Supplementary Material: Semantic Segmentation for Real Point Cloud Scenes via Bilateral Augmentation and Adaptive Fusion

Shi Qiu[1,2], Saeed Anwar[1,2] and Nick Barnes[1]
[1]Australian National University, [2]Data61-CSIRO, Australia
{shi.qiu, saeed.anwar, nick.barnes}@anu.edu.au

1. Overview

This supplementary material provides more network details, experimental results, and visualizations of our semantic segmentation results.

2. Network Details

In Figure 2 of the main paper, we present the general architecture of our semantic segmentation network as well as the structure of the Bilateral Context Block. In this section, we provide more details about the different components of our network.

2.1. Key Modules

Feature Extractor: As stated, we apply a single-layer MLP containing eight 1×1 kernels to extract the semantic context $\mathcal{F}$ from the input information $\mathcal{I} \in \mathbb{R}^{N \times C_{in}}$, where N is the number of input points. Hence, $\mathcal{F}$ is acquired as:

$$\mathcal{F} = \mathrm{ReLU}\Big(\mathrm{BN}\big(\mathrm{Conv}^{8}_{1\times1}(\mathcal{I})\big)\Big), \quad \mathcal{F} \in \mathbb{R}^{N\times 8};$$

where Conv denotes a convolution layer whose subscript is the kernel size, and the superscript is the number of kernels. BN represents a batch normalization layer, while ReLU is a ReLU activation layer. Later on, $\mathcal{F}$ is forwarded to the Bilateral Context Module, together with the 3D coordinates $\mathcal{P} \in \mathbb{R}^{N\times 3}$.

Bilateral Context Module: In practice, we apply five Bilateral Context Blocks with Farthest Point Sampling (FPS) to realize the Bilateral Context Module ($\mathcal{B}$). Using the same annotations of the main paper's Section 4.2, the extracted multi-resolution feature maps are:

$$\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4, \mathcal{S}_5\} = \mathcal{B}(\mathcal{P}, \mathcal{F});$$

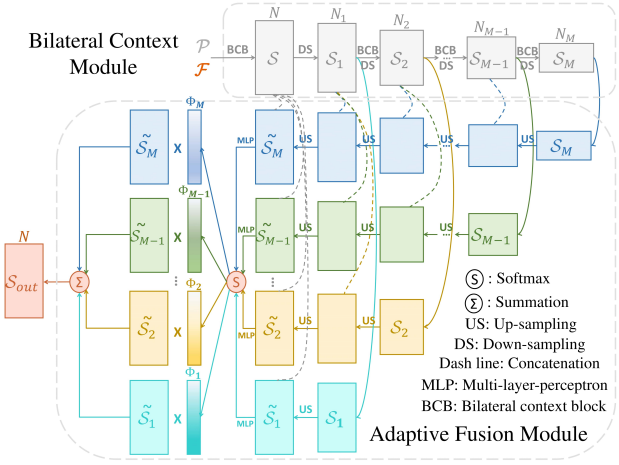

Figure 1: The architecture of the Adaptive Fusion Module. All the annotations are consistent with the items in Section 3 of the main paper.

where:

$$\mathcal{S}_1 \in \mathbb{R}^{\frac{N}{4}\times 32}, \quad \mathcal{S}_2 \in \mathbb{R}^{\frac{N}{16}\times 128}, \quad \mathcal{S}_3 \in \mathbb{R}^{\frac{N}{64}\times 256},$$

$$\mathcal{S}_4 \in \mathbb{R}^{\frac{N}{256}\times 512}, \quad \mathcal{S}_5 \in \mathbb{R}^{\frac{N}{512}\times 1024}.$$

Particularly, the downsampling ratios and feature dimensions are simply adopted from [6], since we mainly focus on the structure design rather than fine-tuning the hyper-parameters in this work.

Adaptive Fusion Module: In addition to Algorithm 1 and Section 3.2 in the main paper, we also illustrate the architecture of the Adaptive Fusion Module in Figure 1 as a complement. As described in Section 4.3 of the main paper, we gradually upsample the extracted feature maps $\{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4, \mathcal{S}_5\}$, respectively.

Testing Area	**mAcc**	**OA**	**mIoU**	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board	clutter
Area 1	87.7	89.5	76.3	96.5	95.4	80.3	65.4	58.8	78.0	84.3	70.7	82.9	78.0	60.9	73.2	67.9
Area 2	71.1	86.6	57.8	87.1	95.1	80.0	19.8	33.3	47.5	69.3	45.6	83.1	52.8	50.7	33.1	54.4
Area 3	89.7	91.7	80.0	95.8	98.2	83.3	74.4	40.5	86.0	88.5	74.4	83.7	79.0	73.6	88.9	73.9
Area 4	77.9	86.1	64.3	94.8	97.1	78.6	53.0	48.6	30.8	61.0	67.4	77.0	70.1	51.3	44.8	61.6
Area 5	73.1	88.9	65.4	92.9	97.9	82.3	0.0	23.1	65.5	64.9	78.5	87.5	61.4	70.7	68.7	57.2
Area 6	92.0	92.5	81.8	96.4	97.5	86.2	79.9	81.0	78.5	90.1	77.1	88.1	65.1	72.4	79.7	71.2
6-fold	83.1	88.9	72.2	93.3	96.8	81.6	61.9	49.5	65.4	73.3	72.0	83.7	67.5	64.3	67.0	62.4

Table 1: Detailed semantic segmentation results (%) on *S3DIS* [1] dataset. (**mAcc**: average class accuracy, **OA**: overall accuracy, **mIoU**: mean Intersection-over-Union."6-fold": 6-fold cross-validation result.)

Method	**OA**	**mIoU**	man-made terrain	natural terrain	high vegetation	low vegetation	buildings	hard scape	scanning artefacts	cars
SnapNet [3]	88.6	59.1	82.0	77.3	79.7	22.9	91.1	18.4	37.3	64.4
ShellNet [10]	93.2	69.3	96.3	90.4	83.9	41.0	94.2	34.7	43.9	70.2
GACNet [9]	91.9	70.8	86.4	77.7	**88.5**	**60.6**	94.2	37.3	43.5	77.8
SPG [7]	94.0	73.2	**97.4**	92.6	87.9	44.0	83.2	31.0	63.5	76.2
KPConv [8]	92.9	74.6	90.9	82.2	84.2	47.9	94.9	40.0	**77.3**	**79.7**
RandLA-Net [6]	**94.8**	**77.4**	95.6	91.4	86.6	51.5	**95.7**	**51.5**	69.8	76.8
Ours	94.3	75.3	96.3	**93.7**	87.7	48.1	94.6	43.8	58.2	79.5

Table 2: Semantic segmentation (reduced-8) results (%) on *Semantic3D* [5] dataset.

In this case, the upsampled full-sized feature maps are $\{\tilde{\mathcal{S}}_1, \tilde{\mathcal{S}}_2, \tilde{\mathcal{S}}_3, \tilde{\mathcal{S}}_4, \tilde{\mathcal{S}}_5\}$, all of which are in $\mathbb{R}^{N\times 32}$.

Then, for each upsampled full-sized feature map, we use a fully-connected layer (FC, and its superscript indicates the number of kernels) to summarize the point-level information:

$$\phi_m = \mathrm{FC}^1(\tilde{\mathcal{S}_m}), \quad \phi_m \in \mathbb{R}^N;$$

where $\forall \tilde{\mathcal{S}_m} \in \{\tilde{\mathcal{S}}_1, \tilde{\mathcal{S}}_2, \tilde{\mathcal{S}}_3, \tilde{\mathcal{S}}_4, \tilde{\mathcal{S}}_5\}$. Subsequently, we concatenate the $\{\phi_1, \phi_2, \phi_3, \phi_4, \phi_5\}$, and point-wisely normalize them using softmax function:

$$\Phi = \mathrm{softmax}\big(\mathrm{concat}(\phi_1, \phi_2, \phi_3, \phi_4, \phi_5)\big), \Phi \in \mathbb{R}^{N\times 5}.$$

Next, we separate Φ channel-by-channel, and obtain the fusion parameters: $\{\Phi_1, \Phi_2, \Phi_3, \Phi_4, \Phi_5\}$, all of which are in $\mathbb{R}^N$. Hence, the point-level adaptively fused feature map is calculated as:

$$\mathcal{S}_{out} = \Phi_1 \times \tilde{\mathcal{S}}_1 + \Phi_2 \times \tilde{\mathcal{S}}_2 + \Phi_3 \times \tilde{\mathcal{S}}_3 + \Phi_4 \times \tilde{\mathcal{S}}_4 + \Phi_5 \times \tilde{\mathcal{S}}_5,$$

where $\mathcal{S}_{out} \in \mathbb{R}^{N\times 32}$.

2.2. Predictions

Based on $\mathcal{S}_{out}$, we utilize three fully-connected layers and a drop-out layer (DP, and the drop-out ratio shows at the superscript) to predict the confidence scores for all Q candidate semantic classes:

$$\mathcal{V}_{pred} = \mathrm{FC}^Q\Big(\mathrm{DP}^{0.5}\Big(\mathrm{FC}^{32}\big(\mathrm{FC}^{64}(\mathcal{S}_{out})\big)\Big)\Big),$$

where $\mathcal{V}_{pred} \in \mathbb{R}^{N\times Q}$.

2.3. Loss Function

Equation 7 of the main paper formulates the overall loss $\mathcal{L}_{all}$ of our network based on the cross-entropy loss $\mathcal{L}_{CE}$ and the augmentation loss $\mathcal{L}_m$ for each Bilateral Context Block.

In practice, our Bilateral Context Module gradually processes a decreasing number of points ($N \rightarrow \frac{N}{4} \rightarrow \frac{N}{16} \rightarrow \frac{N}{64} \rightarrow \frac{N}{256}$) through five blocks. Empirically, we set the weights $\{0.1, 0.1, 0.3, 0.5, 0.5\}$ for the corresponding five augmentation losses, since we aim to

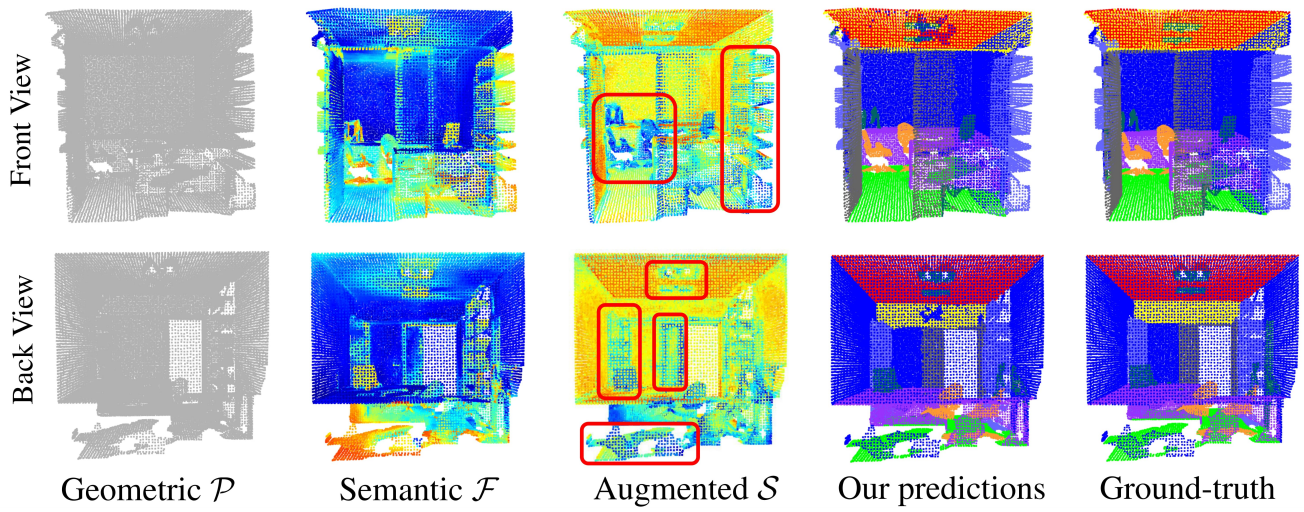

Figure 2: Visualization of intermediate features and semantic segmentation results for an office scene in *S3DIS* [1] dataset. $\mathcal{P}$ denotes the 3D coordinates of the point cloud, and $\mathcal{F}$ presents the semantic information acquired by the Feature Extractor (Section 4.1 in the main paper). Further, $\mathcal{S}$ means the output of our Bilateral Context Block (Section 3.1).

Model	Description	mIoU (%)
N_0	Baseline model	60.8
N_1	Efficient model	64.8
N_2	Dilated model	62.5
N_3	Equal-weighted model	64.0
N_4	Simplified model	63.5
N_5	**Proposed model**	**65.4**

Table 3: Ablation study about different variants of our network, tested on Area 5, *S3DIS* [1] dataset.

provide more penalties for lower-resolution processing. Therefore, the overall loss for our network is:

$$\begin{aligned}\mathcal{L}_{all} =& \mathcal{L}_{CE} + \\ & 0.1 \cdot \mathcal{L}_1 + 0.1 \cdot \mathcal{L}_2 + \\ & 0.3 \cdot \mathcal{L}_3 + 0.5 \cdot \mathcal{L}_4 + 0.5 \cdot \mathcal{L}_5.\end{aligned}$$

3. Experiments

3.1. Areas of S3DIS

We include more experimental data about our network's semantic segmentation performance. To be specific, Table 1 shows our results for each area in the S3DIS dataset, including overall accuracy, average class accuracy, and concrete IoUs for 13 semantic classes. To evaluate each area, we apply the rest five areas as the training set.

Layer		1	2	3	4	5
#Points		40960	10240	2560	640	160
3D Space	**Mean**	$\downarrow 12$	$\downarrow 24$	$\downarrow 47$	$\downarrow 85$	$\downarrow 154$
	Variance	$\downarrow 0.1$	$\downarrow 0.2$	$\downarrow 0.5$	$\downarrow 2$	$\downarrow 13$
Feature Space	**Mean**	$\downarrow 45$	$\downarrow 693$	$\downarrow 814$	$\downarrow 124$	$\downarrow 317$
	Variance	$\downarrow 11.9$	$\downarrow 16.3$	$\downarrow 24.7$	$\downarrow 46.2$	$\downarrow 104$

Table 4: The general *changes* ($\times 10^{-3}$) of neighborhoods by involving bilateral offsets.

3.2. Reduced-8 Semantic3D

Further, Table 2 presents our online evaluation results on the smaller test set (*i.e.*, reduced-8, which has four scenes including about 0.1 billion points) of the Semantic3D dataset. Comparing with Table 2 in the main paper (*i.e.*, results of semantic-8, which contains 15 scenes with 2 billion points), we conclude that our semantic segmentation performance regarding large-scale data is relatively better.

3.3. Ablation Study

In addition to the specific ablation studies (Section 5.3 in the main paper) about our Bilateral Context

Block and Adaptive Fusion Module respectively, we also conduct an ablation study to investigate some variants of our network:

- **Baseline model:** We replace both our Bilateral Context Block and Adaptive Fusion Module with their baseline forms, which are explained in the ablation studies of the main paper.
- **Efficient model:** We apply the random sampling instead of the Farthest Point Sampling (FPS).
- **Dilated model:** We use dilated-knn [4] to search the neighbors of each point, in order to increase the size of point's receptive field. The dilated factor $d = 2$.
- **Equal-weighted model:** We set an equal weight ($\omega_i = 0.3$) for all of the augmentation losses in Equation 7 (*i.e.*, calculating the overall loss $\mathcal{L}_{all}$) of the main paper.
- **Simplified model:** We only study four resolutions of the point cloud through the Bilateral Context Module. The number of points decreases as: $N \rightarrow \frac{N}{4} \rightarrow \frac{N}{16} \rightarrow \frac{N}{64}$, while the number of channels goes as: $16 \rightarrow 64 \rightarrow 128 \rightarrow 256$.

Table 3 indicates that such an efficient random sampling (N_1) cannot perform as effectively as FPS does since the randomly sampled subsets can hardly retain the integrity of inherent geometry. As there is always a trade-off between the network's efficiency and effectiveness, we look forward to better balancing them in future work. Besides, increasing the size of the point's receptive field (N_2) as [4] may not help in our case. Further, we observe that it is not optimal to use the equal-weighted Bilateral Context Blocks (N_3) for multi-resolution point clouds. Moreover, our network can be flexibly assembled: for an instance of model N_4 that consists of fewer blocks, even though the performance is reduced, it consumes less GPU memory.

4. Visualization

4.1. Bilateral Context Block

In Figure 2, we present the Bilateral Context Block's output features in a heat-map view. Particularly, we observe that the Bilateral Context Block can clearly raise different responses for close points (in red frames) that are in different semantic classes.

Besides, we calculate the average neighbor-to-centroid Euclidean-distances and average neighborhood variances in 3D space (Equation 1 in the main paper) and feature space (Equation 2), using the S3DIS samples. Table 4 shows that shifted neighbors get closer to centroids as expected, in both 3D and feature spaces. Further, the variances inside the neighborhoods also drop. In general, the shifted neighbors tend to form compact neighborhoods.

4.2. Visualizations and Failure Cases

We provide more visualizations of our semantic segmentation network's outputs and some failure cases. Specifically, Figure 3 presents our results on six different types of rooms, which are *conference, WC, storage, hallway, lobby, office* rooms, respectively. Unfortunately, we find that the proposed method is not competent enough for distinguishing the objects that are in similar shapes. The main reason is that the network relies on the local neighborhood of each point, while lacks the geometric information about the specific object that each point belongs to. In the 3rd row of Figure 3, *beam* is incorrectly classified as *door* since it looks like the doorframes; while *wall* is wrongly predicted as *board* or *clutter* in the rest of rows.

In Figure 4, we show the general semantic segmentation performances on some large-scale point clouds of typical urban and rural scenes. Although the ground-truths of Semantic3D's test set are unavailable, our semantic predictions of these scenes are visually plausible as the dataset contains a huge amount of points labeled in just a few semantic classes.

In addition, we compare our results against the ground-truths on the validation set (*i.e.*, Sequence 08) of SemanticKITTI dataset in Figure 5. Particularly, we illustrate some 3D point cloud scenes in the views of 2D panorama, in order to clearly show the failure cases (highlighted in red color). In fact, the proposed network is able to find some small objects that are semantically different from the background, however, the predictions are not accurate enough since we only use the 3D coordinates as input. As SemanticKITTI is made up of the sequences of scans, in the future, we will take the temporal information into account.

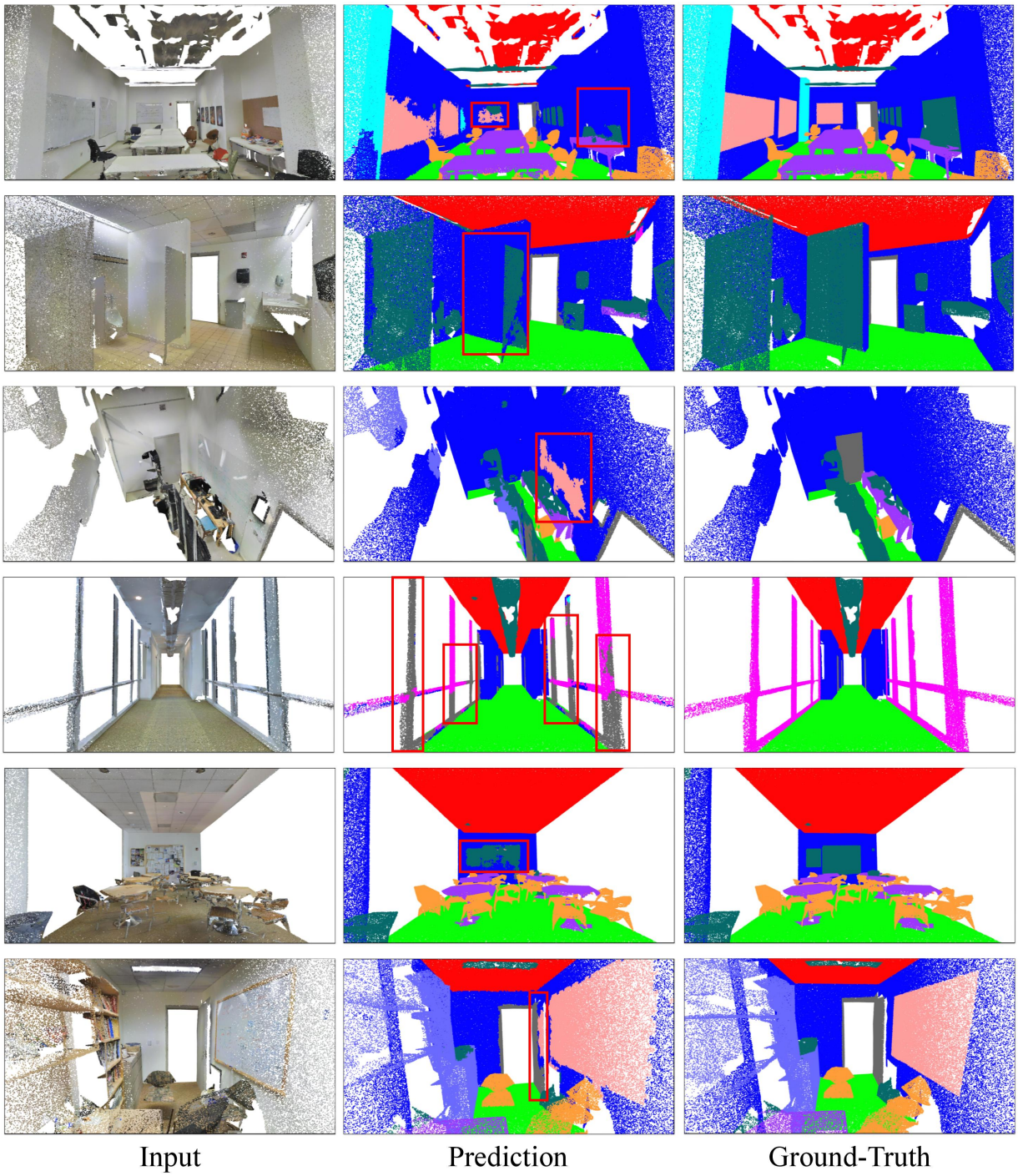

Figure 3: Examples of our semantic segmentation results of *S3DIS* [1] dataset. The first column presents the input point cloud scenes ("Input") of some indoor rooms. The second column shows the semantic segmentation predictions of our network ("Prediction"), while the last column indicates the ground-truths ("Ground-Truth"). The main differences are highlighted in red frames.

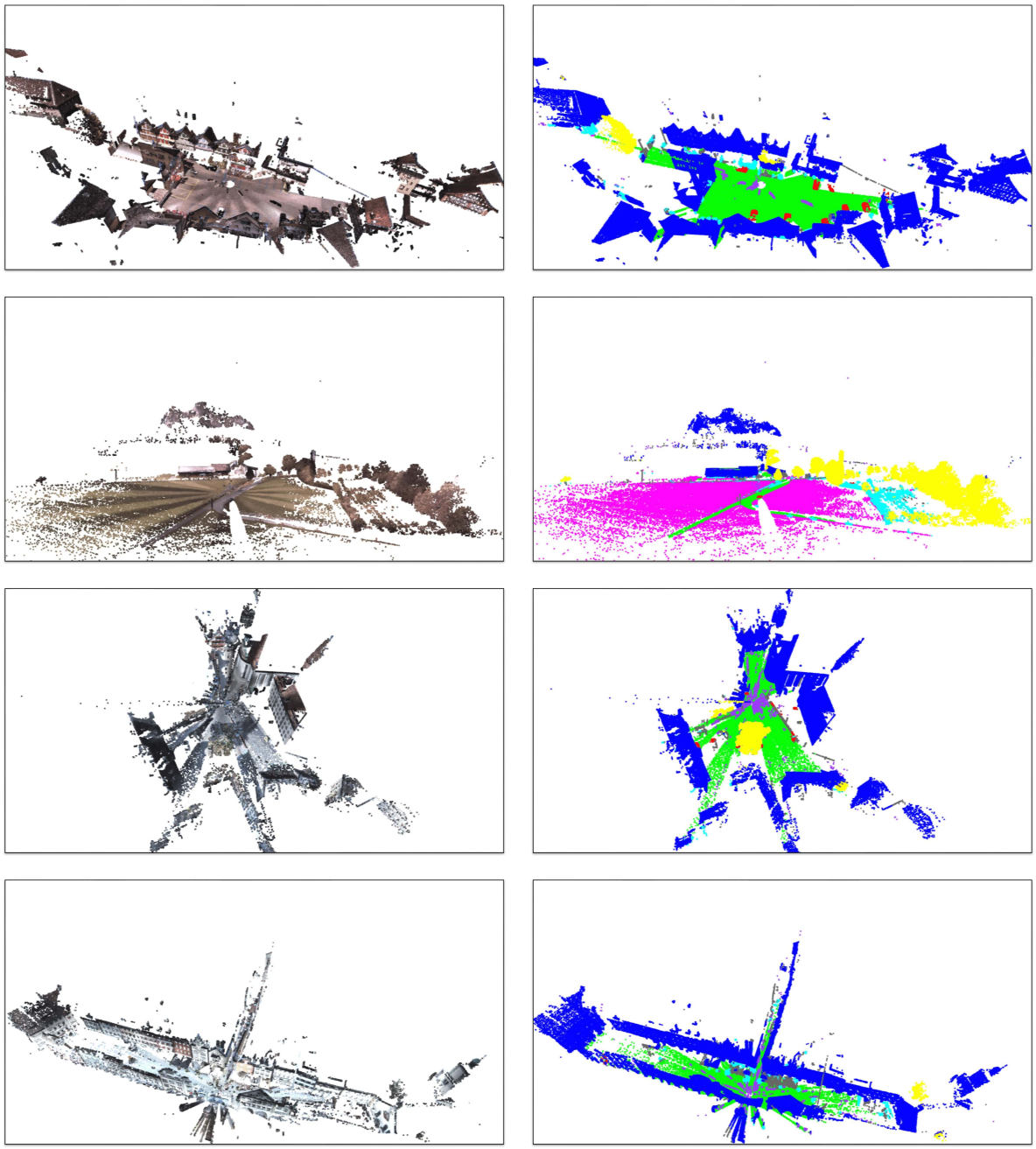

Figure 4: Examples of our semantic segmentation predictions of *Semantic3D* [5] dataset. The first row is about an urban square, the second one shows a rural farm, the third one illustrates a cathedral scene, and the last is scanned from a street view.

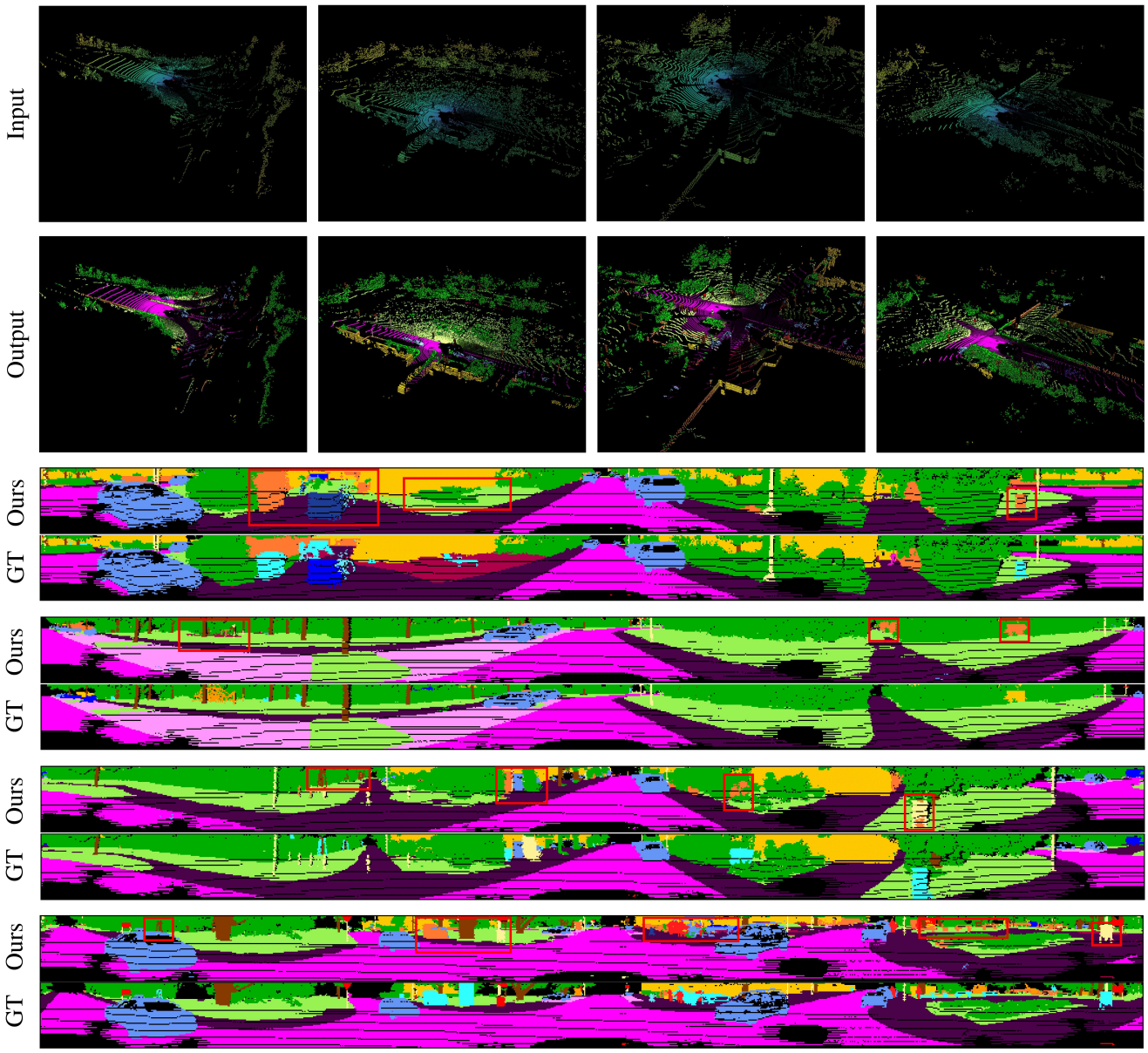

Figure 5: Examples of our semantic segmentation predictions of *SemanticKITTI* [2] dataset. The first two rows show the general 3D views of the input traffic scenarios ("Input") and our semantic segmentation outputs ("Output"), respectively. The remaining rows compare our predictions ("Ours") and the ground-truths ("GT") in 2D panorama views, where the failure cases are highlighted in red frames.

References

[1] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 2, 3, 5

[2] 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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 9297–9307, 2019. 7

[3] Alexandre Boulch, Joris Guerry, Bertrand Le Saux, and Nicolas Audebert. Snapnet: 3d point cloud semantic labeling with 2d deep segmentation networks. *Computers & Graphics*, 71:189–198, 2018. 2

[4] Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. Dilated point convolutions: On the receptive field size of point convolutions on 3d point clouds. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9463–9469. IEEE, 2020. 4

[5] Timo Hackel, Nikolay Savinov, Lubor Ladicky, Jan D Wegner, Konrad Schindler, and Marc Pollefeys. Semantic3d. net: A new large-scale point cloud classification benchmark. *arXiv preprint arXiv:1704.03847*, 2017. 2, 6

[6] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11108–11117, 2020. 1, 2

[7] 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. 2

[8] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6411–6420, 2019. 2

[9] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10296–10305, 2019. 2

[10] Zhiyuan Zhang, Binh-Son Hua, and Sai-Kit Yeung. Shellnet: Efficient point cloud convolutional neural networks using concentric shells statistics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1607–1616, 2019. 2