Less is More: Reducing Task and Model Complexity for 3D Point Cloud Semantic Segmentation - Supplementary

Li Li[1] Hubert P. H. Shum[1] Toby P. Breckon[1,2]
Department of {Computer Science[1] | Engineering[2]}, Durham University, UK
{li.li4, hubert.shum, toby.breckon}@durham.ac.uk

A. Performance with Different Backbones

In Tab. A1, alongside Cylinder3D [8], we also implement our architecture with popular backbone networks [2, 5] widely-used in 3D semantic segmentation.

B. Runtime Comparison

We conduct a runtime comparison between LiM3D and SOTAs on NVIDIA A100 GPU and Graphcore IPU-POD16 for Intelligence Processing Unit (IPU) acceleration. Summary inference times per item: LiM3D (ours, GPU) - 0.51s, Cylinder3D [8] (GPU) - 0.48s, Unal *et al.* [6] (GPU) - 0.53s, LiM3D+SDSC (ours, IPU) - 0.12s. SDSC could be slower on GPU due to its lower arithmetic intensity (ratio of compute to memory operation) [3, 7], but the high-bandwidth on-chip memory in IPU accelerator significantly improves the efficiency.

C. More Quantitative Results on Semi-supervised Segmentation

Besides {5%, 10%, 20%, 40%} labeled frames training, we also report our results with less than 5% label frames shown in Tab. A2. By applying our proposed architecture for semi-supervised and scribble-supervised 3D semantic segmentation, LiM3D and LiM3D+SDSC achieve higher than 80% relative performances (SS/FS) comparing with fully-supervised methods with less than only 1% labeled, *i.e.*, 191 frames (Tab. A2).

D. More Qualitative Results

Figs. A1 and A2 show a higher-resolution version of qualitative results that our method has superior performance. Fig. A3 compares {5%, 10%, 20%, 40%} sampling splits of SemanticKITTI [1] using LiM3D (ours). Note that using our semi-supervised methodology, the results training with very few ground-truth labels (*e.g.*, 5% and 10%) can achieve comparable performance to the training with a large number of labels (see Fig. A3, the 1-st and 2-nd rows *v.s.* 4-th row), with only subtle differences shown in the green and red circle. In Fig. A4, the magnification of regional details shows that our method can achieve better segmentation results than other methods, especially in the category of `vegetation`, `fence`, `sidewalk`, *etc.*

E. Parameters and Computation Costs Analysis on SDSC Sub-module

Given a Tensor $F \in \mathbb{R}^{H_F \times W_F \times L_F \times M}$, where H_F, W_F, L_F and M denote the radius, azimuth, height in the cylinder coordinate [8] and channels respectively. Applying convolution operation only for the active site of the sparse 3D point cloud, the computational cost (in FLOPs) of submanifold sparse convolution (SSC, [4]) is $a \times M \times N$ for the active site, where M is the number of input channels as defined previously, and N is the number of output channels. a is the number of active inputs to the spatial location defined in [4]. The computational cost for the inactive site is 0.

Since our SDSC sub-module consists of a sparse depthwise convolution (SDC) and a sparse pointwise convolution (SPC), the computational cost for SDSC is the sum cost of those two parts. SDC has a computational cost of

$$a \times M \times H_F \times W_F \times L_F. \tag{A1}$$

SPC computes a linear combination of the SDC output via a 1×1 convolution, which has the computational cost of

$$M \times N \times H_F \times W_F \times L_F. \tag{A2}$$

As a result, the computational cost of SDSC is the sum of Eqs. (A1) and (A2), *i.e.*,

$$a \times M \times H_F \times W_F \times L_F + M \times N \times H_F \times W_F \times L_F. \tag{A3}$$

The ratio of computational cost of SDSC to SSC [4] for active site, *i.e.*, $\text{cost}(SDSC) : \text{cost}(SSC)$, is:

$$\begin{aligned} &\frac{a \times M \times H_F \times W_F \times L_F + M \times N \times H_F \times W_F \times L_F}{a \times M \times N \times H_F \times W_F \times L_F} \\ =&\frac{1}{N} + \frac{1}{a} \approxeq \frac{1}{N} \quad (a \gg N). \end{aligned} \tag{A4}$$

Similar to the computational cost analysis, the parameters of SDSC is also the sum of SDC and SPC. The ratio of model

Table A1. 3D semantic segmentation results of LiM3D (ours) evaluated on SemanticKITTI [1] and ScribbleKITTI [6] *valid*-set with 10% labeled data, using different backbones. Alongside the per-class metrics, we show the relative performance of the semi-supervised approach against the fully supervised (SS/FS). S: with SDSC sub-module (✓) or without SDSC sub-module, *i.e.*, with normal sparse convolution.

Model	Dataset	S	mIoU	SS/FF	car	bicycle	motorcycle	truck	other vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other ground	building	fence	vegetation	trunk	terrain	pole	traffic sign
LiM3D (ours) + Cylinder3D [8]	Semantic		**62.2**	**89.5**	95.5	47.6	**65.2**	60.7	42.4	**75.2**	84.3	0.0	**94.2**	42.4	**80.7**	**5.4**	91.0	61.1	86.6	65.7	70.7	64.0	49.2
	Semantic	✓	61.0	87.8	95.3	43.9	59.2	46.2	47.0	71.4	79.8	1.6	93.8	44.0	80.0	4.4	90.8	60.4	87.7	64.5	73.5	63.8	51.4
	Scribble		61.0	87.8	95.0	34.5	52.9	61.5	41.3	71.0	**85.6**	0.0	93.7	44.4	79.9	0.4	90.1	58.1	87.9	61.5	74.8	65.3	44.3
	Scribble	✓	56.7	81.6	**95.6**	**48.9**	45.2	16.0	43.1	66.9	81.8	0.0	91.8	30.9	75.7	1.8	90.0	59.2	86.6	62.6	69.8	63.4	46.8
LiM3D (ours) + MinkowskiNet [2]	Semantic		60.4	86.9	94.6	44.3	47.1	70.2	29.5	68.7	80.8	0.0	93.6	38.4	79.6	0.1	90.2	58.7	88.2	66.1	75.6	**65.5**	**58.9**
	Semantic	✓	59.4	85.5	94.5	43.3	47.1	70.2	29.5	66.7	76.0	0.0	93.5	38.1	79.2	0.1	90.1	58.4	87.8	65.1	74.3	64.1	50.1
	Scribble		56.2	80.9	93.8	42.7	37.6	68.4	33.7	54.2	63.5	0.0	91.9	39.9	76.7	0.1	88.9	61.9	86.9	67.3	74.3	59.1	27.7
	Scribble	✓	50.3	72.4	92.8	08.8	30.3	68.2	36.3	27.2	56.6	**1.7**	88.0	47.9	71.8	2.7	85.3	63.9	86.7	71.2	74.7	56.6	13.6
LiM3D (ours) + SPVCNN [5]	Semantic		62.1	89.4	94.6	48.3	59.1	72.2	45.9	62.2	68.1	0.0	91.7	**59.2**	79.0	1.7	**91.6**	65.0	86.6	70.9	71.4	64.2	42.8
	Semantic	✓	60.8	87.5	92.7	29.6	64.8	72.9	**51.2**	54.0	47.1	0.0	89.0	58.2	75.2	1.9	88.8	**68.6**	**88.9**	**73.4**	**77.8**	64.4	30.8
	Scribble		59.6	85.8	92.5	29.4	63.7	72.8	50.8	52.2	43.2	0.0	89.8	56.3	75.1	0.6	88.6	66.7	87.6	71.0	74.3	62.3	29.6
	Scribble	✓	54.7	78.7	91.0	23.1	61.8	**73.1**	45.5	30.0	36.3	1.6	87.5	56.2	72.7	0.7	86.7	68.3	87.7	72.8	75.8	61.3	19.3

Table A2. 3D semantic segmentation results of LiM3D (ours) evaluated on SemanticKITTI [1] and ScribbleKITTI [6] *valid*-set with %1 and 2% labeled data. Alongside the per-class metrics, we show the relative performance of the semi-supervised approach against the fully supervised (SS/FS). S: with SDSC sub-module (✓) or without SDSC sub-module, *i.e.*, with normal sparse convolution.

% labeled	Dataset	S	mIoU	SS/FF	car	bicycle	motorcycle	truck	other vehicle	person	bicyclist	motorcyclist	road	parking	sidewalk	other ground	building	fence	vegetation	trunk	terrain	pole	traffic sign
LiM3D (ours) 2% / 383 frames	Semantic		**59.3**	**85.3**	**95.6**	**37.6**	50.1	54.4	46.0	**68.8**	77.1	0.0	**87.9**	32.8	**76.6**	2.0	**91.4**	**54.8**	**89.5**	**69.5**	**77.1**	**66.2**	**49.4**
	Semantic	✓	58.7	84.5	94.8	37.1	**55.0**	56.2	45.2	66.1	75.4	0.0	87.0	32.5	75.9	2.1	89.1	49.7	89.3	68.0	76.0	65.8	45.9
	Scribble		58.2	83.7	92.5	35.6	52.0	**57.1**	**49.4**	66.8	78.5	0.0	85.6	30.2	74.4	2.4	89.7	54.0	88.2	66.9	74.4	63.7	47.7
	Scribble	✓	56.8	81.7	93.1	34.9	47.0	50.9	43.8	64.1	75.6	0.0	85.2	29.5	73.9	2.0	88.8	49.5	88.2	66.9	74.8	64.4	47.4
LiM3D (ours) 1% / 191 frames	Semantic		58.4	84.0	92.6	37.5	51.2	50.4	47.9	68.6	**80.3**	0.0	86.3	**33.5**	74.7	**3.9**	89.4	51.4	88.3	67.4	75.1	64.8	45.8
	Semantic	✓	57.2	82.3	92.6	34.5	47.2	54.5	44.3	65.5	76.6	0.0	85.5	29.2	74.3	2.5	88.9	49.7	88.1	67.1	74.9	63.9	47.0
	Scribble		57.0	82.0	93.1	31.7	46.8	55.4	45.2	65.2	71.8	0.0	85.3	29.8	74.0	2.7	89.1	50.9	88.3	67.8	75.4	64.2	45.9
	Scribble	✓	55.8	80.3	92.7	27.6	43.6	50.6	42.3	60.6	73.9	0.0	85.3	29.1	74.2	2.6	87.3	49.7	87.2	68.5	70.2	64.5	43.6

parameters of SDSC to SSC [4] is:

$$\frac{\mathbf{D_K} \times M \times H_F \times W_F \times L_F + M \times N \times H_F \times W_F \times L_F}{\mathbf{D_K} \times M \times N \times H_F \times W_F \times L_F}$$
$$= \frac{1}{N} + \frac{1}{\mathbf{D_K}} \approx \frac{1}{\mathbf{D_K}} \quad (N \gg \mathbf{D_K}), \tag{A5}$$

where $\mathbf{D_K}$ is the dimension of convolution kernel K of size $D_{K,1} \times D_{K,2} \times D_{K,3}$, *i.e.*, $\mathbf{D_K} = D_{K,1} \times D_{K,2} \times D_{K,3}$.

LiM3D+SDSC uses SDSC sub-module as the basic building block for constructing other convolution-based modules in Cylinder3D (*e.g.*, residual block, upsample block, and downsample block). Take the residual block as an example, SDSC uses approximately 32x, 64x, $\cdots$, 512x less computation than SSC for active sites when $N = \{32, 64, \cdots, 512\}$ (Eq. (A4)). SDSC-based residual block with a kernel size of $1 \times 3 \times 1$ has 3x fewer parameters than the SSC-based residual block with the same kernel size, and 9x fewer parameters with $3 \times 1 \times 3$ kernel size (Eq. (A5)).

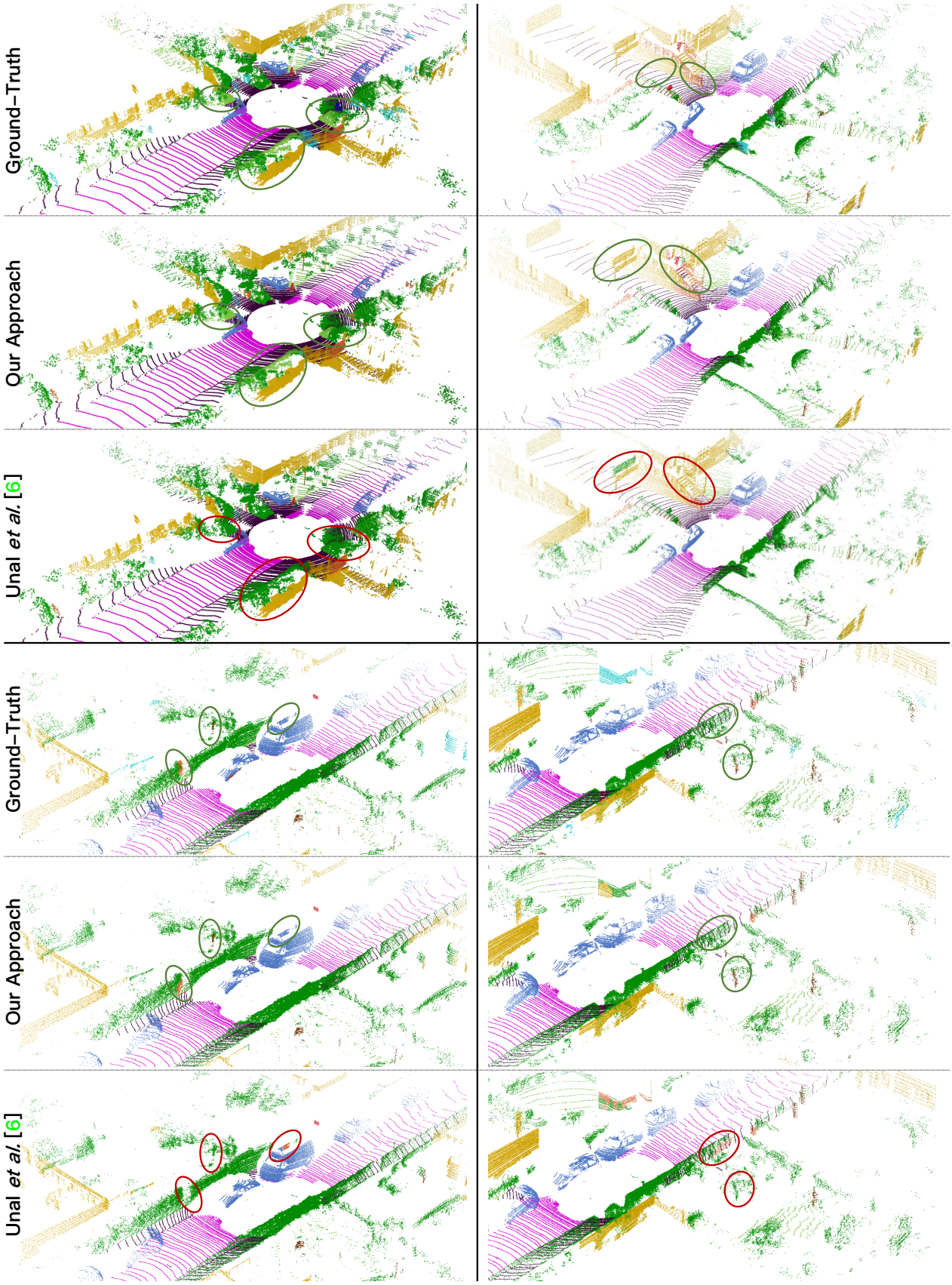

Figure A1. Comparing the 10% sampling split of SemanticKITTI [1] *validation* set with ground-truth (left), our approach (middle) and Unal *et al*. [6] (right) with areas of improvement highlighted in green, and areas of underperformance in red.

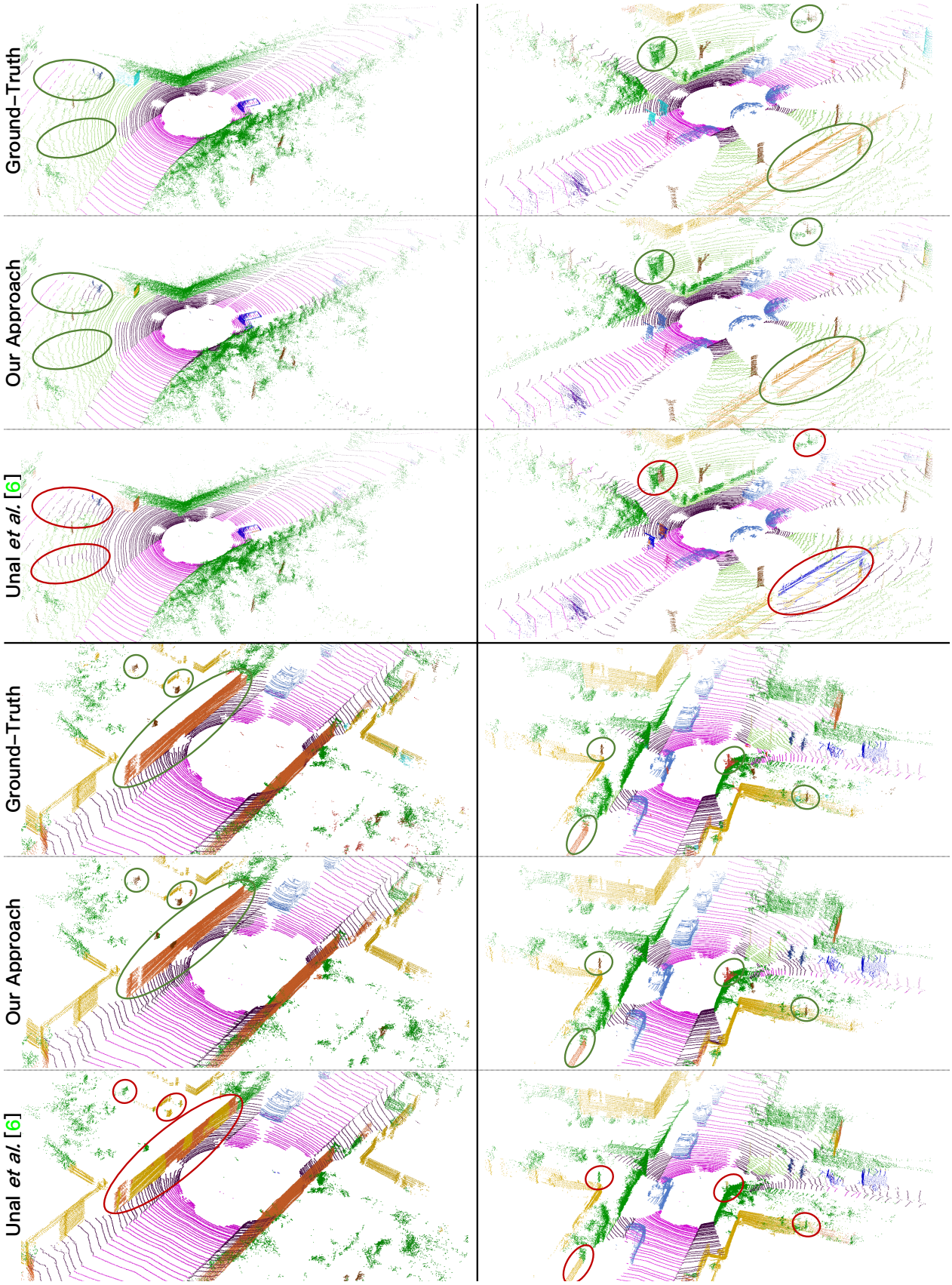

Figure A2. Comparing the 10% sampling split of SemanticKITTI [1] *validation* set with ground-truth (left), our approach (middle) and Unal *et al*. [6] (right) with areas of improvement highlighted in green, and areas of underperformance in red.

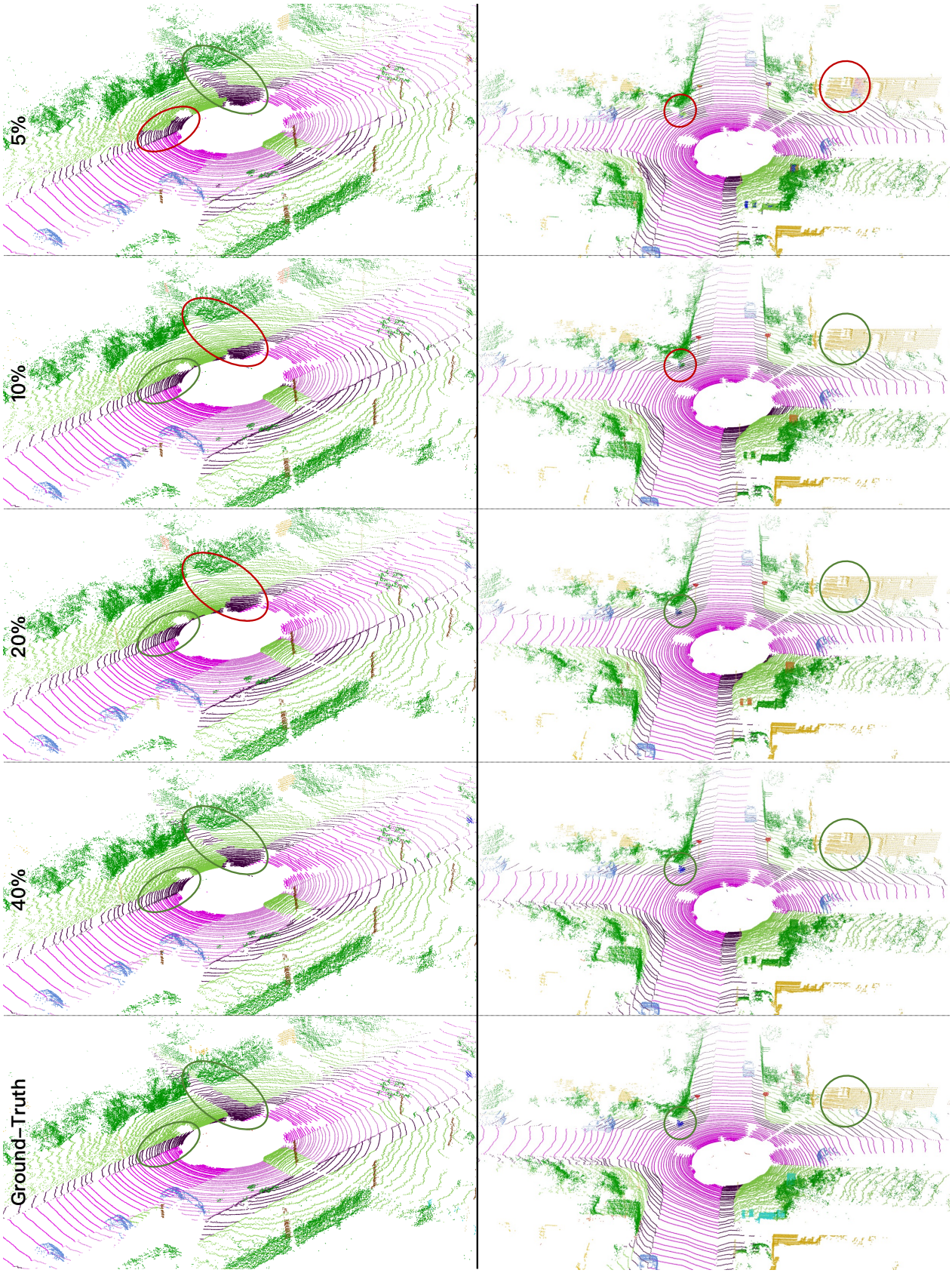

Figure A3. Comparing the 5%, 10%, 20%, 40% sampling split of SemanticKITTI [1] *validation* set with ground-truth (bottom) with areas of improvement highlighted in green, and areas of underperformance in red.

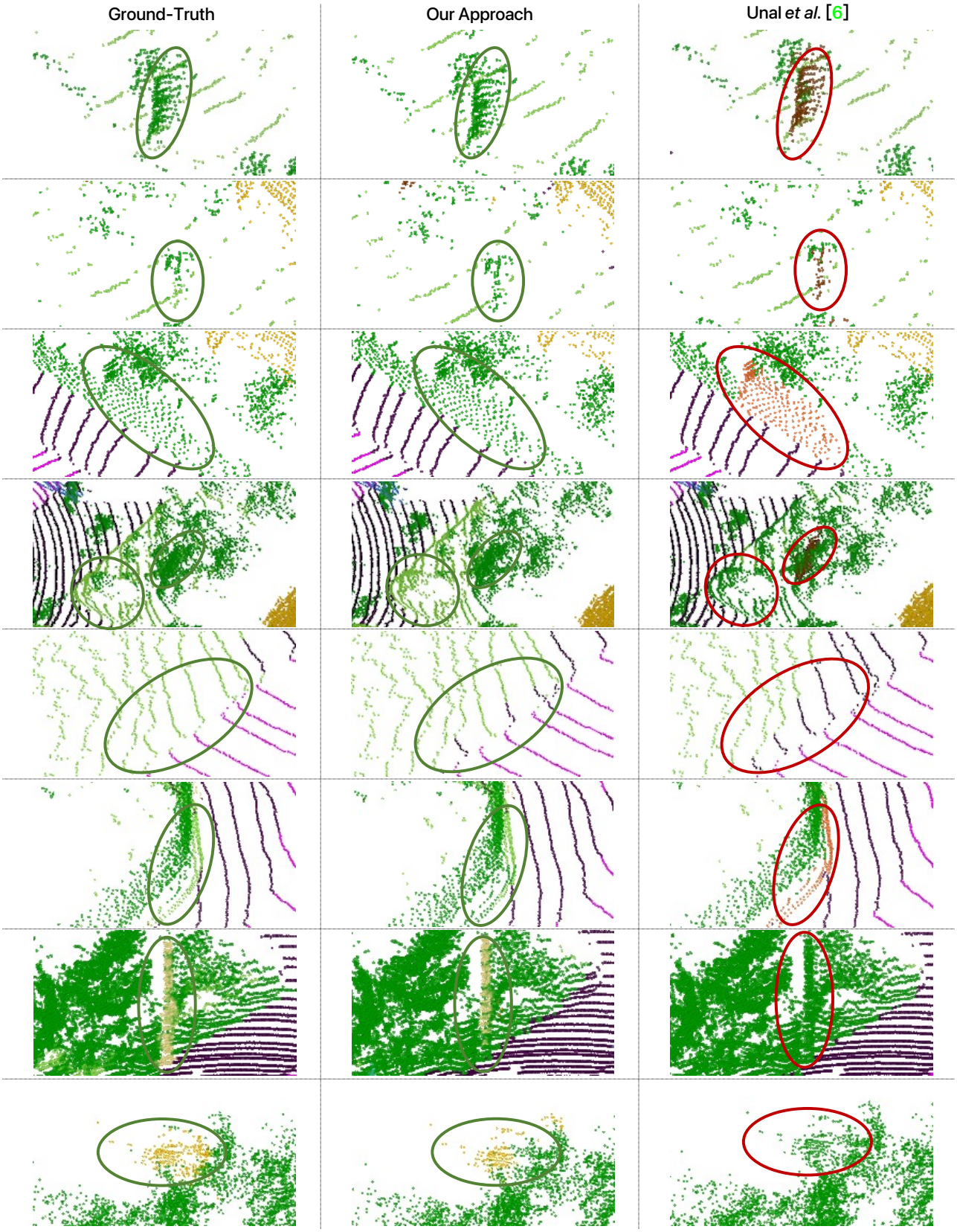

Figure A4. Magnification of regional details: comparing the 10% sampling split of SemanticKITTI [1] *validation* set with ground-truth (left), our approach (middle) and Unal *et al*. [6] (right) with areas of improvement highlighted in green, and areas of underperformance in red.

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 *Int. Conf. Comput. Vis.*, pages 9296–9306, Seoul, Korea (South), Oct. 2019. 1, 2, 3, 4, 5, 6

[2] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3070–3079, 2019. 1, 2

[3] A. Gholami, K. Kwon, B. Wu, Z. Tai, X. Yue, P. Jin, S. Zhao, and K. Keutzer. Squeezenext: Hardware-aware neural network design. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2018. 1

[4] Benjamin Graham, Martin Engelcke, and Laurens van der Maaten. 3D Semantic Segmentation with Submanifold Sparse Convolutional Networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018. 1, 2

[5] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution. In *Eur. Conf. Comput. Vis.*, 2020. 1, 2

[6] Ozan Unal, Dengxin Dai, and Luc Van Gool. Scribble-supervised LiDAR semantic segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022. 1, 2, 3, 4, 6

[7] S. Williams, A. Waterman, and D. A. Patterson. Roofline: an insightful visual performance model for multicore architectures. *Communications of the Acm*, 52(4):65–76, 2009. 1

[8] 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 *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021. 1, 2