# Supplementary Materials for Unsupervised Domain Adaptation for One-Stage Object Detector using Offsets to Bounding Box

## A Binning Strategies

### A.1 The ratio of features assigned to each bin

In Sec. 3.4 of the main paper, we set $m_i$ of each bin to $(m_1, m_2, m_3) = (lv-\frac{1}{2}, lv+\frac{1}{2}, lv+\frac{3}{2})$ for each $lv$-th level feature to convert the offset value into a categorical probability vector. Fig. 1 shows the average of the probabilities that the left $l$ and top $t$ offset values belong to each bin as the training iteration progresses in the CS $\rightarrow$ FoggyCS setting at different feature level. Initially, the probability of the offset value belonging to each bin begins as a uniform distribution so that $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, and it gradually converges to the average of $\tilde{q}$, the probability vector of the offset values, during the warm-up period ($I$ iterations). Since $m_i$ is not finely set to make the the ratio of the offset values corresponding to each bin equal, in the case of $F_3$ and $F_4$, the ratio belonging to the last bin is somewhat high. However, setting the value of $m_i$ as mentioned in the main paper is sufficiently effective. This is because it is not important to ensure that the proportion of features belonging to each bin is equal, but important to similarly match the distribution between the features of the two domains belonging to the same bin.

### A.2 Setting of the $m_i$ values according to the number of bins

Table 3 of Sec. 4.5 in the main paper compares the effect of conditioning the feature more strongly or loosely by changing the number of bins. Table 1 specifies $m_i$ values for each $N_{bin}$ setting at different feature level. We set $m_i$ values so that more than a certain ratio of features can be assigned to each bin.

## B Training Details for the Self-Training

We basically follow the details of the method proposed in Liu et. al [2]. Even though [2] is a work that tackles semi-supervised object detection, its proposed method is applicable to unsupervised domain adaptation as well since the main idea is about how to handle the unlabeled data. It suggests how to train the object detector with the unlabeled data in a self-training manner using the unbiased teacher network. Only difference in the setting of ours from [2] is that in [2], the labeled and the unlabeled data are from the same distribution, but in our case of unsupervised domain adaptation, the unlabeled data are from different distribution of the labeled data. The idea of [2] is to utilize the unbiased teacher

Table 1: $m_i$ values set according to $N_{bin}$ and different feature level.

| | $N_{bin} = 2$ | $N_{bin} = 3$ | $N_{bin} = 4$ | $N_{bin} = 5$ |
|---|---|---|---|---|
| Feature Level | $(m_1, m_2)$ | $(m_1, m_2, m_3)$ | $(m_1, m_2, m_3, m_4)$ | $(m_1, m_2, m_3, m_4, m_5)$ |
| $F_3$ | (3.5, 4.5) | (2.5, 3.5, 4.5) | (2.5, 3.5, 4.5, 5.5) | (2.5, 3.5, 4.5, 5.5, 6.5) |
| $F_4$ | (4.5, 5.5) | (3.5, 4.5, 5.5) | (3.5, 4.5, 5.5, 6.5) | (3.5, 4.5, 5.5, 6.5, 7.5) |
| $F_5$ | (5.5, 6.5) | (4.5, 5.5, 6.5) | (4.5, 5.5, 6.5, 7.5) | (4.5, 5.5, 6.5, 7.5, 8.5) |
| $F_6$ | (6.5, 7.5) | (5.5, 6.5, 7.5) | (4.5, 5.5, 6.5, 7.5) | (4.5, 5.5, 6.5, 7.5, 8.5) |
| $F_7$ | (7.5, 8.5) | (6.5, 7.5, 8.5) | (5.5, 6.5, 7.5, 8.5) | (5.5, 6.5, 7.5, 8.5, 9.5) |

network to produce the pseudo labels for the unlabeled data. The teacher object detector has the same architecture as the student but its parameters are not optimized by gradient descent but are updated by exponential moving average of the student network, as suggested in Tarvainen et. al [3]. The parameters of the teacher detector are updated as follows:

$$\theta_t = \alpha\theta_t + (1 - \alpha)\theta_s \tag{1}$$

where $\theta_t$ and $\theta_s$ represent the parameters of the teacher and the student network, respectively. $\alpha$ is the EMA rate which decides the percentage of the parameters of teacher network in the previous time step to be applied to the updated teacher parameters. The larger the $\alpha$, the slower the teacher network progresses. We empirically find that $\alpha = 0.9999$ works the best in our setting of unsupervised domain adaptation. Also, the EMA interval indicates the number of iterations between EMA updates. EMA interval is set differently for each benchmark, for CS $\rightarrow$ FoggyCS, it is set to 10, while for KITTI $\rightarrow$ CS and Sim10k $\rightarrow$ CS, it is set to 5 and 1. The smaller the value, the more frequent the EMA updates are.

In [2], the predictions of the teacher network are used as the pseudo labels of the target domain to train the student network. [2] feeds weakly augmented unlabeled data into the teacher network and strongly augmented unlabeled data are fed into the student, by differentiating inputs to the two networks, resulting in knowledge gaps between the predictions of the teacher and the student networks. The student tries to narrow this gap by training unlabeled data with pseudo labels generated by the teacher network. However, in our case, we find that applying weak augmentation to the inputs of the teacher network unnecessary and using the original target inputs is effective. For example, when given a target image, $x^T$, the teacher object detector, $f_t$, predicts a set of bounding boxes.

$$\mathbb{B}_t^T = (\hat{y}_i^T, \hat{b}_i^T)_{i=1}^k = f_t(x^T) \tag{2}$$

$\hat{y}_i \in \mathbb{R}^C$ where $0 \leq \hat{y}_{i,c} \leq 1$ and $\hat{b}_i \in \mathbb{R}^4$ indicates the classification confidence and the predicted box coordinates $(l, t, r, b)$, respectively. $k$ is the number of bounding boxes predicted for a given target domain image, $x^T$, in the teacher detector . Note that $T$ in the superscript refers to the 'target domain' and the $t$ in the subscript refers to the 'teacher detector'. Then we threshold the predicted boxes from the teacher detector using the confidence score. Specifically, we set

a threshold $\delta$ and eliminate bounding boxes with the confidence score less then or equal to $\delta$. Therefore, $\mathbb{B}'^{T}_{t} = \{(\hat{y}^{T}_{i}, \hat{b}^{T}_{i}) | \max_{c}(\hat{y}_{i,c}) > \delta\} \subset \mathbb{B}^{T}_{t}$. $\delta$ is set as 0.5 empirically in all of our experiments. Finally, we use this thresholded bounding boxes from the teacher network as the pseudo labels of the target domain to train the student detector. As mentioned earlier, strongly augmented inputs are fed into the student detector, where the same strong augmentation strategy as described in [2] is used. The overall loss function to train the student network, $f_s$, for both the source and the target domain is as follows:

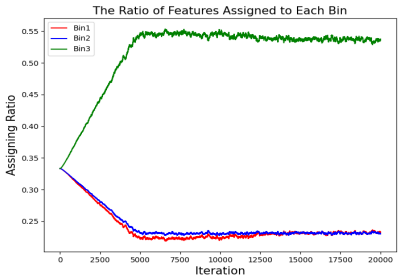
$$\mathcal{L}_{student} = \mathcal{L}_{det}(x^S, (y^S, b^S)) + \lambda_{self}\mathcal{L}_{det}(\mathcal{A}(x^T), \mathbb{B}'^{T}_{t}) \tag{3}$$

where $\mathcal{A}$ refers to the strong augmentation and $\lambda_{self}$ is the weight on the self-training loss for the target domain. Here, we use $\lambda_{self} = 2$ since it shows the best results. While $f_s$ is trained by above loss function, on the other hand, the teacher network is updated via EMA of (1) as explained earlier.

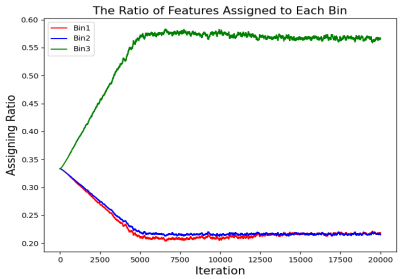
In all of our self-training experiments, the student and the teacher detectors are initialized by the detector that is pre-trained with our proposed OADA (Offset-Left & Top), so that the $f_t$ is able to produce reasonably correct pseudo labels from the beginning of the training. We report the performance of the teacher detector since it shows highly improved performance due to the temporal ensemble effect. We only applied the proposed method of [2] to our problem, so there is no contribution in terms of novelty, but we consider this experiment as an important study result because we applied the proposed method to an anchor-free one-stage detector, FCOS for the first time. Considering that the method was originally proposed for Faster R-CNN, a two-stage detector, this experimental results further prove the applicability of the method to other detector architectures and the unsupervised domain adaptation setting.

## C Qualitative Results

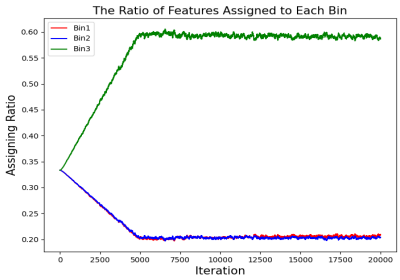
Fig.2, 3, and 4 are the qualitative results of the *SourceOnly*, *EPM* [1] and *OADA* methods in CS $\rightarrow$ FoggyCS, Sim10k $\rightarrow$ CS, and KITTI $\rightarrow$ CS benchmarksets, respectively. As shown in the figure, *OADA* is able to detect distant objects in the center better than baseline methods.



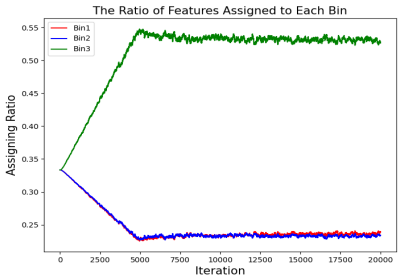

(a) Based on the left offsets in $F_3$



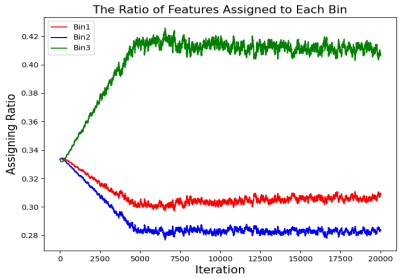

(b) Based on the top offsets in $F_3$



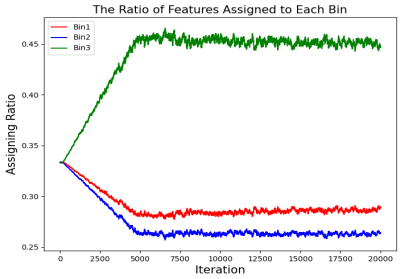

(c) Based on the left offsets in $F_4$



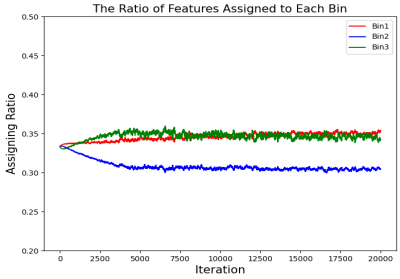

(d) Based on the top offsets in $F_4$



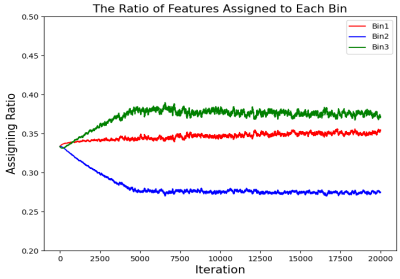

(e) Based on the left offsets in $F_5$




(f) Based on the top offsets in $F_5$




(g) Based on the left offsets in $F_6$




(h) Based on the top offsets in $F_6$

Fig. 1: The ratio of feature allocated to each bin at different feature level as the iteration progresses during training.


Fig. 2: Qualitative results of *SourceOnly*, *EPM* [1] and *OADA* in CS → FoggyCS.


Fig. 3: Qualitative results of *SourceOnly*, *EPM* [1] and *OADA* in Sim10k → CS.


Fig. 4: Qualitative results of *SourceOnly*, *EPM* [1] and *OADA* in KITTI → CS.

## References

1. 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 (ECCV). Springer (2020)
2. Liu, Y.C., Ma, C.Y., He, Z., Kuo, C.W., Chen, K., Zhang, P., Wu, B., Kira, Z., Vajda, P.: Unbiased teacher for semi-supervised object detection. In: Proceedings of the International Conference on Learning Representations (ICLR) (2021)
3. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: Advances in Neural Information Processing Systems (NeurIPS) (2017)