Supplementary Material: Source-Free Domain Adaptation for Semantic Segmentation

Yuang Liu , Wei Zhang*, Jun Wang*
East China Normal University, Shanghai, China
{frankliu624, zhangwei.thu2011, wongjun}@gmail.com

1. Network Architectures

The details of the network architectures of the used generator and discriminator are shown in Table 1.

Generator. We construct our generator following [2]. Specifically, we stack 4 transposed convolutional layers and a convolution layer, and each transposed convolution layer is followed by a leaky ReLU parameterized by 0.2. The input latent dimension of the generator is set to 256, which is the same as the label embedding in the discriminator.

Discriminator. The discriminator network consists of 4 convolutional layers with a kernel size of 4×4 and stride of 2, where the channel numbers are 64, 128, 256, and 64, respectively. Except for the last layer, each convolutional layer is followed by a leaky ReLU parameterized by 0.2. Besides, a label embedding layer is employed at the 4-th layer to extend the discriminator to a conditional version. The last two layers are full-connected layers followed by ReLU and Sigmoid, respectively.

2. Training Scheme

Algorithm 1 illustrates the training scheme of our framework SFDA.

Algorithm 1: Training scheme of SFDA.

Input : Target dataset $D_t = \{x_t | x_t \in \mathbb{R}^{H\times W\times 3}\}$, well-trained source model $\mathcal{S}$, generator $\mathcal{G}$ and discriminator $\mathcal{D}$.
Output : Adapted target model $\mathcal{T}$.

Copy a parameter-fixed source model $\tilde{\mathcal{S}}$ from $\mathcal{S}$, the target model $\mathcal{T}$ shares parameters with $\mathcal{S}$;
for *number of training epochs* **do**
  `// Knowledge Transfer Stage`
  $\mathcal{G}$ synthesizes fake source sample $\tilde{x}_s$ by Eq.(3);
  Forward $\tilde{x}_s$ in $\mathcal{S}$ and $\tilde{\mathcal{S}}$;
  Calculate BNS loss by Eq.(4);
  Calculate MAE and DAD loss by Eq.(5∼7);
  Update $\mathcal{G}$ by Eq.(11);
  `// Model Adaptation Stage`
  Forward a batch of target data x_t in $\mathcal{T}$;
  Calculate target loss by Eq.(2);
  Obtain features of hard and easy patch groups by Eq.(13∼14);
  for *number of patches* **do**
    Train $\mathcal{D}$ with easy patches;
    Update $\mathcal{D}$ by Eq.(16);
    Discriminate hard patches by $\mathcal{D}$;
    Calculate adversarial loss by Eq.(15);
  end
  Update $\mathcal{T}$ and $\mathcal{S}$ by Eq.(16);
end

3. More Experimental Results and Analysis

3.1. Results of Adaptation

We present the qualitative results on SYNTHIA → Cityscapes and Cityscapes → NTHU (including 4 cities: Rome, Rio, Tokyo and Taipei) in Figure 1 and 2, respectively. Since the gap between Cityscapes [1] and NTHU [3] is relatively small, the comparison before and after adaptation is not very obvious. But we can still find the role of IPSM in promoting local segmentation.

We calculate and compare the entropy maps gotten from the SYNTHIA → Cityscapes task. The brighter area has higher confidence of prediction and accuracy. As Figure 3 shows, the entropy maps after adaptation are clearer and more consistent in each class.

3.2. More Analysis of IPSM

To visually demonstrate the principle and effect of IPSM, we track a batch of target images during Cityscapes → Rio adaptation training and calculate mean pixel entropy of 9 patches ($K = 3$) respectively, which are plotted by different colors in Figure 4. Comparing the entropy value of each

*Corresponding author.

Generator	Discriminator
FC, reshape, BN (4,2,1) 512 ConvTrans$\uparrow_{2\times}$, BN, LReLU	(4,2,1) 64 Conv, LReLU (4,2,1) 128 Conv, BN, LReLU
(4,2,1) 256 ConvTrans$\uparrow_{2\times}$, BN, LReLU (4,2,1) 256 ConvTrans$\uparrow_{2\times}$, BN, LReLU	(4,2,1) 256 Conv, BN, LReLU (4,2,1) 128 Conv, BN, LReLU | Label Embedding
(4,2,1) 128 ConvTrans$\uparrow_{2\times}$, BN, LReLU (3,1,1) 3 Conv, Tanh	Concat, FC, ReLU FC, Sigmoid

Table 1. Generator and discriminator architectures. The three numbers in brackets represent kernel size, stride and padding respectively.

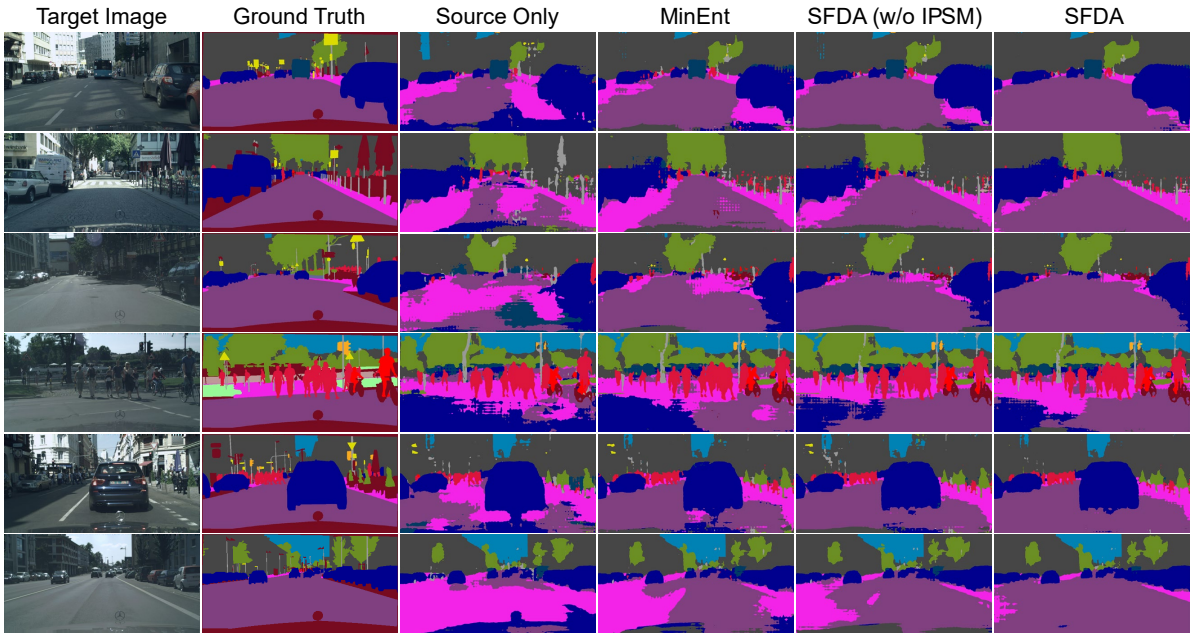

Figure 1. Qualitative results for SYNTHIA → Cityscapes.

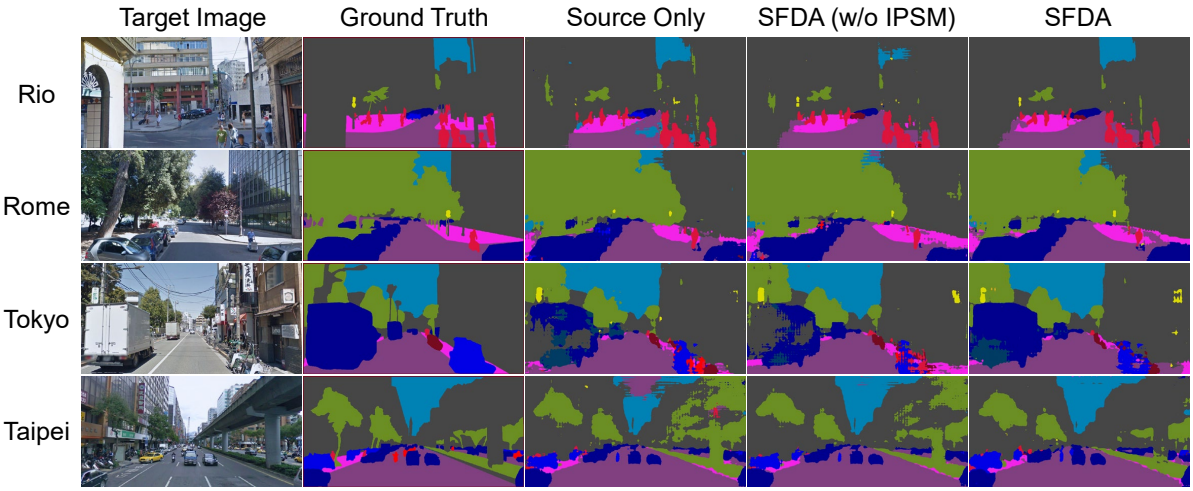

Figure 2. Qualitative results for Cityscapes → NTHU.

patch or each sample (1 to 8) in different training stages, we can see that IPSM can effectively leverage intra-domain information to improve the prediction confidence and accuracy in patches.

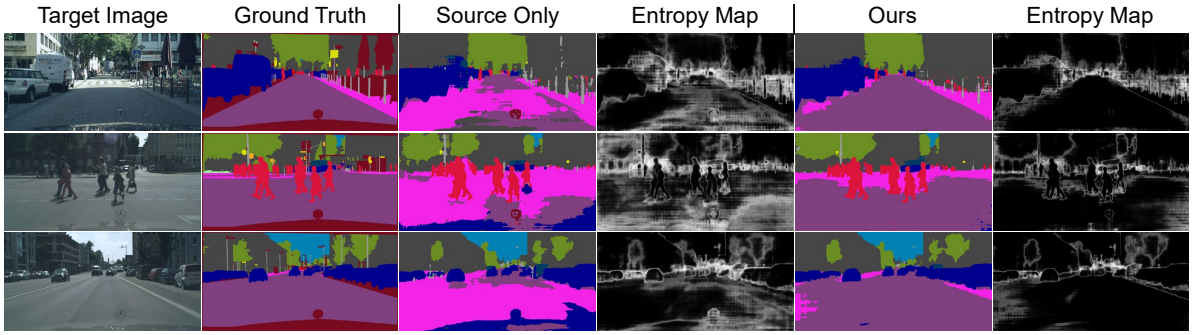

Figure 3. Entropy maps of SYNTHIA → Cityscapes.

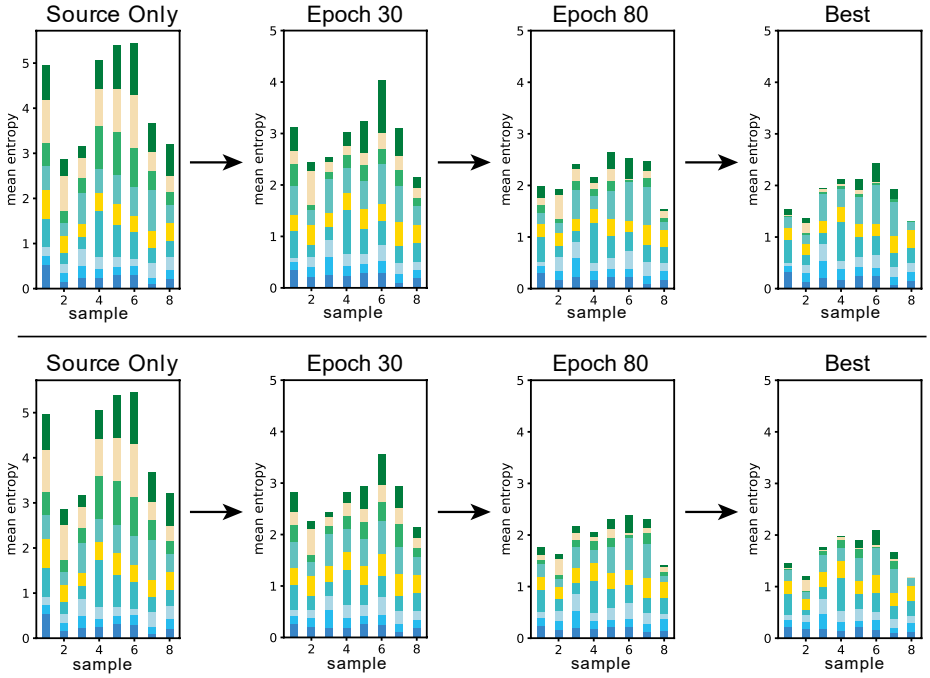

Figure 4. Patch entropy w.r.t. GTA5 → Cityscapes. The upper is trained without IPSM, while the lower is trained with IPSM.

References

[1] 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 *CVPR*, pages 3213–3223, 2016.

[2] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[3] Ching-Hua Weng, Ying-Hsiu Lai, and Shang-Hong Lai. Driver drowsiness detection via a hierarchical temporal deep belief network. In *ACCV*, pages 117–133. Springer, 2016.