SCL: Towards Accurate Domain Adaptive Object Detection via Gradient Detach Based Stacked Complementary Losses

Anonymous authors
Paper under double-blind review

Abstract

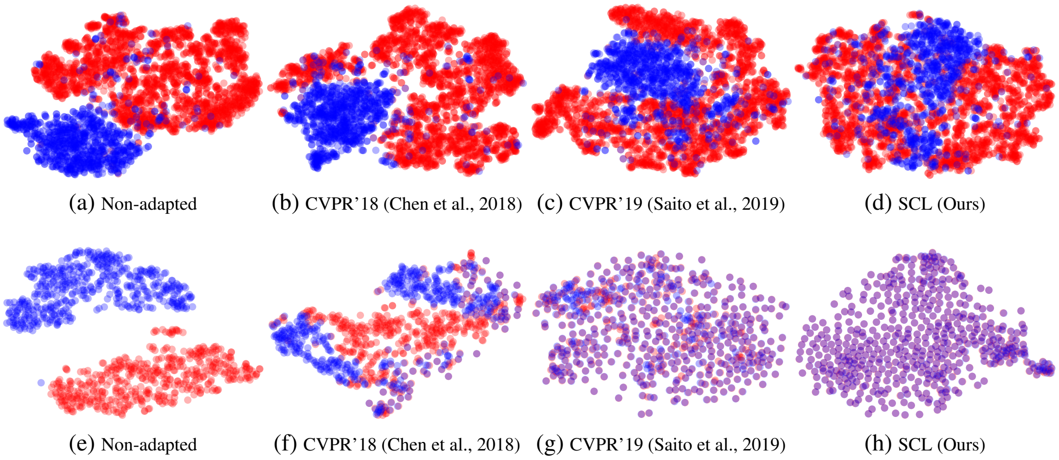
Unsupervised domain adaptive object detection aims to learn a robust detector in the domain shift circumstance, where the training (source) domain is label-rich with bounding box annotations, while the testing (target) domain is label-agnostic and the feature distributions between training and testing domains are dissimilar or even totally different. In this paper, we propose a gradient detach based stacked complementary losses (SCL) method that uses detection objective (Ren et al., 2015) (cross entropy and smooth l_1 regression) as the primary objective, and cuts in several auxiliary losses in different network stages to utilize information from the complement data (target images) that can be effective in adapting model parameters to both source and target domains. A gradient detach operation is applied between detection and context sub-networks with different objectives during training to force networks to learn discriminative representations. We argue that the conventional training with primary objective mainly leverages the information from the source-domain for maximizing likelihood and ignores the complement data in shallow layers of networks, which leads to an insufficient integration within different domains. Thus, our proposed method is a more syncretic adaptation learning process. We conduct comprehensive experiments on seven datasets, the results demonstrate that our method performs favorably better than the state-of-the-art methods by a large margin. For instance, from Cityscapes to FoggyCityscapes, we achieve 37.9% mAP, outperforming the previous art *Strong-Weak* (Saito et al., 2019) by 3.6%[1].

1 Introduction

In real world scenarios, generic object detection always faces severe challenges from variations in viewpoint, background, object appearance, illumination, occlusion conditions, scene change, etc. These unavoidable factors make object detection in domain-shift circumstance becoming a challenging and new rising research topic in the recent years. Also, domain change is a widely-recognized, intractable problem that urgently needs to break through in reality of detection tasks, like video surveillance, autonomous driving, etc. (see Figure 2).

Revisiting Domain-Shift Object Detection. Common approaches for tackling domain-shift object detection are mainly in two directions: (i) training supervised model then fine-tuning on the target domain; or (ii) unsupervised cross-domain representation learning. The former requires additional instance-level annotations on target data, which is fairly laborious, expensive and time-consuming. So most approaches focus on the latter one but still have some challenges. The first challenge is that the representations of source and target domain data should be embedded into a common space for matching the object, such as the hidden feature space (Saito et al., 2019; Chen et al., 2018), input space (Tzeng et al., 2018; Cai et al., 2019) or both of them (Kim et al., 2019b). The second is that a feature alignment/matching operation or mechanism for source/target domains should be further defined, such as subspace alignment (Raj et al., 2015), $\mathcal{H}$-divergence and adversarial learning (Chen et al., 2018), MRL (Kim et al., 2019b), Strong-Weak alignment (Saito et al., 2019), etc. In general, our SCL is also a learning-based alignment method across domains with an end-to-end framework.

[1] Code and models will be publicly available.


(a) Non-adapted (b) CVPR'18 (Chen et al., 2018) (c) CVPR'19 (Saito et al., 2019) (d) SCL (Ours)

(e) Non-adapted (f) CVPR'18 (Chen et al., 2018) (g) CVPR'19 (Saito et al., 2019) (h) SCL (Ours)

Figure 1: Visualization of features from PASCAL to Clipart (first row) and from Cityscapes to FoggyCityscapes (second row) by t-SNE (Maaten & Hinton, 2008). Red indicates the source examples and blue is the target one. If source and target features locate in the same position, it is shown as light blue. All models are re-trained with a unified setting to ensure fair comparisons. It can be observed that our feature embedding results are consistently much better than previous approaches on either dissimilar domains (PASCAL and Clipart) or similar domains (Cityscapes and FoggyCityscapes).

Our Key Ideas. The goal of this paper is to introduce a simple design that is specific to convolutional neural network optimization and improves its training on tasks that adapt on discrepant domains. Unsupervised domain adaptation for recognition has been widely studied by a large body of previous literature (Ganin et al., 2016; Long et al., 2016; Tzeng et al., 2017; Panareda Busto & Gall, 2017; Hoffman et al., 2018; Murez et al., 2018; Zhao et al., 2019; Wu et al., 2019), our method more or less draws merits from them, like aligning source and target distributions with adversarial learning (domain-invariant alignment). However, object detection is a technically different problem from classification, since we would like to focus more on the object of interests (local regions).

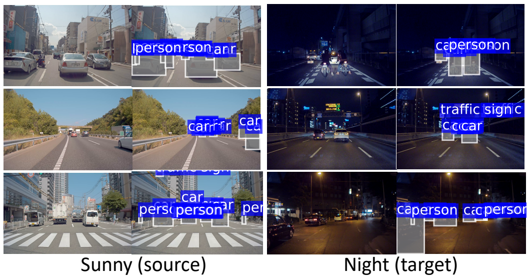


Figure 2: Illustration of domain-shift object detection in autonomous driving scenario. Images are from INIT dataset (Shen et al., 2019).

Some recent work (Zhu et al., 2019) has proposed to conduct alignment only on local regions so that to improve the efficiency of model learning. While this operation may cause a deficiency of critical information from context. Inspired by multi-feature/strong-weak alignment (Saito et al., 2019; Zhang et al., 2018; He & Zhang, 2019) which proposed to align corresponding local-region on shallow layers with small respective field (RF) and align image-level features on deep layers with large RF, we extend this idea by studying the stacked complementary objectives and their potential combinations for domain adaptive circumstance. We observe that domain adaptive object detection is supported dramatically by the deep supervision, however, the diverse supervisions should be applied in a controlled manner, including the cut-in locations, loss types, orders, updating strategy, etc., which is one of the contributions of this paper. Furthermore, our experiments show that even with the existing objectives, after elaborating the different combinations and training strategy, our method can obtain competitive results. By pluging-in a new sub-network that learns the context features independently with gradient detach updating strategy in a hierarchical manner, we obtain the best results on several domain adaptive object detection benchmarks.

The Relation to Complement Objective Training (Chen et al., 2019) and Deep Supervision (Lee et al., 2015). COL (Chen et al., 2019) proposed to involve additional function that complements the primary objective, and updated the parameters alternately with primary and complement objectives.

Specifically, cross entropy is used as the primary objective $\mathbf{H_p}$:

$$\mathbf{H_p}(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{N} \sum_{i=1}^{N} \mathbf{y}_i^T \cdot \log(\hat{\mathbf{y}}_i) \tag{1}$$

where $\mathbf{y}_i \in \{0,1\}^D$ is the label of the i-th sample in one-hot representation and $\hat{\mathbf{y}}_i \in [0,1]^D$ is the predicted probabilities.

Th complement entropy $\mathbf{H_c}$ is defined in COT (Chen et al., 2019) as the average of sample-wise entropies over complement classes in a mini-batch:

$$\mathbf{H_c}(\hat{\mathbf{y}}_{\overline{c}}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{H}(\hat{\mathbf{y}}_{i\overline{c}}) \tag{2}$$

where $\mathcal{H}$ is the entropy function. $\hat{\mathbf{y}}_{\overline{c}}$ is the predicted probabilities of complement classes $\overline{c}$. The training process is that: for each iteration of training, 1) update parameters by $\mathbf{H_p}$ first; then 2) update parameters by $\mathbf{H_c}$. In contrast, we don't use the alternate strategy but update the parameters simultaneously using gradient detach strategy with primary and complement objectives. Since we aim to let the network enable to adapt on both source and target domain data and meanwhile enabling to distinguish objects from them, thus our complement objective design is quite different from COT. We will describe with details in Section 2.

In essence, our method is more likely to be the deeply supervised formulation (Lee et al., 2015) that backpropagation of error now proceeds not only from the final layer but also simultaneously from our intermediate complementary outputs. While DSN is basically proposed to alleviate "vanishing" gradient problem, here we focus on how to adopt these auxiliary losses to promote to mix two different domains through domain classifiers for detection. Interestingly, we observe that diverse objectives can lead to better generalization for network adaptation. Motivated by this, we propose **S**tacked **C**omplementary **L**osses (SCL), a simple yet effective approach for domain-shift object detection. Our SCL is fairly easy and straight-forward to implement, but can achieve remarkable performance. We conjecture that previous approaches that focus on conducting domain alignment on high-level layers only (Chen et al., 2018) cannot fully adapt shallow layer parameters to both source and target domains (even local alignment is applied (Saito et al., 2019)) which restricts the ability of model learning. Also, gradient detach is a critical part of learning with our complementary losses. We further visualize the features obtained by non-adapted model, DA (Chen et al., 2018), Strong-Weak (Saito et al., 2019) and ours, features are from the last layer of backbone before feeding into the Region Proposal Network (RPN). As shown in Figure 1, it is obvious that the target features obtained by our model are more compactly matched with the source domain than any other models.

Contributions. Our contributions in this paper are three-fold.

- We propose an end-to-end learnable framework that adopts complementary losses for domain adaptive object detection. We study the deep supervisions in this task with a controlled manner. Our method allows information from source and target domains to be integrated seamlessly.
- We propose a gradient detach learning strategy to enable complementary losses to learn a better representation and boost the performance. We also provide extensive ablation studies to empirically verify the effectiveness of each component in our framework design.
- To the best of our knowledge, this is a pioneer work to investigate the influence of diverse loss functions and gradient detach for domain adaptive object detection. Thus, this work gives very good intuition and practical guidance with multi-objective learning for domain adaptive object detection. More remarkably, our method achieves the highest accuracy on several domain adaptive or cross-domain object detection benchmarks, which are new records on this task.

2 METHODOLOGY

Following the common formulation of domain adaptive object detection, we define a *source domain* $\mathcal{S}$ where annotated bound-box is available, and a *target domain* $\mathcal{T}$ where only the image can be used in training process without any labels. Our purpose is to train a robust detector that can adapt well to both source and target domain data, i.e., we aim to learn a *domain-invariant* feature representation that works well for detection across two different domains.

2.1 MULTI-COMPLEMENT OBJECTIVE LEARNING

As shown in Figure 3, we focus on the complement objective learning and let $\mathcal{S} = \{(\mathbf{x}_i^{(s)}, \mathbf{y}_i^{(s)})\}$ where $\mathbf{x}_i^{(s)} \in \mathcal{R}^n$ denotes an image, $\mathbf{y}_i^{(s)}$ is the corresponding bounding box and category labels for sample $\mathbf{x}_i^{(s)}$, and i is an index. Each label $\mathbf{y}^{(s)} = (y_{\mathbf{c}}^{(s)}, y_{\mathbf{b}}^{(s)})$ denotes a class label $y_{\mathbf{c}}^{(s)}$ where $\mathbf{c}$ is the category, and a 4-dimension bounding-box coordinate $y_{\mathbf{b}}^{(s)} \in \mathcal{R}^4$. For the target domain we only use image data for training, so $\mathcal{T} = \{\mathbf{x}_i^{(t)}\}$. We define a recursive function for layers $\mathbf{k} = 1, 2, \ldots, \mathbf{K}$ where we cut in complementary losses:

$$\hat{\Theta}_{\mathbf{k}} = \mathcal{F}\left(\mathbf{Z}_{\mathbf{k}}\right), \text{ and } \mathbf{Z}_0 \equiv \mathbf{x} \tag{3}$$

where $\hat{\Theta}_{\mathbf{k}}$ is the feature map produced at layer $\mathbf{k}$, $\mathcal{F}$ is the function to generate features at layer $\mathbf{k}$ and $\mathbf{Z}_{\mathbf{k}}$ is input at layer $\mathbf{k}$. We formulate the complement loss of domain classifier $\mathbf{k}$ as follows:

$$\begin{aligned} \mathcal{L}_{\mathbf{k}}\left(\hat{\Theta}_{\mathbf{k}}^{(s)}, \hat{\Theta}_{\mathbf{k}}^{(t)}; \mathbf{D}_{\mathbf{k}}\right) &= \mathcal{L}_{\mathbf{k}}^{(s)}(\hat{\Theta}_{\mathbf{k}}^{(s)}; \mathbf{D}_{\mathbf{k}}) + \mathcal{L}_{\mathbf{k}}^{(t)}(\hat{\Theta}_{\mathbf{k}}^{(t)}; \mathbf{D}_{\mathbf{k}}) \\ &= \mathbb{E}\left[\log\left(\mathbf{D}_{\mathbf{k}}\left(\hat{\Theta}_{\mathbf{k}}^{(s)}\right)\right)\right] + \mathbb{E}\left[\log\left(1 - \mathbf{D}_{\mathbf{k}}\left(\hat{\Theta}_{\mathbf{k}}^{(t)}\right)\right)\right] \end{aligned} \tag{4}$$

where $\mathbf{D}_{\mathbf{k}}$ is the $\mathbf{k}$-th domain classifier or discriminator. $\hat{\Theta}_{\mathbf{k}}^{(s)}$ and $\hat{\Theta}_{\mathbf{k}}^{(t)}$ denote feature maps from source and target domains respectively. Following (Chen et al., 2018; Saito et al., 2019), we also adopt gradient reverse layer (GRL) (Ganin & Lempitsky, 2015) to enable adversarial training where a GRL layer is placed between the domain classifier and the detection backbone network. During backpropagation, GRL will reverse the gradient that passes through from domain classifier to detection network.

For our instance-context alignment loss $\mathcal{L}_{\mathbf{ILoss}}$, we take the instance-level representation and context vector as inputs. The instance-level vectors are from RoI layer that each vector focuses on the representation of local object only. The context vector is from our proposed sub-network that combine hierarchical global features. We concatenate instance features with same context vector. Since context information is fairly different from objects, joint training detection and context networks will mix the critical information from each part, here we proposed a better solution that uses detach strategy to update the gradients. We will introduce it with details in the next section. Aligning instance and context representation simultaneously can help to alleviate the variances of object appearance, part deformation, object size, etc. in instance vector and illumination, scene, etc. in context vector. We define d_i as the domain label of i-th training image where $d_i = 1$ for the source and $d_i = 0$ for the target, so the instance-context alignment loss can be further formulated as:

$$\mathcal{L}_{\mathbf{ILoss}} = -\frac{1}{N_s}\sum_{i=1}^{N_s}\sum_{i,j}(1-d_i)\log \mathbf{P}_{(i,j)} - \frac{1}{N_t}\sum_{i=1}^{N_t}\sum_{i,j} d_i \log\left(1 - \mathbf{P}_{(i,j)}\right) \tag{5}$$

where N_s and N_t denote the numbers of source and target examples. $\mathbf{P}_{(i,j)}$ is the output probabilities of the instance-context domain classifier for the j-th region proposal in the i-th image. So our total **SCL** objective $\mathcal{L}_{\mathbf{SCL}}$ can be written as:

$$\mathcal{L}_{\mathbf{SCL}} = \sum_{\mathbf{k}=1}^{\mathbf{K}} \mathcal{L}_{\mathbf{k}} + \mathcal{L}_{\mathbf{ILoss}} \tag{6}$$

2.2 GRADIENTS DETACH UPDATING

In this section, we introduce a simple detach strategy which prevents the flow of gradients from context sub-network through the detection backbone path. We find this can help to obtain more discriminative context and we show empirical evidence (see Figure 6) that this path carries information with diversity and hence gradients from this path getting suppressed is superior for such task.

As aforementioned, we define a sub-network to generate the context information from early layers of detection backbone. Intuitively, instance and context will focus on perceptually different parts of an image, so the representations from either of them should also be discrepant. However, if we train

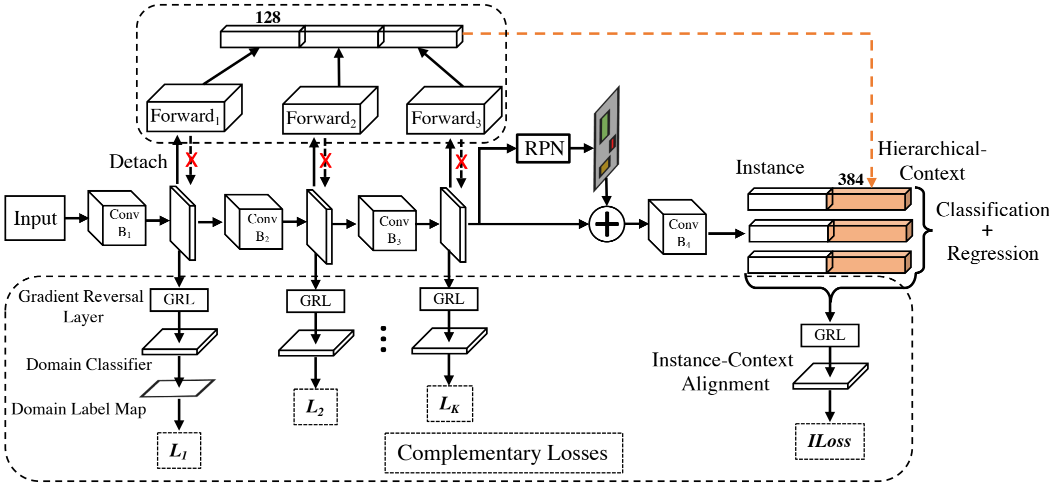


Figure 3: Overview of our SCL framework. More details please refer to Section 2.

with the conventional process, the companion sub-network will be updated jointly with the detection backbone, which may lead to an indistinguishable behavior from these two parts. To this end, in this paper we propose to suppress gradients during backpropagation and force the representation of context sub-network to be dissimilar to the detection network, as shown in Algorithm 1. To our best knowledge, this may be the first work to show the effectiveness of gradient detach that can help to learn better context representation for domain adaptive object detection. Although the detach-based method has been adopted in a few work (Arpit et al., 2019) for better optimization on sequential tasks, our design and motivation are quite different from it. The details of our context sub-network architecture are illustrated in Appendix A.

Algorithm 1: Backward Pass of Our Detach Algorithm

```
INPUT: $\mathbf{G_c}$ is gradient of context network, $\mathbf{G_d}$ is the gradient of detection network, $\mathcal{L}_{det}$ is the detection objective, $\mathcal{L}_{\mathbf{SCL}}$ is the complementary objective;
for $t \leftarrow 1$ to $n_{train\_steps}$ do
    1. Update context net by detection and instance-context objectives: $\mathcal{L}_{det}$(w/o $\mathcal{L}_{rpn}$)+$\mathcal{L}_{\mathbf{ILoss}}$
    2. $\mathbf{G_d} \leftarrow$ stop-gradient($\mathbf{G_c}$;$\mathcal{L}_{det}$)
    3. Update detection net by detection and complementary objectives: $\mathcal{L}_{det}$+$\mathcal{L}_{\mathbf{SCL}}$
```

2.3 Framework Overall

Our framework is based on the Faster RCNN (Ren et al., 2015), including the Region Proposal Network (RPN) and other modules. The objective of the detection loss is summarized as:

$$\mathcal{L}_{det} = \mathcal{L}_{rpn} + \mathcal{L}_{cls} + \mathcal{L}_{reg} \tag{7}$$

where $\mathcal{L}_{cls}$ is the classification loss and $\mathcal{L}_{reg}$ is the bounding-box regression loss. To train the whole model using SGD, the overall objective function in the model is:

$$\min_{\mathcal{F},\mathbf{R}} \max_{\mathbf{D}} \mathcal{L}_{det}(\mathcal{F}(\mathbf{Z}), \mathbf{R}) - \lambda \mathcal{L}_{\mathbf{SCL}}(\mathcal{F}(\mathbf{Z}), \mathbf{D}) \tag{8}$$

where λ is the trade-off coefficient between detection loss and our complementary loss. $\mathbf{R}$ denotes the RPN and other modules in Faster RCNN. Following (Chen et al., 2018; Saito et al., 2019), we feed one labeled source image and one unlabeled target one in each mini-batch during training.

3 Empirical Results

Datasets. We evaluate our approach in three different domain shift scenarios: (1) Similar Domains; (2) Discrepant Domains; and (3) From Synthetic to Real Images. All experiments are conducted

Table 1: Ablation study (%) on Cityscapes to FoggyCityscapes (we use 150m visibility, the densest one) adaptation. Please refer to Section 3.2 for more details.

							AP on a target domain								
Method	Context	L_1	L_2	L_3	ILoss	Detach	person	rider	car	truck	bus	train	mcycle	bicycle	**mAP**
Faster RCNN (Non-adapted)							24.1	33.1	34.3	4.1	22.3	3.0	15.3	26.5	20.3
DA (CVPR'18)	✓						25.0	31.0	40.5	22.1	35.3	20.2	20.0	27.1	27.6
MAF (ICCV'19)							28.2	39.5	43.9	23.8	39.9	33.3	29.2	33.9	34.0
Strong-Weak (CVPR'19)	✓						29.9	42.3	43.5	24.5	36.2	32.6	30.0	35.3	34.3
Kim et al. (2019b) (CVPR'19)							30.8	40.5	44.3	27.2	38.4	34.5	28.4	32.2	34.6
Strong-Weak (Our impl. w/ VGG16)	✓						30.0	40.0	43.4	23.2	40.1	34.6	27.8	33.4	34.1
Strong-Weak (Our impl. w/ Res101)	✓						29.1	41.2	43.8	26.0	43.2	27.0	26.2	30.6	33.4
	✗	LS	FL	✗	✗	✗	29.6	42.2	43.4	23.1	36.4	31.5	25.1	30.5	32.7
	✓	LS	FL	✗	✗	✗	28.5	42.6	43.8	23.2	41.6	24.9	28.3	30.3	32.9
	✓	LS	LS	FL	✗	✗	28.8	**45.5**	44.3	28.6	44.6	29.1	27.8	31.4	35.0
	✓	LS	CE	FL	✗	✗	29.6	42.6	42.6	28.4	46.3	31.0	28.4	33.0	35.3
	✓	LS	FL	FL	✗	✗	30.3	43.1	44.1	26.1	47.8	32.6	27.8	32.4	35.5
	✓	LS	LS	FL	✗	✓	30.0	42.7	44.2	30.0	**50.2**	34.1	27.1	32.2	36.3
	✓	LS	FL	FL	FL	✗	26.3	42.8	44.2	26.7	41.6	36.4	29.2	30.9	34.8
	✓	LS	LS	FL	FL	✓	29.5	43.2	44.2	27.0	42.1	33.3	29.4	30.6	34.9
	✓	LS	FL	FL	FL	✓	29.7	43.6	43.7	26.6	43.8	33.1	30.7	31.5	35.3
	✓	LS	CE	FL	FL	✗	29.8	43.9	44.0	29.4	46.3	30.0	31.8	31.8	35.8
	✓	LS	CE	FL	CE	✓	29.0	42.5	43.9	28.9	45.7	42.4	26.4	30.5	36.2
	✓	LS	CE	FL	FL	✓	30.7	44.1	44.3	30.0	47.9	**42.9**	29.6	33.7	**37.9**
Our full model w/ VGG16	✓	LS	CE	FL	FL	✓	**31.6**	44.0	**44.8**	**30.4**	41.8	40.7	**33.6**	**36.2**	**37.9**
Upper Bound (Saito et al., 2019)	–	–	–	–	–	–	33.2	45.9	49.7	35.6	50.0	37.4	34.7	36.2	40.3

LS: Least-squares Loss; CE: Cross-entropy Loss; FL: Focal Loss; ILoss: Instance-Context Alignment Loss.

on seven domain shift datasets: Cityscapes (Cordts et al., 2016) to FoggyCityscapes (Sakaridis et al., 2018), Cityscapes to KITTI (Geiger et al., 2012), KITTI to Cityscapes, INIT Dataset (Shen et al., 2019), PASCAL (Everingham et al., 2010) to Clipart (Inoue et al., 2018), PASCAL to Watercolor (Inoue et al., 2018), GTA (Sim 10K) (Johnson-Roberson et al., 2016) to Cityscapes.

Implementation Details. In all experiments, we resize the shorter side of the image to 600 following (Ren et al., 2015; Saito et al., 2019) with ROI-align (He et al., 2017). We train the model with SGD optimizer and the initial learning rate is set to 10^{-3}, then divided by 10 after every 50,000 iterations. Unless otherwise stated, we set λ as 1.0 and γ as 5.0, and we use $\mathbf{K} = 3$ in our experiments (the analysis of hyper-parameter $\mathbf{K}$ is shown in Table 7). We report mean average precision (mAP) with an IoU threshold of 0.5 for evaluation.

3.1 How to choose complementary losses

Since there are few pioneer works for exploring the combination of different losses for domain adaptive object detection, here we conduct extensive ablation study for this part to find the best collocation of our SCL method. We follow some objective design from DA and Weak-Strong (Chen et al., 2018; Saito et al., 2019) which provides guidance for us to utilize these losses.

Cross-entropy (CE) Loss. CE loss measures the performance of a classification model whose output is a probability value. It increases as the predicted probability diverges from the actual label:

$$\mathcal{L}_{\mathbf{CE}}(p_{\mathbf{c}}) = -\sum_{\mathbf{c}=1}^{\mathbf{C}} y_{\mathbf{c}} \log p_{\mathbf{c}} \tag{9}$$

where $p_{\mathbf{c}} \in [0, 1]$ is the predicted probability observation of $\mathbf{c}$ class. $y_{\mathbf{c}}$ is the $\mathbf{c}$ class label.

Least-squares (LS) Loss. Following (Saito et al., 2019), we adopt LS loss to stabilize the training of the domain classifier for aligning low-level features. The loss is designed to align each receptive field of features with the other domain. The least-squares loss is formulated as:

$$\mathcal{L}_{\mathbf{LS}} = \mathcal{L}_{loc}^{(s)} + \mathcal{L}_{loc}^{(t)} = \frac{1}{HW} \sum_{w=1}^{W} \sum_{h=1}^{H} \mathbf{D}\left(\hat{\Theta}^{(s)}\right)_{wh}^{2} + \frac{1}{HW} \sum_{w=1}^{W} \sum_{h=1}^{H} \left(1 - \mathbf{D}\left(\hat{\Theta}^{(t)}\right)_{wh}\right)^{2} \tag{10}$$

where $\mathbf{D}\left(\hat{\Theta}^{(s)}\right)_{wh}$ denotes the output of the domain classifier in each location (w, h).

Focal Loss (FL). Focal loss $\mathcal{L}_{\mathbf{FL}}$ (Lin et al., 2017) is adopted to ignore easy-to-classify examples and focus on those hard-to-classify ones during training:

$$\mathcal{L}_{\mathbf{FL}}\left(p_{\mathrm{t}}\right) = -f\left(p_{\mathrm{t}}\right) \log\left(p_{\mathrm{t}}\right), f\left(p_{\mathrm{t}}\right) = \left(1 - p_{\mathrm{t}}\right)^{\gamma} \tag{11}$$

where $p_{\mathrm{t}} = p$ if $d_i = 1$, otherwise, $p_{\mathrm{t}} = 1 - p$.

Table 2: Adaptation results between KITTI and Cityscapes. We report AP of *Car* on both directions, including: K→C and C→K. We re-implemented DA (Chen et al., 2018) and Weak-Strong (Saito et al., 2019) based on the same Faster RCNN framework (Ren et al., 2015).

Method	K→C	C→K
Faster RCNN ((Non-adapted))	30.2	53.5
DA (Chen et al., 2018)	38.5	64.1
DA (Our impl.) (Chen et al., 2018)	35.6	70.8
SW (Our impl.) (Saito et al., 2019)	37.9	71.0
Ours	**41.9**	**72.7**

Table 3: Adaptation results on INIT dataset.

		Car	Sign	Person	**mAP**
s2n	Faster	63.33	63.96	32.00	53.10
	Strong-Weak	67.43	64.33	**32.53**	54.76
	Ours	**67.92**	**65.89**	32.52	**55.44**
	Oracle	80.12	84.68	44.57	69.79
s2r	Faster	70.20	72.71	36.22	59.71
	Strong-Weak	**71.56**	78.07	39.27	62.97
	ours	71.41	**78.93**	**39.79**	**63.37**
	Oracle	71.83	79.42	45.21	65.49
s2c	Faster	–	–	–	–
	Strong-Weak	**71.32**	72.71	43.18	62.40
	Ours	71.28	**72.91**	**43.79**	**62.66**
	Oracle	76.60	76.72	47.28	66.87

3.2 Ablation Studies from Cityscapes to FoggyCityscapes

We first investigate each component and design of our SCL framework from Cityscapes to FoggyCityscapes. Both source and target datasets have 2,975 images in the training set and 500 images in the validation set. We design several controlled experiments for this ablation study. A consistent setting is imposed on all the experiments, unless when some components or structures are examined. In this study, we train models with the ImageNet (Deng et al., 2009) pre-trained ResNet-101 as a backbone, we also provide the results with pre-trained VGG16 model.

The results are summarized in Table 1. We present several combinations of four complementary objectives with their loss names and performance. We observe that “LS—CE—FL—FL” obtains the best accuracy with *Context* and *Detach*. It indicates that LS can only be placed on the low-level features (rich spatial information and poor semantic information) and FL should be in the high-level locations (weak spatial information and strong semantic information). For the middle location, CE will be a good choice. If you use LS for the middle/high-level features or use FL on the low-level features, it will confuse the network to learn hierarchical semantic outputs, so that *ILoss+detach* will lose effectiveness under that circumstance. This verifies that domain adaptive object detection is supported by deep supervision, however, the diverse supervisions should be applied in a controlled manner. Furthermore, our proposed method performed much better than baseline Strong-Weak (Saito et al., 2019) (37.9% *vs.*34.3%) and other state-of-the-arts.

3.3 Similar Domains

Between Cityspaces and KITTI. In this part, we focus on studying adaptation between two real and similar domains, as we take KITTI and Cityscapes as our training and testing data. Following (Chen et al., 2018), we use KITTI training set which contains 7,481 images. We conduct experiments on both adaptation directions K → C and C → K and evaluate our method using AP of *car* as in DA.

As shown in Table 2, our proposed method performed much better than the baseline and other state-of-the-art methods. Since Strong-Weak (Saito et al., 2019) didn't provide the results on this dataset, we re-implement it and obtain 37.9% AP on K→C and 71.0% AP on C→K. Our method is 4% higher than the former and 1.7% higher than latter. If comparing to the non-adapted results (source only), our method outperforms it with a huge margin (about 10% and 20% higher, respectively).

INIT Dataset. INIT Dataset (Shen et al., 2019) contains 132,201 images for training and 23,328 images for testing. There are four domains: sunny, night, rainy and cloudy, and three instance categories, including: car, person, speed limited sign. This dataset is first proposed for the instance-level image-to-image translation task, here we use it for the domain adaptive object detection purpose.

Our results are shown in Table 3. Following (Shen et al., 2019), we conduct experiments on three domain pairs: sunny→night (s2n), sunny→rainy (s2r) and sunny→cloudy (s2c). Since the training images in rainy domain are much fewer than sunny, for s2r experiment we randomly sample the training data in sunny set with the same number of rainy set and then train the detector. It can be observed that our method is consistently better than the baseline method. We don't provide the results of s2c (faster) because we found that cloudy images are too similar to sunny in this dataset (nearly the same), thus the non-adapted result is very close to the adapted methods.

Table 4: Results on adaptation from PASCAL VOC to Clipart Dataset. Average precision (%) is evaluated on target images.

Method	aero	bcycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	hrs	bike	prsn	plnt	sheep	sofa	train	tv	**mAP**
Faster (Non-adapted)	35.6	52.5	24.3	23.0	20.0	43.9	32.8	10.7	30.6	11.7	13.8	6.0	**36.8**	45.9	48.7	41.9	16.5	7.3	22.9	32.0	27.8
BDC-Faster	20.2	46.4	20.4	19.3	18.7	41.3	26.5	6.4	33.2	11.7	26.0	1.7	36.6	41.5	37.7	44.5	10.6	20.4	33.3	15.5	25.6
DA	15.0	34.6	12.4	11.9	19.8	21.1	23.2	3.1	22.1	26.3	10.6	10.0	19.6	39.4	34.6	29.3	1.0	17.1	19.7	24.8	19.8
WST-BSR (Kim et al., 2019a)	28.0	**64.5**	23.9	19.0	21.9	**64.3**	**43.5**	16.4	**42.2**	25.9	**30.5**	7.9	25.5	67.6	54.5	36.4	10.3	**31.2**	**57.4**	43.5	35.7
Strong-Weak (Saito et al., 2019)	26.2	48.5	32.6	**33.7**	38.5	54.3	37.1	18.6	34.8	58.3	17.0	12.5	33.8	65.5	**61.6**	**52.0**	9.3	24.9	54.1	**49.1**	38.1
Ours w/$\mathcal{L}_{ILoss} = FL$	33.4	49.2	36.0	27.1	38.4	55.7	38.7	15.9	39.0	59.2	18.8	23.7	36.9	70.0	60.6	49.7	25.8	34.8	47.2	51.2	40.6
Ours w/$\mathcal{L}_{ILoss} = CE$	**44.7**	50.0	33.6	27.4	**42.2**	55.6	38.3	**19.2**	37.9	**69.0**	30.1	**26.3**	34.4	67.3	61.0	47.9	21.4	26.3	50.1	47.3	**41.5**

Table 5: Adaptation results from PASCAL VOC to WaterColor.

	AP on a target domain						
Method	bike	bird	car	cat	dog	prsn	**mAP**
Source Only	68.8	46.8	37.2	32.7	21.3	60.7	44.6
BDC-Faster	68.6	48.3	47.2	26.5	21.7	60.5	45.5
DA	75.2	40.6	48.0	31.5	20.6	60.0	46.0
Strong-Weak	**82.3**	**55.9**	46.5	32.7	35.5	**66.7**	53.3
Ours	82.2	55.1	**51.8**	**39.6**	**38.4**	64.0	**55.2**

Table 6: Adaptation results on *Car* from Sim10k to Cityscapes Dataset (%). *Source Only* indicates the non-adapted results ($\lambda = 0.1$ and $\gamma = 2.0$ are used).

Method	AP on Car
Source Only	34.6
DA	38.9
Strong-Weak	40.1
Ours	**42.6**

3.4 Discrepant Domains

In this section, we focus on the dissimilar domains, i.e., adaptation from real images to cartoon/artistic. Following (Saito et al., 2019), we use PASCAL VOC dataset (2007+2012 training and validation combination for training) as the source data and the Clipart or Watercolor (Inoue et al., 2018) as the target data. The backbone network is ImageNet pre-trained ResNet-101.

PASCAL to Clipart. Clipart dataset contains 1,000 images in total, with the same 20 categories as in PASCAL VOC. As shown in Table 4, our proposed SCL outperforms all baselines. In addition, we observe that replacing FL with CE loss on instance-context classifier can further improve the performance from 40.6% to 41.5%. More ablation results are shown in our Appendix B.2 (Table 10).

PASCAL to WaterColor. Watercolor dataset contains 6 categories in common with PASCAL VOC and has totally 2,000 images (1,000 images are used for training and 1,000 test images for evaluation). Results are summarized in Table 5, our SCL consistently outperforms other state-of-the-arts.

3.5 From Synthetic to Real Images

Sim10K to Cityscapes. Sim 10k dataset (Johnson-Roberson et al., 2016) contains 10,000 images for training which are generated by the gaming engine Grand Theft Auto (GTA). Following (Chen et al., 2018; Saito et al., 2019), we use Cityscapes as target domain and evaluate our models on *Car* class. Our result is shown in Table 6, which consistently outperforms the baselines.

4 Analysis

Hyper-parameter K. Table 7 shows the results for sensitivity of hyper-parameter $\mathbf{K}$ in Figure 3. This parameter controls the number of SCL losses and context branches. It can be observed that the proposed method performs best when $\mathbf{K} = 3$ on all three datasets.

Table 7: Analysis of hype-parameter $\mathbf{K}$.

Method	$\mathbf{K}$=2	$\mathbf{K}$=3	$\mathbf{K}$=4
from Cityscapes to Foggycityscapes	32.7	**37.9**	34.5
from PASCAL VOC to Clipart	39.0	**41.5**	39.3
from PASCAL VOC to Watercolor	54.7	**55.2**	53.4

Parameter Sensitivity on λ and γ. Figure 4 shows the results for parameter sensitivity of λ and γ in Eq. 8 and Eq. 11. λ is the trade-off parameter between SCL and detection objectives and γ controls the strength of hard samples in *Focal Loss*. We conduct experiments on two adaptations: Cityscapes → FoggyCityscapes (blue) and Sim10K → Cityscapes (red). On Cityscapes → FoggyCityscapes, we achieve the best performance when $\lambda = 1.0$ and $\gamma = 5.0$ and the best accuracy is 37.9%. On Sim10K → Cityscapes, the best result is obtained when $\lambda = 0.1$, $\gamma = 2.0$.

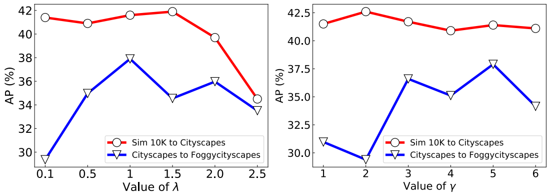
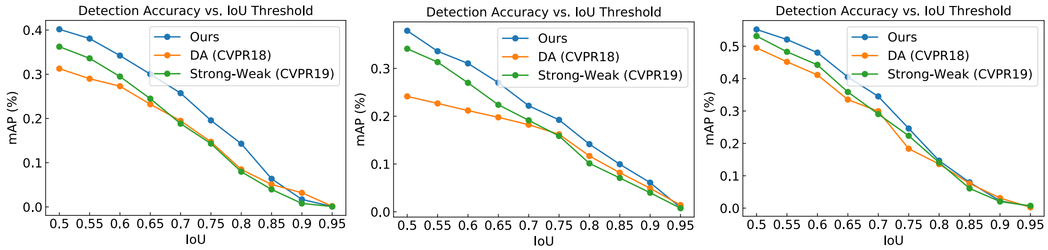


Figure 4: Parameter sensitivity for the value of λ (left) and γ (right) in adaptation from Cityscapes to FoggyCityscapes and from Sim10k to Cityscapes.



(a) from Cityscapes and FoggyCityscapes (b) from PASCAL VOC to Clipart (c) from PASCAL VOC to Watercolor

Figure 5: AP (%) with different IoU thresholds. We show comparisons on three datasets and all results are calculated with different IoU thresholds and illustrated in different colors.

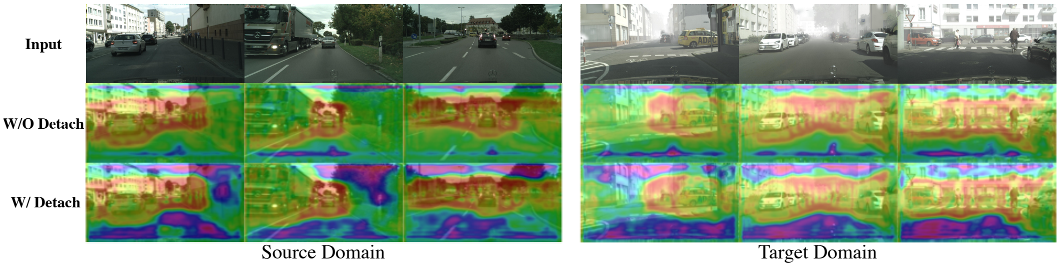
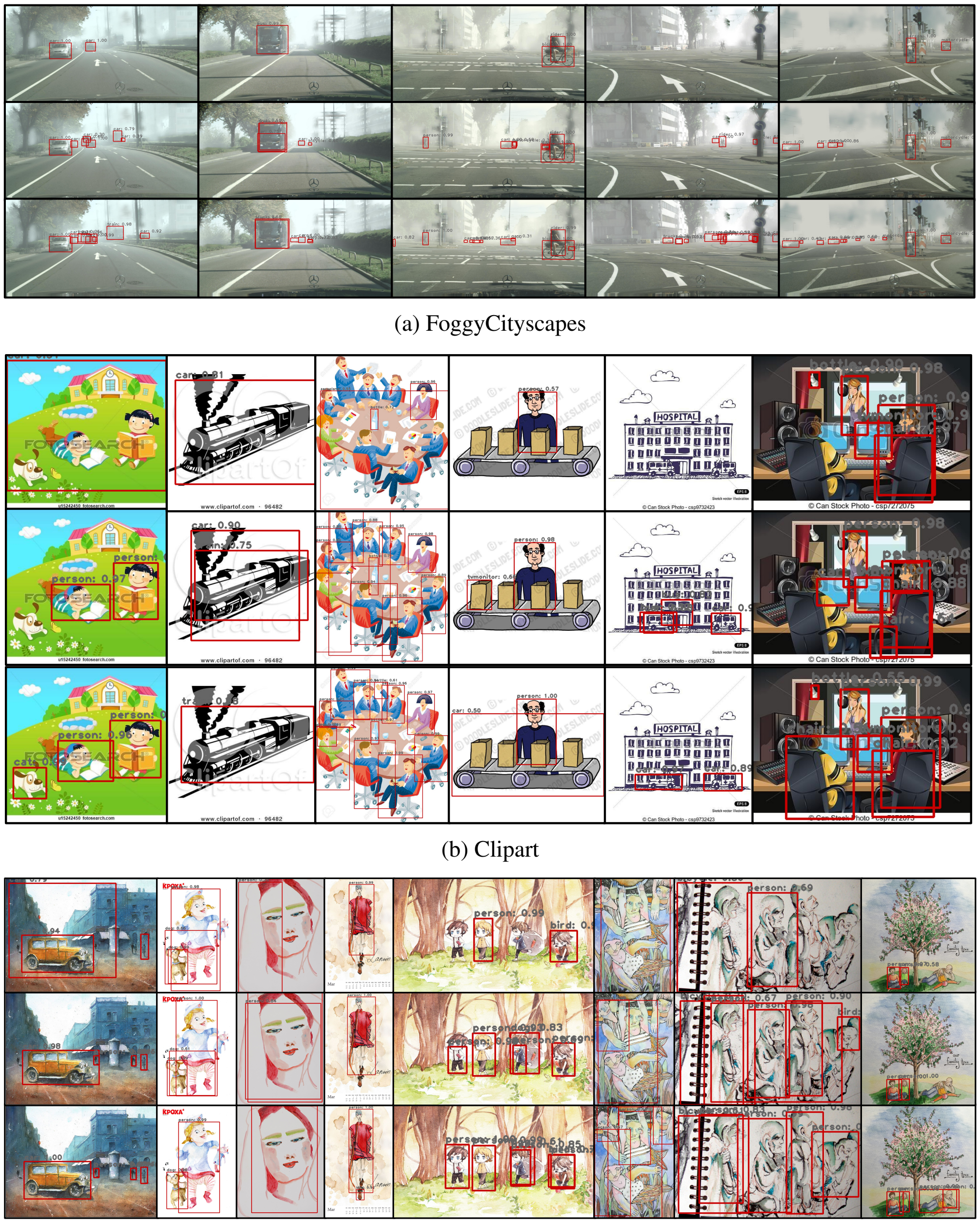


Figure 6: Visualization of *Attention Maps* on source and target domains. We use feature maps after **Conv B3** in Figure 3 for visualizing. Top: Input images; Middle: Heatmaps from models *w/o* gradient detach; Bottom: Heatmaps from models *w/* gradient detach. The colors (red→blue) indicate values from high to low. It can be observed that *w/* detach training, our models can learn more discriminative representation between object areas and background (context).

Analysis of IoU Threshold. The IoU threshold is an important indicator to reflect the quality of detection, and a higher threshold means better coverage with ground-truth. In our previous experiments, we use 0.5 as a threshold suggested by many literature (Ren et al., 2015; Chen et al., 2018). In order to explore the influence of IoU threshold with performance, we plot the performance *vs.* IoU on three datasets. As shown in Figure 5, our method is consistently better than the baselines on different threshold by a large margin (in most cases).

Why Gradient Detach Can Help Our Model? To further explore why gradient detach can help to improve performance vastly and what our model really learned, we visualize the heatmaps on both source and target images from our models *w/o* and *w/* detach training. As shown in Figure 6, the visualization is plotted with feature maps after **Conv B3** in Figure 3. We can observe that the object areas and context from *detach*-trained models have stronger contrast than *w/o* detach model (red and blue areas). This indicates that detach-based model can learn more discriminative features from the target object and context. More visualizations are shown in Appendix C (Figure 8).

Detection Visualization. Figure 10 shows several qualitative comparisons of detection examples on three test sets with DA (Chen et al., 2018), Strong-Weak (Saito et al., 2019) and our SCL models. Our method detects more small and blurry objects in dense scene (FoggyCityscapes) and suppresses more false positives (Clipart and Watercolor) than the other two baselines.



(a) FoggyCityscapes

(b) Clipart

(c) Watercolor

Figure 7: Detection examples with DA (Chen et al., 2018), Strong-Weak (Saito et al., 2019) and our proposed SCL on three datasets. For each group, the first row is the result of DA, the second row is from Strong-Weak and the last row is ours. We show detections with the scores higher than a threshold (0.3 for FoggyCityscapes and 0.5 for other two).

5 Conclusion

In this paper, we have addressed unsupervised domain adaptive object detection through stacked complementary losses. One of our key contributions is gradient detach training, enabled by suppressing gradients flowing back to the detection backbone. In addition, we proposed to use multiple complementary losses for better optimization. We conduct extensive experiments and ablation studies to verify the effectiveness of each component that we proposed. Our experimental results outperform the state-of-the-art approaches by a large margin on a variety of benchmarks. Our future work will focus on exploring the domain-shift detection from scratch, i.e., without the pre-trained models like DSOD (Shen et al., 2017), to avoid involving bias from the pre-trained dataset.

References

Devansh Arpit, Bhargav Kanuparthi, Giancarlo Kerg, Nan Rosemary Ke, Ioannis Mitliagkas, and Yoshua Bengio. h-detach: Modifying the lstm gradient towards better optimization. In *ICLR*, 2019.

Qi Cai, Yingwei Pan, Chong-Wah Ngo, Xinmei Tian, Lingyu Duan, and Ting Yao. Exploring object relation in mean teacher for cross-domain detection. In *CVPR*, 2019.

Hao-Yun Chen, Pei-Hsin Wang, Chun-Hao Liu, Shih-Chieh Chang, Jia-Yu Pan, Yu-Ting Chen, Wei Wei, and Da-Cheng Juan. Complement objective training. In *ICLR*, 2019.

Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *CVPR*, 2018.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 2016.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

Zhenwei He and Lei Zhang. Multi-adversarial faster-rcnn for unrestricted object detection. In *ICCV*, 2019.

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018.

Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *CVPR*, 2018.

Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? *arXiv preprint arXiv:1610.01983*, 2016.

Seunghyeon Kim, Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection. In *ICCV*, 2019a.

Taekyung Kim, Minki Jeong, Seunghyeon Kim, Seokeon Choi, and Changick Kim. Diversify and match: A domain adaptive representation learning paradigm for object detection. In *CVPR*, 2019b.

Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *Artificial intelligence and statistics*, 2015.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, 2016.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 2008.

Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *CVPR*, 2018.

Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *ICCV*, 2017.

Anant Raj, Vinay P Namboodiri, and Tinne Tuytelaars. Subspace alignment based domain adaptation for rcnn detector. *arXiv preprint arXiv:1507.05578*, 2015.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 2015.

Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *CVPR*, 2019.

Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, 2018.

Zhiqiang Shen, Zhuang Liu, Jianguo Li, Yu-Gang Jiang, Yurong Chen, and Xiangyang Xue. Dsod: Learning deeply supervised object detectors from scratch. In *ICCV*, 2017.

Zhiqiang Shen, Mingyang Huang, Jianping Shi, Xiangyang Xue, and Thomas Huang. Towards instance-level image-to-image translation. In *CVPR*, 2019.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

Eric Tzeng, Kaylee Burns, Kate Saenko, and Trevor Darrell. Splat: Semantic pixel-level adaptation transforms for detection. *arXiv preprint arXiv:1812.00929*, 2018.

Yifan Wu, Ezra Winston, Divyansh Kaushik, and Zachary Lipton. Domain adaptation with asymmetrically-relaxed distribution alignment. In *ICML*, 2019.

Weichen Zhang, Wanli Ouyang, Wen Li, and Dong Xu. Collaborative and adversarial network for unsupervised domain adaptation. In *CVPR*, 2018.

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. On learning invariant representations for domain adaptation. In *ICML*, 2019.

Xinge Zhu, Jiangmiao Pang, Ceyuan Yang, Jianping Shi, and Dahua Lin. Adapting object detectors via selective cross-domain alignment. In *CVPR*, 2019.

APPENDIX

A CONTEXT NETWORK

Our context networks are shown in Table 8. We use three branches (forward networks) to deliver the context information and each branch generates a 128-dimension feature vector from the corresponding backbone layers of SCL. Then we naively concatenate them and obtain the final context feature with a 384-dimension vector.

Table 8: Architectures of the forward networks.

Forward Net1	Forward Net2	Forward Net3
Conv $3 \times 3 \times 256$, stride 1, pad 1	Conv $3 \times 3 \times 256$, stride 1, pad 1	Conv $3 \times 3 \times 512$, stride 1, pad 1
ReLU	ReLU	ReLU
Conv $3 \times 3 \times 128$, stride 1, pad 1	Conv $3 \times 3 \times 128$, stride 1, pad 1	Conv $3 \times 3 \times 128$, stride 1, pad 1
ReLU	ReLU	ReLU
Conv $3 \times 3 \times 128$, stride 1, pad 1	Conv $3 \times 3 \times 128$, stride 1, pad 1	Conv $3 \times 3 \times 128$, stride 1, pad 1
ReLU	ReLU	ReLU

B MORE ABLATION STUDIES

Table 9 and 10 show the detailed results on target domains when conducting adaptation from PASCAL VOC to WaterColor and from PASCAL VOC to Clipart dataset. We present results with different combinations of SCL and diverse ablation experiments.

B.1 FROM PASCAL VOC TO WATERCOLOR DATASET

Table 9: AP (%) on adaptation from PASCAL VOC to WaterColor.

	AP on a target domain						
Method	bike	bird	car	cat	dog	prsn	**mAP**
$LS—CE—CE—FL$	76.1	48.8	48.1	29.9	41.2	56.5	50.1
$LS—LS—FL—FL$	72.4	51.8	49.7	41.9	36.6	65.5	53.0
$LS—FL—FL—FL$	77.8	50.6	48.9	40.1	38.7	63.7	53.3
$LS—CE—FL—FL$	82.2	**55.1**	**51.8**	39.6	38.4	64.0	**55.2**
$LS—CE—FL—CE$	64.2	54.8	47.3	38.7	**41.7**	67.9	52.4
W/O Detach	76.2	54.0	49.2	36.7	35.0	**68.6**	53.3
W/O ILoss	76.1	51.7	48.0	31.6	40.4	64.3	52.0
W/O Context	**83.1**	54.5	48.4	34.4	38.8	65.5	54.1
W/O Context&ILoss	69.3	52.8	43.2	**42.7**	36.7	66.0	51.8
W/O CLoss (L_2)	77.1	53.1	49.6	41.0	39.3	67.9	54.7

B.2 FROM PASCAL VOC TO CLIPART DATASET

Table 10: AP (%) on adpatation from PASCAL VOC to Clipart Dataset. Results are evaluated on target images.

Method	aero	bcycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	hrs	bike	prsn	plnt	sheep	sofa	train	tv	**mAP**
$LS—CE—CE—FL$	24.2	48.3	32.6	26.0	31.2	55.3	37.6	12.1	33.0	47.1	23.1	17.0	23.4	57.4	57.3	43.8	19.9	31.7	48.2	42.7	35.4
$LS—CE—FL—CE$	**44.7**	50.0	33.6	27.4	42.2	55.6	38.3	19.2	37.9	**69.0**	**30.1**	26.3	34.4	67.3	61.0	47.9	21.4	26.3	50.1	47.3	**41.5**
$LS—FL—FL—FL$	31.4	52.4	31.5	27.5	39.5	56.9	38.4	13.6	38.3	45.5	23.9	15.8	33.7	73.1	**64.6**	49.5	19.3	26.8	**55.0**	49.9	39.3
$LS—LS—FL—FL$	32.3	56.8	33.2	23.8	39.6	46.0	39.6	17.6	38.7	52.4	14.7	21.2	33.0	72.0	59.6	46.7	21.9	26.9	49.2	**51.8**	38.9
$LS—CE—FL—FL$	33.4	49.2	**36.0**	27.1	38.4	55.7	38.7	15.9	39.0	59.2	18.8	23.7	36.9	70.0	60.6	**49.7**	**25.8**	**34.8**	47.2	51.2	40.6
W/O Detach	33.1	54.5	33.9	**28.2**	**45.3**	**59.4**	31.4	17.4	34.7	39.9	9.8	20.8	33.5	63.0	60.3	40.8	18.7	20.6	51.8	45.6	37.1
W/O ILoss	27.2	54.0	31.9	24.7	38.6	53.7	36.9	15.1	**40.2**	52.4	12.4	**29.6**	36.5	69.3	63.6	43.3	20.2	26.9	50.6	44.3	38.6
W/O Context&ILoss	38.3	**65.4**	25.4	24.6	35.2	47.7	**40.9**	**20.9**	32.6	29.6	4.6	14.7	26.5	**85.2**	60.9	46.6	17.4	22.5	43.9	50.2	36.7
W/O Context	22.5	50.8	33.8	23.5	37.6	48.3	39.4	16.4	38.5	55.7	16.0	23.8	33.0	62.8	59.8	48.4	17.3	28.6	47.6	46.5	37.5
W/O CLoss (L_2)	33.1	57.0	32.5	24.6	39.0	55.9	37.3	15.7	39.5	50.7	20.5	19.8	**37.7**	75.3	60.8	43.9	21.1	26.2	42.9	45.6	39.0

C More Visualizations of Heatmaps

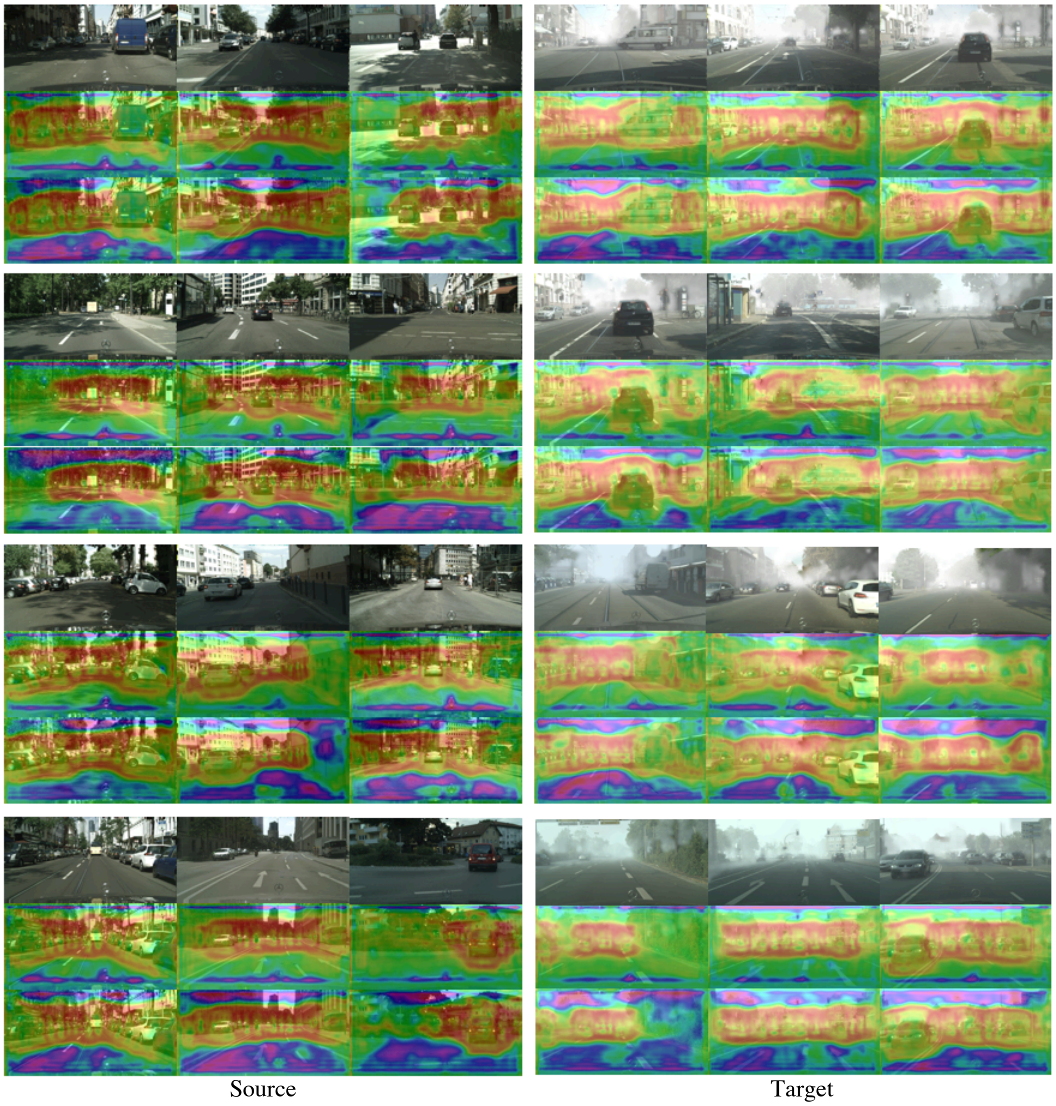


Figure 8: More visualizations of *Attention Maps* on source and target domains. Top: Input images; Middle: Heatmaps from models *w/o gradient detach*; Bottom: Heatmaps from models *w/ gradient detach*. The colors (red→blue) indicate values from high to low.

D Results on Source Domains

In this section, we show the adaptation results on source domains in Table 11, 12, 13 and 14. Surprisingly, we observe that the best-trained models (on target domains) are not performing best on the source data, e.g., from PASCAL VOC to WaterColor, DA (Chen et al., 2018) obtained the highest results on source domain images (although the gaps with Strong-Weak and ours are marginal). We conjecture that the adaptation process for target domains will affect the learning and performing on source domains, even we have used the bounding box ground-truth on source data for training. We will investigate it more thoroughly in our future work and we think the community may also need to rethink whether evaluating on *source domain* should be a metric for domain adaptive object detection, since it can help to understand the behavior of models on both source and target images.

Table 11: AP (%) of adaptation from Cityscapes to FoggyCityscapes. Results are evaluated on source images (Cityscapes) with the same classes as in the target dataset.

	AP on a source domain								
Method	person	rider	car	truck	bus	train	mcycle	bicycle	**mAP**
DA (CVPR'18)	33.5	**48.1**	51.1	37.0	**61.3**	50.0	33.6	36.9	43.9
Strong-Weak (CVPR'19)	**33.7**	47.9	**52.3**	33.5	57.1	39.1	35.1	**37.4**	42.0
Ours	33.0	46.7	51.3	**39.8**	59.2	**51.6**	**36.8**	36.5	**44.4**

Table 12: AP (%) on adaptation from PASCAL VOC to WaterColor. Results are evaluated on source images (PASCAL VOC) with the same classes as in the WaterColor.

	AP on a source domain						
Method	bike	bird	car	cat	dog	prsn	**mAP**
DA (CVPR'18)	**82.0**	78.0	86.3	**89.4**	**83.5**	**82.6**	**83.6**
Strong-Weak (CVPR'19)	81.0	77.4	85.3	89.0	82.9	81.4	82.8
Ours	80.3	**78.1**	**86.5**	87.9	**83.5**	82.0	83.1

Table 13: AP (%) on adaptation from PASCAL VOC to Clipart Dataset. Results are evaluated on source images (PASCAL VOC) with the same classes as in the Clipart.

Method	aero	bcycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	hrs	bike	prsn	plnt	sheep	sofa	train	tv	**mAP**
DA (CVPR'18)	**79.7**	**83.2**	**81.3**	**70.0**	**66.6**	86.0	**87.3**	87.1	57.3	**85.3**	**68.5**	**87.0**	**86.6**	**82.3**	**80.7**	49.7	**80.5**	**75.5**	82.9	**81.6**	**78.0**
Strong-Weak (CVPR'19)	74.3	78.6	66.4	52.7	54.5	80.1	81.4	77.6	43.1	72.9	65.1	74.6	76.5	77.0	75.2	46.3	71.6	64.1	77.0	70.1	69.0
Ours	78.4	81.7	78.4	69.4	60.8	**86.4**	86.0	**87.7**	**57.9**	84.8	68.2	86.4	84.6	82.2	79.3	**50.5**	79.9	73.8	**84.2**	75.2	76.8

Table 14: Adaptation results between KITTI and Cityscapes. We report AP of *Car* on both directions, including: K→C and C→K of source domain.

Method	K→C	C→K
DA (CVPR'18)	**87.9**	52.6
Strong-Weak (CVPR'19)	78.6	**52.9**
Ours	78.5	51.3

E DETAILED RESULTS OF PARAMETER SENSITIVITY ON λ AND γ

We provide the detailed results of parameter sensitivity on λ and γ in Table 15 and 16 with the adaptation of from Cityscapes to FoggyCityscapes and from Sim10K to Cityscapes.

Table 15: AP (%) of adaptation from Cityscapes to FoggyCityscapes with different λ and γ.

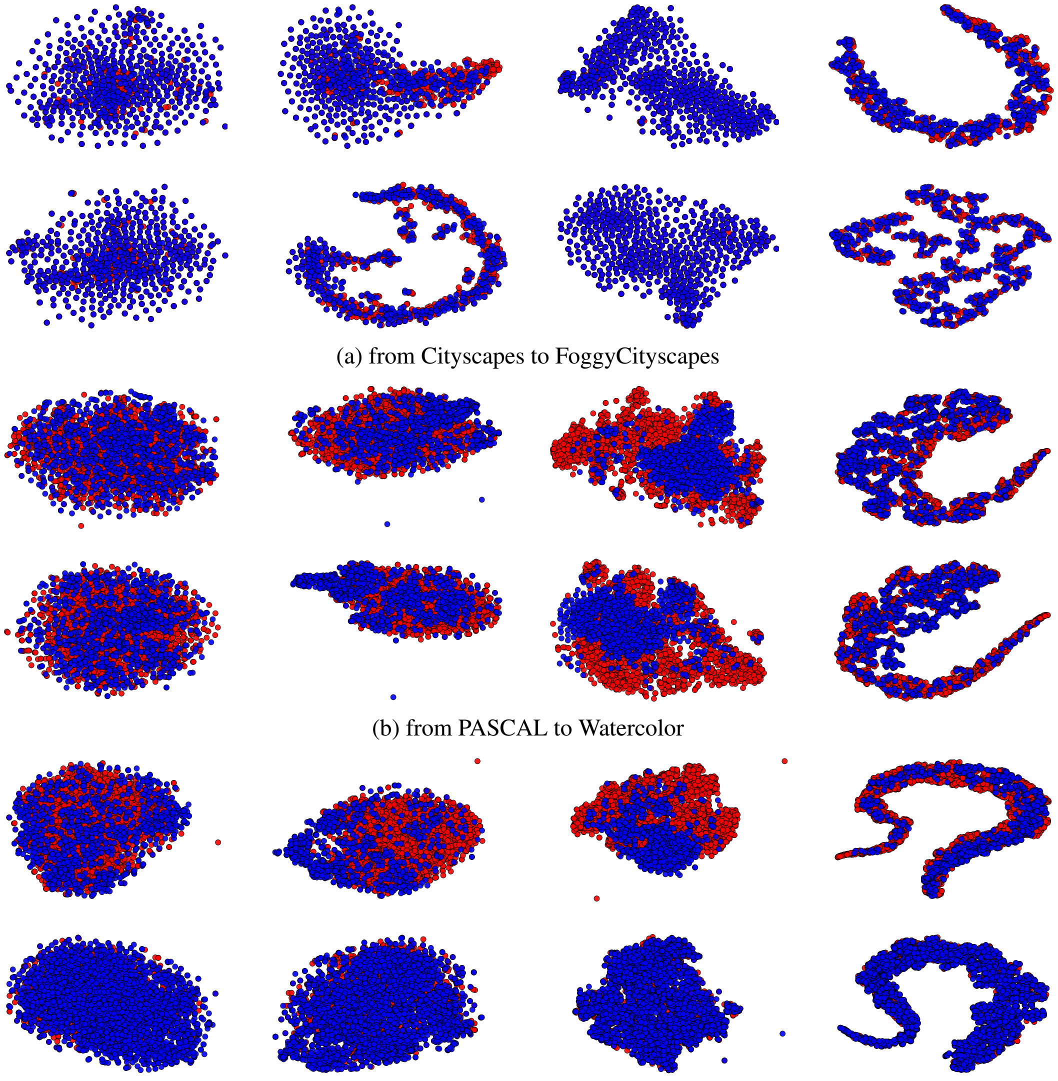
	AP on a target domain								
λ	person	rider	car	truck	bus	train	mcycle	bicycle	**mAP**
0.1	25.8	37.2	24.6	24.2	42.0	33.6	17.5	29.9	29.4
0.5	29.5	42.2	44.4	24.4	45.3	34.1	27.2	32.8	35.0
1.0	30.7	44.1	44.3	30.0	47.9	42.9	29.6	33.7	**37.9**
1.5	26.3	42.2	43.6	25.5	43.8	36.4	26.7	32.0	34.6
2.0	29.5	39.4	43.7	28.7	46.0	39.7	28.7	32.0	36.0
2.5	25.9	40.3	43.3	26.1	40.8	35.2	26.2	30.2	33.5
γ									
1	27.1	41.6	41.3	25.5	41.6	20.3	20.5	30.0	31.0
2	27.8	41.3	36.4	24.2	38.8	12.8	22.9	30.9	29.4
3	29.8	40.7	43.9	29.0	45.0	41.5	30.8	32.0	36.6
4	30.3	42.6	44.2	25.4	45.7	33.9	28.6	30.3	35.1
5	30.7	44.1	44.3	30.0	47.9	42.9	29.6	33.7	**37.9**
6	26.4	42.0	43.8	23.6	45.2	35.2	26.7	30.3	34.2

Table 16: AP (%) of adaptation from Sim10K to Cityscapes with different λ and γ.

AP on a target domain					
λ					
0.1	**0.5**	**1.0**	**1.5**	**2.0**	**2.5**
41.4	40.9	41.6	**41.9**	39.7	34.5
γ					
1	**2**	**3**	**4**	**5**	**6**
41.5	**42.6**	41.7	40.9	41.4	41.1

F Visualization of Intermediate Feature Embedding

In this section, we visualize the intermediate feature embedding on three adaptation datasets. As shown in Figure 9, the gradient *detach*-based models can adapt source and target images to a similar distribution better than *w/o detach* models.



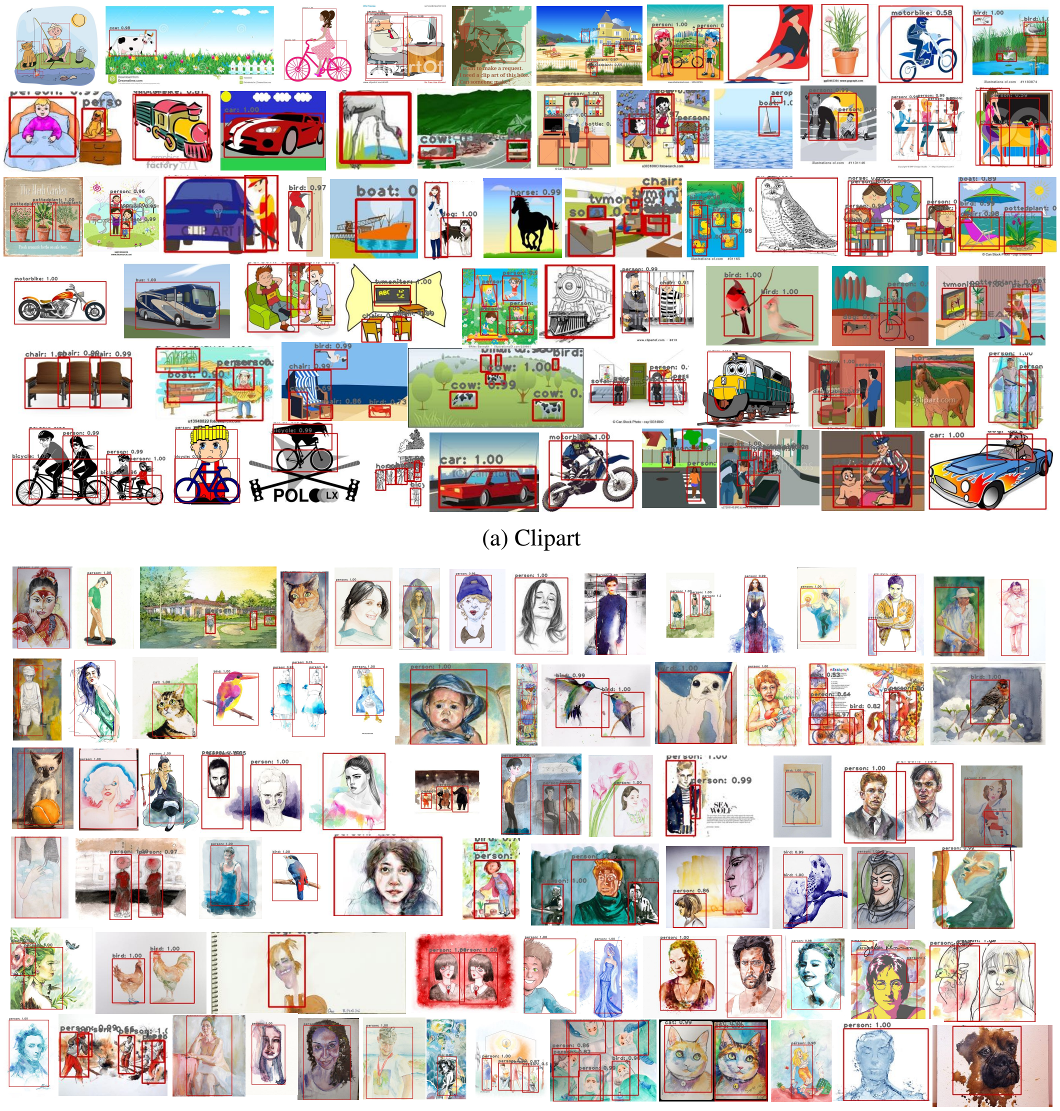
(a) from Cityscapes to FoggyCityscapes

(b) from PASCAL to Watercolor

(c) from PASCAL to Clipart

Figure 9: Visualization of feature embedding on three adaptation datasets by t-SNE (Maaten & Hinton, 2008). Red indicates the source examples and blue indicates the target one. In each group, the first row is the result of *w/o detach* model, the second row is from *with detach* model. In each row, from left to right are results from features after B_1, B_2, B_3 and the 384-dim context features.

G More Detection Visualization



(a) Clipart

(b) Watercolor

Figure 10: More detection examples with our proposed SCL on Clipart and Watercolor. We show detections with the scores higher than 0.5.