Supplementary Material for "MSeg3D: Multi-modal 3D Semantic Segmentation for Autonomous Driving"

Jiale Li[1] Hang Dai[2*] Hao Han[3] Yong Ding[3*]
[1]College of Information Science and Electronic Engineering, Zhejiang University
[2]School of Computing Science, University of Glasgow
[3]School of Micro-Nano Electronics, Zhejiang University
[*]Corresponding authors{Hang.Dai@glasgow.ac.uk, dingyong09@zju.edu.cn}.

1. Implementation Details

This section introduces our implementation details on input data processing and network. The complete implementation can be found in our code.

1.1. Input Data Processing

The processing details of the input point clouds and images for nuScenes [2], Waymo [13], and SemanticKITTI [1] datasets are summarized in Tab. S1. For nuScenes and Waymo datasets, the input size $(H_{\text{in}}, W_{\text{in}})$ of multi-camera images are aligned as (640, 960) by resizing for convenience of batch construction.

Table S1. Details on input data processing.

Dataset	nuScenes [2]	Waymo [13]	SemanticKITTI [1]
Ponit Cloud Range: X-axis	$[-51.2m, +51.2m]$	$[-75.2m, +75.2m]$	$[-75.2m, +75.2m]$
Ponit Cloud Range: Y-axis	$[-51.2m, +51.2m]$	$[-75.2m, +75.2m]$	$[-75.2m, +75.2m]$
Ponit Cloud Range: Z-axis	$[-5.0m, +3.0m]$	$[-2.0m, +4.0m]$	$[-4.0m, +2.0m]$
Voxelization Step d	$(0.1m, 0.1m, 0.2m)$	$(0.1m, 0.1m, 0.15m)$	$(0.1m, 0.1m, 0.15m)$
Image Input Size $(H_{\text{in}}, W_{\text{in}})$	(640, 960)	(640, 960)	(360, 1280)

1.2. Network

We implement all the neural network models in our paper based on the popular deep learning framework Pytorch 1.7.1 and the GPU-cluster server with Tesla-V100 devices and CUDA 11.0 platform.

LiDAR Point Cloud Backbone. To be general enough, we adopt LiDAR point cloud backbone as the sparse 3D U-Net [11] from OpenPCDet toolbox [3], which is a widely-used point cloud backbone in 3D object detection and segmentation. As shown in Fig. S1, the sparse 3D U-Net stacks four down-sampling blocks to make the voxel-wise features more informative with increasing receptive fields, four up-sampling blocks for resolution restoration and feature refinement, and skip-connections between the down-sampling and up-sampling blocks. For the eight convolutional blocks, we configure the output feature dimensions [C_1 - C_8] as [32, 64, 128, 256, 128, 64, 32, 32], respectively. We use the same point cloud backbone configuration for all the experiments.

The 3D convolution operations include the sparse 3D convolution (SparseConv3D), inverse sparse 3D convolution (InverseSparseConv3D) [15], and submanifold 3D convolution (SubMConv3D) [5], implemented by Spconv 1.2 [4].

Camera Image Backbone. In our paper, we employ HRNet-w48 [14] as our default image backbone in all the experiments, which is further validated in the scalability analysis from Tab. 9 in our paper.

SF-Phase. Both the Multi-Head Self-Attention ($MHSA$) and Multi-Head Cross-Attention ($MHCA$) are configured with N_{H} as 4, C_{gfused} as 64, and C_{sfused} as 96.

Training. All our models are trained under the same training schedule: AdamW [9] optimizer and one-cycle learning rate policy [12] with division factor 10, momentum ranges from 0.95 to 0.85, weight decay 0.01, maximum learning rate 0.01, and a batch of 32 random samples are distributed on 16 Tesla V100 GPUs with 24 epochs. During the training process, the data augmentation transformations in Tab. 1 of our paper are used to avoid overfitting.

Inference. During the inference stage, the *argmax* function is applied on the point-wise output $\hat{\mathbf{Y}}$ to obtain the category index with the highest probability value as the segmentation result for each point. Note that the voxel-wise segmentation $\mathbf{D}_{\text{lidar}}$ and pixel-wise segmentation $\mathbf{D}'_{\text{img}}$ are not used for the final result. For preparing the submission results in Tab. 2 and Tab. 3 to online leaderboards, we employ the Test-Time Augmentation (TTA) proposed in SDSeg3D [8] and model ensemble as the common practices as other submissions. We train 4 MSeg3D variants configured with 1/10/20/25 point cloud frames for nuScenes and 3 MSeg3D variants configured with 1/5/10 point cloud frames for Waymo, where each model is applied with TTA. Notably, the rest of the results in our paper except Tab. 2 and Tab. 3 are all evaluated without TTA or model ensemble.

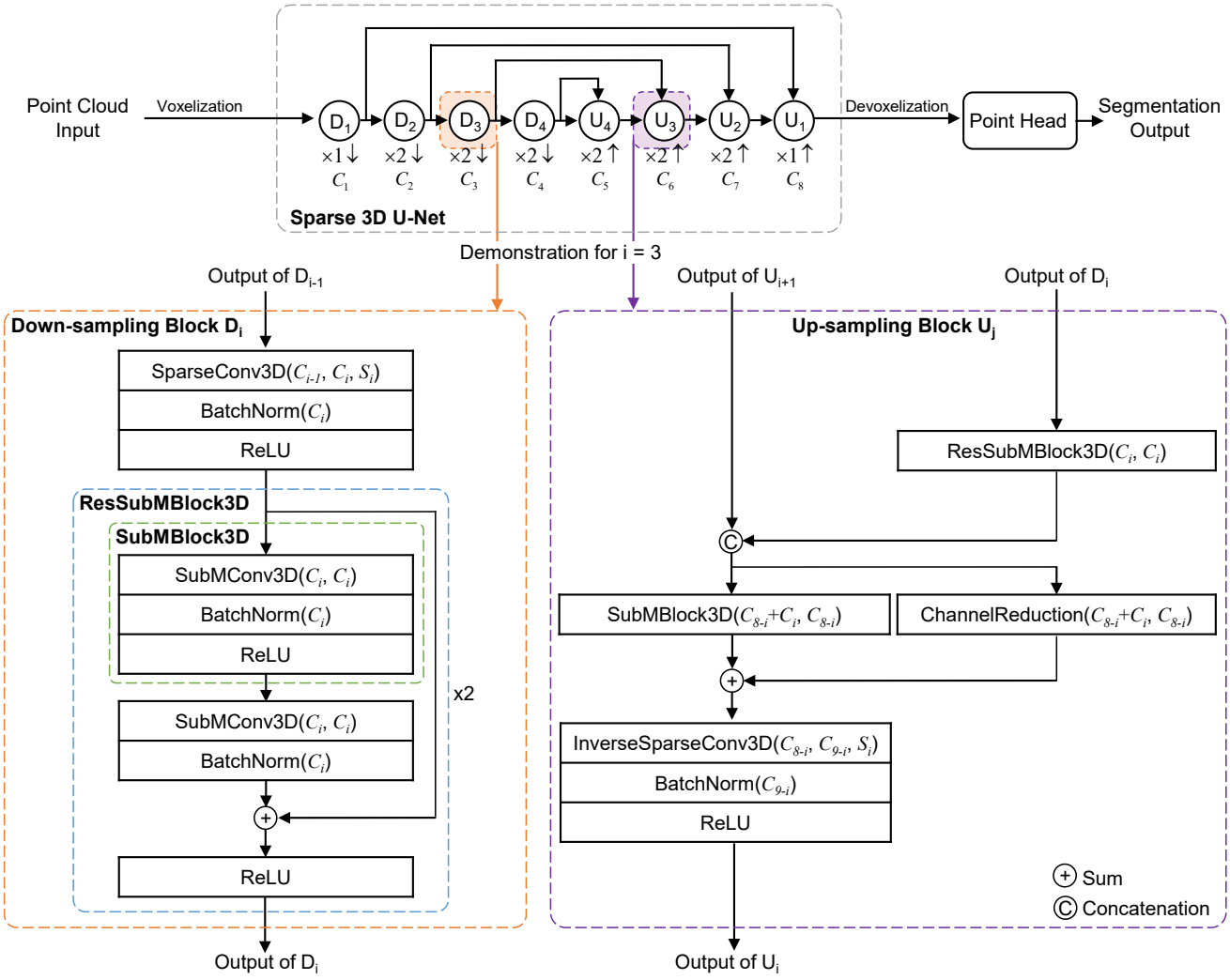

Figure S1. Point cloud backbone: sparse 3D U-Net.

2. Experimental Results

This section includes more qualitative and quantitative experimental results in addition to our paper.

2.1. Qualitative Results

Fig. S2 and Fig. S3 show that both the 3D and 2D semantic segmentation can be jointly achieved in our unified multi-modal MSeg3D. The LiDAR-side segmentation denotes the final output while the camera-side segmentation denotes the $\mathbf{D}'_{\text{img}}$ in Eq. 5 in our paper. Notably, although both the datasets do not provide image segmentation annotations, the camera-side segmentation results are reliable enough, which validates that the sparse point-to-pixel label $\mathbf{Y}_{\text{point2pixel}}$ generated in Eq. 16 provides sufficient supervision to correctly guide the $\mathbf{D}'_{\text{img}}$. Thus, the additional and expensive manual annotations on images are not necessary for training our multi-modal 3D semantic segmentation model.

For both datasets, we provide visualizations on a day sample and a night sample at the top and bottom, respectively. Especially for challenging night samples with insufficient illumination, our segmentation predictions are still reliable and robust due to the active laser-measurement of LiDAR. The Waymo results in Fig. S3 show better segmentation than the nuScenes results in Fig. S2, where the likely reason is that Waymo can provide relatively fine-grained image supervision in the point-to-pixel label $\mathbf{Y}_{\text{point2pixel}}$ from the denser laser points (i.e., 64 beams on Waymo and 32 beams on nuScenes).

2.2. Quantitative Results

Further Analysis on Performance Gap on All Points and Points Inside. The detailed results of Tab. 5 in our

paper are included in Tab. S2 and Tab. S3 for nuScenes and Waymo, respectively, which are also evaluated by the mIoU on all points and mIoU[1] on points inside. Despite the inapparent mIoU improvements, the improvements of mIoU[1] are much more obvious, especially on nuScenes. In Tab. S2, the multi-camera modality shows strong potential to improve the LiDAR-only model across all the categories inside the sensor FOV intersection, which motivates us to dive deeper into the multi-modal segmentation. Without loss of generality, we further conduct similar experiments on Waymo. From Tab. S3, the performance improvements of multi-modal models are relatively weaker but show a consistent trend. On Waymo, the improvements are weakened by the denser leaser points and the missing rear camera, which is also analyzed in Fig. 3 of our paper. Under such an implicit condition, we effectively make efforts to further investigate the inherent difficulties for achieving top-performing multi-modal 3D semantic segmentation by our complete MSeg3D framework.

Further Analysis on Camera Malfunction. From Tab. 7 in our paper, our MSeg3D with no working cameras (denoted as #Camera = 0) still outperforms the LiDAR-only baseline (denoted as #Camera = $\times$), which is benefited from the loss term $\mathbf{L}_{\text{pixel2point}}$ (Eq. 13) in training stage. As we have analyzed in our paper, applying such a loss term effectively transfers the useful image appearance priors to facilitate LiDAR feature learning. From the perspective of knowledge distillation [6], optimizing the $\mathbf{L}_{\text{pixel2point}}$ can also be treated as a feature-level distillation. Such experimental results also hint at an attractive and potential trend that can be deeply investigated: performing cross-modal knowledge distillation can improve point cloud feature learning as well as maintain computational efficiency if the image branch is designed to be detached during inference.

Further Analysis on Multi-frame Point Clouds Input. The category-wise results in Tab. 8 in our paper are detailed in Tab. S4 and Tab. S5 for nuScenes and Waymo, respectively. As can be observed in Tab. S4 and Tab. S5, the optional multi-frame point clouds benefit the segmentation, especially on some static objects (such as pole, cone, sign, sidewalk, road and so on) or slow-moving objects (like pedestrian). Motion blur is more likely to occur on high-speed moving objects, because the relative motion between the object and the ego-vehicle cannot be accurately estimated by the ego-vehicle information alone. The improvements are gradually saturated with the point clouds of more than 10 frames on Waymo and 25 frames on nuScenes, since the motion blur effect also becomes more severe. More neighbor frames of nuScenes point cloud can be used due to the higher sampling frequency of LiDAR sensors (i.e., 20 Hz on nuScenes vs 10 Hz on Waymo). As shown in the last rows, our multi-modal MSeg3D framework is also capable of improving the best-performing multi-frame LiDAR-only model, which is also analyzed in Tab. 8 of our paper.

[1]In the following text, mIoU[1] denotes the segmentation performance evaluated on only the points inside the FOV intersection by excluding the points outside like PMF [16] and other methods [7, 10].

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jürgen Gall. SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences. In *ICCV*, pages 9296–9306, 2019. 1

[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. In *CVPR*, pages 11618–11628, 2020. 1

[3] OpenPCDet Contributors. OpenPCDet: An open-source toolbox for 3D object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2023. 1

[4] Spconv Contributors. SpConv: Spatially sparse convolution library. https://github.com/traveller59/spconv, 2023. 1

[5] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, pages 9224–9232, 2018. 1

[6] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015. 3

[7] Georg Krispel, Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. FuseSeg: LiDAR point cloud segmentation fusing multi-modal data. In *WACV*, pages 1863–1872, 2020. 3

[8] Jiale Li, Hang Dai, and Yong Ding. Self-distillation for robust LiDAR semantic segmentation in autonomous driving. In *ECCV*, volume 13688, pages 659–676, 2022. 1

[9] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 1

[10] Khaled El Madawi, Hazem Rashed, Ahmad El Sallab, Omar Nasr, Hanan Kamel, and Senthil Kumar Yogamani. RGB and LiDAR fusion based 3D semantic segmentation for autonomous driving. In *ITSC*, pages 7–12, 2019. 3

[11] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network. *IEEE TPAMI*, 43(8):2647–2664, 2021. 1

[12] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, page 1100612, 2019. 1

[13] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang,

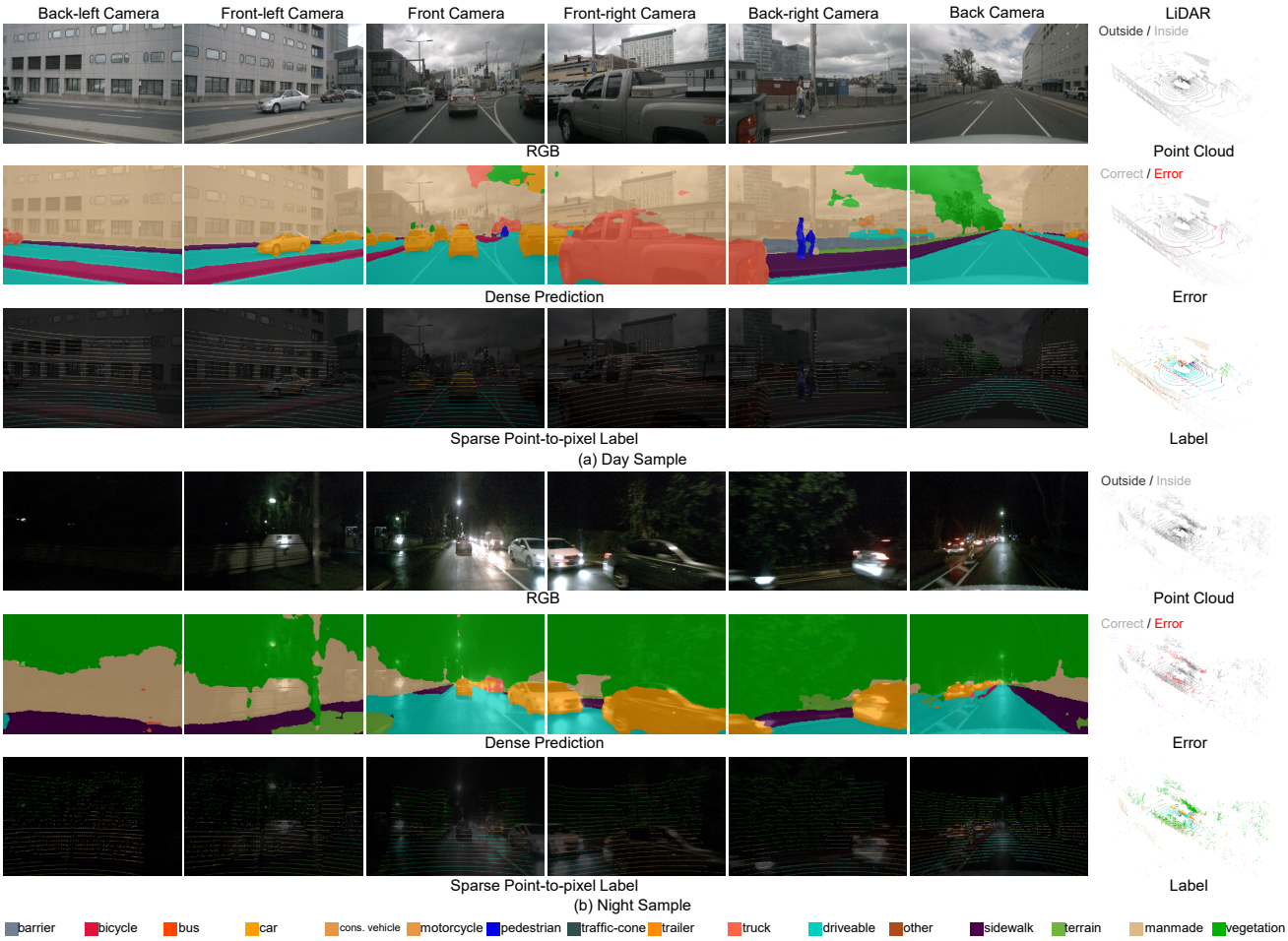

Figure S2. Qualitative visualization on nuScenes validation set, exported from the complete MSeg3D model in the last row of Tab. 6. Best viewed with zoom and color. The dense prediction of the RGB image is $\mathbf{D}'_{\text{img}}$, supervised by the sparse point-to-pixel label $\mathbf{Y}_{\text{point2pixel}}$ in Eq. 16. Note that the points outside are projected below the image bottom boundary. The top half of the image has no labels since no points are projected into this area.

Table S2. Experiments on the performance gap between the mIoU on all points and mIoU^1 on points inside, and the effects of data augmentation (DA), which are the details of Tab. 5 on nuScenes validation set. For simplicity, only the GF-Phase is used for multi-modal fusion (M-Fusion) in this table. Data augmentation (DA) includes the LiDAR DA (L-DA) and Multi-modal DA (M-DA).

LiDAR	M-Fusion	DA	**mIoU**^{1}	**Improv.**^{1}	barrier^1	bicycle^1	bus^1	car^1	construction^1	motorcycle^1	pedestrian^1	traffic-cone^1	trailer^1	truck^1	driveable^1	other^1	sidewalk^1	terrain^1	manmade^1	vegetation^1	**mIoU**	**Improv.**
√	×	L-DA	70.76	-	69.97	38.73	85.65	89.95	48.33	75.55	76.60	38.16	59.62	82.89	91.71	67.34	66.74	70.83	86.46	83.62	72.00	-
√	GF-Phase	×	76.68	+5.92	74.70	50.17	87.97	92.84	51.36	77.54	82.54	70.13	63.11	83.09	94.83	72.63	71.92	74.93	90.26	88.92	68.10	-3.90
√	GF-Phase	L-DA	77.48	+6.72	75.48	54.74	90.95	92.92	50.78	82.59	82.69	69.76	62.47	82.52	94.97	73.35	72.52	75.04	90.03	88.84	71.35	-0.65
√	GF-Phase	M-DA	78.65	+7.89	75.89	54.96	95.29	93.21	56.12	83.48	83.04	67.87	64.53	85.36	95.29	73.66	74.02	75.81	90.66	89.25	72.39	+0.39

Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, pages 2443–2451, 2020. 1

[14] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *IEEE TPAMI*, 43(10):3349–3364, 2021. 1

[15] Yan Yan, Yuxing Mao, and Bo Li. SECOND: sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1

[16] Zhuangwei Zhuang, Rong Li, Kui Jia, Qicheng Wang, Yuanqing Li, and Mingkui Tan. Perception-aware multi-sensor fusion for 3D LiDAR semantic segmentation. In

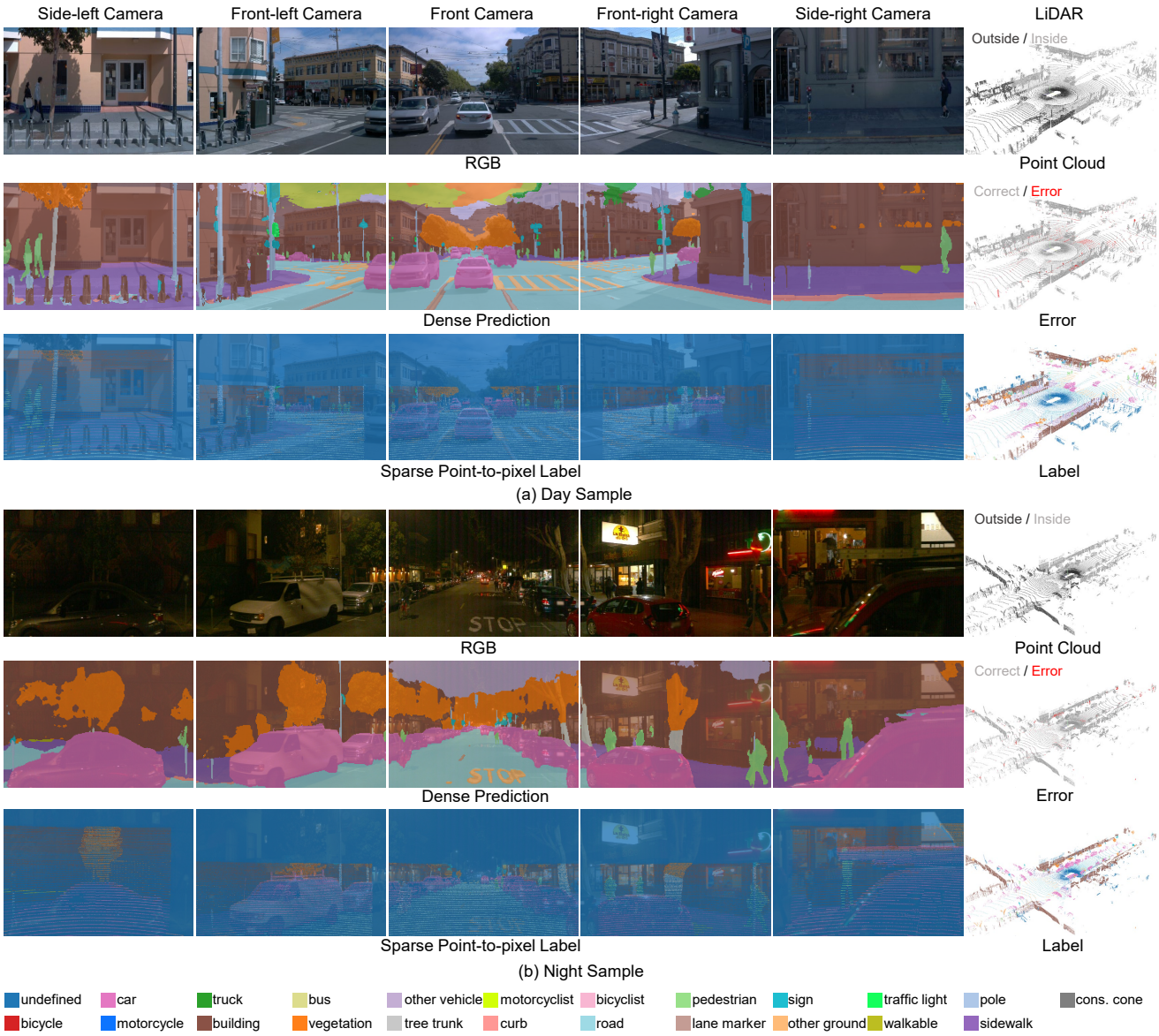

Figure S3. Qualitative visualization on Waymo validation set, exported from the complete MSeg3D model in the last row of Tab. 6. Note that the points within the rearview have no corresponding image, while our model reasonably segments all points in the point cloud with a few errors. The dense prediction of the RGB image is $\mathbf{D}'_{\text{img}}$, supervised by the sparse point-to-pixel label $\mathbf{Y}_{\text{point2pixel}}$ in Eq. 16. The top half of the image has no labels since no points are projected into this area.

Table S3. Experiments on the performance gap between the mIoU on all points and mIoU† on points inside, and the effects of data augmentation (DA), which are the details of Tab. 5 on Waymo validation set. Similar experimental setup to Tab. S2.

LiDAR	M-Fusion	DA	**mIoU**†	**Improv.**†	car†	truck†	bus†	other vehicle†	motorcyclist†	bicyclist†	pedestrian†	sign†	traffic light†	pole†	cons. cone†	bicycle†	motorcycle†	building†	vegetation†	tree trunk†	curb†	road†	lane marker†	other ground†	walkable†	sidewalk†	**mIoU**	**Improv.**
√	×	L-DA	67.41	-	94.72	62.91	82.55	30.35	0.00	71.82	88.80	69.40	23.90	72.65	69.34	70.28	77.47	94.20	90.11	66.40	68.97	92.33	54.28	47.48	79.28	75.67	67.48	-
√	GF-Phase	×	64.79	-2.62	94.58	53.16	63.97	27.03	0.11	67.78	89.29	66.48	29.03	74.26	63.92	66.27	79.96	94.57	90.08	63.99	68.75	91.59	49.28	40.71	78.02	72.64	59.77	-7.71
√	GF-Phase	L-DA	65.70	-1.71	94.67	54.04	66.74	27.97	0.14	67.53	89.45	68.37	34.66	76.43	66.72	63.57	68.66	95.09	90.60	66.00	69.96	92.01	52.18	46.99	78.82	74.85	60.34	-7.14
√	GF-Phase	M-DA	67.94	+0.53	94.59	55.23	76.43	28.06	0.64	71.82	89.99	71.07	33.51	78.11	69.82	70.57	81.37	95.23	90.67	67.01	70.51	92.34	54.06	48.53	79.52	75.62	63.97	-3.51

Table S4. Category-wise results of multi-frame point clouds input on nuScenes validation set, which are the details of Tab. 8 in our paper.

L-Frame	Camera	**mIoU**	barrier	bicycle	bus	car	cons. vehicle	motorcycle	pedestrian	traffic-cone	trailer	truck	driveable	other	sidewalk	terrain	manmade	vegetation
1	×	72.00	73.95	38.48	87.26	88.37	47.69	80.33	78.07	42.37	59.68	83.15	93.89	69.38	68.89	71.60	86.03	82.94
10	×	74.66	76.75	49.98	91.66	86.78	50.54	85.60	81.76	60.66	55.20	84.25	94.20	67.55	65.81	72.27	87.09	84.47
20	×	75.37	77.11	51.47	93.58	85.79	43.03	82.60	83.41	61.93	62.06	81.16	95.63	70.48	72.96	73.01	87.45	84.18
25	×	**75.77**	76.64	50.80	91.84	85.32	52.69	85.06	82.57	64.02	58.91	82.29	95.34	68.55	72.89	72.88	87.37	85.10
30	×	75.28	72.54	47.46	92.55	85.58	55.41	84.03	83.29	60.83	62.19	82.68	94.40	67.98	70.85	71.68	87.76	85.25
40	×	75.15	77.00	42.21	93.65	85.10	53.98	83.00	81.37	64.72	58.89	82.61	95.15	69.78	72.43	71.94	86.30	84.25
25	√	**81.12**	79.61	59.66	97.15	92.09	57.86	89.5	86.24	71.29	73.64	87.53	96.53	75.08	75.26	76.03	91.01	89.32

Table S5. Category-wise results of multi-frame point clouds input on Waymo validation set, which are the details of Tab. 8 in our paper.

L-Frame	Camera	**mIoU**	car	truck	bus	other vehicle	motorcyclist	bicyclist	pedestrian	sign	traffic light	pole	cons. cone	bicycle	motorcycle	building	vegetation	tree trunk	curb	road	lane marker	other ground	walkable	sidewalk
1	×	67.48	93.84	58.07	82.21	34.80	0.00	73.97	89.37	67.19	24.56	72.91	68.09	70.91	78.86	94.18	90.08	66.05	68.89	91.88	53.85	49.33	79.63	75.88
5	×	69.18	94.78	62.65	85.63	30.35	0.00	73.79	90.37	70.13	30.38	75.47	71.88	73.25	85.13	95.36	91.11	67.99	70.86	93.05	55.27	48.63	79.90	76.04
10	×	**69.45**	94.74	62.35	84.04	34.44	0.00	73.99	90.44	70.52	29.52	75.86	71.01	76.49	84.34	95.26	90.87	68.17	70.82	93.17	55.86	49.60	79.81	76.48
15	×	68.88	94.71	59.99	82.98	32.34	0.00	74.04	90.65	69.26	29.84	75.86	71.73	76.26	81.96	94.73	90.84	68.17	70.73	92.68	55.44	48.71	78.79	75.61
20	×	68.78	94.82	60.18	82.44	29.50	0.00	73.32	90.57	70.75	27.62	76.99	70.29	76.27	80.20	95.57	91.15	68.53	71.04	93.22	55.20	49.59	79.72	76.14
30	×	68.64	94.70	58.40	81.43	27.02	0.00	72.25	90.57	70.78	31.34	76.36	69.74	76.52	81.51	95.13	90.84	68.85	71.00	93.04	55.17	49.27	79.79	76.42
10	√	**70.20**	95.45	64.30	87.50	29.95	0.02	73.17	90.75	73.90	37.65	77.85	71.14	75.77	82.40	95.68	91.31	68.67	71.52	93.36	56.07	50.92	79.94	77.20