Machine Vision and Applications (2022) 33:65
https://doi.org/10.1007/s00138-022-01320-y

ORIGINAL PAPER

Attention-based domain adaptation for single-stage detectors

Vidit Vidit[1] · Mathieu Salzmann[1]

Received: 21 December 2021 / Revised: 14 April 2022 / Accepted: 15 June 2022 / Published online: 13 July 2022

Abstract

While domain adaptation has been used to improve the performance of object detectors when the training and test data follow different distributions, previous work has mostly focused on two-stage detectors. This is because their use of region proposals makes it possible to perform local adaptation, which has been shown to significantly improve the adaptation effectiveness. Here, by contrast, we target single-stage architectures, which are better suited to resource-constrained detection than two-stage ones but do not provide region proposals. To nonetheless benefit from the strength of local adaptation, we introduce an attention mechanism that lets us identify the important regions on which adaptation should focus. Our method gradually adapts the features from global, image level to local, instance level. Our approach is generic and can be integrated into any Single-Shot Detector. We demonstrate this on standard benchmark datasets by applying it to both the single-shot detector (SSD) and a recent variant of the You Only Look Once detector (YOLOv5). Furthermore, for equivalent single-stage architectures, our method outperforms the state-of-the-art domain adaptation techniques even though they were designed for specific detectors.

Keywords Domain adaptation · Object detection · Adversarial training · Representation learning

1 Introduction

Modern object detection methods can be grouped into two broad categories: two-stage architectures [35], that first extract regions of interest (ROIs) and then classify and refine them, and single-stage ones [21,28,42], that directly output bounding boxes and classes from the feature maps. While the former yield slightly higher accuracy, the latter are faster and more compact, making them better suited for real-time applications or for mobile devices.

In any event, all object detectors reach their best performance when the training and test data are acquired in the same conditions, such as using the same camera, in similar illumination conditions. When they are not, the resulting domain gap significantly degrades the detection results. Addressing this is the focus of domain adaptation [15,18,29,30,37,46]. In this work, we focus on *unsupervised* domain adaptation, whose goal is to bridge the gap between the source (training) and target (test) domain without having access to any target annotations.

The recent work on domain adaptation for object detection [4,6,36,39,44,49] has focused mostly on two-stage detectors. At the heart of most of these methods lies the intuition that adaptation should be performed locally, focusing on the foreground objects because the background content may genuinely differ between the training and test data, whereas the object categories of interest do not. This process of local adaptation is facilitated by the ROIs used in two-stage detectors. Unfortunately, no counterparts to ROIs exist in single-stage detectors, making local adaptation much more challenging. This has been tackled by [19] for the specific detector of [42], which explicitly extracts objectness maps, and by [5], which introduces complementary modules specifically designed for the SSD architecture [28].

In this paper, we introduce a domain adaptation strategy able to perform local adaptation while generalizing across different single-stage object detectors. Specifically, we introduce an attention mechanism that allows adaptation to focus on the regions that matter for detection, that is, the foreground regions, as depicted in Fig. 1. In essence, our approach leverages attention to perform local-level feature alignment, thus following the intuition that has proven successful in adapting two-stage detectors. Our attention mechanism is generic and can be incorporated into any single-stage detector. Furthermore, and contrarily to [5,19], we gradually modulate

✉ Vidit Vidit
vidit.vidit@epfl.ch

Mathieu Salzmann
mathieu.salzmann@epfl.ch

[1] CVLab, EPFL, Rte Cantonale, Lausanne 1015, Switzerland

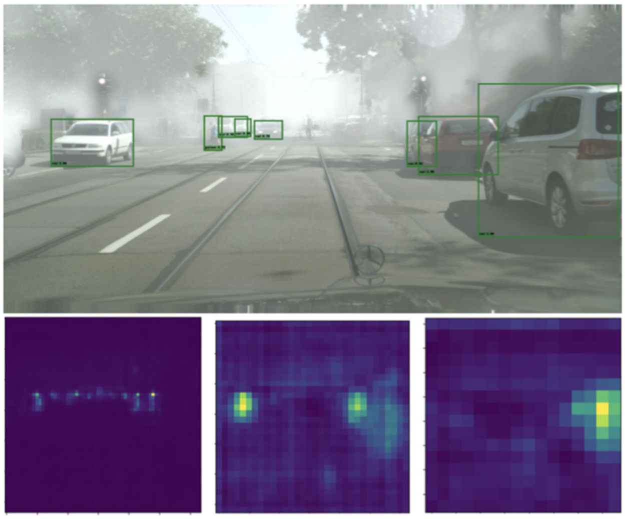

Fig. 1 Leveraging attention for local domain adaptation. Top: target image with predicted detections. Bottom: attention maps output by our approach for feature maps at different scales, allowing us to focus adaptation on the relevant local image regions, ranging from small (left) to large (right) objects. The attention maps are re-scaled to the same size for visualization purpose. Best viewed digitally

the adaptation from global features to local features, which lets us give increasingly more importance to foreground features as training progresses. Consequently, this allows us to use the same domain classifiers for both global and local alignment, thereby leading to a simpler implementation than [5,19]. While [24,50] also propose attention-based adaptation mechanism, in contrast to our work, they are dedicated to specific backbones and thus do not easily transfer to different single-stage architectures.

We demonstrate the benefits of our approach via a series of experiments on several standard domain adaptation detection datasets. Despite its comparative simplicity, our method outperforms the state-of-the-art ones of [5,19]. Furthermore, our results evidence the generalizability of our domain adaptation strategy to different single-stage frameworks, including SSD [28] and YOLOv5 [21], and the importance of local feature alignment over the global ones, particularly in the later training stages. Our code is available at https://github.com/vidit09/adass.

2 Related work

2.1 Object detection

Two-stage object detectors, such as FasterRCNN [35], consist of a feature extractor, a region proposal network (RPN), and a refinement network. The RPN provides foreground regions, via ROI pooling, to the refinement stage for bounding box prediction and classification. Recently, one-stage detectors [3,21,26,28,34,41,42] have emerged as an alternative, becoming competitive in accuracy while faster and more compact than two-stage ones. Most of them [21,26,28,41] rely on predefined bounding box anchors for prediction, and thus do not provide region proposals likely to contain foreground objects as two-stage detectors do. The only exception to this anchor-based approach to single-stage detection is the detectors of [3,42]. Specifically, [42] yields an object centerness map, and [3] learns object regions via a self-attention [43]-based encoder–decoder. Arguably, YOLO [21,34] predicts an objectness score for each anchor box, which could be leveraged to create an objectness map at the feature level. However, we will show in Sect. 4.3.3 that our method is superior to this approach. In any event, in contrast to these approaches, we develop a self-attention framework for domain adaptation. It can be integrated into any anchor-based detector, which we illustrate using SSD [28] and YOLOv5 [21].

2.2 Domain adaptation for object detection

While the bulk of the domain adaptation literature focuses on image classification, several works have nonetheless tackled the task of unsupervised domain adaptation for object detection. In particular, most of them have focused on the two-stage FasterRCNN detector. In this context, [6] uses instance- and image-level alignment to improve the FasterRCNN performance on new domains; [36] shows that a strong local feature alignment improves adaptation, particularly when focusing on foreground regions; [4] performs feature- and image-level adaptation on interpolated domain images generated using a CycleGAN [48]; [9] uses CycleGAN-translated images to remove the source domain bias in the teacher network and generate better pseudo labels for the target domain; [49] clusters the proposed object regions using k-means clustering and uses the centroids to do instance-level alignment; [39] introduces a method to improve the interaction between local and global alignment; [44] learns category-specific attention maps for FasterRCNN using memory modules. In essence, most of these works leverage the RPN proposals to achieve a form of local feature alignment, showing the importance of focusing adaptation on the foreground features. Here, we follow a similar intuition but develop a method applicable to single-stage detectors, which do not rely on an RPN. In Sect. 4.3.2, we nonetheless compare our approach with methods developed for two-stage detectors [4,36], which we adapted to make them compatible with one-stage detectors.

Only few works have tackled domain adaptation for single-stage detectors. Some of these rely on generating better pseudo labels for the target domain and train the detector on them. In particular, [23] proposes to regularize highly confident labels to reduce false positives; [33] develops a domain mixup strategy to gradually adapt the detectors using the generated labels. Pseudo labels, however, are orthogonal to our work; we focus on feature alignment, and while our approach could further benefit from pseudo labels, studying this goes beyond the scope of this paper. Therefore, [5,19] constitute the works closest to our approach. Specifically, [19] uses the object centerness maps predicted by the single-stage detector of [42] to perform local feature alignment. While effective, this approach is therefore restricted to this specific detector. Here, by contrast, we introduce a general approach to local feature alignment in single-stage detection. [5] designs a set of complementary modules, which help global- and local-level alignment in the dissimilar domain setting, implicitly learning foreground regions in the SSD architecture. They formulate their category alignment loss for target domain using the class probabilities of each anchor boxes. SSD, as used in [5], uses softmax-based normalized prediction for each anchor box whereas, YOLOv5 does multiclass prediction using logistic classifiers. Hence, the approach in [5] doesn't translate directly with the multiclass prediction framework of YOLOv5. By contrast, our approach is agnostic to the kind of detection head. Furthermore, we also learn foreground regions implicitly, but rely on a simpler, generalizable strategy, yet outperform both the approaches of [5,19]. Specifically, while [5,19] continuously aim to adapt the global and local features throughout the whole training process, we gradually modulate adaptation from the global to the local level. This lets us focus more strongly on the foreground regions and use the same domain classifiers for global and local adaptation.

2.3 Self-attention

Our approach exploits self-attention (SA). SA was introduced in [43] for natural language processing and has since then become increasingly popular in this field [2,10]. Recently, it has also gained popularity in computer vision, for both image recognition [1,12,32] and object detection [3]. While other attention mechanisms have been proposed [14,20,45,47], they typically require more architectural changes than vanilla SA [43], which motivated us to rely on this strategy in our method. [50] performs domain adaptation with the self-attention-based detector [3]. By contrast, we develop an attention mechanism that can be integrated into a single-stage detector to facilitate adaptation. This makes our approach applicable to the nonattention-based backbones of SSD and YOLOv5, thus making it more general than [50].

3 Method

Let us now introduce our attention-based domain adaptation strategy for single-stage detection.

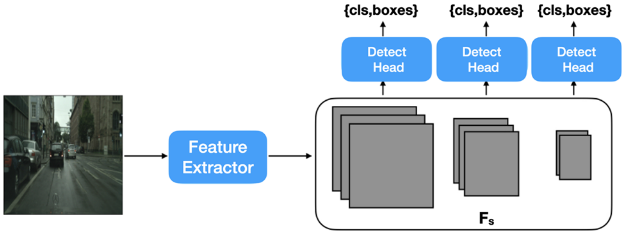

Fig. 2 General single-stage object detection architecture. Both SSD [28] and YOLOv5 [21], used in our experiments, comply with this architecture, and other methods [26,41] also do

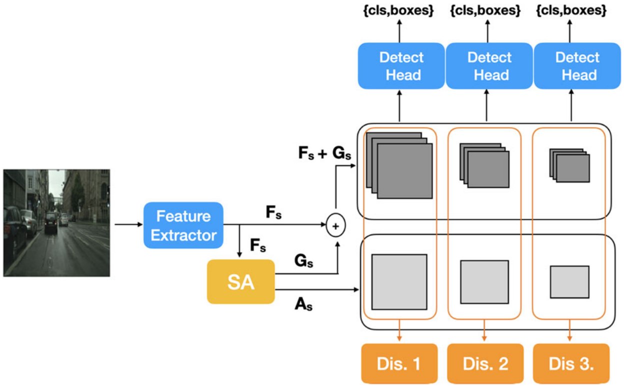

Fig. 3 Overview of our approach. We compute self-attention from the features extracted by the single-stage detector backbone. We then modulate these features with our attention maps so as to encourage the feature alignment achieved by the domain classifiers (abbreviated above as Dis. for discriminator) to focus on the relevant local image regions. The number of domain classifiers matches the number of detection heads in SSD [28] and YOLOv5 [21]

3.1 Attention in single-stage detectors

Single-stage object detectors typically follow the general architecture depicted in Fig. 2, consisting of a feature extractor followed by several detection heads. These detection heads take as input the features F_s at different scales $s \in [1, S]$, with the different scales allowing the detector to effectively handle objects of different sizes. Such an architecture directly predicts bounding boxes and their corresponding class from the feature maps, via the use of bounding box anchors at each spatial location. As such, it does not explicitly provide information about the features corresponding to the objects. This contrasts with two-stage detectors, whose region proposals directly correspond to potential objects.

To automatically extract information about the object locations, we propose to incorporate a self-attention mechanism [43] in the detector. Intuitively, we expect the foreground objects to have higher self-attention than background regions because the detector aims to identify them, and thus exploit self-attention to extract an objectness map. To this end, we use an attention architecture similar to that of [3], but without attention-based decoder because we want to keep the same detector heads as in [21,28].

The attention module takes as input the feature map $F_s \in \mathbb{R}^{H_s \times W_s \times C_s}$ and produces an objectness map $A_s \in \mathbb{R}^{H_s \times W_s}$ and a feature map $G_s \in \mathbb{R}^{H_s \times W_s \times C_s}$. Specifically, F_s is flattened to $\mathbb{R}^{H_s W_s \times C_s}$ and transformed into a query matrix $Q \in \mathbb{R}^{D_q \times D}$, a key matrix $K \in \mathbb{R}^{D_k \times D}$ and a value matrix $V \in \mathbb{R}^{D_v \times C_s}$, with $D_q = D_k = D_v = H_s W_s$, using three separate linear layers. We then compute

$$A'_s = softmax\left(\frac{QK^T}{\sqrt{D}}\right) \quad \in \mathbb{R}^{D_q \times D_k} \tag{1}$$

which, intuitively, represents the similarity between the query and the key at different spatial locations. To compute the objectness map A_s, we then compute the maximum in each row of A'_s, leading to a D_q-dimensional vector, which we min–max normalize, so that each value falls in the range [0, 1]. Finally, A_s is obtained by reshaping this vector to $\mathbb{R}^{H_s \times W_s}$.

Given A'_s, we also compute

$$G'_s = A'_s V \quad \in \mathbb{R}^{H_s W_s \times C_s} \tag{2}$$

which we reshape to $\mathbb{R}^{H_s \times W_s \times C_s}$ to obtain the feature map G_s. We then pass $F_s + G_s$ to the detection head. In addition to this, and as will be discussed in more detail in Sect. 3.2, we further leverage A_s to modulate the $F_s + G_s$ features for domain adaptation. This differs from previous SA works, which do not explicitly exploit the learnt attention maps.

In practice, instead of the single-head attention mechanism discussed above, we rely on the multi-head extension presented in detail in [3,43]. In short, Eq. 1 is computed multiple times using unshared linear layers to obtain different query, key, and value matrices. The resulting independent A'_s matrices are concatenated and linearly transformed to a single matrix of size $\mathbb{R}^{D_q \times D_k}$. Intuitively, and as discussed in [3,43], the multiple heads can extract different representations for the same pair of locations.

As the different detection heads focus on objects of different sizes, we add an attention module at each scale. These modules are trained jointly with the feature extractor and detection heads. Because we do not have access to supervisory signal for the attention/objectness maps, the loss function $\mathcal{L}^{det}$ to train the detector remains the same as that of the original single-stage detector. Typically [21,28], such a loss function incorporates a classification term to categorize predefined anchor bounding boxes, and a regression one to refine these anchors. It can thus be expressed in general as

$$\mathcal{L}^{det}(I) = \mathcal{L}^{cls}(I) + \mathcal{L}^{reg}(I) \ . \tag{3}$$

3.2 Unsupervised domain adaptation

Let us now explain how we exploit the above-mentioned attention mechanism for unsupervised domain adaptation. This process is depicted in Fig. 3. Let I_s be a source image, for which we have the ground-truth bounding boxes and class labels, and I_t be a target image, for which we do not. The source and target images are drawn from two different distributions but depict the same set of classes. Domain adaptation then translates to learning a representation that reduces the gap between both domains.

An effective approach to achieve this consists of jointly training a domain discriminator D in an adversarial manner [15], encouraging the learnt features not to carry any information about the observed domain. In our context, because the detection heads act on features at different scales, we use a separate discriminator D_s for each scale s. However, we do not directly use the feature maps F_s as input to these discriminators, but instead aim to focus the adaptation on the foreground objects, accounting for the fact that the background can genuinely differ across the two domains.

To this end, we leverage the objectness maps from Sect. 3.1 to extract the weighted feature map

$$M_s = (1-\gamma) * (F_s + G_s) + \gamma * (F_s + G_s) \odot A_s \ , \tag{4}$$

where $\odot$ indicates an element-wise product performed independently for each channel of $(F_s + G_s)$, and $\gamma \in [0, 1]$. This formulation combines the global, unaltered features with the local ones obtained by modulating the features by our attention map. During our training, we then gradually increase γ from 0 to 1, which lets us transition from global adaptation to local feature alignment. Intuitively, this accounts for the fact that, at the beginning of training, the predicted attention maps may be unreliable, and a global alignment is thus safer. We also observed such a strategy to facilitate the training of the discriminators. In practice, we compute γ as

$$\gamma = \frac{2}{1 + \exp(-\delta \cdot r)} - 1 \ , \tag{5}$$

where δ controls the smoothness of the change and $r = \frac{\text{current iteration}}{\text{max iteration}}$.

Given the attention-modulated features M_s for each scale s, we then write the discriminator loss as

$$\mathcal{L}^{dis}(I) = -\frac{1}{S}\sum_s t \log(D_s(M_s)) \\ +(1-t)\log(1 - D_s(M_s)), \tag{6}$$

where $t = 0$, resp. $t = 1$, indicates that image I is a source, resp. target image.

During training, the discriminator aims to minimize $\mathcal{L}^{dis}$ while the feature extractor seeks to maximize it. To facilitate such an adversarial training process, we use the gradient reversal layer (GRL) of [15]. Hence, the overall loss function minimized by the feature extractor for a source and a target image can be expressed as

$$\mathcal{L}(I_s) = \mathcal{L}^{det}(I_s) - \mathcal{L}^{dis}(I_s) \; , \tag{7}$$

$$\mathcal{L}(I_t) = -\mathcal{L}^{dis}(I_t) \; , \tag{8}$$

respectively. Note that, unlike [5,19], we do not use pixel-wise domain discriminators, as we found our attention-modulated feature maps to be sufficient to suppress the background features. Moreover, the formulation in Eq. 4 allows us to use the same discriminator for global alignment in the beginning of training and local alignment in the later training stages.

4 Experiments

In this section, we discuss our experimental settings and analyze our results.

4.1 Datasets

We evaluate our method using the following four standard datasets:

Cityscapes [8] contains 2975 images in the training set and 500 in the test set, with annotations provided for eight categories, namely, *person, car, train, rider, truck, motorcycle, bicycle, and bus*. The images depict street scenes taken from a car, mostly in good weather conditions.

Foggy Cityscapes [38] contains synthetic images aiming to mimic the Cityscapes setting, but in foggy weather. It contains 2965 training images and 500 testing ones, depicting the same eight categories as Cityscapes.

Sim10K [22] consists of 9975 synthetic images, with annotations available for the *car* category.

KITTI [16] depicts street scenes similar to those of Cityscapes, but acquired using a different camera setup. In our experiments, we will only use its 6684 training images.

Following [19], we present results for the following domain adaptation tasks:

Sim→Cityscapes (**S→C**): This evaluates the effectiveness of a method to adapt from synthetic data to real images. All Sim10K images are used as source domain, and the Cityscapes training images act as target domain. Following [19], only the *car* class is considered for evaluation.

KITTI→Cityscapes (**K→C**): This task aims to evaluate adaptation to a different camera setup. We use the KITTI training images as source domain and the Cityscapes training images as target one. Again, as in [19], we consider only the *car* class for evaluation.

Cityscapes→Foggy Cityscapes (**C→F**): The goal of this experiment is to test the effectiveness of a method in different weather conditions. We use the Cityscapes training images as source domain and all Foggy Cityscapes images as target data. For this task, all eight object categories are taken into account for evaluation.

4.2 Implementation details

We evaluate our method on two single-stage detectors, SSD [28][1] and YOLOv5 [21][2].We implemented our method in Pytorch [31], and performed all our experiments on a single Nvidia V100 GPU [7]. The batch consists of 8 images, 4 drawn from source and 4 from target domain. We set δ in Eq. 5 to 5. We provide additional training details in the supplementary material.

SSD relies on a similar VGG [40] backbone to that used by the detectors employed in [5,19]. We will therefore focus our comparison with [19] and with [5] to our SSD-based approach. We employ an image resolution of 512×512 because it is the highest resolution available for the SSD architecture. Note that, in [19], larger images were used, i.e., a short image side between 800 and 1333, and that [5] used a lower, 300×300 resolution. For the comparison to be fair, we thus re-trained these methods with this 512×512 image resolution. To further make our SSD architecture comparable to that of [19], we incorporated a Feature Pyramid Network [25] to our SSD backbone. Following [5,19], all backbones were initialized with ImageNet-trained weights.

YOLOv5 is also trained with input images of size of 512×512. This allows us to illustrate the generality of our approach to other single-stage detectors. Specifically, we use the YOLOv5s backbone, which is the smallest model out of all YOLO configurations. We keep the default configuration for preprocessing and data augmentation. We initialize the backbone with COCO-pretrained weights [27] since [21] don't provide ImageNet-trained weights.

4.3 Results

4.3.1 Evaluation metric

Following previous work [5,19,36], we evaluate our method's performance with the Mean Average Precision (mAP) [13]. Specifically, the precision of the detector is computed over 11 equally spaced recall values in the range [0, 1]. We then compute the Average Precision (AP) for each class as the area under the precision–recall curve, and then use the mean of the

[1] https://github.com/lufficc/SSD.

[2] https://github.com/ultralytics/yolov5.

Table 1 Results on Cityscapes to Foggy adaptation

Method	mAP@0.5								
	Person	Car	Train	Rider	Truck	Motorcycle	Bicycle	Bus	Mean
[19] - w/o DA	18	28.3	1.6	18.3	6.5	6.6	15.5	16.5	13.9
[19] - global	25.1	43.3	5.4	27.6	17.8	11.9	22.1	33.5	23.3
[19] - global+local	**26.6**	44.5	4.8	26.2	**21.2**	12.3	19.1	33.9	23.5
I^{3}Net [5]	19.7	37.9	**9.6**	22.9	12.5	**18.3**	22.7	21.1	20.6
SSD - w/o DA	15.1	28.8	0.2	12.9	2.2	5.8	13.7	13.5	11.5
SSD + our DA	23.4	**49.1**	4.9	**27.8**	16.9	17.6	**24.2**	**34.0**	24.8
SWDA [36]	16.6	30.3	0.6	17.9	6.2	9.3	18.5	16.9	14.5
HTCN$^{\psi}$ [4]	11.5	28.8	0.9	9.8	1.7	4.5	12.7	6.4	9.6

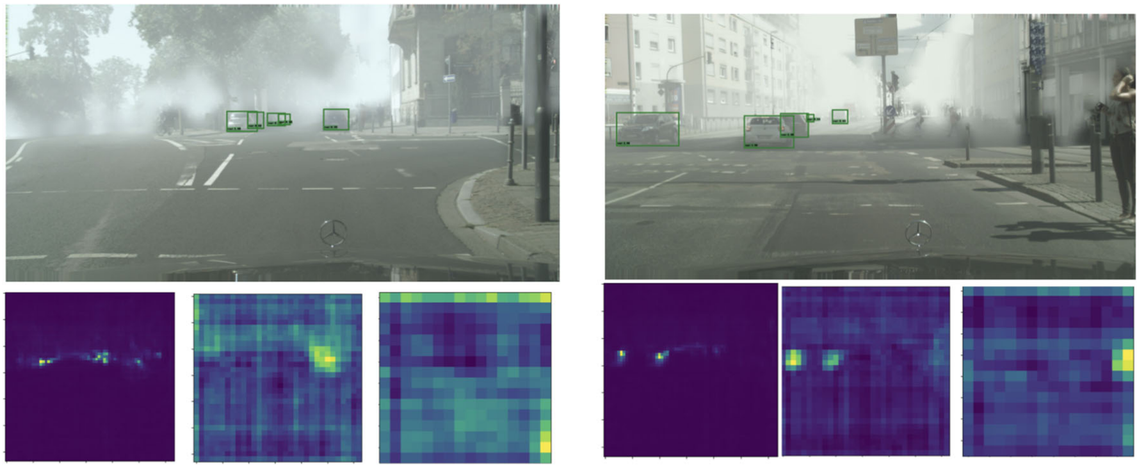

Fig. 4 Qualitative results on C→ F. We show target images with predicted detections, together with attention maps at different scales. While this adaptation task is particularly challenging, our attention maps nonetheless manage to correctly identify the objects at their different scales. Note, when there is no object of interest activation map tends to have activation everywhere. All predictions are with confidence 50% and above

Table 2 Results on Sim10K to Cityscapes adaptation

Method	mAP@0.5
[19] - w/o DA	31.5
[19] - global	33
[19] - global + local	32.8
I^{3}Net [5]	35.1
SSD - w/o DA	29.1
SSD + our DA	**36.7**
SWDA [36]	31.5
HTCN$^{\psi}$ [4]	29.9

APs for the different classes to indicate the overall detector performance on a dataset. In this process, a prediction is considered to be correct if it deemed to contain the right class and has an intersection over union (IOU) score of at least 0.5 with the ground-truth bounding box. We thus refer to our metric as mAP@0.5. In the single-class setting, mAP = AP, and hence we will generically use the term mAP.

4.3.2 Comparison with the state of the art

Let us first compare our SSD-based method with [5] and with the global and local version of [19]. Following [5], we also report the results of SWDA [36] and of HTCN$^{\psi}$ [4], originally developed for two-stage detectors, which we made compatible with single-stage ones. Specifically, we reimplemented both methods within our SSD framework, and further modified the HTCN pixel and image-wise reweighting so as not to use any context vector, as single-stage detectors don't provide access to foreground ROIs. Additionally, we did not use CycleGAN-translated images as in [4] for the comparison to be fair. As a reference point, we also report the results obtained without domain adaptation, as *SSD - w/o DA*.

Table 1 provides the results on **C**→**F**. Our method yields the best results on average (last column). When looking at

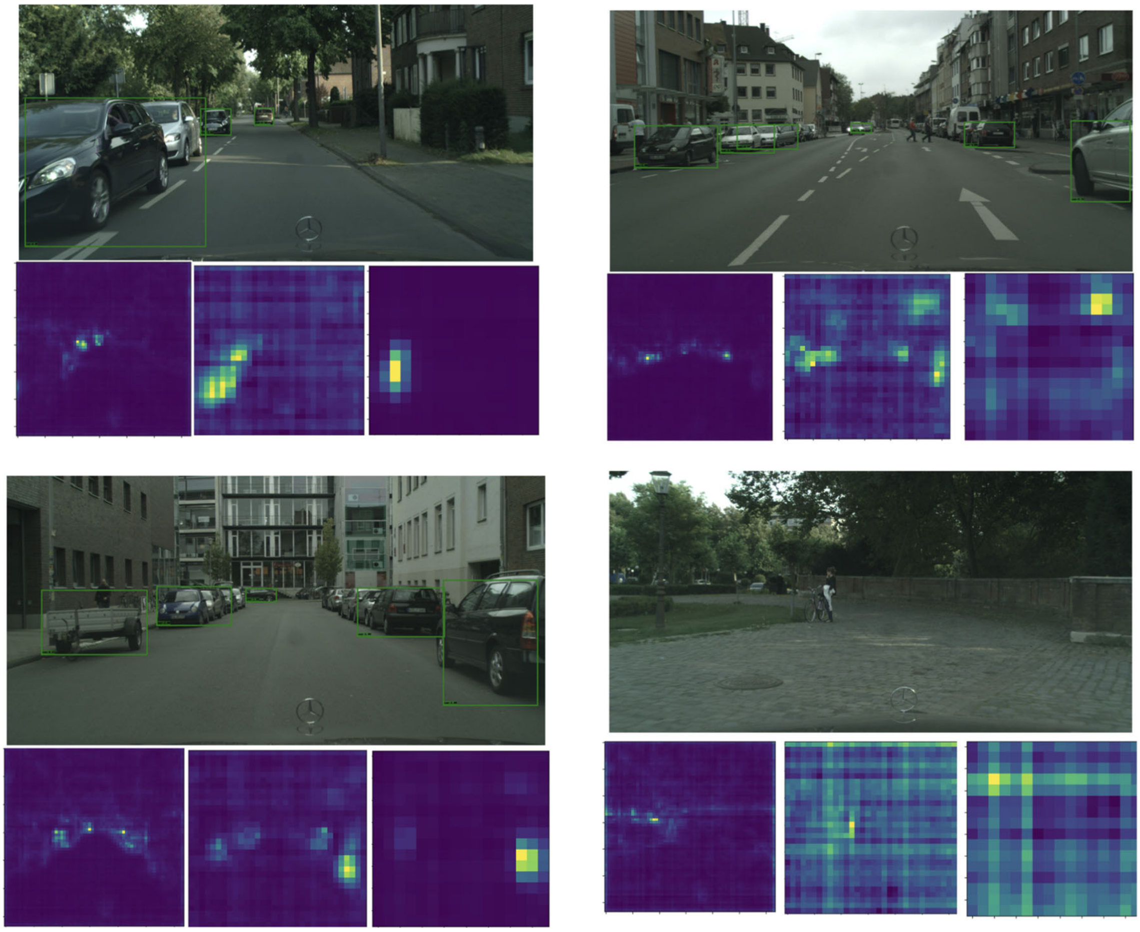

Fig. 5 Qualitative results on S→ C. We show target images with their predicted detections, together with the corresponding attention maps at different scales. Note that the finer map (left) correctly identifies the small cars whereas the coarser one (right) focuses on large cars. Bottom right: Because, this task focuses on cars only, this image does not contain any object of interest. Hence, in this case, the attention maps tend to have either no activation or activations everywhere. Note that the fine attention map nonetheless highlights cars in the background, which, by zooming in, can be verified to truly be present in the image. All predictions are with confidence 50% and above

Table 3 Results on KITTI to Cityscapes adaptation

Method	mAP@0.5
[19] - w/o DA	33.3
[19] - global	23.3
[19] - global + local	27.8
I^{3}Net [5]	40.0
SSD - w/o DA	33.1
SSD + our DA	**40.5**
SWDA [36]	39.0
HTCN$^{\psi}$ [4]	32.3

the individual categories, we observe that we outperform all methods on *car*, *rider*, and additionally yield better results than [19] on *bicycle*, with on par performance on *train* and *bus*. In some categories, such as *car*, our approach yields an increase in mAP by 10% compared to [19]. We attribute our poor performance on *train* and *truck* to the fact that these categories are under-represented in the source domain, and that their similar elongated shapes creates confusion between these classes. We outperform [5] on most of the categories and increase the mAP score by 29.5% and 61% for *car* and *bus*, respectively. This shows the effectiveness of our method. Both SWDA and HTCN$^{\psi}$ suffer from the lack of rich foreground information in SSD, which contrasts with the two-staged detector they were originally developed for. HTCN$^{\psi}$ additionally relies on context vectors trained with ROIs and translated images to improve performance. The unavailability of these leads to even worse performance than our *SSD - w/o DA*.

In Fig. 4, we provide examples of detections and attention maps predicted with our approach on the **C**→**F** task. Despite the challenging nature of this adaptation problem, our method

Table 4 Results on Cityscapes to Foggy adaptation

Method	mAP@0.5								
	Person	Car	Train	Rider	Truck	Motorcycle	Bicycle	Bus	Mean
YOLO - w/o DA	27.1	40.8	4.5	30.8	11.1	9.3	21	24.7	21.1
YOLO + obj w DA	31.8	50.3	4.9	33.9	18.5	**12.7**	**25.8**	34.3	26.5
YOLO + our DA	**32.8**	**51.3**	**16.2**	**35.7**	**18.8**	11.8	25.6	**34.5**	**28.3**

Table 5 Results on Sim10K to Cityscapes adaptation

Method	mAP@0.5
YOLO - w/o DA	42.5
YOLO + obj w DA	43.5
YOLO + our DA	**44.9**

Table 6 Results on KITTI to Cityscapes adaptation

Method	mAP@0.5
YOLO - w/o DA	29.1
YOLO + obj w DA	37.5
YOLO + our DA	**37.7**

correctly highlights the objects in the scene. The attention maps at different scales focus on objects of different sizes. We show additional qualitative results pre- and post-adaptation in Fig. 6.

Table 2 shows the results for the **S→C** adaptation. Our method again yields the best results, outperforming both [19] and [5]. Surprisingly, the global alignment of [19] yields better performance than when further exploiting their local alignment. This suggests that both should not be given equal importance as training progresses. Our method also outperforms our baseline without any attention, hence validating the importance of accounting for the foreground regions during feature alignment. HTCN^{ψ} without instance-aware adaptation performs worse than the other baselines, suggesting its reliance on the foreground adaptation.

In Fig. 5, we provide qualitative results for the **S→C** task. These results evidence that the attention maps we produce correctly focus on the local regions of interest, i.e., the cars in this case. Furthermore, the maps at different scales account for objects at different sizes. Note that attention maps with no activations or activations everywhere indicate the absence of any object of that scale, and will typically lead to predictions with low confidence because the model has learned to ignore those cases during training. We show additional qualitative results pre- and post-adaptation in Fig. 7.

We provide the **K→C** results in Table 3. Note that the method of [19] fails to adapt to the target data, yielding worse performance than their own no-DA baseline. This difference compared to the results provided in [19] arises from the use of a smaller image size here, as discussed above. Note, however, that the fact that the [19]- *w/o DA* baseline, which we also re-trained, yields essentially the same performance as our *SSD - w/o DA* baseline, and that the method of [19] yields reasonable performance in the other source-target pairs evidence that we correctly re-trained this model. For this adaptation task, we achieve comparable results with [5] even though we adopt simpler training and architecture choices. Again, the worse performance of HTCH^{ψ} can be attributed to the lack of instance-specific loss. We show qualitative results for this task in Fig. 8.

4.3.3 Generalization to another architecture

To show the generality of our approach, we use it with the YOLOv5 detector. We compare our method with an additional baseline *YOLO + obj w DA*. This baseline leverages the fact that the YOLO architecture predicts an objectness score for each anchor box at each feature map location. We thus use the maximum score at each location to create an objectness map and replace our A_s, learned using self-attention, with this map. Furthermore, we provide the results of the YOLOv5 architecture without domain adaptation as *YOLO w/o DA*.

The results on **C→F**, **S→C**, and **K→C** are shown in Tables 4, 6, and 7, respectively. As in the SSD case, our method consistently outperforms the baselines, illustrating the generality of our approach. *YOLO + obj w DA* performs worse than us on **S→C**, **C→F** and comparably on **K→C**. This further shows that our attention scheme helps to learn better objectness maps.

4.4 Ablation study

4.4.1 Global versus local alignment

As mentioned in Sect. 3.2, our formulation in Eq. 4 is motivated by the intuition that one should initially perform a global alignment to learn reliable features for the attention module, but that the global features can be gradually dropped to focus on local regions in the later training stages. To further evaluate the benefits of local vs global alignment, we

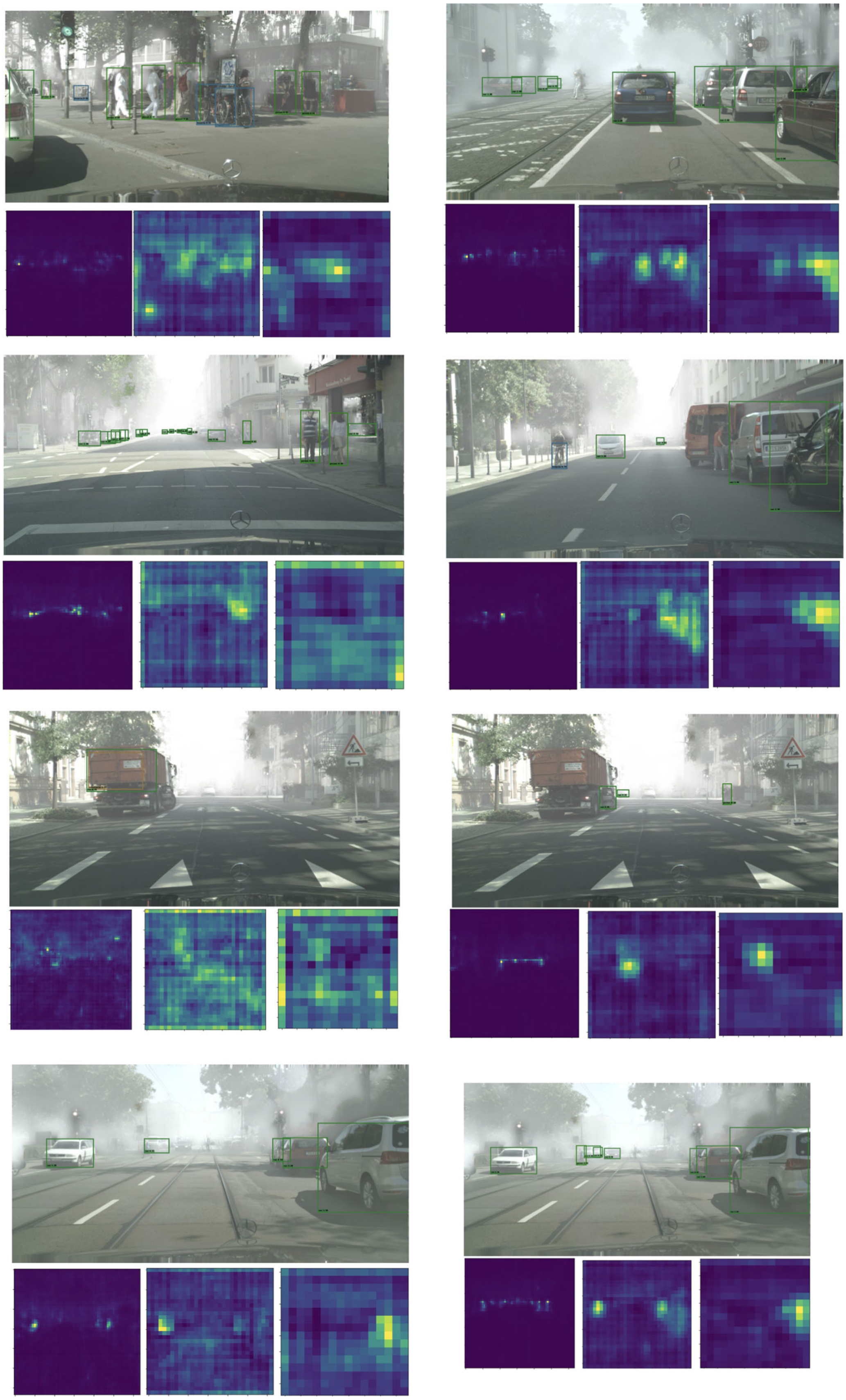

Fig. 6 Qualitative results on C → F. We show targeted images with predicted detections, together with attention maps at different scales. Recall that here we consider multiple classes. All predictions are with confidence 50% and above. Bottom two rows: We show the predictions and attention maps before (left) and after (right) adaptation. We are able to reduce the false positives and improve the detection on smaller objects in this case.

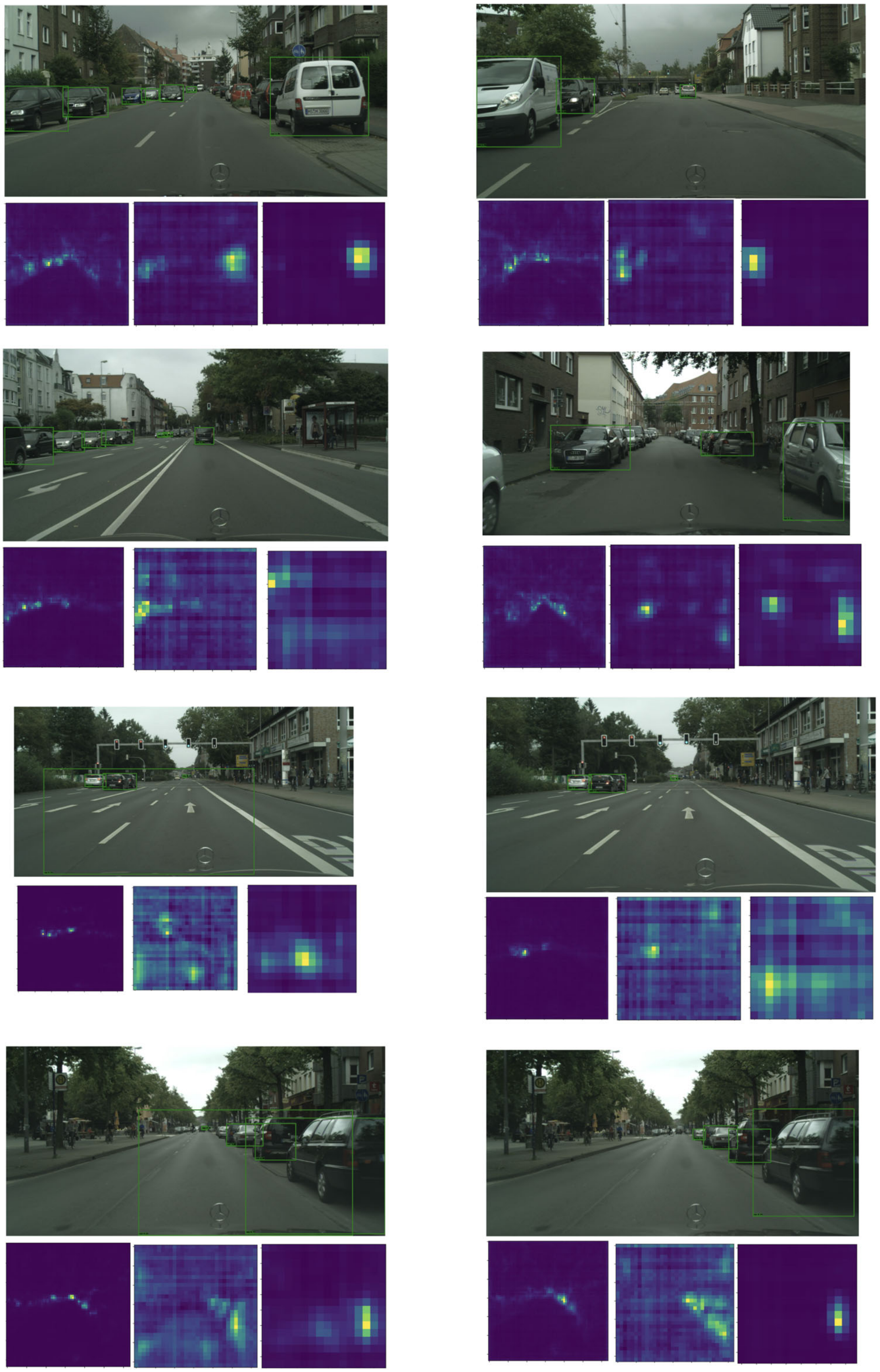

Fig. 7 Qualitative results on S→ C. We show targeted images with predicted detections, together with attention maps at different scales. All predictions are with confidence 50% and above. Bottom two rows: We show the predictions and attention maps before (left) and after (right) adaptation. We can see we suppress the false positives by learning better attention maps (middle)

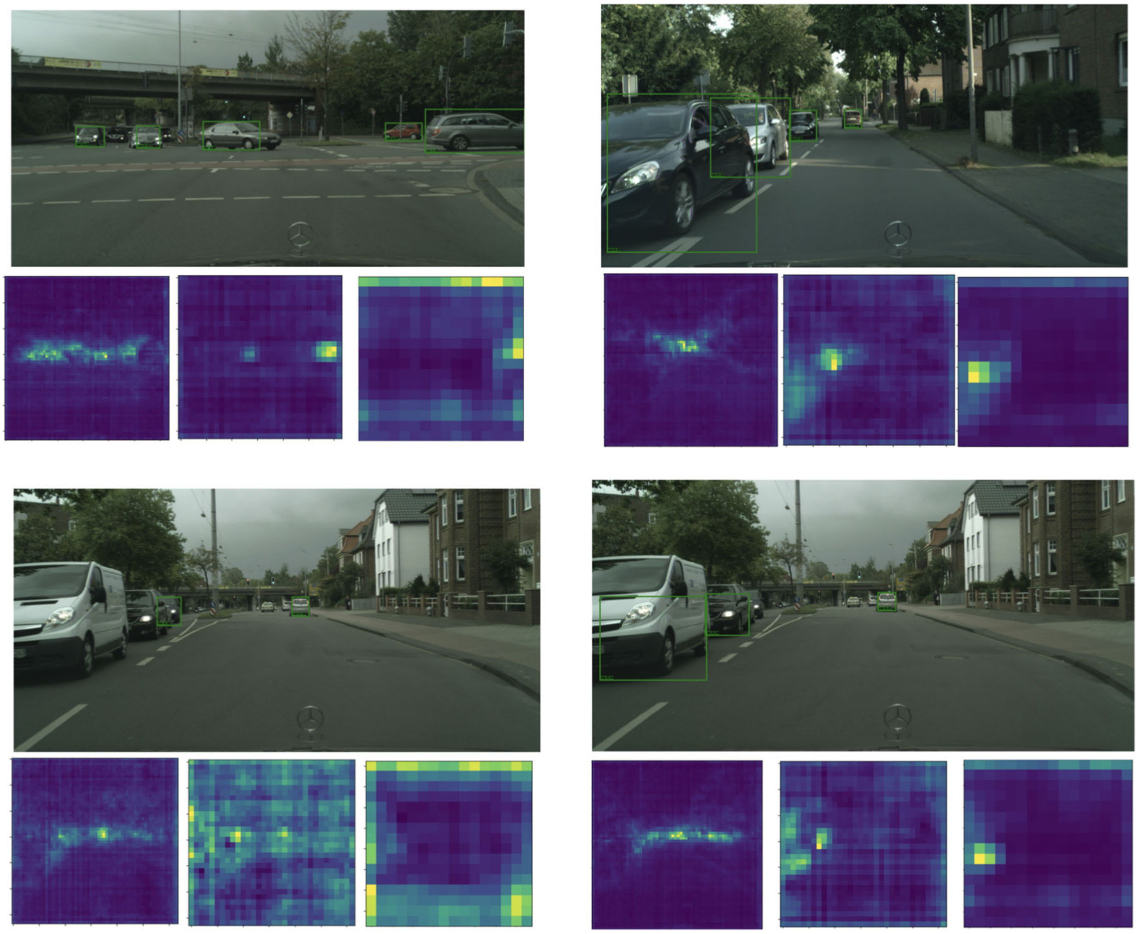

Fig. 8 Qualitative results on K → C. We show targeted images with predicted detections, together with attention maps at different scales. All predictions are with confidence 50% and above. Bottom row: We show the predictions and attention maps before (left) and after (right) adaptation. After adaptation, we see attention maps to be more focused on the foreground objects

implemented three alternative strategies: **(a)** The global features are maintained throughout the whole training process. Concretely, this strategy computes a features map of the form

$$M_d = (F_s + G_s) + \gamma * (F_s + G_s) \odot A_s \ , \qquad (9)$$

where γ follows the same rule as in our approach. **(b)** We set $\gamma = 1$ in Eq. 4, which corresponds to performing adaptation using only local features throughout the whole training process. **(c)** We set $\gamma = 0$ in Eq. 4, which corresponds to a global alignment where the attention block is nonetheless employed via G_s but the attention maps are not used to modulate the features.

As shown in Table 7 for the **S→C** task and with an SSD-based detector, our approach outperforms all of these baselines. This confirms that maintaining a global alignment term throughout training harms the overall performance, suggesting that the transition from global to local is crucial. This is further supported by the fact that local or global alignment on their own performs better than combining both in a suboptimal fashion. Purely local adaptation yields worse results than purely global adaptation because the attention maps do not carry sufficient meaningful information at the beginning of training, which compromises the rest of the training process. This study shows that both global and local alignments are important, and that their interaction affects the overall performance.

4.4.2 Hyperparameter study

In this section, we further investigate the influence of attention on our results. To this end, we first study the effect of δ in $\gamma = \frac{2}{1+\exp(-\delta \cdot r)} - 1$ for **S→C** with SSD. Table 8 shows mAP scores for strategies ranging from local alignment (γ=1) to more global alignment (δ=0.5). Figure 9 depicts the evolution of γ for different values of δ. For $\delta = 10, 5$ we see that the transition from global to local is relatively fast, which yields better results than the slower transition $\delta = 1, 0.5$ and γ=r^3.

Table 7 Global versus local alignment on **S** → **C**

Method	mAP@0.5
Ours w. Eq. 9	32.9
Ours w. γ =1	33.6
Ours w. γ = 0	34.2
Ours	**36.7**

Table 8 Hyperparameter study on **S** → **C**

Method	mAP@0.5
Ours w. γ =1; large δ	33.6
Ours w. δ=10	35.6
Ours w. δ=5	**36.7**
Ours w. $\gamma = x$	35.7
Ours w. $\gamma = x^3$	33.0
Ours w. δ=1	33.4
Ours w. δ=0.5	33.3

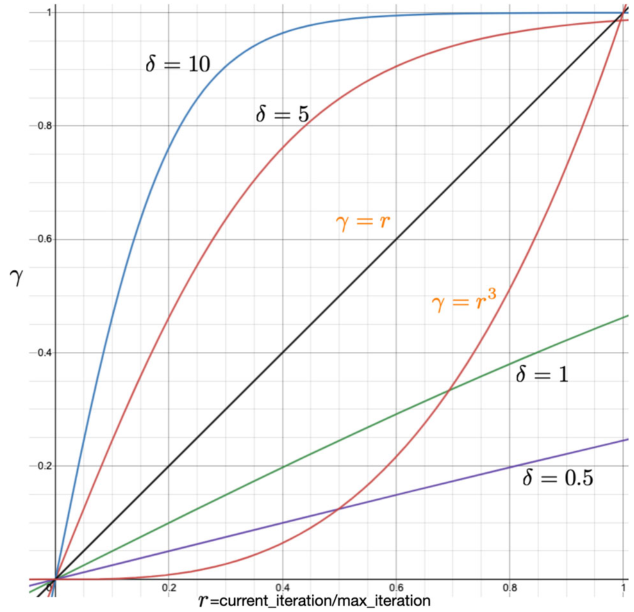

Fig. 9 Study of different variants of γ. We plot the evolution of γ throughout training for different values of δ. We also study other functions highlighted in orange

We attribute this to the fact that the network becomes biased toward global features if the transition is slow. Moreover, for $\delta = 1, 0.5$, the local features are never given much importance as γ is always below 0.5. Finally, we see that a linear function $\gamma = r$ yields a similar score to that obtained with a nonlinear function with $\delta = 10$, suggesting that transition leads to a better result, thereby validating our claim of the importance of global adaptation in the initial training stages and local adaptation toward the end.

Table 9 Effectiveness of our attention mechanism on different adaptation tasks. We report the mAP@0.5 in the target domain

Adapt	Method	SSD	YOLO
S→C	w/o attn	35.1	42.7
	attn	**36.7**	**44.9**
K→C	w/o attn	39.9	37.4
	attn	**40.5**	**37.7**
C→F	w/o attn	24.1	25.9
	attn	**24.8**	**28.3**

4.4.3 Importance of attention

To show the importance of attention, we trained both the SSD and YOLO detectors without and with our attention mechanism, along with domain adversarial training. As shown in Table 9, our attention scheme consistently improves the performance in the target domain for all the adaptation tasks.

5 Conclusion

To conclude, we have proposed to incorporate an attention module acting on the features extracted by the detector backbone, and to modulate these features so as to focus adaptation on the local foreground image regions that truly matter for detection. We have further developed a gradual training strategy that smoothly transitions from global to local feature alignment. Our experiments on several domain adaptation benchmarks have demonstrated that (i) with a comparable architecture, our method outperforms the state-of-the-art domain adaptation techniques for single-stage detection, despite the fact that they were designed for specific architectures; (ii) our approach remains effective across different single-stage detectors; (iii) our gradual training strategy effectively allows the network to benefit from global and local adaptation. In the future, we will study the use of pseudo labels with our local feature alignment strategy. We will also investigate the use of our method for multi-source domain adaptation, similarly to the scenario studied in [11,17,51] for image recognition and semantic segmentation.

Acknowledgements This work was funded in part by Swiss Innovation Agency (Innosuisse) grant 35935.1 IP-ICT.

Funding Open access funding provided by EPFL Lausanne.

References

1. Bello, I., Zoph, B., Vaswani, A., Shlens, J., Quoc, V.L.: Attention augmented convolutional networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3286–3295 (2019)
2. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et,al.: Language models are few-shot learners. arXiv preprint arXiv:2005.14165 (2020)
3. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: Detr. https://github.com/facebookresearch/detr (2020)
4. Chen, C., Zheng, Z., Ding, X., Huang, Y., Dou, Q.: Harmonizing transferability and discriminability for adapting object detectors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8869–8878 (2020)
5. Chen, C., Zheng, Z., Huang, Y., Ding, X., Yu, Y.: I3net: implicit instance-invariant network for adapting one-stage object detectors. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
6. Chen, Y., Li, W., Sakaridis, C., Dai, D., Van Gool, L.: Domain adaptive faster r-cnn for object detection in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3339–3348 (2018)
7. Corporation, NVIDIA. NVIDIA Tesla V100 GPU Architecture. (Technical Report WP-08608.) http://www.nvidia.com/object/volta-architecture-whitepaper.html (2017)
8. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3213–3223 (2016)
9. Deng, J., Li, W., Chen, Y., Duan, L.: Unbiased mean teacher for cross-domain object detection. In: Proceedings Of The IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4091–4101 (2021)
10. Devlin, J., Chang, M., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
11. Dong, J., Fang, Z., Liu, A., Sun, G., Liu, T.: Confident anchor-induced multi-source free domain adaptation. Adv. Neural Inf. Process. Syst. **34** (2021)
12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G.,. Gelly, S, et,al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
13. Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. **88**, 303–338 (2010)
14. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3146–3154 (2019)
15. Ganin, Yaroslav: Ustinova, Evgeniya, Ajakan, Hana, Germain, Pascal, Larochelle, Hugo, Laviolette, François, Marchand, Mario, Lempitsky, Victor: Domain-adversarial training of neural networks. J. Mach. Learn. Res. **17**(1), 2096–30 (2016)
16. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The Kitti vision benchmark suite. In 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 3354–3361. IEEE (2012)
17. He, J., Jia, X., Chen, S. & Liu, J.: Multi-source domain adaptation with collaborative learning for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11008–11017 6 (2021)
18. Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A., Darrell, T.: Cycada: cycle-consistent adversarial domain adaptation. In: International Conference on Machine Learning, pp. 1989–1998. PMLR (2018)
19. Hsu, C.-C., Tsai, Y.-H., Lin, Y.-Y., Yang, M.-H.: Every pixel matters: center-aware feature alignment for domain adaptive object detector. In: European Conference on Computer Vision, pp. 733–748. Springer (2020)
20. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7132–7141 (2018)
21. Jocher, G., Stoken, A., Borovec, J., NanoCode012, Stan, C., Changyu, L., Tkianai, L., Hogan, A., Yxnong, l., AlexWang1900, Diaconu, L., Wanghaoyang0106, M., ml5ah, Doug, Ingham, F., Frederik, Hatovix, G., Poznanski, J., Fang, J., Yu, L., Changyu98, Wang, M., Gupta, N., Akhtar, O., Dvoracek, P., Rai, P.: ultralytics/yolov5: v3.1—bug Fixes and Performance Improvements, (Oct. 2020)
22. Johnson-Roberson, M., Barto, C., Mehta, R., Sridhar, S.N., Rosaen, K., Vasudevan, R.: Driving in the matrix: can virtual worlds replace human-generated annotations for real world tasks? arXiv preprint arXiv:1610.01983 (2016)
23. Kim, S., Choi, J., Kim, T., Kim, C.: Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6092–6101 (2019)
24. Li, C., Du, D., Zhang, L., Wen, L., Luo, T., Wu, Y., Zhu, P.: Spatial attention pyramid network for unsupervised domain adaptation. In: European Conference On Computer Vision, pp. 481–497 (2020)
25. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
26. Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
27. Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D.C., Zitnick, L., Dollár, P.: Microsoft coco: common objects in context (2015)
28. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, C. A.: Ssd: single shot multibox detector. In: European Conference on Computer Vision, pp. 21–37. Springer (2016)
29. Liu, Y.-C., Yeh, Y.-Y., Fu, T.-C., Wang, S.-D., Chiu, W.-C., Frank Wang, Y.-C.: Detach and adapt: Learning cross-domain disentangled deep representation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8867–8876 (2018)
30. Long, M., Cao, Y., Wang, J., Jordan, M.I.: Learning transferable features with deep adaptation networks. In: Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6–11 July 2015, pp. 97–105 (2015)
31. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L.: Others Pytorch: an imperative style, high-performance deep learning library. Adv. Neural Inf. Process. Syst. **32** (2019)

32. Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., Shlens, J.: Stand-alone self-attention in vision models. arXiv preprint arXiv:1906.05909 (2019)
33. Ramamonjison, R., Banitalebi-Dehkordi, A., Kang, X., Bai, X. Zhang, Y.: Simrod: a simple adaptation method for robust object detection. In: Proceedings Of The IEEE/CVF International Conference on Computer Vision, pp. 3570–3579 (2021)
34. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
35. Ren, Shaoqing: He, Kaiming, Girshick, Ross, Sun, Jian: Faster r-cnn: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2016)
36. Saito, K., Ushiku, Y., Harada, T., Saenko, K.: Strong-weak distribution alignment for adaptive object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6956–6965 (2019)
37. Saito, K., Watanabe, K., Ushiku, Y., Harada, T.: Maximum classifier discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3723–3732 (2018)
38. Sakaridis, C., Dai, D., Van Gool, L.: Semantic foggy scene understanding with synthetic data. Int. J. Comput. Vis. **126**(9), 973–992 (2018)
39. Shen, Z., Maheshwari, H., Yao, W., Savvides, M.: Scl: towards accurate domain adaptive object detection via gradient detach based stacked complementary losses. arXiv preprint arXiv:1911.02559 (2019)
40. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
41. Tan, M., Pang, R., Le, V. Q.: Efficientdet Scalable and efficient object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10781–10790 (2020)
42. Tian, Z., Shen, C., Chen, H., He, T.: Fcos: fully convolutional one-stage object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9627–9636 (2019)
43. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. arXiv preprint arXiv:1706.03762 (2017)
44. VS, V., Gupta, V., Oza, P., Sindagi, V., Patel, V.: MeGA-CDA memory guided attention for category-aware unsupervised domain adaptive object detection. In: Proceedings Of The IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4516–4526 (2021)
45. Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., Tang, X.: Residual attention network for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3156–3164 (2017)
46. Wang, X., Li, L., Ye, W., Long, M., Wang, J.: Transferable attention for domain adaptation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 5345–5352 (2019)
47. Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: Cbam: convolutional block attention module. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19 (2018)
48. Zhu, J.-Y., Park, T., Isola, P., Efros, A. A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2223–2232 (2017)
49. Zhu, X., Pang, J., Yang, C., Shi, J., Lin, D.: Adapting object detectors via selective cross-domain alignment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 687–696 (2019)
50. Zhang, J., Huang, J., Luo, Z., Zhang, G., Lu, S.: Da-detr: domain adaptive detection transformer by hybrid attention. ArXiv Preprint ArXiv:2103.17084 (2021)
51. Zhao, S., Li, B., Yue, X., Gu, Y., Xu, P., Hu, R., Chai, H., Keutzer, K.: Multi-source domain adaptation for semantic segmentation. Adv. Neural Inf. Process. Syst. **32** (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Vidit Vidit is a PhD candidate at EPFL-CVLab under the supervision of Dr. Mathieu Salzmann. He completed his M.Sc. from EPFL in 2018. His interest lies in object detection, domain adaptation, and in general, computer vision.

Mathieu Salzmann is a Senior Researcher at EPFL-CVLab, and, since May 2020, a part-time Artificial Intelligence Engineer at ClearSpace. Previously, he was a Senior Researcher and Research Leader in NICTA's computer vision research group. Prior to this, from Sept. 2010 to Jan 2012, he was a Research Assistant Professor at TTI-Chicago, and, from Feb. 2009 to Aug. 2010, a postdoctoral fellow at ICSI and EECS at UC Berkeley under the supervision of Prof. Trevor Darrell. He obtained his PhD in Jan. 2009 from EPFL under the supervision of Prof. Pascal Fua. Mathieu Salzmann's research lies at the intersection of machine learning and visual recognition.