Scene as Occupancy
Appendix

Wenwen Tong[1,2*], Chonghao Sima[1,3*†], Tai Wang[1,4], Li Chen[1,3], Silei Wu[2], Hanming Deng[2], Yi Gu[1], Lewei Lu[2], Ping Luo[3], Dahua Lin[1,4], Hongyang Li[1†]

[1] Shanghai AI Laboratory [2] SenseTime Research
[3] The University of Hong Kong [4] The Chinese University of Hong Kong

[*]Equal contribution [†]Project lead

https://github.com/OpenDriveLab/OccNet
https://github.com/OpenDriveLab/OpenScene

As presented in the main paper, we put the detail of evaluation metrics in the supplementary materials, along with more related work, visualization, implementation / training detail and more ablation of OccNet, and detail about BEVNet, VoxelNet and OpenOcc post-processing.

1. Evaluation Metrics

Semantic Scene Completion (SSC) Metric. For the scene completion task, we predict the semantic label of each voxel in 3D space. The evaluation metric is defined by mean intersection-over-union (mIoU) over all classes:

$$\text{mIoU} = \frac{1}{C}\sum_{c=1}^{C}\frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c + \text{FN}_c}, \qquad (1)$$

where $C = 16$ is the class num in the benchmark, TP_c, FP_c and FN_c represent true positive, false positive, and false negative predictions for class c, respectively. In addition, we consider the class-agnostic metric IoU_{geo} to evaluate the geometrical reconstruction quality of scene.

3D Object Detection Metric. We use the official evaluation metrics for the nuScenes datasets [1], including nuScenes detection score (NDS), mean average precision (mAP), average translation error (ATE), average scale error (ASE), average orientation error (AOE), average velocity error (AVE) and average attribute error (AAE).

Motion Planning Metric. For planning evaluation, we follow the metrics in ST-P3 [3]. In detail, L2 distance is calculated by the planning trajectory and the ground-truth trajectory for the regression accuracy, and collision rate (CR) to other vehicles and pedestrians is applied for the safety of future actions.

2. More Related Work

BEV segmentation [9, 4] implicitly squeezes the height information into each cell in BEV map. However, in some challenging urban settings, explicit height information is necessary to capture entities above the ground, *e.g.* traffic lights and overpass. As an alternative, 3D occupancy is 3D geometry-aware.

3. Implementation Detail of OccNet

Backbone and Multi-scale Features. Following previous works [5, 8], We adopt ResNet101 [2] as the backbone with FPN [6] to extract the multi-scale features from multi-view images. We use the output features from stages S_3, S_4, and S_5 from ResNet101, where S_n means the downsampling factor is $1/2^n$ with the feature dimension $C_n = 256 \times 2^{n-2}$. In the FPN, the features are aggregated and transforms to three levels with sizes of 1/16, 1/32, 1/64 and the dimension of $C_n = 256$.

BEV Encoder. The BEV encoder follows the structure of BEVFormer [5], where the multi-scale features from FPN are transformed into the BEV feature. The BEV encoder includes 2 encoder layers with the temporal self-attention and spatial cross-attention. Then the BEV query Q_t gradually refines in the encoder layers with spatial-temporal-transformer mechanism to learn the scene representation in BEV space.

Feature Transformation in Voxel Decoder. To lift the voxel feature $V^{'}_{t,i} \in \mathbb{R}^{Z_i \times H \times W \times C_i}$ to $V^{'}_{t,i+1} \in \mathbb{R}^{Z_{i+1} \times H \times W \times C_{i+1}}$, we use the MLP to transfer the feature dimension from $Z_i \times C_i$ to $Z_{i+1} \times C_{i+1}$. To implement the spatial cross attention for $V^{'}_{t,i}$, the multi-scale image features from FPN with dimension of $C_n = 256$ are transformed into dimension of C_i utilizing the MLP.

Training Strategy. Following previous works [5, 8], we train OccNet 24 epochs with a learning rate of 2×10^{-4}, a batchsize of 1 per GPU with six images, and AdamW optimizer [7] with a weight decay of 1×10^{-2}. For the implementation of downstream tasks, all the perception tasks (except BEV segmentation) are trained at once, and the others are fine-tuned based on the frozen tasks.

Details of VoxelNet and BEVNet. Different from the OccNet with cascaded feature map, we construct the VoxelNet and BEVNet with single-scale feature map. In detail, VoxelNet uses voxel queries $Q_{voxel} \in \mathbb{R}^{4\times H\times W}$ to construct the voxel feature map $F_{voxel} \in \mathbb{R}^{4\times H\times W\times C_1}$ from the image feature using 3D-DA directly, and expands it to full-scale occupancy $V \in \mathbb{R}^{16\times H\times W\times C_2}$ using fully connected layer. BEVNet generates BEV feature $F_{bev} \in \mathbb{R}^{H\times W\times C_1}$ as in BEVFormer and reshapes it to voxel feature $V \in \mathbb{R}^{16\times H\times W\times C_2}$ directly. Here C_1 and C_2 stand for the number of channels. Both VoxelNet and BEVNet adopt temporal context fusion accordingly.

4. More Detail about OpenOcc

Accumulation of Foreground objects. To accumulate the foreground object, we split the LiDAR points into object points and background points. However, the 3D box annotation of intermediate frame is not provided in the nuScenes dataset [1]. We approximately annotate the 3D box using the linear interpolation based on two adjacent key frames, then we can accumulate dense object points with available intermediate LiDAR points.

Dataset Generation Pipeline. With accumulated dense background points and foreground object points, we generate the occupancy data following the pipeline as shown in Figure 1. We gradually fine tune the occupancy data and obtain the 3D occupancy benchmark with dense and high-quality annotations in Figure 1(d).

Dataset Statistics. We annotate 16 classes in 34149 frames for all 700 training and 150 validation scenes with over 1.4 billion voxels. The label distribution of 16 classes is shown in Figure 2, indicating great diversity in the benchmark. There exists a significant class imbalance phenomenon in the dataset, for example, where the 10 foreground objects only account for 5.33% of the total labels, especially the bicycle and motorcycle, which account for 0.02% and 0.03%, respectively.

We provide the additional flow annotation of eight foreground objects, which is helpful for the downstream task such as motion planning. We split the object into moving state and stationary state based on the velocity threshold $v_{th} = 0.2$m/s, and the percentage of moving object for each class is given in Figure 3. Note that the percentage of moving foreground object is over 50%, indicating the significance of motion information in the autonomous driving scenes.

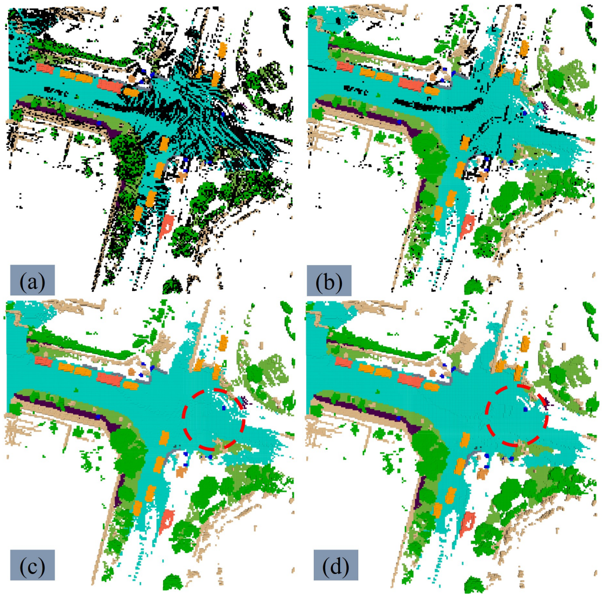

Figure 1. **The generation process of our occupancy data.** (a) Generating the occupancy data based on objects points and partial background points with label, where the black points denotes the unknown background points from the intermediate frame. (b) Annotating partial unknown background points based on generated occupancy data. (c) Removing the remaining unknown background points which are regarded as noise. (d) Postprocessing the occupancy data to ensure the completeness of the scene, such as fill the hole, denoted by the red dashed box.

5. More Experiments

Ablations on Frame Number for Temporal Self-Attention. We investigate the effect of frame number applied for temporal self-attention during training. From Table 1 and Table 2, we find that increasing temporal frames results in better performance, which slows down until a threshold of four frames is reached. Meanwhile, insufficient previous frames would hurt the performance to some extent.

Evaluation on Occupancy Metrics for Planning. We utilize the ground truth of occupancy as the metrics to evaluate the planning model, instead of the bounding box of vehicles and pedestrians. Specifically, all of foreground occupancy voxels and four classes of background occupancy voxels, i.e., other flat, terrain, manmade, and vegetation, are

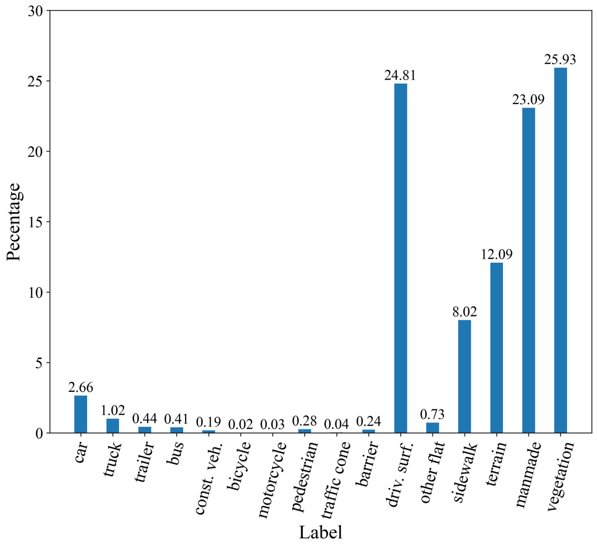

Figure 2. **The distribution of occupancy classes in the OpenOcc benchmark.** We notice that the background stuff is the majority in 3D occupancy data.

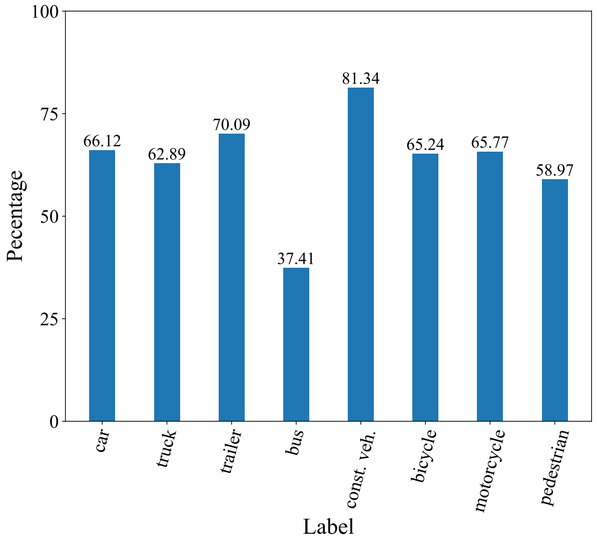

Figure 3. **The percentage of occupancy with velocity for each foreground object.** For 10 foreground objects in the benchmark, we only consider the 8 movable classes.

#Num	IoU$_{geo}$↑	mIoU↑	barrier↑	bicycle↑	bus↑
0	37.49	19.21	20.07	4.70	24.11
1	36.89	18.35	18.77	4.51	21.66
2*	37.69	19.48	20.63	5.52	24.16
3	38.36	20.30	21.39	6.47	24.65
4	39.21	**20.81**	**22.30**	5.66	**25.13**
9	**39.36**	20.68	20.75	**7.83**	24.79

Table 1. **The effect of historical frames on the semantic scene completion task using OccNet with ResNet50 backbone.** The "#Num" denotes the historical frame number used during training. * stands for number used in the main paper.

#Num	mIoU↑	barrier↑	bicycle↑	bus↑	car↑	truck↑
0	53.82	59.02	24.05	67.61	69.59	59.38
1	48.41	57.45	21.52	57.71	67.26	44.60
2	52.33	61.41	26.07	73.97	70.56	52.64
3	53.49	60.98	24.45	70.20	69.37	56.84
4	**54.59**	**62.09**	21.06	75.05	70.20	59.40
9	54.35	60.04	**30.83**	**75.49**	**71.02**	**61.63**

Table 2. **The effect of historical frames on the LiDAR segmentation task using OccNet with ResNet50 backbone.** The "#Num" denotes the historical frame number used during training.

Input	Collision (%)↓			L2 (m)↓		
	1s	2s	3s	1s	2s	3s
Bbox GT	1.66	2.88	4.37	1.33	2.18	3.03
Occupancy GT	**1.63**	**2.85**	**4.29**	**1.29**	**2.13**	**2.99**
Bbox pred. (OccNet)	1.75	**2.85**	4.37	1.33	2.17	3.04
Occupancy pred. (OccNet)	**1.68**	2.94	**4.32**	**1.30**	**2.15**	**3.02**

Table 3. **Planning results with different scene representations under occupancy metrics.** Occupancy representation is still more advantageous most of the intervals.

calculated collision rate with trajectory. As shown in Table 3, using occupancy as input for planning model is still more advantageous in most of the intervals under this metrics. In the future research, a specific design of cost function for occupancy input may further improve the performance of planning.

Pre-training for planning. As evaluating the pre-trained model on 3D detection and BEV segmentation tasks in the main paper, we further compared the impact on the downstream planning task. Specifically, the perception module of ST-P3 [3] is replaced by pre-trained OccNet, and the planning module is fine-tuned. Unfortunately, the pre-training on OccNet does not provide an advantage for planning as shown in Table 4. Therefore, combined with the experiment of planning in the main paper, we should directly apply the scene completion results of occupancy in the planning task instead of these pre-trained features.

Input	Collision (%)↓			L2 (m)↓		
	1s	2s	3s	1s	2s	3s
Det	**0.38**	**0.40**	**0.82**	**0.85**	**1.18**	**1.57**
Occ	0.47	0.68	1.03	0.93	1.26	1.70

Table 4. **Different pretraining tasks for planning.** Pretrained features from occupancy do not directly bring performance benefits to planning.

Ablations in Semantic Scene Completion. Table 5 shows the comparison of BEVNet, VoxelNet, OccNet in the task of semantic scene completion. We can see that the design of cascaded voxel structure can help learn a bettern occupancy descriptor to represent the 3D space.

Effect of Voxel Resolution on LiDAR Segmentation. We voxelize the 3D space with the resolution $\Delta s \in \{1.0\text{m}, 0.5\text{m}, 0.25\text{m}\}$ to investigate the effect of voxel resolution on LiDAR segmention. Since we transfer semantic occupancy prediction to LiDAR segmentation by assigning the point label based on associated voxel label, the performance of LiDAR segmention will increase with the decrease of Δs as shown in Table 6. OccNet with camera input can achieve the performance of LiDAR based method with $\Delta s \rightarrow 0$.

6. Visualization Results

We sample two scenes in the validation set and provide detailed visualization of the occupancy prediction in Figure 5, indicating that OccNet can describe the scene geometry and semantics in detail. As shown in Figure 4, we compare the rasterized occupancy with the rasterized bounding box as the input of planning module, indicating that occupancy is superior to bounding box for motion planning task.

Method	Backbone	IoU$_{geo}$	mIoU	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	driv. surf.	other flat	sidewalk	terrain	manmade	vegetation
BEVNet	ResNet50	36.11	17.37	14.02	5.07	20.85	24.94	8.64	7.75	12.8	8.93	10.21	16.02	44.41	14.42	23.87	27.76	13.73	24.49
VoxelNet	ResNet50	37.59	19.06	19.31	6.25	22.16	26.89	9.96	6.91	12.70	6.27	9.43	16.96	46.7	23.31	26.04	29.08	16.52	26.46
OccNet	ResNet50	37.69	19.48	20.63	5.52	24.16	27.72	9.79	7.73	13.38	7.18	10.68	18.00	46.13	20.60	26.75	29.37	16.90	27.21
BEVNet	ResNet101	40.15	24.62	26.39	15.79	32.07	35.83	11.93	19.72	19.75	15.38	12.82	23.90	49.16	21.52	30.57	31.39	18.99	28.71
VoxelNet	ResNet101	40.73	26.06	27.98	15.95	32.31	36.15	14.88	20.55	20.72	16.52	15.13	25.94	49.07	27.82	31.04	32.43	20.45	29.99
OccNet	ResNet101	**41.08**	**26.98**	**29.77**	**16.89**	**34.16**	**37.35**	**15.58**	**21.92**	**21.29**	**16.75**	**16.37**	**26.23**	**50.74**	**27.93**	**31.98**	**33.24**	**20.8**	**30.68**

Table 5. **Ablation in semantic scene completion with different models.** OccNet is superior to BEVNet and VoxelNet in performance.

Method	Δs(m)	mIOU	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	driv. surf.	other flat	sidewalk	terrain	manmade	vegetation
OccNet	1.00	46.60	52.78	21.04	65.94	62.45	18.31	15.49	30.71	15.82	33.94	50.22	83.93	48.84	50.52	57.89	69.49	68.29
OccNet	0.50	47.29	59.06	20.63	48.32	63.05	24.12	20.24	41.82	18.84	23.38	41.12	86.46	53.12	52.03	59.14	71.55	73.68
OccNet	0.25	53.00	65.93	22.84	64.09	72.69	32.73	28.73	52.21	17.64	22.05	51.26	89.05	57.41	58.06	64.30	75.09	73.92

Table 6. **The performance of OccNet with ResNet50 backbone on nuScenes validation set for LiDAR segmentation task.** The method with the smallest Δs show best performance.

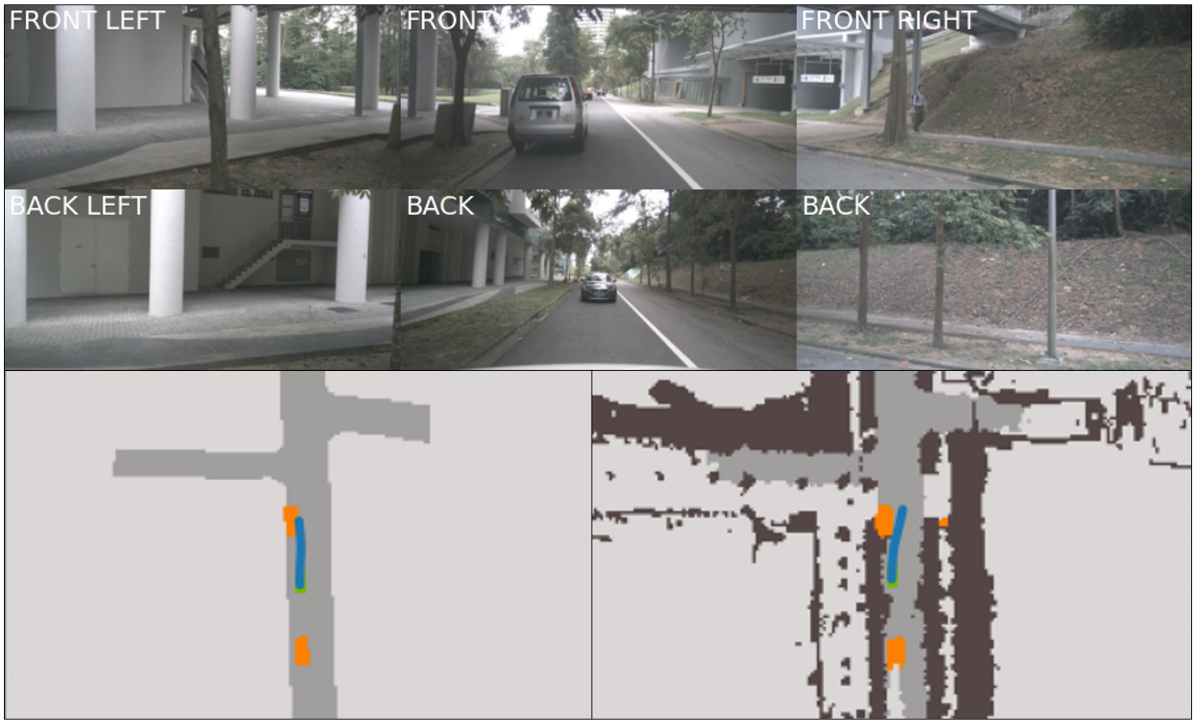

Figure 4. **Visualization of planning.** The blue line represents the planned trajectory, and the lower figures are rasterisation results of bounding box and occupancy, respectively. The trajectory obtained by the rasterized occupancy input can maintain a greater safety distance from the truck, because of the more accurate polygon representation.

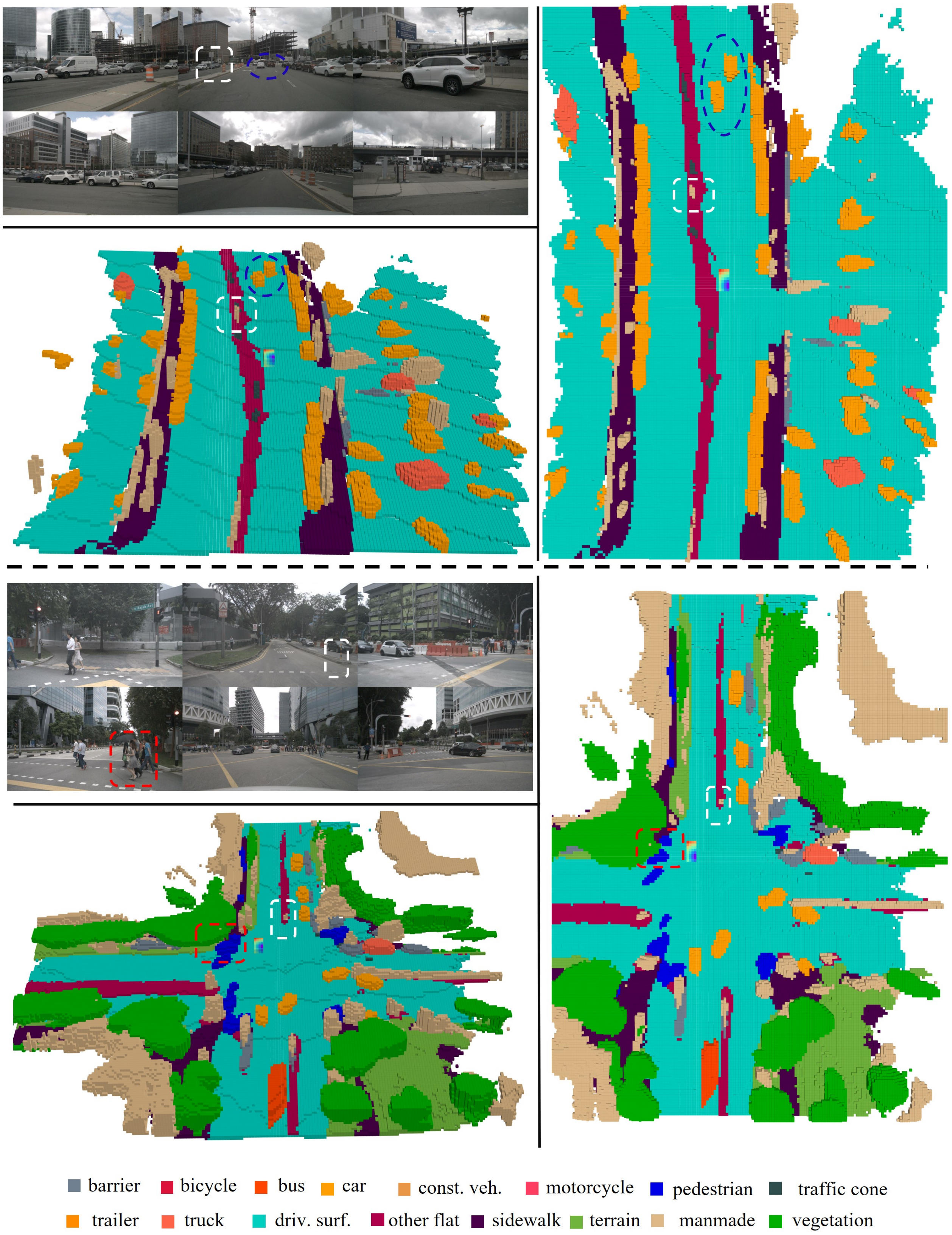

Figure 5. **Visualization of occupancy prediction.** For each scene, the top left figure is the surrounding camera input, and the left bottom figure and right figure represents the perspective view and top view of occupancy prediction result. The dashed region denotes that OccNet can predict the small size target or the distance target well.

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 11621–11631, 2020. 1, 2

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[3] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *ECCV*, pages 533–549. Springer, 2022. 1, 3

[4] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *ICRA*, pages 4628–4634. IEEE, 2022. 1

[5] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, pages 1–18. Springer, 2022. 1, 2

[6] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 1

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[8] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *ICCV*, pages 913–922, 2021. 1, 2

[9] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *CVPR*, pages 13760–13769, 2022. 1