Supplementary Material for Self-Distillation for Robust LiDAR Semantic Segmentation in Autonomous Driving

Jile Li[1], ✉Hang Dai[2], and ✉Yong Ding[1]

[1] Zhejiang University, Hangzhou, China
[2] MBZUAI, Abu Dhabi, United Arab Emirates
hang.dai@mbzuai.ac.ae dingy@vlsi.zju.edu.cn

1 Point Cloud Voxelization Setup

The voxelization step d is set as $(0.1, 0.1, 0.15)$ meters to voxelize the SemanticKITTI point cloud within the range of $[-75.2, +75.2]$, $[-75.2, +75.2]$, $[-4.0, +2.0]$ meters along the X, Y, Z axis. For nuScenes dataset, the range of point cloud is $[-51.2, +51.2]$, $[-51.2, +51.2]$, $[-5.0, +3.0]$ meters, and the step d is $(0.1, 0.1, 0.2)$ meters.

2 Model Training

We train our model with the Adam optimizer and one-cycle learning rate policy [3] with division factor 10, momentum ranges from 0.95 to 0.85, and weight decay 0.01. The maximum learning rate is set as 0.001 or 0.01 for SemanticKITTI or nuScenes. A batch of 8 or 24 random point cloud samples is trained on 2 Tesla V100 GPUs with 10 or 48 epochs for SemanticKITTI or nuScenes. During the training process, the four types of data augmentation transformations are used to avoid overfitting, which can be retrieved in Sec. 3.2 of our paper.

3 Network Architecture Details

Following the Shi *et al.* [2], we build the 3D sparse U-Net backbone by stacking four down-sampling blocks to make the voxel-wise features more informative with increasing receptive fields, and stack four up-sampling blocks for resolution restoration and feature refinement.

Fig.1 demonstrates the details of the backbone, which is implemented with the 3D sparse convolution (SparseConv3D), 3D inverse sparse convolution (InverseSparseConv3D) [4], and 3D submanifold convolution (SubMConv3D) [1]. We decompose a pair of down-sampling block D_i and up-sampling block U_i for a clear illustration.

In the down-sampling block D_i, a SparseConv3D block with a stride of S_i is followed by two stacked ResSubMBlock3D for down-sampling and residual feature learning. Note that the SparseConv3D can down-sample the voxel

Table 1. Robustness test on SemanticKITTI validation set for each components under more types of disturbances: 1) clean point cloud, 2) add point-wise random noise (jitter) uniformly distributed in [-0.05, +0.05] meters to all points, 3) drop some points with a random ratio sampled from [0.0, 0.5], 4) - 7) random rotation, random translation, random flipping, random scaling with the default parameters in our paper.

Disturbance	Clean	Jitter		Dropping		Rotation		Translation		Flipping		Scaling	
Performance	mIoU	mIoU	Delta	mIoU	Delta	mIoU	Delta	mIoU	Delta	mIoU	Delta	mIoU	Delta
Baseline*	64.90	58.48	-6.42	63.69	-1.21	65.01	+0.11	64.83	-0.07	64.88	-0.02	65.07	+0.17
+TransVFE	65.47	59.79	-5.68	65.13	-0.34	65.84	+0.37	65.56	+0.09	65.85	+0.38	65.71	+0.24
++Self-distillation	67.11	65.08	-2.03	66.97	-0.14	67.29	+0.18	67.02	-0.09	67.69	+0.58	67.42	+0.31
+++TTA	68.64	66.76	-1.88	68.49	-0.15	68.55	-0.09	68.46	-0.18	68.64	0	68.68	+0.04

features with a stride larger than 1, while the SubMConv3D only computes 3D submanifold convolution without any changes on the resolution for maintaining the submanifold of the voxels.

The up-sampling block U_i first fuses the output of the previous up-sampling block U_{i+1} and the output of the symmetrical down-sampling block D_i from the skip connection, then restores the resolution via an InverseSparseConv3D with the corresponding stride of S_i.

4 Robustness Analysis

We provide more experimental results on robustness analysis in Tab. 1. Compared with the average based VFE in baseline* model, our TransVFE guarantees less degradation under all disturbances. Rotation, translation, flipping and scaling have slight impacts on the model with TransVFE as: **i**) These four global transformations maintain the local structure on objects. But, point jitter and dropping destroy the local structures, which are severe disturbances that challenge and reflect the model robustness more. **ii**) TransVFE explicitly models and exploits the local relationship among the points in each voxel, achieving more robust point cloud encoding learning than other existing VFEs.

5 Latency Analysis on TTA

Although we employ our new TTA to enhance the teacher with the improved soft labels for better self-distillation, the new TTA can also be used to further improve the LiDAR segmentation performance after self-distillation in the inference stage. The details of the performance improvements and latency of our simple yet effective TTA are reported in Fig. 2, which reuses a proposed compound transform multiple times for input variants generation. The segmentation performance and latency increase as the number of input variants (M) increases,

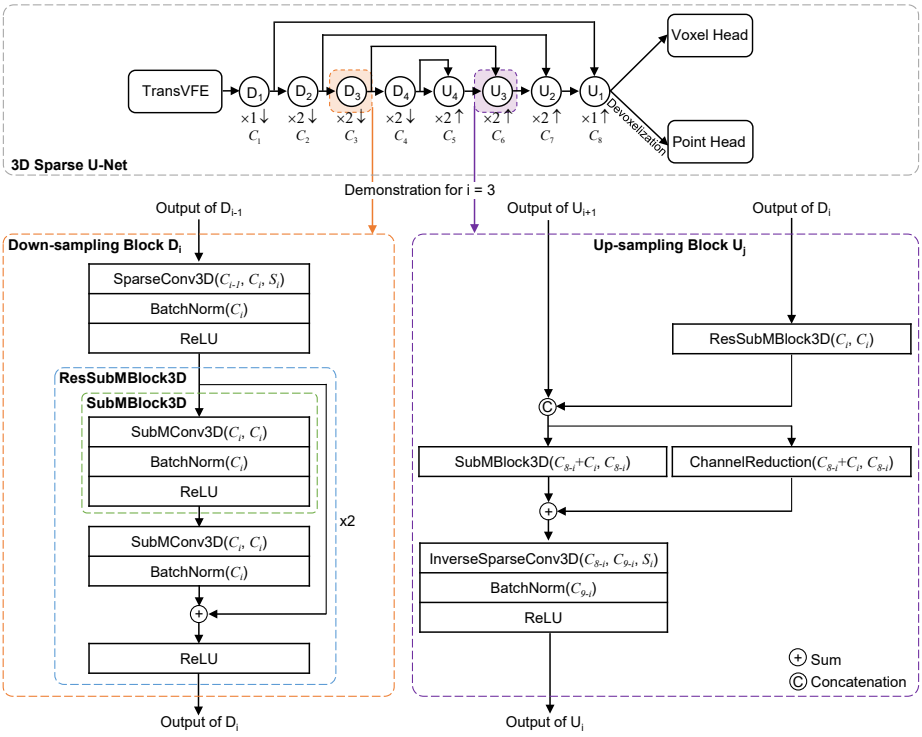

Fig. 1. Details of 3D sparse U-Net.

which is a trade-off between the performance and the inference time. Our new TTA can improve the performance on mIoU by ∼2.5% when $M \geq 6$. If the efficiency is required, our TTA with a small M of 2 can still achieve improvements on mIoU of ∼1.5% with 0.247 seconds inference time. Note that naively combining the four individual transformations in an input variant set as $\{X, X_{\text{scale}}, X_{\text{flip}}, X_{\text{rot}}, X_{\text{tran}}\}$ ($M = 5$) only yields mIoU of 66.64 with +1.17 improvements (see Tab. 6 of our paper), and there exists a mIoU gap of ∼0.25% behind our TTA with a set capacity of $M = 2$ with only one additional input variant generated by the proposed compound transform.

6 Latency Analysis on TransVFE

The TransVFE takes 0.03 seconds out of overall 0.148 seconds (evaluated without TTA) on SemanticKITTI, but yields higher segmentation performance than the commonly used VFE based on the average or PointNet.

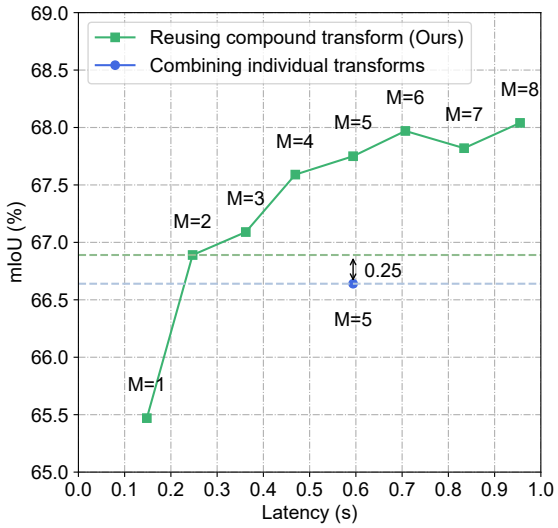

Fig. 2. Segmentation performance v.s. latency of different number of input variants (M) in our TTA on SemanticKITTI validation set.

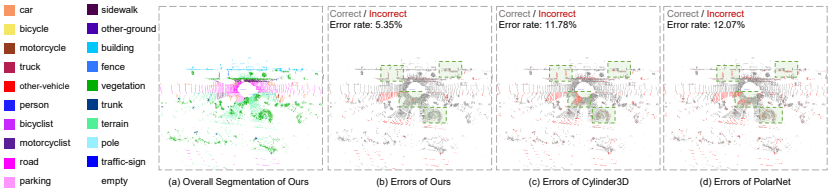

Fig. 3. Qualitative results on sequence 08 of SemanticKITTI. The point cloud is projected to the BEV for clear visualization. The error rate is defined as the number of incorrect points divided by the total number of points in a point cloud. Light green rectangles highlight some significant differences.

7 Qualitative Results

We compare the qualitative results with the representative 3D voxel method Cylinder3D [6] and 2D image method PolarNet [5] using their released models. As shown in Fig. 3, our method can perceive more object details with less incorrect segmentation in understanding complex scenes.

References

1. Graham, B., Engelcke, M., van der Maaten, L.: 3D semantic segmentation with submanifold sparse convolutional networks. In: CVPR. pp. 9224–9232 (2018)
2. Shi, S., Wang, Z., Shi, J., Wang, X., Li, H.: From points to parts: 3D object detection from point cloud with part-aware and part-aggregation network. IEEE TPAMI **43**(8), 2647–2664 (2021)

3. Smith, L.N., Topin, N.: Super-convergence: Very fast training of neural networks using large learning rates. In: Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications. vol. 11006, p. 1100612 (2019)
4. Yan, Y., Mao, Y., Li, B.: SECOND: sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
5. Zhang, Y., Zhou, Z., David, P., Yue, X., Xi, Z., Gong, B., Foroosh, H.: PolarNet: An improved grid representation for online LiDAR point clouds semantic segmentation. In: CVPR. pp. 9598–9607 (2020)
6. Zhu, X., Zhou, H., Wang, T., Hong, F., Ma, Y., Li, W., Li, H., Lin, D.: Cylindrical and asymmetrical 3D convolution networks for LiDAR segmentation. In: CVPR. pp. 9939–9948 (2021)