Tri-Perspective View for Vision-Based 3D Semantic Occupancy Prediction
Supplementary Material

Yuanhui Huang* Wenzhao Zheng* Yunpeng Zhang Jie Zhou Jiwen Lu†
Beijing National Research Center for Information Science and Technology, China
Department of Automation, Tsinghua University, China
{huangyh22,zhengwz18}@mails.tsinghua.edu.cn; yunpengzhang97@gmail.com;
{jzhou,lujiwen}@tsinghua.edu.cn

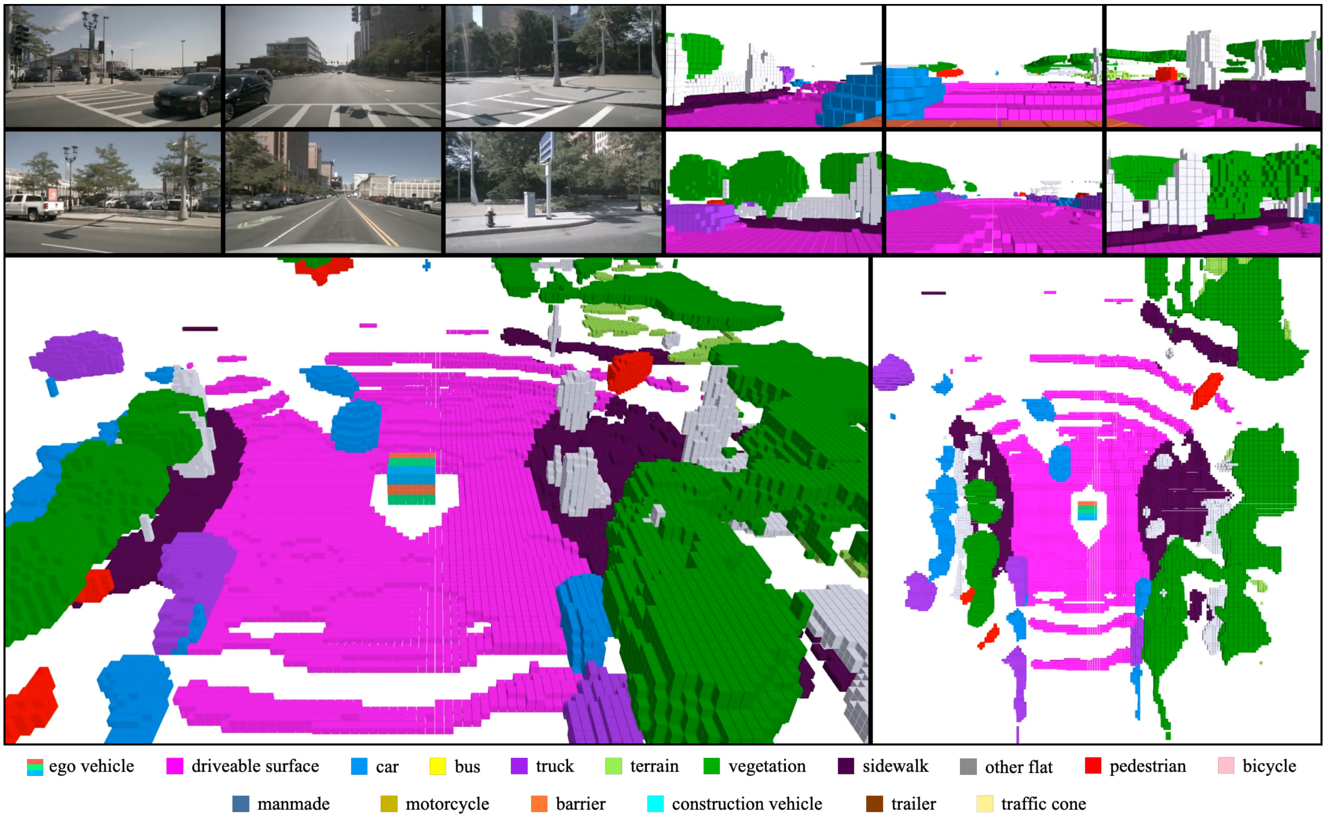

Figure 1. An image sampled from the video demo for 3D semantic occupancy prediction on the nuScenes validation set (not seen in the training phase). The six images in the top left are the inputs to our model captured by the front-left, front, front-right, back-left, back, and back-right cameras. The six images in the top right denote our prediction results with the corresponding views as the inputs. The bottom two images provide a global view of our predictions where the red-green-blue box represents the ego vehicle.

A. Dataset Details

The Panoptic nuScenes dataset [7] collects 1000 driving scenes of 20 seconds duration each, and the keyframes are annotated at 2Hz. Each sample contains RGB images from 6 cameras with 360° horizontal FOV and point cloud data from 32 beams LiDAR sensor. The total of 1000 scenes are officially divided into training, validation and test splits with 700, 150 and 150 scenes, respectively.

The SemanticKITTI dataset [1] includes outdoor-scene automotive LiDAR scans voxelized into $256 \times 256 \times 32$ grids. Each voxel has a side length of 0.2m and is labeled with one of 21 classes (19 semantic, 1 free and 1 unknown). In our experiments, we also use RGB images captured by cam2 from the KITTI odometry benchmark. The voxel and image data is officially arranged as 22 sequences, split into 10/1/11 sequences for training, validation and test.

B. Implementation Details

3D semantic occupancy prediction and LiDAR segmentation. TPVFormer-Base uses the ResNet101-DCN [5, 8] initialized from FCOS3D [15] checkpoint, while TPVFormer-Small adopts the ResNet-50 [8] pretrained on

ImageNet [6]. The TPV resolutions are 200x200x16 and 100x100x8 for the base and small versions, respectively, and we upsample the TPV planes by a factor of 2 in TPVFormer-Small for finer supervision. Although both of them share the same TPV feature dimension of 128, the base model uses multi-scale image features and an input image resolution of 1600x900 instead of single-scale image features and 800x450 input for the small model.

For training, we adopt the AdamW [11] optimizer with initial learning rate as 2e-4 and weight decay as 0.01. We use the cosine learning rate scheduler with a linear warming up in the first 500 iterations, and the same image augmentation strategy as BEVFormer [10]. All models are trained for 24 epochs with a batch size of 8 on 8 A100 GPUs.

Semantic Scene Completion. We adopt the 2D UNet based on a pretrained EfficientNetB7 [14] as 2D backbone to generate multi-scale image features, which is the same as MonoScene. Moreover, we set the resolution of TPV planes as 128x128x16 to generate a 3D voxel feature tensor of the same size as MonoScene, although our TPV planes are 2D feature maps while MonoScene operates directly on 3D voxel features. We use RGB images from cam2 cropped to 1220x370 as input and a feature dimension of 96. For optimization, we employ the losses in MonoScene except for the relation loss, since TPVFormer does not have the 3D CRP module or any downsampling operation. For training, we generally follow the recipe in MonoScene. Specifically, we use a learning rate of 2e-4, a weight decay of 0.01, and a cosine scheduler. We keep the other settings the same. For a fair comparison, we also rerun the official code of MonoScene with a cosine learning rate scheduler.

C. 3D Semantic Occupancy Prediction Results

We provide a video demo on our website[1] for 3D semantic occupancy prediction on nuScenes validation set with a sampled image in Figure 1. Figure 2 provides detailed visualization results of our model for four samples from nuScenes validation set. For each sample, we present the six surround camera images, the top view of the predicted scene, and the zoomed-in results from three different angles. In addition, we highlight predictions for small and rare objects with circles and further link them to corresponding ground truths in RGB images with arrowed dash lines. Specifically, we highlight bicycles, motorcycles, and pedestrians with red, blue, and yellow circles, respectively. Note that although some of these objects are barely visible in RGB images, our model still predicts them successfully.

D. LiDAR segmentation Results

In Table 1, we report the performance of TPVFormer on nuScenes validation set for LiDAR segmentation. For a fair comparison, we replace the temporal module in BEVFormer with self-attention moduel and use a feature dimension of 256 to make the model sizes of BEVFormer-Base and TPVFormer-Base comparable. The mIoU of TPVFormer-Base is on par with LiDAR-based methods despite critical modal differences. Furthermore, our TPVFormer-Base achieves a 12.7% higher mIoU than BEVFormer-Base, which demonstrates the effectiveness of TPV in modeling fine-grained 3D structures of a scene.

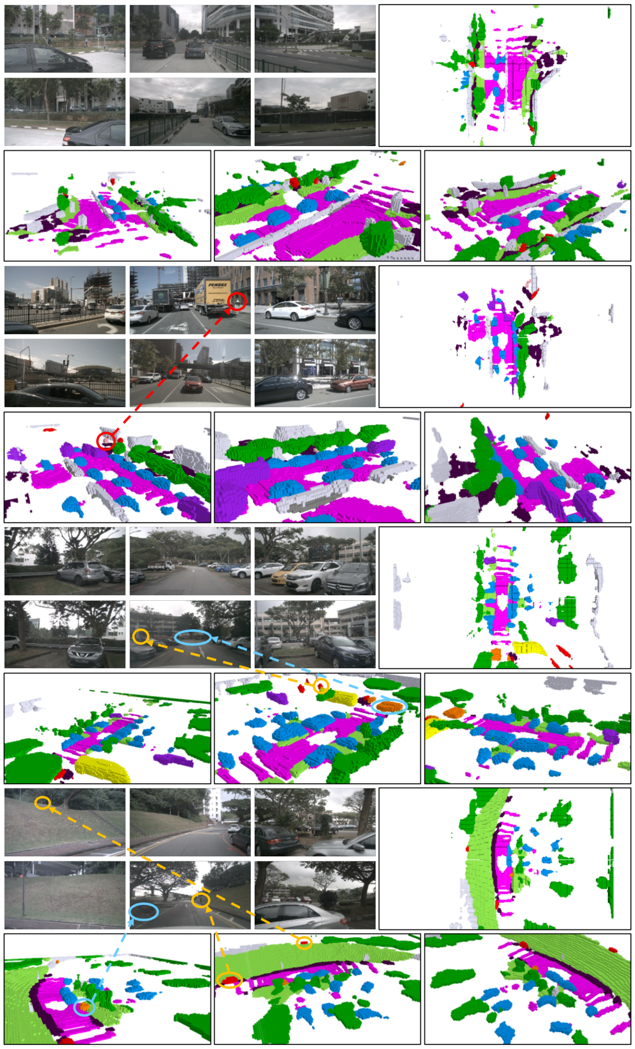

Figure 2. More visualizations of the proposed TPVFormer.

E. Semantic Scene Completion Results

We present the semantic scene completion performance on SemanticKITTI validation set in Table 2. Although TPVFormer does not achieve the highest IoU for scene completion, it outperforms other methods in mIoU with a clear margin for semantic scene completion. We reproduce MonoScene [2] with the official code in our environment and also report its performance using the cosine learning rate following our recipe for a fair comparison.

[1] https://wzzheng.net/TPVFormer/

Table 1. **LiDAR segmentation results on nuScenes validation set.** Despite critical modal difference, our TPVFormer-Base achieves comparable performance with LiDAR-based methods. Moreover, the mIoU gap between BEVFormer and TPVFormer clearly proves the effectiveness of TPV in modelling fine-grained 3D structures of a scene.

Method	Input Modality	mIoU	barrier 🟧	bicycle 🟪	bus 🟨	car 🟦	const. veh. 🟦	motorcycle 🟨	pedestrian 🟥	traffic cone 🟨	trailer 🟫	truck 🟪	drive. suf. 🟪	other flat ⬛	sidewalk 🟪	terrain 🟩	manmade ⬜	vegetation 🟩
RangeNet++ [12]	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 [17]	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++ [18]	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**
BEVFormer-Base [10]	Camera	56.2	54.0	22.8	76.7	74.0	45.8	53.1	44.5	24.7	54.7	65.5	88.5	58.1	50.5	52.8	71.0	63.0
TPVFormer-Small (ours)	Camera	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
TPVFormer-Base (ours)	Camera	68.9	70.0	**40.9**	**93.7**	85.6	49.8	68.4	59.7	38.2	**65.3**	83.0	93.3	64.4	64.3	64.5	81.6	79.3

Table 2. **Semantic scene completion results on SemanticKITTI validation set.** For a fair comparison, we use the performances of RGB-inferred versions of the first four methods reported in MonoScene [2]. * represents the reproduced result using the official code. ** represents result using the cosine learning rate schedule.

Method	Input Modality	SC IoU	SSC mIoU	road (15.30%)	sidewalk (11.13%)	parking (1.12%)	other-grnd (0.56%)	building (14.1%)	car (3.92%)	truck (0.16%)	bicycle (0.03%)	motorcycle (0.03%)	other-veh. (0.20%)	vegetation (39.3%)	trunk (0.51%)	terrain (9.17%)	person (0.07%)	bicyclist (0.07%)	motorcyclist. (0.05%)	fence (3.90%)	pole (0.29%)	traf.-sign (0.08%)
LMSCNet [13]	Camera	28.61	6.70	40.68	18.22	4.38	0.00	10.31	18.33	0.00	0.00	0.00	0.00	13.66	0.02	20.54	0.00	0.00	0.00	1.21	0.00	0.00
3DSketch [3]	Camera	33.30	7.50	41.32	21.63	0.00	0.00	14.81	18.59	0.00	0.00	0.00	0.00	**19.09**	0.00	26.40	0.00	0.00	0.00	0.73	0.00	0.00
AICNet [9]	Camera	29.59	8.31	43.55	20.55	11.97	0.07	12.94	14.71	4.53	0.00	0.00	0.00	15.37	2.90	28.71	0.00	0.00	0.00	2.52	0.06	0.00
JS3C-Net [16]	Camera	**38.98**	10.31	50.49	23.74	11.94	0.07	**15.03**	**24.65**	4.41	0.00	0.00	**6.15**	18.11	**4.33**	26.86	0.67	0.27	0.00	3.94	3.77	1.45
MonoScene* [2]	Camera	36.86	11.08	**56.52**	**26.72**	14.27	0.46	14.09	23.26	6.98	0.61	0.45	1.48	17.89	2.81	29.64	**1.86**	**1.20**	0.00	5.84	**4.14**	**2.25**
MonoScene** [2]	Camera	36.13	10.98	56.30	25.89	15.91	0.75	13.47	23.31	5.36	**0.72**	**0.91**	3.77	17.70	2.45	27.12	1.71	1.08	0.00	**6.34**	3.79	2.03
TPVFormer (ours)	Camera	35.61	**11.36**	56.50	25.87	**20.60**	**0.85**	13.88	23.81	**8.08**	0.36	0.05	4.35	16.92	2.26	**30.38**	0.51	0.89	0.00	5.94	3.14	1.52

F. Inference Time for Each Component

We computed the inference time for each component in Table 3. We see that the segmentation head and point querying mechanism enjoy great efficiency, while the TPV encoder accounts for most of the latency. We think the high latency of the TPV encoder might be due to the slow for loops to filter out the inactive points in image cross-attention.

Table 3. **Detailed inference time for each component.** *: 0.0029s is the point querying time inside the segmentation head.

Image backbone (s)	TPV encoder (s)	Segmentation head (s)	Total (s)
0.026	0.283	0.0032 (0.0029*)	0.312

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*, pages 9297–9307, 2019. 1

[2] Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion. In *CVPR*, pages 3991–4001, 2022. 2, 3

[3] Xiaokang Chen, Kwan-Yee Lin, Chen Qian, Gang Zeng, and Hongsheng Li. 3d sketch-aware semantic scene completion via semi-supervised structure prior. In *CVPR*, pages 4193–4202, 2020. 3

[4] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In *ISVC*, pages 207–222, 2020. 3

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. 1

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 2

[7] Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *arXiv preprint arXiv:2109.03805*, 2021. 1

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[9] Jie Li, Kai Han, Peng Wang, Yu Liu, and Xia Yuan. Anisotropic convolutional networks for 3d semantic scene completion. In *CVPR*, pages 3351–3359, 2020. 3

[10] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera

images via spatiotemporal transformers. In *ECCV*, 2022. 2, 3

[11] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 2

[12] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *IROS*, pages 4213–4220, 2019. 3

[13] Luis Roldao, Raoul de Charette, and Anne Verroust-Blondet. Lmscnet: Lightweight multiscale 3d semantic completion. In *2020 International Conference on 3D Vision (3DV)*, pages 111–119. IEEE, 2020. 3

[14] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114, 2019. 2

[15] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *ICCV*, 2021. 1

[16] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In *AAAI*, volume 35, pages 3101–3109, 2021. 3

[17] 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*, pages 9601–9610, 2020. 3

[18] 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*, pages 9939–9948, 2021. 3