Domain Randomization and Pyramid Consistency: Simulation-to-Real Generalization without Accessing Target Domain Data

Xiangyu Yue[1], Yang Zhang[2], Sicheng Zhao[1], Alberto Sangiovanni-Vincentelli[1],
Kurt Keutzer[1], Boqing Gong[3]
[1] University of California, Berkeley, [2] University of Central Florida, [3] Google
{xyyue,schzhao,alberto,keutzer}@berkeley.edu, yangzhang4065@gmail.com, boqinggo@outlook.com

Abstract

*We propose to harness the potential of simulation for the semantic segmentation of real-world self-driving scenes in a **domain generalization** fashion. The segmentation network is trained without any data of target domains and tested on the unseen target domains. To this end, we propose a new approach of domain randomization and pyramid consistency to learn a model with high generalizability. First, we propose to randomize the synthetic images with the styles of real images in terms of visual appearances using auxiliary datasets, in order to effectively learn domain-invariant representations. Second, we further enforce pyramid consistency across different "stylized" images and within an image, in order to learn domain-invariant and scale-invariant features, respectively. Extensive experiments are conducted on the generalization from GTA and SYNTHIA to Cityscapes, BDDS and Mapillary; and our method achieves superior results over the state-of-the-art techniques. Remarkably, our generalization results are on par with or even better than those obtained by state-of-the-art simulation-to-real domain adaptation methods, which access the target domain data at training time.* [1]

1. Introduction

Simulation has spurred growing interests for training deep neural nets (DNNs) for computer vision tasks [53, 10, 23, 55]. This is partially due to the community's recent exploration to embodied vision [46, 62, 2], in which the perception has to be embodied and purposive for an agent to actively perceive and/or navigate through a physical environment [7, 10]. Moreover, training data generated by simulation is often low-cost and diverse, especially benefiting the tasks that otherwise need heavy human annotations (*e.g.* semantic segmentation [19, 57, 18]). Finally, in the case of autonomous driving, simulation can complement the insufficient coverage of real data by synthesizing rare events and scenes, such as construction sites, lane merges, and accidents. In summary, the promise of simulation is that one may conveniently acquire a large amount of labeled and diverse imagery from simulated environments. This scale is vital for training state-of-the-art deep convolutional neural networks (CNNs) with millions of parameters.

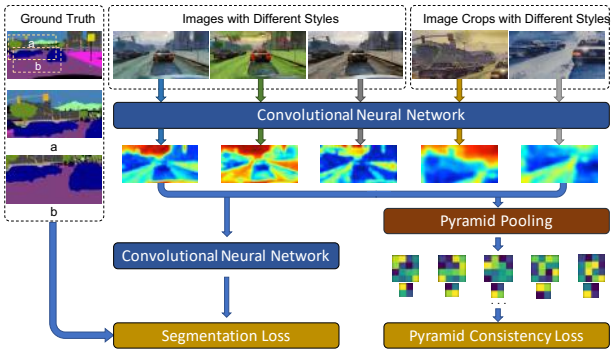

Figure 1. Domain randomization and pyramid consistency enforce the learned semantic segmentation network invariant to the change of domains. As a result, the semantic segmentation network can generalize to various domains, including those of real scenes.

However, when we learn a semantic segmentation neural network from a synthetic dataset, its visual difference from real-world scenes often discounts its performance on real images. To mitigate the domain mismatch between simulation and the real world, existing work often resorts to **domain adaptation** [19, 18, 57], which aims to tailor the model for a particular target domain by jointly learning from the source synthetic data and the (often unlabeled) data of the target real domain. This setting is, unfortunately, very stringent. Take autonomous driving for instance. It is almost impossible for a car manufacturer to know in advance under which domain (which city, what weather, day or night) the vehicle would be used.

In this paper, we instead propose to harness the potential of simulation from a **domain generalization** manner [1, 27, 14, 46], without the need of accessing any target

[1] Our code is available at https://github.com/xyyue/DRPC.

domain data in training and yet aiming to generalize well to multiple real-world target domains. We focus on the semantic segmentation of self-driving scenes, but the proposed method is readily applicable to similar tasks and scenarios. Our main idea is to randomize the labeled synthetic images to the styles of real images. We further enforce the semantic segmentation network to generate consistent predictions, in a pyramid form, over these domains. Our conjecture is that if the network is exposed to a sufficient number of domains in the training stage, it should *interpolate* well to new real-world target domains. In contrast, the domain adaptation work [19, 57, 18] can be seen as *extrapolating* from a single source domain to a single target domain.

Our approach comprises two key steps: domain randomization and consistency-enforced training, as illustrated in Figure 1. Unlike [48, 39], we do not require any control of the simulators for randomizing the source domain imagery. Instead, we leverage the recently advanced image-to-image translation [60] to transfer a source domain image to multiple styles, each dubbed an *auxiliary domain*. This has at least three advantages over manipulating the simulator. First, it enables us to select auxiliary domains from the real world. After all, our goal is to achieve good performance on real data. Second, we have a more concrete anticipation about the look of the randomized images as we view the auxiliary domains. Finally, the randomized images are naturally grouped according to the auxiliary domains. The last point facilitates us to devise effective techniques in the second step to train the networks in a domain-invariant way.

In the second step of our approach, we train a deep CNN for semantic segmentation with a pyramid consistency loss. If the network fits well to not only the synthetic source domain but also the auxiliary domains — synthetic images with the styles of real images, it may become invariant to domain changes to a certain degree and thus generalize well to real-world target domain(s). To ensure consistent performance across different training domains, we explicitly regularize the network's internal activations so that they do not deviate from each other too much for the stylized versions of the same source domain image. We find that it is vital to apply the regularization over average-pooled pyramids rather than the raw feature maps, probably because the pooled pyramid gives the network certain flexibility — the pyramid allows some errors made by the network on the finest-grained pixel level as long as the average activations are about the same across different training domains.

To the best of our knowledge, this is the first work to explore domain randomization for the semantic segmentation problem. Experiments show that the proposed approach gives rise to robust domain-invariant CNNs trained using synthetic images. These CNN models generalize well to multiple datasets of real images. It significantly outperforms the straightforward source-only baseline and the newly designed network [34], where the latter reduces the network's dependency on the training set by a hybrid of batch and instance normalizations. Our results are on par or even better than state-of-the-art domain adaptation results which are obtained by accessing the target data in training.

2. Related Work

We now discuss some related work on semantic segmentation, domain adaptation, domain generalization, domain randomization, and data augmentation.

Domain Adaptation for Semantic Segmentation. Until [19, 57] first studied the domain shift problem in semantic segmentation, most works in domain adaptation had focused on the task of image classification. After that, the problem subsequently became one of the tracks in the Visual Domain Adaptation Challenge (VisDA) 2017 [35] and started receiving increasing attention. Since then, adversarial training has been utilized in most of the following works [18, 3, 41, 58] for feature alignment. Most of these works were inspired by the unsupervised adversarial domain adaptation approach in [13] which shares similar idea with generative adversarial networks. One of their most important objectives is to learn domain-invariant representations by trying to deceive the domain classifier. Zhang *et al.* [57] perform segmentation adaptation by aligning label distributions both globally and across superpixels in an image. Recently, an unsupervised domain adaptation method has been proposed for semantic segmentation via class-balanced self-training [63]. Please refer to [56, Section 5] for a brief survey of other related works.

Domain Generalization In contrast to Domain Adaptation, where the network is tested on a known target domain, and the images in the target domain, although without labels, are accessible during the training process, Domain Generalization is tested on unseen domains [31, 12]. Current domain generalization researches mostly focus on the image classification problem. Image data is hard to manually divide into discrete domains, [15] devised a nonparametric formulation and optimization procedure to discover domains among both training and test data. [28] imposed Maximum Mean Discrepancy measure to align the distributions among different domains and train the network with adversarial feature learning. [26] assigned a separate network duplication to each training domain during training and used the shared parameter for inference. [27] improved generalization performance by using a meta-learning approach on the split training sets.

Domain Randomization. Domain randomization (DR) is a complementary class of techniques for domain adaptation. Tobin *et al.* [46] introduced the concept of Domain Randomization. Their approach randomly varies the texture and color of the foreground object, the background image, the number of lights in the scene, the pose of the lights,

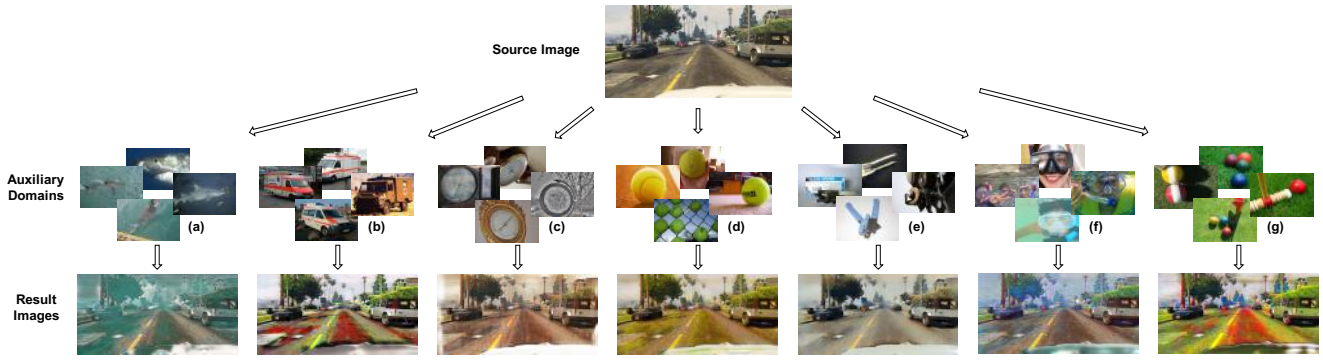

Figure 2. The domain randomization process. Top: an original synthetic image from the source domain; Mid: auxiliary image sets composed of ImageNet classes: (a) great white shark, (b) ambulance, (c) barometer, (d) tennis ball, (e) can opener, (f) snorkel, (g) tennis ball; Bottom: stylized images with same image content as the synthetic image and meanwhile corresponding styles of the ImageNet classes.

the camera position, etc. The goal is to close the reality gap by generating synthetic data with sufficient variation that the network views real-world data as just another variation. Randomization in the visual domain has been used to directly transfer vision-based policies from simulation to the real world without requiring real images during training [39, 46]. DR has also been utilized to do object detection and 6D pose estimation [49, 36, 45]. All the above DR methods require modifying objects inside the simulation environment. We instead propose a different DR method which is orthogonal to all the aforementioned methods.

Data Augmentation. Data augmentation is the process of supplementing a dataset with similar data created from the information in that dataset, which is ubiquitous in deep learning. When dealing with images, it often includes the application of rotation, translation, blurring, and other modifications [4, 51, 42] to existing images that allow a network to better generalize [43]. In [25], a network is proposed to automatically generate augmented data by merging two or more samples from the same class. A Bayesian approach is proposed in [47] to generate data based on the distribution learned from the training set. In [9], simple transformations are used in the learned feature space to augment data. Counterexamples are considered to help data augmentation in [11]. Recently, AutoAugment has been proposed to learn augmentation policies from data [6]. The type of domain randomization we proposed in this paper can also be considered as a type of data augmentation.

3. Approach

The main idea of our approach is twofold, illustrated in Figure 1. The first part is **Domain Randomization with Stylization**: mapping the synthetic imagery to multiple auxiliary real domains (cf. Figure 2) in the training stage, such that, at the test stage, the target domain is not a surprise for the CNN model but merely another real domain. The second part is **Consistency-enforced Training**: enforcing pyramid consistency across domains and within an image to learn representations with better generalization ability.

3.1. Domain Randomization with Stylization

Keeping in mind that the target domain consists of real images, we randomly draw K real-life categories from ImageNet [8] for stylizing the synthetic images. Each category is called an auxiliary domain. We then use the image-to-image translation work [60] to map the synthetic images to each of the auxiliary domains. As a result, the training set is augmented to $K+1$ times the original size.

Figure 2 illustrates this procedure and some qualitative results. We can see that each auxiliary domain stylizes the synthetic images by different real-world elements. Meanwhile, the semantic content of the original image is retained at most parts of the images. Some edge-preserving methods [29] on style transfer may give rise to better results, and are left for future work.

A straightforward method is to train a CNN segmentation model using the augmented training set. Denote by $\mathcal{D}^k, k = 0, 1, \cdots, K$, the training domains, where $\mathcal{D}^0$ stands for the original source domain of synthetic images and $\mathcal{D}^k, k > 0$ the auxiliary domains. A synthetic image $I_n^0 \in \mathcal{D}^0$ has K stylized copies $I_n^k \in \mathcal{D}^k$ in the auxiliary domains, and yet they all share the same semantic segmentation map Y_n as the labels. The objective function for training a segmentation network $f(\cdot;\theta)$ is:

$$\min_{\theta} \quad \mathcal{L} := \frac{1}{Z} \sum_{n} \sum_{k=0}^{K} L\Big(Y_n, f(I_n^k;\theta)\Big), \tag{1}$$

where θ denotes the weights of the network, $L(\cdot,\cdot)$ is the mean of pixel-wise cross-entropy losses, and $Z = (K+1)\big|\mathcal{D}^0\big|$ is a normalization constant.

Our experiments (cf. Section 4) show that the network trained using this augmented training set $\mathcal{D}^0 \cup \mathcal{D}^1 \cdots \cup \mathcal{D}^K$ generalizes better to the unseen target domain than using the single source domain $\mathcal{D}^0$. Two factors may attribute to this result: 1) the training set is augmented in size and

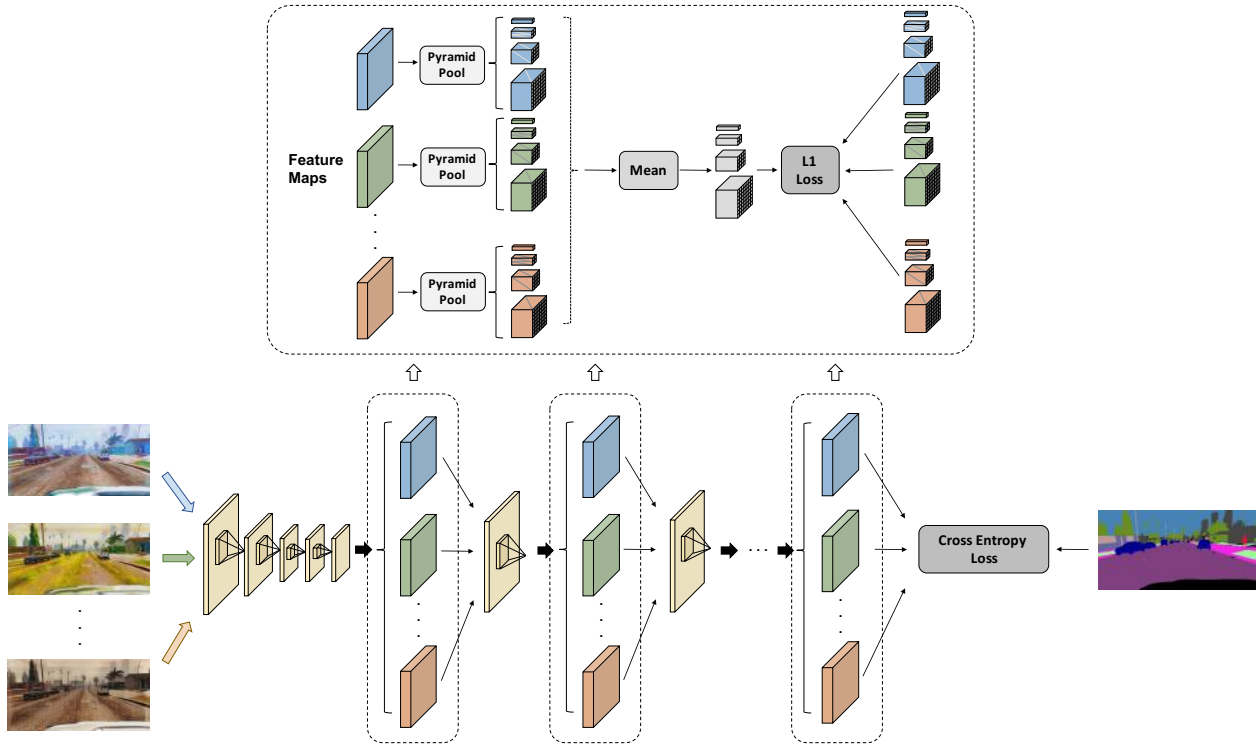

Figure 3. Pyramid Consistency across Domains. After feeding the images from different domains with the same content into the neural network, we impose the pyramid consistency loss on the activation maps at each of the last few layers (shown in blue, green and red).

2) the training set is augmented in style, especially in the styles closer to the real images. Despite being effective, this baseline method fails to track the multi-domain structure of the training set. We improve on it by the following.

3.2. Consistency-enforced Training

We aim to learn an image representation through the segmentation network that is domain-invariant for the semantic segmentation task. However, simply training the network with images from different domains (*i.e.*, Eq. (1), the baseline method) has some problems: a) images from different domains may drive the network toward distinct representations, making the training process not converge well; b) even if the network fits well to the training domains, it could capture the idiosyncrasies of each individual and yet fail at interpolating between them or to the new target domain.

In order to tackle these caveats, we regularize the network by a consistency loss. The intuition is that if the network can generalize well, it should extract similar high-level semantic features and perform similar predictions for the images with the same content regardless of the styles. The consistency loss is simply imposed as the following:

$$\mathcal{R} := \sum_{n,k} \sum_{l \in \mathcal{P}} \lambda_l \, L_1 \Big(\overline{g_l(\mathcal{I}_n; \theta)}, \, g_l(I_n^k; \theta) \Big), \qquad (2)$$

where l indexes an operator $g_l(\cdot; \theta)$ (*e.g.*, average pooling) which maps a hidden layer's activations to a vector of smaller dimension, $\overline{g_l(\mathcal{I}_n; \theta)}$ denotes the target after each operation g_l (cf. Sections 3.2.1 and 3.2.2 for details), and L_1 is the ℓ_1 distance. We argue that, by doing so, the network can be better guided to find a generic and domain-invariant representation for the semantic segmentation task.

The design of the operators $g_l(\cdot; \theta), l \in \mathcal{P}$ is key to the overall performance. The obvious identity mapping — so the ℓ_1 distance is directly calculated over the hidden activations — does not work well in the experiments. One of the reasons is that it strictly requires the network to give about the same representations across different training domains, while some domains may be harder than the others to fit.

3.2.1 Pyramid consistency across domains

We find that the spatial pyramid pooling [16, 59, 24] serves as very effective operators $g_l(\cdot; \theta), l \in P$ in our context probably because it accommodates subtle differences of the network representations and meanwhile enables Eq. (2) to enforce the consistency at multiple scales. Pyramid pooling has been used in supervised visual understanding before, mostly as a part of the backbone networks. In this paper, instead, we use the pooled features to define regularization losses for training the network. The pyramid consistency we consider is over the images of different styles but with

the same semantic content.

Figure 3 illustrates our pyramid consistency scheme across the training domains. Consider a set of images $\mathcal{I}_n = \{I_n^k \mid k = 0, 1, \ldots, K\}$ of $K+1$ different styles with the same annotation Y_n and denote by $M_n^{l,k} \in \mathbb{R}^{C_l \times H_l \times W_l}$ the feature map of input I_n^k at layer l. Then, a spatial pyramid pooling operation is done on $M_n^{l,k}$. The spatial pyramid pooling operation is designed to fuse features under four different pyramid levels. First of all, a global average pooling is of the coarsest level that generates a single bin output. Each other pyramid levels separates the feature map into sub-regions evenly and performs average pooling inside each sub-region. In our design, we use 1×1, 2×2, 4×4 and 8×8 as the pyramid pooling scales, namely the spatial size of the outputs of the pyramid pooling. After the pooling, we squeeze and concatenate the output tensors into a tensor $P_n^{l,k} \in \mathbb{R}^{C_l \times (1+2^2+4^2+8^2)}$, which is much lower-dimensional than the original feature map $M_n^{l,k}$. For a pair of images $I_n^k, I_n^{k'} \in \mathcal{I}_n$, the network is expected to have similar understanding and thus similar high-level features in a deep layer l. Note that simply constraining $M_n^{l,k}$ and $M_n^{l,k'}$ to be the same is too strong and could easily lead to degraded performance. To save computation, we avoided pair-wise terms and instead use the mean of $P_n^{l,k}(k = 0, 1, ..., K)$ as the target value for the loss. Back to equation (2), we have $g_l(I_n^k; \theta) = P_n^{l,k}$, the target is the mean across domains $\overline{g_l(\mathcal{I}_n; \theta)} = \frac{1}{K+1} \sum_k P_n^{l,k}$, and the set $\mathcal{P} = \{l\}$ is the layers down deep of the network.

3.2.2 Pyramid consistency within an image

The pyramid consistency loss across the training domains can guide the network to learn style-invariant features so that it can generalize well to the unseen target domains with different appearances. However, in many cases, style is not the only difference between domains. The view angles and parameters of cameras also lead to systematic domain mismatches in terms of the layout and scale of scenes. Take the *focal length* parameter for instance. With different focal lengths, the same objects may be of different scales as the fields of view vary.

In order to alleviate the issues above, we propose to further apply the pyramid consistency between random crops and full images. The idea is to artificially randomize the scale of the images and, therefore, guide the network to be robust to the domain gap incurred by the scene layouts and scales. Formally, following the notations in Section 3.2.1, each image I_n^k of size (H, W) is first randomly cropped at the same height-width ratio, with the top-left corner at (h_n^k, w_n^k) and with the height $\overline{h_n^k}$. Then the crop is scaled back to the full image size, denoted as C_n^k, and finally fed to the network. Denote by $M_n^{l,k}$ and $MC_n^{l,k} \in \mathbb{R}^{C_l \times H_l \times W_l}$ the feature maps of the image I_n^k and crop C_n^k at layer l, respectively. Denote by $\overline{M_n^{l,k}}$ the part of $M_n^{l,k}$ corresponding to the crop. When there is no significant padding through the layers, then $\overline{M_n^{l,k}}$ is of shape $C_l \times (\rho \cdot H_l) \times (\rho \cdot W_l)$, where $\rho = \overline{h_n^k}/h$.

We perform the spatial pyramid pooling on the cropped feature map $\overline{M_n^{l,k}}$ and the feature map $MC_n^{l,k}$ of the crop. The results are the same-size maps, $\overline{P_n^{l,k}}, PC_n^{l,k} \in \mathbb{R}^{C_l \times (1+2^2+4^2+8^2)}$. Back to Eq. (2), we have $g_l(I_n^k; \theta) = PC_n^{l,k}$ and the target vector is $\overline{g_l(\mathcal{I}_n; \theta)} = \overline{P_n^{l,k}}$.

4. Experiments and Results

In this section, we describe the experimental setup and present results on the semantic segmentation generalization by learning from synthetic data. Experimental analysis and comparison with other methods are also provided.

4.1. Experimental Settings

It should be emphasized that our experiment setting is different from domain adaptation. Since domain adaptation aims to achieve good performance on a particular target domain, it requires unlabeled target domain data during training and also (sometimes) uses some labeled target domain images for validation. In contrast, our model is trained without any target domain data and is tested on unseen domains.

Datasets. In our experiments, we use GTA [37] and SYNTHIA [38] as the source domains and a small subset of ImageNet [8] as well as datasets used in CycleGAN [60] as the auxiliary domains for "stylizing" the source domain images. We consider three target domains of real-world images, whose official validation sets are used as our test sets: Cityscapes [5], Berkeley Deep Drive Segmentation (BDDS) [54], and Mapillary [33].

GTA is a vehicle-egocentric image dataset collected in a computer game with pixel-wise semantic labels. It contains 24,966 images with the resolution 1914×1052. There are 19 classes which are compatible with other semantic segmentation datasets of outdoor scenes *e.g.* Cityscapes.

SYNTHIA is a large synthetic dataset with pixel-level semantic annotations. A subset, SYNTHIA-RAND-CITYSCAPES, is used in our experiments which contains 9,400 images with annotations compatible with Cityscapes.

Cityscapes contains vehicle-centric urban street images taken from some European cities. There are 5,000 images with pixel-wise annotations. The images have the resolution of 2048×1024 and are labeled into 19 classes.

BDDS contains thousands of real-world dashcam video frames with accurate pixel-wise annotations. It has a compatible label space with Cityscapes and the image resolution is 1280×720. The training, validation, and test sets contain 7,000, 1,000 and 2,000 images, respectively.

Mapillary contains street-level images collected from all around the world. The annotations contain 66 object

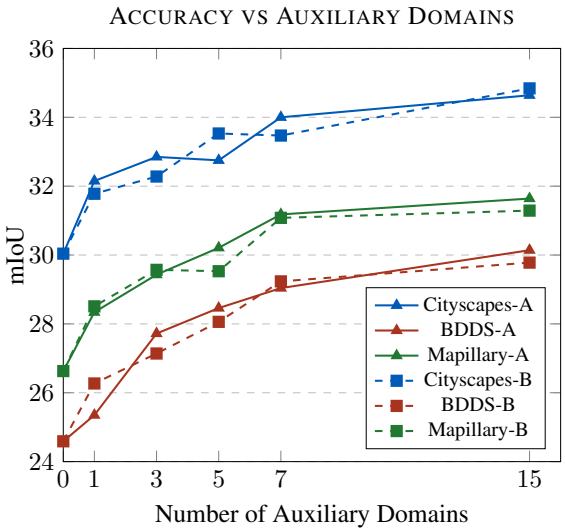

Figure 4. Accuracy of FCN8s-VGG16 with varying numbers of auxiliary domains. Two domain sets A and B are used. Models are trained on GTA and tested on Cityscapes, BDDS, and Mapillary.

classes, but only the 19 classes that overlap with Cityscapes and GTA are used in our experiments. It has a training set with 18,000 images and a validation set with 2,000 images.

Validation. To select a model for a particular real-world dataset $\mathcal{D}_R$ (*e.g.* Cityscapes), we randomly pick up 500 images from the training set of another real-world dataset $\mathcal{D}_{R'}$ (*e.g.* BDDS) as the validation set. This cross-validation is to imitate the following real-life scenarios. When we train a neural network from a randomized source domain without knowing to which target domain it will be applied, we can probably collect a validation set which is as representative as possible of the potential target domains. Still take the car manufacturers for instance. A manufacturer may collect images of Los Angeles and NYC for the model selection while the cars will also be used in San Francisco and many other cities.

Evaluation. We evaluate the performance of a model on a test set using the standard PASCAL VOC intersection-over-union, *i.e.* IoU. The mean IoU (mIoU) is the mean of all IoU values over all categories. To measure the generalizability of a model M, we propose a new metric,

$$G_{perf}(M) = \mathbb{E}_{\mathcal{B}\in\mathcal{P}}\, mIoU(M, \mathcal{B}) \approx \frac{1}{L}\sum_{l} mIoU(M, \mathcal{B}_l)$$

where $\mathcal{B}$ is an unseen domain drawn from a distribution of all possible real-world domains $\mathcal{P}$, and L is the number of unseen test domains, which is 3 in our experiment setting.

Implementation Details In our experiments, we choose to use FCN [30] as our semantic segmentation network. To make it easier to compare with most of other methods, we use VGG-16 [44], ResNet-50, and ResNet-101 [17] as FCN backbones. The weights of the feature extraction layers in the networks are initialized from models trained on ImageNet [8]. We add the pyramid consistency loss across domains on the last 5 layers, with $\lambda = 0.2, 0.4, 0.6, 0.8, 1$, respectively. The pyramid consistency within an image is only added on the last layer. The network is implemented in PyTorch and trained with Adam optimizer [22] using a batch size of 32 for the baseline models and 8 for our models. Our machines are equipped with 8 NVIDIA Tesla P40 GPUs and 8 NVIDIA Tesla P100 GPUs.

Table 1. Performance contribution of each design.

Method	DR	PCD	PCI	mIoU		
				Cityscapes	BDDS	Mapillary
FCN				30.04	24.59	26.63
+DR	✓			34.64	30.14	31.64
+PCD	✓	✓		35.47	31.21	32.06
+PCI	✓		✓	35.12	30.87	32.12
All	✓	✓	✓	**36.11**	**31.56**	**32.25**

4.2. Evaluation of Domain Randomization

In total, we use two sets of 15 auxiliary domains: A) 10 from ImageNet [8] and 5 from CycleGAN [60], and B) 15 from ImageNet with each domain corresponding to one semantic class in Cityscapes. Please see supplementary materials for additional auxiliary domains, including color augmentation as an auxiliary domain.

To evaluate our domain randomization method, we conduct experiments generalizing from GTA to Cityscapes, BDDS, and Mapillary with FCN8s-VGG16. We augment the training set with images from different numbers of auxiliary domains in both setting A and B, and show the result in Figure 4. As we can see from the plot, the accuracy increases with the number of auxiliary domains. The accuracy eventually saturates with the number of auxiliary domains. This is probably because 1) the 15 auxiliary domains are somehow sufficient to cover the appearance domain gap, and 2) as the number of images of the same content goes up, it is harder for the network to converge for the sake of the data scale and data variation.

4.3. Ablation Study

Next, we study how each design in our approach influences the overall performance. The experiments are still adapting from GTA to the 3 tests with FCN8s-VGG16. Table 1 details the mIoU improvement on Cityscapes, BDDS and Mapillary by considering one more factor each time: Domain Randomization (DR), Pyramid Consistency across Domains (PCD) and within an Image (PCI). DR is a generic way to alleviate domain shift. In our case, it helps boost the

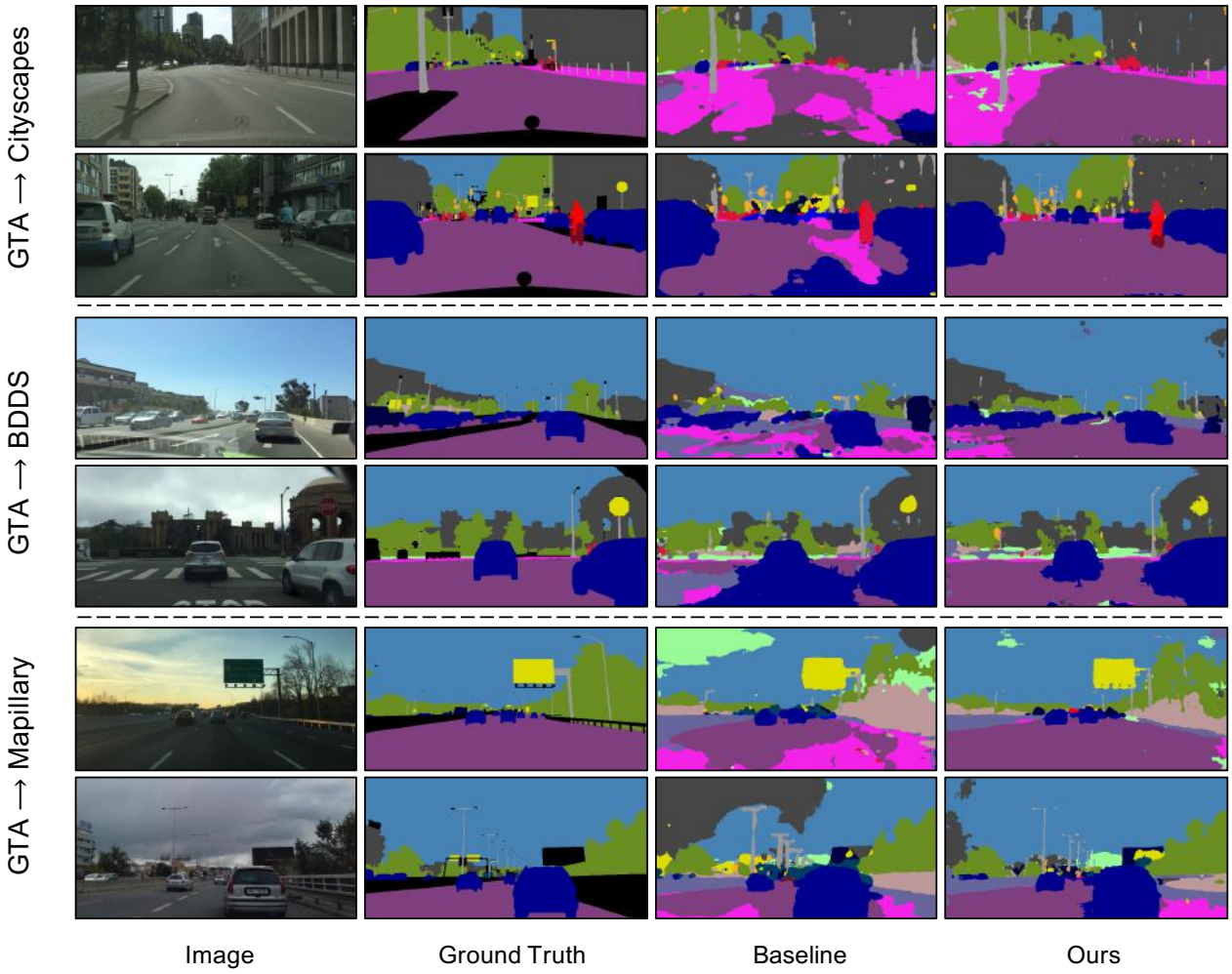

Figure 5. Qualitative semantic segmentation results of the generalization from GTA to Cityscapes, BDDS, and Mapillary.

Table 2. Domain generalization performance from (G)TA and (S)YNTHIA to (C)ityscapes, (B)DDS, and (M)apillary.

	VGG-16		ResNet-50		ResNet-101	
	NonAdapt	Ours	NonAdapt	Ours	NonAdapt	Ours
G→ C	30.04	**36.11**	32.45	**37.42**	33.56	**42.53**
G→ B	24.59	**31.56**	26.73	**32.14**	27.76	**38.72**
G→ M	26.63	**32.25**	25.66	**34.12**	28.33	**38.05**
G_{perf}	27.09	**33.31**	28.28	**34.56**	29.88	**39.77**
S → C	27.26	**35.52**	28.36	**35.65**	29.67	**37.58**
S → B	24.38	**29.45**	25.16	**31.53**	25.64	**34.34**
S → M	24.39	**32.27**	27.24	**32.74**	28.73	**34.12**
G_{perf}	25.34	**32.41**	26.92	**33.31**	28.01	**35.35**

performance from 30.04 to 34.64, from 24.59 to 30.14 and from 26.63 to 31.64, respectively for Cityscapes, BDDS and Mapillary. PCD and PCI further enhance the performance gains. By integrating all methods, our full approach finally reaches 36.11, 31.56 and 32.25 on Cityscapes, BDDS and Mapillary, respectively. Figure 5 showcases some examples of the semantic segmentation results on the 3 test sets.

Table 3. Comparison with other domain generalization methods.

Methods	Base Net	mIoU	mIoU↑
NonAdapt	ResNet-50	22.17	7.47
IBN-Net [34]		29.64	
NonAdapt	ResNet-50	32.45	4.97
Ours		**37.42**	

4.4. Generalization from GTA and SYNTHIA

Then, we conduct extensive experiments to evaluate the generalization ability of our proposed methods. Specifically, we tested *2 source domains*, GTA and SYNTHIA; *3 models with different backbone networks*, VGG-16, ResNet-50 and ResNet-101; *3 test sets*, Cityscapes, BDDS and Mapillary; and *2 sets of auxiliary domains* (cf. Section 4.2). The experiments with ResNet-50 are conducted with auxiliary domain set B, while the rest of the experiments are with set A. The validation set and test set in each experiment are from different domains, *e.g.* using

Table 4. Adaptation from GTA to Cityscapes with FCN-8s.

Network	Method	Train w/ Tgt	Val on Tgt	mIoU	mIoU↑
VGG-19	NonAdapt	✓	✓	22.3	6.6
	Curriculum [57]			28.9	
	NonAdapt	✓	✓	NA	NA
	CGAN [20]			44.5	
VGG-16	NonAdapt	✓	✓	21.1	6.0
	FCN wld [19]			27.1	
	NonAdapt	✓	✓	17.9	17.5
	CYCADA [18]			35.4	
	NonAdapt	✓	✓	29.6	7.5
	LSD [41]			37.1	
	NonAdapt	✓	✓	21.9	14.0
	ROAD [3]			35.9	
	NonAdapt	✓	✓	24.9	3.9
	MCD [40]			28.8	
	NonAdapt	✓	✓	NA	NA
	I2I [32]			31.8	
	NonAdapt	✓	✓	24.3	11.8
	CBST-SP [63]			36.1	
	NonAdapt	✓	✓	27.8	8.4
	DCAN [52]			36.2	
	NonAdapt	✓	✓	30.0	8.1
	PTP [61]			38.1	
	NonAdapt	✓	✓	NA	NA
	AdaptSegNet [50]			35.0	
	NonAdapt	✓	✓	18.8	13.8
	DAM [21]			32.6	
	NonAdapt	✗	✓	30.0	8.6
	Ours			**38.6**	
	NonAdapt	✗	✗	29.8	6.3
	Ours			36.1	

Cityscapes to select the model which will be evaluated on BDDS/Mapillary. The G_{perf} value of each model is computed and the results are shown in Table 2. We can see that the proposed techniques can greatly boost the generalizability by 5%∼12% of different models regardless of dataset combinations.

Then we compare our method with the only known existing state-of-the-art domain generalization method for semantic segmentation IBN-Net [34] under the generalization setting from GTA to Cityscapes. From the comparison shown in Table 3, we can see that our domain generalization method has better final performance. IBN-Net improves domain generalization by fine-tuning the ResNet building blocks. Our method would be complementary with theirs.

4.5. Adaptation from GTA and SYNTHIA

All experiments in the sections above are conducted in the domain generalization setting, where the validation set and the test set are from different domains. Now we conduct more experiments using the domain adaptation setting and compare our results with previous state-of-the-art works. Since most of the previous works conducted adaptation to Cityscapes with VGG backbone networks, we present the adaptation mIoU comparison on GTA → Cityscapes and SYNTHIA → Cityscapes in Table 4 and Table 5, leaving class-wise comparison details in the supplementary material. We can see that our method could outperform the state-of-the-art methods in both settings. Further, we should notice that the domain generalization performance of our method (last row) outperforms the adaptation performance of most other techniques. In addition, since our method is target domain-agnostic, no data is needed from the target domain, resulting in more extensive applicability.

Table 5. Adaptation from SYNTHIA to Cityscapes with FCN-8s.

Network	Method	Train w/ Tgt	Val on Tgt	mIoU	mIoU↑
VGG-19	Non Adapt	✓	✓	22.0	7.0
	Curriculum [57]			29.0	
	Non Adapt	✓	✓	NA	NA
	CGAN [20]			41.2	
VGG-16	Non Adapt	✓	✓	17.4	2.8
	FCN Wld [19]			20.2	
	Non Adapt	✓	✓	25.4	10.8
	ROAD [3]			36.2	
	Non Adapt	✓	✓	26.8	9.3
	LSD [41]			36.1	
	Non Adapt	✓	✓	26.2	9.9
	CBST [63]			36.1	
	Non Adapt	✓	✓	27.8	8.4
	DCAN [52]			36.2	
	Non Adapt	✓	✓	NA	NA
	DAM [21]			30.7	
	Non Adapt	✓	✓	24.9	9.3
	PTP [61]			34.2	
	Non Adapt	✗	✓	27.3	9.1
	Ours			**36.4**	
	Non Adapt	✗	✗	26.8	8.7
	Ours			35.5	

5. Conclusion

In this paper, we present a domain generalization approach for generalizing semantic segmentation networks from simulation to the real world without accessing any target domain data. We propose to randomize the synthetic images with auxiliary datasets and enforce pyramid consistency across domains and within an image. Finally, we experimentally validate our method on a variety of experimental settings, and show superior performance over state-of-the-art methods in both domain generalization and domain adaptation, which clearly demonstrates the effectiveness of our proposed method.

Acknowledgement. This work was partially supported by NSF grants, award 1645964, and by the Berkeley Deep Drive center. We thank Kostadin Ilov for providing system assistance.

References

[1] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *Advances in Neural Information Processing Systems*, pages 1006–1016, 2018.

[2] Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor, Kurt Konolige, et al. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4243–4250. IEEE, 2018.

[3] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7892–7901, 2018.

[4] Dan Ciregan, Ueli Meier, and Jurgen Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, page 36423649, 2012.

[5] 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*, pages 3213–3223, 2016.

[6] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[7] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 5, page 6, 2018.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[9] Terrance DeVries and Graham W Taylor. Dataset augmentation in feature space. *arXiv preprint arXiv:1702.05538*, 2017.

[10] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[11] Tommaso Dreossi, Shromona Ghosh, Xiangyu Yue, Kurt Keutzer, Alberto Sangiovanni-Vincentelli, and Sanjit A Seshia. Counterexample-guided data augmentation. *arXiv preprint arXiv:1805.06962*, 2018.

[12] Chuang Gan, Tianbao Yang, and Boqing Gong. Learning attributes equals multi-source domain generalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 87–97, 2016.

[13] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015.

[14] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pages 2551–2559, 2015.

[15] Boqing Gong, Kristen Grauman, and Fei Sha. Reshaping visual datasets for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1286–1294, 2013.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European conference on computer vision*, pages 346–361, 2014.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle consistent adversarial domain adaptation. In *International Conference on Machine Learning (ICML)*, 2018.

[19] 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.

[20] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional generative adversarial network for structured domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, 2018.

[21] Haoshuo Huang, Qixing Huang, and Philipp Krahenbuhl. Domain transfer through deep activation matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 590–605, 2018.

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[23] Eric Kolve, Roozbeh Mottaghi, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

[24] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2169–2178, 2006.

[25] Joseph Lemley, Shabab Bazrafkan, and Peter Corcoran. Smart augmentation learning an optimal data augmentation strategy. *IEEE Access*, pages 5858–5869, 2017.

[26] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5542–5550, 2017.

[27] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[28] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In

Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5400–5409, 2018.

[29] Shaohua Li, Xinxing Xu, Liqiang Nie, and Tat-Seng Chua. Laplacian-steered neural style transfer. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 1716–1724. ACM, 2017.

[30] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[31] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *International Conference on Machine Learning*, pages 10–18, 2013.

[32] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. *arXiv preprint arXiv:1712.00479*, 13, 2017.

[33] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4990–4999, 2017.

[34] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *European Conference on Computer Vision*, pages 484–500, 2018.

[35] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *arXiv preprint arXiv:1710.06924*, 2017.

[36] Aayush Prakash, Shaad Boochoon, Mark Brophy, David Acuna, Eric Cameracci, Gavriel State, Omer Shapira, and Stan Birchfield. Structured domain randomization: Bridging the reality gap by context-aware synthetic data. *arXiv preprint arXiv:1810.10093*, 2018.

[37] 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, 2016.

[38] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016.

[39] Fereshteh Sadeghi and Sergey Levine. Cad2rl: Real single-image flight without a single real image. *arXiv preprint arXiv:1611.04201*, 2016.

[40] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. *arXiv preprint arXiv:1712.02560*, 3, 2017.

[41] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[42] Ikuro Sato, Hiroki Nishimura, and Kensuke Yokoi. Apac: Augmented pattern classification with neural networks. *arXiv preprint arXiv:1505.03229*, 2015.

[43] PY Simard, D Steinkraus, and JC Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Document Analysis and Recognition, 2003. Proceedings. Seventh International Conference on*, pages 958–963, 2003.

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

[45] Martin Sundermeyer, Zoltan-Csaba Marton, Maximilian Durner, Manuel Brucker, and Rudolph Triebel. Implicit 3d orientation learning for 6d object detection from rgb images. In *European Conference on Computer Vision*, pages 712–729, 2018.

[46] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 23–30, 2017.

[47] Toan Tran, Trung Pham, Gustavo Carneiro, Lyle Palmer, and Ian Reid. A bayesian data augmentation approach for learning deep models. In *Advances in Neural Information Processing Systems*, pages 2797–2806, 2017.

[48] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 969–977, 2018.

[49] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 969–977, 2018.

[50] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7472–7481, 2018.

[51] Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International Conference on Machine Learning*, pages 1058–1066, 2013.

[52] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gkhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S Davis. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. *arXiv preprint arXiv:1804.05827*, 2018.

[53] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: real-world perception for embodied agents. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[54] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018.

[55] Xiangyu Yue, Bichen Wu, Sanjit A Seshia, Kurt Keutzer, and Alberto L Sangiovanni-Vincentelli. A lidar point cloud generator: from a virtual world to autonomous driving. In *Proceedings on International Conference on Multimedia Retrieval*, pages 458–464, 2018.

[56] Yang Zhang, Philip David, Hassan Foroosh, and Boqing Gong. A curriculum domain adaptation approach to the semantic segmentation of urban scenes. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[57] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *IEEE International Conference on Computer Vision*, pages 2039–2049, 2017.

[58] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Fully convolutional adaptation networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6810–6818, 2018.

[59] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2881–2890, 2017.

[60] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE International Conference on Computer Vision*, 2017.

[61] Xinge Zhu, Hui Zhou, Ceyuan Yang, Jianping Shi, and Dahua Lin. Penalizing top performers: Conservative loss for semantic segmentation adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 568–583, 2018.

[62] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali 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.

[63] 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*, pages 289–305, 2018.