Revisiting Retentive Networks for Fast Range-View 3D LiDAR Semantic Segmentation – Supplementary Material –

Simone Mosco Daniel Fusaro Wanmeng Li Alberto Pretto
University of Padova, Padova, Italy
{moscosimon, fusarodani, liwanmeng, alberto.pretto}@dei.unipd.it

A. Additional Implementation Details

Vision Embedding Module. The Vision Embedding module is responsible for extracting patches from the range image features produced by the convolutional stem. We use patch sizes of 7×7 with a stride of 4 in both the vertical and horizontal directions. Given an input range image of size 64×1024, as in the case of PandaSet [11] and SemanticKITTI [1], this produces 15 patches vertically and 255 horizontally, resulting in a total of $M = 3825$ tokens.

Backbone. The backbone of our method consists of $L = 8$ layers of the Retentive Network [9] block. Each layer processes an input of shape (M, C_h), where M is the number of tokens and $C_h = 128$ is the feature dimension. Each layer employs $h = 4$ heads, which in the retention case correspond to 4 different scales, modeled by the γ factors, where $\gamma = 1 - 2^{-5-\text{arange}(0,h)}$. The value matrix $\mathbf{V}$ has a hidden dimension of $C_v = 256$, which is twice the size of the query and key matrices dimensions ($C_h = 128$), as in the original RetNet architecture. Each Feed-Forward Network consists of two linear layers that expand features to a hidden dimension $C_f = 256$ before projecting them back to the original dimension $C_h = 128$, preserving input and output channel size across layers.

Semantic Head. The semantic head consists of two linear layers where the first maps the input features from 128 to 64, and the second projects to the number of output classes, depending on the target dataset.

B. Insights on Circular Retention

In this section, we provide a detailed explanation of how the retention mechanism originally proposed for NLP tasks [9] is adapted to the vision domain, specifically for the range-view representation of LiDAR data. The original retention mechanism is defined as:

$$\text{Retention}(\mathbf{X}) = (\mathbf{Q}\mathbf{K}^T \odot \mathbf{D})\mathbf{V} \tag{1}$$

where, the input sequence $\mathbf{X} \in \mathbb{R}^{N \times d}$ is projected into query, key and value matrices $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{N \times d_k}$ of a hidden dimension d_k, and $\mathbf{D}$ is a decay matrix that incorporates causal masking and exponential decay. This formulation is suitable for NLP tasks, where tokens represent a temporal sequence and the current token should not be influenced from future ones. In the vision domain, this causal constraint is not necessary. To address this, RMT [5] modifies the decay matrix to be bi-directional, as image patches do not have a temporal order but spatial relationships. To better capture these spatial dependencies, it proposes a decay matrix based on the Manhattan distance between tokens, which better capture the 2D structure of images. However, when applying this to LiDAR range images, further adaptation is needed. Range images are generated by projecting a 3D point cloud onto a 2D surface, effectively representing a 360° view of the surrounding environment. This results in a unique spatial structure where the left and right borders of the image are adjacent in the physical world. To model this continuity, we compute token distances using a circular Manhattan distance, capturing spatial relationships across the image boundaries. The impact of these adaptations is illustrated in Fig. 1, which shows the distance matrices used in the decay computation. Each heatmap highlights token distances. The original (causal) decay matrix discards future tokens, making it inefficient for vision tasks. The bi-directional variant improves coverage but fails to capture the circular structure inherent to range images. In contrast, our circular distance formulation produces a smoother and more coherent decay pattern that better reflects the 360° geometry of the range-view representation. This distinction is evident in the heatmaps. While the standard Manhattan distance used in [5] introduces discontinuities, resulting in repetitive sub-patterns, our circular approach preserves a continuous spatial flow across the entire map. These qualitative differences are also supported by the quantitative re-

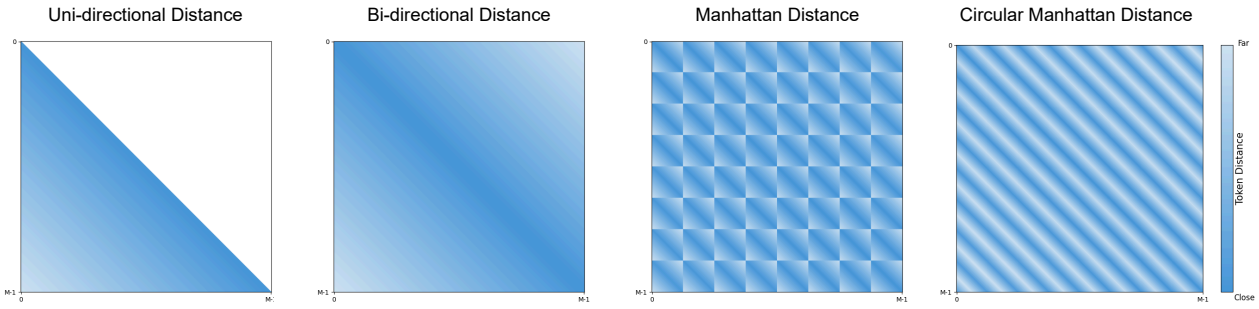

Figure 1. Visualization of heatmaps representing token distances used to construct the decay matrix in the retention mechanism, highlighting spatial relationships among tokens.

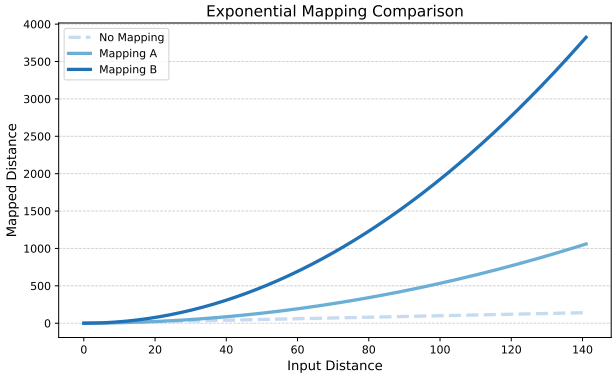

Figure 2. Exponential mapping comparisons using range images 64×1024 with patch size $(7, 7)$ and stride 4.

Method	Map. A	Map. B	mIoU (%)
Attention [10]			58.5
Retention [9]			57.7 (–0.8)
MaSa [5]			58.4 (–0.1)
MaSa [5]	✓		59.2 (+0.7)
CiR (ours)			58.7 (+0.2)
CiR (ours)	✓		59.9 (+1.4)
CiR (ours)		✓	**60.7** (+2.2)

Table 1. Ablation study on exponential mapping on SemanticKITTI validation set.

sults in Tab. 1, which highlights the superior performance of our circular decay matrix compared to previous approaches.

C. Exponential Mapping

Figure 2 illustrates different exponential mapping strategies designed to map circular Manhattan distances between patches and align them with the original range values used in [9]. Given a range image of size $H \times W$, patch size (P_H, P_W) and stride (s_H, s_W), we extract a number $M = (\lfloor \frac{H-P_H}{s_H} \rfloor + 1) \times (\lfloor \frac{W-P_W}{s_W} \rfloor + 1)$ of tokens. Following [9], the retention range is $[0, M-1]$, with $M-1$ the maximum token distance. However, computing circular Manhattan distances yields a much smaller range, up to $\lfloor \frac{w}{2} \rfloor + (h-1)$, where $w = (\lfloor \frac{W-P_W}{s_W} \rfloor + 1)$ and $h = (\lfloor \frac{H-P_H}{s_H} \rfloor + 1)$ are the number of tokens in the horizontal and vertical dimensions, respectively. This smaller scale reduces the effectiveness of the decay matrix, limiting the model ability to capture long-range dependencies. To address this, we re-scale the distances using two exponential mappings: *Mapping A*, which normalizes by the standard Manhattan maximum distance, *Mapping B*, which normalizes by the circular Manhattan maximum distance. As shown in Fig. 2, both methods expand the value distribution effectively, with Mapping B offering the widest spread. Employing this mapping is supported by the improved performance observed in Tab. 1, where also the method in [5] benefits from this remapping strategy.

D. Additional Quantitative Results

We provide additional quantitative results on the PandaSet [11] validation set in Tab. 2, using a range image of size 64×1024 for all methods. These further confirm the effectiveness of our approach, particularly in the circular retention mechanism in capturing fine-grained details within the range-view representation. Our method achieves high mIoU on challenging classes such as *bicycle* and *person*, as well as on difficult categories like *road barriers*. These performances are consistent with the test set results presented in the main paper, where our method outperforms other range-view-based approach, including both CNN and attention-based models.

E. Additional Ablation Study

Patch Extraction We provide a more detailed analysis of the Vision Embedding module, which is responsible for patch extraction, before the Retentive Network backbone.

Method	mIoU %	car	bicycle	motorcycle	truck	other-vehicle	person	road	road barriers	sidewalk	building	vegetation	terrain	background	traffic sign
RangeNet++ [7]	44.7	79.2	2.7	18.9	50.5	24.2	25.2	83.4	20.1	62.0	80.1	66.2	54.6	38.5	19.9
SqueezeSegV3 [12]	53.3	83.8	19.1	20.4	59.2	42.4	39.9	85.9	41.6	67.3	84.9	73.3	60.5	45.1	23.3
RangeFormer [6]	57.4	89.2	25.6	1.7	62.1	52.6	55.6	87.6	39.6	70.4	88.1	77.9	62.0	55.9	35.4
FIDNet [13]	58.4	89.8	18.6	22.1	62.2	55.8	58.5	89.0	35.4	71.2	89.3	79.8	59.5	56.9	28.7
SalsaNext [3]	61.0	89.4	27.8	**41.8**	64.4	53.4	59.4	88.1	43.4	71.5	88.2	78.3	60.8	55.4	32.6
CENet [2]	64.9	91.2	32.9	36.8	68.7	69.8	68.2	89.9	45.5	73.0	90.2	83.2	**67.1**	60.1	32.1
RangeRet (ours)	**66.7**	**91.5**	**50.9**	20.8	**69.9**	**72.3**	**70.4**	**90.3**	**56.9**	**73.1**	**90.9**	**84.1**	65.9	**61.4**	**35.5**

Table 2. Semantic segmentation results on PandaSet validation set for range-view models using inputs of size 64×1024.

Patch size	16×16	16×16	8×8	8×8	7×7	5×5	4×4
Stride	16×16	8×8	8×8	4×4	4×4	4×4	4×4
Tokens	256	889	1024	3825	3825	3825	4096
Params (M)	7.1	7.1	4.0	4.0	3.8	3.4	3.2
Train time	$\times 1$	$\times 1.1$	$\times 1.2$	$\times 3.3$	$\times 3.3$	$\times 3.3$	$\times 3.4$
mIoU (%)	56.6	58.0	58.9	59.6	**60.7**	60.5	60.1

Table 3. Ablation study on the patch size of the Vision Embedding module on SemanticKITTI validation set.

In Tab. 3, we assess how different patch sizes and strides impact model performance. As expected, smaller patches and strides produce more tokens, allowing the backbone to capture finer details and improving overall performance. However, a large number of tokens implies increased training time compared to the standard 16×16 patch dimension [4]. On the other hand, larger patches reduce the number of tokens but increase the number of parameters, making the Vision Embedding module a key component of the network. Our experiments show that a patch size 7×7 with stride 4 provides the best results, with a good balance between performance and efficiency.

k-NN Post-processing. Range-view-based approaches often suffer from information loss during the reprojection step, where per-pixel predictions are mapped back to the 3D domain. To mitigate this issue, we apply the k-NN post-processing technique proposed in [7], which refines predictions by exploiting information from neighboring points in 3D space. As reported in Tab. 4, we evaluate various kernel sizes on the SemanticPOSS [8] validation set, finding that a 7×7 kernel provides the best results. This step significantly alleviates the projection loss and improves overall results. Notably, larger kernels result in performance degradation.

Skip Connection We highlight the importance of the main skip connection introduced in our approach, which connects the output of the convolutional stem directly to the semantic head. This connection enhances the final representation by injecting low-level spatial details. Since the feature map produced by the backbone is upsampled to match the input range image size, it suffers from blurriness and loss of fine-grained details, particularly along object boundaries and for small objects. As demonstrated in the main paper, the skip connection not only improves quantitative results, boosting mIoU, but also yields clear qualitative benefits. As shown in Fig. 3, it enhances the sharpness and spatial layout of range-view features, effectively mitigating side effects introduced by the bilinear interpolation during upsampling.

Kernel	-	3	5	7	9	11
mIoU (%)	51.4	52.4	52.8	**52.9**	52.7	52.6

Table 4. Ablation study on kNN post-processing technique on SemantiPOSS validation set.

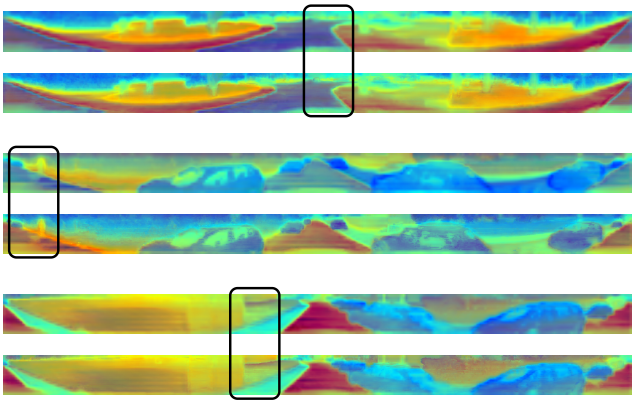

Figure 3. Visualization of the skip connection impact on range image features. Top shows features before, bottom after applying the skip connection. Results from SemanticKITTI [1] validation set using PCA features projection. Black boxes highlight improved border smoothness after the skip connection is applied.

F. Qualitative Results

Figures 4, 5 and 6 show visualization of segmentation results on PandaSet, SemanticPOSS, and SemanticKITTI, comparing ground truth point cloud labels with the predictions produced by our method. The samples are drawn from the validation split of each dataset. Our approach demonstrates strong segmentation performance across all three datasets, with only few misclassifications. Most errors occur in sparse regions far from the LiDAR sensor or in challenging areas such as borders between road and other ground types.

References

[1] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *ICCV*, pages 9297–9307, 2019. 1, 3

[2] Hui Xian Cheng, Xian Feng Han, and Guo Qiang Xiao. Cenet: Toward concise and efficient lidar semantic segmentation for autonomous driving. In *ICME*, pages 01–06, 2022. 3

[3] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In *International Symposium on Visual Computing (ISVC)*, pages 207–222, 2020. 3

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3

[5] Qihang Fan, Huaibo Huang, Mingrui Chen, Hongmin Liu, and Ran He. Rmt: Retentive networks meet vision transformers. In *CVPR*, pages 5641–5651, 2024. 1, 2

[6] Lingdong Kong, Youquan Liu, Runnan Chen, Yuexin Ma, Xinge Zhu, Yikang Li, Yuenan Hou, Yu Qiao, and Ziwei Liu. Rethinking range view representation for lidar segmentation. In *ICCV*, pages 228–240, 2023. 3

[7] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *IROS*, pages 4213–4220, 2019. 3

[8] Yancheng Pan, Biao Gao, Jilin Mei, Sibo Geng, Chengkun Li, and Huijing Zhao. Semanticposs: A point cloud dataset with large quantity of dynamic instances. In *IEEE intelligent vehicles symposium (IV)*, pages 687–693. IEEE, 2020. 3

[9] Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models. *arXiv*, 2023. 1, 2

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. 2

[11] Pengchuan Xiao, Zhenlei Shao, Steven Hao, Zishuo Zhang, Xiaolin Chai, Judy Jiao, Zesong Li, Jian Wu, Kai Sun, Kun Jiang, et al. Pandaset: Advanced sensor suite dataset for autonomous driving. In *ITSC*, pages 3095–3101, 2021. 1, 2

[12] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In *ECCV*, pages 1–19, 2020. 3

[13] Yiming Zhao, Lin Bai, and Xinming Huang. Fidnet: Lidar point cloud semantic segmentation with fully interpolation decoding. In *IROS*, pages 4453–4458, 2021. 3

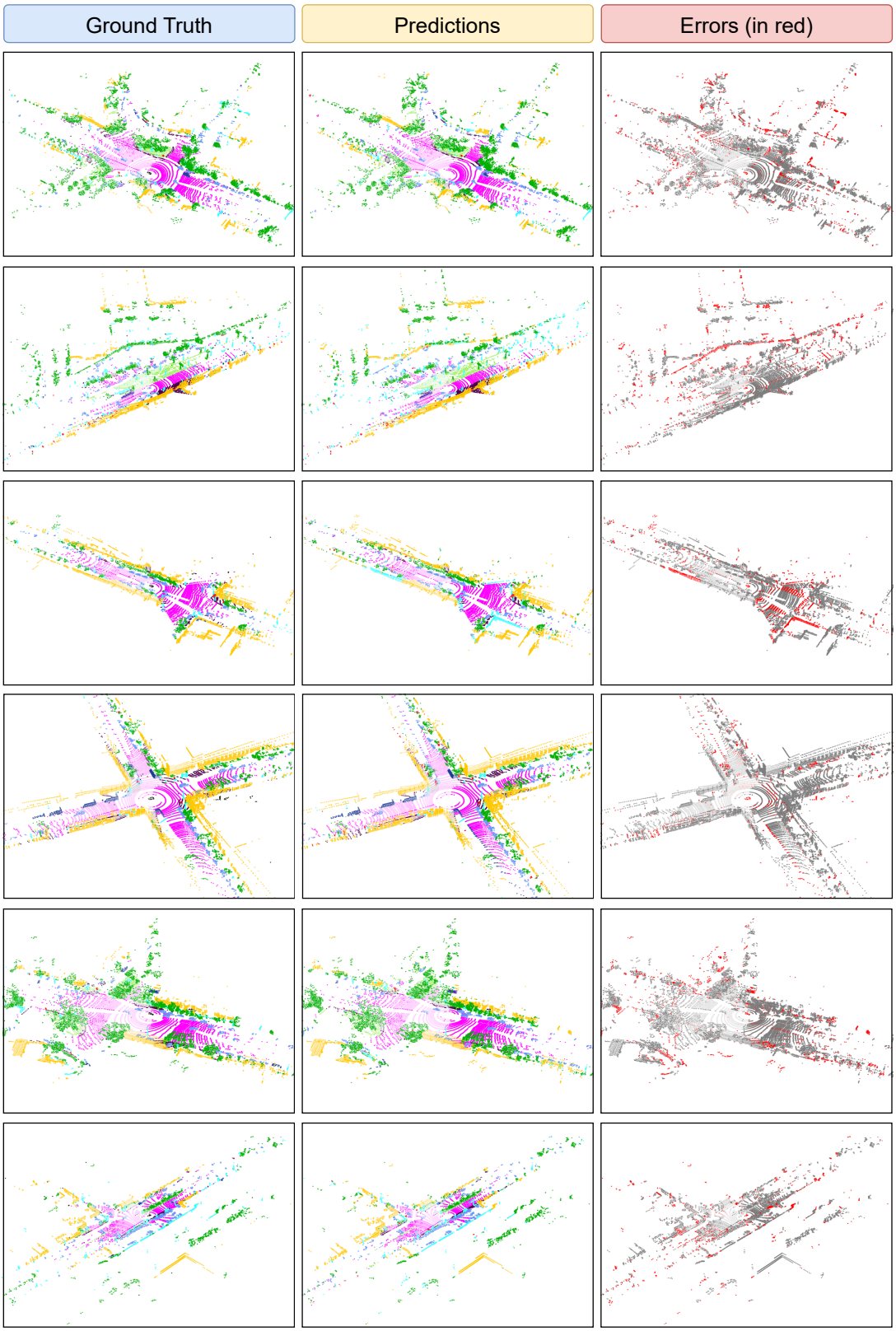

Figure 4. Qualitative results on PandaSet validation set.

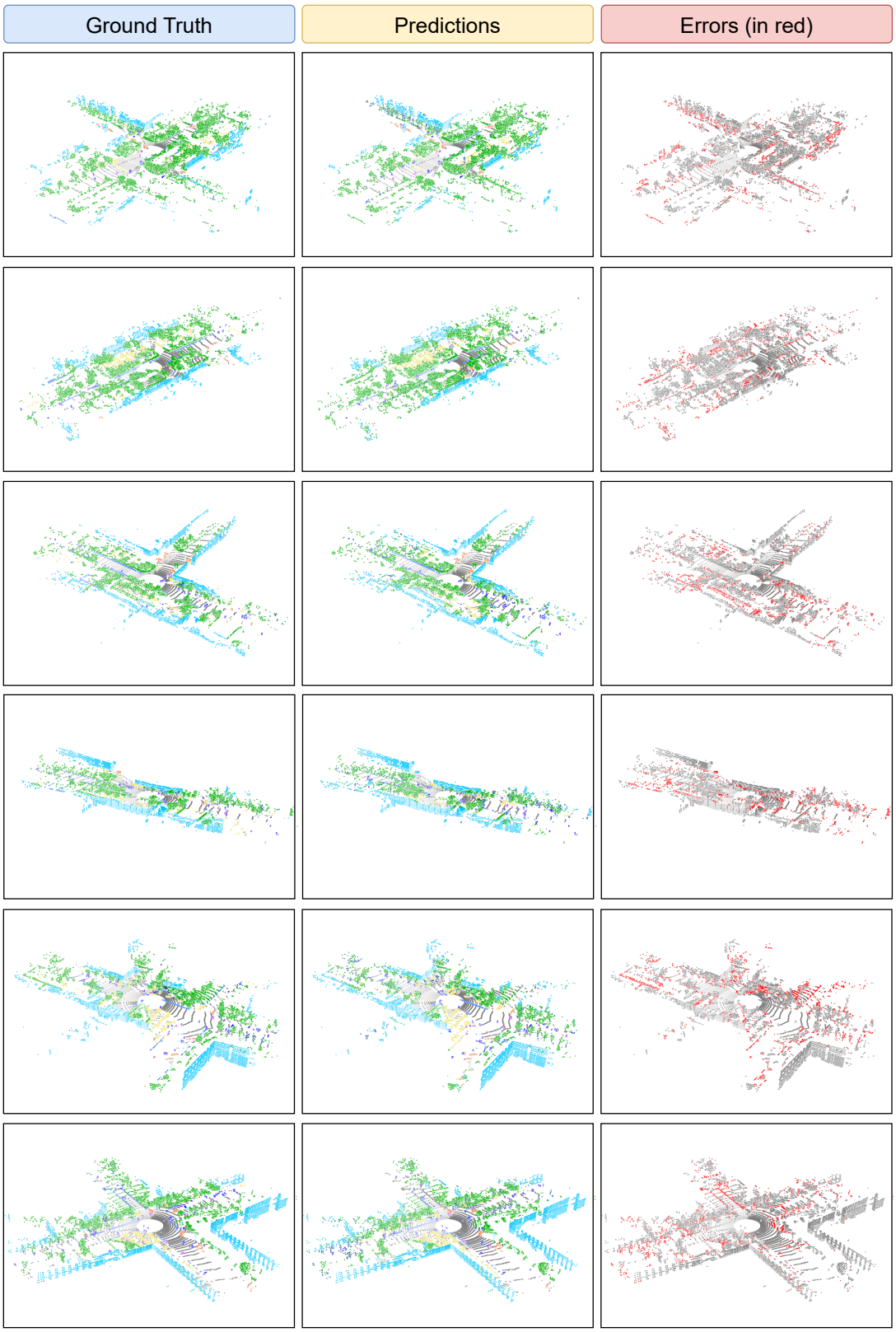

Figure 5. Qualitative results on SemanticPOSS validation set.

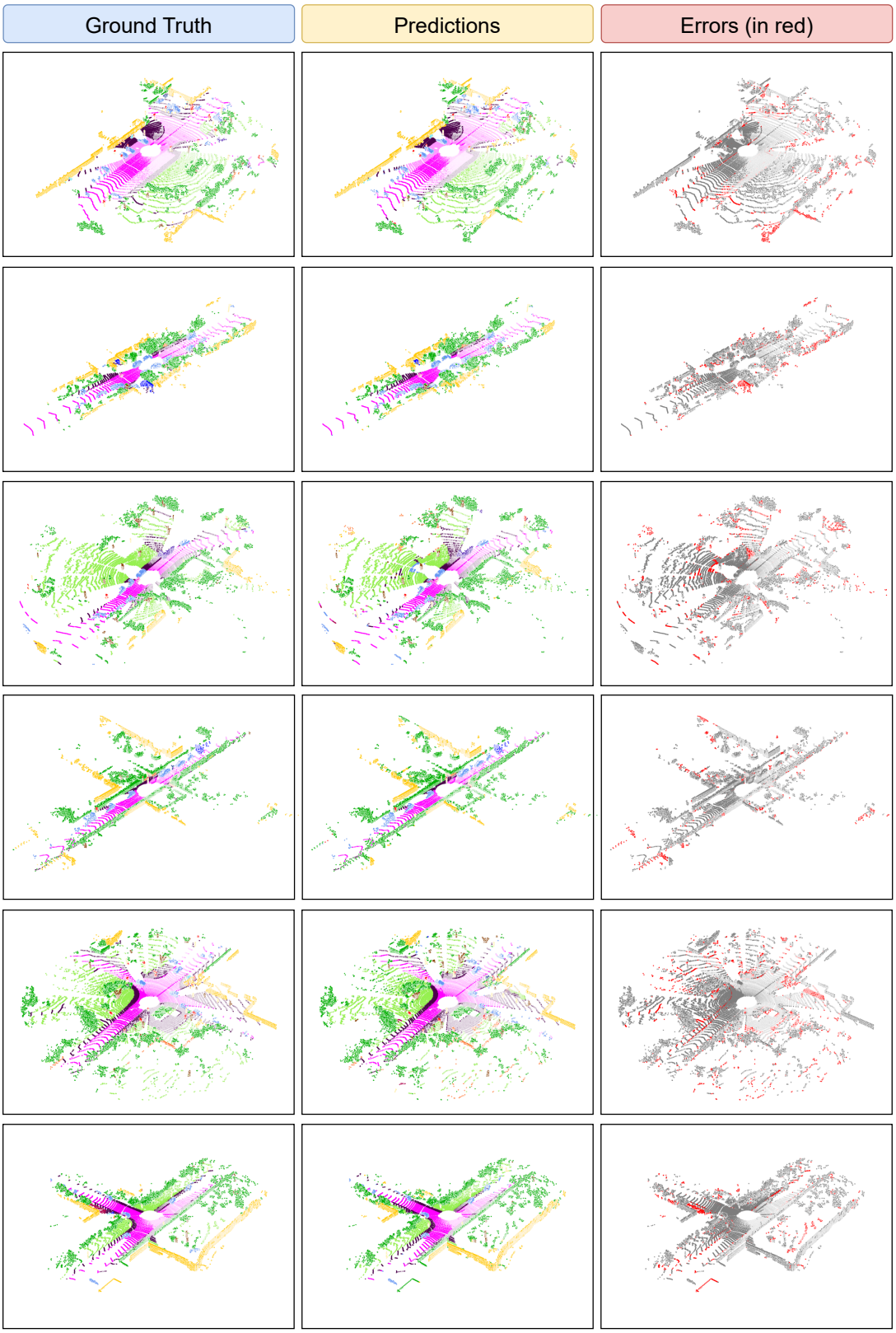

Figure 6. Qualitative results on SemanticKITTI validation set.