SparseOcc: Rethinking Sparse Latent Representation for Vision-Based Semantic Occupancy Prediction —Supplementary Material—

Pin Tang[1] Zhongdao Wang[2] Guoqing Wang[1] Jilai Zheng[1]
Xiangxuan Ren[1] Bailan Feng[2] Chao Ma[1*]
[1] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[2] Huawei Noah's Ark Lab
{pin.tang,guoqing.wang,zhengjilai,bunny_renxiangxuan,chaoma}@sjtu.edu.cn
{wangzhongdao,fengbailan}@huawei.com
Project page: https://pintang1999.github.io/sparseocc.html

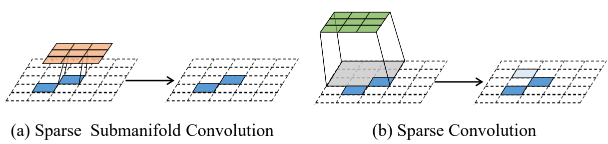

Figure A. **Comparison between sparse submanifold convolution and sparse convolution.** For simplicity, 2D feature map and 2D kernels are utilized.

A. Sparse Convolution Description

We detail the difference between sparse submanifold convolution and sparse convolution [3] in a 2D sparse view in A. As can be seen, submanifold convolution only operates on occupied voxels, thus ensuring that an output location is active only if the corresponding input location is active, thereby maintaining sparsity even when stacking multiple layers. On the other hand, a sparse convolution performs the computation in a local window in which at least one non-empty voxel resides, allowing the diffusion of features from non-empty voxels to their neighbors. Hence, we use sparse convolution for scene completion and submanifold convolution for contextual feature exchange.

B. More Experiments

B.1. Scaling up the 3D resolution.

We utilize LSS [7] to lift the 2D image features to 3D volume from which we extract 3D sparse representation. The 3D feature resolution of the volume output by LSS may affect the performance. For efficiency, we use SparseOcc with a linear layer segmentation head to study it. As shown in Table. A, scaling up the size of LSS output by 2× boosts the mIoU from 11.8 to 12.3 while decreasing the geometry IoU by 0.4. We guess the IoU drop is caused by the number of sparse completion blocks, as it may not be enough to complete the whole scene in a bigger resolution. This problem can be resolved by stacking one more sparse completion block at the last layer of the 3D sparse encoder.

* Corresponding author: chaoma@sjtu.edu.cn.

Type	3D resolution	IoU	mIoU
SparseOcc-Linear	128×128×16	**36.8**	11.8
	256×256×256	36.4	**12.3**

Table A. Scaling up the 3D representation resolution on SemanticKITTI [1] validation set.

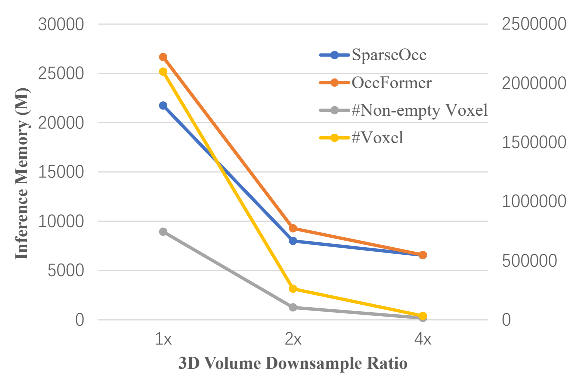

Figure B. **Efficiency analysis when scaling up the 3D representation on SemanticKITTI [1] validation set.** The left axis represents the inference GPU memory. The right axis denotes the number of voxels. The 3D downsampling ratio is considered between the ground-truth and the LSS output. The number of non-empty voxels is measured using max-donwsampled ground-truth.

Method	Input Modality	Backbone	Image Size	mIoU	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. suf.	other flat	sidewalk	terrain	manmade	vegetation
RangeNet++ [6]	LiDAR	-	-	65.5	66.0	21.3	77.2	80.9	30.2	66.8	69.6	52.1	54.2	72.3	94.1	66.6	63.5	70.1	83.1	79.8
PolarNet [8]	LiDAR			71.0	74.7	28.2	85.3	90.9	35.1	77.5	71.3	58.8	57.4	76.1	96.5	71.1	74.7	74.0	87.3	85.7
Salsanext [4]	LiDAR			72.2	74.8	34.1	85.9	88.4	42.2	72.4	72.2	63.1	61.3	76.5	96.0	70.8	71.2	71.5	86.7	84.4
Cylinder3D [10]	LiDAR			**76.1**	**76.4**	**40.3**	**91.2**	**93.8**	**51.3**	**78.0**	**78.9**	**64.9**	**62.1**	**84.4**	**96.8**	**71.6**	**76.4**	**75.4**	**90.5**	**87.4**
TPVFormer [5]	Camera	R50	850×450	59.3	64.9	27.0	83.0	82.8	38.3	27.4	44.9	24.0	55.4	73.6	91.7	60.7	59.8	61.1	78.2	76.5
OccFormer [9]	Camera		704×256	68.1	69.2	36.9	**91.2**	84.4	47.3	59.1	61.9	42.1	58.8	82.8	93.0	67.5	67.4	68.5	81.0	78.5
SparseOcc (ours)	Camera		704×256	68.4	69.1	40.4	89.1	85.0	49.9	71.0	62.0	38.1	58.1	79.4	92.9	65.8	66.2	67.0	80.9	78.9

Table B. **LiDAR segmentation results on nuScenes validation set.** For vision-based methods, we list the utilized image backbone and the input image sizes. The **bold** numbers indicate the best results in the whole table and the green numbers indicate the best results in vision-based methods.

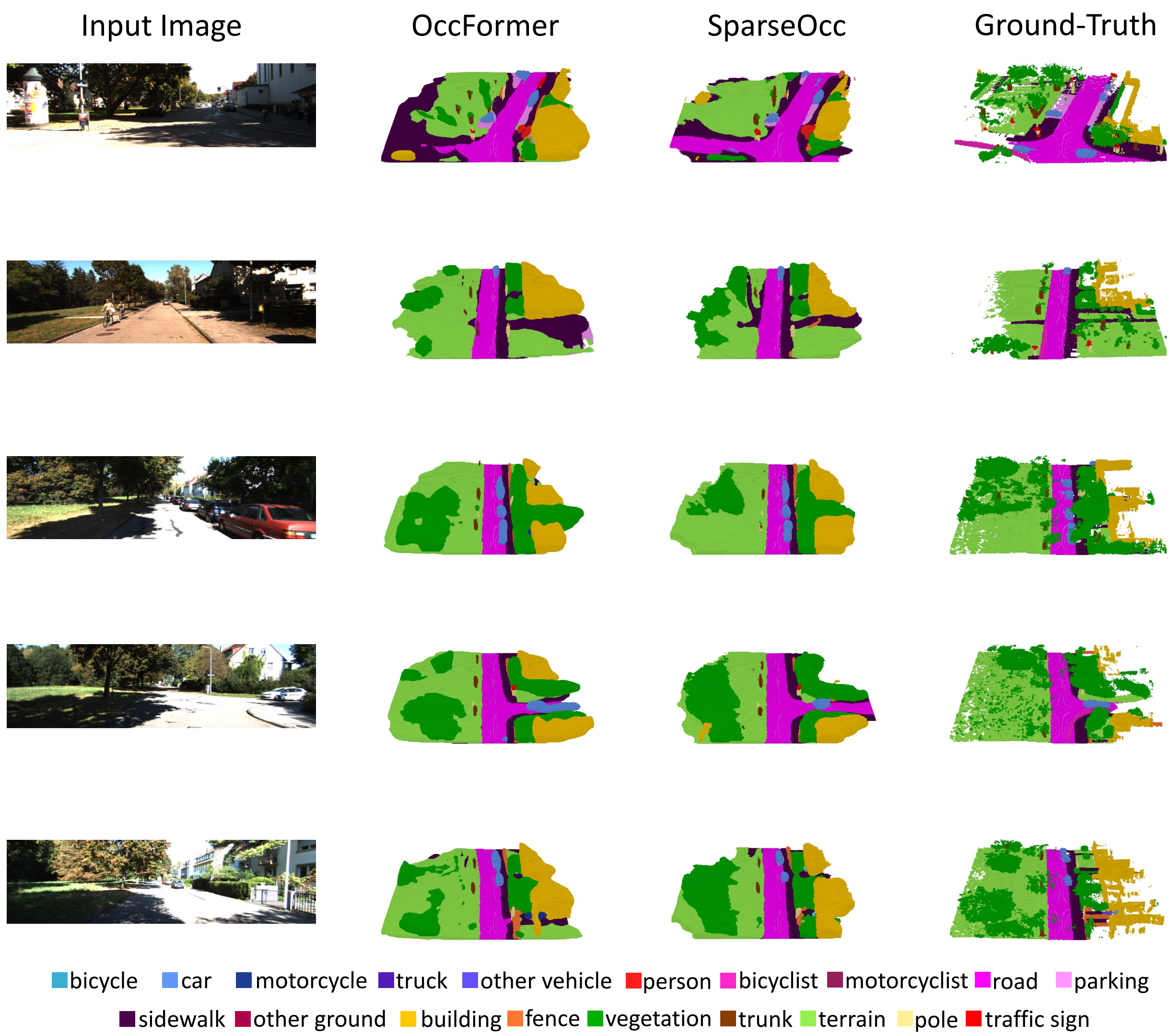

Figure C. **Qualitative results of 3D semantic scene completion on SemanticKITTI [1] validation set.** The input monocular image is shown on the left, and the 3D semantic scene completion results from OccFormer [9], our SparseOcc, and the ground-truth are then visualized sequentially.

B.2. Efficiency analysis.

The complexity of our SparseOcc should be roughly linear to the number of non-empty voxels since it only operates on feature-occupied voxels. We compare the inference GPU memory and the number of non-empty voxels in Fig. B. As can be observed, when scaling up the 3D resolution from $2\times$ to $1\times$, the 3D dense representation based method OccFormer [9] sufferes from a steep inference GPU memory rise while our SparseOcc presents a linear increase to the non-empty voxels, which justifies the superior efficiency of 3D sparse representation.

B.3. Point Cloud Semantic Segmentation

Following the former practices [5, 9], we report the point cloud semantic segmentation results on nuScenes [2] validation set. Different from semantic occupancy prediction, point cloud segmentation does not have to predict the "empty" class and reconstruct the occluded part. We build the model as the same as the semantic occupancy prediction but only use point cloud semantic labels for supervision. As displayed in Table. B, our SparseOcc outperforms the vision-based methods, i.e., TPVFormer [5] and OccFormer [9] by 9.1 and 0.3 mIoU. Note that TPVFormer uses 2D projection based representation and OccFormer uses 3D dense representation, while SparseOcc uses efficient 3D sparse representation. Moreover, SparseOcc also achieves comparable accuracy with the state-of-the-art LiDAR-based methods [4, 6, 8, 10], which further demonstrates the generalization ability and potential of the proposed 3D sparse latent representation.

C. Additional Visualizations.

We visualize the predicted results of semantic scene completion from OccFormer [9] and our proposed SparseOcc on SemanticKITTI [1] validation set. As can be observed from Fig. C, SparseOcc mitigates the hallucinations on empty voxels compared with OccFormer. We blame the hallucinations of OccFormer on dense operators like large-window Swin Transformer blocks and 3D deformable self-attention, while our SparseOcc represents the scene with 3D sparse representation and only operates on feature-occupied voxels, thus relieving the hallucination problem.

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *ICCV*, 2019. 1, 2, 3

[2] 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*, 2020. 3

[3] Spconv Contributors. Spconv: Spatially sparse convolution library. https://github.com/traveller59/spconv, 2022. 1

[4] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In *International Symposium on Visual Computing*, 2020. 2, 3

[5] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. *arXiv:2302.07817*, 2023. 2, 3

[6] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *IROS*, 2019. 2, 3

[7] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, 2020. 1

[8] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *CVPR*, 2020. 2, 3

[9] Yunpeng Zhang, Zheng Zhu, and Dalong Du. Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. *arXiv:2304.05316*, 2023. 2, 3

[10] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *CVPR*, 2021. 2, 3