Focal Sparse Convolutional Networks for 3D Object Detection
Supplementary Material

A. More Implementation Details

Our implementation is based on the open-sourced OpenPCDet [2, 8], and the released code of CenterPoint [12].

A.1. Voxelization

KITTI. The 3D object detectors in this work convert point clouds into voxels as input data. On the KITTI [4] dataset, the range of point clouds is clipped into [0, 70.4m] for *X* axis, [-40m,40m] for *Y* axis, and [-3, 1]m for *Z* axis. The voxelization size for input is (0.05m, 0.05m, 0.1m).

nuScenes. On the nuScenes [1], the detection range is set to [-54m, 54m] for both *X* and *Y* axes, and [-5m, 3m] for the *Z* axis. The voxel size is set as (0.075m, 0.075m, 0.2m).

A.2. Data Augmentations

KITTI. On the KITTI [4] dataset, data transformation and augmentations include random flipping, global scaling, global rotation, and ground-truth (GT) sampling [11]. The random flipping is conducted along the *X* axis. The global scaling factor is sampled from 0.95 to 1.05. The global rotation is conducted around the *Z* axis. The rotation angle is sampled from -45° and 45°. The ground-truth sampling is to copy-paste some new objects from other scenes to the current training data, which enriches objects in the environments. For the multi-modal setting, we do not transform images with the corresponding operations, except ground-truth sampling. We copy-paste the corresponding image crops from other scenes onto the current training images.

nuScenes. On the nuScenes [1] dataset, data augmentations includes random flipping, global scaling, global rotation, GT sampling [11], and an additional translation. The random flipping is conducted along both *X* and *Y* axes. The rotation angle is also randomly sampled in [-45°, 45°]. The global scaling factor is sampled in [0.9, 1.1]. The translation noise is conducted on all three axes, *X*, *Y*, and *Z*, with a factor independently sampled from 0 to 0.5. We also conduct the corresponding point-image GT sampling on the nuScenes. GT sampling is disabled in the last 4 epochs [10].

Table S - 1. Comparisons to Voxel R-CNN in R40 on KITTI *val*.

Method	AP_{BEV}			AP_{3D}		
	Easy	Mod.	Hard	Easy	Mod.	Hard
V. [2]	95.52	91.25	88.99	92.38	85.29	82.86
Ours	95.45	**91.51**	91.21	92.86	**85.85**	85.29

Table S - 2. Objective loss weight in AP_{3D} (R40) on KITTI *val*.

Method	*Weight*	*Car*			*Ped.*	*Cyc.*
		Easy	Mod.	Hard	Mod.	Mod.
Baseline	–	92.10	84.36	82.48	54.49	70.38
Focals Conv	0.1	91.59	84.63	82.42	60.62	71.33
	0.5	92.10	85.16	83.12	**63.74**	69.56
	1.0	92.32	**85.19**	82.62	61.61	**72.76**
	2.0	91.73	84.61	82.38	54.09	71.34

Table S - 3. Improvements upon CenterPoint on Waymo $\frac{1}{5}$.

Method	AP LEVEL 1			AP LEVEL 2		
	Veh.	Ped.	Cyc.	Veh.	Ped.	Cyc.
Baseline	70.9	71.5	69.1	62.8	63.5	66.5
Focals Conv	72.2	72.6	71.1	64.1	64.6	68.5

A.3. Training Settings

KITTI. For model training on the KITTI dataset, *i.e.*, PV-RCNN [8] and Voxel R-CNN [2], we train the network for 80 epochs with the batch size 16. We adopt the Adam [6] optimizer. The learning rate is set as 0.01 and decreases in the cosine annealing strategy. The weight decay is set as 0.01. The momentum is set as 0.9. The gradient norms of training parameters are clipped by 10.

nuScenes. For models trained on the nuScenes datasets, *i.e.*, CenterPoint [12], we also train the network for 20 epochs with batch size 32. They are also trained with the Adam [6] optimizer. The learning rate is initialized as 1e-3 and decreases in the cosine annealing strategy to 1e-4. The weight decay is set as 0.01. The gradient norms of training parameters are clipped by 35.

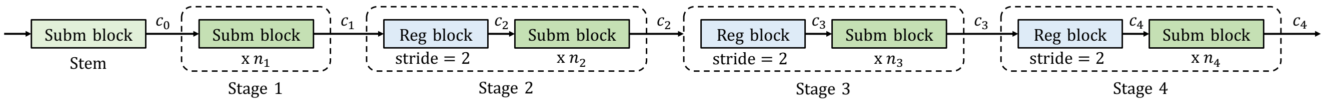

Figure S - 1. The illustration of the VoxelNet [13] backbone networks of PV-RCNN [8], Voxel R-CNN [2], and CenterPoint [12]. $\{c_0, c_1, c_2, c_3, c_4\}$ represent the output channels of the Stem, Stage 1, 2, 3, and 4. $\{n_1, n_2, n_3, n_4\}$ means the numbers of repeated submanifold blocks in these stages. In our approach, for the LIDAR-only setting, focal sparse convolutions (Focals Conv) are used in the last layer of Stage 1, 2, and 3. For the multi-modal setting, the focal sparse convolution with fusion (Focals Conv - F) is used in the last layer of Stage 1.

Table S - 4. Improvements over CenterPoint on the nuScenes *val* split.

Method	mAP	NDS	Car	Truck	Bus	Trailer	C.V.	Ped	Mot	Byc	T.C.	Bar
CenterPoint [12]	61.1	68.3	86.5	60.4	72.5	40.4	19.8	86.0	61.1	45.2	70.8	68.9
Focals Conv	62.3	69.0	86.5	61.2	**73.5**	40.3	21.3	86.5	63.3	48.0	72.0	**70.6**
Focals Conv - F	**67.1**	**71.5**	**87.7**	**62.9**	72.4	**42.6**	**28.1**	**87.8**	**74.4**	**65.5**	**78.9**	70.1

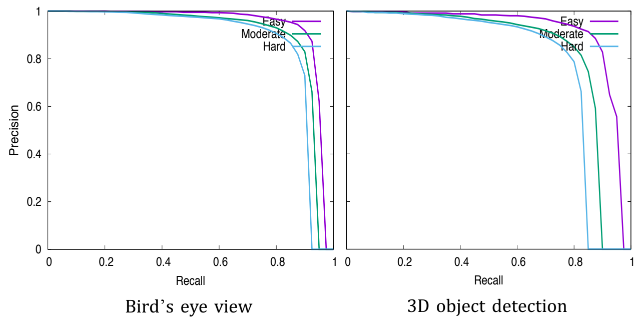

Figure S - 2. PR curves of Focals Conv - F on KITTI *test*.

B. Backbone Networks

We illustrate the structure of the backbone networks in Fig. S - 1. In this illustration, *Reg block* and *Subm block* mean the regular sparse convolutional block and the sub-manifold sparse convolutional block, respectively. The backbone networks are based on VoxelNet [13]. It contains a stem layer and 4 stages. In the last three stages, Stage 1, 2, and 3, there a regular sparse convolutional block with stride as 2 for down-sampling. There are some detailed differences among different frameworks, as the following.

B.1. Architecture settings

PV-RCNN and Voxel R-CNN. In the backbones of PV-RCNN [8] and Voxel R-CNN [2], the channels for the stem and stages, $\{c_0, c_1, c_2, c_3, c_4\}$, are $\{16, 16, 32, 64, 64\}$. The numbers of *Subm blocks* in these stages, $\{n_1, n_2, n_3, n_4\}$, are $\{1, 2, 2, 2\}$. A *Reg* or *Subm block* is a conv-bn-relu layer, which includes a regular or submanifold convolution, a batch normalization layer [5], and a ReLU activation.

CenterPoint. In the backbone network of the CenterPoint [12] detector, the backbone network is larger. The channels for the stem and stages, $\{c_0, c_1, c_2, c_3, c_4\}$, equal to $\{16, 16, 32, 64, 128\}$. The numbers of repeated *Subm blocks* in these stages, $\{n_1, n_2, n_3, n_4\}$, are $\{2, 2, 2, 2\}$. Compared to that in the PV-RCNN [8] and Voxel R-CNN [2] detectors, the *Subm block* is more complicated in this backbone network. Except the stem, it contains two sequential conv-bn-relu layers, with a residual connection.

B.2. Focal Sparse Convolution Usage

In our approach, the above architecture-level settings are directly inherited from the original PV-RCNN [8], Voxel R-CNN [2], and CenterPoint [3] frameworks, without any adjustment, for a fair comparison. In the LIDAR-only task, we apply the *Focals Conv* in the last layer of Stage 1, 2, and 3. In the multi-modal task, we apply the *Focals Conv - F* only at the last layer of Stage 1. This relieves the efficiency and memory issues caused by the RGB feature extraction. Note that, in the CenterPoint [12] detectors, it is also used in the last *layer*, not the total *block*. In other words, although there are two conv-bn-relu layers in each Subm block in CenterPoint [12], we only apply it as the last layer. For simplification, we do not double it as a block.

C. Additional Experiments

C.1. Results on Bird's Eye View on KITTI

We report the accuracy for 3D object detection and Bird's Eye View (BEV) of Focals Conv - F upon Voxel R-CNN [2] on the KITTI [4] dataset in Tab. S - 1. The results are calculated by recall 40 positions with the IoU threshold of 0.7. It performs better than the strong Voxel R-CNN [2] baseline on both AP_{BEV} and AP_{3D} in moderate and hard cases. We also provide the Prevision-Recall (PR) curves of Focals Conv - F on KITTI *test* split in Fig. S - 2.

Table S - 5. Ablations on ground-truth sampling on nuScenes $\frac{1}{4}$. GT Sampl. - Ground-truth Sampling. Trans. - Transformations.

Fusion	GT sampl.	Trans.	mAP	Car	Truck	Bus	Trail.	C.V.	Ped	Mot	Byc	T.C.	Bar
✗	✓	✗	39.3	**70.8**	31.0	**49.0**	20.3	3.5	**73.7**	28.8	13.4	49.3	**50.6**
✓			**43.3**	**70.8**	**32.5**	47.9	**21.0**	**6.7**	72.6	**44.9**	**31.8**	**58.4**	49.1
✗	✗	✓	54.6	80.4	51.9	60.6	31.5	14.2	81.4	57.4	45.4	63.1	60.7
✓			**59.0**	**83.2**	**56.1**	**61.5**	**36.6**	**19.7**	**84.3**	**59.1**	**49.4**	**73.8**	**66.3**

Table S - 6. Ablations on voxel size upon Focals Conv - F on the nuScenes *val* split.

Voxel size (m)	mAP	NDS	Car	Truck	Bus	Trailer	C.V.	Ped	Mot	Byc	T.C.	Bar
(0.05, 0.05, 0.2)	65.6	70.2	85.9	61.6	70.1	35.9	25.0	**88.3**	74.0	**66.1**	**79.4**	**70.1**
(0.075, 0.075, 0.2)	**67.1**	**71.5**	**87.7**	**62.9**	72.4	42.6	28.1	87.8	74.4	65.5	78.9	**70.1**
(0.1, 0.1, 0.2)	66.5	71.4	87.5	62.1	71.8	44.6	27.9	86.9	74.3	63.6	77.4	69.0
(0.125, 0.125, 0.2)	66.6	70.9	87.0	61.1	**74.0**	44.1	**30.2**	85.6	**75.1**	63.8	75.1	69.6
(0.15, 0.15, 0.2)	65.3	70.2	86.9	60.9	72.8	**45.1**	30.0	83.3	70.4	63.5	72.7	67.9

C.2. Objective Loss Weight

The training of the focal sparse convolutional networks involves the objective loss function. We implement it as a focal loss [7] as in Eq (1).

$$L_{obj} = \frac{1}{|N|}\sum_{i\in N} -(1-\bar{p_i})^{\gamma}\log(\bar{p_i}). \tag{1}$$

where $i \in N$ enumerates all sparse features in the current feature space. Following the original focal loss [7], we directly set $\gamma = 2$ and it works well. For notational convenience, we define $\bar{p_i}$ as follow

$$\bar{p_i} = \begin{cases} p_i, & \text{if} \quad y_i = 1, \\ 1 - p_i, & \text{otherwise}, \end{cases} \tag{2}$$

where $p_i \in [0, 1]$ is the estimated probability for the class with label $y_i = 1$. It is the estimation that whether the feature i contributes any foreground objects.

We analyze the loss weight for this objective loss in Tab. S - 2. This ablation study is conducted upon the PV-RCNN [8] detector on the KITTI [4] datasets. The results on AP$_{3D}$ with 40 recall positions are reported as the metric. We change the loss weight values from {0.1, 0.5, 1.0, 2.0}. It shows that too large or too small loss weight values degrade the results. Loss weights 0.5 and 1.0 present competitive performance. We remain the 1.0 loss weight as a default setting for simplification.

C.3. Improvements on the Waymo Open Dataset.

To show our generalization capacity, we conduct further experiments on Waymo dataset. We use $\frac{1}{5}$ training data, following the default setting in the OpenPCdet codebase [1]. As shown in Tab. S - 3, Focals Conv also brings non-trivial improvements on the Waymo [9] dataset.

[1]https://github.com/open-mmlab/OpenPCDet

C.4. Improvements on the nuScenes val split.

Tab. S - 4 presents the improvements over CenterPoint [12] on the nuScenes *val* split. The CenterPoint baseline in Tab. S - 4 is implemented in the same settings to Focals Conv and Focals Conv - F. It shows that both Focals Conv and Focals Conv - F bring non-trivial improvements. Notably, Focals Conv - F improves the plain CenterPoint [12] by 5.9% mAP on the nuScenes *val* split.

C.5. Accuracy loss on some categories after fusion.

A surprising case is that the multi-modal fusion make the performance stay the same or worse on some popular categories, *e.g.*, Car, Ped, Bar (from *Focals Conv* to *Focals Conv - F* in Tab. 11). The improvements over the baseline are consistent on all categories. To analyze this special case, we conduct ablations on augmentations on CenterPoint and the nuScenes $\frac{1}{4}$ training set. We find *ground-truth sampling* (GT Sampl.) is the keypoint. As in Tab. A - 5, when GT Sampl. is used, the performance on some popular categories (*e.g., Car, Bus, Ped, Bar*) stays the same or worse. In contrast, when we disable GT Sampl. and apply all other transformations (flip, rotation, re-scaling, and translation), all categories are benefited from the fusion. We suppose that this is from the image-level copy-paste in GT Sampl. When other objects are pasted onto images, popular objects inevitably have more chance to be covered by the pasted, which degrades the performance on these categories.

C.6. Ablations on Voxel Size.

We ablate the effects of different voxel sizes upon Focals Conv - F on the nuScenes *val* split in Tab. S - 6. We change

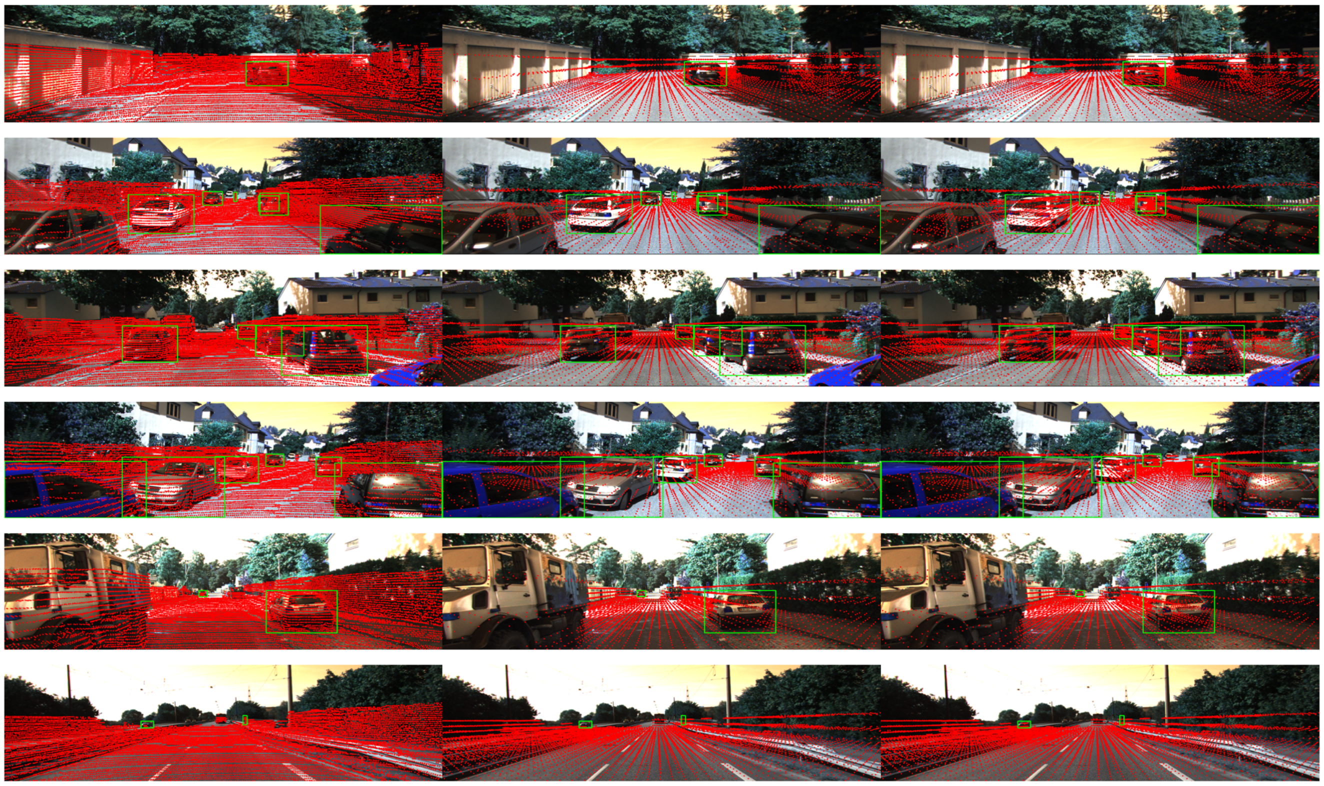
Figure S - 3. The illustration of visual comparisons between the plain and the focal sparse convolutional backbone networks.

the voxel sizes in X and Y axes from 0.05m to 0.15m, with the interval 0.025m. The overall mAP achieves the best performance at the voxel size (0.075, 0.075, 0.2)m. However, the proper voxel sizes vary across different classes. This phenomenon deserves further analysis or a dynamic mechanism design in the future.

D. Visualizations

We provide additional visual comparisons between the plain network and the focal sparse convolutional networks in Fig. S - 3. It shares the same settings to the Fig. 2 in the paper. These visualizations are based on the PV-RCNN [8] detectors and on the KITTI [4] dataset. In each visualization group, the top figure is the distribution of input voxels. The middle and the bottom figures are from the plain and the focal sparse convolutional networks, respectively. We project the coordinate centers of the output voxel features from the backbone networks onto the 2D image plane. The projection is based on the calibration matrices of KITTI [4].

References

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11618–11628, 2020. 1

[2] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel R-CNN: towards high performance voxel-based 3d object detection. In *AAAI*, pages 1201–1209, 2021. 1, 2

[3] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *ICCV*, pages 6568–6577, 2019. 2

[4] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *Int. J. Robotics Res.*, 32(11):1231–1237, 2013. 1, 2, 3, 4

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, volume 37, pages 448–456, 2015. 2

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015. 1

[7] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *T-PAMI*, 42(2):318–327, 2020. 3

[8] 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*, pages 10526–10535, 2020. 1, 2, 3, 4

[9] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, pages 2443–2451, 2020. 3

[10] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *CVPR*, pages 11794–11803, 2021. 1

[11] Yan Yan, Yuxing Mao, and Bo Li. SECOND: sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1

[12] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. In *CVPR*, pages 11784–11793, 2021. 1, 2, 3

[13] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, pages 4490–4499, 2018. 2