# Conditional Generative Adversarial Network for Structured Domain Adaptation

Weixiang Hong
Nanyang Technological University
weixiang.hong@outlook.com

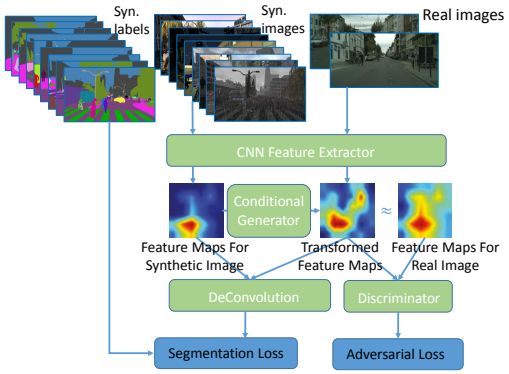
Zhenzhen Wang
Nanyang Technological University
zwang033@e.ntu.edu.sg

Ming Yang
Horizon Robotics, Inc.
ming.yang@horizon-robotics.com

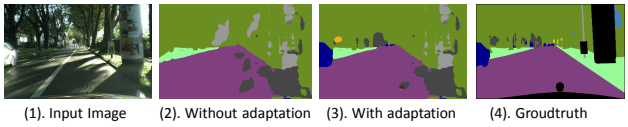
Junsong Yuan
State University of New York at Buffalo
jsyuan@buffalo.edu

## Abstract

*In recent years, deep neural nets have triumphed over many computer vision problems, including semantic segmentation, which is a critical task in emerging autonomous driving and medical image diagnostics applications. In general, training deep neural nets requires a humongous amount of labeled data, which is laborious and costly to collect and annotate. Recent advances in computer graphics shed light on utilizing photo-realistic synthetic data with computer generated annotations to train neural nets. Nevertheless, the domain mismatch between real images and synthetic ones is the major challenge against harnessing the generated data and labels. In this paper, we propose a principled way to conduct structured domain adaption for semantic segmentation,* i.e.*, integrating GAN into the FCN framework to mitigate the gap between source and target domains. Specifically, we learn a conditional generator to transform features of synthetic images to real-image like features, and a discriminator to distinguish them. For each training batch, the conditional generator and the discriminator compete against each other so that the generator learns to produce real-image like features to fool the discriminator; afterwards, the FCN parameters are updated to accommodate the changes of GAN. In experiments, without using labels of real image data, our method significantly outperforms the baselines as well as state-of-the-art methods by* $12\% \sim 20\%$ *mean IoU on the Cityscapes dataset.*




(a) Synthetic images are easy to collect and annotate, yet the semantic segmentation model naively trained on synthetic data may not generalize well to real images. In this work, we introduce a conditional GAN model to close the gap between the representations of synthetic images to those of real images, thus improve the semantic segmentation performance without laborious annotations on real image data.




(b) Our GAN-based domain adaptation boosts the semantic segmentation performance.

Figure 1: (a) The motivation and overview. (b) An illustrative example.

## 1. Introduction

Deep neural networks have dominated many vision tasks such as image recognition [38], object detection [17] and semantic segmentation [26]. These state-of-the-art results are generally achieved by learning very deep networks on large-scale, high-quality, and thoroughly labeled datasets, such as the ImageNet [11], COCO [25], Pascal VOC [13] and Cityscapes [10], *etc*. Nevertheless, building such datasets manually is expensive and not scalable. For instance, annotating pixel-level semantic segmentation in autonomous driving or medical image analysis requires intensive labor and certain expertise. Therefore, it is very appealing to ex-

plore a scalable alternative, *i.e*., the use of synthetic data for neural network training. Recent advances in computer graphics have stimulated the interests to train deep learning models on photo-realistic synthetic data with computer-generated annotations, and apply them in real-world settings [12, 29, 47, 37, 41, 48, 20, 19, 32, 22, 34, 33].

In this work, we aim to train semantic segmentation models for real urban scene parsing without using any manual annotation. The significance of this problem is mainly due to three reasons: 1) Autonomous driving has become such a *hot* theme in both academia and industry, where semantic segmentation is one of the essential techniques [16, 10] in understanding complex inner-city traffic scenes; 2) Training deep neural networks for automatic semantic segmentation requires massive amount of high-quality annotated imagery in order to generalize well to unlimited unseen scenes. Compared with image recognition and object detection, annotating pixel-level training data for semantic segmentation is a much more time-consuming task for human. For example, Cordts *et al*. reports that the annotation and quality control take more than 1.5 hours for a *single* image of the Cityscapes dataset [10]. In contrast, annotating a synthetic image takes only 7 seconds on average through a computer game [34]; 3) Besides, it requires dedicated equipments and takes months, if not years, to collect imagery that covers a large number of diverse urban scenes in different countries, seasons, and lighting conditions, *etc*. Therefore, it is of great practical and theoretical interests to explore the feasibility conducting semantic urban scene segmentation without manual labeling.

Formally, our goal is to learn neural networks for semantic segmentation using synthetic images with *generated* annotations and real images *without* annotations in the training phase, and then we expect the learned model generalizes well to real images in the testing phase. By harnessing photo-realistic simulation of urban environments such as Grand Thief Auto (GTA), practically an unlimited amount of synthetic scene images can be generated for training deep learning models [35, 34, 33]. However, the latest literature findings reveal that a gap between distributions of synthetic and real data does exist, yet the deep features may reduce, but not remove, the cross-domain distribution discrepancy [44]. Deep neural nets naively trained on synthetic data do not readily generalize to real images due to the domain mismatch between the source domain (synthetic) and the target domain (real). This problem almost fits into the setting of unsupervised domain adaptation, except that our goal is to learn pixel-wise labeling classifier, while unsupervised domain adaptation mainly concerns with classification and regression problem [30, 31]. In view of these, we refer our problem as a structured domain adaptation [43].

Generally speaking, unsupervised domain adaptation is a very challenging problem, *i.e*., learning a discriminative classifier in the presence of a shift between training and testing data distribution, where target domain data are completely unlabeled. To tackle the discrepancy between source and target domains, previous work [15, 40, 27, 5] typically assume that there exists a cross-domain feature space, so that the rich labeled data in the source domain can be leveraged to train effective classifiers in the target domain. However, it is practically infeasible to determine whether the classifier learned on cross-domain features generalizes well to the target domain beforehand. Moreover, different classifiers could be indeed necessary in some cases [3]. In contrast to prior arts, the semantic segmentation is a highly structured prediction problem. Can we still achieve reasonable domain adaptation by following the above assumption? The prior attempts [43] indicate that learning a decision function for structured prediction involves an exponentially large label space. As a result, the assumption that the source and target domains share the same prediction function becomes less likely to hold.

How can we achieve unsupervised domain adaptation without relying on the assumption that the source and target domains share a same prediction function in a domain-invariant feature space? In this work, we propose a principled approach to model the residual in the feature space between the source and target domain. We train a deep residual net to transform the feature maps of source domain images to appear as if they were sampled from the target domain while maintaining their semantic spatial layouts. We propose a novel Generative Adversarial Network (GAN) - based architecture that is capable of learning such a transformation in an unsupervised manner. Note, we do not require corresponding image pairs from the two domains, which are not available in practice. Our unsupervised domain adaptation method offers the following advantages over existing approaches:

**No Assumption of Shared Feature Space:** Previous work [15, 40, 27, 5] align two domains via an intermediate feature space and thereby implicitly assume the same decision function for both domains. Our approach effectively relaxes this assumption by learning the residual between the feature maps from both domains [28]. As shown by our experiments, the relaxation of requiring a common feature space for both domains is the key to address the structured domain adaptation problem in an unsupervised setting.

**In-network Architecture:** To our best knowledge, state-of-the-art work [45, 9] typically rely on heuristic observations such as pixel distribution and label statistics to conduct domain adaptation for semantic segmentation. In contrast, we propose to transform the feature map with a conditional generator, hence all components in our method are within one network and trained in an end-to-end fashion. Although a discriminator has been used by [45, 9] to conduct adversarial learning, we show that the *conditional* generator is the

key for structured domain adaptation in Section 4.4.

**Data Augmentation:** Conventional domain adaptation approaches are limited to learning from a finite set of source and target data. However, by conditioning on both source images and a stochastic noise channel, our model enables to create virtually unlimited stochastic samples that appear similar to feature maps from the target domain images.

To demonstrate the efficacy of our method, we conduct structured domain adaptation experiments with the SYNTHIA [35] and GTA [34] as the source domain, Cityscapes [10] as the target domain. Without using labels of real images, our method significantly outperforms the state-of-the-art structured domain adaptation methods [21, 45, 9] by around $12\% \sim 20\%$ mean IoU.

## 2. Related Work

We discuss some related work on domain adaptation and semantic segmentation, with a particular focus on transferring knowledge from virtual images to real photos.

### 2.1. Semantic Segmentation

Semantic segmentation is the task of assigning a semantic category label to each pixel in an image. Traditional methods rely on local image features handcrafted by domain experts [24]. After the seminal work [26] introduced the fully convolutional network (FCN) to semantic segmentation, most recent top-performing methods follow upon FCN [1, 6, 46].

Despite the existence of weakly-supervised semantic segmentation [42, 36], an enormous amount of labor-intensive work is often required to annotate many images in order to achieve state-of-the-art semantic segmentation accuracy. The PASCAL VOC2012 Challenge [13] contains nearly 10,000 annotated images for the segmentation competition, and the MS COCO Challenge [25] includes over 200,000 annotated images. According to [10], it took about 1.5 hours to manually segment each image in Cityscapes; In contrast, annotating a synthetic image took only 7 seconds on average through a computer game.

We instead explore the use of almost effortlessly labeled synthetic images for training high-quality segmentation networks. For the urban scenes, we use the SYNTHIA [35] and GTA [34] dataset which contains images of virtual cities.

### 2.2. Domain Adaptation

Conventional machine learning algorithms rely on the assumption that the training and test data are drawn i.i.d. from the same underlying distribution. However, in practice it is common that there exists some discrepancy between training data and testing data. Domain adaptation aims to rectify this mismatch and tune the models toward better generalization at testing phase [15, 40, 27, 5].

Most of the previous deep domain adaptation methods operate mainly under the assumption that the adaptation is realized by matching the distribution of features from different domains. These methods aim to obtain domain-invariant features by minimizing a task-specific loss on the source domain and the divergence between domains, which is usually measured by MMD [27] or DANN [5].

**Generative Adversarial Net in Domain Adaptation:** The Generative Adversarial Networks (GANs) [18] is a framework for learning generative models. In [14], a discriminator is harnessed to distinguish the source domain and target domain, but no generator is used. They tried to learn features that minimize the discrepancy between two domains by fooling a discriminator. Bousmalis *et al*. [4] utilizes GAN to transfer the style of images from source domain to target domain, so that one shared classifier could accommodate both domains. In [39], the authors employ adversarial learning to train a face classifier on images and adapt it to video domain.

### 2.3. Structured Domain Adaptation

Most existing domain adaptation literatures are limited to classification or regression. Instead, structured domain adaptation, which is significantly challenging, has been seldom discussed. To our best knowledge, this problem is only considered in several recent works [7, 21, 45, 9, 8]. In detail, [7] aims to train a cross-domain image captioning model with adversarial loss. The other four work [21, 45, 9, 8] focus on domain adaptation to enhance the segmentation performance on real images by networks trained on virtual ones, which is also our concern in this work. Although [21, 9] have utilized a discriminator to conduct adversarial learning, they do not resort to a generator to perform domain adaption. Instead, they reply on some heuristic observations about pixel distribution, label statistics, *etc*. In this work, we show the generator is indeed the key for structured domain adaptation.

## 3. Methods

Given a labeled dataset in a source domain and an unlabeled dataset in a target domain, our goal is to train a semantic segmentation model in the source domain that generalizes to the target domain. In this section, we start with explaining the general design of our conditional GAN model in the context of structured domain adaptation. Then, the detailed architecture of each component is elaborated.

### 3.1. Overview

To close the domain gap between synthetic images and real images, we employ a generative adversarial objective to encourage our conditional generator $G$ to produce feature maps that appear to be extracted from the target domain images. Different from a vanilla GAN formulation

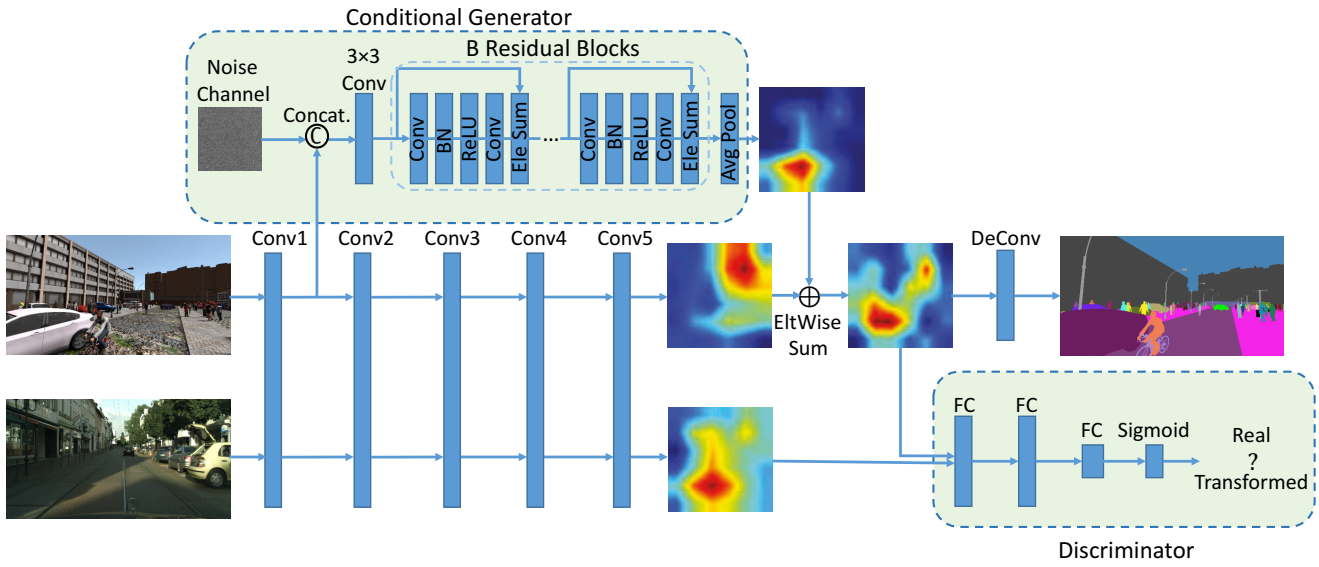



Figure 2: Details of the proposed Structured Domain Adaptation network. (a) The backbone network is FCN-8s [26] initialized with VGG-19 [38], we omit the skip connection for simplification. (b) The generator is a deep residual network which takes the features with fine-grained details from the lower-level layer as input and passes them to $3 \times 3$ convolutional filters to integrate the noise channel. Then B residual blocks are employed to learn the residual representation between the pooled features from "Conv5" for source-domain images and the target-domain representation. (c) The discriminator takes the features of target domain images and the enhanced representation of source domain features as inputs and tries to distinguish them. The discriminator consists of three fully connected layers followed by sigmoid activation, which is used to estimate the probability that the current input representation belongs to real images.

[18] in which the generator only takes a noise vector as input, our generator is conditioned on the extra auxiliary information, *i.e.*, the features maps $x^s$ of the synthetic image. During training, our generator $G(x^s, z; \theta_G) = x^s_{\text{Conv5}} + \hat{G}(x^s, z; \theta_G)$ transforms the feature maps $x^s$ of a synthetic image and a noise map $z$ to an adapted feature map $x^f$. Note that the generator produces the residual representation $\hat{G}(x^s, z; \theta_G)$ between the Conv5 feature maps of real and synthetic image, rather than directly computing $x^f$.

We expect that $x^f$ preserves the semantic of the source feature map $x^s$, meanwhile appears as if it were extracted from a target domain image, *i.e.*, a real image. Therefore, we feed $x^f$ to a discriminator branch $D(x; \theta_D)$, as well as a pixel-wise classifier branch $T(x; \theta_T)$. Specifically, the discriminator branch $D(x; \theta_D)$ aims to distinguish between transformed feature maps $x^f$ produced by the generator, and the feature maps of a real image from the target domain $x^t$, while the pixel-wise classifier branch $T(x; \theta_T)$ assigns a class label to each pixel in input image, which is implemented as deconvolution following FCN [26]. The overall architecture of our model is shown in Figure 2.

Our goal is to optimize the following minimax objective:

$$\min_{\theta_G, \theta_T} \max_{\theta_D} \mathcal{L}_d(G, D) + \alpha \mathcal{L}_t(G, T), \tag{1}$$

where $\alpha$ is a weight that controls the combination of the losses. $\mathcal{L}_d$ represents the domain loss:

$$\begin{aligned} \mathcal{L}_d(D, G) = &\mathbb{E}_{x^t}[\log D(x^t; \theta_D)] + \\ &\mathbb{E}_{x^s, z}[\log(1 - D(G(x^s, z; \theta_G); \theta_D))]. \end{aligned} \tag{2}$$

Following in FCN [26], we define the task loss $\mathcal{L}_t$ as multinomial logistic loss (a.k.a. cross entropy loss):

$$\begin{aligned} \mathcal{L}_t(G, T) = \mathbb{E}_{x^s, y^s, z}\Big[ &- \sum_{i=1}^{|I^s|} \sum_{k=1}^{K} \mathbf{1}^{y_i = k} \log\left(T(x_i^s; \theta_T)\right) \\ &- \sum_{i=1}^{|I^s|} \sum_{k=1}^{K} \mathbf{1}^{y_i = k} \log\left(T(G(x_i^s, z; \theta_G); \theta_T)\right)\Big], \end{aligned} \tag{3}$$

where $\sum_{i=1}^{|I^s|}$ and $\sum_{k=1}^{K}$ indicate the summarization over all $|I^s|$ pixels and $K$ semantic classes, $\mathbf{1}^{y_i = k}$ is a one-hot encoding of the $i$-th pixel.

In our implementation, $G$ is a convolutional neural network with residual connections. Our discriminator $D$ is a multi-layer perceptron. The minimax optimization is achieved by alternating between two steps. During the first step, we update the pixel-wise classifier $T$ and the discriminator $D$, while keeping the conditional generator $G$ and

feature extractor Conv1 $\sim$ Conv5 fixed. During the second step, we fix $T$, $D$ and update $G$ and Conv1 $\sim$ Conv5.

Notice that we train $T$ with both adapted and non-adapted source feature maps. Training $T$ solely on adapted feature maps leads to similar performance, but requires many runs with different initializations and learning rates due to the instability of the GAN. Indeed, without training on source as well, the model is free to shift class assignments (*e.g.* class 1 becomes 2, class 2 becomes 3 *etc.*), meanwhile the objective function is still optimized. Similiar to [4], training classifier $T$ on both source and adapted images avoids this shift and greatly stabilizes training.

### 3.2. Conditional Generator Network Architecture

The generator network aims to generate real-image like representations for synthetic images to reduce the domain gap. To achieve this purpose, we design the generator as a deep residual learning network that augments the representations of synthetic images through residual learning.

As shown in Figure 2, the generator consumes the feature from the bottom convolutional layer that preserves informative low-level details for feature transformation. The Conv1 feature maps are first augmented with an additional noise channel to introduce randomness, then passed into the $3 \times 3$ convolution filters to adjust the feature dimension. Afterwards, we use B residual blocks to learn the residual representation between the features from synthetic and real images, as a generative model. All residual blocks share the identical layout consisting of two $3 \times 3$ convolutional filters followed by batch-normalization layer and ReLU activation. The learned residual representation is then used to enhance the feature pooled from "Conv5" for the source domain images by element-wise sum operations, producing transformed representations that appear to be generated from the target domain.

### 3.3. Discriminator Network Architecture

As shown in Figure 2, the discriminator network is trained to differentiate between the generated feature for the source domain images and the original one from the target domain images. Taking the vectorized feature maps as input, the discriminator passes it through two fully-connected layers followed by a sibling output layer with the sigmoid activation, and predicts the probability that the input representation is transformed. The output dimension of the first two fully-connected layers are 4096 and 1024 respectively.

By trying to distinguish the generated representation from the real image representation, an adversarial loss is introduced to encourage the generator network to produce the representation for the synthetic image similar to that of the real image.

### 3.4. Testing on real images

In testing phase, a real image is passed through the CNN feature extractor Conv1 $\sim$ Conv5 followed by the pixel-wise classifier $T$. The generator $G$ and $D$ would not be involved. Therefore, our network is supposed to have the same inference complexity as the vanilla FCN-8s [26]. We achieve $4.4$ fps with one GeForce GTX 1080 Ti GPU.

## 4. Experiments

In this section, we present the experimental setup and compare the results of our approach, its variations, and some existing baseline methods.

### 4.1. Datasets and Evaluation

In our experiments, we use the Cityscapes datasets [10] as target domain dataset, the SYNTHIA [35] or GTA [34] dataset as source domain datasets. All of them are publicly available.

Cityscapes [10] is a real-world, vehicle-egocentric image dataset collected in 50 cities in Germany and nearby countries. It provides four disjoint subsets: 2,993 training images, 503 validation image, 1,531 test images, and 20,021 auxiliary images. All the training, validation, and test images are accurately annotated with per pixel category labels, while the auxiliary set is coarsely labeled. There are 34 distinct categories in the dataset.

SYNTHIA [35] is a large dataset of synthetic images and provides a particular subset, called SYNTHIA-RANDCITYSCAPES, to pair with Cityscapes. This subset contains 9,400 images that are automatically annotated with 12 object categories, one void class, and some unnamed classes. Note that the virtual city used to generate the synthetic images does not correspond to any of the real cities covered by Cityscapes. We abbreviate SYNTHIA-RANDCITYSCAPES to SYNTHIA hereon.

GTA [34] is a synthetic, vehicle-egocentric image dataset collected from the open world in the realistically rendered computer game Grand Theft Auto V (GTA, or GTA5). It contains 24,996 images, whose semantic segmentation annotations are fully compatible with the classes used in Cityscapes. Hence we use all the 19 official training classes in our experiment.

**Experiment setup.** As in [21, 45], the Cityscapes validation set is used as our test set. We split 500 images out of the Cityscapes training set for validation (*i.e.*, to monitor the convergence of the networks). In training, we randomly sample mini-batches from the images (and their labels) in source domain dataset and the real images (*without* labels) in target domain dataset.

For the SYNTHIA dataset [35], we consider the 16 common classes shared with Cityscapes [10]: sky, building, road, sidewalk, fence, vegetation, pole, car, traffic sign, per-

| Method | IoU (/%) | Class-wide IoU (/%) | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | bike | fence | wall | t-sign | pole | mbike | t-light | sky | bus | rider | veg | building | car | person | sidewalk | road |
| NoAdapt [21] | 17.4 | 0.0 | 0.0 | 1.2 | 7.2 | 15.1 | 0.1 | 0.0 | 66.8 | 3.9 | 1.5 | 30.3 | 29.7 | 47.3 | 51.1 | 17.7 | 6.4 |
| FCN Wld [21] | 20.2 | 0.6 | 0.0 | **4.4** | 11.7 | 20.3 | 0.2 | 0.1 | 68.7 | 3.2 | 3.8 | 42.3 | 30.8 | 54.0 | 51.2 | 19.6 | 11.5 |
| CL [45] | 29.0 | 13.1 | 0.5 | 0.1 | 3.0 | 10.7 | 0.7 | 3.7 | 70.6 | 20.7 | 8.2 | 76.1 | 74.9 | 43.2 | 47.1 | **26.1** | 65.2 |
| CCA [9] | - | 4.6 | - | - | 5.4 | - | 1.2 | 1.2 | **81.0** | 16.1 | 6.4 | **81.3** | **78.3** | 63.5 | 37.4 | 25.6 | 62.7 |
| Ours | **41.2** | **29.5** | **3.0** | 3.4 | **21.3** | **31.5** | **17.9** | **19.5** | 69.4 | **41.6** | **25.0** | 67.4 | 73.5 | **76.5** | **68.5** | 25.8 | **85.0** |

Table 1: Comparison results for the semantic segmentation of the Cityscapes images [10] by adapting from SYNTHIA [35]. The IoUs of CCA [9] for fence, wall and pole are not reported, thus we could not show the mean IoU of CCA [9] for all the 16 classes. For the remaining 13 classes, the mean IoU of CCA [9] is $35.7\%$, while our method achieves $\mathbf{47.7}\%$.

| Method | IoU (/%) | Class-wise IoU (/%) | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | bike | fence | wall | t-sign | pole | mbike | t-light | sky | bus | rider | veg | terrain | train | building | car | person | truck | sidewalk | road |
| NoAdapt [21] | 21.1 | 0.0 | 3.1 | 7.4 | 1.0 | 16.0 | 0.0 | 10.4 | 58.9 | 3.7 | 1.0 | 76.5 | 13 | 0.0 | 47.7 | 67.1 | 36 | 9.5 | 18.9 | 31.9 |
| FCN Wld [21] | 27.1 | 0.0 | 5.4 | **14.9** | 2.7 | 10.9 | 3.5 | 14.2 | 64.6 | 7.3 | 4.2 | **79.2** | 21.3 | 0.0 | 62.1 | 70.4 | 44.1 | 8.0 | 32.4 | 70.4 |
| CL [45] | 28.9 | 14.6 | **11.9** | 6.0 | 11.1 | 8.4 | 16.8 | 16.3 | **66.5** | 18.9 | 9.3 | 75.7 | 13.3 | 0.0 | **71.7** | 55.2 | 38.0 | 18.8 | 22.0 | 74.9 |
| Ours | **44.5** | **35.4** | 10.9 | 13.5 | **33.7** | **38.5** | **25.5** | **29.4** | 65.8 | **45.2** | **32.4** | 77.9 | **37.6** | **0.0** | 70.7 | **77.8** | **75.1** | **39.2** | **49.0** | **89.2** |

Table 2: Comparison results for the semantic segmentation of the Cityscapes images [10] by adapting from GTA [34]. CCA [9] does not report the experimental results on GTA dataset [34], thus is omitted in this table.

son, bicycle, motorcycle, traffic light, bus, wall, and rider. For the GTA dataset [34], we consider the 19 common classes shared with Cityscapes [10]: bike, fence, wall, traffic sign, pole, motorcycle, traffic light, sky, bus, rider, vegetation, terrain, train, building, car, person, truck, sidewalk and road.

**Evaluation.** We use the evaluation code released along with the Cityscapes dataset to evaluate our results. It calculates the PASCAL VOC [13] intersection-over-union, *i.e.*, $\text{IoU} = \frac{\text{TP}}{\text{TP+FP+FN}}$, where TP, FP, and FN are the numbers of true positive, false positive, and false negative pixels, respectively, determined over the whole test set. Since we have to resize the images before feeding them to the segmentation network, we resize the output segmentation mask back to the original image size before running the evaluation against the groundtruth annotations.

## 4.2. Implementation Details

In our experiments, we use FCN-8s [26] as our backbone network. We initialize it with VGG-19 [38], and then train it using the Adam optimizer [23]. Each mini-batch consists of 5 source images and 5 target images, while we use the largest possible minibatch of 15 source images when we train the baseline network with no adaptation. The network is implemented in PyTorch. Our machine is equipped with 190 GB memory and 8 GeForce GTX 1080 Ti GPUs.

All images are resized to $480 \times 960$, thus, the feature map of Conv1 output is $[64, 339, 579]$. Consequently, $z$ is a matrix of $339 \times 579$ elements, each of which is sampled from a uniform distribution $z_{ij} \sim \mathcal{U}(-1, 1)$. $z$ is concatenated to the Conv1 feature map as an extra channel, and fed to a $3 \times 3$ convolutional layer with input channel 65 and output channel 64. We set $\text{B} = 16$ for the number of the residual blocks, where each block contains 64 convolutional filters.

## 4.3. Performance Comparison

We report the final semantic segmentation results on the test data of the target domain in this section. We compare our approach to the following competing methods.

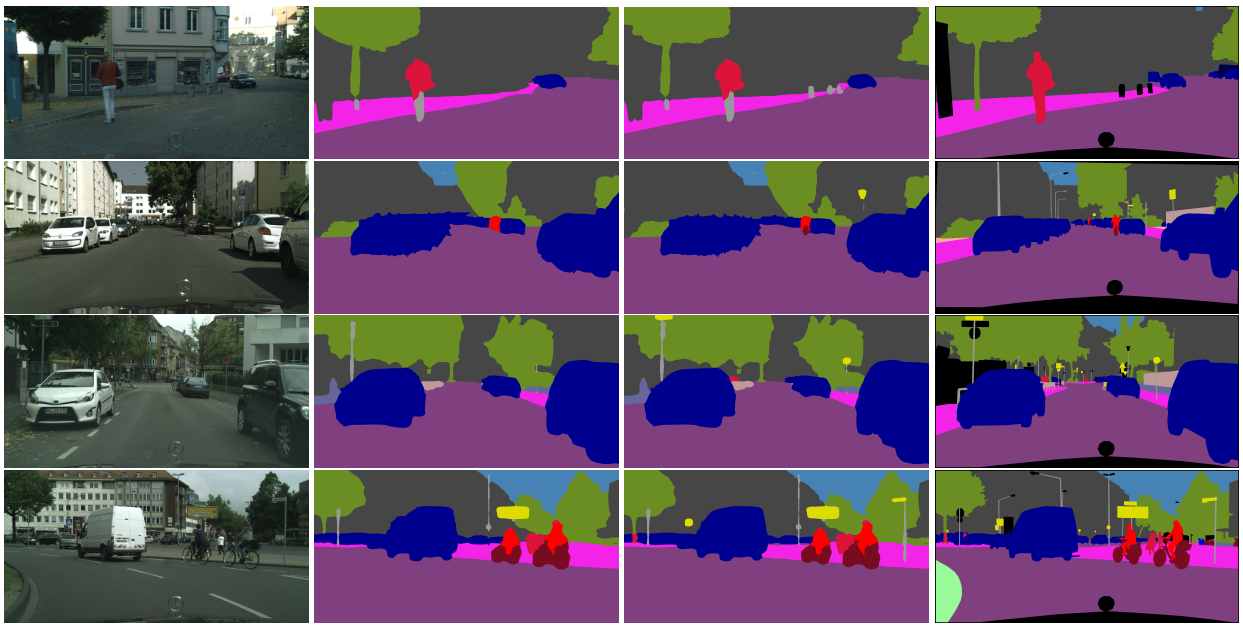
**No adaptation (NoAdapt).** We directly train the FCN-8s model on SYNTHIA and GTA without any domain adaptation. This is the most basic baseline for our experiments.

**FCNs in the wild (FCN Wld)** [21] introduces a pixel-level adversarial loss to the intermediate layers of the network and impose constraints on label statistics to the network output.

**Curriculum learning (CL)** [45] proposes a curriculum-style learning approach to minimize the domain gap in semantic segmentation. The curriculum domain adaptation first solves easy tasks such as estimating label distributions, then infers the necessary properties about the target domain.

**Cross city adaptation (CCA)** [9] refines the pre-trained semantic segmentation network by integrating static-object priors with the global class-specific learning framework. Their method is particularly designed for cross-domain discrimination on road scene images across different cities.

The comparison results are shown in Table 1 and Table 2. We note that all domain adaptation results are significantly better than those without adaptation (NoAdapt), which demonstrates the large domain gap between synthetic urban images and real images. In both datasets, our proposed method achieves higher class-wise IoU than the NoAdapt baseline for any class, and the mean IoU of our method is around $23\%$ higher than that of the NoAdapt baseline. These results verify the effectiveness of the proposed structured domain adaptation method. Compared with the state-of-the-arts approach [21, 45, 9], our method also outperforms them by a large margin of around $12\% \sim 20\%$ IoU. Note that the IoUs of CCA [9] of fence, wall and



(a) Input image. (b) From SYNTHIA. (c) From GTA. (d) Groundtruth.

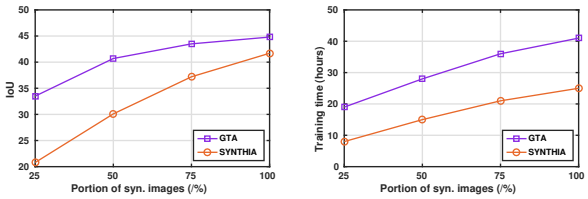
Figure 3: Qualitative Results. (a). Input images. (b). Testing results of the model adapted from SYNTHIA dataset [35]. (c). Testing results of the model adapted from GTA dataset [34]. (d). Groundtruth annotations.

pole are not reported, thus we could not show the mean IoU of CCA [9] for all the 16 classes. For the remaining 13 classes, the IoU of CCA [9] is 35.7%, while our method achieves **47.7**%, a significant improvement of 12%. Several representative results are shown in Figure 3.

#### 4.3.1 On the amount of synthetic data

From Table 1, 2 and Figure 3, we observe that the adaptation from GTA [34] is better than the adaptation from SYNTHIA [35], partly due to the reason that GTA dataset contains more images than SYNTHIA dataset ($24,996$ *v.s.* $9,400$). Thus, a natural question to ask is: given that the synthetic images are easy to collect and annotate, what is the trend of segmentation performance *w.r.t* the amount of synthetic data? We experimentally investigate this problem.

As shown in Figure 4a, the IoUs of models adapted from both SYNTHIA [35] and GTA [34] monotonously increase *w.r.t* the portion of source dataset (*i.e.*, the number of training synthetic images), yet the gains are dropping slowly when more synthetic images are used, especially for GTA as it contains more images. The trends in Figure 4a indicate: 1) Using more synthetic training data does improve the segmentation performance considerably, *e.g.*, using 25% *v.s.* 50% of SYNTHIA data, the IoU improves by about 10%; 2) The IoU tends to be saturate when more synthetic images are added. Beyond a substantial amount of synthetic data, the diversity of the scenes in the training images may matter more than the number of the training samples. Figure 4b presents the training time using different portions of source dataset.




(a) The IoU. (b) The training time.

Figure 4: (a) The change of IoU *w.r.t* various portions of synthetic images. (b) The change of training time *w.r.t* various portions of synthetic images.

### 4.4. Ablation Studies

We investigate the effectiveness of different components of the proposed structured domain adaptation framework. The performance achieved by different variants and parameter settings are reported in the following.

#### 4.4.1 The effectiveness of conditional generator

To verify the importance of our conditional generator in enhancing the feature maps, we compare it with a simple baseline "Skip Pooling" proposed in [2]. We also try to train our

| IoU (/%) | SYNTHIA | GTA |
|---|---|---|
| Skip Pooling | 22.7 | 24.9 |
| Without Generaror | 17.1 | 20.5 |
| With Generator | 41.2 | 44.5 |

Table 3: With/Without the conditional generator.

network without the conditional generator, *i.e.*, we remove the conditional generator $G$, and only keep the feature extractor, pixel-wise classifier $T$ and the discriminator $D$. We note that this shares the spirit with [14] that tries to learn domain-invariant features with a domain classifier.

The experimental results are shown in Table 3. Our conditional generator outperforms "Skip Pooling" by around $20\%$ mean IoU in both datasets, which validates that our method can effectively incorporate fine-grained details from low-level layers to improve semantic segmentation.

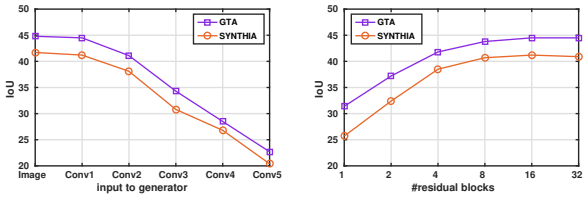
When the conditional generator is removed, the IoU significantly drops around $24\%$. This outcome verifies our previous hypothesis, *e.g.*, simply learning domain-invariant features fails to handle the structured domain adaptation problem due to the large prediction space. In view of the importance of the conditional generator, we investigate its different design choices by the following experiments.

#### 4.4.2 Different lower layers for learning generator

The proposed generator leverages fine-grained details of synthetic images from representations of lower-level convolutional layers. In particular, we employ the features from "Conv1" as the inputs for learning the generator. To validate the effectiveness of this setting, we conduct additional experiments using features from "Conv2" to "Conv5", as well as from the original input image directly for learning the generator, respectively. As shown in Figure 5a, the performance consistently decreases by employing the representations from higher convolutional layers.

In general, deep features in standard CNNs eventually evolve from general to specific along the network, and the transferability of features and classifiers will decrease when the cross-domain discrepancy increases [44]. In other words, the shifts in the data distributions linger even after multilayer feature abstractions, and the lower layers can capture more low-level details than the higher layers. Therefore, using low-level features from "Conv1" provides the best performance among all convolutional layers.

In addition, we observe that directly learning the generator from the input images produces similar or slightly-higher IoU in testing phase. However, the size of "Conv1" feature maps is only half of the original image due to the fact that the "Conv1" layer has stride 2, hence we feed "Conv1" layer outputs rather than the original images to the generator for efficient computation.




(a) Different lower layer. (b) Number of residual blocks.

Figure 5: Different design choices for the conditional generator. (a) The tesing IoU for domain adaptation from different lower layers. (b) The testing IoU *w.r.t* different numbers of the residual blocks.

| IoU (/%) | SYNTHIA | GTA |
|---|---|---|
| Without Noise | 40.7 | 43.2 |
| With Noise | 41.2 | 44.5 |

Table 4: With/Without the noise channel.

#### 4.4.3 On the number of residual blocks

We also vary the number of residual blocks. As shown in Figure 5b, the testing IoU grows w.r.t. the number of residual blocks B, but the gains are diminishing when B is relatively large. Due to the minor IoU gap between $\mathrm{B} = 16$ and $\mathrm{B} = 32$, we simply use $\mathrm{B} = 16$ for computation efficiency.

#### 4.4.4 How much does the noise channel contribute?

We conduct controlled experiments to verify the effectiveness of the noise channel. As shown in Table 4, the noise channel can marginally improve the IoU of the proposed methods. By conditioning on both source image feature map and noise input, our model can even create an unlimited number of training samples.

## 5. Conclusion

In this paper, we address structured domain adaptation for the semantic segmentation of urban scenes. We propose a GAN-based approach to this problem, and learn a conditional generator to transform the feature maps of source domain images as if they were extracted from target domain images. We use a discriminator to encourage realistic transformations. Our method outperforms other state-of-the-art approaches that concern domain adaptation from simulated images to real photos of urban traffic scenes.

## Acknowledgements

This work is supported in part by start-up grants of University at Buffalo, Computer Science and Engineering Department.

## References


[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481, 2017. 3

[2] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2874–2883, 2016. 7

[3] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010. 2

[4] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3, 5

[5] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems*, pages 343–351, 2016. 2, 3

[6] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 3

[7] T.-H. Chen, Y.-H. Liao, C.-Y. Chuang, W.-T. Hsu, J. Fu, and M. Sun. Show, adapt and tell: Adversarial training of cross-domain image captioner. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 3

[8] Y. Chen, W. Li, and L. Van Gool. ROAD: Reality oriented adaptation for semantic segmentation of urban scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[9] Y.-H. Chen, W.-Y. Chen, Y.-T. Chen, B.-C. Tsai, Y.-C. Frank Wang, and M. Sun. No more discrimination: Cross city adaptation of road scene segmenters. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2, 3, 6, 7

[10] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 1, 2, 3, 5, 6

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 1

[12] D. Dwibedi, I. Misra, and M. Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[13] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015. 1, 3, 6

[14] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015. 3, 8

[15] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016. 2, 3

[16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2

[17] R. Girshick. Fast R-CNN. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 1440–1448. IEEE, 2015. 1

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 3, 4

[19] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2315–2324. IEEE, 2016. 2

[20] H. Hattori, V. Naresh Boddeti, K. M. Kitani, and T. Kanade. Learning scene-specific pedestrian detectors without real data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3819–3827, 2015. 2

[21] J. Hoffman, D. Wang, F. Yu, and T. Darrell. FCNs in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016. 3, 5, 6

[22] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 746–753. IEEE, 2017. 2

[23] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representation*, 2015. 6

[24] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, pages 109–117, 2011. 3

[25] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 3

[26] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 3431–3440. IEEE, 2015. 1, 3, 4, 5, 6

[27] M. Long, Y. Cao, J. Wang, and M. Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning*, pages 97–105, 2015. 2, 3

[28] M. Long, H. Zhu, J. Wang, and M. I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, pages 136–144, 2016. 2

[29] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison. SceneNet RGB-D: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[30] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010. 2

[31] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE signal processing magazine*, 32(3):53–69, 2015. 2

[32] W. Qiu and A. Yuille. UnrealCV: Connecting computer vision to unreal engine. In *European Conference on Computer Vision Workshop VARVAI*, pages 909–916. Springer, 2016. 2

[33] S. R. Richter, Z. Hayder, and V. Koltun. Playing for benchmarks. In *International Conference on Computer Vision (ICCV)*, 2017. 2

[34] S. R. Richter, V. Vineet, S. Roth, and V. Koltun. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision*, pages 102–118. Springer, 2016. 2, 3, 5, 6, 7

[35] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2016. 2, 3, 5, 6, 7

[36] F. Sadat Saleh, M. Sadegh Aliakbarian, M. Salzmann, L. Petersson, and J. M. Alvarez. Bringing background into the foreground: Making all classes equal in weakly-supervised video semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2106–2116, 2017. 3

[37] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[38] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representation*, 2015. 1, 4, 6

[39] K. Sohn, S. Liu, G. Zhong, X. Yu, M.-H. Yang, and M. Chandraker. Unsupervised domain adaptation for face recognition in unlabeled videos. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 3

[40] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4068–4076, 2015. 2, 3

[41] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[42] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1568–1576, 2017. 3

[43] M. Yamada, L. Sigal, and Y. Chang. Domain adaptation for structured regression. *International journal of computer vision*, 109(1-2):126–145, 2014. 2

[44] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014. 2, 8

[45] Y. Zhang, P. David, and B. Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2, 3, 5, 6

[46] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[47] Z. Zheng, L. Zheng, and Y. Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2

[48] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 3357–3364. IEEE, 2017. 2