PanoOcc: Unified Occupancy Representation for Camera-based 3D Panoptic Segmentation

Supplementary Material

A. Implementation Details

In this section, we introduce the implementation details of PanoOcc.

Image Backbone. The backbone used in our approach includes R50 [14], R101-DCN [10], and InternImage-XL [53], with output multi-scale features from FPN [30] at sizes of 1/8,1/16,1/32 and 1/64.

Voxel Queries. The initial resolution of the voxel queries is 50x50x16 for H, W, Z. We use an embedding dimension D of 256, and learnable 3D position encoding is added to the voxel queries.

Occupancy Encoder. The camera view encoder includes 3 layers, with each layer consisting of voxel self-attention, voxel cross-attention, norm layer, and feed-forward layer, with both M_1 and M_2 set to 4. The temporal encoder fuses 4 frames (including the current frame) with a time interval of 0.5s. Our key difference from previous BEV-based methods primarily lies in the learning of voxel features. We designed voxel cross-attention and voxel self-attention to facilitate the interaction between multi-scale image features and voxel queries.

- **Voxel Cross-Attention**: Specifically, for a voxel query $\mathbf{q}$ located at (i, j, k), the process of voxel cross-attention (VCA) can be formulated as follows:

$$\text{VCA}(\mathbf{q}, \mathbf{F}) = \frac{1}{|v|} \sum_{n \in v} \sum_{m=1}^{M_1} \text{DA}(\mathbf{q}, \pi_{\mathbf{n}}(\mathbf{Ref}^m_{i,j,k}), \mathbf{F}_n) \tag{5}$$

where n indexes the camera view, m indexes the reference points, and M_1 is the total number of sampling points for each voxel query. v is the set of image views for which the projected 2D point of the voxel query can fall on. $\mathbf{F}_n$ is the image features of the n-th camera view. $\pi_{\mathbf{n}}(\mathbf{Ref}^m_{i,j,k})$ denotes the m-th projected reference point in n-th camera view, projected by projection matrix $\pi_{\mathbf{n}}$ from the voxel grid located at (i, j, k). DA represents deformable attention. The real position of a reference point located at voxel grid (i, j, k) in the ego-vehicle frame is (x^m_i, y^m_j, z^m_k). The projection between m-th projected reference point $\mathbf{Ref}^m_{i,j,k}$ and its corresponding 2D reference point $(u^{n,m}_{ijk}, v^{n,m}_{ijk})$ on the n-th view can be formulate as:

$$\mathbf{Ref}^m_{i,j,k} = (x^m_i, y^m_j, z^m_k) \tag{6}$$

$$d^{n,m}_{ijk} \cdot [u^{n,m}_{ijk}, v^{n,m}_{ijk}, 1] = \mathbf{P_n} \cdot [x^m_i, y^m_j, z^m_k, 1]^T \tag{7}$$

where $\mathbf{P}_n \in \mathbb{R}^{3\times4}$ is the projection matrix of the n-th camera. $(u^{n,m}_{ijk}, v^{n,m}_{ijk})$ denotes the m-th 2D reference point on n-th image view. $d^{n,m}_{ijk}$ is the depth in the camera frame.

- **Voxel Self-Attention**: Voxel self-attention (VSA) facilitates the interaction between voxel queries. For a voxel query $\mathbf{q}$ located at (i, j, k), it only interacts with the voxel queries at the reference points nearby. The process of voxel self-attention can be formulated as follows:

$$\text{VSA}(\mathbf{q}, \mathbf{Q}) = \sum_{m=1}^{M_2} \text{DA}(\mathbf{q}, \mathbf{Ref}^m_{i,j,k}, \mathbf{Q}) \tag{8}$$

where m indexes the reference points, and M_2 is the total number of reference points for each voxel query. DA represents deformable attention. Contrary to the reference points on the image plane in voxel cross-attention, $\mathbf{Ref}^m_{i,j,k}$ in voxel self-attention is defined on the BEV plane.

$$\mathbf{Ref}^m_{i,j,k} = (x^m_i, y^m_j, z_k) \tag{9}$$

where (x^m_i, y^m_j, z_k) denotes the m-th reference point for query $\mathbf{q}$. These sampling points share the same height z_k, but with different learnable offsets for (x^m_i, y^m_j). This encourages the voxel queries to interact in the BEV plane, which contains more semantic information.

Occupancy Decoder. The voxel upsample module employs 3 layers of 3D deconvolutions to upscale 4x for H and W, and 2x for Z, with detailed parameters in the Table 9. The upsampled voxel features have dimensions of 200x200x32 for H', W', Z', and a feature dimension D' of 64.

Task Head. The segmentation head has 2 MLP layers with a hidden dimension of 128 and uses *softplus* [66] as the activation function. The number of object queries for the detection head is set to 900, and has 6 layers decoder, similar to [26].

B. Test Set Performance

3D Semantic Segmentation. In Table 8, we adopt the R101-DCN [10] initialized from FCOS3D [52] checkpoint,

Method	Input Modality	mIoU	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. suf.	other flat	sidewalk	terrain	manmade	vegetation
MINet [22]	LiDAR	56.3	54.6	8.2	62.1	76.6	23.0	58.7	37.6	34.9	61.5	46.9	93.3	56.4	63.8	64.8	79.3	78.3
PolarNet [64]	LiDAR	69.4	72.2	16.8	77.0	86.5	51.1	69.7	64.8	54.1	69.7	63.5	96.6	67.1	77.7	72.1	87.1	84.5
PolarSteam [6]	LiDAR	73.4	71.4	27.8	78.1	82.0	61.3	77.8	75.1	72.4	79.6	63.7	96.0	66.5	76.9	73.0	88.5	84.8
JS3C-Net [59]	LiDAR	73.6	80.1	26.2	87.8	84.5	55.2	72.6	71.3	66.3	76.8	71.2	96.8	64.5	76.9	74.1	87.5	86.1
AMVNet [32]	LiDAR	77.3	80.6	32.0	81.7	88.9	67.1	84.3	76.1	73.5	84.9	67.3	97.5	67.4	79.4	75.5	91.5	88.7
SPVNAS [47]	LiDAR	77.4	80.0	30.0	91.9	90.8	64.7	79.0	75.6	70.9	81.0	74.6	97.4	69.2	80.0	76.1	89.3	87.1
Cylinder3D++ [70]	LiDAR	77.9	82.8	33.9	84.3	89.4	69.6	79.4	77.3	73.4	84.6	69.4	97.7	70.2	80.3	75.5	90.4	87.6
AF2S3Net [7]	LiDAR	78.3	78.9	52.2	89.9	84.2	77.4	74.3	77.3	72.0	83.9	73.8	97.1	66.5	77.5	74.0	87.7	86.8
DRINet++ [63]	LiDAR	80.4	85.5	43.2	90.5	92.1	64.7	86.0	83.0	73.3	83.9	75.8	97.0	71.0	81.0	77.7	91.6	90.2
LidarMultiNet [62]	LiDAR	81.4	80.4	48.4	94.3	90.0	71.5	87.2	85.2	80.4	86.9	74.8	97.8	67.3	80.7	76.5	92.1	89.6
TPVFormer [18]	Camera	69.4	74.0	27.5	86.3	85.5	**60.7**	68.0	62.1	49.1	81.9	68.4	94.1	59.5	66.5	63.5	83.8	79.9
OccFormer [65]	Camera	70.8	72.8	29.9	87.9	**85.6**	57.1	74.9	63.2	53.4	**83.0**	67.6	94.8	61.9	70.0	66.0	**84.0**	**80.5**
PanoOcc(Ours)	Camera	**71.4**	**82.5**	**32.3**	**88.1**	83.7	46.1	**76.5**	**67.6**	**53.6**	82.9	**69.5**	**96.0**	**66.3**	**72.3**	**66.3**	80.5	77.3

Table 8. **LiDAR semantic segmentation results on nuScenes test set.** Our method achieves new state-of-the-art performance on camera-based semantic segmentation. For a fair comparison, we use the same backbone R101-DCN and train for 24 epochs.

Hyperparameters	Values
#Input features	50x50x16x256 (H,W,Z,D)
#Output features	200x200x32x64 (H',W',Z',D')
ConvTranspose3D#1	kernel:(1,5,5), stride:(1,1,1)
ConvTranspose3D#2	kernel:(1,4,4), stride:(1,2,2)
ConvTranspose3D#3	kernel:(2,4,4), stride:(2,2,2)
Activate function	ReLU
Normalize	BN3D

Table 9. **Network hyper-parameters of voxel upsample module.**

the same setting as TPVFormer [18] and OccFormer [65]. Without bells and whistles, our PanoOcc surpasses all previous camera-based methods.

C. Ablation Studies on Model Design

Initial Voxel Resolution. Table 10 compares the results of different initial resolutions used for voxel queries in our experiments. In experiments (b), (c), and (d), we maintained fixed dimensions of H and W while varying the resolution of Z. Our findings clearly demonstrate that encoding height information is a crucial factor in achieving superior performance in both segmentation(+5.3 mIoU) and detection tasks(+1.2 mAP and +1.6 NDS), with a more significant impact observed in segmentation tasks. Furthermore, we observed that (a) and (b) have the same number of query parameters and perform similarly in detection tasks. However, there is a significant gap in the segmentation tasks between these two. Specifically, the mIoU gain from (d) to (a) is much less compared to that from (d) to (b). The experiment (e) results suggest that when the dimensions of H and W are too small, there will be a significant reduction in the performance of both detection and segmentation tasks. Overall, our findings emphasize the importance of encoding height information to achieve fine-grained scene understanding.

	Query Resolution	mIoU	mAP	NDS
(a)	100x100x4	0.617	**0.276**	**0.327**
(b)	50x50x16	**0.661**	0.271	0.324
(c)	50x50x8	0.631	0.267	0.316
(d)	50x50x4	0.608	0.259	0.308
(e)	25x25x16	0.591	0.244	0.294

Table 10. **Ablation study for different initial query resolutions.** Height information is important to achieve fine-grained 3D scene understanding.

Design of Camera View Encoder. Table 11 presents the ablation study conducted on the design choices in the camera view encoder. Specifically, we experimented with different combinations of attention modules in (b), (c), and (d). The results demonstrated that incorporating voxel self-attention (VSA) enhanced the interaction between queries, leading to improved performance. Considering both performance and parameters, we choose 3 layers as default.

Design of Temporal Encoder. Table 12 presents extensive ablation studies on the design of the temporal encoder,

	Layers	Attention module	mIoU	mAP	NDS
(a)	1	VSA + VCA	0.648	0.251	0.294
(b)	3	VCA	0.644	0.264	0.312
(c)	3	VSA + VCA	0.653	0.267	0.314
(d)	3	VSA×2 + VCA	0.661	**0.271**	**0.324**
(e)	6	VSA×2 + VCA	**0.662**	0.267	0.319

Table 11. **Ablation study for camera view encoder.** VSA denotes voxel self-attention, while VCA means voxel cross-attention.

including different time intervals, number of frames, fusion methods, and encoder network architectures. Compared to (a) and (b) designs, both detection and segmentation tasks show a significant improvement (+2.5 mIoU, +2.4 mAP, and +7.1 NDS), which suggests the importance of temporal information. In (b)(c)(d), we compared the influence of different time intervals and found that longer intervals do not improve the fine-grained segmentation performance. In (e) and (f), we also compared different ways to fuse the historical features and found that directly concatenating the features performs better than using temporal self-attention [26].

	Temp.	Intv.	Frames	Fuse	Arch.	mIoU	mAP	NDS
(a)		/	1	/	C3D×1	0.656	0.269	0.319
(b)	✓	0.5s	4	Cat.	C3D×1	**0.681**	0.293	**0.390**
(c)	✓	1s	4	Cat.	C3D×1	0.657	**0.294**	0.385
(d)	✓	2s	4	Cat.	C3D×1	0.660	0.294	0.375
(e)	✓	1s	4	Cat.	C3D×3	0.658	0.290	0.379
(f)	✓	0.5	4	TSA	DA	0.648	0.271	0.323

Table 12. **Ablation study for temporal encoder.** Temp. stands for temporal fusion, while ✓denotes using temporal fusion. Intv. denotes time interval. Arch. refers to the architecture used in temporal encoder. C3D represents 3D convolution. ×3 means using 3 blocks of the architecture. Cat. means concatenating features from different frames, and TSA represents the temporal self-attention structure in [26]. DA means deformable attention [69].

The Supervision for Voxel Representation. Table 13 ablates the effects of different resolutions for segmentation loss supervision. The experiment results indicate that resolution at $400 \times 400 \times 64$ has the best performance.

Loss Terms and Weights. Table 14 presents the comparison of various combinations of loss terms and weights. It indicates that the $\mathcal{L}_{lovasz}$ plays a crucial role in the segmentation learning process, as its removal led to a significant drop in performance (from 65.6 to 59.6 mIoU). We also

Supervision	Voxel feats	Loss Resolution	mIoU	mAP	NDS
LiDAR	200x200x32	400x400x64	**0.661**	**0.271**	**0.324**
LiDAR	200x200x32	200x200x32	0.644	0.267	0.316
LiDAR	100x100x16	100x100x16	0.609	0.264	0.317

Table 13. **Supervision for voxel representation.** We utilize sparse LiDAR point labels as the supervision for voxel representation.

$\mathcal{L}_{focal}$	$\mathcal{L}_{lovasz}$	$\mathcal{L}_{thing}$	λ_1	λ_2	λ_3	mIoU	mAP	NDS
✓			10.0	/	/	0.596	0.259	0.315
✓	✓		10.0	10.0	/	0.656	0.266	0.319
	✓	✓	/	10.0	5.0	0.643	0.260	0.311
✓	✓	✓	10.0	10.0	5.0	**0.661**	**0.271**	**0.324**
✓	✓	✓	10.0	10.0	10.0	0.652	0.265	0.317
✓	✓	✓	5.0	10.0	5.0	0.656	0.266	0.315
✓	✓	✓	15.0	10.0	5.0	0.650	0.265	0.314
✓	✓	✓	10.0	15.0	5.0	0.654	0.263	0.312

Table 14. **Ablation for loss terms and weights.** We ablates different loss combinations and its weight.

experimented with various weight combinations and found that $\lambda_1 = 10, \lambda_2 = 10, \lambda_3 = 5$ performs best.

Temporal Enhancement. In Table 15, we compared the impact of temporal information on different categories. The findings revealed that the semantic segmentation performance improved for almost all categories except for the barrier category. The motorcycle and trailer categories demonstrated a significant improvement, with a boost of 11.7 mIoU and 8.2 mIoU, respectively. These two categories are typically affected by occlusion, and thus, the utilization of temporal information can enhance the model's ability to accurately detect and segment occluded objects.

D. Training and Inference Details

Training. We trained the model on 8 NVIDIA A100 GPUs with a batch size of 1 per GPU. Throughout training, we employed the AdamW optimizer [36] for 24 epochs, starting with an initial learning rate of 3×10^{-4} and following a step schedule at epochs 20 and 23. Additionally, we employed several data augmentation techniques, including image scaling, color distortion, and Gridmask [5]. The input image size is cropped to 640×1600. When using the R101-DCN [10] or InternImage [53] as the backbone, we default to the 1.0 image scale (640×1600). However, when using the R50 [14] backbone, we adopt a 0.5 image scale

mIoU	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. suf.	other flat	sidewalk	terrain	manmade	vegetation
65.6	72.3	35.8	91.4	84.4	47.2	52.6	57.7	31.5	55.6	80.6	94.0	64.3	63.2	66.5	77.7	73.9
68.1	70.7	37.9	92.3	85.0	50.7	64.3	59.4	35.3	63.8	81.6	94.2	66.4	64.8	68.0	79.1	75.6
(2.5↑)	(1.6↓)	(2.1↑)	(0.9↑)	(0.6↑)	(3.5↑)	(11.7↑)	(1.7↑)	(3.8↑)	(8.2↑)	(1.0↑)	(0.2↑)	(2.1↑)	(1.6↑)	(1.5↑)	(1.4↑)	(1.7↑)

Table 15. **Effect of temporal enhancement on different categories.** The findings indicated that incorporating temporal information improved segmentation performance for most categories.

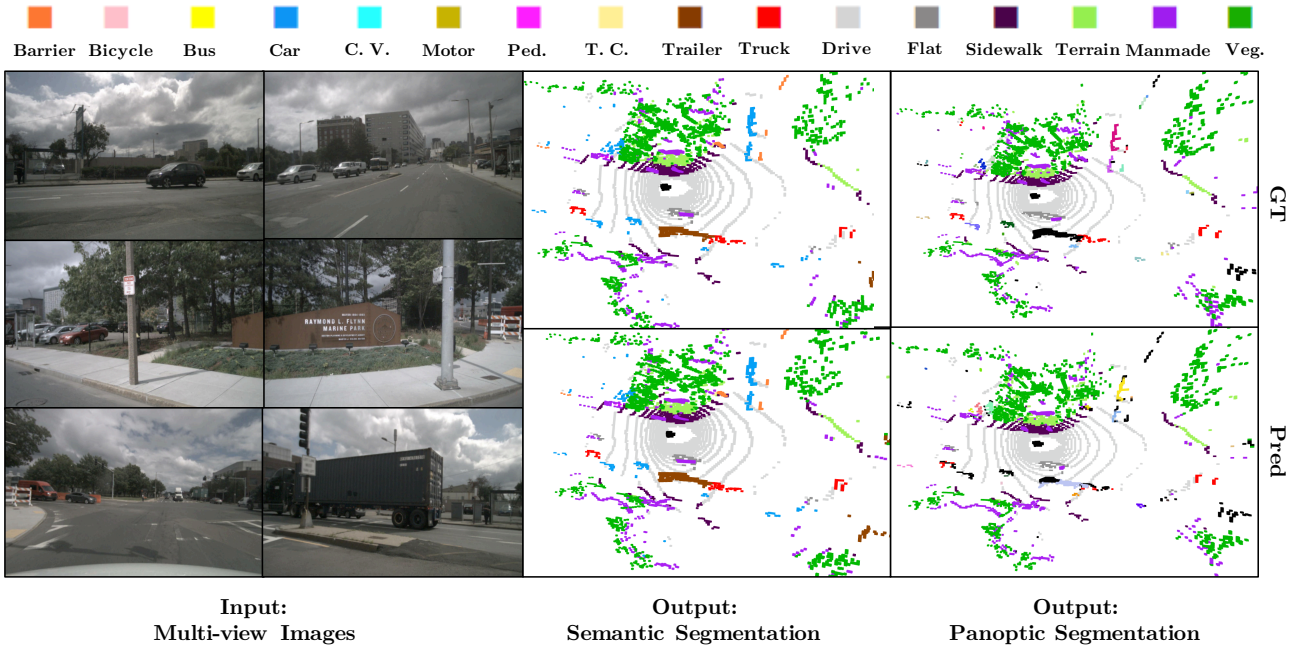

Figure 5. **Qualitative results on nuScenes validation set.** Our PanoOcc takes multi-view images as input and produces voxel predictions, which are visualized at a resolution of 200 × 200 × 32. We evaluate 3D segmentic segmentation and panoptic segmentation on LiDAR points.

(320 × 800). The loss weights used in our approach are λ_1=10.0, λ_2=10.0, λ_3=5.0, λ_4=2.0,and λ_5=0.25.

Supervision. For the detection head, we use object-level annotations as the supervision. We employ sparse LiDAR point-level semantic labels for the segmentation head to supervise voxel prediction. When multiple semantic labels are present within a voxel grid, we prioritize the category label with the highest count of LiDAR points. As for the occupancy prediction, we rely on the occupancy label as the source of supervision.

E. Visualization

Figure 5 showcases qualitative results achieved by PanoOcc on the nuScenes validation set. The voxel predictions are visualized at a resolution of 200 × 200 × 32 and assign to LiDAR points. These visualizations highlight the accuracy and reliability of our predictions for 3D semantic segmentation and panoptic segmentation.

F. Reproducibility Statements

We are committed to providing the research community with the necessary resources to replicate our work. We will release the training and inference codes, accompanied by well-documented instructions to facilitate the replication process. Our codebase is built upon mmdetection3D[2], ensuring that it is user-friendly and accessible to the wider community. The data and annotations of nuScenes[3] are publicly available.

[2] https://github.com/open-mmlab/mmdetection3d
[3] https://nuscenes.org