Supplementary Materials for "FusionNet"

1. Details of the Architecture and Layers

1.1. Input

Point Feature: For the Semantic KITTI dataset, we use the 4-channel feature for input which consists of (p_x, p_y, p_z, r) where p_x, p_y and p_z are the locations of the point p which are divided/normalized by the maximum absolute values $(70, 70, 20)$ respectively. r is the reflection intensity of the point p which is also normalized into the range of $[0, 1]$.

For the 3DSIS and ScanNet datasets, we directly use the RGB as the 3-channel input features. The RGB values are normalized by the mean and standard deviation of the dataset by subtracting the mean and dividing the deviation.

Voxel Feature: For the voxel feature, we directly use the mean of the point features as input which is calculated by averaging the points in each voxel.

1.2. Fusion Module

As presented in Table 2, each fusionnet layer is implemented by i) neighborhood voxel feature aggregation of "step (1)"; ii) neighborhood point feature aggregation of "step 1"; and iii) inner-voxel fine-grain aggregation of "step (2)$\sim$(4), 2$\sim$5".

1.3. Architecture Parameters

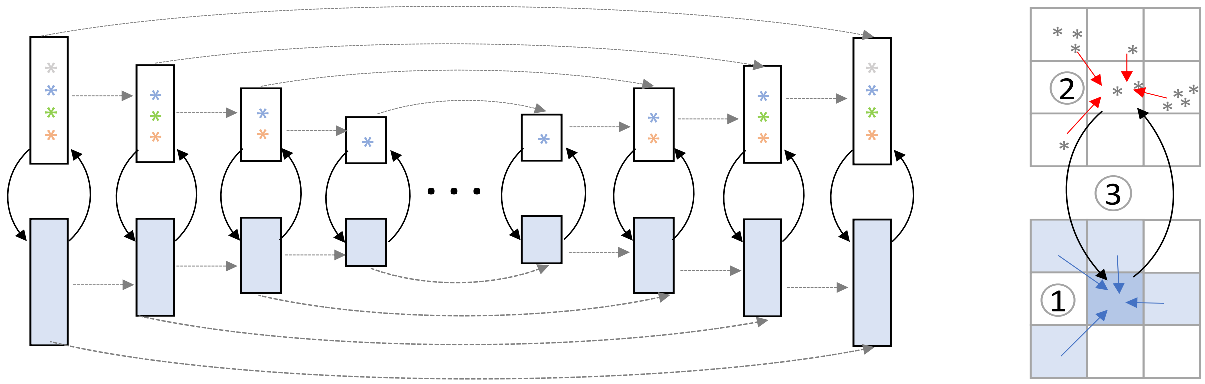
For the FusionNet architecture (illustrated in Fig. 1 and presented in Table 3), we use a total of 11 fusion models and 10 down- or up-sampling layers. Finally, the point-wise feature is refined by a point-wise fully-connected layer (linear layer) for point-wise classification.

2. More Results

More visualized results are presented in Fig. 2. Our FusionNet has many advantages for the large-scale LiDAR point cloud segmentation. Compared to state-of-the-art voxel-based networks [1], FusionNet can predict point-wise labels and avoid those ambiguous/wrong predictions at object boundaries when a voxel has points from different classes. It can give more accurate predictions for many small objects (*e.g.* cyclist, pedestrian and bicycles). When compared to state-of-the-art point-wise convolutions (*e.g.* [5]), our FusionNet gets much better segmentation accuracy in the large-scale LiDAR dataset. This is because our FusionNet is realized with more effective feature aggregation operations (including the effective voxel-level neighborhood aggregations and the fine-grain inner-voxel point-level aggregations).

Table 1: ScanNet 3D Segmentation Benchmark Results

Method	**mIoU**	bath	bed	bksf	cab	chair	cntr	curt	desk	door	floor	othr	pic	ref	show	sink	sofa	tab	toil	wall	wind
ScanNet [2]	30.6	20.3	36.6	50.1	31.1	52.4	21.1	0.2	34.2	18.9	78.6	14.5	10.2	24.5	15.2	31.8	34.8	30.0	46.0	43.7	18.2
PointNet++ [3]	33.9	58.4	47.8	45.8	25.6	36.0	25.0	24.7	27.8	26.1	67.7	18.3	11.7	21.2	14.5	36.4	34.6	23.2	54.8	52.3	25.2
TangetConv [4]	43.8	43.7	64.6	47.4	36.9	64.5	35.3	25.8	28.2	27.9	91.8	29.8	14.7	28.3	29.4	48.7	56.2	42.7	61.9	63.3	35.2
PointConv [5]	66.6	78.1	75.9	69.9	64.4	82.2	47.5	77.9	56.4	50.4	95.3	42.8	20.3	58.6	75.4	66.1	75.3	58.8	90.2	81.3	64.2
PointASNL [6]	66.6	70.3	78.1	75.1	65.5	83.0	47.1	76.9	47.4	53.7	95.1	47.5	27.9	63.5	69.8	67.5	75.1	55.3	81.6	80.6	70.3
MinNet42 (5cm) [1]	67.9	81.1	73.4	73.9	64.1	80.4	41.3	75.9	69.6	54.5	93.8	51.8	14.1	62.3	75.7	68.0	72.3	68.4	89.6	82.1	65.1
Our FusionNet (5cm)	**68.8**	70.4	74.1	**75.4**	**65.6**	82.9	**50.1**	74.1	60.9	**54.8**	95.0	**52.2**	**37.1**	63.3	75.6	**71.5**	**77.1**	62.3	86.1	81.4	65.8



(a) FusionNet Architecture (b) Fusion Module

Figure 1: Illustration of the FusionNet architecture and the fusion layer/module. For top-row layers, each voxel consists of a "mini-PointNet" to learn the point representation, while the bottom row learns the voxel representation. (a) FusionNet architecture with 3D UNet as the backbone, in which the fusion modules replace all the original convolutional layers. (b) Illustration of one fusion layer/module. Blank squares represent empty/invalid voxels. One fusion module consists of three efficient feature aggregation steps: 1) regular voxel-based convolutional aggregation (blue arrows), 2) neighborhood-voxel aggregation of point features (red arrows), and 3) inner-voxel point-level circulated aggregation (**black arrows**).

Table 2: Parameters of One Fusion Module in Our FusionNet

Step	**Point Feature Layer**	**Output Shape**	**Step**	**Voxel Feature Layer**	**Output Shape**
input	Point feature as input	$N\times C$	input	Voxel feature as input	$H\times W\times C$
1	$3\times3\times3$ Voxel-MLP	$N\times C$	(1)	$3\times3\times3$ sparse conv, BN, ReLU	$H\times W\times D\times C$
2	concat: layer 1 and point locations	$N\times(C+3)$	(2)	**from layer 3**, Point Avg-pooling	$H\times W\times D\times C$
3	Point-wise FC layer, BN, ReLU	$N\times C$	(3)	concat: (1) and (2)	$H\times W\times D\times 2C$
4	**from layer (3)**, expand/repeat	$N\times C$	(4)	$3\times3\times3$ sparse conv, BN, ReLU	$H\times W\times D\times C$
5	concat: 3 and 4, FC layer, BN, ReLU	$N\times C$	–		——-

Table 3: Parameters of the FusionNet architecture

No.	**Layer Description**	**Output Feature Shapes**	
input	N points as input	$N\times3$ or $N\times 4$	$H\times W\times D\times3$ or 4
1	$3\times3\times3$ Fusion Module	$N\times32$	$H\times W\times D\times32$
2	down-sampling: $2\times2\times2$ conv stride 2, point sample: 1/4	$\frac{1}{4}N\times32$	$\frac{1}{2}H\times\frac{1}{2}W\times\frac{1}{2}D\times32$
3	$3\times3\times3$ Fusion Module	$\frac{1}{4}N\times48$	$\frac{1}{2}H\times\frac{1}{2}W\times\frac{1}{2}D\times48$
4-5	repeat layer 2-3	$\frac{1}{16}N\times64$	$\frac{1}{4}H\times\frac{1}{4}W\times\frac{1}{4}D\times64$
6-7	repeat layer 2-3	$\frac{1}{64}N\times96$	$\frac{1}{8}H\times\frac{1}{8}W\times\frac{1}{8}D\times96$
8-9	repeat layer 2-3	$\frac{1}{256}N\times128$	$\frac{1}{16}H\times\frac{1}{16}W\times\frac{1}{16}D\times128$
10-11	repeat layer 2-3, stride 2, point sample: 1/2	$\frac{1}{512}N\times256$	$\frac{1}{32}H\times\frac{1}{32}W\times\frac{1}{32}D\times256$
12	up-sampling: $2\times2\times2$ deconv stride 2, point upsample: $\times2$	$\frac{1}{256}N\times128$	$\frac{1}{16}H\times\frac{1}{16}W\times\frac{1}{16}D\times128$
13	concat: 12 and 9, $3\times3\times3$ Fusion Module	$\frac{1}{256}N\times128$	$\frac{1}{16}H\times\frac{1}{16}W\times\frac{1}{16}D\times128$
14-15	repeat 12-13 (concat: 14 and 7, point upsample: $\times4$)	$\frac{1}{64}N\times96$	$\frac{1}{8}H\times\frac{1}{8}W\times\frac{1}{8}D\times96$
16-17	repeat 12-13 (concat: 16 and 5)	$\frac{1}{16}N\times64$	$\frac{1}{4}H\times\frac{1}{4}W\times\frac{1}{4}D\times64$
18-19	repeat 12-13 (concat: 18 and 3)	$\frac{1}{4}N\times48$	$\frac{1}{2}H\times\frac{1}{2}W\times\frac{1}{2}D\times48$
20-21	repeat 12-13 (concat: 20 and 1)	$N\times32$	$H\times W\times D\times32$
output	from point feature, point FC-layer (no BN or ReLU)	$N\times$num of classes	

References

[1] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3075–3084, 2019.

[2] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5828–5839, 2017.

[3] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5099–5108, 2017.

[4] Maxim Tatarchenko, Jaesik Park, Vladlen Koltun, and Qian-Yi Zhou. Tangent convolutions for dense prediction in 3d. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3887–3896, 2018.

[5] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[6] Xu Yan, Chaoda Zheng, Zhen Li, Sheng Wang, and Shuguang Cui. Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

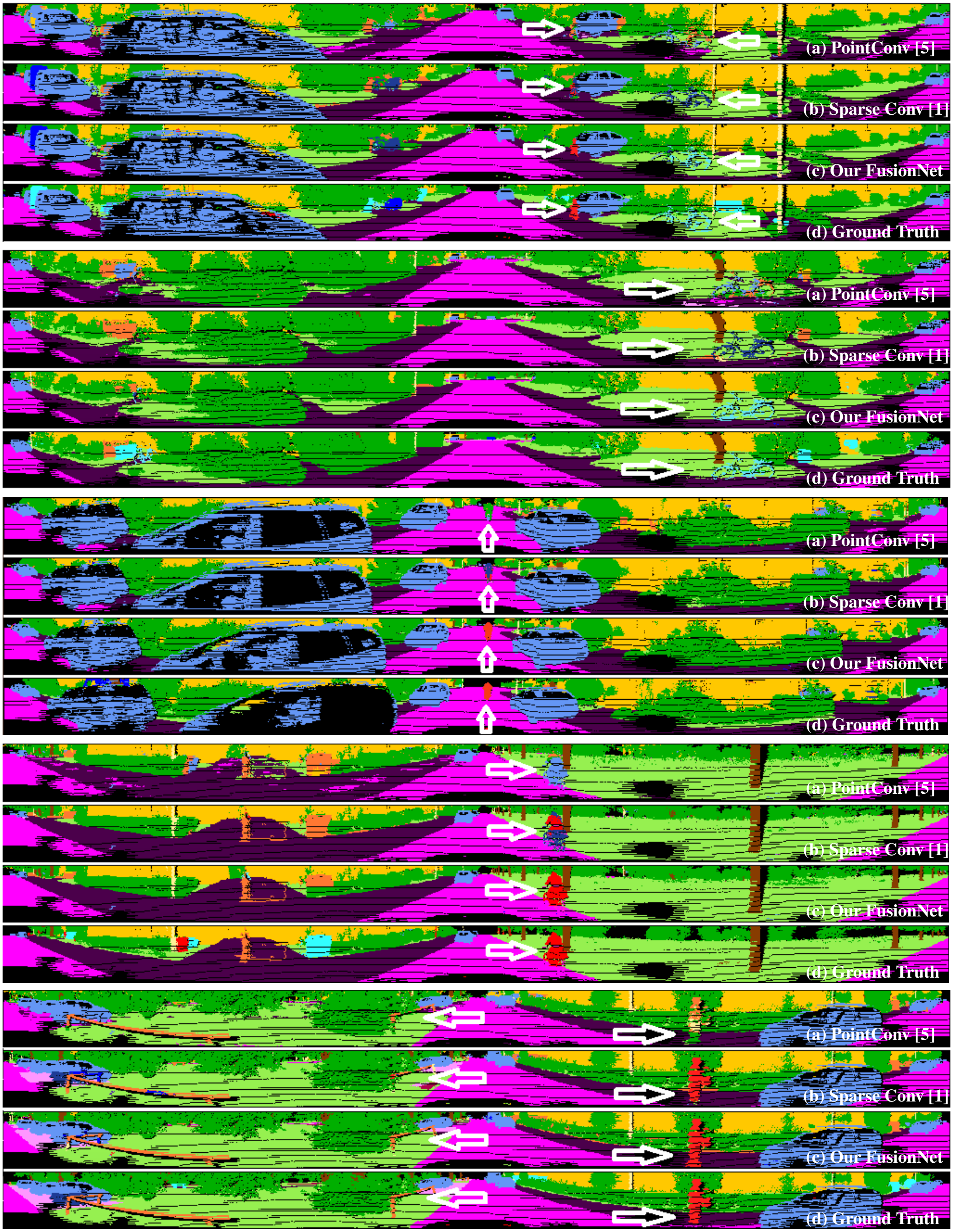


Figure 2: Visualization of the segmentation results of LiDAR point clouds. Points are projected to cylindrical images. (a) State-of-the-art point-wise convolutions [5], (b) state-of-the-art sparse convolutions [1], (c) our FusionNet, (d) ground truths. The improvements are as illustrated by the white arrows.