PointASNL: Robust Point Clouds Processing using Nonlocal Neural Networks with Adaptive Sampling

Supplementary Material

A. Overview

In this supplementary material, we first provide more additional experiments to further verify the superiority of our model in Section B. Besides, we show the our network architecture details in Section C.

B. Additional Experiment

B.1. Part Segmentation

Due to space limitation, we illustrate the part segmentation experiments using man-made synthetic dataset ShapeNet [15], which contains 16,881 shapes from 16 classes and 50 parts. We use the data provided by [11] and adopt the same training and test strategy, i.e., randomly pick 2048 points as the input and concatenate the one-hot encoding of the object label to the last layer.

The quantitative comparisons with the state-of-the-art point-based methods are summarized in Tab. 5. Note that we only compare with methods use 2048 points. When compared with the state-of-the-arts, PointASNL achieves comparable result, which is only slightly lower than RS-CNN [9] using different sampling and voting strategy (as the same reason for classification task).

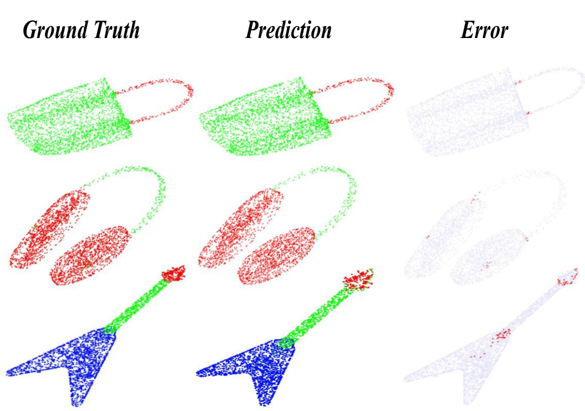

Figure 1. Selected results of part segmentation.

Table 1. The results (%) of four selection strategies on adaptive sampling. For a fair comparison, the number of neighbors is set to be equal in each layer between the two models.

Model	RS	FPS	Average	GF	*ModelNet40*
A	✓		✓		87.9
B		✓	✓		91.5
C	✓			✓	92.3
D		✓		✓	**93.2**

B.2. Selection of Adaptive Sampling

Two variable conditions, i.e., the sampling strategy for initial sampled points and deformation method, are investigated for this issue. Tab. 1 summarizes the results. For the initial sampling points, we chose two strategies, i.e., FPS and random sampling (RS). Also for local coordinate points and feature updates, we compare the effects of using the weight learning by group feature (GF) and simple average of all neighbors' coordinates and features. Note that the number of neighbors is set to be equal for a fair comparison.

As Tab. 1 shows,, if we just use RS sample the initial points and then average their coordinates and features (model A), we will get very low accuracy of 87.9%. However, if we use FPS instead of RS (model B), it can increase to 91.5%. Furthermore, model C and D illustrate the weight learning using group features can largely increase the inference ability of our model. However, if we use RS as sampling strategy, it will cause some accuracy loss while we add the group features learning. This shows that AS module can only finely adjust the distribution of the sampled point cloud instead of 'creating' the missing information.

B.3. Visualization of L-NL Module

We further demonstrate the local-global learning of PointASNL in Fig. 2. In the first layer of the network, PNL can find global points that have similar characteristics with sampled points (e.g., edge and normal vectors). In

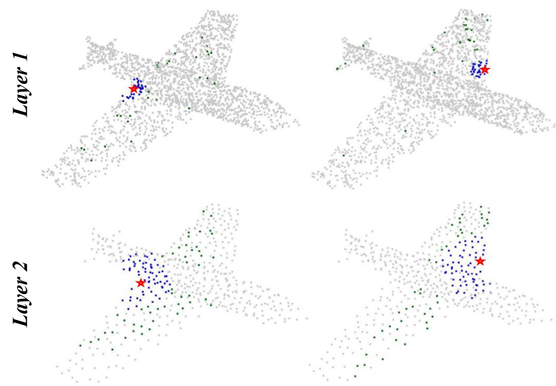

Figure 2. Visualization of local-global learning. For each sampled point (red), we search its local neighbors (blue) and the K points with the highest global response value (green), where K is equal to the number of local neighborhoods.

Table 2. The results (mIoU, (%)) on *ScanNet* v2 validation set with other model setting.

Model	input	grid sample	deeper	mIoU
A	8192 pnt			63.5
B	8192 pnt	✓		64.5
C	10240 pnt	✓		64.8
D	10240 pnt	✓	✓	**66.4**

the second layer, these global highly responsive points have the same semantics information with sampled points, even when sampled points are at the junction of the two different semantics. This is why global features can help sampled points to better aggregate local features.

B.4. Visualization of Adaptive Sampling

When the input point cloud has a lot of noise, adaptive sampling has the ability to ensure the distribution of the sample point manifold. We give some examples of comparative visualization in Fig. 6 to prove the robustness of the AS module. As can be seen from Fig. 6, AS module can effectively reduce noise in the sample points and maintain the shape of the sampled manifold.

B.5. Further Improvement of PointASNL

The result in manuscript only conducts a fair comparison (same model structure and training strategy) against appealing recent methods under the same setting of PointNet++ [11]. However, our PointASNL can still achieve further improvement if we use other data pre-processing or deeper structure.

As shown in Tab. 2, our PointASNL can still improve its performance if we use grid sampling pre-processing, more input points and deeper structure. As for the structure of deeper PointASNL, we add an additional point local cell at the end of each layer. Furthermore, by conducting ensemble learning with model from different training epochs, we can finally achieve 66.6% on *ScanNet* benchmark.

Table 3. Network Configurations.

Layer	npoint	nsample	as_neighbor	mlp
Task	*Classification*			
1	512	32	12	[64,64,128]
2	128	64	12	[128,128,256]
3	1	-	-	[256,512,1024]
Task	*Segmentation*			
1	1024	32	8	[32,32,64]
2	256	32	4	[64,64,128]
3	64	32	0	[128,128,256]
4	36	32	0	[256,256,512]

B.6. Concrete Results

In this section, we give our detailed results on the S3DIS (Tab.6 and Tab.7) and SemanticKITTI (Tab.8) dataset as a benchmark for future work. *ScanNet* [1] is an online benchmark, the class scores can be found on its website. Furthermore, we provide more visualization results to illustrate the performance of our model in complicated scenes.

C. Network Architectures

C.1. Layer Setting

For each encoder layer, it can be written as the following form: *Abstraction(npoint, nsample, as_neighbor, mlp)*, where *npoint* is the number of sampled points of layer. *nsample* and *as_neighbor* are number of group neighbors in point local cell and AS module, and they share the same k-NN query. *mlp* is a list for MLP construction in our layers and used in both PL and PNL. Tab. 3 shows the configuration of PointASNL on both classification and segmenttaion tasks.

C.2. Loss Function

Like other previous works, we use cross entropy (CE) loss in classification and part segmentation, and consider the number of each category as weights in semantic segmentation. Furthermore, in order to avoid the sampled points being too close to each other in some local areas after the AS module transformation, we also use Repulsion Loss [16] to restrict the deformation of sampled point clouds. In particular, we only use this loss in the first layer since it has the highest point density. The Repulsion loss does not bring any performance improvement, but the training procedure is significantly accelerated.

Altogether, we train the PointASNL in an end-to-end

Table 4. The running time on *ModelNet40* and *ScanNet* datasets.

Dataset	process	input	time (s/sample)
ModelNet40	Training	1024 pnt	0.00046
ScanNet	Training	8192 pnt	0.17611
ModelNet40	Inference	1024 pnt	0.00024
ScanNet	Inference	8192 pnt	0.11363

manner by minimizing the following joint loss function:

$$L(\theta) = L_{\text{CE}} + \alpha L_{\text{Rep}} + \beta ||\theta||^2,$$
$$L_{\text{Rep}} = \sum_{i=0}^{N} \sum_{i' \in N(x_i)} w(||x_{i'} - x_i||), \tag{1}$$

where θ indicates the parameters in our network, $\alpha = 0.01$ balances the CE loss and Repulsion loss, and β denotes the multiplier of the weight decay. For Repulsion loss, it penalizes the sampled point x_i only when it is too close to its neighboring points $x_{i'} \in N(x_i)$. $w(r) = e^{r^2/h^2}$ is a fast-decaying weight function and N is the number of sampled points.

The Repulsion loss also ensures that each sample point itself has a larger weight in the AS module in a relatively constant density, which makes them cannot move too far.

C.3. Model Speeds

Tab. 4 shows the statistics of our models on different datasets. Since our L-NL module only uses sampled points as query points instead of the whole point cloud, the AS module and NL cell can be both efficient and effective with the bottleneck structures (only around 30% extra time).

References

[1] 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*, pages 5828–5839, 2017. 2

[2] Qiangui Huang, Weiyue Wang, and Ulrich Neumann. Recurrent slice networks for 3d segmentation of point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2626–2635, 2018. 4

[3] Li Jiang, Hengshuang Zhao, Shu Liu, Xiaoyong Shen, Chi-Wing Fu, and Jiaya Jia. Hierarchical point-edge interaction network for point cloud semantic segmentation. 2019. 4

[4] Artem Komarichev, Zichun Zhong, and Jing Hua. A-cnn: Annularly convolutional neural networks on point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7421–7430, 2019. 4

[5] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4558–4567, 2018. 4

[6] Jiaxin Li, Ben M Chen, and Gim Hee Lee. So-net: Self-organizing network for point cloud analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9397–9406, 2018. 4

[7] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems*, pages 820–830, 2018. 4

[8] Xinhai Liu, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Point2sequence: Learning the shape representation of 3d point clouds with an attention-based sequence to sequence network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8778–8785, 2019. 4

[9] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019. 1, 4

[10] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 4

[11] 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*, pages 5099–5108, 2017. 1, 2, 4

[12] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. Splatnet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2530–2539, 2018. 4

[13] 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*, pages 3887–3896, 2018. 4

[14] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *arXiv preprint arXiv:1801.07829*, 2018. 4

[15] Li Yi, Vladimir G Kim, Duygu Ceylan, I Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, Leonidas Guibas, et al. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (TOG)*, 35(6):210, 2016. 1

[16] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2799, 2018. 2

[17] Hengshuang Zhao, Li Jiang, Chi-Wing Fu, and Jiaya Jia. Pointweb: Enhancing local neighborhood features for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5565–5573, 2019. 4

Table 5. Part segmentation performance with part-avaraged IoU on *ShapeNetPart*.

Method	pIoU	areo	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate board	table
#shapes		2690	76	55	898	3758	69	787	392	1547	451	202	184	286	66	152	5271
PointNet [10]	83.7	83.4	78.7	82.5	74.9	89.6	73.0	91.5	85.9	80.8	95.3	65.2	93.0	81.2	57.9	72.8	80.6
SO-Net [6]	84.9	82.8	77.8	**88.0**	77.3	90.6	73.5	90.7	83.9	82.8	94.8	69.1	94.2	80.9	53.1	72.9	83.0
PointNet++ [11]	85.1	82.4	79.0	87.7	77.3	90.8	71.8	91.0	85.9	83.7	95.3	71.6	94.1	81.3	58.7	76.4	82.6
DGCNN [14]	85.1	**84.2**	83.7	84.4	77.1	90.9	78.5	91.5	87.3	82.9	96.0	67.8	93.3	82.6	59.7	75.5	82.0
P2Sequence [8]	85.2	82.6	81.8	87.5	77.3	90.8	77.1	91.1	86.9	83.9	95.7	70.8	94.6	79.3	58.1	75.2	82.8
PointCNN [7]	86.1	84.1	**86.5**	86.0	**80.8**	90.6	79.7	92.3	**88.4**	85.3	**96.1**	77.2	95.2	**84.2**	**64.2**	**80.0**	83.0
RS-CNN [9]	**86.2**	83.5	84.8	88.8	79.6	91.2	**81.1**	**91.6**	88.4	**86.0**	96.0	73.7	94.1	83.4	60.5	77.7	**83.6**
PointASNL	86.1	84.1	84.7	87.9	79.7	**92.2**	73.7	91.0	87.2	84.2	95.8	**74.4**	**95.2**	81.0	63.0	76.3	83.2

Table 6. Semantic segmentation results on *S3DIS* dataset evaluated on Area 5.

Method	OA	mAcc	mIoU	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
PointNet [10]	-	49.0	41.1	88.8	97.3	69.8	**0.1**	3.9	46.3	10.8	52.6	58.9	40.3	5.9	26.4	33.2
PointCNN [7]	85.9	63.9	57.3	92.3	98.2	**79.4**	0.0	17.6	22.8	62.1	74.4	80.6	31.7	66.7	62.1	**56.7**
PointWeb [17]	87.0	66.6	60.3	92.0	**98.5**	79.4	0.0	21.1	59.7	34.8	76.3	**88.3**	46.9	**69.3**	64.9	52.5
HPEIN [3]	87.2	68.3	61.9	91.5	98.2	81.4	0.0	23.3	**65.3**	40.0	75.5	87.7	**58.5**	67.8	**65.6**	49.7
PointASNL	**87.7**	**68.5**	**62.6**	**94.3**	98.4	79.1	0.0	**26.7**	55.2	**66.2**	**83.3**	86.8	47.6	68.3	56.4	52.1

Table 7. Semantic segmentation results on the *S3DIS* dataset with 6-fold cross validation.

Method	OA	mAcc	mIoU	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
PointNet [10]	78.5	66.2	47.6	88.0	88.7	69.3	42.4	23.1	47.5	51.6	42.0	54.1	38.2	9.6	29.4	35.2
RSNet [2]	-	66.5	56.5	92.5	92.8	78.6	32.8	34.4	51.6	68.1	59.7	60.1	16.4	50.2	44.9	52.0
A-CNN [4]	87.3	-	62.9	92.4	96.4	79.2	59.5	34.2	56.3	65.0	66.5	**78.0**	28.5	56.9	48.0	56.8
PointCNN [7]	88.1	75.6	65.4	94.8	97.3	75.8	**63.3**	**51.7**	58.4	57.2	**71.6**	69.1	39.1	61.2	52.2	58.6
PointWeb [17]	87.3	76.2	66.7	93.5	94.2	80.8	52.4	41.3	64.9	68.1	71.4	67.1	**50.3**	**62.7**	62.2	58.5
PointASNL	**88.8**	**79.0**	**68.7**	**95.3**	**97.9**	**81.9**	47.0	48.0	**67.3**	**70.5**	71.3	77.8	50.7	60.4	**63.0**	**62.8**

Table 8. Semantic segmentation results on the *SemanticKITTI*.

Method	mIoU	road	sidewalk	parking	other-ground	building	car	truck	bicycle	motorcycle	other-vehicle	vegetation	trunk	terrain	person	bicyclist	motorcyclist	fence	pole	traffic sign
PointNet [10]	14.6	61.6	35.7	15.8	1.4	41.4	46.3	0.1	1.3	0.3	0.8	31.0	4.6	17.6	0.2	0.2	0.0	12.9	2.4	3.7
SPGraph [5]	17.4	45.0	28.5	0.6	0.6	64.3	49.3	0.1	0.2	0.2	0.8	48.9	27.2	24.6	0.3	2.7	0.1	20.8	15.9	0.8
SPLATNet [12]	18.4	64.6	39.1	0.4	0.0	58.3	58.2	0.0	0.0	0.0	0.0	71.1	9.9	19.3	0.0	0.0	0.0	23.1	5.6	0.0
PointNet++ [11]	20.1	72.0	41.8	18.7	5.6	62.3	53.7	0.9	1.9	0.2	0.2	46.5	13.8	30.0	0.9	1.0	0.0	16.9	6.0	8.9
TangentConv [13]	40.9	83.9	63.9	**33.4**	**15.4**	**83.4**	**90.8**	15.2	**2.7**	16.5	12.1	79.5	49.3	58.1	23.0	28.4	**8.1**	**49.0**	35.8	28.5
PointASNL	**46.8**	**87.4**	**74.3**	24.3	1.8	83.1	87.9	**39.0**	0.0	**25.1**	**29.2**	**84.1**	**52.2**	**70.6**	**34.2**	**57.6**	0.0	**43.9**	**57.8**	**36.9**

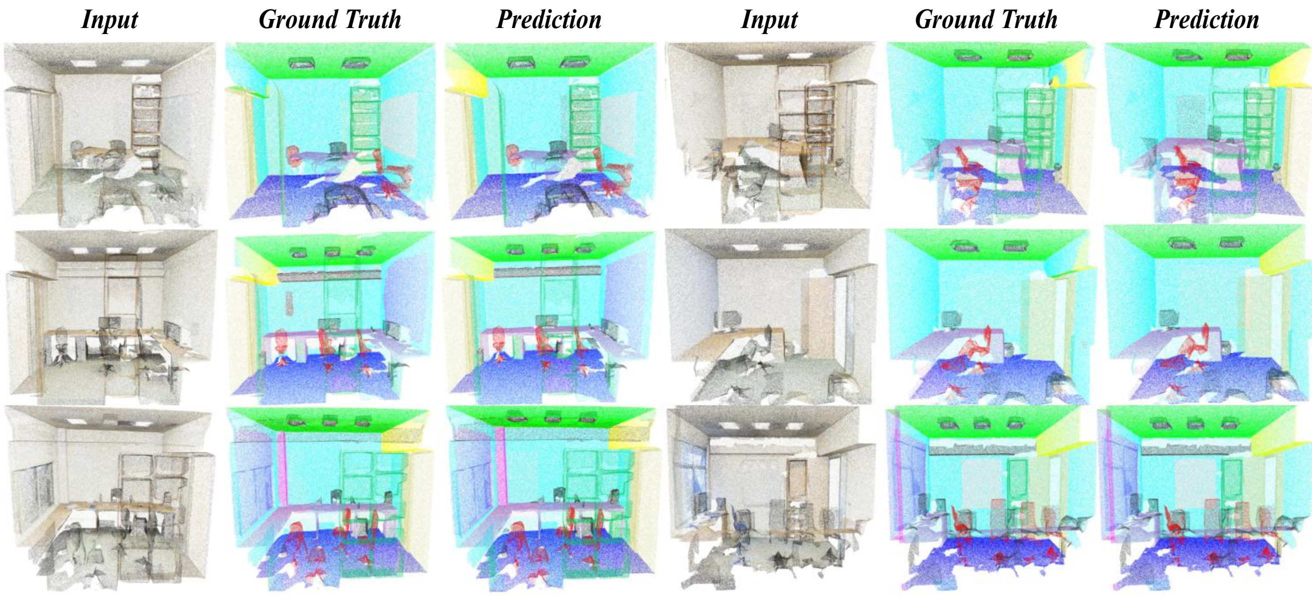

Figure 3. More examples on *S3DIS* datasets.

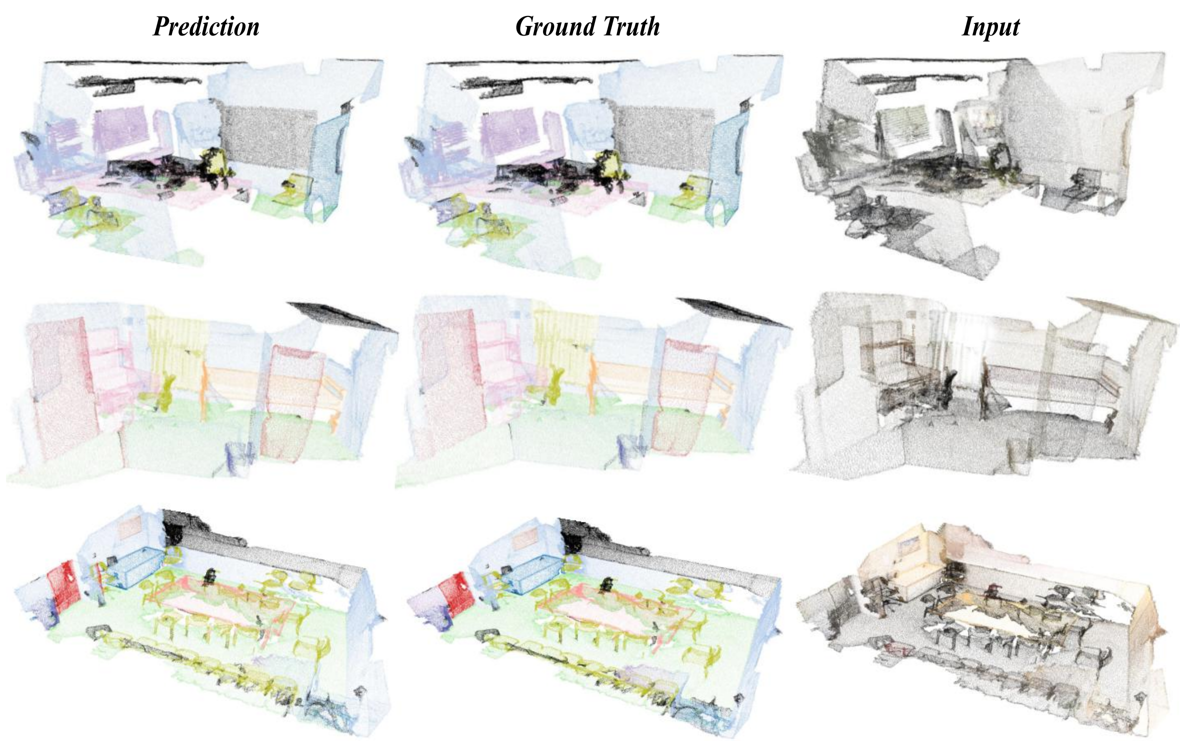

Figure 4. More examples on *ScanNet* datasets.

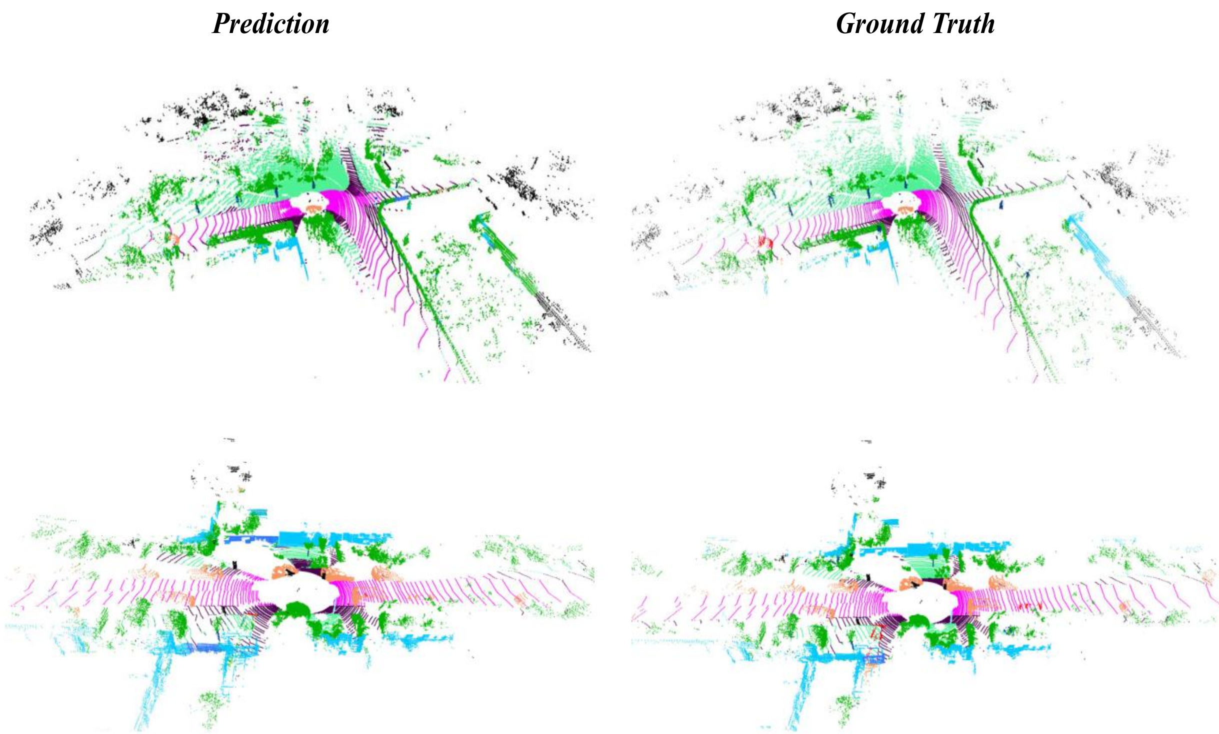

Figure 5. More examples on *SemanticKITTI* datasets.

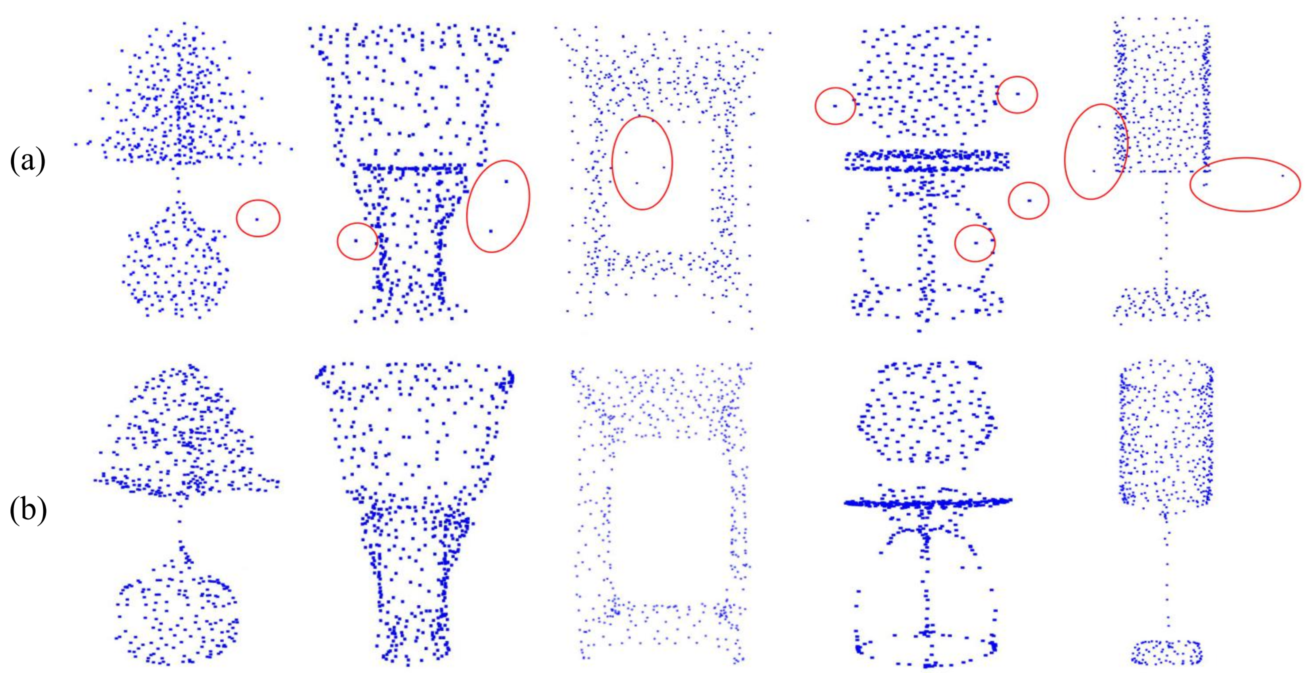

Figure 6. Visualized results of AS module. (a) Sampled points via farthest point sampling (FPS). (b) Sampled points ajusted by AS module.