Supplementary Material

Learning from Synthetic Data: Addressing Domain Shift for Semantic Segmentation

1. Architecture and Hyperparameters

The details of the architectures used in our experiments are shown in the Figure. 1. As described in Section 3 of the main paper, the entire pipeline of our approach consists of 4 networks - F, C, G and D networks. We perform experiments with two architectures for $F - C$ pair: FCN-8s and Deeplab-Resnet-101. For FCN-8s, we denote the network till $fc7$ layer as F network, and the final classification layers are treated as C network. For Deeplab-Resnet-101, the network till $res5c$ layer is denoted as F network, the rest of the layers are treated as C network.

The architectures of G and D networks are described in Figure. 1. The G network is a multi-stage network - it accepts inputs from intermediate layers of the F network to generate the reconstructed image. We observed that fusing information from the earlier layers of F network produced good quality generations compared to using only the response of the final layer. For both G and D networks, we use residual blocks inspired by the recent success of several generative models [32]

There are two hyper-parameters in our approach: α - the weight of auxiliary classification loss and β - the weight of adversarial component (Refer to Section 3 in the main paper). We observed that the network is not very sensitive to these parameters. We found that the parameter setting $\alpha = 0.1$ and $\beta = 0.1$ worked well across all settings, and used it for all our experiments.

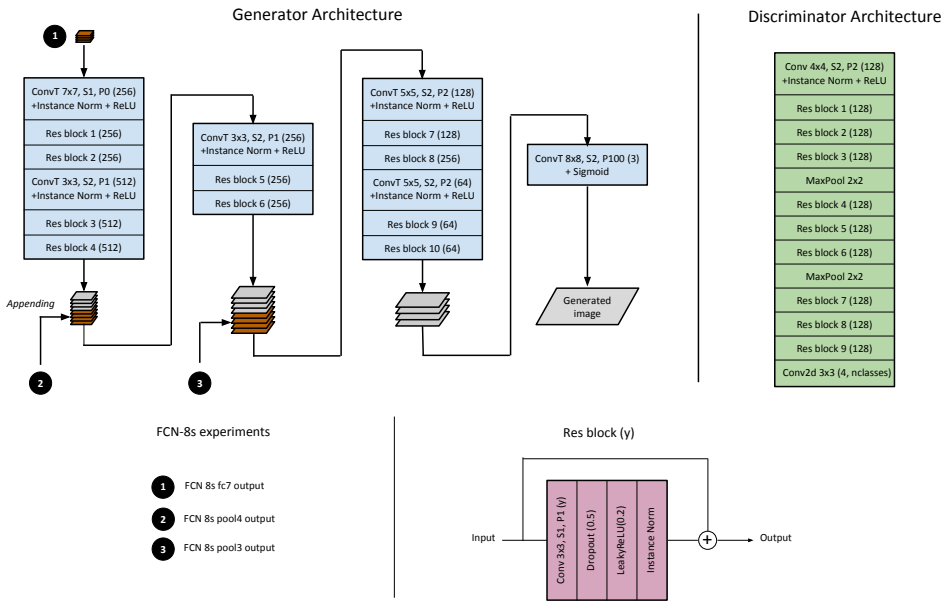

Figure 1: Details of the network architectures used in our experiments. Conv - Convolution layer, ConvT - Transposed convolution layer, S - stride, P - padding. For each Conv/ConvT layer, the numbers in the parenthesis denote the number of filters.

2. Generator Visualizations

In figures 2 and 3, we present examples of the generator reconstructions of source and target images during the course of the training procedure. This provides a visual representation of the domain shift happening in the generator space; this is especially clear in the source domain images which look more realistic as training progresses.

Figure 2: Querying the generator space for source images - Progress across iterations

Figure 3: Querying the generator space for target images - Progress across iterations

3. Qualitative Comparison of Label predictions

The label map visualization of the segmentation results obtained by the baseline model and our adapted model are shown in the Figure. 4. We observe that our adapted model improves the quality of the label map predictions significantly compared to the source-only model. This improvement is predominant for classes that occupy the major portion of the image like *road* and *car*.

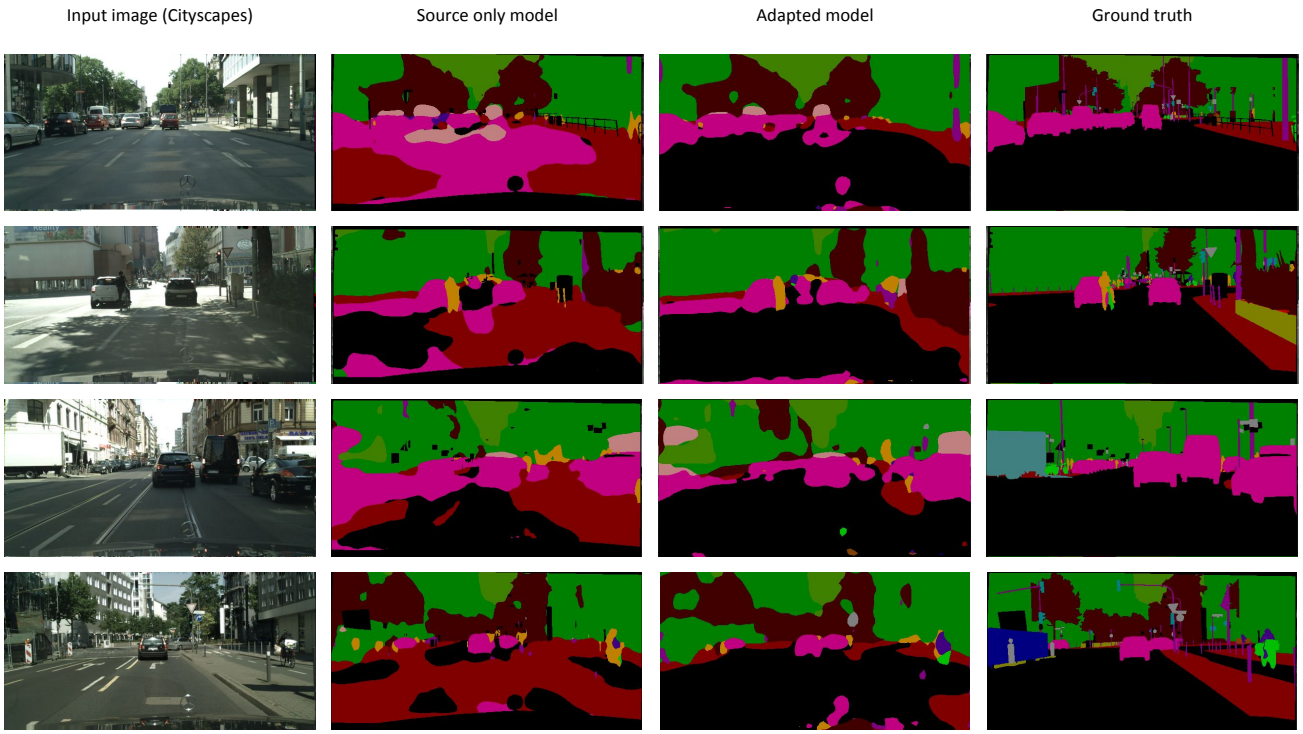

Figure 4: Visualization of label map predictions for SYNTHIA $\rightarrow$ CITYSCAPES experiment. In each row, the first column corresponds to the input image sampled from the target domain (Cityscapes). The second and the third column corresponds to the segmentation results of the baseline model (source-only model) and our adapted model respectively. The last column corresponds to the ground truth

4. Common Classes for CamVid experiment

For the CamVid experiment in Section 5.5 from the original paper, we chose the 10 common classes between the CamVid, SYNTHIA and CITYSCAPES datasets. To begin with, we would like to note that for the results presented in Table 2a of the main paper for the SYNTHIA $\rightarrow$ CITYSCAPES setting, 16 common classes among the two datasets were chosen following previous works. Now, for the CamVid experiment, we choose 10 among these 16 classes which are common with the CamVid dataset. They are the following: *building*, *vegetation*, *t sign*, *sky*, *car*, *road*, *person*, *fence*, *pole*, *sidewalk*.