Progressive Domain Adaptation for Object Detection

Han-Kai Hsu[1], Chun-Han Yao[1], Yi-Hsuan Tsai[2], Wei-Chih Hung[1],
Hung-Yu Tseng[1], Maneesh Singh[3], and Ming-Hsuan Yang[1,4]

[1]University of California, Merced [2]NEC Laboratories America [3]Verisk Analytics [4]Google

Abstract

Recent deep learning methods for object detection rely on a large amount of bounding box annotations. Collecting these annotations is laborious and costly, yet supervised models do not generalize well when testing on images from a different distribution. Domain adaptation provides a solution by adapting existing labels to the target testing data. However, a large gap between domains could make adaptation a challenging task, which leads to unstable training processes and sub-optimal results. In this paper, we propose to bridge the domain gap with an intermediate domain and progressively solve easier adaptation subtasks. This intermediate domain is constructed by translating the source images to mimic the ones in the target domain. To tackle the domain-shift problem, we adopt adversarial learning to align distributions at the feature level. In addition, a weighted task loss is applied to deal with unbalanced image quality in the intermediate domain. Experimental results show that our method performs favorably against the state-of-the-art method in terms of the performance on the target domain.

1. Introduction

Object detection is an important computer vision task aiming to localize and classify objects in images. Recent advancement in neural networks has brought significant improvement to the performance of object detection [9, 24, 21, 22, 23, 17]. However, such deep models usually require a large-scale annotated dataset for supervised learning and do not generalize well when the training and testing domains are different. For instance, domains can differ in scenes, weather, lighting conditions and camera settings. Such domain discrepancy or domain-shift can cause unfavorable model generalization issues. Although using additional training data from the target domain can improve the performance, collecting annotations is usually time-consuming and labor-intensive.

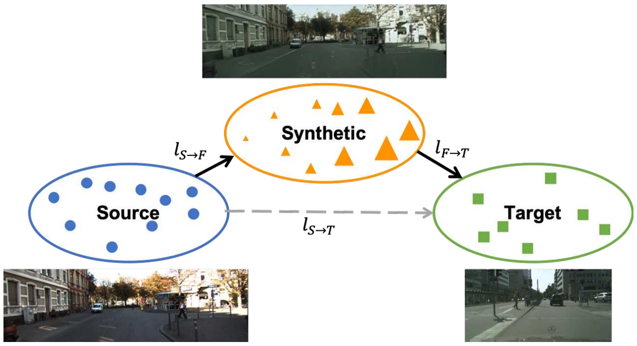

Figure 1. An illustration of our progressive adaptation method. Conventional domain adaptation aims to solve domain-shift problem from source to target domain, which is denoted as $l_{\mathbb{S}\to\mathbb{T}}$. We propose to bridge this gap with an intermediate synthetic domain that allows us to gradually solve separate subtasks with smaller gaps (shown as $l_{\mathbb{S}\to\mathbb{F}}$ and $l_{\mathbb{F}\to\mathbb{T}}$). In addition, we treat each image in the synthetic domain unequally based on its quality with respect to the target domain, where the size of the yellow triangles stand for their weights (i.e., the closer to target, the higher of the weight).

Unsupervised domain adaptation methods address the domain-shift problem without using ground truth labels in the target domain. Given the source domain annotations, the objective is to align source and target distributions in an unsupervised manner, so that the model can generalize to the target data without annotation effort. Numerous methods are developed in the context of image classification [32, 18, 19, 28, 10, 31, 7, 2], while fewer efforts have been made on more complicated tasks such as semantic segmentation [13, 29] and object detection [11, 3, 15]. Such domain adaptation tasks are quite challenging as there usually exists a significant gap between source and target domains.

In this paper, we aim to ease the effort of aligning different domains. Inspired by [10] which addresses the domain-shift problem via aligning intermediate feature representations, we utilize an intermediate domain that lies between source and target, and hence avoid direct mapping across two distributions with a significant gap. Specifically, the source images are first transformed by an image-to-image translation network [36] to have similar appearance as the target ones. We refer to the domain containing synthetic target images as the intermediate domain. We then construct an intermediate feature space by aligning the source and intermediate distributions, which is an easier task than aligning to the final targets. Once this intermediate domain is aligned, we use it as a bridge to further connect to the target domain. As a result, via the proposed progressive adaptation through the intermediate domain, the original alignment between source and target domains is decomposed into two subtasks that both solve an easier problem with a smaller domain gap.

During the alignment process, since the intermediate space is constructed in an unsupervised manner, one potential issue is that each synthetic target image may contribute unequally based on the quality of the translation. To reduce the outlier impact of the low-quality translated images, we propose a weighted version in our adaptation method, where the weight is determined based on the distance to the target distribution. That is, an image closer to the target domain should be considered a more important sample. In practice, we obtain the distance from the discriminator in the image translation model and incorporate it into the detection framework as a weight in the task loss.

We evaluate our method on various adaptation scenarios using numerous datasets, including KITTI [8], Cityscapes [4], Foggy Cityscapes [26] and BDD100k [35]. We conduct experiments on multiple real-world domain discrepancy cases, such as weather changes, camera differences and the adaptation to a large-scale dataset. With the proposed progressive adaptation, we show that our method performs favorably against the state-of-the-art algorithm in terms of accuracy in the target domain. The main contributions of the work are summarized as follows: 1) we introduce an intermediate domain in the proposed adaptation framework to achieve progressive feature alignment for object detection, 2) we develop a weighted task loss during domain alignment based on the importance of the samples in the intermediate domain, and 3) we conduct extensive adaptation experiments under various object detection scenarios and achieve state-of-the-art performance.

2. Related Work

Object Detection. Recently, state-of-the-art object detection methods are predominately based on the deep convolutional neural networks (CNNs). These methods can be categorized into region proposal-based and single-shot detectors, depending on the network forwarding pipelines. Region proposal-based methods [9, 24] perform prediction on a variable set of candidate regions. Fast R-CNN [9] applies selective search [33] to obtain region proposals, while Faster R-CNN [24] proposes to learn a Region Proposal Network (RPN) to accelerate the proposal generation process. To further reduce the computational need of proposal generation, single-shot approaches [21, 22, 23, 17] employ a fixed set of predefined anchor boxes as proposals and directly predict the category and offsets for each anchor box. Although these methods achieve state-of-the-art performance, such success hinges on the substantial amount of labeled training data which requires a high labor cost. Also, these methods can overfit on the training domain, which makes them difficult to generalize to many real-world scenarios. As a result, the vision community has recently started showing a great interest in employing domain adaptation techniques to object detection.

Domain Adaptation. Domain adaptation techniques aim to tackle domain-shift between the source and target domains with unlabeled or weakly labeled images in the target domain. In recent years, adversarial learning has played a critical role in domain adaptation methods. Since the emergence of the Domain Adversarial Neural Network (DANN) [7], numerous works [2, 31, 3] have been proposed to utilize adversarial learning for the feature distribution alignment between two domains. Furthermore, several methods attempt to perform alignment in the pixel space, based on the unpaired image-to-image translation approaches [36]. For image classification, PixelDA [1] synthesizes additional images in the target domain by learning one-to-many mapping. For semantic segmentation, CyCADA [12] and AugGAN [14] both design a CycleGAN [36]-like network to transform images from the source domain to the target one. The transformed images are then treated as simulated training images for the target domain with the same label mapped from the source domain. Instead of performing alignment in the feature/pixel space, Tsai *et al*. [29, 30] adopt adversarial learning in the structured output space for solving domain adaptation on semantic segmentation.

To address domain adaptation for object detection in a weakly-supervised manner, LSDA [11] finetunes a fully-supervised classification model for object detection with limited bounding box resources. Alternatively, Naoto *et al*. [15] train the network with synthetic data and finetune it with pseudo-labels in the target domain. In an unsupervised domain adaptation setting, Chen *et al*. [3] propose to close the domain gap on both image level and instance level via adversarial learning. To emphasize on matching local features, Zhu *et al*. [37] mines discriminative regions for alignment, while Saito *et al*. [25] focus on aligning local receptive fields at low-level features along with weak alignment on global regions. On the other hand, Kim *et al*. [16] utilize image translation network to generate multiple domains and use a multi-domain discriminator to adapt all domains simultaneously, but this method does not consider the distance between the generated ones and the final target.

In this work, we observe that simply applying image

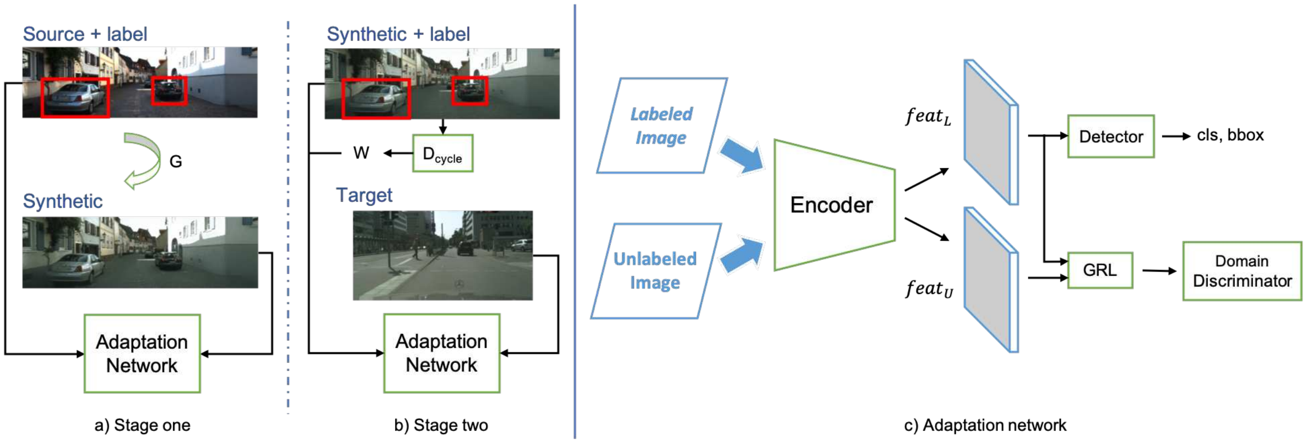

Figure 2. The proposed progressive adaptation framework. The algorithm includes two stages of adaptation as shown in a) and b). In a), we first transform source images to generate synthetic ones by using the generator G learned via CycleGAN [36]. Afterward, we use the labeled source domain and perform first stage adaptation to the synthetic domain. Then in b), our model applies a second stage adaptation which takes the synthetic domain with labels inherited from the source and aligns the synthetic domain features with the target distribution. In addition, a weight w is obtained from the discriminator D_{cycle} in CycleGAN to balance the synthetic image qualities in the detection loss. The overall structure of our adaptation network is shown in c). Labeled and unlabeled images are both passed through the encoder network E to extract CNN features $feat_L$ and $feat_U$. We then use them to: 1) learn supervised object detection with the detector network from $feat_L$, and 2) forward both features to GRL and a domain discriminator, learning domain-invariant features in an adversarial manner.

translation without knowing the distance between each generated sample and the target domain may result in less effective adaptation. To handle this issue, we first introduce an intermediate domain to reduce the effort of mapping two significantly different distributions and then adopt a two-stage alignment strategy with sample weights to account for the sample quality.

3. Progressive Domain Adaptation

We propose to decompose the domain adaptation problem into two smaller subtasks, bridged by a synthetic domain sitting in between the source and target distribution. Taking advantage of this synthetic domain, we adopt a progressive adaptation strategy which closes the gap gradually through the intermediate domain. We denote the source, synthetic, and target domains as $\mathbb{S}$, $\mathbb{F}$ and $\mathbb{T}$, respectively. The conventional adaptation from a labeled domain $\mathbb{S}$ to the unlabeled domain $\mathbb{T}$ is denoted as $\mathbb{S} \rightarrow \mathbb{T}$, while the proposed adaptation subtasks are expressed as $\mathbb{S} \rightarrow \mathbb{F}$ and $\mathbb{F} \rightarrow \mathbb{T}$. An overview of our progressive adaptation framework is shown in Figure 2. We discuss the details of the proposed adaptation network and progressive learning in the following sections.

3.1. Adaptation in the Feature Space

In order to align distributions in the feature space, we propose a deep model which consists of two components; a detection network and a discriminator network for feature alignment via adversarial learning.

Detection Network. We adopt the Faster R-CNN [24] framework for the object detection task, where the detector has a base encoder network E to extract image features. Given an image $\mathbf{I}$, the feature map $E(\mathbf{I})$ is extracted and then fed into two branches: Region Proposal Network (RPN) and Region of Interest (ROI) classifier. We refer to these branches as the detector, which is shown in Figure 2. To train the detection network, the loss function $\mathcal{L}_{det}$ is defined as:

$$\mathcal{L}_{det}(E(\mathbf{I})) = \mathcal{L}_{rpn} + \mathcal{L}_{cls} + \mathcal{L}_{reg}, \tag{1}$$

where $\mathcal{L}_{rpn}$, $\mathcal{L}_{cls}$, and $\mathcal{L}_{reg}$ are the loss functions for the RPN, classifier and bounding box regression, respectively. We omit the details of the RPN and ROI classifier here as we focus on solving the domain-shift We omit the details of the RPN and ROI classifier here as we focus on solving the domain-shift problem. The readers are encouraged to refer to the original paper [24] for further details.

Domain Discriminator. To align the distributions across two domains, we append a domain discriminator D after the encoder E. The main objective of this branch is to discriminate whether the feature $E(\mathbf{I})$ is from the source or the target domain. Through this discriminator, the probability of each pixel belonging to the target domain is obtained as $\mathbf{P} = D(E(\mathbf{I})) \in \mathbb{R}^{H \times W}$. We then apply a binary cross-entropy loss to $\mathbf{P}$ based on the domain label d of the input image, where images from the source distributions are given the label $d = 0$ and the target images receive label $d = 1$.

The discriminator loss function $\mathcal{L}_{disc}$ can be formulated as:

$$\mathcal{L}_{disc}(E(\mathbf{I})) = -\sum_{h,w} d \log \mathbf{P}^{(h,w)} \\ + (1-d)\log(1-\mathbf{P}^{(h,w)}). \quad (2)$$

Adversarial Learning. Adversarial learning is achieved using the Gradient Reverse Layer (GRL) proposed in [6] to learn the domain-invariant feature $E(\mathbf{I})$. GRL is placed in between the discriminator and the detection network, only affecting the gradient computation in the backward pass. During backpropagation, GRL negates the gradients that flow through. As a result, the encoder E receives gradients that force it to update in an opposite direction which maximizes the discriminator loss. This allows E to produce features that fools the discriminator D while D tries to distinguish the domain of the features. For the adaptation task $\mathbb{S} \to \mathbb{T}$, given source images $\mathbf{I}_{\mathbb{S}}$ and target images $\mathbf{I}_{\mathbb{T}}$, the overall min-max loss function of the adaptive detection model is defined as the following:

$$\min_{E}\max_{D} \mathcal{L}(\mathbf{I}_{\mathbb{S}}, \mathbf{I}_{\mathbb{T}}) = \mathcal{L}_{det}(\mathbf{I}_{\mathbb{S}}) + \lambda_{disc}\big[\mathcal{L}_{disc}(E(\mathbf{I}_{\mathbb{S}})) \\ + \mathcal{L}_{disc}(E(\mathbf{I}_{\mathbb{T}}))\big], \quad (3)$$

where λ_{disc} is a weight applied to the discriminator loss that balances the loss.

3.2. Progressive Adaptation

Aligning feature distributions between two distant domains is challenging, and hence we introduce an intermediate feature space to make the adaptation task easier. That is, instead of directly solving the gap between the source and the target domains, we progressively perform adaptation to the target domain bridged by the intermediate domain.

Intermediate Domain. The intermediate domain is constructed from the source domain images to synthesize the target distributions on the pixel-level. We apply an image-to-image translation network, CycleGAN [36] to learn a function that maps the source domain images to the target ones, and vice versa. Since ground truth labels are only available in the source domain, we only consider the translation from source images to the target domain (i.e., synthetic target images) after training CycleGAN.

Synthetic target images have been utilized to assist with domain adaptation tasks [1, 14, 15] as additionally augmented target training data. Different from these approaches, we define this set of synthetic images as an individual domain $\mathbb{F}$ to connect the labeled domain $\mathbb{S}$ with the unlabled domain $\mathbb{T}$ via adversarial learning. One motivation behind this is that the similarity between source domain $\mathbb{S}$ and $\mathbb{F}$ is the image content, only diverging in the visual appearances, while $\mathbb{F}$ and the target domain $\mathbb{T}$ are different in image details but have similar distributions on the pixel-level. Consequently, this synthetic domain "sits" in between the source and target domains and thus can help reduce the adaptation difficulty of a large domain gap between $\mathbb{S}$ and $\mathbb{T}$. Figure 3 is one example of feature space visualization using the KITTI and Cityscapes datasets. This figure shows a distribution plot by mapping the features from $E(\mathbf{I})$ to a low dimensional 2-D space via t-SNE [34]. The plot demonstrates that in the feature space, the synthetic domain $\mathbb{F}$ (blue) is located in between the KITTI (red) and Cityscapes (green) distributions.

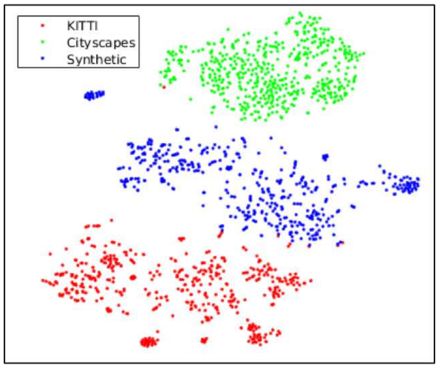

Figure 3. Visualization of the feature distributions via t-SNE [34], showing that our synthetic images serve as an intermediate feature space between the source and target distributions. Each dot represents one image feature extracted from E. We take 500 images from the Cityscapes validation set and 500 from the KITTI training set for comparison.

Adaptation Process. Our domain adaptation network involves obtaining knowledge from a labeled source domain $\mathbb{S}$ then map that knowledge to an unlabeled target domain $\mathbb{T}$ by aligning the two distributions, solving the adaptation task $\mathbb{S} \to \mathbb{T}$, i.e., via (3) in this paper. To take advantage of the intermediate feature space during alignment, our algorithm decomposes the problem into two stages: $\mathbb{S} \to \mathbb{F}$ and $\mathbb{F} \to \mathbb{T}$, as shown in Figure 2 a) and b). At the first stage, we use $\mathbb{S}$ as the labeled domain, adapting to $\mathbb{F}$ without labels. Due to the underlying similarity between $\mathbb{S}$ and $\mathbb{F}$ in image contents, the network focuses on aligning the feature distributions with respect to the appearance difference on the pixel-level. After aligning pixel discrepancies between $\mathbb{S}$ and $\mathbb{F}$, we take $\mathbb{F}$ as the source domain for supervision and adapts to $\mathbb{T}$ as stage two in the proposed method. During this step, the model can take advantage of the appearance-invariant features from the first step and focus on adapting the object and context distributions. In summary, the proposed progressive learning separates the adaptation task into two subtasks and pays more attention to individual discrepancies during each adaptation stage.

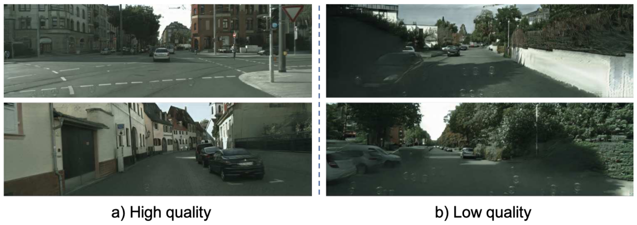

Figure 4. Image quality examples from the KITTI dataset synthesized to be in the Cityscapes domain. a) shows the ones that are translated with better quality. Images in b) contain artifacts and fail to preserve details of the car, almost blend into the background.

Weighted Supervision. We observe that the quality of synthetic images differs in a wide range. For instance, some images fail to preserve details of objects or contain artifacts when translated, and these failure cases may have a larger distance to the target distribution (see Figure 4 for an example). This phenomenon can be also visualized in the feature space in Figure 3, where some blue dots are far away from both the source and target domains.

As a result, when performing supervised detection learning on $\mathbb{F}$ during $\mathbb{F} \rightarrow \mathbb{T}$, these defects may cause confusions to our detection model, leading to false feature alignment across domains. To alleviate this problem, we propose an importance weighting strategy for synthetic samples based on their distances to the target distribution. Specifically, synthetic outliers that are further away from the target distribution will receive less attention than the ones that are closer to the target domain. We obtain the weights by taking the predicted output scores from the target domain discriminator D_{cycle}. This discriminator is trained to differentiate between the source and target images with respect to the target distribution, in which the optimal discriminator is obtained with:

$$D^*_{cycle}(\mathbf{I}) = \frac{p_{\mathbb{T}}(\mathbf{I})}{p_{\mathbb{S}}(\mathbf{I}) + p_{\mathbb{T}}(\mathbf{I})}, \tag{4}$$

where $\mathbf{I}$ is the synthetic target image generated via CycleGAN, and $p_{\mathbb{T}}(\mathbf{I})$ and $p_{\mathbb{S}}(\mathbf{I})$ are the probability of $\mathbf{I}$ belonging to the source and the target domain, respectively. Here, the higher score of $D_{cycle}(\mathbf{I})$ represents a closer distribution to the target domain, thus providing a higher weight. On the other hand, lower quality images which are further away from the target domain will be treated as outliers and receive a lower weight. For each image $\mathbf{I}$, the importance weight is defined as:

$$w(\mathbf{I}) = \begin{cases} D_{cycle}(\mathbf{I}), & \text{if } \mathbf{I} \in \mathbb{F} \\ 1, & \text{otherwise.} \end{cases} \tag{5}$$

We then apply this weight to the detection loss function in (1) when learning from synthetic images with labels during the second stage. Thus, the final weighted objective function given images $\mathbf{I}_{\mathbb{F}}$ and $\mathbf{I}_{\mathbb{T}}$ is re-formulated based on (3) as:

$$\min_E \max_D \mathcal{L}(\mathbf{I}_{\mathbb{F}}, \mathbf{I}_{\mathbb{T}}) = w(\mathbf{I}_{\mathbb{F}})\mathcal{L}_{det}(\mathbf{I}_{\mathbb{F}}) + \lambda_{disc}\big[\mathcal{L}_{disc}(E(\mathbf{I}_{\mathbb{F}})) + \mathcal{L}_{disc}(E(\mathbf{I}_{\mathbb{T}}))\big]. \tag{6}$$

4. Experimental Results

In this section, we validate our method by evaluating the performance in three real-world scenarios that result in different domain discrepancies: 1) *cross-camera adaptation*, 2) *weather adaptation*, and 3) *adaptation to large-scale dataset*. Figure 5 shows examples of the detection results from the three tasks before and after applying our domain adaptation method.

For each adaptation scenario, we show a baseline Faster R-CNN result trained on the source data without applying domain adaptation, and a supervised model trained fully on the target domain data (oracle) to illustrate the existing gap between domains. Then we train the proposed model on the selected source and target domain to demonstrate the effectiveness of the proposed method. We also conduct ablation study to analyze the effectiveness of individual proposed components. More results will be available in the supplementary material. All the source code and trained models will be made available to the public[1].

4.1. Implementation Details

Adaptation Network. In our experiments, we adopt VGG16 [27] as the backbone for the Faster R-CNN [24] detection network, following the setting in [3]. We design the discriminator network D using 4 convolution layers with filters of size 3×3. The first 3 convolution layers have 64 channels, each followed by a leaky ReLU [20] with α set to 0.2. The final domain classification layer has 1 channel that outputs the binary label prediction. Our synthetic domain is generated by training CycleGAN [36] on the source and target domain images.

Training Details. Before applying the proposed adaptation method, we pre-train the detection network using source domain images with ImageNet [5] pre-trained weights. When training the adaptation model, we use all available annotations in the source domain including the training and validation set. We optimize the network using Stochastic Gradient Descent (SGD) with a learning rate of 0.001, weight decay of 0.0005 and momentum of 0.9. We use $\lambda_{disc} = 0.1$ based on a validation set to balance the discriminator loss with the detection loss. Batch size is 1 during training. The proposed method is implemented with Pytorch and the networks are trained using one GTX 1080 Ti GPU with 12 GB memory.

[1] https://github.com/kevinhkhsu/DA_detection

4.2. Datasets

KITTI. The KITTI dataset [8] contains images taken while driving in cities, highways, and rural areas. There are a total of 7,481 images in the training set. The dataset is only used as the source domain in the proposed experiments, and we utilize the full training set.

Cityscapes. The Cityscapes dataset [4] is a collection of images with city street scenarios. It includes instance segmentation annotation which we transform into bounding boxes for our experiments. It contains 2,975 training images and 500 validation images. We use Cityscapes with the KITTI dataset in Section 4.3 to evaluate the cross camera adaptation and compare our results with the state-of-the-art method.

Foggy Cityscapes. As self-explanatory by the name, the Foggy Cityscapes dataset [26] is built upon the images in the Cityscapes dataset [4]. This dataset simulates the foggy weather using depth maps provided in Cityscapes with three levels of foggy weather. The simulation process can be found in the original paper [26]. Section 4.4 shows the experiments conducted on this simulated dataset for cross weather adaptation.

BDD100k. The BDD100k dataset [35] consists of 100k images which are split into training, validation, and testing sets. There are 70k training images and 10k validation images with available annotations. This dataset includes different interesting attributes; there are 6 types of weather, 6 different scenes, 3 categories for the time of day and 10 object categories with bounding box annotation. In our experiment, we extract a subset of the BDD100k with images labeled as $daytime$. It includes 36,728 training and 5,258 validation images. We use this subset to demonstrate the adaptation from a smaller dataset, Cityscapes, to a large-scale dataset using the proposed method in Section 4.5.

4.3. Cross Camera Adaptation

Different datasets exhibit distinct characteristics such as scenes, objects, and viewpoint. In addition, the underlying camera settings and mechanisms can also lead to critical differences in visual appearance as well as the image quality. These discrepancies are where the domain-shift takes place. In this experiment, we show the adaptation between images taken from different cameras and with distinctive content differences. The KITTI [8] and Cityscapes [4] datasets are used as source and target respectively to conduct the cross camera adaptation experiment. During training, all data in the KITTI training set and raw training images from Cityscapes dataset is used and further evaluated on the Cityscapes validation set. In Table 1, we show experimental results evaluated on the *car* class in terms of the average precision (AP). Compared to the state-of-the-art method [3] that learns to adapt in the feature space, our baseline denoted as "Ours (w/o synthetic)" matches their performance using our own implementation.

Table 1. Cross camera adaptation using KITTI and Cityscapes datasets. The results show the average precision (AP) of the *car* class shared between the two domains.

KITTI → Cityscapes	
Method	AP
Faster R-CNN	28.8
FRCNN in the wild [3]	38.5
Ours (w/o synthetic)	38.2
Ours (synthetic augment)	40.6
Ours (progressive)	**43.9**
Oracle	55.8

Table 2. Analysis of our weighted task loss compared to several arbitrary weight settings. We show that by setting each image weight with respect to the distance from the target distribution improves the model performance.

KITTI → Cityscapes						
weight	0.8	0.9	1	1.1	1.2	Ours
AP	39.8	42.8	42.2	41.1	42.6	**43.9**

In order to validate our method, we also conduct ablation studies using several settings. First, we demonstrate the benefit of utilizing information from the synthetic domain. When we directly augment synthetic data in the training set and include them in the source domain to perform feature-level adaptation, denoted as "Ours (synthetic augment)", there is a 2.1% performance gain compared to [3]. In the proposed method, by adopting our progressive training scheme with the importance weights,we show that our model further improves the AP by 5.4%. In addition, we present the advantage of our weighted task loss in balancing the uneven quality of synthetic images. In Table 2, we show the analysis for using different fixed weights and our importance weighting method. Our method dynamically determines the weight of each image[2] based on the distance from the target distribution. Compared to the one without using any weight (i.e., weight is equal to 1), our importance weight improves the AP by 1.7% and performs better than others that use fixed weights. Overall, we show that our model can reduce the domain-shift problem caused by the camera along with other content differences across two distinct datasets and achieves state-of-the-art performance.

4.4. Weather Adaptation

Under real-world scenarios, supervised object detection models can be applied in different weather conditions where

[2] In this case, the averaged weight obtained from the discriminator is around 0.9.

Table 3. Weather adaptation focusing on clear weather to foggy weather using the Cityscapes and Foggy Cityscapes datasets respectively. Performance is evaluated using the mean average precision (mAP) across 8 classes.

Cityscapes → Foggy Cityscapes									
Method	person	rider	car	truck	bus	train	motorcycle	bicycle	mAP
Faster R-CNN	23.3	29.4	36.9	7.1	17.9	2.4	13.9	25.7	19.6
FRCNN in the wild [3]	25.0	31.0	40.5	22.1	35.3	20.2	20.0	27.1	27.6
Diversify & Match [16]	30.8	40.5	44.3	27.2	38.4	**34.5**	28.4	32.2	34.6
Strong-Weak Align [25]	29.9	42.3	43.5	24.5	36.2	32.6	**30.0**	35.3	34.3
Selective Align [37]	33.5	38.0	48.5	**26.5**	39.0	23.3	28.0	33.6	33.8
Ours (w/o synthetic)	30.2	37.9	46.1	14.7	26.9	7.0	20.8	31.5	26.9
Ours (synthetic augment)	**36.6**	45.3	**55.0**	24.2	43.9	18.5	28.4	**37.1**	36.1
Ours (progressive)	36.0	**45.5**	54.4	24.3	**44.1**	25.8	29.1	35.9	**36.9**
Oracle	37.8	48.4	58.8	25.2	53.3	15.8	35.4	39.0	39.2

they may not have sufficient knowledge of. However, it is difficult to obtain a large number of annotations in every weather condition for the models to learn. This section studies the weather adaptation from clear weather to a foggy environment. The Cityscapes dataset [4] and the Foggy Cityscapes dataset [26] are used as the source domain and the target domain, respectively.

Table 3 shows that our method reduces the domain gap across weather conditions and performs favorably against the state-of-the-art methods [3, 25, 37, 16]. When synthetic images are introduced during our progressive adaptation, there is a 10% improvement in mAP compared to the baseline method. We note that the target Foggy Cityscapes dataset fundamentally contain same images as the source Cityscapes dataset, but with synthesized fogs. Thus, the synthetic target domain $\mathbb{F}$ via image translation is already closely distributed to the target domain and inherits informative labels for the network to learn. Given such information learned from the synthetic domain, both our method and the synthetic augmented one climbs closely to the oracle result. Although the synthetic domain lies close to the target distribution, we show in the results that our progressive training can still assist the adaptation process, improving performance and at the same time generalizing well to different categories. To sum up, this experiment not only demonstrates the adaptation to a foggy weather condition but also shows the capability of using synthetic images to facilitate the distribution alignment process.

4.5. Adaptation to Large-scale Dataset

Digital cameras have developed quickly over the years and collecting a large number of images is not a difficult task in the modern world. However, labeling the collected images is a major issue when it comes to building a dataset for supervised learning methods. In this experiment, we examine the adaptation from a relatively smaller dataset to a large unlabeled domain containing distinct attributes. We show that our method can harvest more from existing resources and adapt them to complicated environments. To this end, we use the Cityscapes [4] and BDD100k [35] datasets as the source and target domains, respectively. We choose a subset of the BDD100k dataset annotated as $daytime$ to be our target domain and consider the city scene as the adaptation factor, since there only exists daytime data in the Cityscapes dataset.

From the baseline and oracle results shown in Table 4, we can observe the difficulty and the significant performance gap between the source and target domains. Without using the synthetic data, the network has a harder time in adapting to a much diverse dataset with only 0.4% improvement after directly aligning the source and target domains using the method in [3]. When synthetic data is introduced to the source training set, the model learns to generalize better to the target domain and increases the performance by 2.5%. Finally, our method progressively adapts to the target domain by utilizing the intermediate feature space and receives an 3.1% gain in mAP compared to the baseline method [3]. We show in this experiment that our progressive adaptation can squeeze more juice out of the available knowledge and generalize better to a diverse environment, which is a critical issue in real-world applications. Qualitative results are shown in Figure 5 and more results are provided in the supplementary material.

5. Conclusions

In this paper, we propose a progressive adaptation method that bridges the domain gap using an intermediate domain, decomposing a more difficult task into two easier subtasks with a smaller gap. We obtain the intermediate domain by transforming the source images to target ones. Using this domain, our method progressively solves the adaptation subtasks by first adapting from source to the intermediate domain and then finally to the target domain. In addition, we introduce a weighted loss during stage two of our

Table 4. Adaptation from a smaller Cityscapes dataset to a larger and diverse BDD100k dataset. A subset of the BDD100k dataset labeled as $daytime$ is used as the target domain. We evaluate the mean average precision (mAP) of 10 classes which are available across the two domains.

Cityscapes → BDD100k daytime											
Method	bike	bus	car	motor	person	rider	light	sign	train	truck	mAP
Faster R-CNN	19.4	20.4	49.0	17.2	31.1	26.5	11.5	14.6	0	18.9	20.8
Ours (w/o synthetic)	20.4	20.2	49.2	**16.6**	32.1	27.8	11.9	14.9	0	19.2	21.2
Ours (synthetic augment)	23.1	**25.3**	**51.9**	15.7	36.0	31.6	12.7	20.8	0	**20.2**	23.7
Ours (progressive)	**25.3**	23.7	51.8	16.1	**37.6**	**32.9**	**14.0**	**22.2**	0	19.3	**24.3**
Oracle	36.2	58.2	62.3	36.1	46.2	43.6	43.5	49.7	0	57.6	43.3

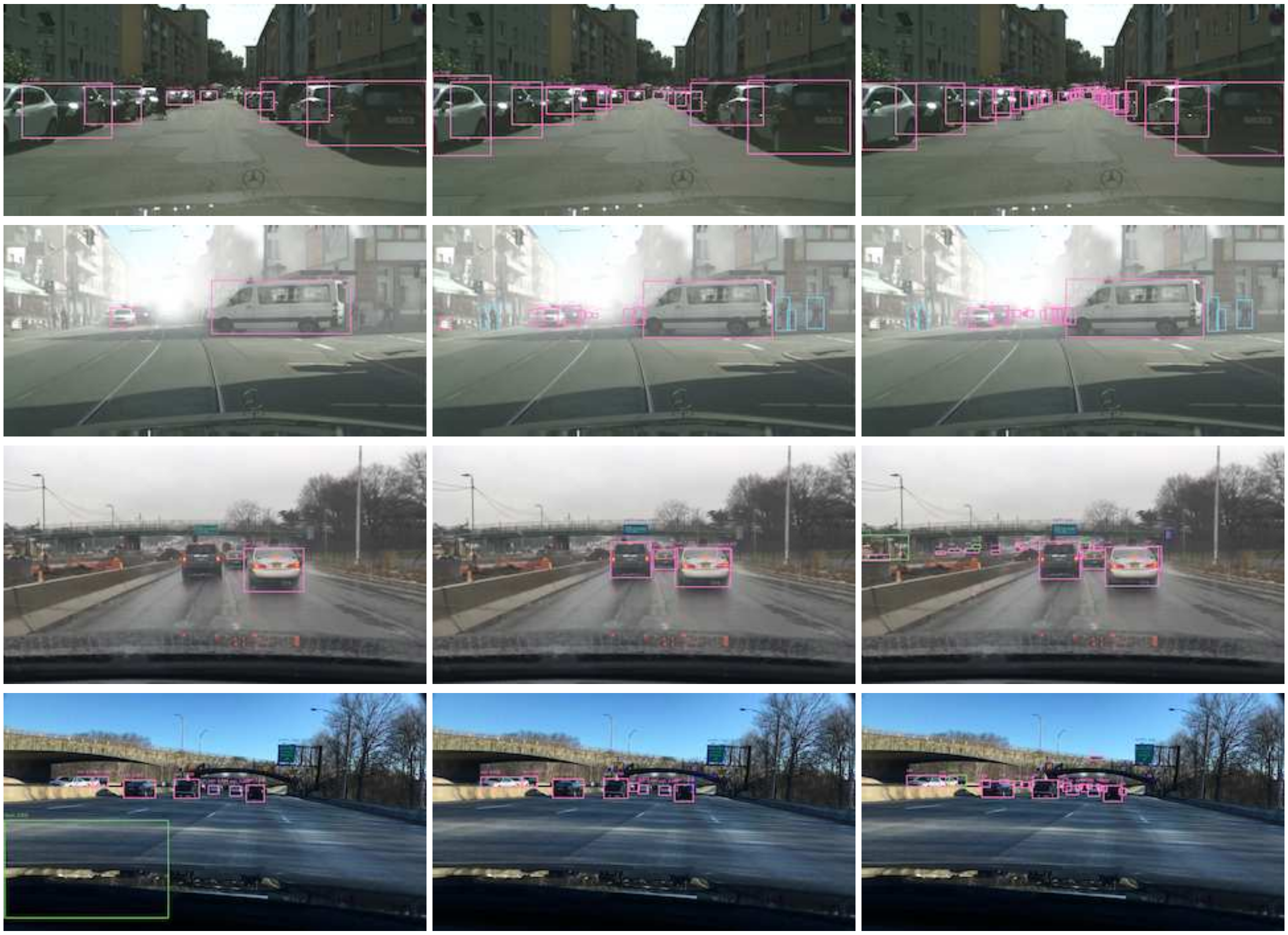

Figure 5. Examples of the detection results from our three adaptation tasks. The first two rows are the tasks KITTI → Cityscapes and Cityscapes → Foggy Cityscapes respectively, while the last two rows are the task Cityscapes → BDD100k. We show the detection results on the target domain before and after applying our adaptation method as well as the ground truth labels.

method to balance different image qualities in the intermediate domain. Experimental results show that our method performs favorably against the state-of-the-art method and can further reduce the domain discrepancy under various scenarios, such as the cross-camera case, weather condition, and adaption to a large-scale dataset.

Acknowledgement. This work is supported in part by the NSF CAREER Grant 1149783, gifts from Adobe, Verisk, and NEC.

References

[1] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, 2017. 2, 4

[2] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *NIPS*, 2016. 1, 2

[3] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. V. Gool. Domain adaptive faster r-cnn for object detection in the wild. In *CVPR*, 2018. 1, 2, 5, 6, 7

[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 2, 6, 7

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[6] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015. 4

[7] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016. 1, 2

[8] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 2, 6

[9] R. Girshick. Fast r-cnn. In *ICCV*, 2015. 1, 2

[10] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *ICCV*, 2011. 1

[11] J. Hoffman, S. Guadarrama, E. S. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, and K. Saenko. Lsda: Large scale detection through adaptation. In *NIPS*, 2014. 1, 2

[12] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018. 2

[13] J. Hoffman, D. Wang, F. Yu, and T. Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *CoRR*, abs/1612.02649, 2016. 1

[14] S.-W. Huang, C.-T. Lin, S.-P. Chen, Y.-Y. Wu, P.-H. Hsu, and S.-H. Lai. Auggan: Cross domain adaptation with gan-based data augmentation. In *ECCV*, 2018. 2, 4

[15] N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *CVPR*, 2018. 1, 2, 4

[16] T. Kim, M. Jeong, S. Kim, S. Choi, and C. Kim. Diversify and match: A domain adaptive representation learning paradigm for object detection. In *CVPR*, 2019. 2, 7

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. 1, 2

[18] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015. 1

[19] M. Long, J. Wang, and M. I. Jordan. Deep transfer learning with joint adaptation networks. In *ICML*, 2017. 1

[20] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013. 5

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *CVPR*, 2016. 1, 2

[22] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. *CVPR*, 2017. 1, 2

[23] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. 1, 2

[24] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *PAMI*, 2017. 1, 2, 3, 5

[25] K. Saito, Y. Ushiku, T. Harada, and K. Saenko. Strong-weak distribution alignment for adaptive object detection. In *CVPR*, 2019. 2, 7

[26] C. Sakaridis, D. Dai, and L. V. Gool. Semantic foggy scene understanding with synthetic data. *IJCV*, 2018. 2, 6, 7

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 5

[28] B. Sun and K. Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV Workshops*, 2016. 1

[29] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018. 1, 2

[30] Y.-H. Tsai, K. Sohn, S. Schulter, and M. Chandraker. Domain adaptation for structured output via discriminative patch representations. *CoRR*, abs/1901.05427, 2019. 2

[31] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017. 1, 2

[32] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014. 1

[33] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 2013. 2

[34] L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *JMLR*, 2008. 4

[35] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *CoRR*, abs/1805.04687, 2018. 2, 6, 7

[36] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2, 3, 4, 5

[37] X. Zhu, J. Pang, C. Yang, J. Shi, and D. Lin. Adapting object detectors via selective cross-domain alignment. In *CVPR*, 2019. 2, 7