to add learnable momentum variables to the memory prototypes so that they can be updatable, which is represented by:

$$M_j^d = \hat{M}_j^d + \Delta_{mom} \tag{3}$$

where Δ_{mom} is the learnable momentum variable with the same shape as $\hat{M}_j^d$, and M_j^d is the final memory that can be directly used for the memory adaptation module. The above process is shown in the left box of Fig. 2.

Memory Adaptation After obtaining the memory, the memory adaptation module aims at transferring it into feature alignment for both domains adaptively. To this end, we propose a novel efficient adaptation module to associate the memory with the multi-scale features of both source and target domains. As shown in the middle box of Fig. 2, inspired by meta-learning (Vinyals et al. 2016), which lets the network learn strong adaptability on new tasks with limited data, we mix the source and limited labeled target data in a meta-learning manner, allowing the detector learning domain adaptation. In particular, the feature alignment of different domains is guided by the memory corresponding to the domain because of the following two reasons. Intuitively, the memory and features of the same domain are more compatible, so it is more suitable to use the memory of the corresponding domain to guide the feature alignment. On the other hand, since the source domain memory (M_j^s) and the target domain memory (M_j^t) are naturally close, we can align features from different domains together by aligning the features to the memory of the corresponding domain.

Specifically, the training data is first fed into the backbone and the neck architecture (that is inherited from the base detector) to extract multi-scale features. We use $\{P_j^d\}_{j=1}^3, P_j^d \in \mathbb{R}^{\frac{H}{2^{j-1}} \times \frac{W}{2^{j-1}} \times C}$ to stand for the extracted multi-scale features, where d and j index to the domain and scale respectively. Then, the efficient adaptation module associates the memory with the multi-scale features, aligning them together.

Efficient Adaptation Module. It has been demonstrated by existing studies (Vaswani et al. 2017; Dosovitskiy et al. 2020) that the attention mechanism has a strong association ability, therefore we choose the attention mechanism as our association operator. However, the vanilla attention (Vaswani et al. 2017) takes up too much computational cost when it is applied to 2D spatial features. To eliminate this concern, we propose a novel window attention mechanism, reducing the computational cost significantly while preserving powerful associative capabilities. Given the memory M_j^d and the multi-scale features P_j^d, we first use a 1×1 convolutional layer to project M_j^d, getting $\bar{M}_j^d$ with the same channel dimension as P_j^d. Then, we split them into window features with the window size (h, w), which can be represented by:

$$\begin{aligned} \dot{P}_j^d &= \mathbf{Split}(P_j^d, (h, w)) \\ \dot{M}_j^d &= \mathbf{Split}(\bar{M}_j^d, (h, w)), \\ \dot{P}_j^d, \dot{M}_j^d &\in \mathbb{R}^{\dot{N}_j \times h \times w \times C}, \dot{N}_j = \frac{H}{2^{j-1}h}\frac{W}{2^{j-1}w}. \end{aligned} \tag{4}$$

Here, $\mathbf{Split}(F, s)$ means using the window size s to split F, W_j is the number of split windows. The attention calculation needs the inputs to be batches of vectors, hence we must find a vector to represent each window feature. Previous researches (Zhao et al. 2017; Liu, Rabinovich, and Berg 2015) show that the average pooling operation can filter out a discriminative feature to represent a region. Thus, we obtain

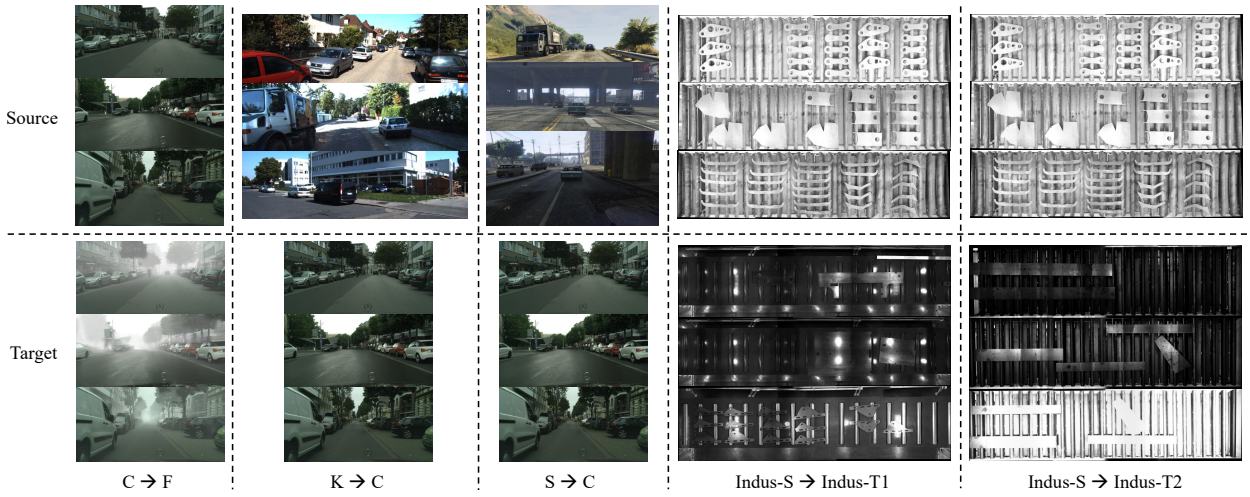

Figure 3: Comparisons between different datasets. "C→F", "K→C" and "S→C" are the existing datasets while "Indus-S→Indus-T1" and "Indus-S→Indus-T2" are the proposed datasets. For existing datasets, the differences between the source and target data are primarily in the background. As a comparison, our datasets face the challenge of changing foreground and background simultaneously.

the window feature vectors by utilizing the average pooling applied on $\dot{P}_j^d$ and $\dot{M}_j^d$:

$$\begin{aligned} \ddot{P}_j^d &= \mathbf{AvgPool}(\dot{P}_j^d), \\ \ddot{M}_j^d &= \mathbf{AvgPool}(\dot{M}_j^d), \\ \ddot{P}_j^d, \ddot{M}_j^d &\in \mathbb{R}^{\dot{N}_j \times C} \end{aligned} \tag{5}$$

Now we use the multi-head attention to build strong relationships between $\ddot{P}_j^d$ and $\ddot{M}_j^d$. The multi-head attention takes the query, key, and value features as inputs, and performs attention calculations for L times in parallel. The multi-head outputs are concatenated and projected by the learnable weight $\mathcal{W}_o$, resulting in the final output:

$$\begin{aligned} \mathbf{MHA}(\ddot{P}_j^d, \ddot{M}_j^d) &= [head_1; ...; head_l; ...; head_L] \cdot \mathcal{W}_o, \\ head_l &= Softmax(\frac{\mathbf{q}_l {\mathbf{k}_l}^T}{\sqrt{C}})\mathbf{v}_l, \end{aligned} \tag{6}$$

where the subscript l represents the index of head, $\mathbf{q}_l$ is generated from $\ddot{M}_j^d$ while $\mathbf{k}_l, \mathbf{v}_l$ are from $\ddot{P}_j^d$. The output of the multi-head attention is reshaped back to a 2-D feature map with shape $\frac{H}{2^{j-1}h} \times \frac{W}{2^{j-1}w} \times C$ and upsampled to the same size as P_j^d so that P_j^d can be connected as a shortcut. In this way, the final output of the efficient adaptation module is calculated by:

$$E_j^d = \mathbf{Up}(\mathbf{MHA}(\ddot{P}_j^d, \ddot{M}_j^d)) + P_j^d, \tag{7}$$

where $\mathbf{Up}(\cdot)$ means the bilinear upsampling function. The whole process is shown in the right box of Fig. 2. Finally, the output features are then sent to the head architecture and used for obtaining the detection results.

Foreground Enhancement. Due to the domain gap, objects of the same category may vary widely across different domains, so we propose to enhance the features of foreground objects. To this end, we elicit an auxiliary branch from the feature of the last layer in the backbone to predict the bounding box mask, which is a binary image of values 1 inside the bounding box.

Training Objective Our training objective consists of three components:

$$L = \lambda_1 L_{box} + \lambda_2 L_{cls} + \lambda_3 L_{mask} \tag{8}$$

where L_{box} denotes the IoU loss (Yu et al. 2016) applied to the bounding box, L_{cls} is the Focal loss (Lin et al. 2017) applied to the class logits and L_{mask} is the Dice loss (Milletari, Navab, and Ahmadi 2016) used for the foreground enhancement. $\{\lambda_i\}_{i=1}^3$ are the weights used to balance different components, and we set $\lambda_1 = 2, \lambda_2 = 1, \lambda_3 = 2$.

4 Experiments

4.1 Datasets and Implementation Details

Public Datasets **Cityscapes→Foggy Cityscapes (C→F)**. There are 2975 images used for training and 500 images for validation in Cityscapes (Cordts et al. 2016). The Foggy Cityscapes (Sakaridis, Dai, and Van Gool 2018) is a synthesized dataset with hand-crafted foggy modification based on Cityscapes. Following (Gao et al. 2022), we select 8 labeled images in the Foggy Cityscapes to train our model. **SIM10k→Cityscapes (S→C) and KITTI→Cityscapes (K→C)**. SIM10k (Johnson-Roberson et al. 2016) is a simulated dataset with 10k synthetic images and 58,701 bounding boxes of car. KITTI (Geiger, Lenz, and Urtasun 2012) is used for autonomous driving scenes and contains 7481 images of the car category. For these two settings, we use the

Method	Architecture	person	rider	car	truck	bus	train	motorcycle	bicycle	mAP50	SO/GAIN
FAFRCNN (2019)	F-RCNN V	27.9	37.8	42.3	20.1	31.9	13.1	24.9	30.6	28.6	19.9/8.7
PICA (2022)	F-RCNN V	28.3	41.3	43.0	23.8	38.1	24.3	25.4	33.7	32.2	20.3/11.9
OA-FSUI2IT (2022)	F-RCNN R	47.5	53.8	64.1	27.8	45.9	11.5	35.9	52.3	42.3	30.0/12.3
SimRoD (2021)	YOLOv5 X	34.3	35.8	55.9	9.6	18.0	5.9	10.6	29.2	24.9	21.9/5.0
FsDet (2020)	YOLOv5 X	32.3	29.8	44.0	14.1	24.2	8.4	22.9	26.2	25.2	21.9/3.3
AcroFOD (2022)	YOLOv5 X	46.2	47.3	63.5	20.1	41.5	34.2	36.1	39.6	41.1	21.9/19.2
AsyFOD (2023)	YOLOv5 X	46.9	48.7	66.8	26.3	45.1	40.6	40.6	39.2	44.3	21.9/22.4
MS2A (ours)	YOLOv5 X	48.6	45.4	64.8	29.2	51.2	46.3	25.6	42.6	**44.8**	21.9/**22.9**
MS2A (ours)	YOLOX X	48.8	45.6	64.7	29.0	51.6	46.1	31.8	44.6	**45.3**	22.1/**23.2**

Table 1: Comparisons to state-of-the-art methods on C→F. "V"/"R" stand for VGG16/ResNet50 backbone networks. "X" stands for the variant architecture of YOLOv5/YOLOX model. "SO" denotes the source-only results, and "GAIN" represents gains after adaptation compared with the source-only model.

Method	Indus-S → Indus-T1						Indus-S → Indus-T2					
	10-anno		30-anno		50-anno		10-anno		30-anno		50-anno	
	AP	SO/GAIN	AP	SO/GAIN	AP	SO/GAIN	AP	SO/GAIN	AP	SO/GAIN	AP	SO/GAIN
FsDet	74.5	65.8/8.7	74.8	65.8/9.0	75.2	65.8/9.4	79.6	71.1/8.5	80.0	71.1/8.9	80.2	71.1/9.1
AcroFOD	79.2	65.8/13.4	80.4	65.8/14.6	81.0	65.8/15.2	84.0	71.1/12.9	84.9	71.1/13.8	85.6	71.1/14.5
OA-FSUI2IT	72.3	65.6/6.7	79.9	65.6/14.3	81.9	65.6/16.3	-	-	-	-	-	-
MS2A (ours)	**89.6**	65.6/**24.0**	**90.5**	65.6/**24.9**	**91.5**	65.6/**25.9**	**93.8**	71.2/**22.6**	**94.1**	71.2/**22.9**	**94.5**	71.2/**23.3**

Table 2: Comparisons to state-of-the-art methods on the proposed Indus-S→Indus-T1 and Indus-S→Indus-T2.

Cityscapes as the target domain and select 8 labeled target images for training. We only consider the car instances for evaluation.

Proposed Industrial Datasets Indus-S→Indus-T1 and Indus-S→Indus-T2. We collect the images of different domains from different factories and use LabelMe (Russell et al. 2008) for labeling. Similar to S→C and K→C, we annotate all the objects as the part class. Indus-S consists of 4614 images for training and 1153 images for validation; Indus-T1 and Indus-T2 have 269 and 432 images for validation respectively. For the training data of the target domain, different from previous datasets, we compare the performances of using three different settings: 10-anno, 30-anno, and 50-anno. The industrial datasets are more challenging than general scenes since the foreground and background are changing simultaneously as shown in Fig. 3.

Method	K → C	SO/GAIN	S → C	SO/GAIN
FARRCNN	-	-	41.2	33.5/7.7
PICA	-	-	42.1	34.6/7.5
FsDet	52.9	47.4/5.5	52.9	49.0/3.9
SimROD	55.8	47.4/8.4	54.2	49.0/5.2
AcroFOD	62.6	47.4/15.2	62.5	49.0/13.5
AsyFOD	64.1	47.4/16.7	65.4	49.0/16.4
MS2A (ours)	**64.3**	47.4/**16.9**	**65.5**	49.0/**16.5**

Table 3: Performance comparisons on K→C and S→C.

Implementation Details We adopt YOLOX (Ge et al. 2021) as our base detector and implement the experiments by MMDetection (Chen et al. 2019). All the images are resized to 640×640 during training. We set the hyper-parameter k in KNN to 100. We use the SGD as the optimizer with an initial learning rate of 0.0001 and the same decay policy as (Ge et al. 2021). For evaluation, we report average precision with an IoU threshold of 0.5 as AP50/mAP50 for single/multi classes on public datasets, and AP for 10 averaged IoU thresholds of 0.5:0.05:0.95 on proposed datasets.

4.2 Comparisons with state-of-the-arts

Results on C→F. As reported in Tab. 1, MS2A exceeds the second-best AsyFOD by 0.5 in mAP50 with the same architecture of YOLOv5-X and achieves the best mAP50 of 45.3 using YOLOX-X architecture. Previous methods aim to augment the target data (Gao et al. 2022; Zhao, Meng, and Xu 2022; Ramamonjison et al. 2021) or align the source data with the limited target data (Gao et al. 2023; Wang et al. 2020, 2019; Zhong et al. 2022), which fails to utilize the prior knowledge present in the unlabeled target data. Differently, we propose to mine prior memory and transfer the prior information into feature alignment, obtaining significant gains (22.9 for YOLOv5 and 23.3 for YOLOX).

Results on Indus-S→Indus-T1 and Indus-S→Indus-T2. We implement three recent coda-available methods on the proposed datasets. For a fair comparison, we use YOLOX as the architecture of OA-FSUI2IT, and the architectures of FsDet and AcroFOD are still YOLOv5 since its performance is comparable to YOLOX. As shown in Fig. 2, previous methods can not work well on challenging industrial scenes they only reach up to a gain of 16.3 As a comparison, the proposed MS2A achieves the best results in all settings. Especially, our method achieves a gain of more than 22 with only 10 labeled target images, which demonstrates MS2A has great superiority in addressing the CFOD problems. Moreover, such a high performance has met the requirements of deploying the model practically.

Results on S→C and K→C. We report the performances on S→C and K→C in Tab. 3. The proposed MS2A outper-

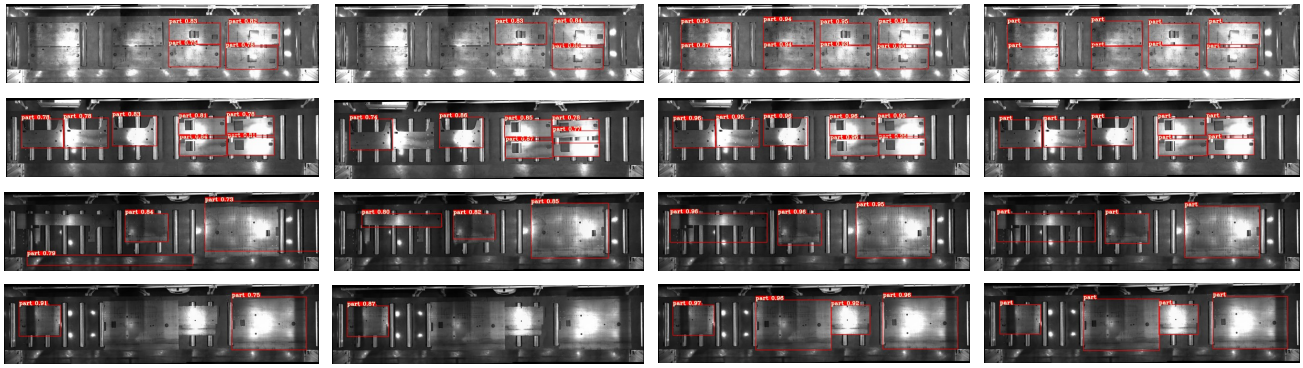

Figure 4: Quantitative results on Indus-S→Indus-T1. Our method can detect objects accurately in such a challenging scene.

EMA	Mask Branch	10-anno		30-anno	
		AP	SO/GAIN	AP	SO/GAIN
×	✓	71.7	65.6/6.1	73.3	65.6/7.7
✓	×	85.9	65.6/20.3	89.4	65.6/23.8
✓	✓	89.6	65.6/24.0	90.5	65.6/24.9
w/o momentum update		88.3	65.6/22.7	90.0	65.6/24.4
w momentum update		89.6	65.6/24.0	90.5	65.6/24.9
window size	(5, 5)	89.9	65.6/24.3	90.3	65.6/24.7
	(8, 8)	90.6	65.6/25.0	90.3	65.6/24.7
	(10, 10)	89.6	65.6/24.0	90.5	65.6/24.9
	(20, 20)	89.5	65.6/23.9	90.2	65.6/24.6

Table 4: Ablation of architecture components. "EAM" means the efficient adaptation module.

forms all the existing methods and gets the best gain of 16.9 and 16.5 on K→C and S→C respectively.

Visual results. Firstly, we compare the quantitative detection results on Indus-S→Indus-T1. As shown in Fig. 4, there are some false and missed bounding boxes in AcroFOD and FsDet, while our method detects all the objects labeled in the ground truth. Moreover, we plot the feature maps after the neck as shown in Fig. 5. Without the memory adaptation module, the network hardly focuses on the region of the object, which indicates the efficient adaptation module helps the model extract more discriminative features.

4.3 Ablation Studies

We conduct the ablation studies on Indus-S→Indus-T1. **Architecture components and momentum update**. Based on YOLOX, our network additionally introduces an efficient adaptation module and an auxiliary mask branch. To show the effectiveness of these components, we gradually add individual modules to the basic YOLOX and test the performance. As shown in Tab. 4, the efficient adaptation module improves the AP greatly (+20.3 for 10-anno and +23.8 for 30-anno). With the mask branch together, we can get further improvements (+24.0 for 10-anno and +24.9 for 30-anno). On the other hand, updating the memory can improve the AP by 1.3 and 0.5 for 10-anno and 30-anno respectively.

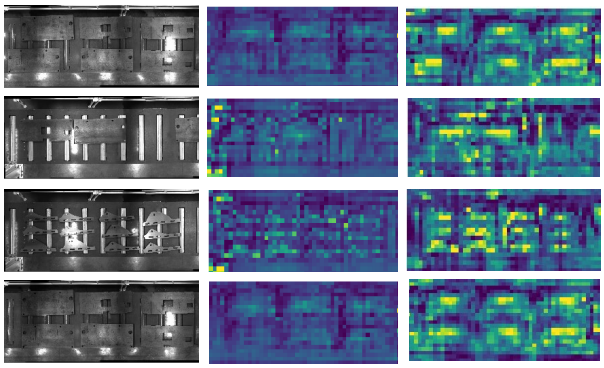

Figure 5: Feature maps visualization of target domain data. We select the feature of the largest scale (80×80) and resize it back to the original image size for visualization.

Window size of the efficient adaptation module. We also conduct ablations to show the effectiveness of the window size (h, w) used in the efficient adaptation module. We do not compare to the vanilla attention since the GPU memory ($2\times$ RTX 2080Ti) is still not enough though the batch size is 1. As shown in Tab. 4, the size of 8×8 is best for 10-anno and 10×10 for 30-anno.

5 Conclusion

In this paper, we propose a novel memory storage-to-adaptation mechanism to learn the prior memory and transfer the memory into domain adaptation. Helped by the robust prior memory, we can guide the feature alignment to be more accurate and extract more discriminative features for the complex industrial environments. Moreover, we build a challenging industrial dataset to bridge the gap of CFOD in industrial settings. We perform experiments on both public and proposed datasets and achieve state-of-the-art performances on all datasets.

References

Bochkovskiy, A.; Wang, C.-Y.; and Liao, H.-Y. M. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Chen, C.; Zheng, Z.; Ding, X.; Huang, Y.; and Dou, Q. 2020. Harmonizing transferability and discriminability for adapting object detectors. In *CVPR*, 8869–8878.

Chen, K.; Wang, J.; Pang, J.; Cao, Y.; Xiong, Y.; Li, X.; Sun, S.; Feng, W.; Liu, Z.; Xu, J.; et al. 2019. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*.

Chen, Y.; Li, W.; Sakaridis, C.; Dai, D.; and Van Gool, L. 2018. Domain adaptive faster r-cnn for object detection in the wild. In *CVPR*, 3339–3348.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 3213–3223.

Cover, T.; and Hart, P. 1967. Nearest neighbor pattern classification. *TIT*, 13(1): 21–27.

Ding, N.; Xu, Y.; Tang, Y.; Xu, C.; Wang, Y.; and Tao, D. 2022. Source-free domain adaptation via distribution estimation. In *CVPR*, 7212–7222.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Dwibedi, D.; Misra, I.; and Hebert, M. 2017. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *ICCV*, 1301–1310.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 1126–1135.

Gao, Y.; Lin, K.-Y.; Yan, J.; Wang, Y.; and Zheng, W.-S. 2023. AsyFOD: An Asymmetric Adaptation Paradigm for Few-Shot Domain Adaptive Object Detection. In *CVPR*, 3261–3271.

Gao, Y.; Yang, L.; Huang, Y.; Xie, S.; Li, S.; and Zheng, W.-S. 2022. Acrofod: An adaptive method for cross-domain few-shot object detection. In *ECCV*, 673–690.

Ge, Z.; Liu, S.; Wang, F.; Li, Z.; and Sun, J. 2021. Yolox: Exceeding yolo series in 2021. *arXiv preprint arXiv:2107.08430*.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 3354–3361.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. 2012. A kernel two-sample test. *JMLR*, 13(1): 723–773.

Han, G.; Ma, J.; Huang, S.; Chen, L.; and Chang, S.-F. 2022. Few-shot object detection with fully cross-transformer. In *CVPR*, 5321–5330.

He, Z.; and Zhang, L. 2019. Multi-adversarial faster-rcnn for unrestricted object detection. In *ICCV*, 6668–6677.

Hu, H.; Bai, S.; Li, A.; Cui, J.; and Wang, L. 2021. Dense relation distillation with context-aware aggregation for few-shot object detection. In *CVPR*, 10185–10194.

Jin, Y.; Wang, X.; Long, M.; and Wang, J. 2020. Minimum class confusion for versatile domain adaptation. In *ECCV*, 464–480.

Johnson-Roberson, M.; Barto, C.; Mehta, R.; Sridhar, S. N.; Rosaen, K.; and Vasudevan, R. 2016. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? *arXiv preprint arXiv:1610.01983*.

Keshari, R.; Vatsa, M.; Singh, R.; and Noore, A. 2018. Learning structure and strength of CNN filters for small sample size training. In *CVPR*, 9349–9358.

Kim, T.; Jeong, M.; Kim, S.; Choi, S.; and Kim, C. 2019. Diversify and match: A domain adaptive representation learning paradigm for object detection. In *CVPR*, 12456–12465.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Koch, G.; Zemel, R.; Salakhutdinov, R.; et al. 2015. Siamese neural networks for one-shot image recognition. In *ICML*, volume 2.

Kozerawski, J.; and Turk, M. 2018. Clear: Cumulative learning for one-shot one-class image recognition. In *CVPR*, 3446–3455.

Lee, K.; Maji, S.; Ravichandran, A.; and Soatto, S. 2019. Meta-learning with differentiable convex optimization. In *CVPR*, 10657–10665.

Li, A.; and Li, Z. 2021. Transformation invariant few-shot object detection. In *CVPR*, 3094–3102.

Li, S.; Ye, M.; Zhu, X.; Zhou, L.; and Xiong, L. 2022. Source-free object detection by learning to overlook domain style. In *CVPR*, 8014–8023.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *ICCV*, 2980–2988.

Liu, W.; Rabinovich, A.; and Berg, A. C. 2015. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*.

Liu, W.; Ren, G.; Yu, R.; Guo, S.; Zhu, J.; and Zhang, L. 2022a. Image-adaptive YOLO for object detection in adverse weather conditions. In *AAAI*, volume 36, 1792–1800.

Liu, X.; Li, W.; Yang, Q.; Li, B.; and Yuan, Y. 2022b. Towards robust adaptive object detection under noisy annotations. In *CVPR*, 14207–14216.

Long, M.; Cao, Y.; Wang, J.; and Jordan, M. 2015. Learning transferable features with deep adaptation networks. In *ICML*, 97–105.

Long, M.; Zhu, H.; Wang, J.; and Jordan, M. I. 2016. Unsupervised domain adaptation with residual transfer networks. In *NeurIPS*.

Long, M.; Zhu, H.; Wang, J.; and Jordan, M. I. 2017. Deep transfer learning with joint adaptation networks. In *ICML*, 2208–2217.

Milletari, F.; Navab, N.; and Ahmadi, S.-A. 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, 565–571.

Mirza, M. J.; Micorek, J.; Possegger, H.; and Bischof, H. 2022. The norm must go on: Dynamic unsupervised domain adaptation by normalization. In *CVPR*, 14765–14775.

Ramamonjison, R.; Banitalebi-Dehkordi, A.; Kang, X.; Bai, X.; and Zhang, Y. 2021. Simrod: A simple adaptation method for robust object detection. In *ICCV*, 3570–3579.

Ren, M.; Triantafillou, E.; Ravi, S.; Snell, J.; Swersky, K.; Tenenbaum, J. B.; Larochelle, H.; and Zemel, R. S. 2018. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*.

Russell, B. C.; Torralba, A.; Murphy, K. P.; and Freeman, W. T. 2008. LabelMe: a database and web-based tool for image annotation. *IJCV*, 77: 157–173.

Saito, K.; Ushiku, Y.; Harada, T.; and Saenko, K. 2019. Strong-weak distribution alignment for adaptive object detection. In *CVPR*, 6956–6965.

Sakaridis, C.; Dai, D.; and Van Gool, L. 2018. Semantic foggy scene understanding with synthetic data. *IJCV*, 126: 973–992.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. In *NeurIPS*.

Sugiyama, M.; Nakajima, S.; Kashima, H.; Buenau, P.; and Kawanabe, M. 2007. Direct importance estimation with model selection and its application to covariate shift adaptation. In *NeurIPS*.

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M. 2018. Learning to compare: Relation network for few-shot learning. In *CVPR*, 1199–1208.

Van den Oord, A.; Kalchbrenner, N.; Espeholt, L.; Vinyals, O.; Graves, A.; et al. 2016. Conditional image generation with pixelcnn decoders. In *NeurIPS*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. In *NeurIPS*.

Wang, T.; Zhang, X.; Yuan, L.; and Feng, J. 2019. Few-shot adaptive faster r-cnn. In *CVPR*, 7173–7182.

Wang, X.; Huang, T. E.; Darrell, T.; Gonzalez, J. E.; and Yu, F. 2020. Frustratingly simple few-shot object detection. *arXiv preprint arXiv:2003.06957*.

Wu, A.; and Deng, C. 2022. Single-domain generalized object detection in urban scene via cyclic-disentangled self-distillation. In *CVPR*, 847–856.

Xu, Y.; Sun, Y.; Yang, Z.; Miao, J.; and Yang, Y. 2022. H2fa r-cnn: Holistic and hierarchical feature alignment for cross-domain weakly supervised object detection. In *CVPR*, 14329–14339.

Yan, X.; Chen, Z.; Xu, A.; Wang, X.; Liang, X.; and Lin, L. 2019. Meta r-cnn: Towards general solver for instance-level low-shot learning. In *ICCV*, 9577–9586.

Yoo, D.; Fan, H.; Boddeti, V.; and Kitani, K. 2018. Efficient k-shot learning with regularized deep networks. In *AAAI*, volume 32.

Yoo, J.; Chung, I.; and Kwak, N. 2022. Unsupervised domain adaptation for one-stage object detector using offsets to bounding box. In *ECCV*, 691–708.

Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; and Huang, T. 2016. Unitbox: An advanced object detection network. In *ACM MM*, 516–520.

Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 6023–6032.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid scene parsing network. In *CVPR*.

Zhao, L.; Meng, Y.; and Xu, L. 2022. Oa-fsui2it: A novel few-shot cross domain object detection framework with object-aware few-shot unsupervised image-to-image translation. In *AAAI*, volume 36, 3426–3435.

Zheng, Y.; Huang, D.; Liu, S.; and Wang, Y. 2020. Cross-domain object detection through coarse-to-fine feature adaptation. In *CVPR*, 13766–13775.

Zhong, C.; Wang, J.; Feng, C.; Zhang, Y.; Sun, J.; and Yokota, Y. 2022. Pica: Point-wise instance and centroid alignment based few-shot domain adaptive object detection with loose annotations. In *ICCV*, 2329–2338.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2223–2232.