In the supplementary material, we first elaborate on the proof of non-overlapped set partition in §A, and then provide more implementation details of the network architecture, training schemes, and ablation baselines in §B. Finally, we present more ablative studies for hyper-parameter analyses in §C and visualization of quantitative results in §D. We also discuss the difference of Axis-attention in §E and limitation of DSVT in §F.

A. Proof of Non-overlap Set Partition

Definition A.1 (***Dynamic set partition***) N is the number of non-empty voxels for a specific window, and τ is the maximum number of sparse voxels allocated to each local set. The required number of sub-sets in this window is computed as follows,

$$S = \lfloor \frac{N}{\tau} \rfloor + \mathbb{I}[(N \,\%\, \tau) > 0]. \tag{1}$$

For the j-th set (denoted as $\mathcal{Q}_j = \{q_k^j\}_{k=0}^{\tau-1}$), the index of voxel can be computed:

$$q_k^j = \left\lfloor \widetilde{q}_k^j \right\rfloor = \left\lfloor \frac{(j \times \tau + k)}{S \times \tau} \times N \right\rfloor, \quad \text{for} \;\; k = 0, ..., \tau - 1. \tag{2}$$

The following theorems show that our algorithm formulation in the main paper satisfies all the necessary concepts of the proposed dynamic set partition. The case of $N = S \times \tau$ is trivial. Now suppose the case of $(S-1) \times \tau < N < S \times \tau$.

Theorem A.2. (***Non-overlap***): *For any two local sets,* $0 \leq i, j \leq S - 1$, *then* $\mathcal{Q}_i \cap \mathcal{Q}_j = \emptyset$.

Proof. Obviously, this theorem can be converted to verify the non-overlap of two neighboring sets. Specifically, for any $1 \leq j \leq S-1$, $\mathcal{Q}_{j-1} \cap \mathcal{Q}_j = \emptyset$. We formulate q_0^j (the first voxel index in Q_j) as follows,

$$q_0^j = \left\lfloor \frac{j \times N}{S} \right\rfloor = \left\lfloor N_0^j + \frac{k_0^j}{S} \right\rfloor = N_0^j, \;\; 0 \leq k_0^j \leq S - 1. \tag{3}$$

Then we can compute the last voxel index in Q_{j-1}:

$$\begin{aligned} q_{\tau-1}^{j-1} &= \left\lfloor \frac{(j-1) \times \tau + \tau - 1}{S \times \tau} \times N \right\rfloor \\ &= \left\lfloor \frac{j \times N}{S} - \frac{N}{S \times \tau} \right\rfloor = \left\lfloor N_0^j + \frac{k_0^j}{S} - \frac{N}{S \times \tau} \right\rfloor. \end{aligned} \tag{4}$$

Note that $(S-1) \times \tau < N < S \times \tau$, thus $\frac{S-1}{S} < \frac{N}{S \times \tau} < 1$ and $0 \leq \frac{k_0^j}{S} \leq \frac{S-1}{S}$. So we have

$$\left\lfloor N_0^j - 1 \right\rfloor < \left\lfloor N_0^j + \frac{k_0^j}{S} - \frac{N}{S \times \tau} \right\rfloor < \left\lfloor N_0^j \right\rfloor. \tag{5}$$

To this end, we can prove that $q_{\tau-1}^{j-1} = N_0^j - 1$. For any $1 \leq j \leq S-1$, $q_{\tau-1}^{j-1} \neq q_0^j$, thus $\mathcal{Q}_{j-1} \cap \mathcal{Q}_j = \emptyset$.

Theorem A.3. (***Completeness***): $\mathcal{Q}_0 \cup \mathcal{Q}_1 \cup ... \cup \mathcal{Q}_{S-1} = U$, *where* $U = \{0, 1, ..., N-1\}$.

Proof. If $u \in \{0, 1, ..., N-1\}$ but $u \notin U$, there must exist two continuous indexes q_{k-1}^j and q_k^j satisfying $q_{k-1}^j < u$ and $q_k^j \geq u + 1$. Thus we have:

$$\frac{(j \times \tau + k)}{S \times \tau} \times N - \frac{(j \times \tau + k - 1)}{S \times \tau} \times N > 1. \tag{6}$$

This yields a contradiction, because $N < S \times \tau$, and concludes the proof.

Theorem A.4. (***Equivalent***): *For any j-th subset,* $0 \leq j \leq S-1$, *we have* $\lfloor \frac{N}{S} \rfloor \leq |\mathcal{Q}_j| \leq \lfloor \frac{N}{S} \rfloor + 1$. $|\mathcal{Q}_j|$ *denotes the number of valid and unique voxels belonging to j-th set, which needs to be distinguished from* τ.

Proof. We reformulate N as:

$$N = \left\lfloor \frac{N}{S} \right\rfloor \times S + res = l \times S + res, \tag{7}$$

where $l = \lfloor \frac{N}{S} \rfloor$ and $0 \leq res \leq S - 1$. In the case of $res = 0$, we have:

$$q_0^0 = 0, \; q_0^1 = l, \; ..., \; q_0^{S-1} = (S-1) \times l. \tag{8}$$

Following the proof of Theorem A.2, we can have:

$$q_{\tau-1}^0 = l - 1, \; q_{\tau-1}^1 = 2 \times l - 1, \; ..., \; q_{\tau-1}^{S-1} \quad = S \times l - 1, \tag{9}$$

which indicates for any j, we obtain $|\mathcal{Q}_j| = l = \lfloor \frac{N}{S} \rfloor$. In the case of $res \neq 0$, we compute the difference between $\widetilde{q}_0^j$ and $\widetilde{q}_0^{j+1}$:

$$\begin{aligned} \widetilde{q}_0^{j+1} - \widetilde{q}_0^j &= \frac{(j+1) \times \tau}{S \times \tau} \times N - \frac{j \times \tau}{S \times \tau} \times N \\ &= \frac{N}{S} = l + \frac{res}{S}. \end{aligned} \tag{10}$$

By Eq. (3) and Eq. (10), we have

$$\begin{aligned} q_0^{j+1} &= \left\lfloor \frac{(j+1) \times N}{S} \right\rfloor = \left\lfloor \frac{j \times N}{S} + \frac{N}{S} \right\rfloor \\ &= \left\lfloor N_0^j + \frac{k_0^j}{S} + l + \frac{res}{S} \right\rfloor = N_0^j + l + \left\lfloor \frac{res + k_0^j}{S} \right\rfloor. \end{aligned} \tag{11}$$

Note that $0 \leq k_0^j \leq S - 1$ and $0 < res \leq S - 1$, therefore we can easily obtain:

$$N_0^j + l \leq q_0^{j+1} \leq N_0^j + l + 1. \tag{12}$$

Finally, following the proof of Theorem A.2, we have:

$$N_0^j + l - 1 \leq q_{\tau-1}^j \leq N_0^j + l, \tag{13}$$

which implies for any j, we have $l \leq |\mathcal{Q}_j| \leq l + 1$ and concludes the proof.

B. Implementation Details

In this section, we provide more implementation details about network architecture (§B.1), ablation baselines (§B.2) and training schemes (§B.3).

B.1. Network Architecture.

B.1.1 3D Perception on Waymo

As mentioned in the main paper, our detection approach follows the framework of CenterPoint-Pillar [13] and only appends our DSVT before BEV backbone while other components remained unchanged.

DSVT-P is a single-stride pillar-based sparse backbone, which adopts the pillar size of (0.32m, 0.32m, 6m) with four DSVT blocks. Each block is equipped with two DSVT layers with different set partitioning configurations, (*i.e*, X-Axis Partition and Y-Axis Partition). The DSVT layers contains a rotated set based Multi-Head Self-Attention (MHSA) module, followed by a 2-layer MLP with GELU nonlinearity in between. A LayerNorm (LN) layer is applied after each MHSA module and each MLP, and a residual connection is applied after each module. All the attention modules are equipped with 8 heads, 192 input channels, and 384 hidden channels. The hybrid window sizes are set to (12, 12, 1) and (24, 24, 1) by default, and the maximum number of voxels belonging to each set (τ) is 36, as introduced in main paper.

DSVT-V is a voxel-based variant of our proposed backbone, which follows the pillar-based framework and splits along the Z-Axis. The input voxel size is (0.32m, 0.32m, 0.1875m). Moreover, its backbone also has four stages with block numbers $\{1, 1, 1, 1\}$ and the number of voxels along the Z-Axis is reduced by our attention-style 3D pooling module with the stride $\{4, 4, 2\}$. The window sizes along the Z-Axis are $\{32, 8, 2, 1\}$, which covers all of the Z-Axis. Different from DSVT-P, to adapt more voxels, τ is set to 48.

B.1.2 3D Perception on nuScenes

3D object detection and BEV Map Segmentation both utilizes DSVT-P in nuScenes benchmark. We set window size and set the maximum number of tokens assigned to each set (τ) to (30, 30, 1) and 90, respectively. The attention modules in use were equipped with 8 heads, 128 input channels, and 256 hidden channels.

B.2. Ablation Baselines.

B.2.1 ResBackbone1x

ResBackbone1x is built upon sparse convolution (Spconv 2.0) [3], a widely used auto-differentiation library for sparse tensors. This baseline adopts the same network designs (*i.e.*, depth, width, and kernel size) as VoxelResBackBone8x implemented by OpenPCDet [11] except for replacing all the downsampling SparseConv blocks with conventional SubMConv to hold the single stride architecture. The input voxel size is set to (0.32m, 0.32m, 6m), which is the same as our DSVT pillar version. For a fair comparison, this variant only substitutes the DSVT sparse backbone with ResBackbone1x while other settings remained unchanged, (*e.g.*, detection head, loss functions, and post-processing). As shown in Table 1, thanks to the single stride design, this baseline is very strong, which is +4.89 better than the original CenterPoint-Voxel(8x) [13] and +2.80 higher than its residual modification version on L2 mAPH. Even on such a strong baseline, our DSVT performs +1.78 better, which demonstrates its powerful modeling ability.

Backbone	#param.	LEVEL_2 (3D)	
		mAP	mAPH
VoxelBackBone8x†	58M	64.51	61.92
VoxelResBackBone8x†	80M	66.47	64.01
ResBackbone1x	88M	69.61	66.81
DSVT(Pillar, dim128)	71M	**71.14**	**68.59**

Table 1. Comparison with sparse convolution. † denotes the results implemented by OpenPCDet [11]. All models are trained on 20% Waymo data with 30 epochs.

B.2.2 3D Pooling

Linear. As for a specific downsampling sparse region, we first convert it into dense and flatten it to a vector with fixed length. Then a one-layer MLP is applied to project it. Finally, a layer normalization is adopted after MLP module.

Max Pooling. Similar to the linear variant, after being converted into a dense format, the native max-pooling operation is applied on voxel dimension for processing downsampling.

Attention+Mask. This variant follows the same design as our attention-style 3D pooling module except for adding key padding masks of the empty space in the pooling region.

B.3. Training and Inference Schemes.

B.3.1 Waymo

One Stage Detection. As mentioned in the main paper, we follow the same training schemes as [13] to optimize the model using Adam [7] optimizer with weight decay 0.05, one-cycle learning rate policy [5], and max learning rate 3e-3. All the models are trained with batch size 24 for 24 epochs on 8 NVIDIA A100 GPUs. During inference, following [6, 10], we use class-specific NMS with the IoU threshold of 0.7, 0.6 and 0.55 for vehicle, pedestrian and cyclist, respectively. Besides, we also use the ground-truth copy-paste data augmentation during training and disable this data augmentation in the last one epoch following [4] (*e.g.*, using the fade strategy).

DSVT-P	DSVT-V	Size	LEVEL_2 (3D)	
			mAP	mAPH
✓		24	70.71	68.14
✓		36	**71.14**	**68.59**
✓		48	70.95	68.43
	✓	36	71.65	69.31
	✓	48	**72.01**	**69.67**
	✓	60	71.90	69.60

Table 2. Effect of set size.

Two Stage Detection. The two-stage of our DSVT is built upon CT3D [9] and trained separately. We fix the 1^{st}-stage model and finetune the 2^{nd}-stage refinement module for 12 epochs with the same training schedule.

B.3.2 NuScenes

3D Object Detection. We follow the same training scheme adopted in Transfusion-L [1]. All the models are trained by AdamW optimizer with weight decay 0.05, one-cycle learning rate policy, max learning rate 5e-3, and batch size 32 for 20 epochs.We adopt the same fade strategy in [1] in last 5 epochs.

BEV Map Segmentation. We also adopt the same training strategy as BEVFusion [8], including training epoch, learning rate and hyper-parameter of optimizer.

C. Hyper-parameter Analyses

Our DSVT also works well in a wide range of hyper-parameters, such as the set size and network depth. All the experiments are trained on 20% Waymo training data with 30 epochs.

Set Size. Table 2 shows the performance of our approach with different set sizes. With the increase of the set size (from 24 to 36 in DSVT-P, 36 to 48 in DSVT-V), the performance gradually improves. However, a very large set size will also slightly decrease the mAP/mAPH. We argue that our regional local set attention can better encode the part-aware geometric information, which enhances the performance of tiny objects. The large set coverage may involve lots of noise points. Thus, we set the set sizes to 36 and 48 for our DSVT-P and DSVT-V respectively.

Network Depth. DSVT is relatively shallow by design thanks to the large receptive fields of the transformer architecture. As shown in Table 3, we provide the results with a greater number of DSVT blocks for investigating the influence of network depth. We observe that the performance is gradually saturated with the increase of block number. A deeper network will decrease the running speed. Considering the trade-off between the computation cost and performance improvement, we choose 4 blocks as the default setting.

# of Blocks	LEVEL_2 (3D)	
	mAP	mAPH
2	70.66	68.10
4	71.14	68.59
6	71.12	68.56
8	**71.24**	**68.68**

Table 3. Effect of network depth.

D. Qualitative Results

We visualize the qualitative results on Waymo Validation Set in Figure 1. Thanks to the large receptive field of Transformer and fine-grained geometric information provided by the attention-style 3D pooling module, our DSVT performs well on the large scenes and can locate 3D objects with sparse points accurately.

E. Compared to Axial-attention

Our method cannot be considered as an extension of Axial-attention [12]. DSVT is specifically designed for efficiently processing sparse data in parallel with dynamically assigned and size-equivalent local sets. The axis-based rotated partitioning is a replaceable strategy for intra-window fusion (please see Table 6 in the main paper for alternative strategies). In contrast, axial-attention [2, 12] aims to reduce the attention computation cost due to the dense data (*e.g.*, 2D image, or video) and enlarge receptive field by axis-based factorization.

F. Limitation

Although our method achieves promising performance and running speed on Waymo Open dataset, there are still some limitations. DSVT mainly focuses on point cloud processing for 3D object detection in outdoor autonomous driving scenarios, where objects (*i.e.*, car, pedestrian and cyclist) are distributed on a 2D ground plane. It is still an open problem to design more general-purposed backbones for the 3D community.

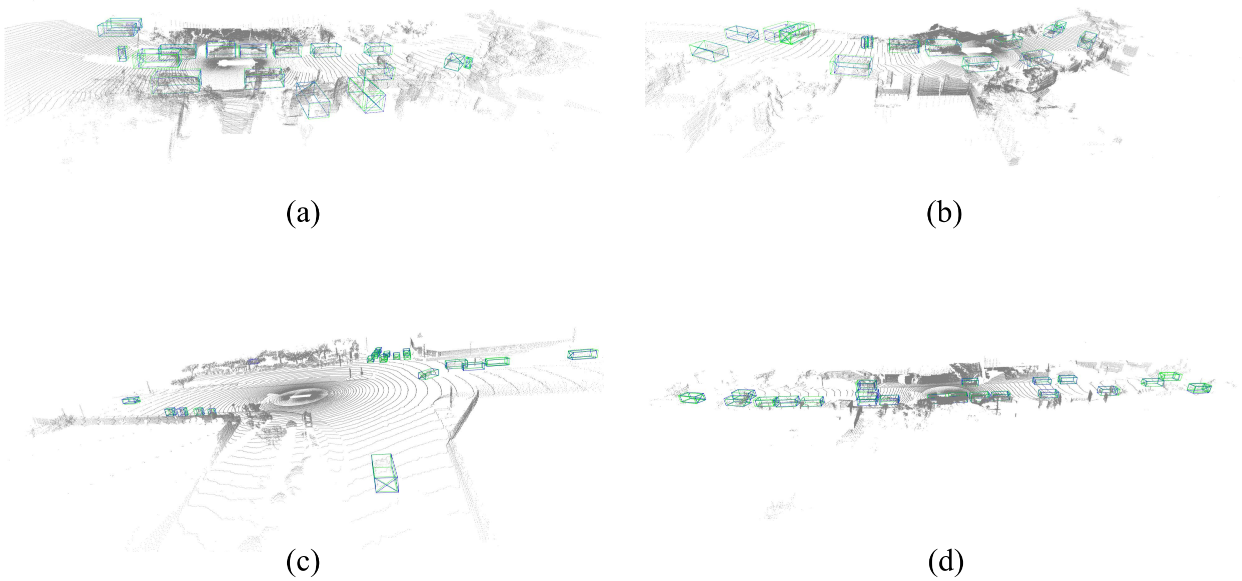
References

[1] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *CVPR*, 2022. 3

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021. 3

[3] Spconv Contributors. Spconv: Spatially sparse convolution library. `https://github.com/traveller59/spconv`, 2022. 2

[4] Lue Fan, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Fully sparse 3d object detection. In *NeurIPS*, 2022. 2



(a)

(b)

(c)

(d)

Figure 1. Qualitative visualization on Waymo validation set. Blue boxes and green boxes are ground-truth and predictions, respectively.

[5] Sylvain Gugger. The 1cycle policy. https://sgugger.github.io/the-1cycle-policy.html, 2018. 2

[6] Yihan Hu, Zhuangzhuang Ding, Runzhou Ge, Wenxin Shao, Li Huang, Kun Li, and Qiang Liu. Afdetv2: Rethinking the necessity of the second stage for object detection from point clouds. *NCAI*, 2022. 2

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[8] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *ICRA*, 2023. 3

[9] Hualian Sheng, Sijia Cai, Yuan Liu, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, and Min-Jian Zhao. Improving 3d object detection with channel-wise transformer. In *ICCV*, 2021. 3

[10] Guangsheng Shi, Ruifeng Li, and Chao Ma. Pillarnet: Real-time and high-performance pillar-based 3d object detection. In *ECCV*, 2022. 2

[11] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. 2

[12] Huiyu Wang, Yukun Zhu, Bradley Green, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In *ECCV*, 2020. 3

[13] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *CVPR*, 2021. 2