Adversarially Masking Synthetic to Mimic Real: Adaptive Noise Injection for Point Cloud Segmentation Adaptation
–Supplementary Materials–

Guangrui Li[1,2*], Guoliang Kang[3†], Xiaohan Wang[4], Yunchao Wei[5], Yi Yang[4]
[1] ReLER, AAII, University of Technology Sydney [2]Baidu Inc. [3] Beihang University
[4] ReLER, CCAI, Zhejiang University [5] Beijing Jiaotong University.
www.guangrui.li/projects/ASM

In the following sections, we first present more quantitative results for a better comparison against previous methods in Appendix A. Then in Appendix B, we show more qualitative results to enable more intuitive understanding and comparisons. Finally, in Appendix C, we provide more implementation details.

A. More Quantitative Results

Table A. Experiments results of GTA-LiDAR [6] → SemKITTI [1] with SqueezeSegV2 as backbone.

Methods	Car	Pedestrian	mIoU
SqueezeSegV2 [6]	63.2	12.8	38.0
ePointDA [8]	70.7	12.9	41.8
Ours	**73.1**	**13.3**	**43.2**

Table B. Comparison with CosMix following their setup on SynLiDAR [7] → SemKITTI [1].

Method	**car**	**bi.cle**	**mt.cle**	**truck**	**oth-v.**	**pers.**	**b.clst**	**m.clst**	**road**	**park.**	**sidew.**	**oth-g.**	**build.**	**fence**	**veget.**	**trunk**	**terra.**	**pole**	**traff.**	**mIoU**
ADDA [4]	52.5	4.5	11.9	0.3	3.9	9.4	27.9	0.5	52.8	4.9	27.4	0.0	61.0	17.0	57.4	34.5	42.9	23.2	4.5	23.0
Ent-Min [5]	58.3	5.1	14.3	0.3	1.8	14.3	**44.5**	0.5	50.4	4.3	34.8	0.0	48.3	19.7	67.5	34.8	**52.0**	33.0	6.1	25.8
ST [9]	62.0	5.0	12.4	1.3	9.2	16.7	44.2	0.4	53.0	2.5	28.4	0.0	57.1	18.7	69.8	**35.0**	48.7	32.5	6.9	26.5
PCT [7]	53.4	5.4	7.4	0.8	10.9	12.0	43.2	0.3	50.8	3.7	29.4	0.0	48.0	10.4	68.2	33.1	40.0	29.5	6.9	23.9
ST-PCT [7]	70.8	**7.3**	13.1	1.9	8.4	12.6	44.0	0.6	56.4	4.5	31.8	0.0	**66.7**	**23.7**	**73.3**	34.6	48.4	**39.4**	11.7	28.9
CosMix [3]	75.1	6.8	29.4	**27.1**	**11.1**	**22.1**	25.0	24.7	**79.3**	14.9	46.7	**0.1**	53.4	13.0	67.7	31.4	32.1	37.9	**13.4**	32.2
Ours	**75.8**	**7.3**	**34.6**	26.8	10.8	21.3	40.3	**25.1**	60.4	**18.3**	**48.1**	**0.1**	58.4	14.3	72.3	33.3	40.2	36.6	8.2	**33.3**

Here we present the quantitative comparisons with previous methods on more benchmarks.

In Table A, following the setting of ePointDA [8], we run our method on the transfer GTA-LiDAR → SemKITTI and compare with previous solutions. Apparently, our method still maintains its superiority against them, further proving the effectiveness of the proposed approach.

Then in Table B, we run experiments following the configuration of CoSMix [3], and our method surpasses CoSMix by 1.1 % mIoU, validating a better adaptation performance on both benchmarks.

Besides, in Fig. A, we compare the bar plots of per-class IoUs to provide a more intuitive illustration of the gain we made against the baseline.

*Work done during an internship at Baidu.
†corresponding author

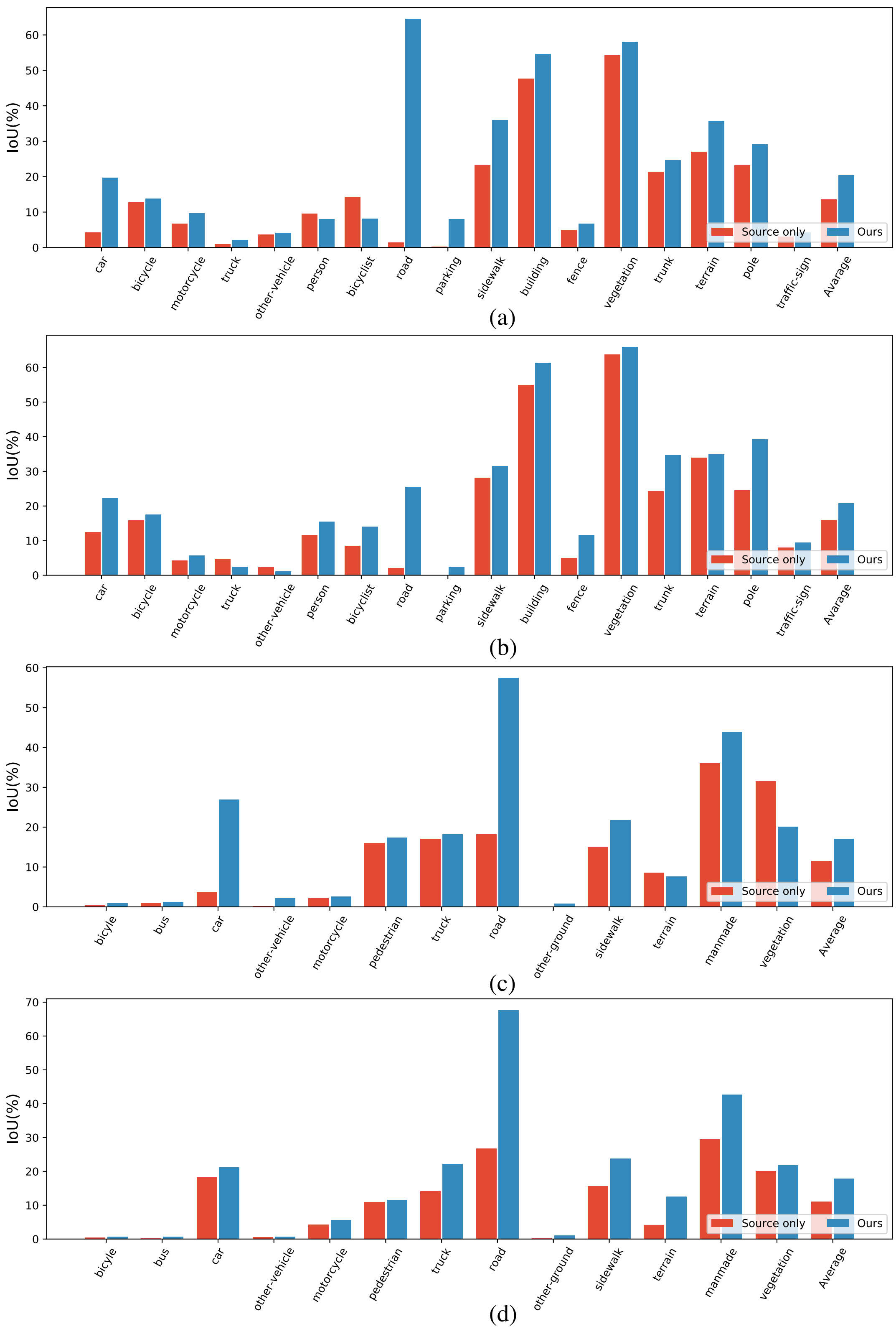

Figure A. **Comparison on Per-class IoU with the baseline.** SynLiDAR → SemKITTI with SqueezeSegV2 (a) and SalsaNext (b), and SynLiDAR → nuScenes with SqueezeSegV2 (c) and SalsaNext (d).

B. More Qualitative Analysis

In Fig. B and Fig. C, we present more visualizations regarding both raw points and projected LiDAR images, and compare our results with the source-only model, AdaptSeg, and the ground truth.

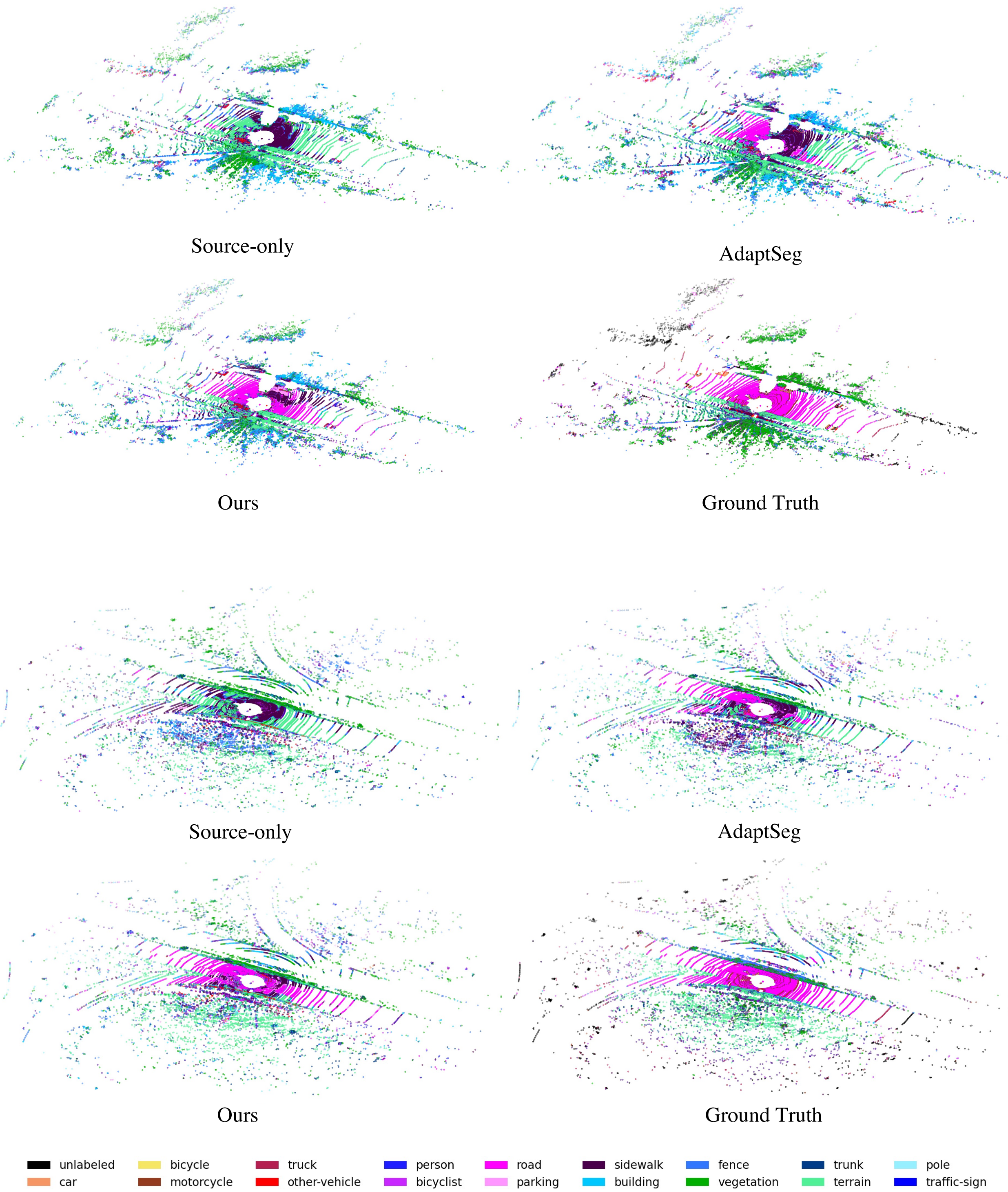

Figure B. **Visualization results for SynLiDAR → SemKITTI.**

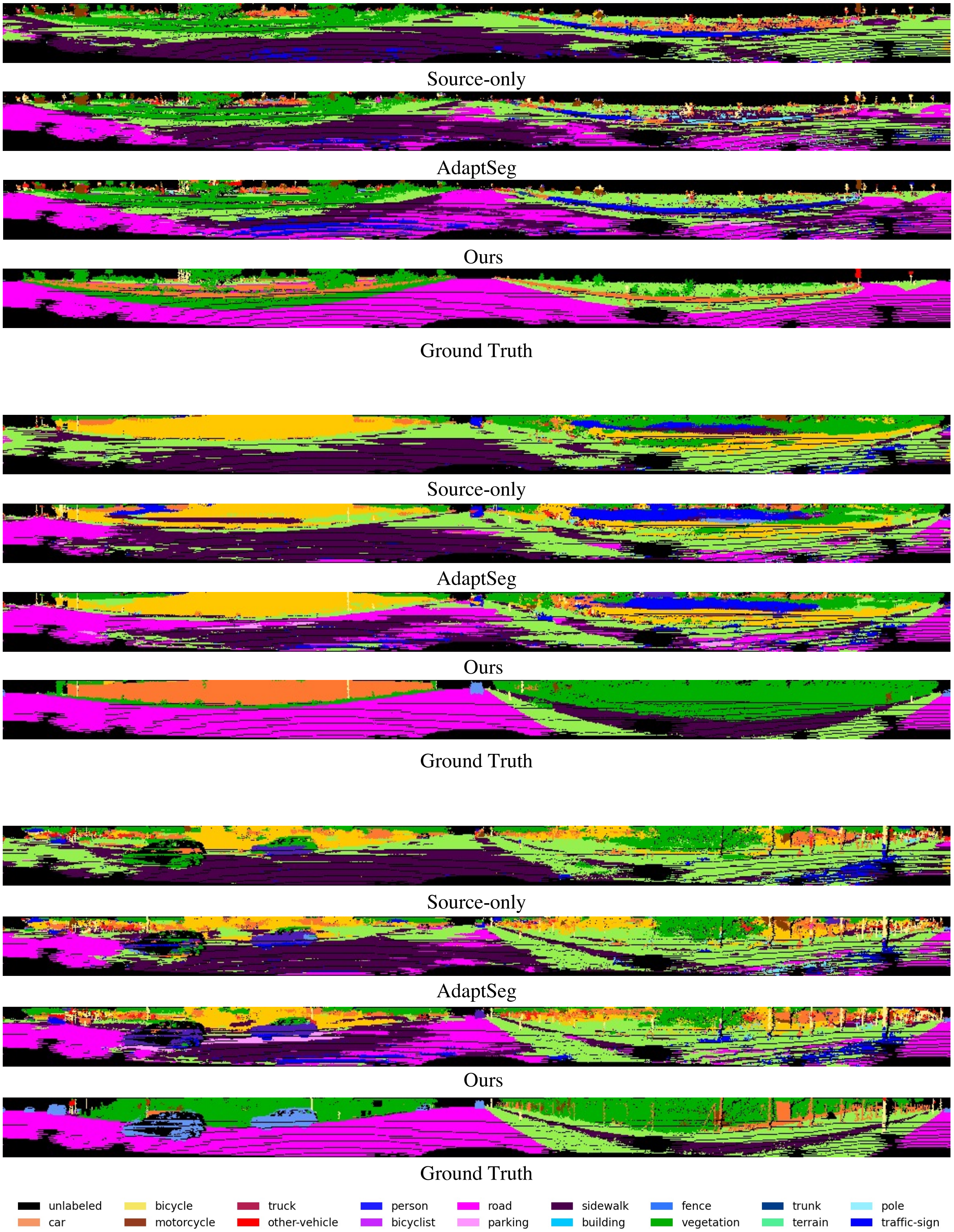

Figure C. **Visualization results for SynLiDAR → SemKITTI with projected LiDAR images.**

C. More Implementation Details

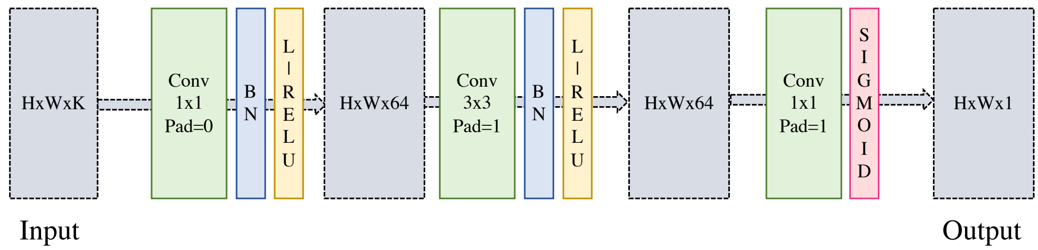

Figure D. **Architecture of the Domain Discriminator**, where "L-RELU" denotes the leaky-relu with slope of 0.1 and "BN" denotes BatchNorm layer.

Architecture of the Domain Discriminator. In Fig. D, we present the architecture of the domain discriminator, which consists of three convolution layers and several non-linear layers. Note that we employ BatchNorm (BN) after each convolution layer because it can stablize the training process. We use "L-RELU" to denote Leaky-ReLU with slope of 0.1.

Algorithm 1 PyTorch-style Pseudocode for Adaptive Spatial Masking (ASM) Module

```
import torch.nn as nn
import torch.nn.functional as F
class ASM_module(nn.Module):
    def __init__(self, ):
        super().__init__()
        mid = 32
        # Embedding Branch e
        self.conv1 = nn.Conv2d(3, mid, kernel_size=1, stride=1)
        self.bn1 = nn.BatchNorm2d(mid)
        # Embedding Branch o
        self.conv2 = nn.Conv2d(32, mid, kernel_size=1, stride=1)
        self.bn2 = nn.BatchNorm2d(mid)
        # Fusion Head
        self.conv3 = nn.Conv2d(mid*2, 2, kernel_size=1, stride=1)
    def forward(self, coord, feat):
        # Embed coordinates and features accordingly.
        ori_feat = feat
        cod = self.conv1(coord)
        cod = self.bn1(cod)
        cod = torch.relu(cod)
        feat = self.conv2(feat)
        feat = self.bn2(feat)
        feat = torch.relu(feat)
        # Fusion with concatenation
        fused = torch.cat([cod, feat], dim=1)
        fused = self.conv3(fused)
        # Apply Gumbel-Softmax
        fused = F.gumbel_softmax(fused, hard=True, dim=1)
        mask = fused[:, :1, :, :]
        feat = ori_feat * mask
        return feat
```

Implementation of Adaptive Spatial Masking Module. In Algorithm 1, we present the implementation of ASM module to enable a better understanding of the proposed method.

Label Mappings. In Table C and Table D, we present label mappings for two transfer tasks respectively.

Table C. Label Mappling for SynLIDAR → SemKITTI

SynLIDAR [7]				SemKITTI [1]			
ID	Name	Mapped ID	Mapped Name	ID	Name	Mapped ID	Mapped Name
0	unlabeled	0	unlabeled	0	unlabeled	0	unlabeled
1	car	1	car	1	outlier	0	unlabeled
2	pick-up	4	truck	10	car	1	car
3	truck	4	truck	11	bicycle	2	bicycle
4	bus	5	other-vehicle	13	bus	5	other-vehicle
5	bicycle	2	bicycle	15	motorcycle	3	motorcycle
6	motorcycle	3	motorcycle	16	on-rails	5	other-vehicle
7	other-vehicle	5	other-vehicle	18	truck	4	truck
8	road	8	road	20	other-vehicle	5	other-vehicle
9	sidewalk	10	sidewalk	30	person	6	person
10	parking	9	parking	31	bicyclist	7	bicyclist
11	other-ground	9	parking	32	motorcyclist	3	motorcycle
12	female	6	person	40	road	8	road
13	male	6	person	44	parking	9	parking
14	kid	6	person	48	sidewalk	10	sidewalk
15	crowd	6	person	49	other-ground	9	parking
16	bicyclist	7	bicyclist	50	building	11	building
17	motorcyclist	3	motorcycle	51	fence	12	fence
18	building	11	building	52	other-structure	0	unlabeled
19	other-structure	0	unlabeled	60	lane-marking	8	road
20	vegetation	13	vegetation	70	vegetation	13	vegetation
21	trunk	14	trunk	71	trunk	14	trunk
22	terrain	15	terrain	72	terrain	15	terrain
23	traffic-sign	17	traffic-sign	80	pole	16	pole
24	pole	16	pole	81	traffic-sign	17	traffic-sign
25	traffic-cone	0	unlabeled	99	other-object	0	unlabeled
26	fence	12	fence	252	moving-car	1	car
27	garbage-can	0	unlabeled	253	moving-bicyclist	7	bicyclist
28	electric-box	0	unlabeled	254	moving-person	6	person
29	table	0	unlabeled	255	moving-motorcyclist	8	road
30	chair	0	unlabeled	256	moving-on-rails	5	other-vehicle
31	bench	0	unlabeled	257	moving-bus	5	other-vehicle
32	other-object	0	unlabeled	258	moving-truck	4	truck
				259	moving-other-vehicle	5	other-vehicle

Table D. Label Mappling for SynLIDAR → nuScenes

SynLIDAR [7]				nuScenes [2]			
ID	Name	Mapped ID	Mapped Name	ID	Name	Mapped ID	Mapped Name
0	unlabeled	0	unlabeled	1	animal	0	unlabeled
1	car	3	car	5	human·pedestrian·personal mobility	0	unlabeled
2	pick-up	7	truck	7	human·pedestrian·stroller	0	unlabeled
3	truck	7	truck	8	human·pedestrian·wheelchair	0	unlabeled
4	bus	2	bus	10	movable object·debris	0	unlabeled
5	bicycle	1	bicycle	11	movable object·pushable pullable	0	unlabeled
6	motorcycle	5	motorcycle	13	static object·bicycle rack	0	unlabeled
7	other-vehicle	4	other vehicles	19	vehicle·emergency·ambulance	4	other vehicles
8	road	8	road	20	vehicle·emergency·police	4	other vehicles
9	sidewalk	10	sidewalk	0	unlabeled	0	unlabeled
10	parking	9	other ground	29	static·other	0	unlabeled
11	other-ground	9	other ground	31	vehicle·ego	0	unlabeled
12	female	6	pedestrian	9	movable object·barrier	12	manmade
13	male	6	pedestrian	14	vehicle·bicycle	1	bicycle
14	kid	6	pedestrian	15	vehicle·bus·bendy	2	bus
15	crowd	6	pedestrian	16	vehicle·bus·rigid	2	bus
16	bicyclist	1	bicycle	17	vehicle·car	3	car
17	motorcyclist	5	motorcycle	18	vehicle·construction	4	other vehicles
18	building	12	manmade	21	vehicle·motorcycle	5	motorcycle
19	other-structure	12	manmade	2	human·pedestrian·adult	6	pedestrian
20	vegetation	13	vegetation	3	human·pedestrian·child	6	pedestrian
21	trunk	7	truck	4	human·pedestrian·construction worker	6	pedestrian
22	terrain	11	terrain	6	human·pedestrian·police officer	6	pedestrian
23	traffic-sign	12	manmade	12	movable object·trafficcone	12	manmade
24	pole	12	manmade	22	vehicle·trailer	4	other vehicles
25	traffic-cone	12	manmade	23	vehicle·truck	7	truck
26	fence	12	manmade	24	flat·driveable surface	8	road
27	garbage-can	12	manmade	25	flat·other	9	other ground
28	electric-box	12	manmade	26	flat·sidewalk	10	sidewalk
29	table	0	unlabeled	27	flat·terrain	11	terrain
30	chair	0	unlabeled	28	static·manmade	12	manmade
31	bench	12	manmade	30	static·vegetation	13	vegetation
32	other-object	0	unlabeled				

References

[1] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *ICCV*, 2019. 1, 6

[2] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *CVPR*, 2020. 7

[3] Cristiano Saltori, Fabio Galasso, Giuseppe Fiameni, Nicu Sebe, Elisa Ricci, and Fabio Poiesi. Cosmix: Compositional semantic mix for domain adaptation in 3d lidar segmentation. *ECCV*, 2022. 1

[4] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017. 1

[5] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Mathieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*, 2019. 1

[6] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *ICRA*, 2019. 1

[7] Aoran Xiao, Jiaxing Huang, Dayan Guan, Fangneng Zhan, and Shijian Lu. Synlidar: Learning from synthetic lidar sequential point cloud for semantic segmentation. *AAAI*, 2022. 1, 6, 7

[8] Sicheng Zhao, Yezhen Wang, Bo Li, Bichen Wu, Yang Gao, Pengfei Xu, Trevor Darrell, and Kurt Keutzer. epointda: An end-to-end simulation-to-real domain adaptation framework for lidar point cloud segmentation. In *AAAI*, 2021. 1

[9] Yang Zou, Zhiding Yu, Xiaofeng Liu, B.V.K. Vijaya Kumar, and Jinsong Wang. Confidence regularized self-training. In *ICCV*, 2019. 1