SC-UDA: Style and Content Gaps aware Unsupervised Domain Adaptation for Object Detection

Fuxun Yu[†], Di Wang[‡], Yinpeng Chen[‡], Nikolaos Karianakis[‡], Tong Shen[‡], Pei Yu[‡], Dimitrios Lymberopoulos[‡], Sidi Lu[*], Weisong Shi[*], Xiang Chen[†]
[†] George Mason University, [‡] Microsoft, [*] Wayne State University

Abstract

Current state-of-the-art object detectors can have significant performance drop when deployed in the wild due to domain gaps with training data. Unsupervised Domain Adaptation (UDA) is a promising approach to adapt detectors for new domains/environments without any expensive label cost. Previous mainstream UDA works for object detection usually focused on image-level and/or feature-level adaptation by using adversarial learning methods. In this work, we show that such adversarial-based methods can only reduce domain style gap, but cannot address the domain content gap that is also important for object detectors. To overcome this limitation, we propose the SC-UDA framework to concurrently reduce both gaps: We propose fine-grained domain style transfer to reduce the style gaps with finer image details preserved for detecting small objects; Then we leverage the pseudo label-based self-training to reduce content gaps; To address pseudo label error accumulation during self-training, novel optimizations are proposed, including uncertainty-based pseudo labeling and imbalanced mini-batch sampling strategy. Experiment results show that our approach consistently outperforms prior state-of-the-art methods (up to 8.6%, 2.7% and 2.5% mAP on three UDA benchmarks).

1. Introduction

Past few years have witnessed significant breakthroughs on object detection using deep learning [11, 13]. However, most deep learning based object detectors are highly data dependent and thus are susceptible to the *domain gap* emerging in between the training and testing dataset, particularly in real deployment where the environment factors change over time (e.g., weather, light condition, and ambient environment) [20, 29, 30]. Retraining deep learning models with refreshed data, on the other hand, is not always feasible due to labor-intensive and expensive data labeling [16]. To this end, *Unsupervised Domain Adaptation* (UDA) technique becomes a promising alternative solution [37, 5, 17, 35, 22].

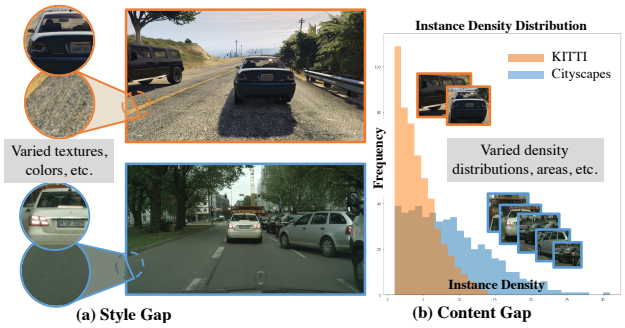

Figure 1: We separate the concept of domain gaps into (a) style gap and (b) content gap. As we will show, both gaps can incur performance drops in domain adaptation.

Existing UDA methods [8, 29, 22] are primarily designed for reducing the *style gap* (e.g., the difference in color, texture, and brightness) between the well-learned training dataset (source domain) and the unpredictable testing dataset (target domain), as shown in Fig. 1 (a). Recent works on the image classification task, on the other hand, reveal that the *content gap* severely undermines the classification performance. An illustrative example is shown in Fig. 1 (b) where two popular benchmark datasets KITTI [9] and Cityscapes [6] manifest distinct distribution on label density. While the content gap has been proved detrimental to image classification in literature [34, 27, 19, 1], the impact of content gap on object detection task remains an open question, even though it could be prevalent in practice, such as building styles, traffic patterns, and landscapes, *etc*., may differ across geographic areas and even change over time.

Our Contribution: In this paper, we quantitatively analyze the significant impact of style and/or content gaps on object detection. Motivated by it, we propose SC-UDA — a *Style & Content Gaps aware UDA* framework to reduce both style and content gaps in object detection. We summarize the key design components below.

- To reduce the *style gap*, we propose a *Fine-grained*

Domain Style Transfer technique. Such design improves the style transfer with finer granularity and better preserves low-level styles (*e.g.*, local edges, textures) for small objects, which are beneficial to the downstream object detection task.

- To address the *content gap*, we annotate the dataset in the target domain with high-quality pseudo labels and conduct iterative *Self-Training* to train detectors on both source-domain and target-domain datasets. Such pseudo labels are optimized to approximate the content distribution in the target domain, thus helping reduce the content gap in the domain adaptation.
- As the pseudo labels in self-training can introduce label errors and influence the adaptation quality, we further propose two optimizations: (i) *Uncertainty-based pseudo label fusion* to generate high-quality pseudo labels; (ii) *Imbalanced mini-batch sampling* to adjust the supervision ratio of real and pseudo labels to balance the influence of potential label errors.

We evaluate our proposed SC-UDA framework on several detection adaptation benchmarks, including *synthetic-to-real*, *cross-camera*, and *normal-to-foggy*, that represent different degrees of domain style and content gaps. Our approach demonstrates consistently better results than the best prior works by up to 8.6%, 2.7%, and 2.5% mAP on three benchmarks, achieving the new state-of-the-art domain adaptive object detection performance. We also ablate the importance of each key component of SC-UDA towards domain adaptation improvements on object detection tasks.

2. Related Work

Unsupervised Domain Adaptation: State-of-the-art deep neural networks often face significant performance drop due to the changing environments. Many unsupervised domain adaptation (UDA) techniques are proposed, e.g., MMD distance minimization [12], sub-space alignment [7]. Recently, adversarial learning based methods with gradient reverse layer for feature-invariant learning achieved great performance for classification adaptation problems [8]. Many works also generalized it into other vision tasks including semantic segmentation [14, 33], object detection [5], *etc.* However, UDA for classification concerns only single object per image, while detector adaptation targets at images with multiple objects. Thus, UDA for detection requires more fine-grained adaptation than classification.

Domain Adaptation for Detection: Similar to UDA in classification, the predominant trends of UDA works for object detection also used adversarial learning based methods. For example, [5] first used adversarial learning methods to align the image-level and instance-level features. Following that, [37] proposed a region-level alignment method to target at a middle granularity. Some other adversarial methods

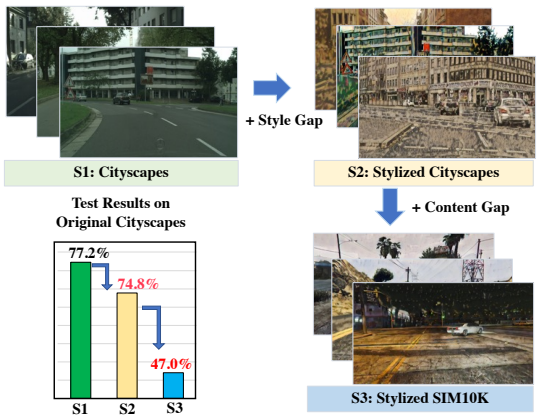

Figure 2: We conduct style randomization [10] to introduce style gaps (S1 to S2). Then the source dataset is changed to further involve the content gaps (S2 to S3). In the test results, both gaps are shown to incur performance drops in domain adaptation.

have also targeted at different levels with weighting strategies [25, 31, 15]. These methods can well solve the style gaps between different domains. But due to the challenges like varied object classes, locations and densities, adversarial methods may cause certain feature misalignment. Moreover, these methods mostly focused on feature alignment on image style gaps and did not account for content distribution gap (e.g., different object density distribution) which are also important for object detection.

Self-Training for UDA: Self-Training with pseudo labels utilizing both labeled and unlabeled data has been shown to be an effective way of using unlabeled data to boost the end-task performance [21, 18, 2]. Such combined dataset setting also applies to the UDA problem, where we have labeled source domain and unlabeled target domain. Recently, there are a few works utilizing self-training for UDA problems in classification, segmentation, and detection tasks. For example, [38, 39] utilized self-training for classification and segmentation adaptation. [17] utilized weak self-training for single-shot detector adaptation. [24] used pseudo-labeled data but the labels are from extra video data annotation, which belongs to weakly-supervised domain adaptation.

One common shortcoming of previous methods is that they mostly used classification confidence as the label selection criteria. However, such confidence-based selection is usually sub-optimal in detection as it fails to represent the localization accuracy. Therefore, we re-innovate the pseudo labeling method with uncertainty-based box selection and fusion method for detection tasks.

3. Analysis: Style and Content Gaps

In this section, we confirm the existence and quantify the influence of style/content gap in object detection. Specifically, we leverage style randomization [10] and content manipulation to disentangle the performance degradation

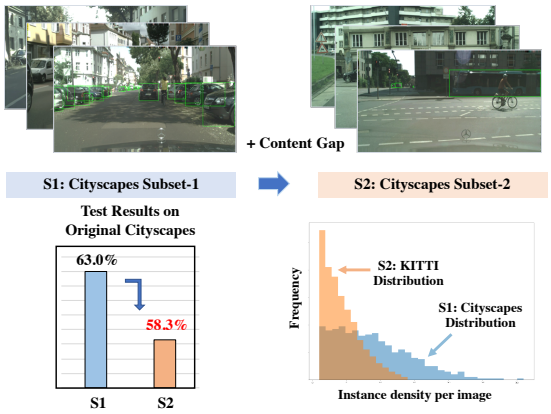

Figure 3: We test content gap by making two Cityscapes subsets with the same number of images but following two different density distributions. The content gap alone is also shown to affect the adaptation performance. (Detailed settings in supp. material.)

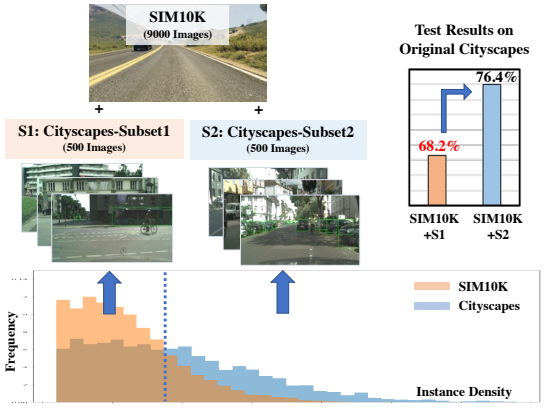

Figure 4: We reduce content gap by combining two Cityscapes subsets following two density distributions into SIM10K dataset. Here adding the subset (S2) whose distribution is non-overlapped with SIM10K leads to higher performance than the other (S1).

caused by style and content gaps, respectively.

Impact of Style Gap: First, we transform the style of Cityscapes dataset (S1-Cityscapes) to different styles (S2-Cityscapes) through style randomization. The visualization examples are shown in Fig. 2. By doing so, we witness that the style gap alone (without content change) results in certain mAP drop on detection performance (77.2% $\rightarrow$ 74.8%).

Impact of Style + Content Gap: In Fig. 2, we then introduce content gap on S3-SIM10K: a style transferred dataset sharing the same randomized styles with S2-Cityscapes but differs in their content distribution. We test the S3-trained model on the original Cityscapes testset and observe that the style and content gaps together incur more significant (74.8% $\rightarrow$ 47.0%) performance drop.

Impact of Content Gap: We further analyze content gap alone by selecting two subsets from Cityscapes with same number of images (500) but following two density distributions (Cityscapes and KITTI distribution). Models trained on two subsets are then tested on the Cityscapes testset. As Fig. 3 shows, the detector trained on KITTI-distribution shows non-negligible mAP drop (63.0% $\rightarrow$ 58.3%) due to certain content gap with the Cityscapes distribution.

Motivation: Reducing Both Style & Content Gaps: The above results demonstrate that both style and content gaps are detrimental to object detection performance. Unfortunately, previous UDA works usually tackle the style gap while ignoring the content distribution gaps, thus only achieving sub-optimal performance.

To reduce the content gap influence, one intuitive way is to involve the real target-domain data distribution into the training process. To verify its effectiveness, we conduct a preliminary test by SIM10K to Cityscapes domain adaptation with two auxiliary Cityscapes subsets. Specifically as shown in Fig. 4, one Cityscapes subset (S1) is sampled from the overlapped distribution, while the other one (S2) is sampled from the less-overlapped distribution. Training models using these two settings, SIM10K+S2 (76.4%) shows much better performance than SIM10K+S1 (68.2%), indicating the advantages of involving the missing distributions.

Such results show the effectiveness of involving missing content distributions to reduce the content gaps. Based on this, our work proposes to involve the target-domain dataset into training by adopting pseudo labels for content gap reduction. Combined with the fine-grained style transfer for style gap reduction, we propose a holistic framework that could concurrently reduce style & content gaps, achieving the new SOTA performance in UDA for detection.

4. SC-UDA Framework

Framework Overview: Our framework consists of two major steps as shown in Fig. 5:

(a) *Fine-grained Domain Transfer*: We first conduct domain style transfer to reduce the style gaps. Considering the detector needs to detect multi-scale objects from large to small, we conduct style transfer with the finest granularity to better preserve small object details, which is shown to greatly boost the detection adaptation performance. An initial annotator will be trained on the transferred domain to generate initial pseudo labels on the target domain to launch the following self-training process.

(b) *Iterative Self-Training*: We then run iterative self-training to reduce the content gap by combing the source domain data and the target domain data (with pseudo labels). As the pseudo label may contain errors, we also propose two optimization techniques: (1) *Uncertainty-based Pseudo Label Optimization*, a novel uncertainty-based pseudo label selection and fusion method, which outperforms the confidence-based labeling by large margins; (2) *Imbalanced Mini-batch Sampling* to balance the training error and stabilize the self-training process.

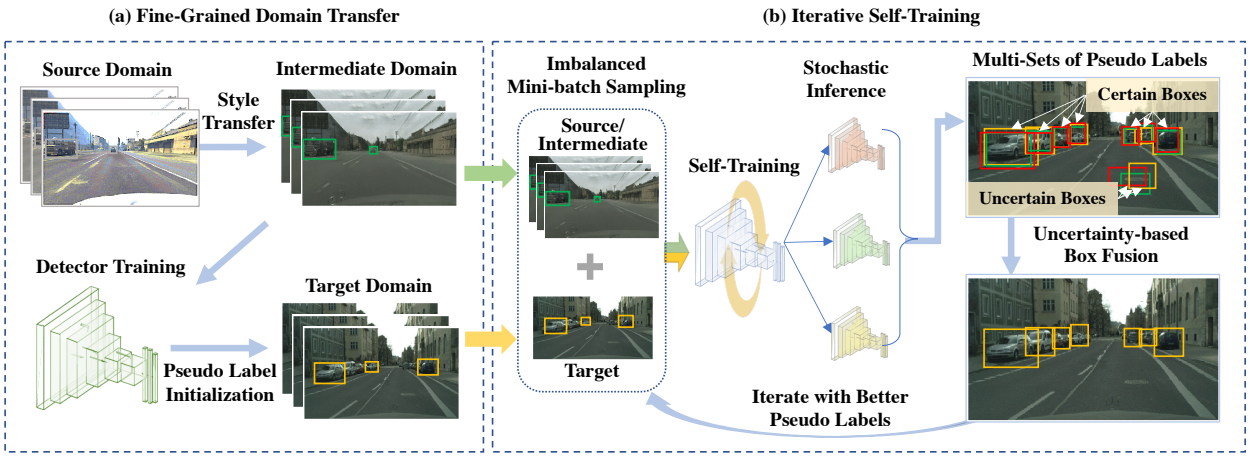

Figure 5: SC-UDA Framework Overview. (a) Fine-grained domain transfer first transfers the source to an intermediate domain to reduce style gaps. A detector is trained on the intermediate domain as the initial pseudo label annotator. (b) Then we conduct iterative self-training with combined source/target domain data to reduce content gaps. Two optimizations are proposed to mitigate label error's influence: (b-1) imbalanced mini-batch sampling; (b-2) uncertainty-based box fusion to generate high-quality labels.

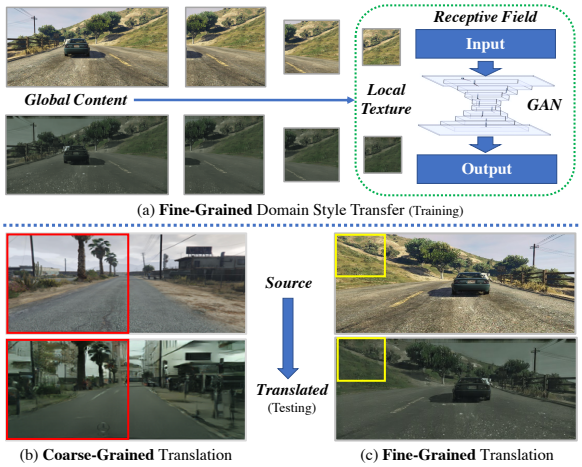

Figure 6: We optimize CycleGAN with receptive field restrictions to conduct detection-oriented fine-grained style transfer.

4.1. Domain Transfer for Style Gap Reduction

In SC-UDA, we first conduct domain style transfer to reduce the style gap by an optimized CycleGAN model [36]. The initial CycleGAN design is often *coarse-grained* that can dramatically change the image backgrounds and edges of objects, as shown in Fig. 6 (b). Such coarsely transferred images can become harmful for detectors to detect small objects. Therefore, we propose to address this limitation by optimizing it with receptive field restriction to conduct detection-oriented *fine-grained* style transfer.

4.1.1 Receptive Field Restriction

To benefit the detector adaption, the style translator should less touch the global contents but focus more on the low-level styles (e.g., local textures, object details, etc). To this end, we optimize CycleGAN with a simple yet effective method by imposing restrictions on the model's receptive field. Specifically, given the training loss of native GAN (same for the cycle consistency loss):

$$L_{GAN}(G, D, X_s, X_t) = \mathbb{E}_{x\sim p(t)}[\log D_T(x_t^{'})] + \mathbb{E}_{x\sim p(s)}[\log(1 - D_T(G(x_s^{'})))], \quad (1)$$

instead of using the full-size source images $x_s \sim p(s)$ and target images $x_t \sim p(t)$, we crop random patches $x_s^{'}$ and $x_t^{'}$ from the image pairs as the input, as shown in Fig. 6 (c). In this way, the receptive field of CycleGAN can be then restricted to be much smaller than most objects, so that it learns to translate only local textures instead of objects, fulfilling the goal of fine-grained style transfer.

Figure 6 (b) and (c) compare the results of coarse-/fine-grained translation. For coarse-grained translation, the background/objects of seed image are dramatically changed. While for the fine-grained translation, only local textures are changed to match the target domain and the objects details are well-preserved. As we will show later, such fine-grained design can greatly benefit the small object detection adaptation performance in various of scenarios.

Source to Intermediate Domain Transfer: Based on the fine-grained style translation model, we then translate the data from source domain $(X_s,\ Y_s)$ to the intermediate domain $(X_m,\ Y_s)$. With smaller style gaps from the target domain, the intermediate domain data $(X_m,\ Y_s)$will then be utilized for pseudo label initialization to launch the following self-training process.

4.2. Self-Training for Content Gap Reduction

To further reduce the content gaps, we next conduct self-training with both optimized pseudo labels and a error-balanced training process.

4.2.1 Iterative Self-Training with Pseudo Labels

The basic idea of self-training is combining both source and target data into training process, where the target domain data is pseudo labeled. This process follows several steps: (a) pseudo label initialization, (b) self-training, and (c) iterate with better pseudo labels.

Pseudo Label Initialization: To get the pseudo labels, we first train an initial annotator (model) $F_{init}(\theta, \cdot)$ on the style transferred data in the intermediate domain (X_m, Y_s):

$$\text{Minimize } \mathbb{E}_{x,y\sim(X_m,Y_s)}[\,\text{Loss}(F_{init}(\theta,x),y)\,], \tag{2}$$

where θ is the detector weights, and (X_m, Y_s) is the style-transferred intermediate domain data. The annotator is then applied on the target domain images to get the prediction results as pseudo labels:

$$Y_t^{psd} = F(\theta, X_t). \tag{3}$$

Self-Training with Pseudo Labels: Although pseudo labels are not perfectly accurate, they enable us to involve the real target domain data distribution during training. We therefore retrain the new model $F(\theta, \cdot)$ by combining both source/target domain data to further boost the performance:

$$\begin{aligned}\text{Minimize } &\mathbb{E}_{x,y\sim(X_s,Y_s)}[\,\text{Loss}_s(F(\theta,x),y)\,] \\ &+\mathbb{E}_{x,y\sim(X_t,Y_t^{psd})}[\,\text{Loss}_t(F(\theta,x),y)\,].\end{aligned} \tag{4}$$

Iterate with Better Labels: With the target domain data involved, self-training could often yield better performance. Therefore, we can use the new model as a better annotator to improve the pseudo label quality. Such a "self-labeling" and "self-teaching" process could be conducted for multiple rounds, forming the iterative self-training process.

4.2.2 Uncertainty-based Pseudo Label Optimization

Due to the prediction inaccurateness, the pseudo labels can inevitably contain some errors. Traditional pseudo label optimization commonly chooses labels above high confidence thresholds, e.g., in many classification problems [23, 3]. However, as detectors contain both classification and localization heads, the classification confidence can fail to indicate the localization accuracy, making the confidence thresholding ineffective for detection tasks.

To avoid such issues, we propose an *uncertainty-based box selection & fusion* method to enhance the pseudo label quality for the detection problem. The general idea is shown in Fig. 7: We obtain multi-sets of predictions from different detectors. If the predictions show high localization agreement on the same box, we consider it has higher certainty and is more likely to be true. We then conduct box fusion to generate the final coordinates, otherwise these boxes are removed to exclude inaccurate pseudo labels.

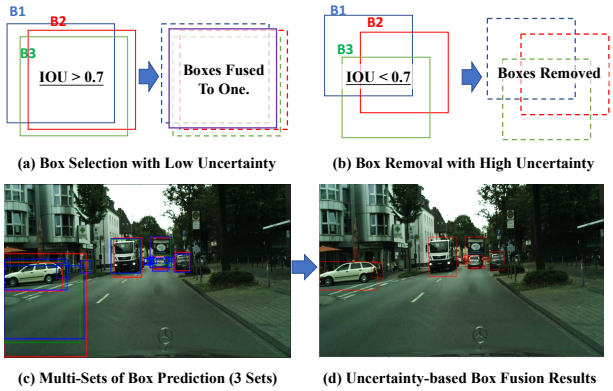

Figure 7: Uncertainty-based Detection Pseudo Labeling. We select boxes with high certainty (higher IoUs) and fuse their coordinates to generate accurate box labels.

Multi-Detector Simulation via Stochastic Inference: To get the uncertainty estimation, we first need to get multiple sets of predictions and detectors. Previous uncertainty methods like Mean-Teacher [28] and Co-Teaching [4] train multiple models or multiple network heads, bringing many training/parameter overheads.

By contrast, we propose to conduct multi-detector simulation in a overhead-free way via stochastic inference. Specifically, stochastic dropout is added in detector's ROI feature extractor layers during both training and testing phases. To get multi-sets of pseudo labels, we conduct model inferences with different dropout masks ξ:

$$Y_t^{psd}(\xi) = F(\theta,\ \xi,\ X_t), \quad \text{where } \xi \sim Ber(p). \tag{5}$$

Here $Ber(p)$ is the Bernoulli distribution with dropout ratio p (usually set as 0.5) to generate the dropout mask.

With such stochastic inference, we can thus get multi-sets of pseudo labels but without any extra training/parameter overheads[1].

Uncertainty-based Box Selection & Fusion: With multi-sets of pseudo labels via stochastic inference, the next step is to select the boxes with higher localization agreement (i.e., lower uncertainty).

To do so, we evaluate the boxes' localization agreement by the *Intersection-over-Union (IoU)* metric. As shown in Fig. 7 (a), for the n boxes with higher localization IoUs than the given threshold, we regard them as with lower uncertainty. We then retain the boxes and fuse them into one

[1]Notably, the drop out layer is also commonly used in detection model training and incurs no accuracy influence on the detector performance.

final box by coordinate averaging. By contrast in Fig. 7 (b), the less-overlapped boxes with smaller IoUs are removed from the pseudo label sets as they are more likely to contain inaccurate localization coordinates, and thus are considered harmful to the training process. Therefore, the pseudo label selection is defined as:

$$y_i^{psd} = \begin{cases} 1, & \text{IoU}(\text{box}_{0,1,\dots n}) > \delta; \\ 0, & \text{Otherwise.} \end{cases} \quad (6)$$

Here δ is the IoU threshold among n boxes to estimate the localization uncertainty of the boxes. If the box is selected, the final localization coordinates are obtained by averaging:

$$\text{Fused coordinates} = \mathbb{E}_{box_i \sim Y_t^{psd}(\xi_i)}[\text{box}_{0,1,\dots n}] \quad (7)$$

And the classification label is obtained by major voting of n boxes, which we find accurate for most boxes.

Fig. 7 (c)(d) show an example of the above uncertainty-based box selection process. Specifically, we use 3 sets of pseudo labels generated by our stochastic inference. As shown in Fig. 7 (c), the true boxes with objects inside usually show high IoUs, while the wrong boxes did not. By our uncertainty-based box selection and fusion method, we could thus remove such wrong predictions and generate more accurate pseudo labels as shown in Fig. 7 (d).

4.2.3 Balancing Error by Imbalanced Sampling

Besides the uncertainty-based methods, we also propose an imbalanced sampling strategy to reduce the influence of pseudo label errors. Specifically, during self-training, the target domain loss $Loss_t$ in Eq. 4 can become inaccurate due to the pseudo label error. By contrast, the source-domain samples have ground-truth labels (X_s, Y_s), $Loss_s$ is more accurate. Therefore, we propose a weighted supervision loss in source/target domains for self-training.

To do so, we statistically over-sample the source images and under-sample the target images in each mini-batch to adjust the ratio of source/target-domain supervision:

$$Loss = \sum_0^i Loss_s + \sum_0^j Loss_t, \ \ \textbf{s.t.} \ i > j, \quad (8)$$

where i and j denote the sampling number of source and target domain images in each mini-batch.

As we will analyze later, the pseudo labels usually contain localization errors. In such case, the source-domain labels provide correct localization loss as supervision, helping improve the final adaptation performance.

5. Experimental Evaluation

Experiments Setup: We follow the experiment settings of previous works [5, 37, 25, 31]. The detector is Faster-RCNN with VGG16 backbone. Three benchmarking domain adaptation scenarios are evaluated, namely Synthetic-to-Real (*Sim2City*), Cross-Camera (*Kitti2City*) and Normal-to-Foggy (*City2Foggy*). For the image size, previous works mainly use two settings: 512 pixels or 600 pixels as the image's shorter side. Works using higher resolution (600 pixels) usually achieve better performance [25, 31]. For fair comparison, we report and compare our results under both settings. We use the mean average precision (mAP) at IoU threshold 0.5 for evaluation.

Table 1: KITTI to Cityscapes Adaptation Performance. We implement both resolution settings (512 & 600) for fair comparisons.

Methods	*Car AP*
^{512}Baseline [non-adapt]	36.4
^{512}CVPR'18 [5]	38.5
^{512}CVPR'19 [37]	42.5
^{512}Ours w/ ST	42.6 (+6.2)
^{512}Ours w/ DT	41.4 (+5.0)
^{512}Ours w/ (DT + ST)	**45.2** (+8.8)
^{512}Oracle Performance	61.6
^{600}Baseline [non-adapt]	37.5
^{600}Ours w/ (DT + ST)	**46.4** (+8.9)
^{600}Oracle Performance	62.7

Table 2: SIM10K to Cityscapes Adaptation Performance.

Methods	*Car AP*
^{512}Baseline [non-adapt]	33.0
^{512}CVPR'18 [5]	39.0
^{512}CVPR'19 [37]	43.0
^{512}Ours w/ ST	40.6 (+7.6)
^{512}Ours w/ DT	48.1 (+15.1)
^{512}Ours w/ (DT + ST)	**49.0** (+16.0)
^{512}Oracle Performance	61.6
^{600}Baseline [non-adapt]	34.6
^{600}CVPR'19 [25]	42.3
^{600}ICCV'19 [31]	42.8
^{600}CVPR'20 [35]	43.8
^{600}Ours w/ (DT + ST)	**52.4** (+17.8)
^{600}Oracle Performance	62.7

5.1. Overall Domain Adaptation Performance

Cross-Camera Adaptation: We first evaluate our framework in cross camera scenarios, KITTI $\rightarrow$ Cityscapes [9]. Task of single-class *car* detection is evaluated as per the setting in [5, 37, 36, 25, 31]. *Baseline* represents source-domain trained models. The *ST* denotes applying our iterative self-training only, and *DT* denotes the fine-grained domain transfer (*DT*) only. *Oracle* represents the performance trained on fully-labeled target domain.

The results are shown in Table 1. Compared to baseline, ST alone and DT alone improve AP by +6.2% and +5.0%, respectively. Finally, combining ST+DT brings +8.8% im-

Table 3: Multi-Class Cityscapes to Foggy-Cityscapes Adaptation Performance

Methods	Person	Rider	Car	Truck	Bus	Train	Motor	Bicycle	mAP
^{512}Baseline [non-adapt]	24.4	30.5	32.6	10.8	25.4	9.1	15.2	28.3	22.0
^{512}CVPR'18 [5]	25.0	31.0	40.5	22.1	35.3	20.2	20.0	27.1	27.6
^{512}CVPR'19 [37]	33.5	38.0	48.5	26.5	39.0	23.3	28.0	33.6	33.8
^{512}Ours w/ (DT + ST)	33.9	38.7	52.1	26.3	43.4	32.9	27.5	35.5	**36.3 (+14.3)**
^{512}Oracle Performance	40.7	44.7	61.9	28.2	51.3	33.0	31.4	40.9	41.5
^{600}Baseline [non-adapt]	29.7	32.2	44.6	16.2	27.0	9.1	20.7	29.7	26.2
^{600}CVPR'19 [25]	29.9	42.3	43.5	24.5	36.2	32.6	30.0	35.3	34.3
^{600}ICCV'19 [31]	33.2	44.2	44.8	28.2	41.8	28.7	30.5	36.5	36.0
^{600}CVPR'20 [32]	32.9	43.8	49.2	27.2	**45.1**	**36.4**	30.3	34.6	37.4
^{600}CVPR'20 [35]	34.0	**46.9**	52.1	**30.8**	43.2	29.9	**34.0**	37.4	38.6
^{600}Ours w/ (DT + ST)	**38.5**	43.7	**56.0**	27.1	43.8	29.7	31.2	**39.5**	**38.7 (+12.5)**
^{600}Oracle Performance	42.7	49.2	63.4	35.8	53.1	22.7	33.5	39.7	42.5

provement over baseline, which outperforms the previous best result [37] by **+2.7%**.

Synthetic-to-Real Adaptation: We then evaluate the synthetic to real adaptation scenario with SIM10K $\rightarrow$ Cityscapes datasets. SIM10K is a synthetic dataset generated by the GTA-V game engine [16], and Cityscapes consists of images of real street scenes taken at different cities [6]. The results are shown in Table 2.

Compared to baseline, our methods of ST and DT improves *car* AP by +7.6% and +15.1%, respectively. By combining both methods (DT+ST), the result boosts up to 49.0% AP (+16.0%). Compared to prior SOTA works, our approach achieves the current best performance, **+6.0%** and **+8.6%** better than the best prior works in 512 and 600 pixel resolutions, respectively.

Multi-Class Normal to Foggy Adaptation: We finally evaluate our framework on Cityscapes $\rightarrow$ Foggy-Cityscapes [26] as a multi-class scenario. As shown in Table 3, our approach achieves the best performance under both resolution settings (512 and 600), achieving +14.3% and +12.5% mAP compared to baseline performance. Compared to other SOTA works [37, 31, 35, 17], our method achieves consistent mAP improvement (e.g., **+2.5%** in 512 settings, **+0.1%**$\sim$**4.4%** in 600 settings). Note that, our performance (38.7%) has nearly achieved the oracle model performance (42.5%) with only a 3.8% mAP margin, demonstrating the adaptation effectiveness.

5.2. Self-Training Improvement Analysis

In this part, we aim to understand why source + target-domain self-learning helps improve the overall detection performance. To do so, we evaluate the contribution of different loss components (classification & localization loss) from different labels (source-domain GT labels & target-domain pseudo labels). The results are shown in Table 4.

Pseudo Labels Help Classification: We first evaluate the target-domain pseudo label's contribution. As the first four rows in Table 4 show, pseudo labels benefit the detector (+7.1%) most when only classification loss is used. By contrast, the localization loss alone hurts the performance (-5.1%). This implies that most pseudo labels correctly covered target objects, which indeed promote detectors to learn genuine object features in the target domain.

Table 4: Localization/Classification Loss Analysis.

Train Settings (Kitti2City)	AP
Source only	36.4
Source + Target (localization loss only)	31.3 (-5.1)
Source + Target (classification loss only)	**43.5 (+7.1)**
Source + Target (both)	44.3 (+7.9)
Target only	40.8
Source (localization loss only) + Target	**44.4** (+3.6)
Source (classification loss only) + Target	43.5 (+2.7)
Source (both) + Target	44.3 (+3.5)

GT Labels Help Localization: By contrast, the GT labels in the source domain bring more gain from the localization loss (+3.6%). This implies the importance of GT labels' localization supervision, i.e., to provide accurate localization loss and also mitigate the pseudo labels' localization errors.

The above results show an interesting complementary learning effect in self-training, that is learning from the pseudo labels' genuine classification features as well as GT labels' accurate localization features, which empirically explains the performance improvement of our method.

5.3. Improvement beyond Adversarial Methods

In this part, we show that previous adversarial-based methods can only address the style gap, but our approach can further reduce the content gap by self-training. We compare our approach with *Frcnn in the wild* [5], one of the representative feature adversarial learning methods. The results are shown in Table 5. Both methods are using MaskR-CNN detector with ResNet50 backbone, and trained/tested

Table 5: Improvement beyond Adversarial Methods.

Methods	*Car AP*
Baseline [non-adapt]	45.4
Frcnn in the wild [5]	52.5
Fine-Grained Domain Trans. (DT)	62.8
DT + Frcnn in the wild	63.1 (+0.3)
DT + ST-1st Iter	66.1 (+3.3)
DT + ST-2nd Iter	68.1 (+5.3)
DT + ST-3rd Iter	**69.8** (+7.0)

Table 6: Benefits of Fine Granularity in Domain Transfer.

Granularity	512^2 (coarse)	256^2	128^2 (fine)
Sim2City	42.4	44.3	**48.1 (+5.7)**
Kitti2City	40.5	40.0	**41.4 (+0.9)**
City2Foggy	30.0	30.6	**34.3 (+4.3)**

on full resolution images in SIM10K and Cityscapes.

As Table 5 shows, *Frcnn* approach can achieve 52.5% AP with +7.1% improvement over baseline (45.4%). By comparison, our fine-grained domain transfer achieves 62.8% (+17.4% gain), demonstrating the performance advantages of our fine-graiend DT.

Meanwhile, we can notice that, on top of our DT style translation (62.8%), adversarial learning method *Frcnn in the wild* can hardly bring any further performance gain (only +0.3%). By contrast, our self-training method could still continually improve performance and finally outperform DT by a large margin (+7.0%).

We hypothesize this is because *Frcnn* and our style transfer (DT) are both targeting at reducing the style gap. As DT has already reduced the style gap to a large extent in the pixel level, adversarial method *Frcnn* in feature level thus cannot yield further improvement. In contrast, the iterative self-training (ST) can further reduce the content gap by involving the real data into training, suggesting a new exploration space in improving UDA for detection.

6. Ablation Study for Design Modules

Effectiveness of Fine Granularity: We first show the benefits of the fine granularity design in the domain style transfer. Specifically, we trained three style translation models with coarse-to-fine granularities: 512^2, 256^2, 128^2. Their final adaptation results are shown in Table 6. As we can see, detector's adaptation performance consistently improves with finer granularity (*e.g.*, **+5.7%** in *Sim2City*, **+4.3%** in *City2Foggy*). The reason is that with finer granularity, the contexts and small objects are better maintained during the style translation. More visualizations could be found in supp. materials.

Effectiveness of Uncertainty-based Labeling: To demonstrate the effectiveness of our uncertainty-based labeling, we compare our method with common confidence-based thresholding methods (*CF*). Throughout our experiments, we use the fixed IoU threshold 0.6 for box uncertainty estimation, which is robust on all of our datasets.

Table 7: Benefits of Uncertain-based Pseudo Labeling.

Methods	CF0.5	CF0.6	CF0.7	Ours
Sim2City	51.4	51.9	51.6	**52.4 (+1.0)**
Kitti2City	44.6	44.4	44.5	**46.4 (+1.8)**
City2Foggy	38.2	38.0	38.0	**38.7 (+0.5)**

Table 8: Benefits of Imbalanced Sampling in Self-Training.

GT : Pseudo	0:4	1:3	2:2	**3:1**
Sim2City	46.2	47.0	48.2	**49.0 (+2.8)**
Kitti2City	40.8	41.0	41.8	**44.3 (+3.5)**
City2Foggy	32.5	35.1	35.6	**35.7 (+3.2)**

Table 7 quantifies the performance gain: In three adaptation scenarios, our uncertainty-based labeling could consistently achieve **+0.5%** to **+1.8%** mAP improvement than CF-based labeling with different thresholds.

Effectiveness of Imbalanced Sampling: As pseudo labels inevitably contain errors, imbalanced sampling by weighing more on the GT labels is designed to mitigate the pseudo label biases and improve the training performance.

The results in Table 8 verify our design: First, the GT and pseudo label sampling ratio at 0:4 (*i.e.*, no GT labels being sampled) gives the least performance for all settings. By sampling more GT data in each mini-batch, the adaptation performance consistently improves and achieves the best at 3:1 ratio, **+2.8%**$\sim$**3.5%** mAP improvement.

By detailed analysis, we find the imbalanced sampling mainly helps mitigate the label errors in the localization: Due to that pseudo labels' coordinates are usually deviated, involving more GT labels wtih accurate coordinates provides important localization supervision during training, and thus helps the overall performance. The detailed localization/classification study could be found below.

7. Conclusion

In this work, we propose SC-UDA: a style & content gaps aware UDA framework to address the unsupervised domain adaptation for object detection. Specifically, we conduct fine-grained domain transfer to reduce the style gap first and then launch our iterative self-training to reduce the content gaps. Optimizations including uncertainty-based pseudo labeling and imbalanced sampling are proposed to mitigate the pseudo label errors' influence. Experiments demonstrate the effectiveness of our framework, which outperforms previous SOTA by large margins in various adaptation scenarios.

References

[1] Kamyar Azizzadenesheli, Anqi Liu, Fanny Yang, and Animashree Anandkumar. Regularized learning for domain adaptation under label shifts. *arXiv preprint arXiv:1903.09734*, 2019.

[2] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[3] Olivier Chapelle and Alexander Zien. Semi-supervised classification by low density separation. In *AISTATS*, volume 2005, pages 57–64. Citeseer, 2005.

[4] Minmin Chen, Kilian Q Weinberger, and John Blitzer. Co-training for domain adaptation. In *Advances in neural information processing systems*, 2011.

[5] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3339–3348, 2018.

[6] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[7] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, pages 2960–2967, 2013.

[8] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014.

[9] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[10] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

[11] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[12] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[14] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016.

[15] Han-Kai Hsu, Wei-Chih Hung, Hung-Yu Tseng, Chun-Han Yao, Yi-Hsuan Tsai, Maneesh Singh, Ming-Hsuan Yang, Wayne Treible, Philip Saponaro, Yi Liu, et al. Progressive domain adaptation for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–5, 2019.

[16] 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? In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017.

[17] Seunghyeon Kim, Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6092–6101, 2019.

[18] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*, volume 3, page 2, 2013.

[19] Zachary C Lipton, Yu-Xiang Wang, and Alex Smola. Detecting and correcting for label shift with black box predictors. *arXiv preprint arXiv:1802.03916*, 2018.

[20] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016.

[21] David McClosky, Eugene Charniak, and Mark Johnson. Effective self-training for parsing. In *Proceedings of the main conference on human language technology conference of the North American Chapter of the Association of Computational Linguistics*, pages 152–159. Association for Computational Linguistics, 2006.

[22] Marc'aurelio Ranzato and et al. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007.

[23] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pages 102–118. Springer, 2016.

[24] Aruni RoyChowdhury, Prithvijit Chakrabarty, Ashish Singh, SouYoung Jin, Huaizu Jiang, Liangliang Cao, and Erik Learned-Miller. Automatic adaptation of object detectors to new domains using self-training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 780–790, 2019.

[25] Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, and Kate Saenko. Strong-weak distribution alignment for adaptive object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6956–6965, 2019.

[26] 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.

[27] Shuhan Tan, Xingchao Peng, and Kate Saenko. Generalized domain adaptation with covariate and label shift coalignment. *arXiv:1910.10320*, 2019.

[28] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.

[29] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017.

[30] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[31] Rongchang Xie, Fei Yu, Jiachao Wang, Yizhou Wang, and Li Zhang. Multi-level domain adaptive learning for cross-domain detection. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[32] Chang-Dong Xu, Xing-Ran Zhao, Xin Jin, and Xiu-Shen Wei. Exploring categorical regularization for domain adaptive object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11724–11733, 2020.

[33] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2020–2030, 2017.

[34] Han Zhao, Remi Tachet des Combes, Kun Zhang, and Geoffrey J Gordon. On learning invariant representation for domain adaptation. In *Proceedings of the IEEE international conference on machine learning*, 2019.

[35] Yangtao Zheng and et al. Cross-domain object detection through coarse-to-fine feature adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13766–13775, 2020.

[36] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[37] Xinge Zhu, Jiangmiao Pang, Ceyuan Yang, Jianping Shi, and Dahua Lin. Adapting object detectors via selective cross-domain alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 687–696, 2019.

[38] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5982–5991, 2019.

[39] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018.