Supplementary Materials for "Embracing Single Stride 3D Object Detector with Sparse Transformer"

A. Submission on Test Server

Due to the submission frequency limit of the Waymo test server, we only report the results of our best model. We compare SST with the three most competitive methods and report their performances in the multi-frame setting from the official leaderboard. The results are shown in Table A and Table B. The performance of SST on vehicle class is comparable with these methods, and the performance of SST on pedestrian class significantly outperforms other methods.

Methods	LEVEL_1 3D AP/APH	LEVEL_2 3D AP/APH
PVRCNN_2f [10]	81.06/80.57	73.69/73.23
CenterPoint_2f [18]	81.05/80.59	73.42/72.99
RSN_3f [13]	80.70/80.30	71.90/71.60
SST_TS_3f (Ours)	80.99/80.62	73.08/72.74

Table A. Performance of **vehicle** detection on test split of Waymo Open Dataset.

Methods	LEVEL_1 3D AP/APH	LEVEL_2 3D AP/APH
PVRCNN_2f [10]	80.31/76.28	73.98/70.16
CenterPoint_2f [18]	80.47/77.28	74.56/71.52
RSN_3f [13]	78.90/75.60	70.70/67.80
SST_TS_3f (Ours)	83.05/79.38	76.65/73.14

Table B. Performance of **pedestrian** detection on test split of Waymo Open Dataset.

B. Discussion of Sparse Operations

Due to the space limit of the main paper, we leave the discussion on sparse operations in the supplementary materials. In this section, we discuss two problems for sparse operations: (1) *insufficient receptive field of submanifold sparse convolution (SSC)* [4], and (2) *the difficulties of downsampling/upsampling in sparse data*.

B.1. Insufficient Receptive Field of Submanifold Sparse Convolution (SSC)

In Sec. 1 and Table 7 in our main paper, we briefly point out that the SSC-based single-stride architecture faces a severe problem of the insufficient receptive field. We demonstrate this issue here in Fig. A by comparing the behaviors of SSC and standard 2D convolution in sparse data. Both the SSC and standard convolutions have two layers with a kernel size of three. However, the SSC could not reach the voxel on the top-left corner from the voxel marked with a star, while the standard convolution is capable of doing this. This example intuitively illustrates the insufficiency of receptive fields for SSC, and we explain the reasons in detail as follows.

The SSC do not "fill" empty voxels for the sake of efficiency, which largely constrains the information communication between voxels. Under such conditions, in Fig. A (a), only one voxel (the pink one) in has information communication with the one marked by a red star if the kernel size is 3×3. On the contrary, Fig. A (b) shows that the 2D convolution can gradually enlarge the receptive field by involving the empty voxels in the convolution process, which is more effective for aggregating information compared to the SSC.

To give an experimental illustration, we conduct experiments on the class of vehicles, which require sufficient receptive field for detection. In the Table 7 of the main paper, replacing the 3×3 standard convolutions with SSC will cause a significant drop of AP from 64.69 to 51.57. We further increase the receptive field by expanding the kernel size of SSC to 5×5 and 7×7. These improve the performance from the 3D AP 51.57 to 55.40 and 56.77, but there is still a large gap to the variant using standard convolutions. Therefore, these numbers support our analyses on the insufficient receptive fields of SSC.

B.2. Downsampling/Upsampling in Sparse Data

Although downsampling and upsampling are common in dense data, *e.g.*, pooling in CNN, token merge in Swin-Transformer, it is non-trivial to transfer these techniques to sparse data like point clouds. A variant of SSC named Sparse Convolution (SC) follows the standard convolu-

tion to implement the downsampling and upsampling in sparse data. With such implementation, data loses sparsity rapidly [17, 18] and this leads to high computational overhead.

In our sparse Transformer, downsampling/upsampling by token merge [7] also needs careful consideration. First, the downsampling operation is still an open problem for point clouds: what is the best way to merge the varied number of tokens scattered in different spatial locations? Second, the upsampling operation is also non-trivial and requires future research: how to recover a couple of tokens in different locations from a single token effectively and efficiently? In developing the SST, we encounter these challenges and find it difficult to offer satisfying solutions. Although we have bypassed these difficulties by adopting the single-stride architecture, we hope future research may work on this downsampling/upsampling question and better utilizes sparse data.

C. Potential Improvements

In order to rule out unimportant factors and present a clean architecture, we only inherit the basic framework of PointPillars [5]. So there is a large room for further performance improvements, and we list some of them as follows. We will adopt these techniques in our future work.

IoU Prediction. In detection, the classification score of a bounding box are not always consistent with the real regression quality. So many recent methods [3, 10, 11, 18] use another branch to predict the IoU between output bounding boxes and the corresponding ground-truth boxes, and use the predicted IoU to correct the classification scores.

More Powerful Second Stage. We use LiDAR-RCNN [6] as our second stage, which is a lightweight PointNet-like module only takes the raw point cloud as input. So it has no effect on our first stage and is convenient for our analysis of single-stride architecture. However, its performance is inferior to some other elaborately designed RCNNs, *e.g.*, CenterPoint [18], PartA2 [11], PVRCNN [10], PyramidRCNN [8], which reuse the features from the single stage to achieve better refinement. With the point-level features interpolated from feature maps in the first stage, SST can be equipped with most of these methods and aim for better abilities.

Incorporating Advanced Techniques in Vision Transformer. We have witnessed the fast progress of vision transformers. Many advanced techniques can be borrowed to enhance the performance of SST. **(1) Better efficiency**: There are a lot of techniques can be adopted to improve our efficiency, for example, token selection [9, 15], attention simplication [14]. **(2) Better efficacy**: Some techniques can be used to make SST more effective, *e.g.*, relative positional encoding [16], different attention mechanism [2].

D. Computational Complexity Compared with Convolutions

We investigate the computational complexity of the SST architecture and convolutional architectures. Our analyses demonstrate that SST has a unique advantage in efficiency by utilizing the sparsity of point clouds and the regional grouping.

Following the calculation in Swin-Transformer [7], we inspect the computational complexities of convolutional architectures and SST. For an input scene size of $h \times w$, a convolution layer with kernel size $k \times k$ and channel number C has the complexity as Equation 1. On the same scene, an SRA operation has the complexity as Equation 2, where it has H-heads, region size of $R \times R$, and the average sparsity as S, which is the ratio for non-empty voxels[1].

$$\Omega(\text{Conv}) = hwk^2C^2, \tag{1}$$

$$\Omega(\text{SRA}) = 4ShwC^2 + 2HS^2R^2hwC, \tag{2}$$

As shown in the equations, the computational complexities for convolutions and SRA operations are all $O(hw)$, thus are both linear to the scale of input. However, the SRA operations have the linear factor of S, which is generally small due to the sparsity of point clouds. According to our statistics, S roughly equals to 0.09 on Waymo Open Dataset with our voxelization. Such an analysis indicates that our SRA operations is efficient by properly exploiting the sparsity of LiDAR data.

E. Use of existing assets

Codebase We use MMDetection3D [1] for all of our experiments. MMDetection3D offers solid implementation of a wide variety of 3D detection algorithms. MMDetection3D is licensed under Apache License, Version 2.0.

Dataset We use Waymo Open Dataset [12] as the benchmark for our experiments. The Waymo Open Dataset is licensed under separate terms. (See https://waymo.com/open/terms/ for details.)

References

[1] MMDetection3D Contributors. MMDetection3D: OpenMMLab Next-generation Platform for General 3D Object Detection. https://github.com/open-mmlab/mmdetection3d, 2020. 2

[2] Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. XCiT: Cross-Covariance Image Transformers. *arXiv preprint arXiv:2106.09681*, 2021. 2

[1] Our calculation is approximate because we assume non-empty voxels uniformly scatter in the space.

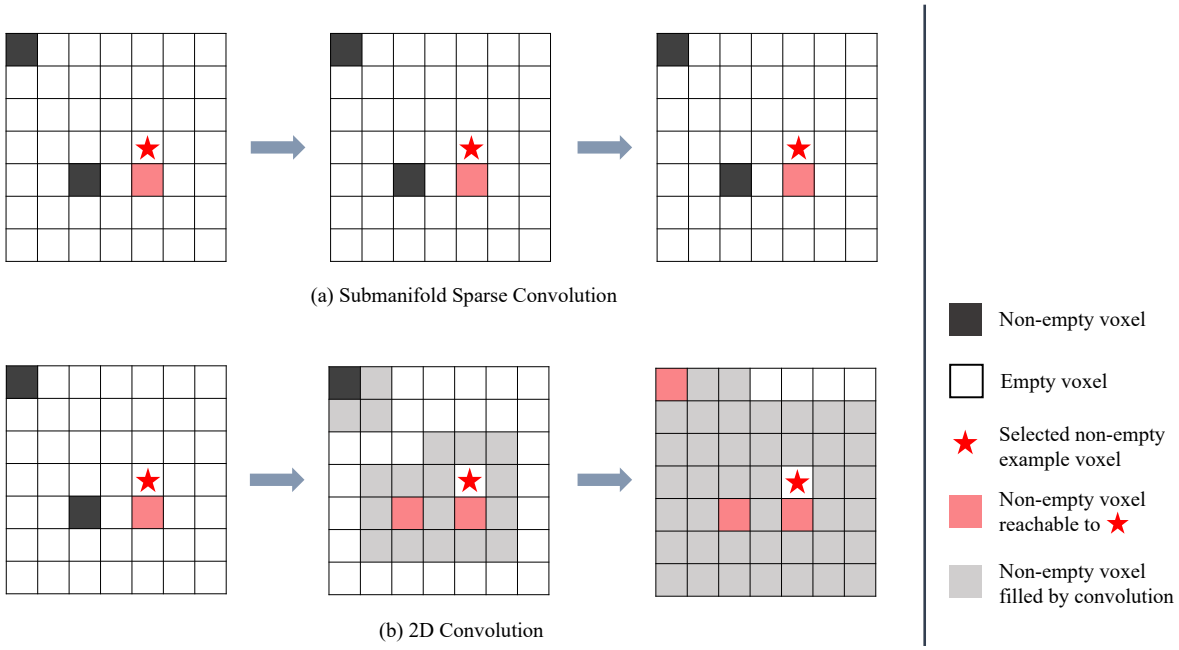

Figure A. Illustration of receptive enlarging in the 3×3 submanifold sparse convolution (SSC) and the standard 3×3 convolution. In SSC, only the information of the voxel (the pink one) covered by the kernel can reach to the red star. In 2D convolution, all non-empty voxels can reach to the red star after 2 convolution layers, because the empty locations are "filled" by the convolution.

[3] Lue Fan, Xuan Xiong, Feng Wang, Naiyan Wang, and ZhaoXiang Zhang. RangeDet: In Defense of Range View for LiDAR-Based 3D Object Detection. In *ICCV*, 2021. 2

[4] Benjamin Graham and Laurens van der Maaten. Submanifold Sparse Convolutional Networks. *arXiv preprint arXiv:1706.01307*, 2017. 1

[5] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast Encoders for Object Detection from Point Clouds. In *CVPR*, 2019. 2

[6] Zhichao Li, Feng Wang, and Naiyan Wang. LiDAR R-CNN: An Efficient and Universal 3D Object Detector. In *CVPR*, 2021. 2

[7] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In *ICCV*, 2021. 2

[8] Jiageng Mao, Minzhe Niu, Haoyue Bai, Xiaodan Liang, Hang Xu, and Chunjing Xu. Pyramid R-CNN: Towards Better Performance and Adaptability for 3D Object Detection. In *ICCV*, 2021. 2

[9] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification. *arXiv preprint arXiv:2106.02034*, 2021. 2

[10] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. In *CVPR*, 2020. 1, 2

[11] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From Points to Parts: 3D Object Detection from Point Cloud with Part-aware and Part-aggregation Network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[12] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *CVPR*, 2020. 2

[13] Pei Sun, Weiyue Wang, Yuning Chai, Gamaleldin Elsayed, Alex Bewley, Xiao Zhang, Cristian Sminchisescu, and Dragomir Anguelov. RSN: Range Sparse Net for Efficient, Accurate LiDAR 3D Object Detection. In *CVPR*, 2021. 1

[14] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with Linear Complexity. *arXiv preprint arXiv:2006.04768*, 2020. 2

[15] Tao Wang, Li Yuan, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. PnP-DETR: Towards Efficient Visual Analysis with Transformers. In *ICCV*, 2021. 2

[16] Kan Wu, Houwen Peng, Minghao Chen, Jianlong Fu, and Hongyang Chao. Rethinking and Improving Relative Position Encoding for Vision Transformer. In *ICCV*, 2021. 2

[17] Yan Yan, Yuxing Mao, and Bo Li. SECOND: Sparsely Embedded Convolutional Detection. *Sensors*, 18(10), 2018. 2

[18] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3D Object Detection and Tracking. *arXiv preprint arXiv:2006.11275*, 2020. 1, 2