Appendix: Model Adaptation

We provide supplementary information for the main paper here. We first introduce the network architectures used in our experiments. Next, additional experimental results are presented and discussed.

1. Detailed Architecture

For the digit and sign benchmarks, we apply spectral normalization [6] (SN) to each layer of D for training stability, and leaky-ReLU is adopted as the activation function except for the last dense layer. The UpResBlock in G is a similar architecture as the one used in improved WGAN [3], in which the Upsample layer contains a nearest-neighbor interpolation and a convolutional layer for avoiding checkerboard artifacts [7]. The corresponding architectures of D and G are shown in Table 1.

For the office-31 and VisDA-17 datasets, we adopt the standard protocols used in [10, 9, 5]. ResNet50 and ResNet101 pre-trained on ImageNet [1] are used to extract corresponding features. Following MCD [10], the final dense layer of the classifier C is replaced by two dense layers. The number of hidden layer neurons is set to 1024 in both experiments. Accordingly, the generator G and discriminator D consist of two dense layers, and ReLU and leaky-ReLU are used as activation functions, respectively.

Discriminator (D)	Generator (G)
Input: $x \in \mathbb{R}^{32\times32\times3}$	**Input:** y (one-hot label), z (noise)
	Concate $[z, y]$
Conv.(k4n128s2), SN, LReLU(0.2)	Dense $4 \times 4 \times 512$
Conv.(k4n256s2), SN, LReLU(0.2)	UpResBlock (256)
Conv.(k4n512s2), SN, LReLU(0.2)	UpResBlock (128)
	UpResBlock (64)
Flatten	BN, ReLU
Dense 1	Conv.(k3n3s1), Tanh
Output: Probability of Real/Fake	**Output:** Generated Images

Table 1. Architectures of the discriminator D and generator G used in the digit and sign benchmarks. 'SN' denotes spectral normalization [6] and 'BN' denotes batch normalization [4]. k, n and s denote kernel size, the number of filters and stride in each convolutional layer (Conv.), respectively.

2. Toy Example

We further use a toy dataset to demonstrate the collaborative behavior of the generator G and the classifier C during the adaptation process. Similar to DANN [2], we use scikit-learn [8] to generate two interleaving moons as the source dataset which has 500 samples per class, and the target dataset is synthesized by rotating the source dataset by 60 degrees. All the classifier, generator and discriminator consist of three dense layers, and we set the number of neurons to be 300 for all the hidden layers.

In Figure 1, we visualize the decision boundary of our proposed method during each stage of adaptation. As shown in Figure 1(a), the Source-Only model delivers good performance on the source data (red and green samples), but suffers on the target data (blue samples).

The adaptation process is illustrated in Figure 1(b) (from 1 to 10). The yellow and purple points denote the generated samples of class 0 and 1, respectively. We observe that during training, the generated data can match the target distribution with reliable labels. As the adaptation proceeds, the generated data can progressively guide the decision boundary of the prediction model to correctly separate the target samples, and it is clear that the class conditional generation becomes more accurate with the prediction model becoming better on the target domain. Therefore, collaboration between the generator and the prediction model are expected during the adaptation process.

As shown in Figure 1(c), the decision boundary of the adapted model correctly classifies all the target samples. The generated data points can match the target domain well, which indicates that the generator reliably learns the class conditional target data distribution.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 1

[2] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17:59:1–59:35, 2016. 1

[3] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. In *NeurIPS*, pages 5769–5779, 2017. 1

[4] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 1

[5] Vinod Kumar Kurmi, Shanu Kumar, and Vinay P. Namboodiri. Attending to discriminative certainty for domain adaptation. In *CVPR*, pages 491–500, 2019. 1

[6] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018. 1

[7] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. 1

[8] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Edouard Duchesnay. Scikit-learn: Machine learning in python. *JMLR*, 12:2825–2830, 2011. 1

[9] Zhongyi Pei, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. Multi-adversarial domain adaptation. In *AAAI*, pages 3934–3941, 2018. 1

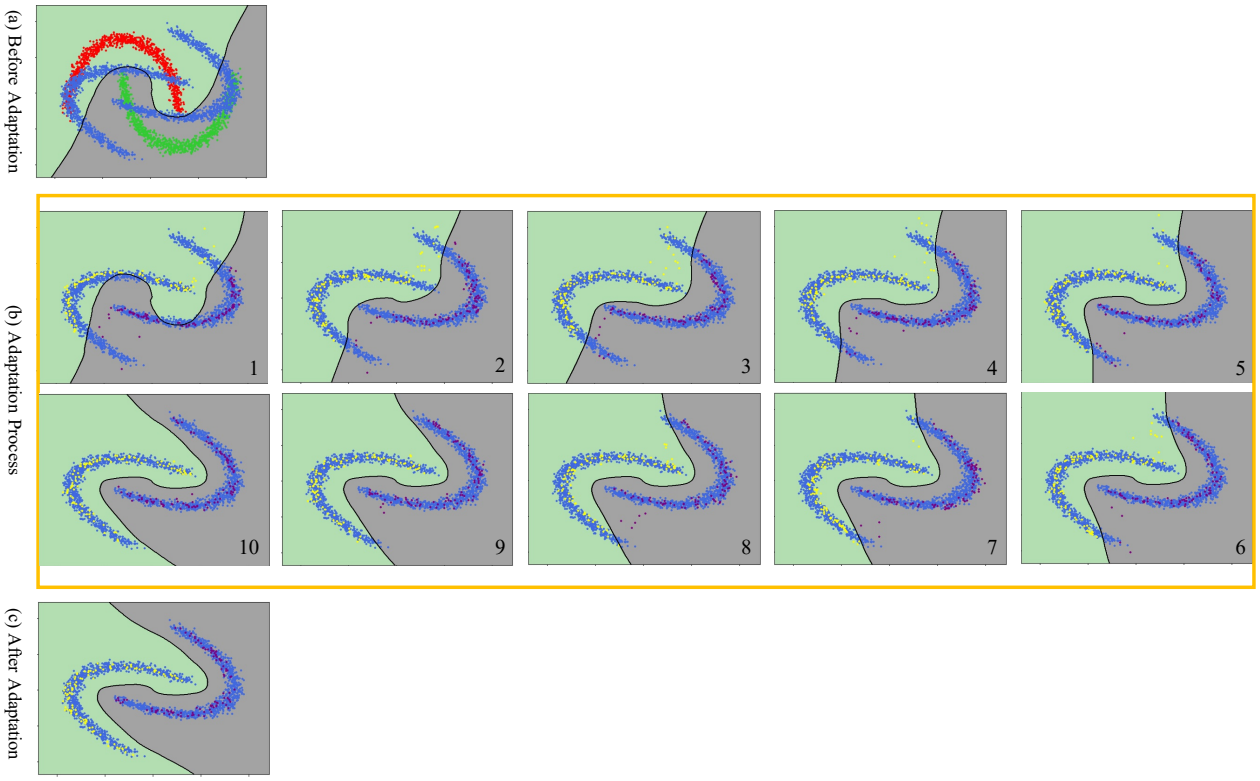

Figure 1. (Best viewed in color.) Behaviors of 3C-GAN during model adaptation on the two intertwining moons problems. (a) presents the model before adaptation, which is trained on the source dataset. The red and green points indicate the source samples with class 0 and 1, respectively. The blue points indicate the unlabeled target samples, which are generated by rotating the source data by 60 degrees. During the adaptation process (b), source dataset is not used. The yellow and purple points indicate the generated samples with class 0 and 1, respectively. Model adaptation proceeds according to the order from 1 to 10 as denoted in the bottom right corner. (c) shows the results of the adapted model.

[10] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *CVPR*, pages 3723–3732, 2018. 1