Towards Online Domain Adaptive Object Detection

Vibashan VS, Poojan Oza, and Vishal M. Patel
Johns Hopkins University, Baltimore, MD, USA
{vvishnu2, poza2, vpatel36}@jhu.edu

Abstract

Existing object detection models assume both the training and test data are sampled from the same source domain. This assumption does not hold true when these detectors are deployed in real-world applications, where they encounter new visual domains. Unsupervised Domain Adaptation (UDA) methods are generally employed to mitigate the adverse effects caused by domain shift. Existing UDA methods operate in an offline manner where the model is first adapted toward the target domain and then deployed in real-world applications. However, this offline adaptation strategy is not suitable for real-world applications as the model frequently encounters new domain shifts. Hence, it is critical to develop a feasible UDA method that generalizes to the new domain shifts encountered during deployment time in a continuous online manner. To this end, we propose a novel unified adaptation framework that adapts and improves generalization on the target domain in both offline and online settings. Specifically, we introduce MemXformer - a cross-attention transformer-based memory module where items in the memory take advantage of domain shifts and record prototypical patterns of the target distribution. Further, MemXformer produces strong positive and negative pairs to guide a novel contrastive loss, which enhances target-specific representation learning. Experiments on diverse detection benchmarks show that the proposed strategy produces state-of-the-art performance in both offline and online settings. To the best of our knowledge, this is the first work to address online and offline adaptation settings for object detection. Source code: https://github.com/Vibashan/memXformer-online-da

1. Introduction

The ability to train deep network models on large-scale annotated datasets [35, 14, 41, 29] has accelerated the progress for multiple computer vision tasks such as classification [35, 19, 13], segmentation [45, 72, 54], and detection [48, 47, 42]. Despite this success, these models have limited generalization capabilities [56, 23, 17]. Specifically, the model performance drops when the test data (target domain) is sampled from a different distribution than that of the training data (source domain) [1]. For example, when a model is deployed in real-world applications such as autonomous navigation, it could encounter images with weather-based degradations, camera artifacts, etc., unknown during training.

Unsupervised Domain Adaptation (UDA) methods [15, 58, 51, 9, 25, 24, 8, 28, 50] are generally employed to improve model generalization under domain shift condition. Existing UDA methods assume that both labelled source data and unlabeled target data are available during adaptation. This scenario is often not feasible in current real-world applications, as the labelled source data is often restricted due to privacy regulations, data transmission constraints, or proprietary data concerns. To overcome this drawback, recently, some works have explored Source-Free Domain Adaptation (SFDA) [37, 34, 68, 40, 39] setting, where a source-trained model is adapted towards the target domain without requiring access to the source data. However, in both UDA and SFDA settings, adaptation is performed in an offline manner where the model is first adapted towards the target domain and then deployed in real-world applications. In addition, it is often impossible to have prior knowledge about the target domain in most real-world applications. In other words, the deployed model could encounter a diverse set of target domains and offline adaptation to every distribution shift would be infeasible. Therefore, we propose a unified adaptation framework which utilizes a source-trained detector and adapts to the target domain in both offline and online manner.

Figure 1. Left: Unsupervised Domain Adaptation - labeled source data and unlabeled target data are available during adaptation. Middle: Source-Free Domain Adaptation - source-trained model is adapted to the target domain. Right: Online Source-Free Domain Adaptation - source-trained model is adapted to target distribution shift during real-world deployment via online updates.
In recent years, few works have explored various test-time adaptation settings where adaptation is performed during test-time [66, 70, 59]. Wang [66] proposed a fully test-time adaptation strategy which performs entropy minimization during test-time and only updates the model batch-norm parameters for the classification task. However, extending TENT to detection framework [66] has two critical drawbacks: 1) TENT use a very large batch size during test-time adaptation, which is not feasible during real-time deployment as images arrive one by one sequentially. 2) Updating only the batch norm parameter of a network with batch size 1 essentially degrades the model performance [70]. Although existing test-time adaptation settings are closer to online-SFDA settings, they are not suitable for adapting a detection model during real-world deployment. To overcome these issues, we explore an Online Source-Free Domain Adaptation (Online-SFDA) setting, where a model is adapted to any distribution shifts encountered during deployment in an online manner with batch size 1. Fig. 1 illustrate the online source-free domain adaptation setting for detection and its differences against the other adaptation settings.

Source-free domain adaptive object detection is relatively new and a more challenging setting than UDA. Existing SFDA methods [39, 27] for detection adapt to the target domain by training on the pseudo-labels generated by the source-trained model. Due to domain shifts, these generated pseudo-labels are noisy and training a model on top of them would lead to noise overfitting [44, 12]. To alleviate these issues, we employ a mean-teacher framework where the student model is supervised using pseudo-labels generated by the teacher network and the teacher network is slowly updated via the exponential moving average (EMA) of student weights. Therefore, the student network is trained on consistent pseudo-labels leading to less overfitting and the teacher network is a gradual ensemble of target adapted student weights [44]. However, this strategy is inefficient in learning two critical aspects required for optimal online adaptation: 1) They fail to learn robust target feature representation, 2) They fail to fully exploit the online target samples. Hence, we propose a novel memory module and a contrastive loss to fully utilize online target samples and learn robust target feature representation.

Contrastive Learning (CL) [5, 6, 18, 10, 31] aims to learn high-quality features from unlabeled data by forcing similar object instances to stay close and push dissimilar ones apart in an unsupervised manner. This is especially useful for online-SFDA as source-labelled data are unavailable during adaptation. Existing CL methods are designed for classification tasks where they operate on image-level features and require multiple image views (or augmentations) [5] to learn robust feature representation. Consequently, obtaining these large sets of views through input augmentations is computationally expensive for adapting detector models. However in detector models, it is possible to obtain different views for an object in an input image without heavy input augmentations. More precisely, the detector provides multiple object proposals generated by Region Proposal Network (RPN), which in turn provides multiple cropped views around the object instance at different locations and various scales. Therefore, applying CL loss on RPN cropped views guides the model to learn object-level feature representation on the target domain. Note this CL loss is used to supervise the student network, where the object-level features are obtained from the student RoI features. However, to perform contrastive learning, these student RoI features require positive and negative pairs. To this end, we propose MemXformer, a cross-attention transformer-based memory module where items in the memory record prototypical patterns of the continuous target distribution. The proposed MemXformer solves two important problems for online adaptation: 1) store the target distribution during online adaptation, which is utilized for future adaptation. 2) stored temporal ensemble of target representations provides positive and negative pairs to guide the contrastive learning process. Further, we introduce a cross-attention based read and write technique which models better target distribution and provides strong positive and negative pairs for contrastive learning. Note that the proposed method is not only suitable for online adaptation but also for offline adaptation. In a nutshell, this paper makes the following contributions:

- To the best of our knowledge, this is the first work to consider both online and offline adaptation settings for detector models.
- We propose a novel unified adaptation framework which makes the detector models robust against online target distribution shifts.
- We introduce the MemXformer module, which stores prototypical patterns of the target distribution and provides contrastive pairs to boost contrastive learning on the target domain.
- We consider multiple detection benchmarks for experimental analysis and show that the proposed method outperforms existing UDA, and SFDA methods for both online and offline settings.

2. Related works

Unsupervised domain adaptation. Existing unsupervised domain adaptation methods can be categorized into three groups based on adversarial training [7, 50, 53, 63], self-training [30, 67] and image-to-image translation [33, 49]. The first domain adaptive object detection was studied in [7], where they followed an adversarial-based strategy to perform feature alignment at both image-level and instance-level to mitigate the domain shift. Later, Saito [50] pro-
posed an adversarial-based strategy where strong alignment of the local features and weak alignment of the global features. Kim [33] introduced an image-to-image translation-based where multiple target domain images are created by stylizing the labelled source images. Multiple discriminators are used to performing adversarial alignment to reduce domain discrepancy by utilizing these target-styled source images. In [30], a pseudo-label based training strategy was formulated to counter noise in pseudo-labels to perform robust training of object detectors on the target domain. However, all these works assume to have access to labelled source data and unlabeled target data during adaptation, and they operate in an offline setting.

**Source-free domain adaptation.** In the source-free domain adaptation setting, we have a source-trained model which adapts to the target domain without having access to source data. Multiple works have addressed the source-free domain adaptation (SFDA) setting for classification [40, 38], segmentation [43, 36, 64] and object detection [39, 27, 61, 21, 22] tasks. In detail, for classification task [39] proposed a self-supervised method to learn target domain representation via information maximization. Further for segmentation [43, 36] and object detection [39, 27], the proposed methods are based on pseudo-label self-training to learn target-specific representation. However, similar to existing UDA works, these SFDA methods operate in an offline setting. Thus, we explore online adaptation, which is a more practical way to tackle domain shifts for real-world applications.

**Online adaptation.** Sun [55] proposed a Test-time training (TTT) strategy, where a model is trained on source data along with an auxiliary task (eg: rotation prediction) which is utilized during test-time to fine-tune the model on target test distribution. The major drawback of this adaptation strategy is training an auxiliary task along with source training just to perform adaptation during test-time is not a feasible solution and effective solution for real-world application. Later, Wang [66] proposed a fully test-time adaptation setting, where the given source trained model adapts to the target domain by entropy minimization during test-time in an online manner by entropy minimization. In this way, Tent [66] adapts to the target domain with test-time loss. Here, the major limitation of [66] is a requirement of a large batch size during test-time adaptation, which is not feasible during real-time deployment as images arrive one by one sequentially. Although existing test-time adaptation settings provide close resemblance to online-SFDA settings, these test-time settings are not suitable for adapting a detection model during real-world deployment. Therefore, in this work, we explore both online and offline adaptation settings for the object detection tasks.

**Contrastive representation learning.** Contrastive representation learning has shown huge progress towards unsupervised feature learning. The standard way of formulating contrastive learning for an anchor is by pulling together the feature embedding of anchor’s positive pairs and pushing apart from the anchor’s negative pair [46, 5, 18]. These positive and negative pairs are formed by augmenting the anchor image and sampling from the input batch of images. Thus, for a given anchor, the positive pair are augmented anchor images and the negative pairs are other images from the batch [46, 5, 18]. On top of this, by exploiting the task-specific label information, [31] performed contrastive learning in a supervised manner. Nonetheless, all these tasks require a large batch size to perform contrastive learning effectively and it is not feasible to have more than one image during online adaptation. Thus, we propose a memory-based contrastive learning framework suitable for adapting object detectors during deployment in an online manner.

3. Proposed method

The online-SFDA setting considers a source-trained model with parameters $\Theta_{src}$ and adapts to any target distribution shifts during real-world deployment as illustrated in Fig. 1. Let us consider a stream of online target data denoted as $T = \{x_1, x_2, ..., x_n\}$, where $x_n$ is the $n^{th}$ online sample. Since these samples arrive sequentially, the model gets adapted to each sample and the adapted weights are used for future online samples. Specifically, the model parameters during adaptation on the $n^{th}$ sample $x_n$, i.e. $\Theta_{src}^{(n)}$, are initialized with the model parameters updated through online adaptation of previous $x_{n-1}^{th}$ sample. To summarize, online-SFDA performs continuous online adaptation, i.e., adaptation will be continued as long as there is a stream of data and necessity.

**Student-teacher training.** In online-SFDA, the model parameters need to be continuously updated in an online unsupervised manner. Consequently, the model risks forgetting the original hypothesis learned through supervised source training [44, 12]. To overcome this, prior works [57, 44] have employed a student-teacher framework. Specifically, the student parameters ($\Theta_{std}$) are adapted to the target domain by minimizing the detection loss supervised through the teacher-generated pseudo-labels. The adapted student parameters are then transferred to the teacher parameters ($\Theta_{tch}$) via Exponential Moving Average (EMA). This can be formally written as:

$$L_{pl}(x_n) = L_{rpn}(x_n, \tilde{y}_n) + L_{rcnn}(x_n, \tilde{y}_n) \tag{1}$$

$$\Theta_{std}^{(n+1)} ← \Theta_{std}^{(n)} + \gamma \frac{\partial L_{pl}(x_n)}{\partial \Theta_{std}^{(n)}} \tag{2}$$

$$\Theta_{tch}^{(n+1)} ← \alpha \Theta_{tch}^{(n)} + (1 - \alpha) \Theta_{std}^{(n+1)} \tag{3}$$

where $x_n$ and $\tilde{y}_n$ are the $n^{th}$ test sample and corresponding pseudo-label generated by teacher network, $L_{pl}$ is the pseudo-label supervision loss, $\gamma$ is the student learning rate, and $\alpha$ is teacher EMA rate. However, the student-teacher framework is still not sufficient to learn robust features to
mitigate target distribution shifts. Hence, we explore contrastive learning-based strategies further to improve the robustness of feature representations in an online setting.

**Contrastive Learning (CL).** SimCLR [5] is a commonly used CL framework which learns representations for an image by maximizing agreement between differently augmented views of the same sample. For given an anchor image $x_i$, the SimCLR loss can be written as:

$$
\mathcal{L}_{\text{SimCLR}}(x_i) = -\log \left( \frac{\exp(\text{sim}(z_i, z_j))}{\sum_{l=1, l \neq i}^{2N} \exp(\text{sim}(z_i, z_l))} \right),
$$

(4)

where $N$ is the batch size, $z_i$ and $z_j$ are the features of two different augmentations of the same sample $x_i$, whereas $z_l$ represents the feature of the $l^{th}$ batch sample $x_l$, where $l \neq i$. Also, $\text{sim}(\cdot, \cdot)$ indicates a similarity function, e.g., cosine similarity. Note that in general, the CRL framework assumes that each image contains one category/object [5]. Moreover, it requires large batch sizes that could provide multiple positive/negative pairs for training [6]. In contrast for object detection, each image will have multiple objects and a large batch size or multiple views are computationally not feasible. Hence, existing CRL methods are more suited for classification tasks.

### 3.1. Memory-based contrastive learning

Though existing contrastive learning methods like SimCLR are exceptional at learning high-quality representations, they are more suitable for the classification task. For detection, these CL methods require large batch size and heavy input augmentation, which are computationally expensive to apply for online parameter updates (discussed in Sec. 1). Therefore, we utilize a computationally efficient memory-based approach to make contrastive learning feasible and effective for online model updates. The proposed online-SFDA strategy is illustrated in Fig. 2.

**MemXformer.** A cross-attention transformer-based memory module which stores target distribution shift and guides the contrastive learning for target domain representation during online adaptation. Specifically, we employ a Global Memory Bank $M = \{m^i \in \mathbb{R}^{1 \times C} \}_{i=1}^{N_f}$, where $N_f$ is number of memory items and $C$ is memory item feature dimension. These memory items are used to store target representation and record prototypical patterns of the target distribution during the adaptation. In addition, these memory items are used to retrieve strong positive and negative pairs for guiding contrastive learning. The MemXformer module has two operations: write and read, which are based on cross-attention. In the MemXformer write operation, the teacher RoI features are used to update the memory elements appropriately. In the MemXformer read operation, the student RoI features are queried to the memory and a weighted sum of similar memory elements is retrieved, which essentially provide strong positive pairs. The read and write operations of MemXformer are illustrated in Fig. 3.

**Write.** To update the memory elements, we consider only the teacher RoI features $R_t = \{f^i_t \in \mathbb{R}^{1 \times C} \}_{i=1}^{N_f}$, where $N_f$ is number of RoI features and $C$ is RoI feature dimension. The teacher RoI features are considered because in the student-teacher framework, the teacher pipeline has input with weak augmentations resulting in accurate RPN proposals compared to the student pipeline. As shown in Fig. 3 (a), first the teacher RoI features are projected as key $K_t = \{k^i_t \}_{i=1}^{N_f}$ and value $V_t = \{v^i_t \}_{i=1}^{N_f}$ using two FC layer with weight $W_k$ and $W_v$, respectively. Now each memory item is considered as query $Q_m = \{m^j \}_{j=1}^{N_f}$, and we compute a cross-attention map $S_t$ between the teacher RoI features and memory items as follows:

$$
k^i_t = W_k \cdot f^i_t, \\
v^i_t = W_v \cdot f^i_t,
$$

(5)
where the cross-attention map \( S_t \) is a 2D matrix of size \( N_m \times N_f \) and \( s^{i,j} \) represents how \( j^{th} \) memory items is related to \( i^{th} \) teacher RoI features. We utilize this cross-attention map \( S_t \) and \( V_t \) to update \( j^{th} \) memory item using following equation:

\[
m^j \leftarrow F \left( m^j + \sum_{i \in V} s^{i,j} v^i \right).
\]

where \( F(\cdot) \) is \( L_2 \) norm. Therefore, using attention-based weighted average and global memory bank update for each online sample makes the MemXformer effectively store and model the target distribution.

**Read.** To read the memory elements, we consider only the student network RoI features \( F_s = \{ f^i_s \in \mathbb{R}^{f \times C}, i = 1 \} \), where \( N_f \) is number of RoI features and \( C \) is RoI feature dimension. In addition, the MemXformer Read operation is performed to obtain strong positive pairs given student RoI features as a query. As shown in Fig. 3 (b), first the student RoI features are projected as query \( Q_s = \{ q^i_s \} \) by one FC layer with weight \( W_q \). Now each memory items are considered as key \( K_m = \{ m^j \} \) and we compute a cross-attention map \( S_s \) between the student RoI features and memory items as follows:

\[
s^i_s = W_k \cdot f^i_s,
\]

\[
s^{i,j}_s = \frac{\exp \left( q^i_s (m^j)^T \right)}{\sum_{m \in M} \exp \left( q^i_s (m^j)^T \right)},
\]

where the cross-attention map \( S_s \) is a 2D matrix of size \( N_m \times N_f \) and given \( i^{th} \) student RoI features as query, the \( s^{i,j}_s \) th row presents \( N_f \) memory items attention score. Therefore, given \( i^{th} \) student RoI features as query, we generate its corresponding positive pair by attention guided weighted sum of most similar memory items. Thus, utilizing the cross-attention map \( S_s \) and considering memory items as value \( V_m = \{ m^j \} \), we compute the strong positive pair for \( i^{th} \) student RoI features using following equation:

\[
p^i_s = \sum_{j \in M} s^{i,j}_s m^j.
\]

where \( P_s = \{ p^i_s \} \) corresponds to set of strong positive pair for student RoI features \( F_s \). In detail, the retrieved positive pairs are temporal ensembles of the prototypical target distribution, which gives more information regarding the online target distribution shifts. This essentially guides contrastive learning to model the target distribution.

**Negative Pair Mining.** As explained earlier from MemXformer read operation, we obtain a set of strong positive pairs for a given student RoI feature. These strong positive pairs are essentially an ensemble of most similar memory items. However, these ensembled similar memory items also contain dissimilar memory items but are scaled with less attention weights. This restricts the contrastive learning capability to effectively model the target domain representation. To mitigate the dissimilar item’s effect on CL, we propose negative pair mining. Specifically in negative pair mining, given a student RoI feature as query and cross-attention map \( S_s \), we mine the least similar 10% of the memory items and label them as negative pairs \( N_s = \{ m^n \} \). As a result, by performing negative pair mining, we obtain \( N_s \) negative samples for one positive sample, where \( N_s \) is top 10% of least similar memory items.

**Memory contrastive loss.** Given student RoI feature \( f^i_s \) as anchor, utilizing MemXformer Read operation and negative pair mining we obtain strong positive \( P_s \) and negative pairs \( N_s \) from MemXformer. Therefore, given an image \( x_n \) with student RoI feature \( F_s \), the MemCLR loss is calculated as:

\[
\mathcal{L}_{\text{MemCLR}}(x_n) = - \log \left\{ \frac{1}{|F_s|} \sum_{i \in F_s} \frac{\exp(f^i_s, p^i_s)}{\sum_{n \in N_s} \exp(f^i_s, m^n)} \right\}.
\]

Therefore, minimizing the MemCLR loss guided by strong positive and negative pairs enhance the student model to learn better target representation in an online-SFDA setting.

**Overall loss.** We illustrate our overall architecture for online source-free domain adaptation in Fig. 2. The proposed method utilizes a global memory bank to perform memory-based contrastive learning to robustify the representations under varying target distribution shifts. Therefore, the overall online-SFDA loss for any online sample \( x_n \) can be calculated as:

\[
\mathcal{L}_{\text{FTTA}}(x_n) = \mathcal{L}_{\text{CL}}(x_n) + \mathcal{L}_{\text{MemCLR}}(x_n).
\]

4. Experiments and Results

To validate the proposed method, we consider four domain shift scenarios where the source train model is adapted to the unlabelled target domain, typically used for comparison in UDA and SFDA literature. Specifically, we evaluate the proposed method with the existing UDA, SFDA and Test-time works under four domain shifts, 1) clear-weather to foggy-weather, 2) real to artistic, 3) synthetic to real, and 4) cross-camera adaptation. Note that, to show the effectiveness of our proposed approach, we evaluate both online and offline settings. Specifically, the offline setting follows the standard SFDA setting. The source-trained model is adapted towards the target domain using an unlabelled target train-set for multiple iterations and evaluated on the target test-set. Whereas in the online setting, the model
W e report the mean Average Precision (mAP) with an IoU learning rate of 0.001 and momentum of 0.9 for 10 epochs. The source model is trained using an SGD optimizer with a weight momentum update parameter \( \alpha \) of 0.9. Further, the student model is set equal to 0.99. The pseudo-labels generated by the teacher network with confidence greater than 0.5 are selected for student training. W e set the batch size to 1 and offline training. The Global Memory Bank contains \( N_m \) memory items, which are set to 1024. Further, the source model is trained using an SGD optimizer with a learning rate of 0.001 and momentum of 0.9 for 10 epochs. We report the mean Average Precision (mAP) with an IoU threshold of 0.5 for the teacher network on the distribution-shifted target domain test data during the evaluation.

### 4.1. Implementation details

For the Online adaptation setting, we adopt Faster-RCNN [48] with ResNet50 [19] as the backbone pre-trained on ImageNet [35]. In all of our experiments, the input images are resized with a shorter side to be 600 pixels while maintaining the aspect ratio. We set the batch size to 1 for all experiments. For the student-teacher framework, the weight momentum update parameter \( \alpha \) of the EMA for the teacher model is set equal to 0.99. The pseudo-labels generated by the teacher network with confidence greater than the threshold \( T=0.9 \) are selected for student training. We utilize an SGD optimizer to train the student network with a learning rate of 0.001 and momentum of 0.9 for both online and offline training. The Global Memory Bank contains \( N_m \) memory items, which are set to 1024. Further, the source model is trained using an SGD optimizer with a learning rate of 0.001 and momentum of 0.9 for 10 epochs. We report the mean Average Precision (mAP) with an IoU threshold of 0.5 for the teacher network on the distribution-shifted target domain test data during the evaluation.

#### 4.1.1 Clear-weather to foggy-weather adaptation

When the source-trained models are deployed in real-world applications such as autonomous navigation, they are likely to encounter data from multiple weather conditions such as fog, haze, etc. In most cases, the deployed detector models would be trained for clear weather conditions. W e propose to formulate this as an online adaptation problem, as it is difficult to pre-determine what kind of weather conditions will occur. Subsequently, we update the detector model in an online manner to adapt to any weather shifts the model might observe after deployment. To evaluate the proposed method under such conditions, we experiment on Cityscapes [11] → FoggyCityscapes [52] dataset. Here, we have a detection model trained on the Cityscapes dataset consisting of 2,975 normal weather images and 500 test images with 8 object categories: person, rider, car, truck, bus, train, motorcycle and bicycle. During inference, images from FoggyCityscapes are sequentially sent and the object detection model is adapted in an online manner to improve generalization on foggy/hazy weather.

*Table 1.* Quantitative results (mAP) for Cityscapes → FoggyCityscapes. S: Source-only, O: Oracle, UDA: Unsupervised Domain Adaptation, SFDA: Source-Free Domain Adaptation, O-SFDA: Online Source-Free Domain Adaptation.

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Offline</th>
<th>Online</th>
<th>Sim10K → Cityscapes</th>
<th>Cityscapes → KITTI</th>
<th>FoggyCityscapes</th>
</tr>
</thead>
<tbody>
<tr>
<td>S Source-Only</td>
<td>✔</td>
<td>✓</td>
<td>32.0</td>
<td>33.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UDA</td>
<td>Faster [7] (CVPR 2018)</td>
<td>✔</td>
<td>✓</td>
<td>38.9</td>
<td>38.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAF [20] (ICCV 2019)</td>
<td>✔</td>
<td>✓</td>
<td>41.1</td>
<td>41.0</td>
<td></td>
</tr>
<tr>
<td>SFDA</td>
<td>Robust DA [30] (ICCV 2019)</td>
<td>✔</td>
<td>✓</td>
<td>42.5</td>
<td>42.9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Strong-Weak [50] (ICCV 2019)</td>
<td>✔</td>
<td>✓</td>
<td>40.1</td>
<td>37.6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Harmonizing [3] (CVPR 2020)</td>
<td>✔</td>
<td>✓</td>
<td>42.5</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cycle DA [71] (ECCV 2020)</td>
<td>✔</td>
<td>✓</td>
<td>41.5</td>
<td>41.7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MeGA CDA [62] (CVPR 2021)</td>
<td>✔</td>
<td>✓</td>
<td>44.8</td>
<td>43.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unbiased DA [12] (CVPR 2021)</td>
<td>✔</td>
<td>✓</td>
<td>43.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SFOD [39] (AAAI 2021)</td>
<td>✔</td>
<td>✓</td>
<td>42.3</td>
<td>41.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MeGA CDA [62] (CVPR 2021)</td>
<td>✔</td>
<td>✓</td>
<td>44.2</td>
<td>46.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unbiased DA [12] (CVPR 2021)</td>
<td>✔</td>
<td>✓</td>
<td>43.7</td>
<td>38.5</td>
<td></td>
</tr>
</tbody>
</table>

*Table 2.* Quantitative results for Sim10K → Cityscapes and KITTI → Cityscapes.

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Offline</th>
<th>Online</th>
<th>Sim10K → Cityscapes</th>
<th>Cityscapes → KITTI</th>
</tr>
</thead>
<tbody>
<tr>
<td>S Source-Only</td>
<td>✔</td>
<td>✓</td>
<td>32.8</td>
<td>34.5</td>
<td></td>
</tr>
<tr>
<td>SFOD [39] (AAAI 2021)</td>
<td>✔</td>
<td>✓</td>
<td>42.3</td>
<td>43.6</td>
<td></td>
</tr>
<tr>
<td>MeGA CDA [62] (CVPR 2021)</td>
<td>✔</td>
<td>✓</td>
<td>44.2</td>
<td>46.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unbiased DA [12] (CVPR 2021)</td>
<td>✔</td>
<td>✓</td>
<td>37.2</td>
<td>38.5</td>
</tr>
</tbody>
</table>
methods such as Tent [66], our best-performing model surpasses it by a huge margin of by 3.0 mAP. Therefore, for Cityscapes→FoggyCityscapes adaptation scenario, our proposed method produces state-of-the-art results in both online and offline SFDA settings.

4.1.2 Synthetic to real world adaptation
Collecting and annotating detection data is computationally intensive, where on top of assigning a category, one needs to add bounding boxes to every object location in the image. On the other hand, creating a synthetic dataset through simulation is much less computation-intensive and generates annotations for free. Hence, training a detector model on a synthetically generated dataset makes sense and then deploying it in real-world conditions. However, stylistic/appearance differences between real and synthetic data limit such deployment due to performance issues. Here, we formulate it as an online adaptation problem to update a synthetic data trained model on the real-world test data. Particularly, we consider a source model trained on Sim10k [29] on 10,000 training images with 58,701 bounding boxes of car category, rendered by the gaming engine Grand Theft Auto. For real-world test data we use the Cityscapes [11] validation set for online model adaptation. In Table 2, we report Sim10K→Cityscapes adaptation results on the existing UDA, SFDA, and O-SFDA methods. In an offline setting, compared to the existing UDA works such as DA Faster [8], SWDA [50] and RobustDA [30], the proposed method outperforms all of them by a considerable margin. Furthermore, when compared to SFOD [39] the proposed method is better by 0.7 mAP. In an online setting, compared to Tent [66], our proposed method outperforms it by 4.0 mAP. Therefore, our proposed method is able to perform well under synthetic to real-world domain shifts.

4.1.3 Cross-camera adaptation
In most real-world applications, it is assumed that both training and test data would be collected using a camera with the same parameters. However, the camera parameters are often different, which causes the collected images to have different appearances, such as radial distortions, tangential distortions, etc. This can cause the model to perform poorly due to changes in the camera parameters. Hence, to tackle any such camera distortions, we formulate the problem as an online adaptation problem and show that the proposed approach succeeds in generalizing to such cases. Here, we have access to only the source model, trained on the KITTI [16] dataset with 7,481 training images with bounding boxes for the car category. To emulate cross-camera scenario, we consider online adaptation on the Cityscapes [11] validation set containing 500 images. We report the results of the cross-camera adaptation experiment in Table 2. Similar to Sim→Cityscapes adaptation even for Kitti→Cityscapes adaptation, we show similar performance improvements compared to UDA, SFDA and O-SFDA methods. Specifically, in the O-SFDA setting, the proposed method outperforms Tent [66] by 5.6 mAP. Thus, our proposed method is able to model the cross-camera domain shifts effectively.

4.1.4 Real to artistic adaptation
Here, we evaluate the proposed method for the case where there is a concept shift in duration inferences. By concept shift, we refer to the case where there is a complete change in the object, e.g., going from real-world to artistic images. Unlike previous scenarios where the objects go through stylistic/appearance changes, the entire concept of an object is different, e.g., a real-world car vs a cartoon car [28]. We show that even in this challenging scenario, the proposed approach is able to improve model generalization through online updates. We consider a model trained on the PascalVOC data [14] which adapts to test set of Watercolor [28]. Specifically, the Watercolor consists of 1K training and 1K testing images with six categories. We compare PASCAL-VOC→Watercolor results with the existing methods in Table 3. From Table 3, we can infer that the proposed method outperforms most of the existing UDA methods and SFDA methods in offline settings. Further, in the online setting, when compared to TENT[65] the proposed method is able to outperform by a significant margin. This demonstrates the capability of the proposed method to generalize even for both online and offline settings.

4.2. Ablation analysis
Quantitative analysis. The Cityscapes→FoggyCityscapes ablation experiment results are reported in Table 4 for the offline-SFDA setting. We first consider a student-teacher offline update baseline which, compared to the source-only baseline, provides significant improvements. To have a fair comparison, we also consider utilizing supervised contrastive loss [31] for offline updates. In particular, we utilize predictions provided by student-teacher training as label information needed for applying the supervised contrastive

Table 3. Quantitative results for PASCAL-VOC → Watercolor.

<table>
<thead>
<tr>
<th>Method</th>
<th>Student only</th>
<th>256</th>
<th>512</th>
<th>1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>DA Faster [7]</td>
<td>✓</td>
<td>75.8</td>
<td>40.6</td>
<td>31.5</td>
</tr>
<tr>
<td>BDC Faster [50]</td>
<td>✓</td>
<td>65.8</td>
<td>43.8</td>
<td>52.6</td>
</tr>
<tr>
<td>RSR [32]</td>
<td>✓</td>
<td>82.8</td>
<td>43.2</td>
<td>30.9</td>
</tr>
<tr>
<td>UDA</td>
<td>✓</td>
<td>77.8</td>
<td>48.0</td>
<td>50.4</td>
</tr>
<tr>
<td>SWDA</td>
<td>✓</td>
<td>71.5</td>
<td>52.0</td>
<td>46.6</td>
</tr>
<tr>
<td>HTCN [13]</td>
<td>✓</td>
<td>78.6</td>
<td>47.5</td>
<td>58.6</td>
</tr>
<tr>
<td>TNet [4]</td>
<td>✓</td>
<td>81.1</td>
<td>49.3</td>
<td>46.2</td>
</tr>
<tr>
<td>Unbiased DA [12]</td>
<td>✓</td>
<td>88.2</td>
<td>55.3</td>
<td>51.7</td>
</tr>
<tr>
<td>DA Faster [8]</td>
<td>✓</td>
<td>69.4</td>
<td>45.9</td>
<td>47.6</td>
</tr>
<tr>
<td>MemCLR (Ours)</td>
<td>✓</td>
<td>73.6</td>
<td>47.6</td>
<td>46.6</td>
</tr>
<tr>
<td>MemCLR (Ours)</td>
<td>✓</td>
<td>70.7</td>
<td>48.5</td>
<td>51.3</td>
</tr>
<tr>
<td>MemCLR (Ours)</td>
<td>✓</td>
<td>62.3</td>
<td>48.3</td>
<td>47.4</td>
</tr>
</tbody>
</table>

Table 4. Ablation analysis on Cityscapes→FoggyCityscapes.

<table>
<thead>
<tr>
<th>Method</th>
<th>Source-only</th>
<th>256</th>
<th>512</th>
<th>1024</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student-teacher</td>
<td>✓</td>
<td>35.1</td>
<td>42.2</td>
<td>51.8</td>
</tr>
<tr>
<td>MemCLR (Ours)</td>
<td>✓</td>
<td>37.2</td>
<td>41.7</td>
<td>51.3</td>
</tr>
<tr>
<td>MemCLR (Ours)</td>
<td>✓</td>
<td>37.2</td>
<td>41.7</td>
<td>51.3</td>
</tr>
<tr>
<td>MemCLR (Ours)</td>
<td>✓</td>
<td>37.2</td>
<td>41.7</td>
<td>51.3</td>
</tr>
<tr>
<td>MemCLR (Ours)</td>
<td>✓</td>
<td>37.2</td>
<td>41.7</td>
<td>51.3</td>
</tr>
</tbody>
</table>

Table 4. Ablation analysis on Cityscapes→FoggyCityscapes.
loss over object proposals. Denoted as SupCon in Table 4, the addition of supervised contrastive learning further improves the performance by 1.3 mAP. However, the proposed memory-based contrastive learning outperforms the supervised contrastive learning by 1.4 mAP, indicating the utility of the proposed method to learn better target representations. Finally, we analyze the performance of the proposed method by varying global memory bank capacity from 256 to 1024 memory items. As shown in Table 4, memory-based contrastive loss with 1024 memory items performs the best when compared to 256 and 512 memory items. Further, note that our model takes around 1 second to perform online adaptation for one sample.

**Qualitative analysis.** Fig. 4 shows t-SNE visualization for source-only, student-teacher training and the proposed method for the Cityscapes→FoggyCityscapes online-SFDA setting. The t-SNE [60] visualizations are created from the RoI features extracted from the predictions for 500 test images. Due to the distribution shift, the features are dispersed for the source-only baseline and classification boundaries are weak. With the help of student-teacher training, the model learns better classification boundaries, resulting in better quantitative performance. However, the features in the student-teacher training have a large variance and do not have compact features. Whereas the proposed method has even better classification boundaries and learns compact features for each category, resulting in a more robust model. Further qualitative comparison is performed to analyze the effect of the order of input sequence during online adaptation. We can observe from the variance that the input sequence order does not affect the model performance much. Note that, in online adaptation, the test samples are seen only once and adaptation happens in an unsupervised manner.

We can observe from the variance that the order of input sequence does not much affect the model’s performance. Further, we can observe the model performance increase as it encounters more test samples during online adaptation, showing the MemXformer effectiveness in exploiting online target distribution. Note that, in online adaptation, the test samples are seen only once and adaptation happens in an unsupervised manner.

**5. Conclusion**

In this work, we introduced a practical domain adaptation setting for the object detection task, which is feasible for real-world settings. Particularly, we proposed a novel unified adaptation framework which makes the detector models robust against online target distribution shifts. Further, we introduce the MemXformer module, which stores prototypical patterns of the target distribution and provides contrastive pairs to boost the contrastive learning on the target domain. We conducted extensive experiments on multiple detection benchmark datasets and compared existing unsupervised domain adaptation, source-free domain adaptation and test-time adaptation methods to show the effectiveness of the proposed approach for both online and offline adaptation of object detection models.

**Acknowledgment:** Research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-23-2-0008. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.
References


