Appendix

In this supplementary material, we first introduce implementation details in Sec. A. It includes data processing, training settings, and network architectures. After that, in Sec. B, we introduce additional experimental results on the nuScenes dataset. We also provide additional visualizations on effective receptive fields (ERFs) in Sec. C and the illustration manner. Note that the rank of LargeKernel3D on nuScenes *test* is reported at the paper submission time. Methods that released afterwards are not counted. We visualize group manners of Tab. 4 in the paper in Fig. S - 1.

A. More Implementation Details

A.1. Data Processing

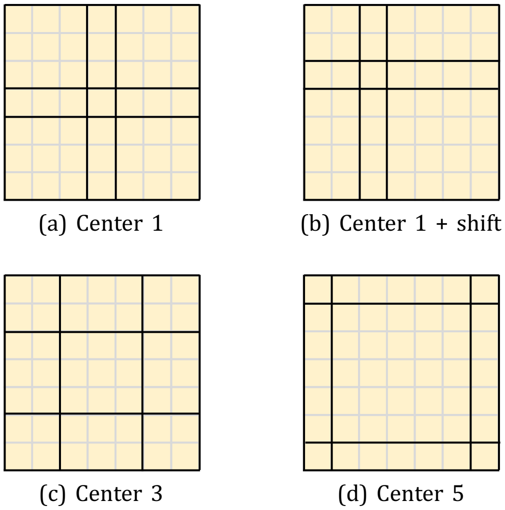
ScanNetv2 We convert point clouds into voxels as input data for the ScanNetv2 dataset. The voxelization sizes are all 0.02m for all *X*, *Y*, *Z* axes. In terms of data augmentations, we exactly follow our baseline method, MinkowskiNet [4]. Specially, input data is randomly dropped out with a ratio of 0.2. For spatial augmentations, we also conduct random horizontal flipping. For intensity augmentations, we conduct auto-contrast, color translation, and jittering. **nuScenes** We convert point clouds into voxels as input data.

We clip point clouds into [-54m, 54m] for both *X* and *Y* axes, and [-5m, 3m] for the *Z* axis, on nuScenes [2]. The voxel size is set as (0.075m, 0.075m, 0.2m). Data augmentations include random flipping, global scaling, global rotation, GT sampling [12], and an additional translation on the nuScenes [2] dataset. Random flipping is performed in *X* and *Y* axes. Rotation angle is sampled in [-45°, 45°]. Global scaling is conducted in the [0.9, 1.1] ratio. Translation noise is conducted on all three axes from the ratio [0, 0.5]. GT sampling is also conducted on the nuScenes.

Waymo Point clouds is clipped into [-75.2m, 75.2m] *X* and *Y* axis, and [-2m, 4m] for *Z* axis, on Waymo [10] for ranges. The input voxel size is set as (0.1m, 0.1m, 0.15m). The data augmentations include random flipping, global scaling, global rotation, and ground-truth (GT) sampling [12] for the Waymo dataset. Random flipping is applied along *X* and *Y* axes. Global scaling is sampled from the [0.95, 1.05] ratio. Global rotation is performed around the *Z* axis. Rotation angle is sampled from [-45°, 45°]. Ground-truth sampling copies objects from other training data, and pastes them onto the current scene. It enriches data variance during training. These settings follow baseline methods [9, 14].

A.2. Training Settings

ScanNetv2 For models trained on the ScanNetv2 dataset, we train networks for 600 epochs with batch size 16. The learning rate is initialized as 0.1 and decays with a poly scheduler. We adopt SGD optimizer. The momentum is set as 0.9. Hyper-parameters directly follow our baseline [4].


(a) Center 1

(b) Center 1 + shift

(c) Center 3

(d) Center 5

Figure S - 1. The center group manners in Tab. 4 in the paper. We study the center sizes for group splitting and center shifting.

nuScenes We train CenterPoint [14] on the nuScenes datasets for 20 epochs with batch size 32. This network is trained by Adam. The learning rate is set as 1e-3 and decays in the cosine annealing strategy to 1e-4. The weight decay is set as 0.01. The gradient norms are clipped by 35.

Waymo We train the network for 30 epochs and batch size 16 on Waymo. The learning rate is initialized as 0.003. Gradient norms are clipped by 10. We adopt the Adam optimizer, with weight decay 0.01 and momentum 0.9. These settings follow the CenterPoint [14] baseline.

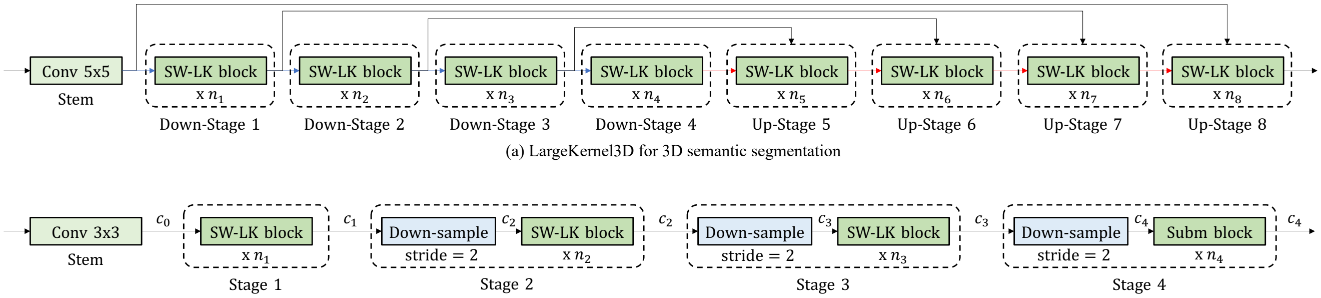
A.3. Network Architecture Settings

3D Semantic Segmentation We use MinkowskiNet-34 [4] as the baseline, for the ScanNetv2 dataset in the paper. In MinkowskiNet-34, we set the channel numbers as $\{32, 64, 128, 256, 256, 128, 96, 96\}$. The block numbers, $\{n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8\}$, are $\{2, 3, 4, 6, 2, 2, 2, 2\}$. The meanings of these notations are shown in Fig. S - 2. LargeKernel3D directly follows MinkowskiNet-34 for these settings. They substitute the plain sparse convolutional blocks to the proposed SW-LK Conv with spatial size 7^3 and groups 3^3. The tiny version, LargeKernel3D-T, has half channel numbers of the original in the last two stages.

3D Object Detection The backbone network of CenterPoint [14] has channels $\{c_0, c_1, c_2, c_3, c_4\}$, equal to $\{16, 16, 32, 64, 128\}$. The block numbers in these stages, $\{n_1, n_2, n_3, n_4\}$, are $\{2, 2, 2, 2\}$. Each block contains two convolutional layers, with a residual connection, except the stem. LargeKernel3D also substitutes the plain blocks for SW-LK

Table S - 1. Comparison with other methods and the ground-truth sampling fading (GT-S Fading) trick on the nuScenes *val* split.

Method	NDS	mAP	Car	Truck	Bus	Trailer	C.V.	Ped	Mot	Byc	T.C.	Bar
CenterPoint [14]	66.4	59.0	85.6	57.2	71.2	37.3	16.2	85.1	58.4	41.0	69.2	68.2
TransFusion [1]	66.8	60.0	85.8	57.6	71.6	37.3	19.3	86.7	57.2	42.3	71.0	69.7
Focals Conv [3]	67.2	60.2	85.7	58.4	71.5	37.8	19.0	85.5	58.5	45.6	70.4	69.2
LargeKernel3D	**67.5**	**60.3**	85.2	58.3	71.6	37.9	19.8	85.4	60.8	44.3	70.7	68.6
+ GT-S Fading	**69.1**	**63.9**	85.1	60.1	72.6	41.4	24.3	85.6	70.8	59.2	72.3	67.7



(a) LargeKernel3D for 3D semantic segmentation

(b) LargeKernel3D for 3D object detection

Figure S - 2. Architectures of LargeKernel3D for 3D semantic segmentation and object detection.

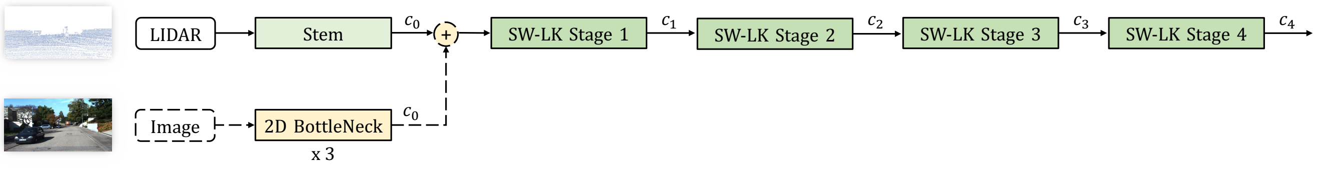


Figure S - 3. Architectures of LargeKernel3D-F with image fusion for 3D object detection.

blocks for stages 1, 2, 3. Because the last stage has heavy channel numbers and its receptive field is already sufficient.

We also present the multi-modal network with our large kernel backbone, *i.e.*, LargeKernel3D-F. As shown in Fig. S - 3, we conduct a direct voxel-wise summation between LIDAR and RGB features. The RGB branch only contains a conv-bn-relu-pooling stem and 3 residual bottlenecks [5]. We intentionally make the RGB branch lightweight to fully demonstrate the capacity of our large-kernel LIDAR backbone. These settings follow [3].

B. Additional Experimental Results

We present further improvements on the nuScenes dataset by additional techniques in Tab. 1. These techniques are removing gt-sampling in the last 5 training epochs (GT-S Fading). This trick has been used by some previous state-of-the-art methods [3, 11] for performance boosting. As shown in Tab. 1, LargeKernel3D achieves 63.9% mAP on the *val* split. This technique is included for test submission. For the multi-modal LargeKernel3D-F, it has the potential to achieve better performance if equipped with more advanced and heavier fusion methods [6, 7, 13]. We would like to try these extensions in future work.

C. Visualizations

We provide additional visual comparisons between the plain 3D network and our LargeKernel3D in Fig. S - 4. It shares the same setting as the Fig. 2 in the paper. In each group, the left one is the original image, the middle one is the ERFs of plain 3D CNNs, and the right one is the ERFs of our LargeKernel3D. We follow the definition of ERFs [8]. We calculate the gradient of every input voxel data regarding the feature of interest. It illustrates the intensity of feature changes as the input value changes. We normalize the gradient norms to [0, 1] and project them onto the image plane via calibration matrices.

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. *CoRR*, abs/2203.11496, 2022. 2

[2] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan,

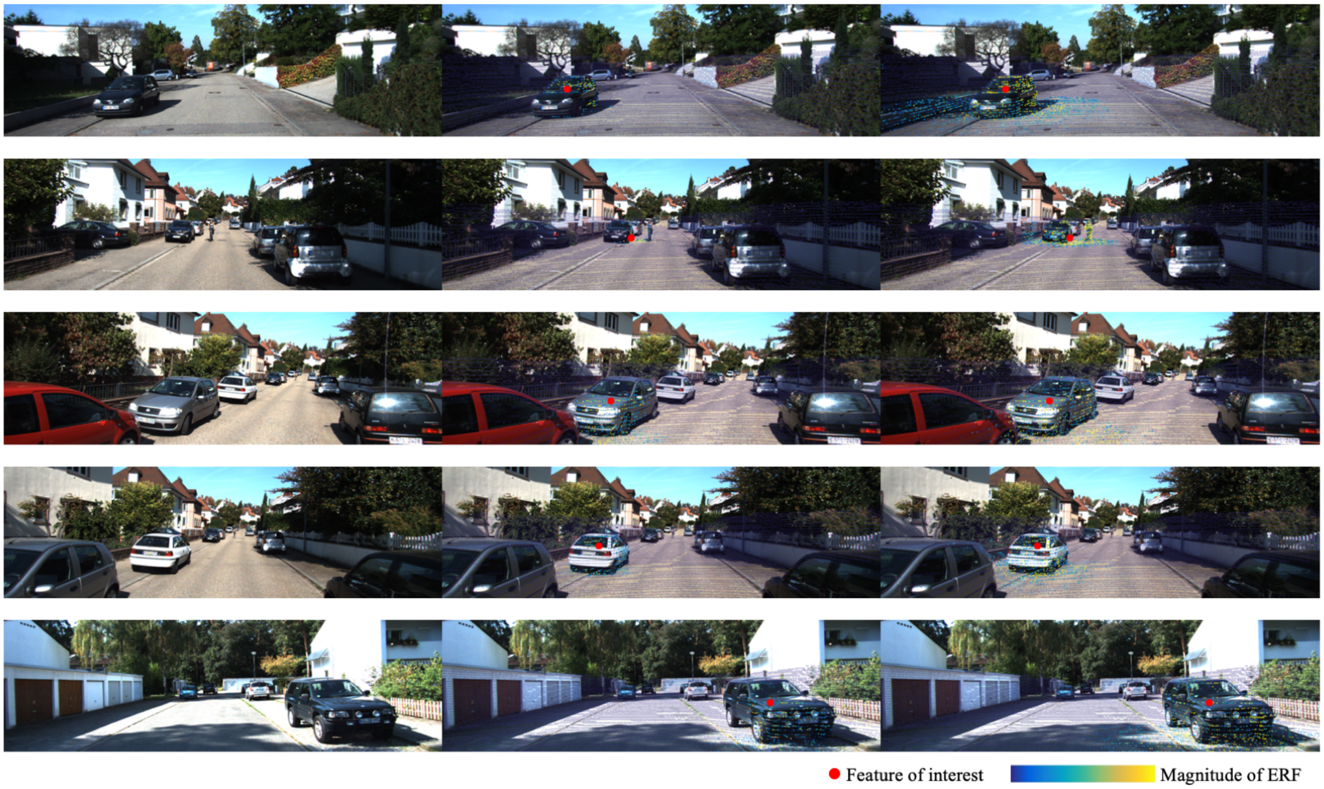


Figure S - 4. Additional illustrations on effective receptive fields. Left - original images, mid - plain 3D CNNs, right - LargeKernel3D. It is best viewed in color and by zooming in.

Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11618–11628, 2020. 1

[3] Yukang Chen, Yanwei Li, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Focal sparse convolutional networks for 3d object detection. *CoRR*, abs/2204.12463, 2022. 2

[4] Christopher B. Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *CVPR*, pages 3075–3084, 2019. 1

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 2

[6] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, and Zhi Tang. Bevfusion: A simple and robust lidar-camera fusion framework. *CoRR*, abs/2205.13790, 2022. 2

[7] 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. *CoRR*, abs/2205.13542, 2022. 2

[8] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard S. Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *NeurIPS*, pages 4898–4906, 2016. 2

[9] 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

[10] 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. 1

[11] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *CVPR*, pages 11794–11803, 2021. 2

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

[13] Zeyu Yang, Jiaqi Chen, Zhenwei Miao, Wei Li, Xiatian Zhu, and Li Zhang. Deepinteraction: 3d object detection via modality interaction. *CoRR*, abs/2208.11112, 2022. 2

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