Self-Ensembling with GAN-based Data Augmentation for Domain Adaptation in Semantic Segmentation: Supplementary material

Jaehoon Choi
KAIST

Taekyung Kim
KAIST

Changick Kim
KAIST

{whdns44, tkkim93, changick}@kaist.ac.kr

A. Experiment results on BDD100K dataset

In this section, we report extra experimental results, in which experiments on BDD100K [10] is involved. BDD100K is a large-scale driving dataset collected from the United States, under various driving circumstances, weather conditions, and lighting conditions. It has a training set with 7000 images, and a validation set with 1000 images. In our experiment, we used 19 categories of the annotations, which are compatible with the classes of Cityscapes. Since BDD100K is a new dataset compared to GTA5 and SYNTHIA in the main text, there are only a few works [2] that study domain adaptation for semantic segmentation. Our method achieves state-of-the-art segmentation performance. We report IoU for each class and mIoU in Table 1. Compared to the baseline, our method improves the mIoUs by 10.9 %.

B. Implementation Details

B.1. TGCF-DA

At first, we pretrain FCN-VGG16 [7] using the labeled source data. With this pretrained segmentation model f_{seg}, we train the augmentation network for Target-Guided and Cycle-Free Data Augmentation (TGCF-DA). We adapt the generator and discriminator architectures from MUNIT [5] and multi-scale discriminators with 3 scales in [9]. We apply spectral normalization [8] to the weights in discriminator for training stability. Following an auto-encoder structure in MUNIT [5], the number of residual blocks in the generator is set to eight, and four adaptive instance normalization (AdaIN) layers [4] are added to the last four residual blocks. The source encoder and target encoder includes three strided convolutional layers to downsample the source and target images, respectively. Additionally, the target encoder has two fully connected layers after convolutional layers to produce the learnable AdaIN parameters. Decoder consists of three transposed convolutional layers. For the semantic constraint, we fix the weight of semantic constraint to 10. We run 100k iterations using Adam [6] with learning rate 1e-4, $\beta_1 = 0.5$, and $\beta_2 = 0.999$.

B.2. Self-Ensembling

We train the segmentation network for 200K iterations using the Adam [6] gradient descent with a learning rate 1e-5 and momentum 0.9. In one batch, we use two labeled source data, two labeled augmented data, and four unlabeled target data. DeepLab [1] based on VGG-16 is used as the base model of the teacher and student network. We use the prediction maps produced from the ASPP (Atrous Spatial Pyramid Pooling) with four atrous rates (r = 6, 12, 18, 24) to compute the consistency loss. We add two dropout layers after 'fc6' and 'fc7' layers.

C. Comparison to other Image-to-Image translation methods

Figure. 3 shows example images of GTA5 synthesized in the style of Cityscapes (Target) from CycleGAN, UNIT, MUNIT, and Ours (TGCF-DA). Figure. 4 shows example images of SYNTHIA synthesized in the style of Cityscapes (Target) from CycleGAN, UNIT, MUNIT, and Ours (TGCF-DA).

D. Additional experiment results of hyperparameter sensitivity on TGCF-DA

We present additional experiment results of hyperparameter sensitivity on TGCF-DA. When $\lambda_{seg} = 50$ in Fig. 5, generated images are almost replica of the source images. When λ_{seg} is too large, the GAN loss is ignored during the training process. Thus, the generator fails to learn the style representations of target images. When $\lambda_{seg} = 1$ in Fig. 5, the augmentation network is not able to preserve objects in image. Also, we can observe the artifacts like road texture in the sky or tree. As we mentioned in the main text, we find that the augmentation network fails to maintain the semantic consistency without the semantic constraint. Since we apply multiple adaptive instance normalizations (AdaIN) [4] to the source feature maps like MUNIT [5], the

Cityscapes → BDD																					
Method	Backbone	road	sidewalk	building	wall	fence	pole	light	sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorbike	bike	**mIoU**
Baseline (Source Only)	V	71.9	28.9	59.1	6.2	23.5	23.0	25.2	26.4	63.7	24.9	85.3	19.8	14.7	67.9	9.9	12.8	0.0	21.9	15.2	31.6
Strategic curriculum [2]	R	87.9	39.8	75.0	**15.3**	**24.6**	29.1	0.4	23.1	77.5	24.2	**87.0**	53.7	9.3	79.6	0.0	**36.4**	0.0	0.0	**29.8**	36.7
Ours (TGCF-DA + SE)	V	**90.2**	**51.5**	**81.1**	15.0	10.7	**37.5**	**35.2**	**28.9**	**84.1**	**32.7**	75.9	**62.7**	**19.9**	**82.6**	**22.9**	28.3	**0.0**	**23.0**	25.4	**42.5**
Target Only	V	90.9	65.9	82.5	44.1	29.9	45.1	41.3	46.2	78.5	47.8	93.5	52.5	29.0	79.8	56.5	21.2	4.4	36.1	0.0	49.7

Table 1. The semantic segmentation results on BDD100K validation set when evaluating the model trained on GTA5. "Source Only" denotes the evaluation result of models only trained on source data. "Target Only" denotes the segmentation results in supervised settings. The backbone "R" and "V" stand for ResNet-18 [3] and VGG-16.

augmentation network tends to match the style of the dominant class in the target images with the content of different classes in the source image. The semantic constraint guides the augmentation network to learn the semantic contents of different categories and preserve the semantic consistency.

References

[1] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[2] Kashyap Chitta, Jianwei Feng, and Martial Hebert. Adaptive semantic segmentation with a strategic curriculum of proxy labels. *arXiv preprint arXiv:1811.03542*, 2018.

[3] 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.

[4] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[5] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 172–189, 2018.

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[7] 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.

[8] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[9] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018.

[10] 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.

Figure 1. Qualitative segmentation results of GTA5 → Cityscapes. From left to right: Original image, ground truth annotation, NoAdapt baseline, results of our method.

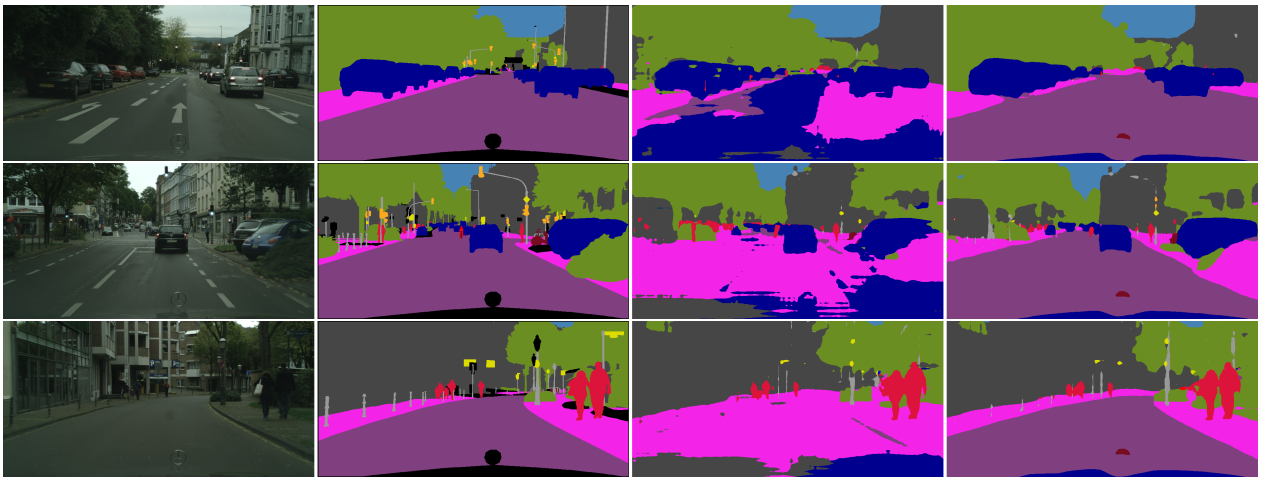

Figure 2. Qualitative segmentation results of SYNTHIA → Cityscapes. From left to right: Original image, ground truth annotation, NoAdapt baseline, results of our method.

Figure 3. Example images of GTA5 synthesized in the style of Cityscapes with CycleGAN, UNIT, and MUNIT.

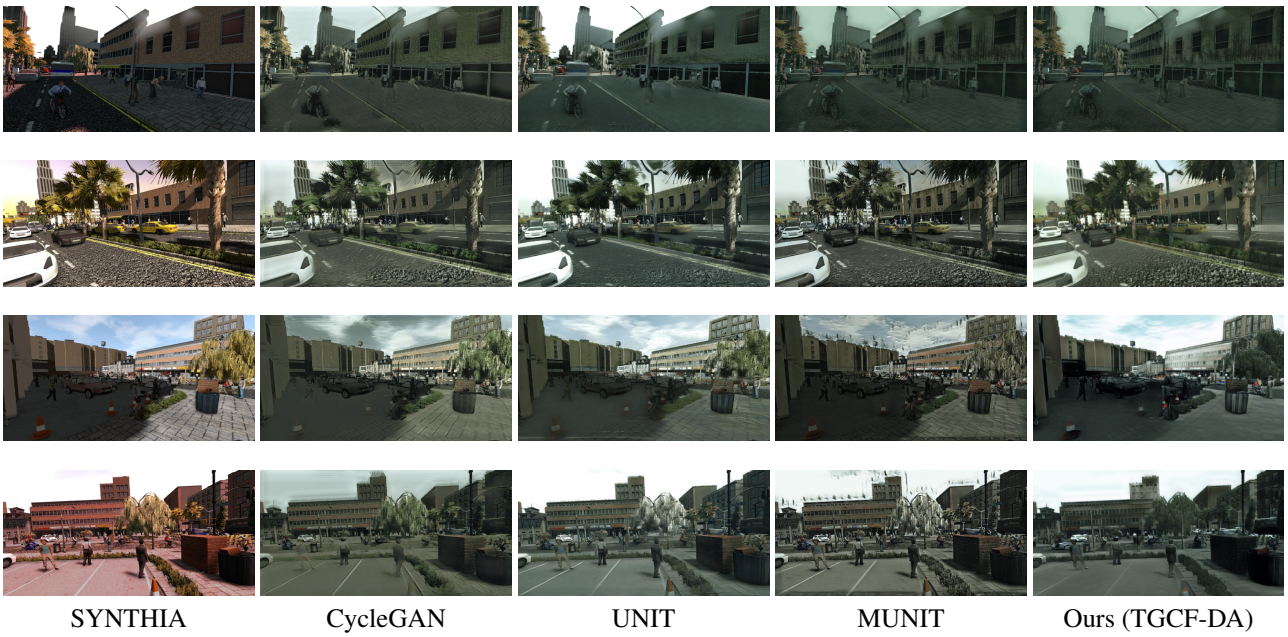

Figure 4. Example images of SYNTHIA synthesized in the style of Cityscapes with CycleGAN, UNIT, and MUNIT.

Figure 5. Hyperparameter sensitivity on TGCF-DA. From left to right: source input, target input, output with λ_{seg} = 1, output with λ_{seg} = 10, output with λ_{seg} = 100.