Spherical Transformer for LiDAR-based 3D Recognition
Supplementary Material

Introduction

This is the supplementary material, which is divided into the following sections.

1. Analysis of model efficiency and parameters is given in Sec. 1.
2. Diagram of network structure is shown in Sec. 2.
3. More experiments are shown in Sec. 3.
4. Introduction of nuScenes [2], SemanticKITTI [1] and Waymo Open Dataset [4] are given in Sec. 4.
5. More visual comparisons are shown in Sec. 5.
6. Limitation analysis and future work are shown in Sec. 6.

ID	stage 1	stage 2	stage 3	stage 4	stage 5	mIoU (%)	inference time	parameters
I						75.2	30.4 ms	30.2 M
II	✓					76.4	34.8 ms	30.2 M
III		✓				76.4	35.8 ms	30.2 M
IV			✓			76.5	35.2 ms	30.4 M
V				✓		77.1	34.1 ms	31.1 M
VI					✓	77.2	34.3 ms	31.1 M
VII	✓	✓				76.8	39.8 ms	30.3 M
VIII	✓	✓	✓			77.3	44.4 ms	30.5 M
IX	✓	✓	✓	✓		77.8	48.5 ms	31.4 M
X	✓	✓	✓	✓	✓	78.4	53.6 ms	32.3 M

Table 1. Ablation on inserting our module into different stages on nuScenes *val* set, in terms of performance, inference time, and parameters

1. Analysis of Efficiency and Parameters

As shown in Table 1, we demonstrate the performance and inference time, as well as parameters when our proposed module is inserted into different stages. From the Experiments I and VI, we notice that the inference time and parameters only increase 3.9 ms and 0.9 M, respectively, when only inserting SphereFormer into the stage 5. Also, when inserted into any single stage, SphereFormer improves the result by a large margin, at a low cost of both speed and parameters. Note that the inference time is yielded by forwarding a single random scene of nuScenes *val* set into a single RTX 3090 GPU.

Also, we make a comparison with Cylinder3D [5] on SemanticKITTI, as illustrated in Table 2. It is worth noting that our method outperforms Cylinder3D in both efficiency and model size, which demonstrates the superiority of our method.

Method	inference time	parameters
Cylinder3D	170 ms	55.9 M
Ours	123 ms	32.3 M

Table 2. Comparison with Cylinder3D on SemanticKITTI *val* set

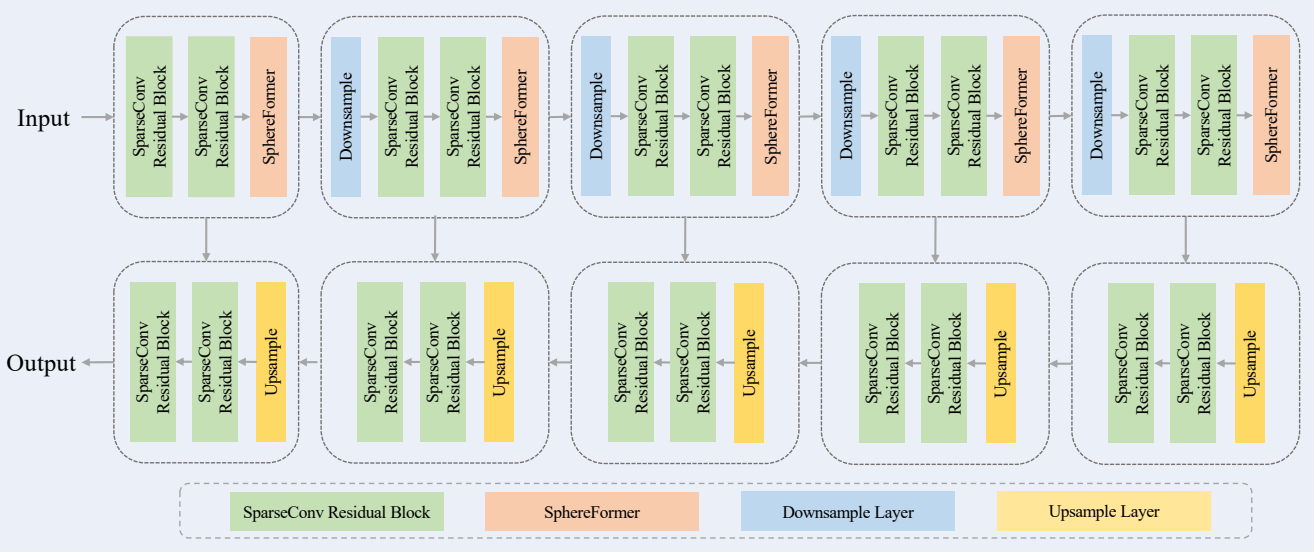


Figure 1. The framework structure. Our proposed module (*i.e.*, SphereFormer) is inserted into the end of each encoding stage.

2. Network Stucture

As shown in Fig. 1, we illustrate the structure of our framework. There are a total of 5 stages, and our proposed module (*i.e.*, SphereFormer) is inserted into the end of each encoding stage. The whole structure is based on U-Net [3]. For the downsample layer, we adopt a sparse convolution with both kernel size and stride set to 2. As for the upsample layer, we use an inverse sparse convolution.

3. More Experiments

3.1. Effect of inserting into different stages

As shown in Table 1, from Exp.II to VI, we apply SphereFormer into a single stage from stage 1 to stage 5, respectively. We can consistently obtain performance improvement. It is worth noting that in Exp.VI, by inserting SphereFormer into stage 5 only, we yield a 2.0% mIoU performance gain. Meanwhile, it incurs merely negligible inference time and model parameters. From Exp.VII to X, we add SphereFormer to more stages one by one. We also yield consistent improvement by including one more stage.

This ablation also indicates that we can achieve the tradeoff between performance and efficiency by inserting SphereFormer into different stages. When efficiency is in higher priority, we can choose to add SphereFormer to the latter one or two stages. And when performance is more needed, we can include more stages to yield more performance gain.

3.2. Experimental Result on SemanticKITTI Validation Set

The semantic segmentation result on SemanticKITTI *val* set is shown in Table 3. Our method outperforms the baseline model by a large margin. Again, the result demonstrates the effectiveness of our proposed module.

4. Datasets Introduction

nuScenes. The nuScenes dataset [2] comprises 1000 driving scenes. 850 scenes of them form the training and validation set, and the other 150 scenes are for testing. It contains 16 semantic classes for the LiDAR semantic segmentation task. The scene is scanned by 32-line LiDAR, so the point cloud is relatively sparser compared to the following two datasets.

Method	mIoU	car	bicycle	motorcycle	truck	other-vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other-ground	building	fence	vegetation	trunk	terrain	pole	traffic sign
Baseline	66.6	96.3	44.6	76.3	89.6	58.6	77.3	91.3	0.0	94.3	51.7	81.8	1.2	91.0	62.5	88.3	70.2	75.3	64.6	51.4
Ours	67.8	96.8	50.7	75.1	93.1	65.7	76.9	92.0	0.0	94.7	53.2	82.2	3.2	90.7	58.4	88.7	71.2	75.9	64.6	54.5
Ours‡	**69.0**	97.0	53.4	77.2	95.1	67.0	78.2	93.7	0.0	95.2	55.5	83.1	2.8	91.0	60.4	89.2	72.5	76.9	66.3	55.9

Table 3. Semantic segmentation results on SemanticKITTI *val* set. ‡ denotes using testing-time augmentations.

SemanticKITTI. The SemanticKITTI dataset [1] is composed of 22 point cloud sequences. The sequences from 00 to 10 are used for training. And the sequence 08 is used for validation. The remaining sequences 11 to 21 are adopted for testing. It has 19 classes which are frequently seen in driving scenes.

Waymo Open Dataset. The Waymo Open Dataset [4] contains 798, 202, and 150 point cloud sequences for training, validation and testing, respectively. Only $29,667$ frames of the training and validation sets are annotated with semantic segmentation labels. It has 22 semantic classes that are common in the driving scenes.

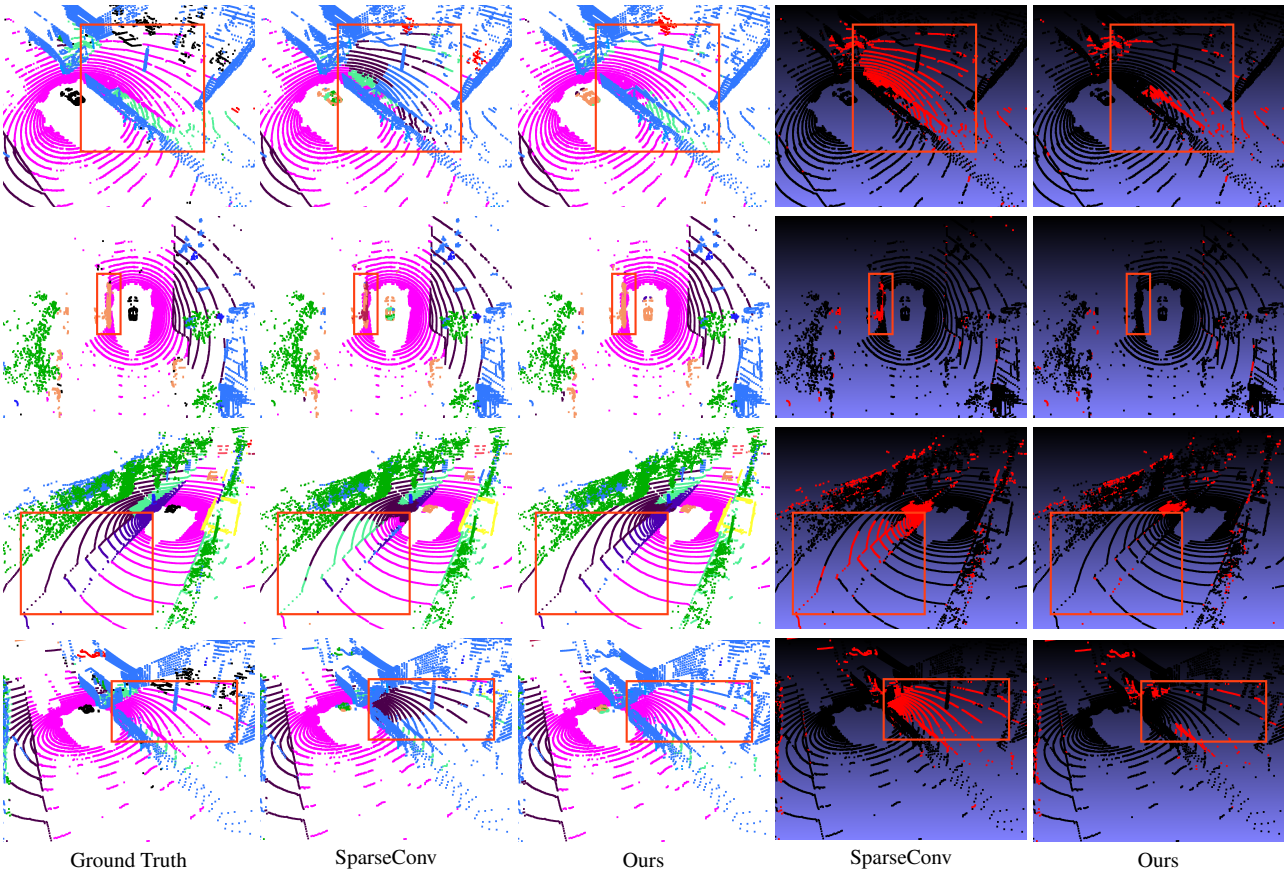
5. More Visual Comparisons

As shown in Fig. 2, we illustrate more examples for visual comparison. As highlighted with the red boxes, our method is superior to the baseline model (*i.e.*, SparseConv). Ours avoids large-scale artifacts, and can accurately recognize the *road*, *car*, *sidewalk*, and so on. Also, with our proposed module, the model can recognize distant objects more easily.

6. Limitation and Future Work

Limitation. Our method has been verified to work well on large-scale LiDAR point cloud data, but the performance on small-scale datasets is still unconfirmed. Also, although when inserting our proposed module into only stage 5, the performance increases by a large margin (*i.e.*, 2.0% mIoU) at a low cost of efficiency as shown in Table 1. But, when inserting our proposed module into all stages, the inference time increases a lot. We explain that the efficiency of our method highly relies on the CUDA kernel optimization, and currently, we only adopt several common optimization tricks such as memory coalesce and broadcast, but still have not explored more advanced methods. We believe the inference delay can be greatly reduced with proper CUDA optimization techniques.

Future Work. First, we will extend our method to more datasets and more LiDAR point cloud tasks such as panoptic segmentation. Second, we will adopt more advanced CUDA optimization techniques to improve efficiency.


Ground Truth
SparseConv
Ours
SparseConv
Ours
barrier
bicycle
truck
car
driveable surface
manmade
terrain
sidewalk
vegetation

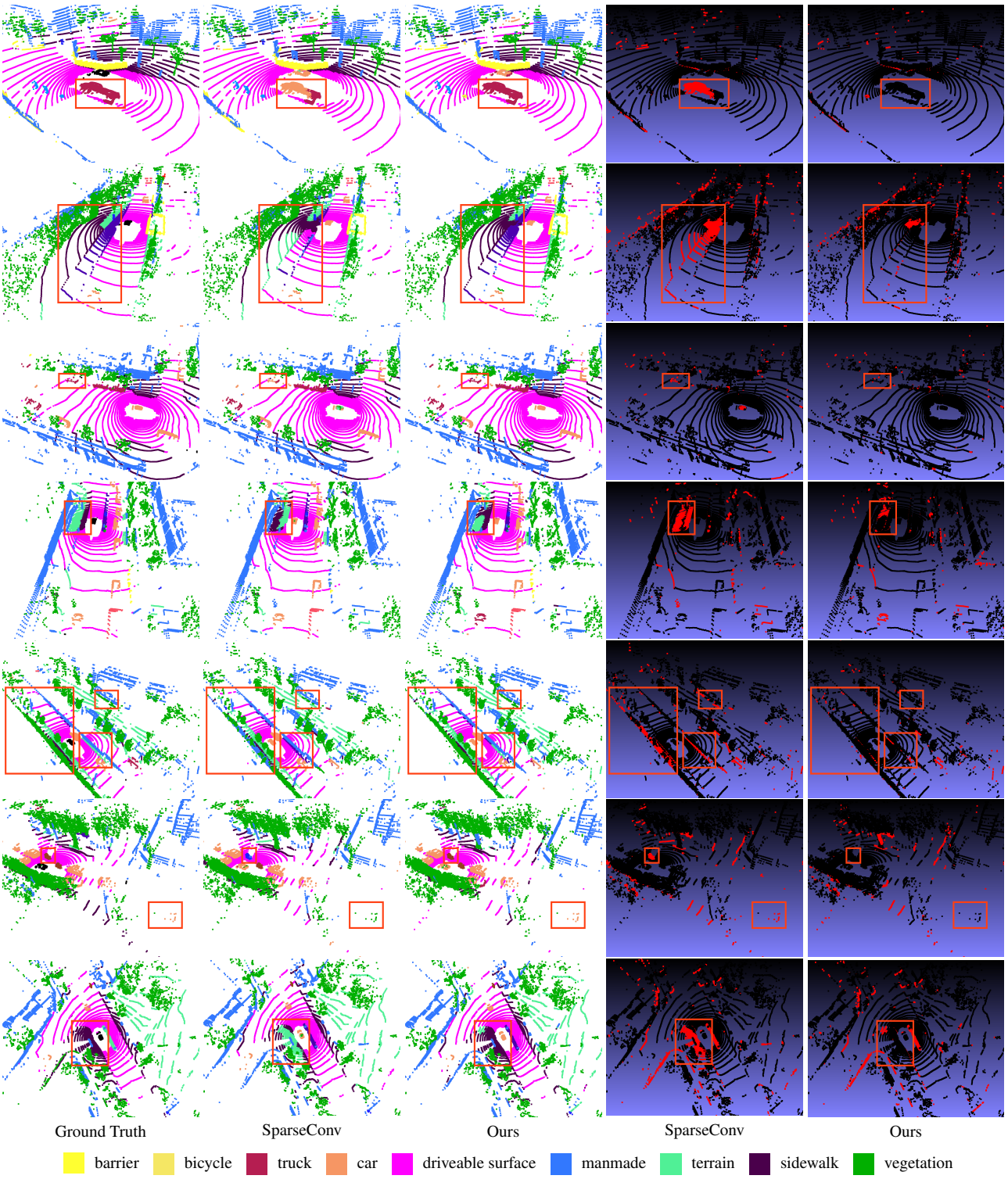


Figure 2. More visual comparison between vanilla SparseConv and ours (best viewed in color and by zoom-in). The last two columns are the difference maps with the ground truth.

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *ICCV*, 2019. 1, 3

[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*, 2020. 1, 2

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2

[4] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 1, 3

[5] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *CVPR*, 2021. 1