ConfMix: Unsupervised Domain Adaptation for Object Detection via Confidence-based Mixing

Giulio Mattolin\textsuperscript{1}, Luca Zanella\textsuperscript{2}, Elisa Ricci\textsuperscript{1,2}, Yiming Wang\textsuperscript{2}
\textsuperscript{1}University of Trento, Trento, Italy \textsuperset{2}Fondazione Bruno Kessler, Trento, Italy
lzanella@fbk.eu

Abstract

Unsupervised Domain Adaptation (UDA) for object detection aims to adapt a model trained on a source domain to detect instances from a new target domain for which annotations are not available. Different from traditional approaches, we propose ConfMix, the first method that introduces a sample mixing strategy based on region-level detection confidence for adaptive object detector learning. We mix the local region of the target sample that corresponds to the most confident pseudo detections with a source image, and apply an additional consistency loss term to gradually adapt towards the target data distribution. In order to robustly define a confidence score for a region, we exploit the confidence score per pseudo detection that accounts for both the detector-dependent confidence and the bounding box uncertainty. Moreover, we propose a novel pseudo labelling scheme that progressively filters the pseudo target detections using the confidence metric that varies from a loose to strict manner along the training. We perform extensive experiments with three datasets, achieving state-of-the-art performance in two of them and approaching the supervised target model performance in the other. Code is available at https://github.com/giuliomattolin/ConfMix.

1. Introduction

Object detection is a fundamental task in computer vision which involves the classification and localisation, e.g. by bounding boxes, of objects of interest belonging to certain predefined categories. Due to its importance in many applications such as autonomous driving, video surveillance and robotic perception, object detection has received significant attention, leading to the development of several different models [15, 34, 41, 35]. However, as detectors mostly rely on deep learning, it is a well known fact that they suffer from severe performance degradation when being tested on images that are visually different from the ones encountered during training, due to the domain shift [4].

To address this problem, recent research efforts have been put on devising Unsupervised Domain Adaptation (UDA) techniques for building deep models that can adapt from an annotated source dataset to a target one without tedious manual annotations [40, 11, 1, 16, 9]. The vast majority of UDA methods for detection resort on adversarial training and on exploiting the Gradient Reversal Layer (GRL) [11] to perform adaptation both at image-level and instance-level [4, 37, 55, 39, 43]. Other approaches mostly focus on robustly producing pseudo detections in order to effectively finetune the model on the target data [49, 52, 44]. In general, while over the last few years several solutions have been proposed in the literature for adapting two-stage object detectors, we argue that devising UDA approaches which can also be applied to one-stage detectors would be desirable. Indeed, the latter methods are more appropriate in applications such as autonomous driving that necessitate of real-time processing and high computational efficiency.
Concurrently, recent works in computer vision have shown the benefit of adopting sophisticated data augmentation techniques by synthesising mixed samples with target and source images in order to improve generalisation ability of deep architectures [51, 50, 17]. These methods have been considered in the context of UDA for classification [45, 30] and semantic segmentation [10, 32, 20, 5], demonstrating some empirical advantage. However, extending these approaches to UDA for detection is far from trivial.

Inspired by these previous works, in this paper we propose ConfMix, the first mixing-based UDA approach for object detection based on the regional confidence of pseudo detections. The main idea behind ConfMix is illustrated in Figure 1. Specifically, we propose to artificially generate samples by combining the region of target images where the model is most confident with source images. We also introduce during training an associated consistency loss to enforce coherent predictions among generated images. Our intuition is that, by combining source and target images and forming new mixed samples, we are training our model on novel, synthetically generated sample images with reliable pseudo detections and with visual appearance close to the samples of target domain, thus improving the generalisation capabilities of the detector. Moreover, the quality of pseudo detections plays an essential role during adaptation, and is tightly related to the confidence metric. By exploiting a stricter confidence metric, e.g. enriching the detector-dependent confidence with bounding box uncertainty [6], one can obtain more reliable pseudo detections, however with a reduced number. To mitigate this, we propose to progressively restrict the confidence metric for pseudo labelling. With a less strict confidence metric at the initial adaptation phase, we allow more pseudo detections in order to learn the representation of the target domain, while with a gradually stricter confidence metric, we aim to improve the detection accuracy with more trustworthy pseudo detections. We conduct extensive experiments on different datasets (Cityscapes [7] → FoggyCityscapes [38], Sim10K [19] → Cityscapes and KITTI [13] → Cityscapes) and we show that our approach outperforms existing algorithms in most setups.

We summarise our main contributions as below:

- We introduce the first sample-mixing UDA method for object detection. Our approach, named ConfMix, mixes samples from source and target domains based on the regional confidence of target pseudo detections.
- We propose a novel Progressive Pseudo-labelling scheme by gradually restricting the confidence metric along the adaptive learning, which allows for a smooth transition when learning target representation, thus improving detection accuracy.
- ConfMix scores the new state-of-the-art adaptation performance, achieving +1.7% on Sim10k → Cityscapes, and +3.7% on KITTI → Cityscapes in terms of mean Average Precision (mAP).

2. Related work

Object Detection. Current object detection models can be grouped into two main categories: one-stage and two-stage approaches. One-stage object detectors, such as YOLO [34] and FCOS [41], adopt a unified framework to obtain final results directly from the feature maps generated by a CNN backbone. These frameworks are very computationally efficient and are able to achieve near real-time speed during inference. On the other hand, two-stage object detectors, such as RCNN [15], generate predictions by first extracting region proposals and then, leveraging this information, produce classification labels and bounding box coordinates. Such models are widely adopted for their high performance but, although research has been conducted to improve detection speed [14, 35, 8], they are considerably slower compared to one-stage detectors.

Unsupervised Domain Adaptation. Given a labelled source domain and an unlabelled target domain, UDA aims to use the available data to produce a model that is able to generalise and perform well on the target domain. A conventional approach is to reduce the domain gap by directly minimising the distance between feature distributions using discrepancy loss functions [29, 40]. On the other hand, adversarial-based methods [11, 12, 42], employ a domain discriminator and a feature extractor that learns to produce domain-invariant feature representations by fooling the discriminator. Many works demonstrated the benefit of using pseudo labels to maximally leverage information from the target domain [28, 23, 24], eventually considering a gradual scheme for incorporating them [48]. Other works have focused on adopting sample mixing techniques, such as mixup [51] or CutMix [50], to improve generalisation. For instance, in [45, 47] domain-level mixup regularisation is applied to ensure domain invariance in the learned feature representations, while in [3, 33] the model’s attention is used to re-assign the confidence of saliency-guided samples and labels. Similar ideas are implemented in previous works considering the segmentation task [10, 32, 20, 5, 31]. However, to the best of our knowledge, no previous works have been proposed to exploit mixing techniques for UDA in the context of object detection.

UDA for Object Detection. In the context of object detection, UDA was recently introduced by [4], which proposed image- and instance-level alignment using two GRLs [11] on Faster R-CNN. Subsequently, several methods started to address this problem mainly using two-stage detectors. Focusing on image-level, [37] showed that strong-local alignment and weak-global alignment of the features extracted from the backbone improve adaptation, while [55], focusing on instance-level, exploited RPN proposals to perform...
region-level alignment. To adapt the source-biased decision boundary to the target data, [2] combined adversarial training with image-to-image translation by generating interpolated samples using Cycle-GAN [54]. Other recent works have proposed applying self-training with pseudo detections to perform the adaptation. To address the risk of performance degradation caused by overfitting noisy pseudo detections, [49] introduces an uncertainty-based fusion of pseudo detections sets generated via stochastic inference, [27] proposes self-entropy descent (SED) as a metric to search for an appropriate confidence threshold for reliable pseudo detections, while [44] uses a student-teacher framework and gradually updates the source-trained model.

Few works have addressed UDA for one-stage detectors, e.g., FCOS [25, 26, 18] or SSD [21]. In particular, adopting a self-training procedure reduces the negative effects of inaccurate pseudo detections by performing hard negative pseudo detections mining followed by a weak negative mining strategy, where instance-level scores are computed for each detection considering all neighbouring boxes [21]. In addition, adversarial learning is employed using GRL [11] and a discriminator with the aim of extracting discriminative background features and reducing the domain shift. However, our approach is radically different, as it does not require additional architectural components to the network, but proposes a mixing-based data augmentation strategy to promote regularisation of the model.

3. Method

The proposed ConfMix, as illustrated in Figure 2, synthesises an image \( x_M \in \mathbb{R}^{W \times H \times C} \) by mixing a source image \( x_S \in \mathbb{R}^{W \times H \times C} \) and the local region of a target image \( x_T \in \mathbb{R}^{W \times H \times C} \) with the most reliable pseudo detections. We first predict a set of \( N_T \) pseudo detections \( \hat{y}_T = \{ \hat{y}_T^i \mid i \in [1, N_T] \} \) on the target image and compute the confidence per pseudo detection using the detector network \( F(\Theta) \) that is parameterised with \( \Theta \) and is originally trained only on the source data. We opt to follow a Gaussian modelling of the bounding box predictions, instead of the deterministic one, in order to improve the reliability of the detector confidence with the uncertainty of the bounding box prediction. Next, we divide the target image \( x_T \) into regions of equal size and select the region with the highest average confidence of pseudo detections to mix with the source sample \( x_S \), forming the mixed sample \( x_M \).

We pass \( x_T, x_S, \) and \( x_M \) to the detector \( F(\Theta) \) and obtain their corresponding detections \( \hat{y}_T, \hat{y}_S \) and \( \hat{y}_M \), respectively. The detector then learns to adapt to the target domain by imposing a consistency loss \( L_{cons} \) which promotes the similarity between \( \hat{y}_M \) and the combined detections \( \hat{y}_{S,T} \) by merging the source \( \hat{y}_S \) and target \( \hat{y}_T \) detections according to how the two sample images are mixed. The supervision of source ground-truth detections \( y_S \) is achieved with the detector-related loss \( L_{det} \) in order to maintain the detector capability during adaptation.

In the following sections, we describe our proposed ConfMix in details, where we first introduce the estimation of the Gaussian-based detection confidence in Sec. 3.1, followed by the confidence-based region mixing strategy for synthesising training samples in Sec. 3.2 and the progressive pseudo labelling in Sec. 3.3. Finally, we present the training objectives with losses in Sec. 3.4.

3.1. Gaussian-based detection confidence

Conventional object detectors, such as YOLO [34], Faster R-CNN [35] and FCOS [41], compute and assign to each detection a confidence score \( C_{det} \in [0, 1] \) that is often detector-dependent and is used to filter out unreliable predictions via non-maximum suppression. However, such confidence score does not account for the reliability of the predicted bounding box \( \hat{bx} = [bx, by, bh, bw] \), where \( [bx, by] \) are the position of bounding box on the image and \( bh \) and \( bw \) represent the height and width, respectively. As suggested in [6], by taking into consideration both the detector-dependent confidence and the confidence that is derived from the uncertainty of bounding box prediction, one can improve the reliability of pseudo detections and reduce the number of false positives.

In order to compute the bounding box uncertainty, \( \hat{b} \) requires a Gaussian-based modelling. Specifically, for each element in \( \hat{b} \), the detector model predicts both a mean \( \mu \) and a variance \( \Sigma \), where the variance expresses the localisation uncertainty. Thus, we can express the Gaussian-based bounding box \( \hat{b} \) as:

\[
\hat{b} = [\mu_{bx}, \mu_{by}, \mu_{bh}, \mu_{bw}, \Sigma_{bx}, \Sigma_{by}, \Sigma_{bh}, \Sigma_{bw}],
\]

where both the means \( \hat{b}_\mu = [\mu_{bx}, \mu_{by}, \mu_{bh}, \mu_{bw}] \) and the variances \( \hat{b}_\Sigma = [\Sigma_{bx}, \Sigma_{by}, \Sigma_{bh}, \Sigma_{bw}] \) are predicted by the detector with an updated regression loss (see details in Sec. 3.4). Note that a sigmoid function \( \sigma(\cdot) \) is applied to the predicted variance value to ensure its range is between 0 and 1.

As a larger variance value implies a higher uncertainty, the confidence of a bounding box is computed as:

\[
C_{bbx} = 1 - \text{mean}(\hat{b}_\Sigma),
\]

where \( \text{mean}(\cdot) \) computes the average variance of \( \hat{b}_\Sigma \).

The combined confidence can thus be computed as:

\[
C_{comb} = C_{det} \cdot C_{bbx}.
\]
Figure 2. Overview of the proposed ConfMix method. We pass the source sample $x_S$ and target sample $x_T$ to the detector model $F(\Theta)$, obtaining the prediction $\tilde{y}_S$ and $\tilde{y}_T$, respectively. We select the target region with the highest regional confidence to form the mixed sample $x_M$ with the source image, which is then fed to the detector model $F(\Theta)$, producing predictions $\tilde{y}_M$. We train the model with the supervised detection loss using the source annotations $y_S$ and the self-supervised consistency loss by comparing $\tilde{y}_M$ with the combined source and target predictions $\tilde{y}_{S,T}$.

detections. Instead of extracting only pseudo detections or randomly selecting part of the target image [50] to mix, we propose a novel region-level mixing strategy whose synthesised samples contain both the foreground and background features from the two domains, contributing to a more effective adaptation towards the target domain.

Specifically, we randomly sample a source image $x_S$ and a target image $x_T$. The target $x_T$ is then passed to the object detector $F(\Theta)$, producing the predictions $\tilde{y}_T$. The target image $x_T$ is then evenly divided into 4 regions as shown in Figure 1. Each region is considered to contain a prediction $\tilde{y}^i_T$ if its centre coordinate resides within the region. The region confidence is computed as the average of the confidences of all the pseudo detections that lie within the region. We select the region with the highest region confidence to mix with the sampled source image $x_S$ and generate the synthesised image $x_M$:

$$x_M = M_T \odot x_S + (1 - M_T) \odot x_T,$$

where $M_T \in R^{W \times H}$ is the mask matrix indicating which pixels of the target image should be masked.

3.3. Progressive Pseudo Labelling

The correctness of the pseudo detections are tightly related to the confidence metric which is used for filtering the detections. At the early stage of the adaptation, the confidence tends to be less reliable and in general of a lower value due to the large domain gap. Thus, the non-maximum suppression ends up in filtering out most of the pseudo detections if a strict confidence metric, such as $C_{comb}$, is applied. Therefore, we propose to perform a gradual transition of the confidence metric from a loose to strict manner, to first learn an initial representation of the target domain by allowing more pseudo detections and then gradually shift towards a stricter confidence metric to improve detection accuracy with more reliable pseudo detections.

To this end, we start with the loose confidence metric $C_{det}$ for filtering the pseudo detections. As iterations continue, we progressively assign more importance to $C_{comb}$ with a shifting weight $\delta$:

$$C = (1 - \delta) \cdot C_{det} + \delta \cdot C_{comb}.$$  \hspace{1cm} (5)

The shifting weight $\delta$ varies based on the progress of the training, thus it is dependent on the iteration $t$, epoch $e$ and the number of batches in one epoch $N_b$. We devise $\delta$ with a non-linear function to gradually increase from 0 to 1:

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

$$r = \frac{t}{N_b \cdot e},$$

where $r$ is the ratio of the current iteration to the total number of iterations, with its scale modulated by $\alpha$.

The pseudo detection with a confidence value that is higher than a predefined threshold value $C_{th}$, i.e. $C > C_{th}$, is considered a valid detection, and it will be accounted during the confidence-based region mixing and the training for detector adaptation.

3.4. Adaptive detector training

To facilitate the adaptive learning of the detector $F(\Theta)$, we rely on two main losses: a self-supervised consistency
loss term $L_{\text{cons}}$ on the mixed samples and a supervised detector loss term $L_{\text{det}}$ on the labelled source samples. $L_{\text{det}}$ aims to maintain the task-specific knowledge during adaptation, while the consistency loss $L_{\text{cons}}$ aims to adapt towards the target representation by penalising the difference between $\hat{y}_M$ and the combined detections $\tilde{y}_{S,T}$ by merging the source $\hat{y}_S$ and target $\tilde{y}_T$ detections based on how the mixed samples are formed.

Specifically, let $\tilde{y}_{S,T}^R$ be the set of target pseudo detections residing within the selected target region, while $\tilde{y}_S^R$ be the set of source pseudo detections residing outside the selected target region. The combined detections $\tilde{y}_{S,T}$ is the union of the two sets, i.e. $\tilde{y}_{S,T} = \{\tilde{y}_{S,T}^R, \tilde{y}_S^R\}$. It can happen that the bounding box dimension of $\tilde{y}_{S,T}^R$ (or $\tilde{y}_S^R$) can exceed the selected target (or source) region, leading to inaccurate pseudo detections. We therefore clip such bounding boxes by their corresponding region boundary.

We define $L_{\text{cons}}$ and $L_{\text{det}}$ as $L_{\text{cons}} = L(\hat{y}_M, \tilde{y}_{S,T})$ and $L_{\text{det}} = L(\hat{y}_S, \tilde{y}_T)$, where both the supervised detection loss $L_{\text{det}}$ and the self-supervised consistency loss $L_{\text{cons}}$ share the same loss function $L(\cdot)$. While $L_{\text{det}}$ aims to penalise the difference between the predicted detections $\hat{y}_S$ and the ground-truth detections $y_S$ on the source samples, $L_{\text{cons}}$ aims to penalise the difference between the predicted detections $\hat{y}_M$ and the pseudo detections $\tilde{y}_{S,T}$ on the mixed samples. Note that $L(\cdot)$ is dependent on the employed object detector. In the case of the one-stage YOLOv5, $L(\cdot)$ is a combination of three terms: $L_{\text{box}}$ is a Complete-IoU (CIoU) loss for regressing the bounding box coordinates, $L_{\text{obj}}$ is the Binary Cross Entropy (BCE) loss for the objectness score and $L_{\text{cls}}$ is a BCE loss for the classification score.

In particular, as our predicted bounding box follows a Gaussian modelling, the regression loss per sample image is updated as follows:

$$L_{\text{box}} = \frac{1}{N} \sum_{i=1}^{N} (1 - \text{mean}(N(y^i_{\mu}, b_{\mu}^i, b_{\sigma}^i))),$$  \hspace{1cm} (8)

where $N(\cdot)$ is the probability density function of a normal distribution for calculating the conditional probability of obtaining the ground-truth $\hat{y}^i_S \in y_S$ for $L_{\text{det}}$, or pseudo detection $\tilde{y}^i_S \in \tilde{y}_S$ for $L_{\text{cons}}$, given the respective means $\hat{b}^i_{\mu}$ and variances $\hat{b}^i_{\sigma}$ predicted by the object detector. $N$ represents the total number of $\{y^i_S\}$.

Finally, the total loss is expressed as a weighted sum of $L_{\text{det}}$ and $L_{\text{cons}}$:

$$L_{\text{total}} = L_{\text{det}} + \gamma L_{\text{cons}},$$ \hspace{1cm} (9)

where $\gamma$ is a hyperparameter to balance the supervised and self-supervised terms. The consistency loss $L_{\text{cons}}$ can have a greater importance when the pseudo detections are more reliable, and vice versa. We therefore define $\gamma$ as the ratio of the number of pseudo detections on $x_M$ with confidence greater than $C_{th}$, and the total number of pseudo detections after non-maximum suppression, to reflect the reliability of the pseudo detections:

$$\gamma = \frac{\left| \tilde{y}_{S,T}^R \in \tilde{y}_{S,T} : C^i \geq C_{th} \right|}{|\tilde{y}_{S,T}|},$$ \hspace{1cm} (10)

where $|\cdot|$ is the cardinality of a set.

### 4. Experiments

We evaluate our proposed method ConfMix against state-of-the-art methods on three common benchmark adaptation scenarios, together with extensive ablation studies to prove the effectiveness of our design choices.

#### Datasets

We evaluate our method on the four datasets:

- **Cityscapes** [7] is a collection of urban street scenes for semantic understanding. Images were collected in 50 cities over several months, during the day, and in good weather conditions. Single instance annotations are available for the following 8 categories: person, rider, car, truck, bus, train, motorcycle and bicycle.
- **FoggyCityscapes** [38] is an extension of Cityscapes in which images are augmented by applying a fog filter. FoggyCityscapes includes the same images and 8 categories as Cityscapes.
- **Sim10K** [19] is a synthetic dataset consisting of 10,000 images derived from the video game Grand Theft Auto V, including only the car category.
- **KITTI** [13] is a dataset of several hours of traffic video recorded by high-resolution colour and greyscale cameras, containing 7481 training images with annotations provided for 8 categories: car, van, truck, pedestrian, person sitting, cyclist, tram and misc.

Following [18], we experiment on the benchmark Cityscapes $\rightarrow$ FoggyCityscapes regarding weather adaptation, Sim10K $\rightarrow$ Cityscapes regarding synthetic-to-real adaptation, and KITTI $\rightarrow$ Cityscapes regarding cross-camera adaptation. In the latter synthetic-to-real and cross-camera adaptations, we only consider the car category, while for Cityscapes $\rightarrow$ FoggyCityscapes, we consider the complete 8 categories.

#### Evaluation Metrics

We evaluate our proposed method on the target domain in terms of Average Precision (AP), which is computed by combining precision and recall for each object category separately. We obtain the mean AP (mAP) by averaging the AP across all object categories.

#### Implementation details

We based our experiments on the YOLOv5s architecture for its lightness among the YOLOv5 series, using PyTorch and the default settings. We set the batch size to 2, with each batch containing a source image and a target image of the size of $600 \times 600$. In all our experiments, we pre-train the model on the source domain for
20 epochs with the COCO-pretrained weights as initialisation, and perform adaptive learning for 50 epochs. In the non-maximum suppression stage, we set the IoU threshold to 0.5, and the confidence threshold \( C_{th} \) to 0.25 for producing pseudo detections. For the computation of \( \gamma \), we set the confidence threshold \( C_{th} \) to 0.5. Please refer details regarding the hyper-parameters in the Supplementary Material.

4.1. Comparisons

We compare ConfMix against recent state-of-the-art UDA approaches for adaptive object detection on three benchmarks. In particular, we compare with adversarial feature learning methods, such as MGA [53], MeGA-CDA [43], SSOD [36], EPM [18], CDN [39], SAPN [22]; pseudo-label based self-training techniques such as SCUDA [49], IRL [44], FL-UDA [27], CTRP [52]; and graph reasoning works such as SCAN [25], SIGMA [26], GIPA [46]. We also include “Source only”, the detector model that is trained only with labelled source data, serving as the performance lower-bound, and “Oracle”, the detector model that is trained with labelled target data, serving as the performance upper-bound.

Result Discussion. Table 1 reports the results of ConfMix and all compared methods in the synthetic-to-real scenario Sim10k → Cityscapes and the cross-camera scenario KITTI → Cityscapes. On both benchmarks, our ConfMix scores the new state-of-the-art adaptation performance, achieving +1.7% on Sim10k → Cityscapes, and +3.7% on KITTI → Cityscapes in terms of mAP.

Table 2 reports the per-class detection performance of ConfMix and all compared methods in the weather adaptation scenario Cityscapes → FoggyCityscapes. The gap between ConfMix and our upper-bound “Oracle” is rather narrow, i.e. -1.5%. Limited by the low “Oracle” performance on this benchmark, which is lower than MGA [53], it is quite unlikely our method can outperform existing approaches regarding mAP. Nevertheless, our method achieves a +1% AP gain in the person class compared to SIGMA [26] and a +2% AP gain in the car class compared to MGA [53], which are the approaches obtaining the second best AP in these classes, respectively.

Real-time analysis. We perform all experiments on a single NVIDIA Tesla V100. At adaptive training time, each epoch took ~13 minutes with a batch size of 2, while training YOLOv5 per epoch the time is ~6 minutes on Sim10K → Cityscapes. On the Cityscape dataset, the detection speed of ConfMix is 76 frames per second, almost equal to the 79 frames per second of YOLOv5.

4.2. Ablation study

We verify the impact of the main design choices of our method with an ablation study on Sim10K → Cityscapes. Does the confidence-based region mixing help? We analyse a variety of different mixing strategies and their impact in terms of object detection performance after adaptation. Specifically, we run our method using the \( C_{comb} \) confidence without the proposed progressive pseudo labelling. We vary the mixing strategy with 5 different options (as shown in Fig. 4): CutMix [50] randomly cuts a target region and mixes it with the source image; ConfMix (Vertical Mix) vertically cuts both source and target images in the middle and mixes the most confident target region; ConfMix (Horizontal Mix) horizontally cuts both source and target images in the middle and mixes the most confident target region; ConfMix (2-region Mix) selects the two most confident regions of the target image for mixing; and ConfMix selects only the most confident region of the target image for mixing. We further examine the 4-division scheme of ConfMix by varying the number of divided regions into 6 \((2 \times 3)\) and 9 \((3 \times 3)\), respectively.

As can be seen in Table 3, our ConfMix achieves a mAP gain of +5.6% compared to CutMix, meaning that considering the most confident target region to mix is more beneficial than randomly cutting a target region and mixing it with the source image for adaptation. ConfMix shows also a superior performance than ConfMix (Vertical Mix), ConfMix (Horizontal Mix), and ConfMix (2-region Mix). This means that mixing more target regions with the source image can impact the adaptation performance negatively, which might be due to the inclusion of a larger amount of less confident target pseudo detections. Interestingly, we notice that the cutting direction, i.e. vertical or horizontal, impacts the adaptation performance, where a vertical mixing demonstrates a better adaptation performance than the horizontal mixing; we believe that this phenomenon is scenario-dependent. In particular, for datasets concerning autonomous driving scenarios, a vertical mix always includes the road area of both target and source samples, thus being more likely to include objects, while a horizontal mix might only include further scenes where it is less likely to
Table 2. Quantitative results (mAP) for Cityscapes → Foggy Cityscapes benchmark.

<table>
<thead>
<tr>
<th>Method</th>
<th>Detector</th>
<th>Backbone</th>
<th>person</th>
<th>rider</th>
<th>car</th>
<th>truck</th>
<th>bus</th>
<th>train</th>
<th>motorcycle</th>
<th>bicycle</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source only</td>
<td>YOLOv5</td>
<td>CSP-Darknet53</td>
<td>34.8</td>
<td>37.6</td>
<td>48.7</td>
<td>14.3</td>
<td>30.1</td>
<td>8.8</td>
<td>14.6</td>
<td>28.1</td>
<td>27.1</td>
</tr>
<tr>
<td>SC-UDA [49]</td>
<td>Faster R-CNN</td>
<td>VGG-16</td>
<td>38.5</td>
<td>43.7</td>
<td>56.0</td>
<td>27.1</td>
<td>43.8</td>
<td>29.7</td>
<td>31.2</td>
<td>39.5</td>
<td>38.7</td>
</tr>
<tr>
<td>MeGA-CDA [43]</td>
<td>Faster R-CNN</td>
<td>VGG-16</td>
<td>37.7</td>
<td>49.0</td>
<td>52.4</td>
<td>25.4</td>
<td>49.2</td>
<td>46.9</td>
<td>34.5</td>
<td>39.0</td>
<td>41.8</td>
</tr>
<tr>
<td>FL-UDA [27]</td>
<td>Faster R-CNN</td>
<td>VGG-16</td>
<td>30.4</td>
<td>51.9</td>
<td>44.4</td>
<td>34.1</td>
<td>25.7</td>
<td>30.3</td>
<td>37.2</td>
<td>41.8</td>
<td>37.0</td>
</tr>
<tr>
<td>CDN [39]</td>
<td>Faster R-CNN</td>
<td>VGG-16</td>
<td>35.8</td>
<td>45.7</td>
<td>50.9</td>
<td>30.1</td>
<td>42.5</td>
<td>29.8</td>
<td>30.8</td>
<td>36.5</td>
<td>36.6</td>
</tr>
<tr>
<td>SAPN [22]</td>
<td>Faster R-CNN</td>
<td>VGG-16</td>
<td>40.8</td>
<td>46.7</td>
<td>59.8</td>
<td>24.3</td>
<td>46.8</td>
<td>37.5</td>
<td>30.4</td>
<td>40.7</td>
<td>40.9</td>
</tr>
<tr>
<td>CTRP [52]</td>
<td>Faster R-CNN</td>
<td>VGG-16</td>
<td>32.7</td>
<td>44.4</td>
<td>50.1</td>
<td>21.7</td>
<td>45.6</td>
<td>25.4</td>
<td>30.1</td>
<td>36.8</td>
<td>35.9</td>
</tr>
<tr>
<td>MGA [53]</td>
<td>Faster R-CNN</td>
<td>VGG-16</td>
<td>43.9</td>
<td>49.6</td>
<td>60.6</td>
<td>29.6</td>
<td>50.7</td>
<td>39.0</td>
<td>38.3</td>
<td>42.8</td>
<td>44.3</td>
</tr>
<tr>
<td>IRGG [44]</td>
<td>Faster R-CNN</td>
<td>VGG-16</td>
<td>37.4</td>
<td>45.2</td>
<td>51.9</td>
<td>24.4</td>
<td>39.6</td>
<td>25.2</td>
<td>31.5</td>
<td>41.6</td>
<td>37.1</td>
</tr>
<tr>
<td>SSOD [36]</td>
<td>Faster R-CNN</td>
<td>VGG-16</td>
<td>38.8</td>
<td>45.9</td>
<td>57.2</td>
<td>29.9</td>
<td>50.2</td>
<td>51.9</td>
<td>31.9</td>
<td>40.9</td>
<td>43.3</td>
</tr>
<tr>
<td>GIPA [46]</td>
<td>Faster R-CNN</td>
<td>ResNet-50</td>
<td>32.9</td>
<td>46.7</td>
<td>54.1</td>
<td>24.7</td>
<td>45.7</td>
<td>41.1</td>
<td>32.4</td>
<td>38.7</td>
<td>39.5</td>
</tr>
<tr>
<td>CDN [39]</td>
<td>Faster R-CNN</td>
<td>VGG-16</td>
<td>35.8</td>
<td>45.7</td>
<td>50.9</td>
<td>30.1</td>
<td>42.5</td>
<td>29.8</td>
<td>30.8</td>
<td>36.5</td>
<td>36.6</td>
</tr>
<tr>
<td>SAPN [22]</td>
<td>Faster R-CNN</td>
<td>VGG-16</td>
<td>40.8</td>
<td>46.7</td>
<td>59.8</td>
<td>24.3</td>
<td>46.8</td>
<td>37.5</td>
<td>30.4</td>
<td>40.7</td>
<td>40.9</td>
</tr>
<tr>
<td>CTRP [52]</td>
<td>Faster R-CNN</td>
<td>VGG-16</td>
<td>32.7</td>
<td>44.4</td>
<td>50.1</td>
<td>21.7</td>
<td>45.6</td>
<td>25.4</td>
<td>30.1</td>
<td>36.8</td>
<td>35.9</td>
</tr>
<tr>
<td>MGA [53]</td>
<td>Faster R-CNN</td>
<td>VGG-16</td>
<td>43.9</td>
<td>49.6</td>
<td>60.6</td>
<td>29.6</td>
<td>50.7</td>
<td>39.0</td>
<td>38.3</td>
<td>42.8</td>
<td>44.3</td>
</tr>
<tr>
<td>IRGG [44]</td>
<td>Faster R-CNN</td>
<td>ResNet-50</td>
<td>37.4</td>
<td>45.2</td>
<td>51.9</td>
<td>24.4</td>
<td>39.6</td>
<td>25.2</td>
<td>31.5</td>
<td>41.6</td>
<td>37.1</td>
</tr>
<tr>
<td>SSOD [36]</td>
<td>Faster R-CNN</td>
<td>VGG-16</td>
<td>38.8</td>
<td>45.9</td>
<td>57.2</td>
<td>29.9</td>
<td>50.2</td>
<td>51.9</td>
<td>31.9</td>
<td>40.9</td>
<td>43.3</td>
</tr>
<tr>
<td>GIPA [46]</td>
<td>Faster R-CNN</td>
<td>ResNet-50</td>
<td>32.9</td>
<td>46.7</td>
<td>54.1</td>
<td>24.7</td>
<td>45.7</td>
<td>41.1</td>
<td>32.4</td>
<td>38.7</td>
<td>39.5</td>
</tr>
</tbody>
</table>

Table 3. Target detection accuracy with various mixing strategies.

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>CutMix [50]</td>
<td>49.1</td>
</tr>
<tr>
<td>ConfMix</td>
<td>53.6</td>
</tr>
<tr>
<td>ConfMix (Vert)</td>
<td>39.6</td>
</tr>
<tr>
<td>ConfMix (Hor)</td>
<td>41.1</td>
</tr>
<tr>
<td>ConfMix (2-reg)</td>
<td>54.1</td>
</tr>
<tr>
<td>ConfMix (6-div)</td>
<td>53.5</td>
</tr>
<tr>
<td>ConfMix (9-div)</td>
<td>54.7</td>
</tr>
</tbody>
</table>

Figure 3. Qualitative results on Sim10K → Cityscapes scenario of ConfMix models trained with different confidence settings. We visualise true positives in blue bbx, false negatives in red bbx and false positives in orange bbx. By introducing a gradual transition between confidence metrics, we achieve less false positive detections compared to training only using $C_{det}$ and less false negative detections compared to training only using $C_{comb}$.

include objects. Finally, our 4-region division scheme leads to the best adaptation performance, with a mAP improvement of +0.6% and +1.2% compared to 6 and 9 divisions, respectively. This is supported by the intuition that smaller regions produce mixed samples containing a larger portion of the source domain, and increase the probability of occluded target objects, thus limiting the adaptive learning of their complete representation.

Does the progressive pseudo labelling help? We investigate a set of variants of pseudo labelling in order to verify the proposed progressive pseudo labelling strategy. Specifically, we ablate the usage of only $C_{det}$ or $C_{comb}$ for thresholding the pseudo detections. We also investigate different directions for the weight adjustment, i.e. $C_{det} \rightarrow C_{comb}$ and $C_{comb} \rightarrow C_{det}$, as well as different shifting weights with $r$ (in Equation 7) representing a linear decay and $\delta$ (in Equation 6) representing a non-linear decay. Our proposed strategy ($C_{det} \rightarrow C_{comb}$ ($\delta$)) gradually shifts from $C_{det}$ to the stricter $C_{comb}$ using the proposed shifting weight $\delta$.

As shown in Table 4, using only $C_{comb}$ demonstrates to be more advantageous than using only $C_{det}$ for adaptation, and this is mainly due to more reliable pseudo detections. Moreover, we achieve the best result by gradually exploiting from $C_{det}$ to $C_{comb}$ with the non-linear weight $\delta$. A less restrictive confidence metric $C_{det}$ at the early adaptation allows more target pseudo detections and this can help
with the target representation learning, while by gradually shifting to the usage of $C_{comb}$, we improve the reliability of pseudo detection, thus benefiting the detector accuracy.

Figure 3 shows qualitative detection results when using only $C_{det}$, only $C_{comb}$, and our proposed strategy $C_{det} \rightarrow C_{comb} (\delta)$. As can be observed, the model is more likely to predict false positives when using only $C_{det}$, whereas the model using only $C_{comb}$ generates more false negatives. Our proposed gradual transition strategy can better combine the two confidence metrics for adaptive training, achieving the best adaptation performance on the target domain.

<table>
<thead>
<tr>
<th>Confidence</th>
<th>$C_{det}$</th>
<th>$C_{comb}$</th>
<th>$C_{comb} (\delta)$</th>
<th>$C_{det}$</th>
<th>$C_{comb}$</th>
<th>$C_{comb} (\delta)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>mAP</td>
<td>52.7</td>
<td>54.7</td>
<td>55.6</td>
<td>54.3</td>
<td>54.3</td>
<td>56.3</td>
</tr>
</tbody>
</table>

Table 4. Target detection accuracy with various confidence-based pseudo labelling.

Does the weight of consistency loss matter? As pseudo detections are inevitably noisy, we are motivated to weight the consistency loss appropriately in order to avoid the introduction of pseudo detection errors. We therefore investigate how the weight of consistency loss impacts the adaptation performance by using a set of constant weights, in comparison to our dynamic weight $\gamma$. Regarding the constant weight, we vary it from 0.2 to 1 in the step of 0.2. As shown in Table 5, using $\gamma$ as the consistency weight leads to the best mAP performance, with an improvement of $+0.7\%$ compared to the best constant weight of 0.6. Therefore, the use of a dynamic weight whose value varies depending on the pseudo detections confidence, can improve the performance of the model by stabilising the training and mitigating the problem of over-fitting unreliable pseudo detections.

<table>
<thead>
<tr>
<th>Weight</th>
<th>0.2</th>
<th>0.4</th>
<th>0.6</th>
<th>0.8</th>
<th>1</th>
<th>$\gamma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>mAP</td>
<td>53.8</td>
<td>54.7</td>
<td>55.6</td>
<td>54.9</td>
<td>47.7</td>
<td>56.3</td>
</tr>
</tbody>
</table>

Table 5. Target detection accuracy with different weights on the consistency loss.

Does the number of pseudo detections before mixing matter? We analyse the number of pseudo detections before mixing by retaining with only 25%, 50%, 75% and 100% (i.e. our setting) of the most reliable pseudo detections in $\tilde{y}_T$, with the confidence threshold $C_{th}$ fixed for filtering the detections. As shown in Table 6, decreasing the number of pseudo detections consistently leads to a worse result than using all of them, i.e. our setting. Thanks to our progressive pseudo labelling scheme, most of the false positives could be filtered out already, thus detections with a confidence greater than $C_{th}$ are generally useful for the model to learn the target features.

<table>
<thead>
<tr>
<th>Pseudo detections (%)</th>
<th>25%</th>
<th>50%</th>
<th>75%</th>
<th>100%</th>
</tr>
</thead>
<tbody>
<tr>
<td>mAP</td>
<td>45.1</td>
<td>48.9</td>
<td>51.7</td>
<td>56.3</td>
</tr>
</tbody>
</table>

Table 6. Target detection accuracy at varying number of pseudo detections.

Does sample mixing performs better than simple self-training? We compare ConfMix with a simple baseline that consists in applying naive finetuning with pseudo detections. With the model trained on the source domain, we generate pseudo detections on the target dataset using the same confidence threshold $C_{th}$ used in ConfMix to filter the boxes on top of non-maximum suppression. We then expand the training dataset containing both source and target samples for further training. With such self-training, we achieve a mAP of 30.5 (Cityscapes→Foggy Cityscapes), 55.4 (Sim10K→Cityscapes) and 46.4 (KITTI→Cityscapes), which are much inferior than ConfMix as reported in Table 1 and 2, confirming the effectiveness of our proposal.

5. Conclusion

We proposed ConfMix, a novel confidence-based mixing method for adapting object detectors trained on a source domain to a target domain in an unsupervised manner. We introduced a region-level strategy for sample data augmentation by mixing the region of the target image with the most confident pseudo detections with the source image, and achieved adaptation with a consistency loss on the pseudo detections. We also introduced the progressive pseudo labelling scheme by gradually restricting the confidence metric in order to facilitate the smooth transition from learning the target representation to improving detection accuracy. We compared our approach with state-of-the-art methods, demonstrating its superior performance on two benchmarks. As future work, we will apply our method in other practical scenarios, other than autonomous driving, and improve its compatibility with different object detection frameworks.
References


[52] Ganlong Zhao, Guanbin Li, Ruijia Xu, and Liang Lin. Collaborative training between region proposal localization and...

