Supplementary Material
GIPSO: Geometrically Informed Propagation for Online Adaptation in 3D LiDAR Segmentation

Cristiano Saltori[1], Evgeny Krivosheev[1], Stéphane Lathuilière[2], Nicu Sebe[1], Fabio Galasso[3], Giuseppe Fiameni[4], Elisa Ricci[1,5], and Fabio Poiesi[5]

[1] University of Trento, Trento, Italy
[2] LTCI, Télécom-Paris, Intitute Polytechnique de Paris, Palaiseau, France
[3] Sapienza University of Rome, Rome, Italy
[4] NVIDIA AI Technology Center
[5] Fondazione Bruno Kessler, Trento, Italy
cristiano.saltori@unitn.it

1 Introduction

We provide supplementary material in support of the main paper. The content is organized as follows:

- Sec. 2 reports the architecture details of the main modules used in GIPSO;
- Sec. 3 provides additional ablations of GIPSO, analysing the performance with a different propagation size and time-window length;
- Sec. 4 goes beyond GIPSO and shows that our proposed strategies can be used to improve baselines in SF-OUDA;
- Sec. 5 reports the class mapping used in our experiments for compatibility between synthetic and real domains;
- In Sec. 6, additional qualitative results are reported on Synth4D → SemanticKITTI, SynLiDAR → SemanticKITTI, and Synth4D → nuScenes.

2 Architecture details

We implemented GIPSO in PyTorch by using minkowski/sparse convolutions in MinkowskiEngine [4]. For the backbone and segmentation network we used the existing implementation of MinkUNet18 [4] by setting the dimension of the input space to $D = 3$, *i.e.* the dimensionality of an input point cloud. For the self-supervised temporal consistency loss (Sec. 4, Eq. 6) we implemented the encoder $h()$ with two consecutive MinkowskiConvolution layers interleaved by a ReLU activation function and a batch-normalization layer. The input size of the first layer is set to 96 - the output feature size of the backbone network - while the output size is set to 128. The last encoding layer is set to have the same input and output size of 128. We implemented the predictor $f()$ with the same structure of $h()$ with the difference that input and output sizes are set to 128. In both $h()$ and $f()$ we used a kernel of size 1, biases activated and $D = 3$.

Table 1. Online adaptation on Synth4D → SemanticKITTI with different propagation size K.

Model	K	vehicle	pedestrian	road	sidewalk	terrain	manmade	vegetation	Avg
Source	-	22.54	14.38	42.03	28.39	15.58	38.18	54.14	30.75
Target	-	+3.76	+0.92	+9.41	+16.95	+19.79	+10.92	+10.71	+10.35
Ours	1	+14.18	-1.13	+1.08	+2.11	+2.74	+5.49	+5.39	+4.27
Ours	5	+13.42	-0.51	+0.91	+2.16	+2.66	+5.54	+5.62	+4.26
Ours	10	+13.12	-0.54	+1.19	+2.45	+2.78	+5.64	+5.54	+4.31
Ours	50	+12.01	-1.00	+0.73	+2.01	+3.02	+5.51	+5.66	+3.99
Ours	100	+12.25	-2.49	+0.62	+1.93	+3.39	+5.99	+5.68	+3.91

3 GIPSO components

We provide two additional ablation studies to complement the ablation study in the main manuscript in Sec. 5.4. We perform an ablation study for different components of GIPSO on Synth4D → SemanticKITTI. Sec. 3.1 reports the results when the propagation size K is increased up to 100 for each seed pseudo-label. Sec. 3.2 reports how GIPSO performs by varying the time window w. Results report the performance on Source (gray) in absolute mIoU while the others are reported as relative mIoU improvement over the Source model. Target is the supervised upper bound of our task in our setting.

3.1 Propagation size

We study the effect of different propagation steps by using our geometry-based propagation. Tab. 1 shows the results with a K of $1, 5, 10, 50, 100$. We can see that mIoU starts to decrease when a higher number of propagation steps are used, i.e., $K = 50$, whereas we reach the best improvement of +4.31 with $K = 10$. These results show that K should be set such that to both preserve pseudo-labelling accuracy while propagating seed labels towards new informative points.

3.2 Time-window length

We study the effect of different time window length w in our self-supervised temporal consistency loss. Tab. 2 shows that w should be selected neither too large ($w = 8$) nor too small ($w = 1$) for the best performance. The time window w should be set based on the sampling rate of the sensor and the overlap between adjacent frames.

4 Improving state-of-the-art with GIPSO

We show that our proposed modules also improve state-of-the-art methods, such as CBST [8], ProDA [7] and, TPLD [7], providing additional evidence that our propositions are steps forward in SF-OUDA not just in GIPSO. First, we show that our adaptive sampling strategy can be used in state-of-the-art methods to obtain more reliable pseudo-labels. Second, we propose modifications

Table 2. Online adaptation on Synth4D → SemanticKITTI with a different time window w.

Model	w	vehicle	pedestrian	road	sidewalk	terrain	manmade	vegetation	Avg
Source	-	22.54	14.38	42.03	28.39	15.58	38.18	54.14	30.75
Target	-	+3.76	+0.92	+9.41	+16.95	+19.79	+10.92	+10.71	+10.35
Our	1	+9.73	-0.63	+0.56	+1.79	+2.86	+4.88	+4.27	+3.35
Our	2	+11.76	-1.09	+0.78	+1.97	+2.50	+5.01	+5.23	+3.74
Our	3	+12.89	-0.37	+0.79	+1.84	+2.70	+5.20	+5.12	+4.02
Our	4	+13.84	-0.84	+0.94	+2.24	+2.57	+5.37	+5.49	+4.23
Our	5	+13.12	-0.54	+1.19	+2.45	+2.78	+5.64	+5.54	+4.31
Our	6	+13.95	-0.48	+0.95	+2.01	+2.77	+5.69	+5.93	+4.40
Our	7	+13.32	-0.90	+1.11	+2.16	+3.14	+5.43	+5.74	+4.28
Our	8	+13.16	-1.16	+0.95	+1.88	+2.67	+5.75	+6.20	+4.21

to further improve baselines performance in SF-OUDA. We propose the following modifications:

- CBST* uses a confidence based sampling strategy to select class-balanced pseudo-labels. We improve CBST* by using our adaptive selection strategy based on uncertainty;
- TPLD* builds upon CBST* by increasing pseudo-label number through densification and voting. We improve TPLD* with our more robust adaptive pseudo-label selection and substitute the spatial nearest neighbor with our geometrically informed propagation strategy.
- ProDA* exploits a centroid-based weighting strategy to denoise pseudo-labels. Moreover, momentum update is performed between source $F_{\mathcal{S}}$ and target model $F_{\mathcal{T}}$. We improve ProDA* in its three main parts. First, we remove source model momentum update as it promotes domain drift. Second, we substitute pseudo-labelling with our iterative dropout based pseudo-labeling strategy. Third, we compute more robust centroids by considering the mean of point-features in our iterative pseudo-labelling strategy.

Tab. 3 shows that GIPSO components can be used to successfully improve the performance of existing methods. ProDA* improves from -32.63 to $+1.48$, we deem this is due to the more robust centroid computation and to the lower adaptation drift obtained with a non-updated source model. CBST* benefits from a better pseudo-label selection improving from $+0.28$ to $+1.07$. TPLD* benefits from a better pseudo-labels and the geometrically informed propagation improving from $*0.56$ to $+1.38$.

5 Class mapping

In Sec. 5.1 we detail the class mapping to make Synth4D compatible with SemanticKITTI [2] and nuScenes [3]. In Sec. 5.2 we report the class mapping used in SynLiDAR [1].

Table 3. Ablation study on Synth4D → SemanticKITTI reporting the improvement of state-of-the-art methods by using GIPSO adaptive selection strategy and propagation strategy.

Model	vehicle	pedestrian	road	sidewalk	terrain	manmade	vegetation	Avg
Source	22.54	14.38	42.03	28.39	15.58	38.18	54.14	30.75
Target	+3.76	+0.92	+9.41	+16.95	+19.79	+10.92	+10.71	+10.35
ProDA*	-58.92	-12.08	-36.74	-45.32	-15.46	-20.69	-39.24	-32.63
CBST*	-0.13	0.58	-1.00	-1.12	0.88	1.69	1.03	0.28
TPLD*	0.36	1.18	-0.76	-0.71	0.95	1.74	1.15	0.56
ProDA* (Ours)	2.04	4.40	0.24	0.62	0.29	1.07	1.71	1.48
CBST* (Ours)	2.72	-2.53	-0.19	0.56	1.48	3.02	2.46	1.07
TPLD* (Ours)	2.81	-2.33	-0.05	0.65	2.30	3.44	2.82	1.38

5.1 Synth4D

Tab. 4 reports the class mapping from Cityscapes [5] format of CARLA [6] to the classes of Synth4D. Tab. 5 reports the class mapping from SemanticKITTI to Synth4D. Tab. 6 reports the class mapping from nuScenes to Synth4D.

Tab. 4-6 maps input labels into the eight Synth4D labels: *vehicle, pedestrian, road, sidewalk, terrain, manmade, vegetation* and, *unlabelled*. This class mapping corresponds to the label intersections between CARLA, SemanticKITTI and nuScenes. All the classes that do not intersect with other datasets are considered as *unlabelled*.

Using the mapping in Tab. 4, the resulting class distributions for Synth4D are reported in Tab. 7. It is important to notice that class distributions differ among sensors as they have been acquired with independent runs. During each run, the simulator is set to randomly initialise the ego-vehicle re-spawn position, agents' positions (i.e., vehicles and pedestrians) and agents' trajectories. Therefore, the same class distribution cannot be ensured.

5.2 SynLiDAR

To make results compatible, we mapped SynLIDAR [1] classes to Synth4D classes. Tab. 8 reports the class mapping used in our experiments.

6 Qualitative results

We report additional adaptation results of GIPSO in Synth4D→SemanticKITTI (Fig. 1-2), SynthLiDAR→SemanticKITTI (Fig. 3-4) and, in Synth4D→nuScenes (Fig. 5-6). In all the cases, we include large and small improvement cases. Large improvement cases have a positive mIoU improvement over +20.0 mIoU, for Synth4D→SemanticKITTI and SynLiDAR→SemanticKITTI while over +10.0 mIoU for Synth4D→nuScenes. Small improvement cases have an improvement lower than +3.0 mIoU on all the adaptation scenarios. For a fair comparison, we also include the predictions of the source model not adapted (source) and the ground truth annotations (ground truth).

Table 4. Class mapping from CARLA [6] format to Synth4D.

CARLA-ID	CARLA-Name	Synth4D-Name	Synth4D-ID
0	unlabelled	unlabelled	0
1	building	manmade	6
2	fences	manmade	6
3	other	unlabelled	0
4	pedestrian	pedestrian	2
5	pole	manmade	6
6	roadlines	road	3
7	road	road	3
8	sidewalk	sidewalk	4
9	vegetation	vegetation	7
10	vehicle	vehicle	1
11	wall	manmade	6
12	trafficsign	manmade	6
13	sky	unlabelled	0
14	ground	unlabelled	0
15	bridge	manmade	6
16	railtrack	manmade	6
17	guardrail	manmade	6
18	trafficlight	unlabelled	0
19	static	unlabelled	0
20	dynamic	unlabelled	0
21	water	unlabelled	0
22	terrain	terrain	5

Table 5. Class mapping from SemanticKITTI [2] format to Synth4D.

SemanticKITTI-ID	SemanticKITTI-Name	Synth4D-Name	Synth4D-ID
0	unlabelled	unlabelled	0
1	car	vehicle	1
2	bicycle	unlabelled	0
3	motorcycle	unlabelled	0
4	truck	unlabelled	0
5	other-vehicle	unlabelled	0
6	person	pedestrian	2
7	bicyclist	unlabelled	0
8	motorcyclist	unlabelled	0
9	road	road	3
10	parking	road	3
11	sidewalk	sidewalk	4
12	other-ground	unlabelled	0
13	building	manmade	6
14	fence	manmade	6
15	vegetation	vegetation	7
16	trunk	vegetation	7
17	terrain	terrain	5
18	pole	manmade	6
19	traffic-sign	manmade	6

Table 6. Class mapping from nuScenes [3] format to Synth4D.

nuScenes-ID	nuScenes-Name	Synth4D-Name	Synth4D-ID
0	unlabelled	unlabelled	0
1	barrier	unlabelled	0
2	bicycle	unlabelled	0
3	bus	unlabelled	0
4	car	vehicle	1
5	construction-vehicle	unlabelled	0
6	motorcycle	unlabelled	0
7	pedestrian	pedestrian	2
8	traffic-cone	unlabelled	0
9	trailer	unlabelled	0
10	truck	unlabelled	0
11	driveable-surface	road	3
12	other-flat	unlabelled	0
13	sidewalk	sidewalk	4
14	terrain	terrain	5
15	manmade	manmade	6
16	vegetation	vegetation	7

Table 7. Number of annotated points for each adaptation category for the simulated Velodyne HDL32E and Velodyne HDL64E. Each sensor setup was acquired in a different run.

Velodyne	**# labels** (10^8)						
	vehicle	pedestrian	road	sidewalk	terrain	manmade	vegetation
HDL32E	2.52	0.04	4.35	1.07	0.95	1.48	1.24
HDL64E	1.15	0.03	6.09	1.25	1.51	1.11	0.75

Table 8. Class mapping from SynLiDAR [1] format to Synth4D.

SynliDAR-ID	SynLiDAR-Name	Synth4D-Name	Synth4D-ID
0	unlabelled	unlabelled	0
1	car	vehicle	1
2	pickup	vehicle	1
3	truck	unlabelled	0
4	bus	unlabelled	0
5	bicycle	unlabelled	0
6	motorcycle	unlabelled	0
7	other-vehicle	unlabelled	0
8	road	road	3
9	sidewalk	sidewalk	4
10	parking	road	3
11	other-ground	unlabelled	0
12	female	pedestrian	2
13	male	pedestrian	2
14	kid	pedestrian	2
15	crowd	pedestrian	2
16	bicyclist	unlabelled	0
17	motorcyclist	unlabelled	0
18	building	manmade	6
19	other-structure	unlabelled	0
20	vegetation	vegetation	7
21	trunk	vegetation	7
22	terrain	terrain	5
23	traffic-sign	manmade	6
24	pole	manmade	6
25	traffic-cone	unlabelled	0
26	fence	manmade	6
27	garbage-can	unlabelled	0
28	electric-box	unlabelled	0
29	table	unlabelled	0
30	chair	unlabelled	0
31	bench	unlabelled	0
32	other-object	unlabelled	0

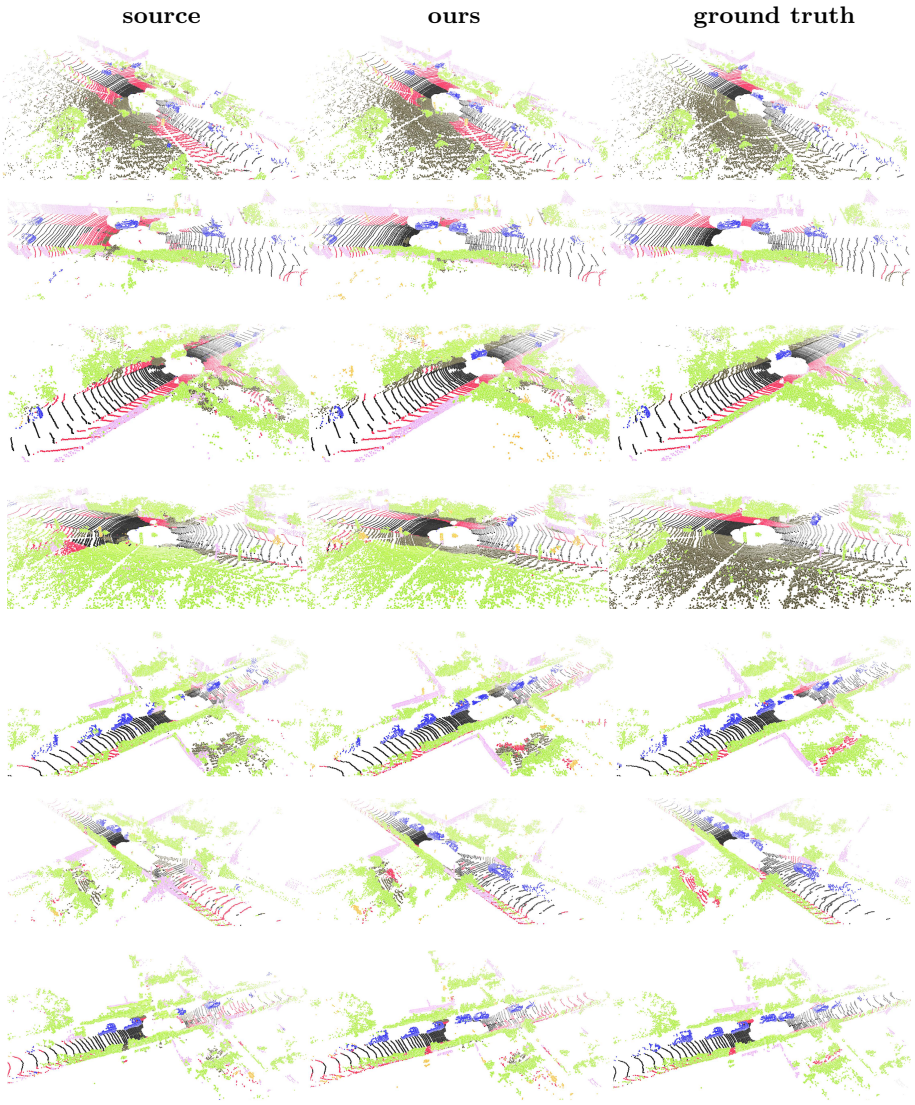

Fig. 1. Qualitative adaptation results on Synth4D→SemanticKITTI reporting large improvement cases. We compare GIPSO predictions during SF-OUDA (ours) with source model predictions (source) and with ground truth annotations (ground truth).

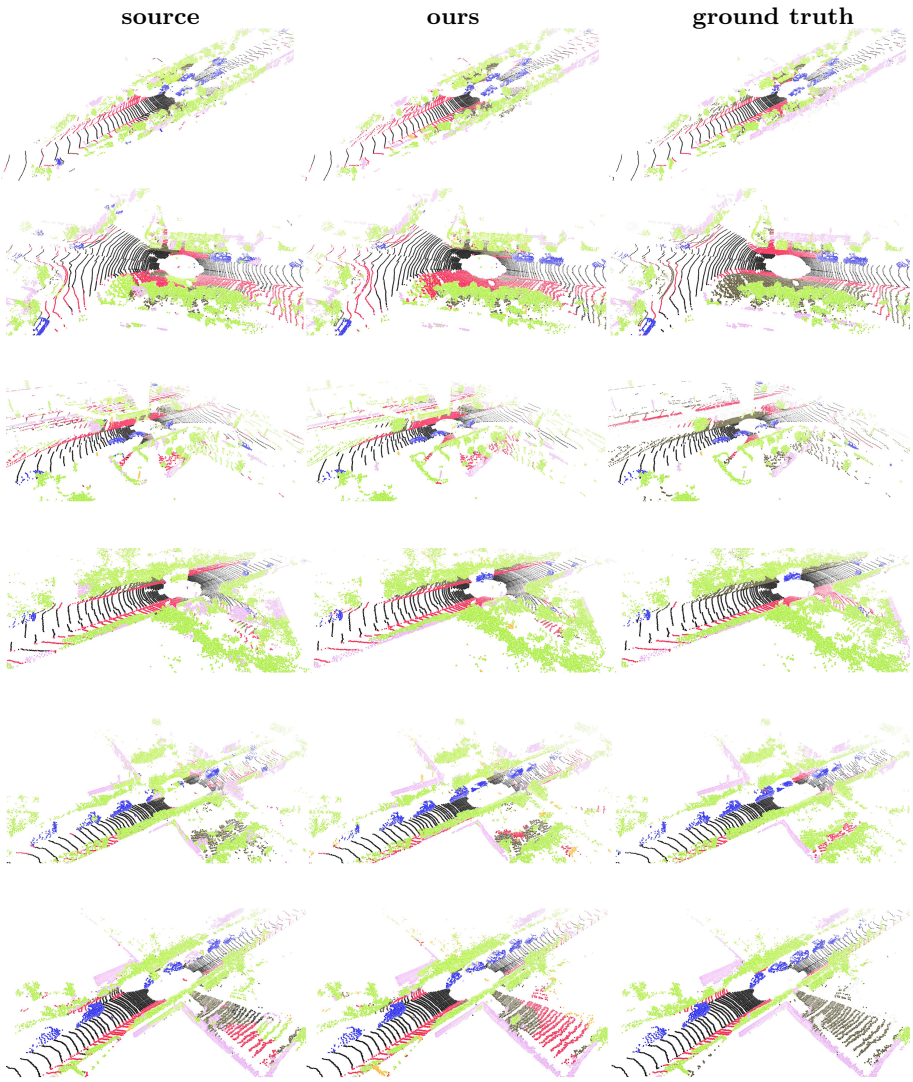

Fig. 2. Qualitative adaptation results on Synth4D→SemanticKITTI reporting small improvement cases. We compare GIPSO predictions during SF-OUDA (ours) with source model predictions (source) and with ground truth annotations (ground truth).

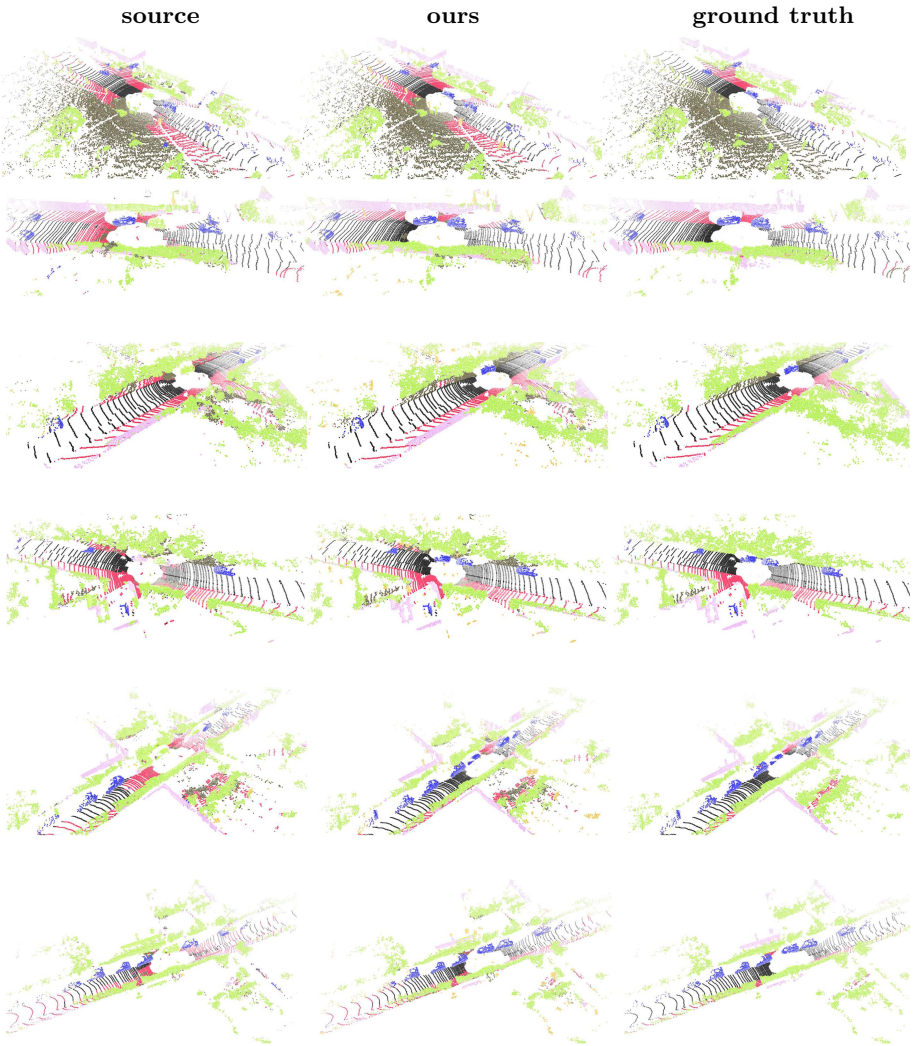

Fig. 3. Qualitative adaptation results on SynLiDAR→SemanticKITTI reporting large improvement cases. We compare GIPSO predictions during SF-OUDA (ours) with source model predictions (source) and with ground truth annotations (ground truth).

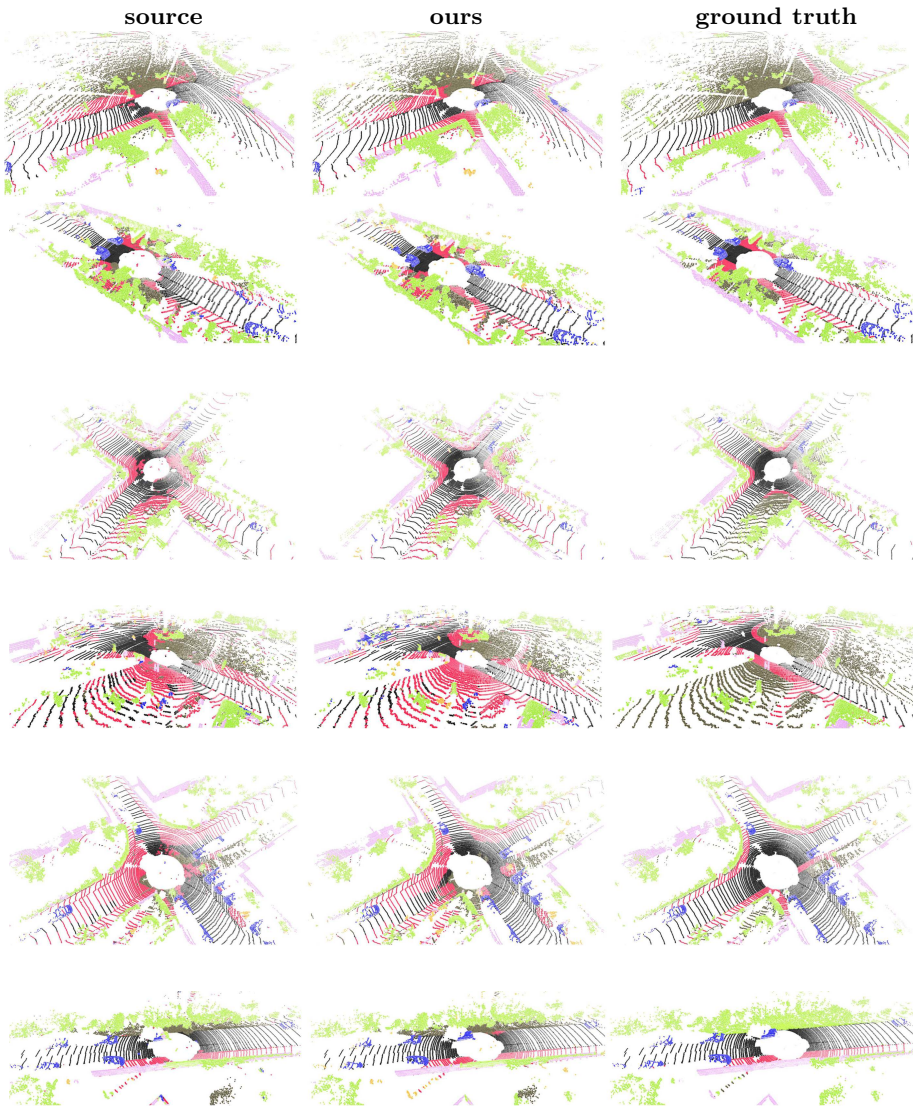

Fig. 4. Qualitative adaptation results on SynLiDAR→SemanticKITTI reporting small improvement cases. We compare GIPSO predictions during SF-OUDA (ours) with source model predictions (source) and with ground truth annotations (ground truth).

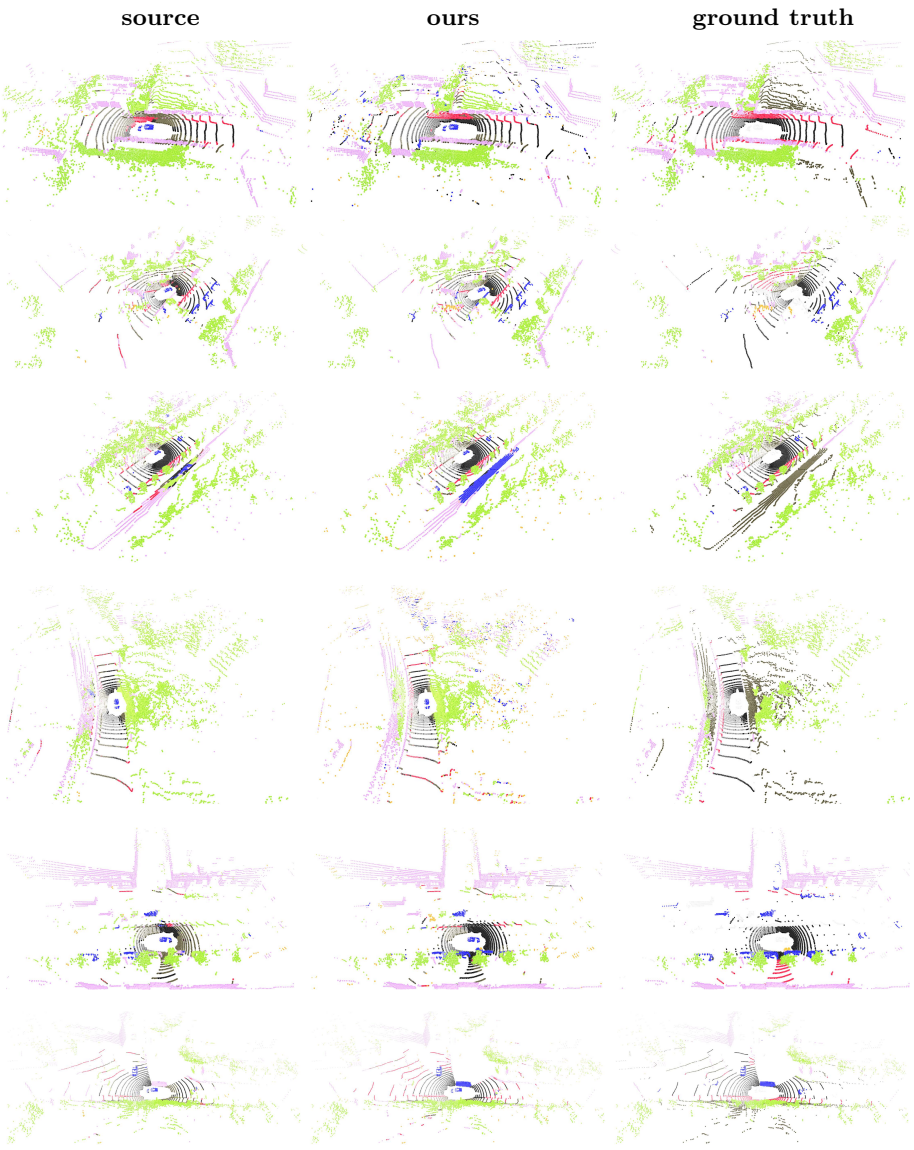

Fig. 5. Qualitative adaptation results on Synth4D→nuScenes reporting large improvement cases. We compare GIPSO predictions during SF-OUDA (ours) with source model predictions (source) and with ground truth annotations (ground truth).

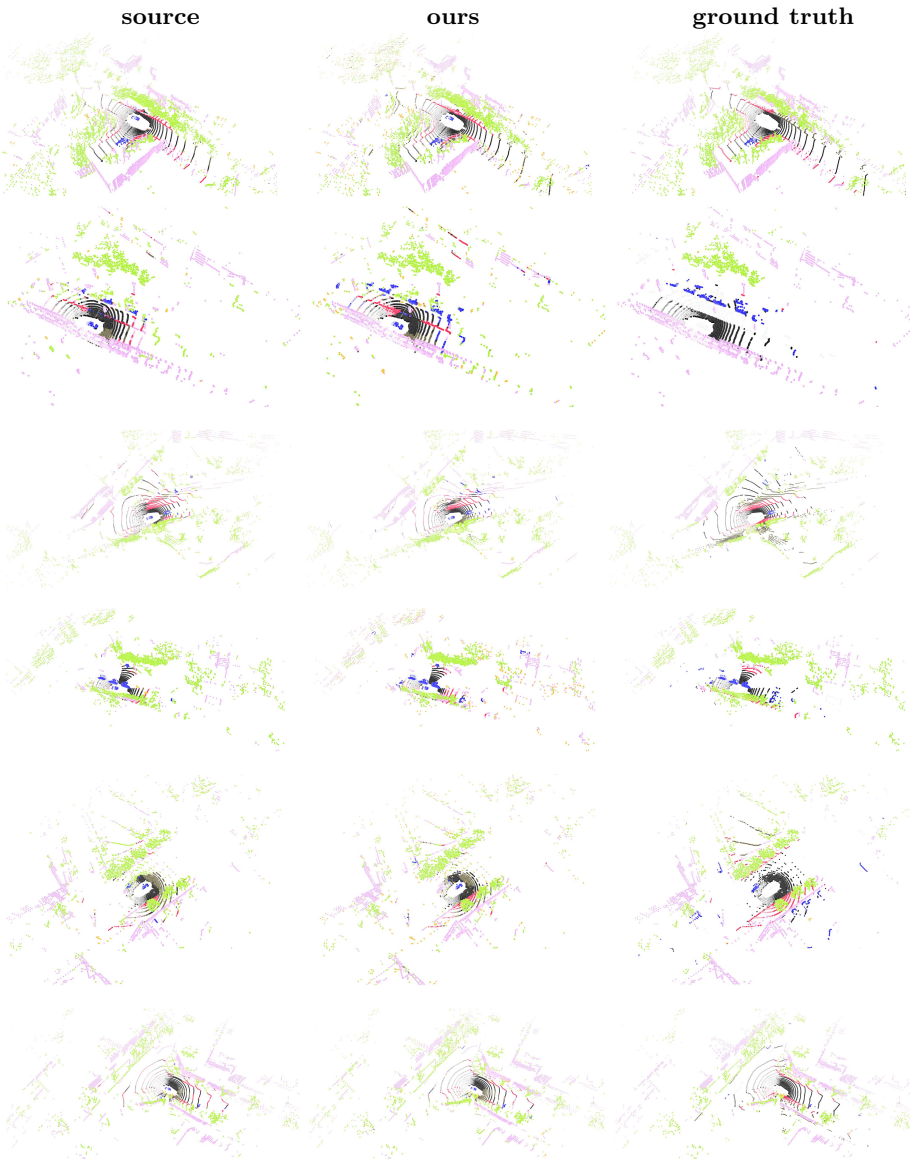

Fig. 6. Qualitative adaptation results on Synth4D→nuScenes reporting small improvement cases. We compare GIPSO predictions during SF-OUDA (ours) with source model predictions (source) and with ground truth annotations (ground truth).

References

1. Aoran, X., Jiaxing, H., Dayan, G., Fangneng, Z., Shijian, L.: Synlidar: Learning from synthetic lidar sequential point cloud for semantic segmentation. arXiv (2021) 3, 4, 7
2. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In: ICCV (2019) 3, 6
3. Caesar, H., Bankiti, V., Lang, A., Vora, S., Liong, V., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuScenes: A multimodal dataset for autonomous driving. In: CVPR (2020) 3, 6
4. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: CVPR (2019) 1
5. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016) 4
6. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: ACRL (2017) 4, 5
7. Zhang, P., Zhang, B., Zhang, T., Chen, D., Wang, Y., Wen, F.: Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In: CVPR (2021) 2
8. Zou, Y., Yu, Z., Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: ECCV (2018) 2