Supplementary material for Detect, Augment, Compose, and Adapt: Four Steps for Unsupervised Domain Adaptation in Object Detection

1 Introduction

In this document, we provide more insights and carry out additional experiments in support of the main paper's content. In particular:

- We detail a step-by-step algorithm of how DACA works.
- We assess the impact of the pseudo-label selection confidence threshold.
- We analyze the effect of different initialization schemes of the detector's weights.

2 Algorithm

Algorithm 1 details the key steps of DACA.

3 Additional ablations

3.1 Pseudo-label selection

Pseudo-label selection is a key step in DACA. Intuitively, increasing the detector's confidence threshold imposes a strict pseudo-label selection criteria where only highly confident labels are selected and subsequently exploited for the adaptation routine. Conversely, decreasing the detector's confidence implies a loose selection mechanism where less confident pseudo-labels are leveraged. Therefore, it is necessary to evidence if high selection confidence is a better option than a low confidence.

For fairness of comparison, in the default experiments we kept the selection confidence set to 0.25 similarly to ConfMix [1]. In this experiment, we further evaluate DACA by using

Algorithm 1: Pseudocode of DACA.

Input: Φ_Θ: detector; $(\mathbf{X}_S, \mathbf{G}_S)$: source image and its ground truth; $\mathbf{X}_T$: target image; g: image transformation function; ℓ: loss function; S_{row}, S_{col}: number of rows and columns for grid division, respectively; N_{it}: number of adaptation iterations.
Output: Φ_Θ adapted via DACA.

for $i = 1, \ldots, N_{\text{it}}$ **do**:
  Compute $\mathbf{D}_S \leftarrow \Phi_\Theta(\mathbf{X}_S)$.
  Compute $\ell_S \leftarrow \ell(\mathbf{G}_S, \mathbf{D}_S)$.
  Compute $\mathbf{P}_T \leftarrow \Phi_\Theta(\mathbf{X}_T)$.
  Divide $\mathbf{X}_T$ into $S_{\text{row}} \times S_{\text{col}}$ regions.
  Associate each detection in $\mathbf{P}_T$ to a region based on the center of its bounding box.
  Compute the average detection confidence of each region.
  Select the region with the highest average confidence and use its detections as pseudo-labels $(\tilde{\mathbf{X}}_T, \tilde{\mathbf{P}}_T)$.
  Generate $S_{\text{row}} \times S_{\text{col}}$ different (random) augmentations of the selected region and its pseudo-labels.
  Merge the augmentation regions to obtain a composite image $\hat{\mathbf{X}}_T$ along with its pseudo-labels $\hat{\mathbf{P}}_T$.
  Compute $\mathbf{D}_T \leftarrow \Phi_\Theta(\hat{\mathbf{X}}_T)$.
  Compute $\ell_T \leftarrow \ell(\hat{\mathbf{P}}_T, \mathbf{D}_T)$.
  Compute $\ell \leftarrow \ell_S + \ell_T$.
  Minimize ℓ to find the optimal parameters Θ for Φ_Θ.
end for
return Adapted Φ_Θ.

0.1, 0.5 and 0.8 as confidences. We also include Precision and Recall metrics to investigate the trend of False Positives and True Positives, respectively [2].

From Tab. 1, we can notice that that using a selection threshold of 0.25 scores the highest mAP on average (51.4%). Decreasing the confidence threshold to 0.1 causes a 2.7% decline, which is owed to the sharp decline in terms of Precision (-7.4%). This is a reasonable behavior provided that a low threshold entails more detections, which results in a higher rate of False Positives while favoring also True positive detections (i.e. $+1.5\%$ in terms of Recall). By contrast, increasing the confidence threshold from 0.25 to 0.5 incurs fewer detections, which is manifested in a 8.8% drop in terms of Recall, but also favors fewer False Positives ($+2\%$ precision). When the selection threshold is further increased to 0.8, by far fewer detections are obtained (30.6% Recall), which also affect significantly the Precision (59.2%). In conclusion, a threshold of 0.25 seems to be the best trade-off. Figs. 1, 2, and 3 depict comparison instances between the four selection threshold options from the three adaptation benchmarks. In particular, it can be observed that for a 0.1 confidence threshold, more False Positive detections are obtained. However, when the confidence threshold is increased to 0.5, the False Positives are reduced but often at the cost of missed True positives.

3.2 Weight initialization

DACA leverages self-training to perform UDA. Prior to adaptation, the detector first infers pseudo-labels that would serve as self-supervision to learn the target modality. Yet, he quality

	C → F			K → C			S → C			Average		
Conf.	P	R	mAP	P	R	mAP	P	R	mAP	P	R	mAP
0.1	59.9	36.7	38.8	71.5	41.5	50.6	74.1	52.4	56.6	68.5	43.5	48.7
0.25	65.2	35.6	39.4	81.0	38.4	54.2	81.5	52.0	60.6	75.9	42.0	51.4
0.5	66.4	31.1	36.6	84.2	33.6	53.0	83.2	34.8	49.6	77.9	33.2	46.4
0.8	55.7	21.4	26.3	43.2	39.7	47.6	78.7	30.8	47.1	59.2	30.6	40.3

Table 1: Effect of pseudo-label selection confidence threshold on the performance for all the adaptation benchmarks. Keys. Conf: Confidence threshold for pseudo-label selection, P: Precision, R: Recall, mAP: Mean Average Precision, C: Cityscapes, F: FoggyCityscapes, S: Sim10K, and K: KITTI.

Figure 1: Qualitative instances of DACA with different pseudo-label selection confidence thresholds for the K → C benchmark.

of the pseudo-labels also depends on the initialization of the detector's weights. As in [1], in the default experiments we train the detector on the source dataset for 20 epochs starting with the COCO pre-trained model, and then we perform adaptation for 50 more epochs.

In this experiment we compare three other initialization alternatives: (i) We discard the pre-training phase on the source and perform adaptation from scratch (i.e. no COCO pre-trained model); (ii) We train the detector on the source for 20 epochs starting from scratch, and then do adaptation as in the default setting. We also report the baseline (i.e. the detector is trained on the source dataset from scratch using its ground truth) and oracle (i.e. the detector is trained on the target dataset from scratch using its ground truth) performances in this setting; (iii) We perform adaptation starting with COCO pre-trained detector weights.

As shown Tab. 2, training the model from scratch on the source and then adapting it entails a significant mAP decay compared to training the model on the source starting with COCO pre-trained weights. In particular, a 12.4% decline is observed on average over the three adaptation benchmarks. Both the lower and upper bound (i.e. baseline and oracle) scores have dropped drastically when training from scratch.

When adapting the model from scratch without pre-training on the source dataset, we get the lowest mAP of 28.6% on average over the three adaptation scenarios. However, when adapting the model starting with COCO pre-trained weights, we obtain a mAP of 49% on average, which is significantly higher, indicating again that pre-training plays a pivotal role in UDA.

Another worth-mentioning fact is that performing adaptation without pre-training on the source dataset starting with COCO weights outperforms, by far (+10% mAP), the case when the model is first trained on the source from scratch and then adapted. This is rationale since COCO dataset is larger and carries much more semantic information, which qualifies it better

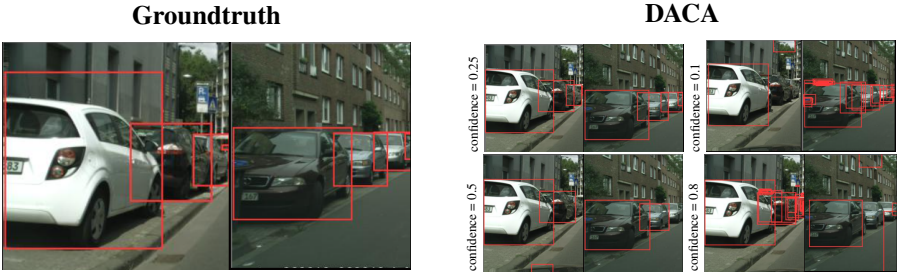

Figure 2: Qualitative instances of DACA with different pseudo-label selection confidence thresholds for the S → C benchmark.

Figure 3: Qualitative instances of DACA with different pseudo-label selection confidence thresholds for the C → F benchmark.

for transfer learning tasks. The default case of training on the source dataset starting with COCO weights and then adapting is 2.4% higher in terms of mAP across the three adaptation scenarios and 5.9% higher on the C→F scenario, which explains that source data is also necessary when addressing UDA.

Configuration	Pre-training	C→F	K→C	S→C	Avg.
Baseline (Source only)	None	7.9	27.5	31.5	22.3
DACA (Adaptation on source and target)	None	17.6	36.4	31.9	28.6
DACA (Finetuning on source + Adaptation on source and target)	None	22.8	50.0	44.3	39.0
Oracle (Target only)	None	24.8	57.7	57.7	46.7
Baseline (Source only)	COCO	29.7	42.9	50.4	41.0
DACA (Adaptation on source and target)	COCO	33.5	53.7	60.0	49.0
DACA (Finetuning on source + Adaptation on source and target)	COCO	39.4	54.2	60.6	51.4
Oracle (Target only)	COCO	42.7	69.5	69.5	60.6

Table 2: Effect of different weights initialization schemes on the detection performance (mAP) for the three adaptation benchmarks. Keys: C: Cityscapes, F: FoggyCityscapes, S: Sim10K, K: KITTI, Avg.: average across the three adaptation scenarios.

References

[1] G. Mattolin, L. Zanella, E. Ricci, and Y. Wang. ConfMix: Unsupervised Domain Adaptation for Object Detection via Confidence-based Mixing. In *WACV*, 2023.

[2] R. Padilla, S.L. Netto, and E.A.B. Da Silva. A survey on performance metrics for object-detection algorithms. In *IWSSIP*, 2020.