Geometric Unsupervised Domain Adaptation for Semantic Segmentation Supplementary Material

Vitor Guizilini Jie Li Rareș Ambruș Adrien Gaidon
Toyota Research Institute (TRI), Los Altos, CA
{first.lastname}@tri.global

1. Network Architectures

In Tab. 1 we describe in details the shared depth and semantic segmentation network used in our experiments. This architecture is based on recent developments in monocular depth estimation [8]. Note that our proposed algorithm can be generalized to any multi-scale backbones. We leave the exploration of architectures more suitable to jointly predict semantic segmentation [2, 3, 14] and monocular depth [8, 7] for future work. For the shared backbone we use a ResNet101 [9] encoder, that produces feature maps with varying number of channels at increasingly lower resolutions (#1, #2, #3, #4, #5 in Tab. 1). These feature maps are used as skip connections for both the depth and the semantic segmentation decoders, through a series of convolutional layers followed by bilinear upsampling. For the depth decoder, at the final four upsampling stages (#10, #13, #16, #19 in Tab. 1) an inverse depth layer is used to produce estimates within a minimum and maximum depth range:

$$\frac{1}{d_{u,v}} = \frac{1}{d_{max}} + \left(\frac{1}{d_{min}} - \frac{1}{d_{max}}\right) Sigmoid(f_{u,v}) \quad (1)$$

All four scales are used to calculate the self-supervised photometric loss (with results averaged per-batch, per-scale and per-pixel), and only the final scale is used to calculate the supervised depth loss. During inference, only the final scale is used as depth prediction estimates. The semantic network is similar, with the difference that the outputs at each of the upsampling stages (#9, #11, #13, #15 in Tab. 1) are instead concatenated (after bilinear upsampling to the highest resolution) and processed using a final convolutional layer to produce a C-dimensional logits vector for each pixel.

Our pose network is described in Tab. 2, and follows closely [8]. It uses a ResNet18 backbone as encoder, followed by four convolutional layers with 256 channels. Finally, a global pooling layer outputs a 6-dimensional vector, containing (x, y, z) translation and *(roll, pitch, yaw)* rotation. We have experimented with a shared encoder for depth, semantic segmentation and pose, however as pointed out in [8] performance degraded in this configuration.

	Layer Description	Out. Dimension
	RGB image	3×H×W
	ResNet101 Encoder	
#1	Intermediate Features #1	256×H/2×W/2
#2	Intermediate Features #2	256×H/4×W/4
#3	Intermediate Features #3	512×H/8×W/8
#4	Intermediate Features #4	1024×H/16×W/16
#5	Latent Features	2048×H/32×W/32
	Depth Decoder	
#6	Conv2d (#5) → ELU → Upsample	256×H/16×W/16
#7	Conv2d (#6 ⊕ #4) → ELU	256×H/16×W/16
#8	Conv2d (#7) → ELU → Upsample	128×H/8×W/8
#9	Conv2d (#8 ⊕ #3) → ELU	128×H/8×W/8
#10	Conv2d (#9) → InvDepth	1×H/8×W/8
#11	Conv2d (#9) → ELU → Upsample	64×H/4×W/4
#12	Conv2d (#11 ⊕ #2) → ELU	64×H/4×W/4
#13	Conv2d (#12) → InvDepth	1×H/4×W/4
#14	Conv2d (#12) → ELU → Upsample	32×H/2×W/2
#15	Conv2d (#14 ⊕ #1) → ELU	32×H/2×W/2
#16	Conv2d (#15) → InvDepth	1×H/2×W/2
#17	Conv2d (#15) → ELU → Upsample	16×H×W
#18	Conv2d (#17) → ELU	16×H×W
#19	Conv2d (#18) → InvDepth	1×H×W
	Semantic Decoder	
#6	Conv2d (#5) → ELU → Upsample	256×H/16×W/16
#7	Conv2d (#6 ⊕ #4) → ELU	256×H/16×W/16
#8	Conv2d (#7) → ELU → Upsample	128×H/8×W/8
#9	Conv2d (#8 ⊕ #3) → ELU	128×H/8×W/8
#10	Conv2d (#9) → ELU → Upsample	64×H/4×W/4
#11	Conv2d (#10 ⊕ #2) → ELU	64×H/4×W/4
#12	Conv2d (#11) → ELU → Upsample	32×H/2×W/2
#13	Conv2d (#12 ⊕ #1) → ELU	32×H/2×W/2
#14	Conv2d (#13) → ELU → Upsample	16×H×W
#15	Conv2d (#14) → ELU	16×H×W
#16	Conv2d (#9 ⊕ #11 ⊕ #13 ⊕ #15)	C×H×W

Table 1: **Depth and semantic segmentation multi-task network**. We use a ResNet101 backbone as encoder, that outputs intermediate features at different resolutions. These intermediate features are used as skip connections in different stages of the semantic and depth decoders. ELU are Exponential Linear Units [5], *Upsample* denotes bilinear interpolation, *InvDepth* is an inverse depth layer (Eq. 1), and ⊕ denotes feature concatenation.

Figure 1: **The *Parallel Domain* dataset**: sample images.

	Layer Description	Out. Dimension
	2 Stacked RGB images	6×H×W
	ResNet18 Encoder	
#1	Latent Features	256×H/8×W/8
	Pose Decoder	
#2	Conv2d → ReLU	256×H/8×W/8
#3	Conv2d → ReLU	256×H/8×W/8
#4	Conv2d → ReLU	256×H/8×W/8
#5	Conv2d → Global Pooling	6

Table 2: **Pose network**. Two concatenated RGB images are used as input for a ResNet18 encoder (the first convolutional layer is duplicated to account for that). The output is a 6-dimensional vector estimating the rigid transformation between frames (translation and rotation in Euler angles).

2. Parallel Domain

This dataset is procedurally generated using the *Parallel Domain* synthetic data generation service [1]. It contains 5000 10-frame sequences, for a total of 50000 frames. Each frame consists of an RGB image from a front-facing vehicle-mounted camera along with associated per-pixel depth and semantic segmentation labels. The dataset consists of urban and highway environments with varying number of agents, time of day, and weather conditions. We present reference images from the dataset in Fig. 1. Each image is rendered with a 1936×1216 resolution. The high degree of fidelity and perceptual quality allows us to investigate the following questions: (i) how does the quality of the simulation affect the *sim-to-real* domain gap; and (ii) can we decrease the *sim-to-real* domain gap with additional synthetic data. As reported in the main paper, Tab. 1 and Fig. 7, we conclude that high quality synthetic data can indeed help narrow the *sim-to-real* gap, and the gap is further narrowed as additional data is made available.

3. Qualitative Results

In Fig. 2 we present semantic pointclouds estimated using GUDA+PL for unsupervised domain adaptation from *Parallel Domain* to *Cityscapes*. Because our multi-task network (Tab. 1) produces both depth and semantic segmentation estimates, we can lift the predicted semantic labels to

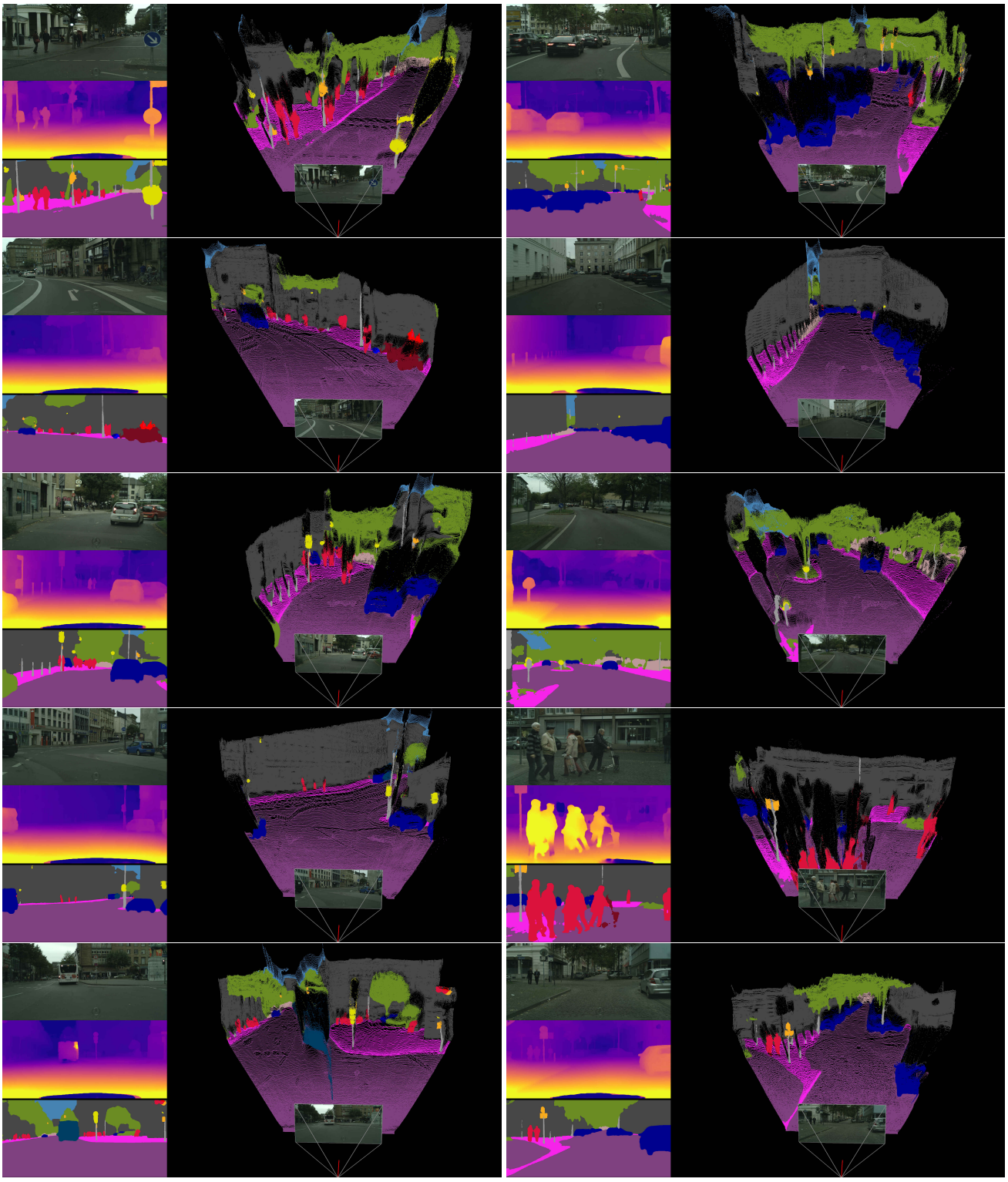

Figure 2: **Qualitative depth and semantic segmentation results**, using GUDA+PL to perform unsupervised domain adaptation from *Parallel Domain* to *Cityscapes*. The same multi-task network was used to generate depth and semantic segmentation estimates, that were combined into a 3D pointcloud using camera intrinsics. No real-world labels (depth or semantic) were used during training.

Method	Road	S.walk	Build.	Wall*	Fence*	Pole*	T.Light	T.Sign	Vegt.	Sky	Person	Rider	Car	Bus	Motor.	Bike	mIoU	mIoU*
Source (SY)	70.2	35.0	74.7	2.1	0.2	27.8	1.7	4.4	76.9	83.4	44.4	9.9	51.3	7.9	4.0	12.8	31.7	36.7
Source (PD)	85.5	39.4	70.6	0.0	0.8	37.6	25.4	11.9	79.9	80.9	47.0	25.0	70.1	10.7	9.8	15.3	38.1	44.0
Target	97.1	82.9	90.6	47.3	51.7	57.1	60.8	72.5	91.6	93.3	75.8	54.3	93.4	77.5	48.5	71.9	72.9	77.8
(a) Comparison with other depth-based UDA methods (SYNTHIA → Cityscapes)																		
SPIGAN [11]	71.1	29.8	71.4	3.7	0.3	33.2	6.4	15.6	81.2	78.9	52.7	13.1	75.9	25.5	10.0	20.5	36.8	42.4
GIO-Ada [4]	78.3	29.2	76.9	11.4	0.3	26.5	10.8	17.2	81.7	81.9	45.8	15.4	68.0	15.9	7.5	30.4	37.3	43.0
DADA [17]	**89.2**	44.8	**81.4**	6.8	0.3	26.2	8.6	11.1	**81.8**	84.0	54.7	**19.3**	79.7	**40.7**	**14.0**	**38.8**	42.6	49.8
GUDA	85.4	**49.5**	80.8	**13.8**	**0.9**	**36.2**	**21.8**	**35.2**	78.8	**84.7**	**59.9**	13.5	**84.0**	33.8	2.8	30.9	**44.5**	**50.9**
(b) Comparison with other UDA methods (SYNTHIA → Cityscapes)																		
Xu et al. [18]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	38.8	—
CLAN [13]	81.3	37.0	80.1	—	—	—	16.1	13.7	78.2	81.5	53.4	21.2	73.0	32.9	22.6	30.7	—	47.8
CBST [23]	53.6	23.7	75.0	12.5	0.3	36.4	23.5	26.3	84.8	74.7	67.2	17.5	84.5	28.4	15.2	55.8	42.5	48.4
CRST [22]	67.7	32.2	73.9	10.7	1.6	37.4	22.2	**31.2**	80.8	80.5	60.8	29.1	82.8	25.0	19.4	45.3	43.8	50.1
ESL[15]	84.3	39.7	79.0	9.4	0.7	27.7	16.0	14.3	78.3	83.8	59.1	26.6	72.7	35.8	23.6	45.8	43.5	50.7
FDA [20]	79.3	35.0	73.2	—	—	—	19.9	24.0	61.7	82.6	61.4	31.1	83.9	40.8	**38.4**	51.1	—	52.5
CCMD [12]	79.6	36.4	80.6	13.3	0.3	25.5	22.4	14.9	81.8	77.4	56.8	25.9	80.7	45.3	29.9	52.0	45.2	52.6
Yang et al. [19]	85.1	44.5	81.0	—	—	—	16.4	15.2	80.1	84.8	59.4	31.9	73.2	41.0	32.6	44.7	53.1	
USAMR [21]	83.1	38.2	81.7	9.3	1.0	35.1	30.3	19.9	82.0	80.1	62.8	21.1	84.4	37.8	24.5	53.3	46.5	53.8
IAST [10]	81.9	41.5	83.3	17.7	**4.6**	32.3	**30.9**	28.8	**83.4**	**85.0**	65.5	**30.8**	86.5	38.2	33.1	52.7	49.8	57.0
GUDA+PL	**88.1**	**53.0**	**84.0**	**22.0**	1.4	**39.6**	28.2	24.8	82.7	81.5	**65.5**	22.7	**89.3**	**50.5**	25.1	**57.5**	**51.0**	**57.9**
(c) Comparison with the state of the art (Varying Sources → Cityscapes)																		
UDAS [16]	86.6	37.8	80.8	29.7	16.4	28.9	30.9	22.2	37.1	76.9	60.1	7.8	84.1	32.1	23.2	13.3	44.3	49.2
USAMR (G5) [21]	90.5	35.0	84.6	34.3	24.0	36.8	44.1	42.7	84.5	82.5	63.1	34.4	85.8	38.2	27.1	41.8	53.1	58.0
IAST (G5) [10]	**94.1**	**58.8**	85.4	**39.7**	**29.2**	25.1	43.1	34.2	84.8	**88.7**	62.7	30.3	87.6	50.3	**35.2**	40.2	55.6	61.2
GUDA(PD)+PL(G5)	92.9	50.5	**86.0**	17.9	24.0	**45.4**	**50.9**	**44.5**	**87.7**	87.0	**66.6**	**36.9**	**89.5**	**52.1**	28.5	**54.0**	**57.2**	**63.2**

Table 3: **Semantic segmentation results on *Cityscapes*** using different unsupervised domain adaptation (UDA) methods and synthetic datasets. The *mIoU* metric considers all 16 classes, and *mIoU** considers only the 13 classes present in SYNTHIA (removing the ones marked with *). *Source* shows results without any adaptation, and *Target* shows results with semantic supervision on the target domain. Synthetic datasets include: *SYNTHIA (SY)*, *Parallel Domain (PD)*, and *GTA5 (G5)*.

Method	Road	Building	Pole	T. Light	T. Sign	Vegetat.	Terrain	Sky	Car	Truck	**mIoU**
Source	64.9	28.3	37.8	18.8	11.7	63.7	21.6	78.7	55.3	1.5	38.6
DANN	70.3	49.4	39.5	28.0	22.2	67.0	23.1	82.0	69.4	5.1	45.6
GUDA	**86.8**	**72.7**	**46.2**	**41.4**	**44.6**	**77.3**	**29.1**	**88.5**	**86.1**	**9.8**	**58.25**

Table 4: **Semantic segmentation results on *VKITTI2* → *KITTI***, using GUDA and DANN [6].

Method	Road	S.walk	Build.	Pole	T.Light	T.Sign	Vegetat.	Sky	Person	Rider	Car	Truck	Bus	Motor.	Bike	**mIoU**
Source	93.9	30.7	49.3	35.7	50.7	20.8	87.2	89.3	10.0	28.7	63.2	38.4	14.3	8.5	7.3	41.9
DANN	95.3	36.1	53.0	35.6	52.8	20.7	88.3	90.3	15.2	38.7	67.5	44.1	36.5	19.5	11.1	47.0
GUDA	**96.1**	**48.0**	**58.9**	**37.1**	**55.8**	**22.0**	**89.6**	**93.0**	**30.6**	**54.8**	**70.8**	**47.2**	**58.7**	**41.6**	**29.6**	**55.6**

Table 5: **Semantic segmentation results on *Parallel Domain* → *DDAD***, using GUDA and DANN [6].

3D space using depth estimates and camera intrinsics. Each pixel is assigned a 3D coordinate in the camera frame of reference, as well as RGB colors and semantic logits. We emphasize that no real-world labels (depth maps or semantic classes) were used at any point during the training of this network, only image sequences. All labeled information was obtained from synthetic datasets, and adapted to better align with real-world data using our proposed GUDA approach to geometric unsupervised domain adaptation.

4. Detailed Tables

We also present detailed tables to complement some results from the main paper. In particular, Table 3 expands Table 1 from the main paper, showing per-class results on the *Cityscapes* dataset of the various methods we use as comparison to validate the improvements of our proposed GUDA approach. Similarly, Tables 4 and 5 expand Figures 5 and 6 from the main paper, showing respectively GUDA results from our *VKITTI2* to *KITTI* and *PD* to *DDAD* experiments relative to *source-only* and *DANN* [6].

References

[1] Parallel domain. https://paralleldomain.com/, March 2021. 2

[2] 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. *CoRR*, 2016. 1

[3] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, 2016. 1

[4] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4

[5] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *ICLR*, 2016. 1

[6] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 17(1):2096–2030, Jan. 2016. 4, 5

[7] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017. 1

[8] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. In *ICCV*, 2019. 1

[9] 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. 1

[10] Jiaqi Zou Ke Mei, Chuang Zhu and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In *European Conference on Computer Vision (ECCV)*, 2020. 4

[11] Kuan-Hui Lee, German Ros, Jie Li, and Adrien Gaidon. SPIGAN: privileged adversarial learning from simulation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. 4

[12] Guangrui Li, Guoliang Kang, Wu Liu, Yunchao Wei, and Yi Yang. Content-consistent matching for domain adaptive semantic segmentation. In *European Conference on Computer Vision (ECCV)*, 2020. 4

[13] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 4

[14] George Papandreou, Liang-Chieh Chen, Kevin Murphy, and Alan L Yuille. Weakly- and semi-supervised learning of a dcnn for semantic image segmentation. In *ICCV*, 2015. 1

[15] Antoine Saporta, Tuan-Hung Vu, M. Cord, and P. Pérez. Esl: Entropy-guided self-supervised learning for domain adaptation in semantic segmentation. *ArXiv*, 2020. 4

[16] Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A Efros. Unsupervised domain adaptation through self-supervision. *arXiv preprint arXiv:1909.11825*, 2019. 4

[17] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Mathieu Cord, and Patrick Pérez. Dada: Depth-aware domain adaptation in semantic segmentation. In *ICCV*, 2019. 4

[18] Jiaolong Xu, Liang Xiao, and Antonio M López. Self-supervised domain adaptation for computer vision tasks. *IEEE Access*, 7:156694–156706, 2019. 4

[19] J. Yang, Weizhi An, S. Wang, Xin liang Zhu, Chao chao Yan, and Junzhou Huang. Label-driven reconstruction for domain adaptation in semantic segmentation. In *ECCV*, 2020. 4

[20] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 4

[21] Zhedong Zheng and Yi Yang. Unsupervised scene adaptation with memory regularization in vivo. In *IJCAI*, 2020. 4

[22] Yang Zou, Zhiding Yu, Xiaofeng Liu, B.V.K. Vijaya Kumar, and Jinsong Wang. Confidence regularized self-training. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 4

[23] 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 (ECCV)*, pages 289–305, 2018. 4