OccFormer: Dual-path Transformer for Vision-based 3D Semantic Occupancy Prediction

Yunpeng Zhang
PhiGent Robotics
yunpengzhang97@gmail.com

Zheng Zhu*
PhiGent Robotics
zhengzhu@ieee.org

Dalong Du
PhiGent Robotics
dalong.du@phigent.ai

Abstract

The vision-based perception for autonomous driving has undergone a transformation from the bird-eye-view (BEV) representations to the 3D semantic occupancy. Compared with the BEV planes, the 3D semantic occupancy further provides structural information along the vertical direction. This paper presents OccFormer, a dual-path transformer network to effectively process the 3D volume for semantic occupancy prediction. OccFormer achieves a long-range, dynamic, and efficient encoding of the camera-generated 3D voxel features. It is obtained by decomposing the heavy 3D processing into the local and global transformer pathways along the horizontal plane. For the occupancy decoder, we adapt the vanilla Mask2Former for 3D semantic occupancy by proposing preserve-pooling and class-guided sampling, which notably mitigate the sparsity and class imbalance. Experimental results demonstrate that OccFormer significantly outperforms existing methods for semantic scene completion on SemanticKITTI dataset and for LiDAR semantic segmentation on nuScenes dataset. Code is available at https://github.com/zhangyp15/OccFormer.

1. Introduction

The accurate perception of 3D surroundings constitutes the foundation of modern autonomous driving systems. Though LiDAR-based methods [23, 44, 62, 43, 39, 53], with explicit depth measurements, have been dominating the leading performance on public datasets [15, 3, 46, 2], vision-based approaches still offer advantages in terms of cost-effectiveness, stability, and generality. The past years have witnessed the prosperity of Bird-Eye-View representations for vision-based 3D perception. With the multi-view camera images as input, various attempts for 2D-to-3D transformation [38, 30, 19, 28] have been proposed for applications including 3D object detection [19, 30, 33], semantic map construction [38, 60, 42, 37], and motion prediction [18, 1, 58]. Considering these tasks require either rigid bounding boxes or BEV-oriented predictions, the collapse of 3D scenes into 2D ground planes has demonstrated an excellent trade-off between performance and efficiency. However, the holistic understanding of the 3D scene, especially for real-world obstacles with variable shapes, can hardly be recovered with the condensed BEV feature maps. To this end, this paper focuses on building a fine-grained 3D representation, namely 3D semantic occupancy, for the surrounding environment with multi-view images.

The task of 3D semantic occupancy prediction aims to reconstruct the surrounding 3D environment with fine-grained geometry and semantics, which is also known as 3D semantic scene completion when the LiDAR point cloud is taken as input. For the driving scenes, most existing methods [43, 11, 8, 25, 50] still rely on the expensive LiDAR sensors for explicit depth measurements. The seminar work MonoScene [4] proposed the first monocular framework for 3D semantic occupancy prediction. It first constructs the 3D feature with sight projection and then processes it with a classical 3D UNet. However, the 3D convolution suffers from several limitations. First, it reasons the semantics within a relatively fixed receptive field, while different semantic classes may distribute following various patterns. Also, its spatial invariance cannot well process the sparse and discontinuous 3D features, generated from the state-of-the-art practices for image-to-3D transformation [38, 19, 28]. Finally, the 3D convolution filters can consume massive parameters. Therefore, we believe a long-range, dynamic, and efficient method for encoding 3D features is needed to pave the way.

Inspired by the widespread success of vision transformers [13, 34] in various vision tasks [5, 59, 16, 54, 34, 29], we are motivated to utilize the attention mechanism for building the encoder-decoder network for 3D semantic occupancy prediction. For the encoder part, we propose the dual-path transformer block to unleash the capacity of self-attention while limiting the quadratic complexity. Specifically, the local path operates along each 2D BEV slice with the shared windowed attention to capture the fine-grained

[*] Corresponding author.

details, while the global path performs on the collapsed BEV feature to obtain scene-level understanding. Finally, the dual-path outputs are adaptively fused to generate the output 3D feature volume. The dual-path designs appropriately break down the challenging processing of 3D feature volumes and we demonstrate its clear advantage over the classic 3D convolutions. For the decoder part, we are the first to adapt the state-of-the-art method Mask2Former [9] for 3D semantic occupancy prediction. We further propose to use max-pooling rather than the default bilinear for computing the masked regions for attention, which can better preserve the minor classes. Additionally, the class-guided sampling is proposed to capture the foreground areas for more effective optimization. Experimental results demonstrate the superiority of OccFormer over existing state-of-the-art methods. For 3D semantic scene completion on SemanticKITTI [2] dataset, OccFormer outperforms MonoScene by 1.24% mIoU, which makes an 11% relative improvement and ranks first on the test leaderboard among all monocular methods. We also evaluate OccFormer on nuScenes [3] dataset for LiDAR semantic segmentation, following TPVFormer [20]. Our method surpasses TPVFormer by 1.4% mIoU and generates more complete and realistic predictions for 3D semantic occupancy prediction.

2. Related Work

2.1. Camera-based BEV Perception

Considering the dimension gap between the 2D image input and the 3D prediction, recent studies for vision-based 3D perception first construct the BEV feature representations and then perform various downstream tasks on the BEV space [19, 28, 30, 37, 58, 38, 60, 40, 18, 1, 42]. To transform the perspective image features into the BEV features, LSS [38] and its follow-ups [40, 28, 18, 58] predict the pixel-wise depth distribution to project the image features into 3D points, which are then voxelized into the BEV features. Other methods like BEVFormer [30] utilize the deformable attention [61, 48] to update the BEV queries with corresponding image features. In this paper, we extend the BEV-based perception to 3D semantic occupancy prediction, which further contains the structural information along the height dimension.

2.2. 3D Semantic Occupancy Prediction

Since 3D semantic occupancy prediction is also known as 3D semantic scene completion (SSC), we also review the related SSC methods. SSCNet [45] first proposes the problem of semantic scene completion, which jointly reasons the geometry and semantics. The follow-ups usually employ the geometrical inputs with explicit depth information [43, 25, 50, 11, 8, 41]. Recently, MonoScene [4] builds the first monocular method for semantic scene completion, which employs the 3D UNet to process the voxel features generated by sight projection. TPVFormer [20] proposes the tri-perspective view representation to describe the 3D scene for semantic occupancy prediction. Despite its simplicity, the tri-plane format is susceptible to the deficiency of fine-grained semantic information, leading to inferior performance. In this paper, we re-advocate the representation power of dense 3D features and propose the transformer-based encoder-decoder network for 3D semantic occupancy prediction.

2.3. Efficient 3D Network

On the field of 3D semantic scene completion, extensive attempts have been proposed to improve the efficiency of 3D networks. EsscNet [55] partitions the non-empty voxels into different groups and conduct 3D sparse convolution within each group. DDRNet [26] replaces the 3D convolution with three consecutive 1D convolution layers along each dimension. AIC-Net [25] further equips each 1D layer with various kernel sizes for anisotropic processing. LMSCNet [43] uses the 2D UNet to process the collapsed BEV features and finally expands the height dimension for 3D segmentation. S3CNet [11] turns to the sparse convolution for outdoor point clouds. These methods are mostly targeted for LiDAR points with convolutional structures. In this paper, we propose the dual-path transformer to efficiently process the camera-generated 3D feature volumes with transformer-based modules.

3. Approach

3.1. Overview

The overall pipeline of OccFormer is illustrated in Fig. 1. With the monocular image or multi-camera images as the input, the multi-scale features are first extracted by the image encoder, and then lifted to 3D feature volume, which are briefly introduced in the following paragraphs. The 3D feature is further processed by the dual-path transformer encoder (Sec. 3.2) to produce multi-scale voxel features with local and global semantics. Finally, the transformer occupancy decoder (Sec. 3.3) fuses multi-scale features and formulates the occupancy prediction as the transformer-based mask classification for decoding.

Image Encoder. The image encoder aims to extract geometric and semantic features within the perspective view, which provides the foundation of the later-generated 3D feature volume. The image encoder consists of a backbone network for extracting multi-scale features and a neck for further fusion. The output of the image encoder is one fused feature map with $\frac{1}{16}$ of the input resolution. We use $\mathbf{F}^{2d} \in \mathbb{R}^{N \times C \times H \times W}$ to represent the extracted features, where N is the number of camera views, C is the channel number, and (H, W) refers to the resolution.

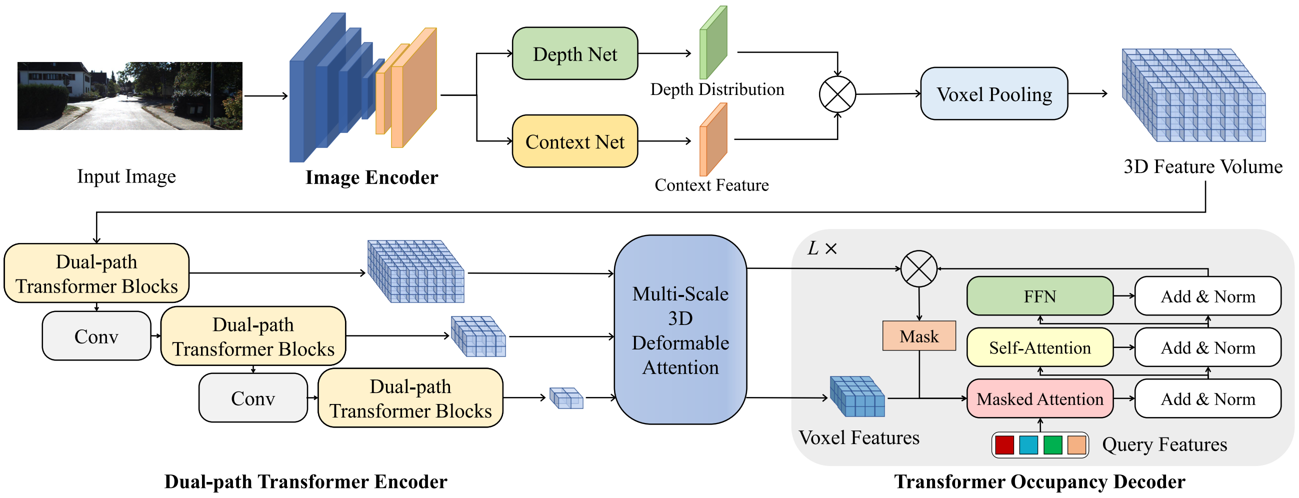

Figure 1: The framework of the proposed OccFormer for camera-based 3D semantic occupancy prediction. The pipeline consists of the image encoder for extracting multi-scale 2D features, the image-to-3D transformation for lifting the 2D features to 3D volumes, and the transformer-based encoder-decoder for obtaining 3D semantic features and predicting the 3D semantic occupancy.

Image-to-3D Transformation. Inspired by recent studies on lifting multi-view images to the Bird-Eye-View representations [38, 19, 28, 30], we extend the LSS [38] paradigm for image-to-3D transformation. Specifically, the encoded image features $\mathbf{F}^{2d}$ are processed to generate the context feature $\mathbf{F}^{2d}_{con} \in \mathbb{R}^{N \times C_{con} \times H \times W}$ and the discrete depth distribution $\mathbf{D} \in \mathbb{R}^{N \times D \times H \times W}$. Then the outer product $\mathbf{F}^{2d}_{con} \otimes \mathbf{D}$ is employed to create the point cloud representation $\mathbf{P} \in \mathbb{R}^{NDHW \times C_{con}}$. Finally, the voxel-pooling is conducted to create the 3D feature volume $\mathbf{F}^{3d} \in \mathbb{R}^{C_{con} \cdot X \cdot Y \cdot Z}$, where (X, Y, Z) denotes the resolution of the 3D volume.

3.2. Dual-path Transformer Encoder

To pursue long-range, dynamic, and efficient processing of the 3D feature volumes, we propose the dual-path transformer block to build the 3D encoder. Inspired by recent advances that introduce locality into the transformer [49, 56, 24], we also design the encoder as a hybrid structure. The encoder consists of a series of dual-path transformer blocks, while one 3D convolution layer is inserted between two consecutive blocks to introduce locality and optionally perform the downsampling. The detailed structure of the dual-path transformer block is shown in Fig. 2. With the input 3D feature, the local and global pathways first aggregate semantic information along the horizontal direction in parallel. Next, the dual-path outputs are fused through the sigmoid-weighted summation. Finally, the skip connection is applied to ensure the residual learning [17]. We introduce the dual-path processing with more details in the following paragraph.

The local path is mainly targeted to extract the fine-grained semantic structures. Since the horizontal direction contains the most variations, we believe the parallel processing of all BEV slices with one shared encoder is able to keep most of the semantic information. Specifically, we merge the height dimension into the batch dimension and employ the windowed self-attention [34] as the local feature extractor, which can dynamically attend to long-range regions with moderate computations. On the other hand, the global path aims to efficiently capture the scene-level semantic layouts. To this end, the global path starts by getting the BEV feature by average pooling along the height dimension. The same windowed self-attention from the local path is utilized to process the BEV feature for neighbouring semantics. Since we find the global self-attention on the BEV plane can consume excessive memories, the ASPP [6] is applied instead to capture the global contexts. In practice, we employ the bottleneck structure [17] to reduce the channel number by 4× for ASPP. Finally, the scene-level information from the global path is propagated to the entire 3D volume from the local path. Assume the dual-path outputs are $\mathbf{F}_{\text{local}} \in \mathbb{R}^{C \cdot X \cdot Y \cdot Z}$ and $\mathbf{F}_{\text{global}} \in \mathbb{R}^{C \cdot X \cdot Y}$, the combined output $\mathbf{F}_{\text{out}}$ is computed as:

$$\mathbf{F}_{\text{out}} = \mathbf{F}_{\text{local}} + \sigma(\mathbf{W}\mathbf{F}_{\text{local}}) \cdot \text{unsqueeze}(\mathbf{F}_{\text{global}}, -1) \quad (1)$$

where $\mathbf{W}$ refers to the FFN for generating the aggregation weights along the height dimension, $\sigma(\cdot)$ is the sigmoid function, and "unsqueeze" expands the global 2D feature along the height. Although the dual-path processing only performs 2D reasoning along the horizontal direction, their combination effectively aggregates essential informa-

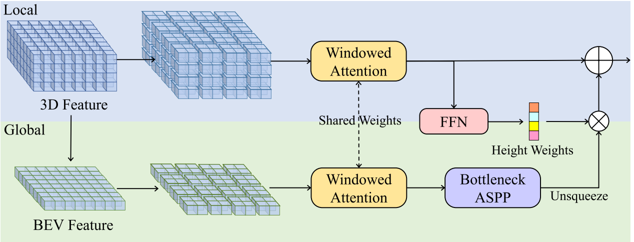

Figure 2: Illustration of the dual-path transformer block. The local path processes the 3D feature by applying the shared windowed attention to each horizontal slice, while the global path operates on the collapsed BEV feature for scene-level semantic layouts. The dual-path outputs are finally fused through the weighted summation. The skip connection is omitted.

tion for semantic reasoning, including local semantic structures and global semantic layouts. Additionally, the dual-path transformer encoder has fewer parameters and requires less computation than classic 3D convolutions, benefiting from shared modules and mostly 2D reasoning.

3.3. Transformer Occupancy Decoder

Inspired by the recent mask classification models [10, 9] for image segmentation, we also formulate the 3D semantic occupancy as predicting a set of binary 3D masks associated with corresponding class labels. Following Mask2Former [9], our transformer occupancy decoder includes the pixel decoder (Sec. 3.3.1) for per-voxel embeddings and the transformer decoder (Sec. 3.3.2) for per-query embeddings and class predictions. The final mask predictions are derived from the dot product between these two embeddings. Also, we introduce two essential modifications to effectively improve the occupancy predictions, including the preserve-pooling (Sec. 3.3.3) and the class-guided sampling (Sec. 3.3.4). Formally, the input multi-scale feature volumes from the transformer encoder are denoted as $\{\mathbf{F}_i^{3d} \in \mathbb{R}^{C_i \cdot X_i \cdot Y_i \cdot Z_i}\}_{i=1}^{N_l}$, where N_l is the level number, C_i is the channel number, and (X_i, Y_i, Z_i) is the volume size.

3.3.1 Pixel Decoder

With multi-scale 3D features as input, the pixel decoder is tasked with aggregating multi-level semantics and creating high-resolution voxel embeddings. Since each feature level places different emphasis on low-level details and high-level semantics, we employ the multi-scale deformable attention [61], tailored for 3D, to facilitate effective intra-scale and inter-scale interactions. Take the level-i feature $\mathbf{F}_i^{3d}$ as an example, its corresponding real-world coordinates $\mathbf{P}_i^{3d} \in \mathbb{R}^{X_i \cdot Y_i \cdot Z_i \cdot 3}$ are first computed. Then the features are processed to create the sampling offsets $\mathbf{\Delta}_j^{3d}$ and the attention weights $\mathbf{W}_j^{3d}$ for all levels $j = 1, \cdots, N_l$. Finally, the updating process is formulated as in Eq. (2):

$$\mathbf{F}_i^{3d} = \mathbf{F}_i^{3d} + \sum_{j=1}^{N_l} \left[\mathbf{W}_j^{3d} \mathbf{F}_j^{3d} \left(\mathbf{P}_i^{3d} + \mathbf{\Delta}_j^{3d}\right)\right] \tag{2}$$

where $\mathbf{F}^{3d}\left(\mathbf{P}^{3d} + \mathbf{\Delta}^{3d}\right)$ conducts the trilinear feature sampling at the corresponding positions. With the above interactions, each processed feature volume is enhanced by the multi-scale semantic information, which facilitates the following transformer decoder. The feature volume with the highest resolution is projected to generate the per-voxel embeddings $\mathcal{E}_{\text{voxel}} \in \mathbb{R}^{C_{\mathcal{E}} \cdot X \cdot Y \cdot Z}$, where $C_{\mathcal{E}}$ is the embedding dimension.

3.3.2 Transformer Decoder

With the input multi-scale voxel features and the parameterized query features, the transformer decoder performs an iterative updating of the query features towards the desired class segments, as shown in Fig. 1. Within each iteration layer l, the queries features $\mathbf{Q}_l$ first attends to their corresponding foreground regions through the masked attention:

$$\mathbf{Q}_{l+1} = \text{softmax}\left[\mathcal{M}_{l-1} + \mathbf{W}_q \mathbf{Q}_l \left(\mathbf{W}_k \mathbf{F}_l^{3d}\right)^T\right] \mathbf{W}_v \mathbf{F}_l^{3d} + \mathbf{Q}_l \tag{3}$$

where $\mathbf{F}_l^{3d}$ is the 3D voxel feature, $\mathcal{M}_{l-1}$ is the attention mask from the previous layer, and $(\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v)$ are linear projection layers. The self-attention is then conducted to exchange context information, followed by the FFN for feature projection. At the end of each iteration, each query feature $\mathbf{q}_i$ is projected to predict its semantic logits $\mathbf{p}_i$ and the mask embedding $\mathcal{E}_{\text{mask}_i}$. The latter is further transformed into the binary 3D mask $\mathbf{M}_i$ by a dot product with the per-voxel embeddings $\mathcal{E}_{\text{voxel}}$ and a sigmoid function. The final 3D semantic occupancy prediction $\mathbf{Y}$ is formulated as:

$$\mathbf{Y} = \sum_{i=1}^{N_q} \mathbf{p}_i \cdot \mathbf{M}_i \tag{4}$$

where N_q is the number of query features.

3.3.3 Preserve-Pooling

When converting the high-resolution mask predictions into the low-resolution attention masks for the next iteration, Mask2Former [9] employs the bilinear interpolation for downsampling. The operation is sufficient to protect the local structures because the image segmentation masks are more complete and contiguous. However, we found its trivial adaptation, namely trilinear interpolation, cannot well handle the 3D semantic occupancy prediction. Since the LiDAR-generated segmentation masks for 3D objects are usually partial and sparse, the trilinear downsampling can remove the local structures or even the entire objects. To

Table 1: **Semantic scene completion results on SemanticKITTI test set.** * represents these methods are adapted for the RGB inputs, which are implemented and reported in MonoScene [4]. Our method outperforms all existing monocular methods for semantic scene completion in both the SC IoU and the SSC mIoU.

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* [43]	Camera	31.38	7.07	46.70	19.50	13.50	3.10	10.30	14.30	0.30	0.00	0.00	0.00	10.80	0.00	10.40	0.00	0.00	0.00	5.40	0.00	0.00
3DSketch* [8]	Camera	26.85	6.23	37.70	19.80	0.00	0.00	12.10	17.10	0.00	0.00	0.00	0.00	12.10	0.00	16.10	0.00	0.00	0.00	3.40	0.00	0.00
AICNet* [25]	Camera	23.93	7.09	39.30	18.30	19.80	1.60	9.60	15.30	0.70	0.00	0.00	0.00	9.60	1.90	13.50	0.00	0.00	0.00	5.00	0.10	0.00
JS3C-Net* [50]	Camera	34.00	8.97	47.30	21.70	19.90	2.80	12.70	20.10	0.80	0.00	0.00	4.10	14.20	3.10	12.40	0.00	0.20	0.20	8.70	1.90	0.30
MonoScene [4]	Camera	34.16	11.08	54.70	27.10	24.80	5.70	14.40	18.80	3.30	0.50	0.70	**4.40**	14.90	2.40	19.50	1.00	1.40	**0.40**	11.10	3.30	2.10
TPVFormer [20]	Camera	34.25	11.26	55.10	27.20	27.40	**6.50**	14.80	19.20	**3.70**	1.00	0.50	2.30	13.90	2.60	20.40	1.10	**2.40**	0.30	11.00	2.90	1.50
OccFormer (ours)	Camera	**34.53**	**12.32**	**55.90**	**30.30**	**31.50**	**6.50**	**15.70**	**21.60**	1.20	**1.50**	**1.70**	3.20	**16.80**	**3.90**	**21.30**	**2.20**	1.10	0.20	**11.90**	**3.80**	**3.70**

Table 2: **Semantic scene completion results on SemanticKITTI [2] validation set.** * represents these methods are adapted for the RGB inputs, which are implemented and reported in MonoScene [4]. † represents the reproduced result from [20].

Method	SSC Input	SC	SSC																		
		IoU	road (15.30%) ■	sidewalk (11.13%) ■	parking (1.12%) ■	other-ground (0.56%) ■	building (14.1%) ■	car (3.92%) ■	truck (0.16%) ■	bicycle (0.03%) ■	motorcycle (0.03%) ■	other-vehicle (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%) ■	traffic-sign (0.08%) ■	mIoU
LMSCNet* [43]	$\hat{x}^{\text{occ}}_{\text{3D}}$	28.61	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	6.70
3DSketch* [8]	x^{rgb},$\hat{x}^{\text{TSDF}}$	33.30	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	7.50
AICNet* [25]	x^{rgb},$\hat{x}^{\text{depth}}$	29.59	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	8.31
JS3C-Net* [50]	$\hat{x}^{\text{pts}}$	**38.98**	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	10.31
MonoScene† [4]	x^{rgb}	36.86	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	11.08
TPVFormer [20]	x^{rgb}	35.61	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	11.36
OccFormer	x^{rgb}	36.50	**58.85**	**26.88**	19.61	0.31	14.40	**25.09**	**25.53**	**0.81**	**1.19**	**8.52**	**19.63**	3.93	**32.62**	**2.78**	**2.82**	0.00	5.61	**4.26**	**2.86**	**13.46**

this end, we propose the preserve-pooling by simply using the max-pooling for downsampling the attention masks. Despite a minor modification, we demonstrate its effectiveness in the ablation studies (Sec. 4.5).

3.3.4 Class-Guided Sampling

For efficient training, Mask2Former uniformly (or further with importance sampling [21]) samples K points in the image space when computing the matching costs and final losses. However, in the 3D occupancy space, the uniform sampling struggles to capture foreground regions, particularly the minor classes, due to sparsity and class imbalance. To address this issue, we propose the class-guided sampling method. More specifically, we first compute the class frequencies $\mathbf{n}_c \in \mathbb{R}^{N_c}$ from the training set, where N_c is the number of classes. Then we compute their reciprocal $\mathbf{w}_c = 1/\mathbf{n}_c$ and normalize its minimum to 1 with $\mathbf{w}_c = \mathbf{w}_c/\min(\mathbf{w}_c)$. Finally, the sampling weights are computed as $\mathbf{w}_c = (\mathbf{w}_c)^\beta$, where β is a hyper-parameter.

During training, each voxel is assigned a sampling weight according to its ground-truth class. We then use the multinomial distribution to sample K voxel positions for matching and supervision. Note that for nuScenes dataset with only sparse LiDAR point supervisions, we simply use the LiDAR points and random coordinates in a 1:1 ratio as the sampled points.

3.4. Loss Functions

Following Mask2Former [10], we compute the bipartite matching between the predicted and ground-truth segments, considering only the sampled positions. The matching cost includes the class loss and the binary mask loss. With the optimal matching computed by the Hungarian algorithm [22], the mask classification loss $\mathcal{L}_{\text{mask-cls}}$ is computed following the matching cost. Besides, the intermediate depth distribution for view transformation is supervised by the projections of LiDAR points, with the binary cross-entropy loss $\mathcal{L}_{\text{depth}}$ following BEVDepth [28]. The final training loss is a simple summation: $\mathcal{L} = \mathcal{L}_{\text{mask-cls}} + \mathcal{L}_{\text{depth}}$.

4. Experiments

4.1. Datasets

The SemanticKITTI dataset [2] is based on the popular KITTI Odometry Benchmark [15] and focuses on the se-

Table 3: **LiDAR segmentation results on nuScenes test set.** The proposed OccFormer outperforms the only vision-based method TPVFormer [20] and achieves comparable performance with LiDAR-based methods.

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 [27]	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 [57]	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 [7]	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 [50]	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++ [62]	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 [12]	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++ [52]	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 [51]	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 [20]	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 (ours)	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**

mantic scene understanding with LiDAR points and front cameras. OccFormer is evaluated by its task of semantic scene completion, but with the monocular left camera as input following MonoScene [4]. Specifically, the ground-truth semantic occupancy is represented as the 256 × 256 × 32 voxel grids. Each voxel is 0.2m×0.2m×0.2m large and annotated with 21 semantic classes (19 semantics, 1 free, 1 unknown). Following [4, 20], the 22 sequences are split into 10/1/11 for train/val/test.

The nuScenes dataset [3] is a large-scale autonomous driving dataset, collected in Boston and Singapore. The dataset includes 1000 driving sequences from various scenes. Each sequence lasts for around 20 seconds and the key-frames are annotated at 2Hz with 3D bounding boxes. The Panoptic nuScenes dataset [14] further extends the nuScenes dataset to provide the annotations for LiDAR semantic segmentation. Similar to TPVFormer [20], we train OccFormer with sparse LiDAR point supervisions for 3D semantic occupancy prediction. We follow the official protocol to split the total scenes into train/val/test splits with 700/150/150 scenes. We report quantitative results for the LiDAR segmentation and qualitative visualizations for the 3D semantic occupancy prediction.

4.2. Implementation Details

Network Structures. Considering the image backbone network, we adopt EfficientNetB7 [4] on SemanticKITTI and ResNet-101 [17] on nuScenes, following the compared methods [4, 20]. The view transformer creates the 3D feature volume of size 128×128×16, with 128 channels. The transformer encoder consists of 4 stages with 2 dual-path transformer blocks each. The generated multi-scale 3D features are projected to 192 channels and processed the multi-scale deformable self-attention with 6 layers. The transformer decoder mainly follows the implementation from Mask2Former [9]. We increase the number of sampling points to 50176 (4×) and set β as 0.25 for the class-guided sampling. The predicted occupancy is upsampled 2× to 256×256×32 for full-scale evaluation.

Table 4: Ablation study on the dual-path encoder.

Local	Global	Params	GFLOPs	IoU↑	mIoU↑
✓		74.1M	494.2	36.42	12.95
	✓	81.4M	407.4	36.37	12.93
✓	✓	81.4M	515.3	**36.50**	**13.46**
3D ResNet-16 [17]		132.5M	825.8	36.12	12.89
3D Swin-T [35]		82.3M	437.9	36.32	12.80

Table 5: Ablation study on the pixel decoder.

Method	Layer	params	GFLOPs	IoU↑	mIoU↑
MsDeAttn3D	3	2.74M	329.3	35.74	13.22
MsDeAttn3D	6	4.07M	379.2	**36.50**	**13.46**
FPN-3D [31]	-	4.35M	307.0	36.12	12.89

Training Setup. Unless specified, we train the model for 30 epochs on SemanticKITTI dataset and 24 epochs on nuScenes dataset. The AdamW [36] optimizer with initial learning rate 1e-4 and weight decay 0.01 is used. The learning rate is decayed by a multi-step scheduler. All models are trained with a batch size of 8 on 8 RTX 3090 GPUs with 24G memory. For data augmentation, we use random resize, rotation, and flip for the image space and 3D flip for the 3D volume space, following recent practices for BEV-based 3D object detection [19, 28, 58].

4.3. Metrics

We report the mean intersection over union (mIoU) for both the semantic scene completion (SSC) and the LiDAR segmentation tasks. Also, the intersection over union (IoU)

Table 6: Ablation study on the transformer decoder.

Resize method	Sampling method	IoU↑	mIoU↑
Tri-linear	Uniform	35.04	11.61
Max-pool	Uniform	35.41	12.13
Tri-linear	Class-guided	36.21	13.01
Max-pool	Class-guided	**36.50**	**13.46**

Table 7: Ablation study on encoder modules.

Method	IoU↑	mIoU↑
OccFormer	**36.50**	**13.46**
w.o. soft sum.	35.83	12.73
w.o. shared attn.	36.28	12.93
w.o. ASPP	36.12	12.92

Table 8: Ablation study on augmentations.

Image Aug.	3D Aug.	IoU↑	mIoU↑
✓		36.37	12.72
	✓	35.73	12.94
✓	✓	**36.50**	**13.46**

for the class-agnostic scene completion (SC) task is reported. To infer the LiDAR segmentation results, the LiDAR points are only used to query their corresponding semantic logits from the predicted 3D semantic occupancy volume.

4.4. Main Results

Semantic Scene Completion. As shown in Tab. 1, we report the quantitative comparison of existing monocular methods for the semantic scene completion task on SemanticKITTI test set. We can observe that OccFormer outperforms all existing competitors, especially for the more challenging task of semantic scene completion. Compared with the recent TPVFormer [20], our method achieves a remarkable boost of 1.06 mIoU, demonstrating the effectiveness of OccFormer for semantic scene completion. Also, we report the results on SemanticKITTI validation set in Tab. 2. OccFormer achieves comparable IoU for scene completion and significantly better performance for the SSC mIoU.

LiDAR Semantic Segmentation. Following the practices from TPVFormer [20], the LiDAR semantic segmentation task is utilized as a quantitative indicator for the 3D semantic occupancy prediction. As shown in Tab. 3, our method outperforms the only vision-based method TPVFormer and achieves comparable performance with the state-of-the-art LiDAR-based methods. Note that our method requires only one model to perform both the LiDAR segmentation and the semantic occupancy prediction, while the TPVFormer [20] model trained for LiDAR segmentation cannot produce reasonable occupancy predictions.

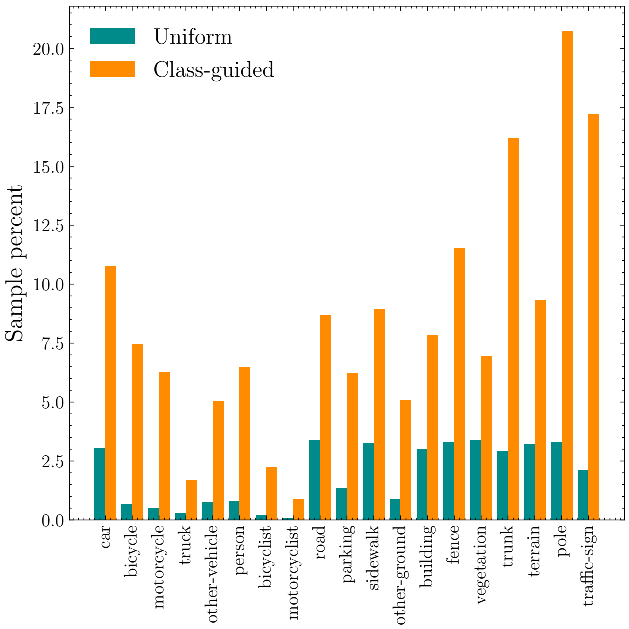

Figure 3: Comparisons of the uniform sampling and the proposed class-guided sampling. The sample percent is computed as the average sample ratio with 10k times of sampling. The class-guided sampling can significantly improve the quality of supervision.

4.5. Ablation Studies

To validate the effectiveness of proposed module, we conduct extensive ablation studies on SemanticKITTI validation set.

Ablation on the Dual-path Encoder. In Tab. 4, we ablate the dual-path design for the 3D feature extraction and compare it with other baseline methods. First, both the local and global paths contribute to the final performance positively. Since the local and global pathways focus on the fine-grained structures and the scene-level semantic layouts respectively, their complementary influence is quite understandable. Also, our dual-path transformer encoder achieves a better trade-off than the vanilla 3D convolution and the 3D windowed attention proposed in [35].

Ablation on the Pixel Decoder. In Tab. 5, we compare different structures for the pixel decoder, which aims to fuse multi-scale features and generate the per-voxel mask embeddings. Thanks to the dynamic receptive field and multi-scale aggregation, the multi-scale 3D deformable attention performs better than the classic FPN [31], tailored for 3D. Therefore, we utilize the 6-layer multi-scale 3D deformable attention as the pixel decoder for OccFormer.

Ablation on the Transformer Decoder. In Tab. 6, we ablate the methods of resizing attention masks and sam-

Table 9: **Detailed Comparison between sampling methods on SemanticKITTI [2] validation set.**

	SC	SSC																			
Sampling Method	IoU	road (15.30%)	sidewalk (11.13%)	parking (1.12%)	other-ground (0.56%)	building (14.1%)	car (3.92%)	truck (0.16%)	bicycle (0.03%)	motorcycle (0.03%)	other-vehicle (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%)	traffic-sign (0.08%)	mIoU
Uniform	35.41	**59.39**	**30.01**	**21.16**	0.18	**14.96**	**25.80**	7.10	0.16	**2.69**	7.94	18.77	2.43	30.14	0.00	0.00	0.00	**6.29**	3.53	0.00	12.13
Class-Guided	**36.50**	58.85	26.88	19.61	**0.31**	14.40	25.09	**25.53**	**0.81**	1.19	**8.52**	**19.63**	**3.93**	**32.62**	**2.78**	**2.82**	0.00	5.61	**4.26**	**2.86**	**13.46**

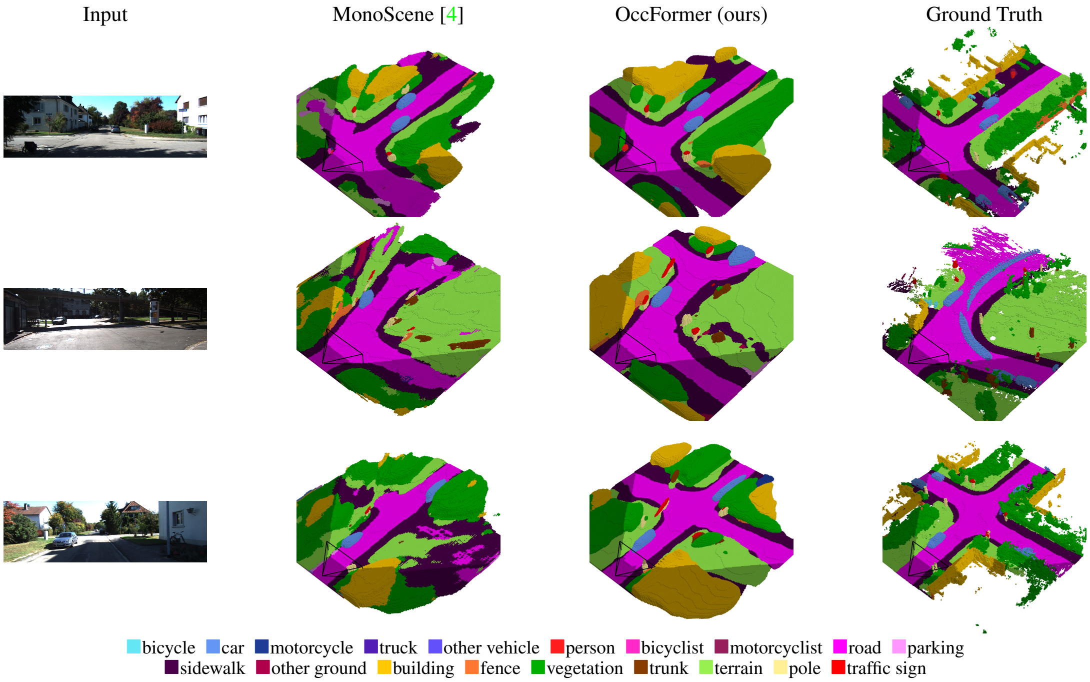

Figure 4: **Qualitative results on SemanticKITTI validation set.** The input monocular image is shown on the left and the 3D semantic occupancy results from MonoScene [4], our OccFormer, and the annotations are then visualized sequentially. The darker colors within the occupancy images represent the unseen parts out of the camera FOV.

pling points for supervision. Despite the state-of-the-art performance for 2D segmentation, the naive adaptation of Mask2Former [9] for 3D semantic occupancy prediction achieves inferior performance, only 11.61 mIoU. Compared with the tri-linear interpolation, we employ the max-pooling to preserve the fine-grained 3D predictions during downsampling, which achieves a boost of about 0.5 mIoU. On the other hand, the proposed class-guided sampling significantly outperforms the default uniform sampling because it can better adapt to the task of 3D semantic occupancy prediction, with a lot more "pixels" but much sparser supervisions than the 2D counterpart.

Ablation on Detailed Network Structures. As shown in Tab. 7, more detailed structures in the dual-path transformer encoder are ablated. First, the soft weight for fusing the dual-path outputs is removed and we observe an obvious drop in SSC mIoU from 13.46 to 12.73. Second, we remove the windowed attention in the global path, whose weights are shared with the local path, and observe a degradation of around 0.5 mIoU. Finally, we demonstrate the effectiveness of the bottleneck ASPP from the global path, which can extract long-range information for scene-level semantic layouts.

Ablation on Augmentations In Tab. 8, we ablate the employed augmentation techniques to train OccFormer. Since the attention mechanism, with strong capacities, is prone to over-fitting, these augmentation techniques are essential for reducing over-fitting and improving performance. Also, we find that the 3D augmentation which jointly transforms the 3D feature and the ground-truth semantic occupancy is

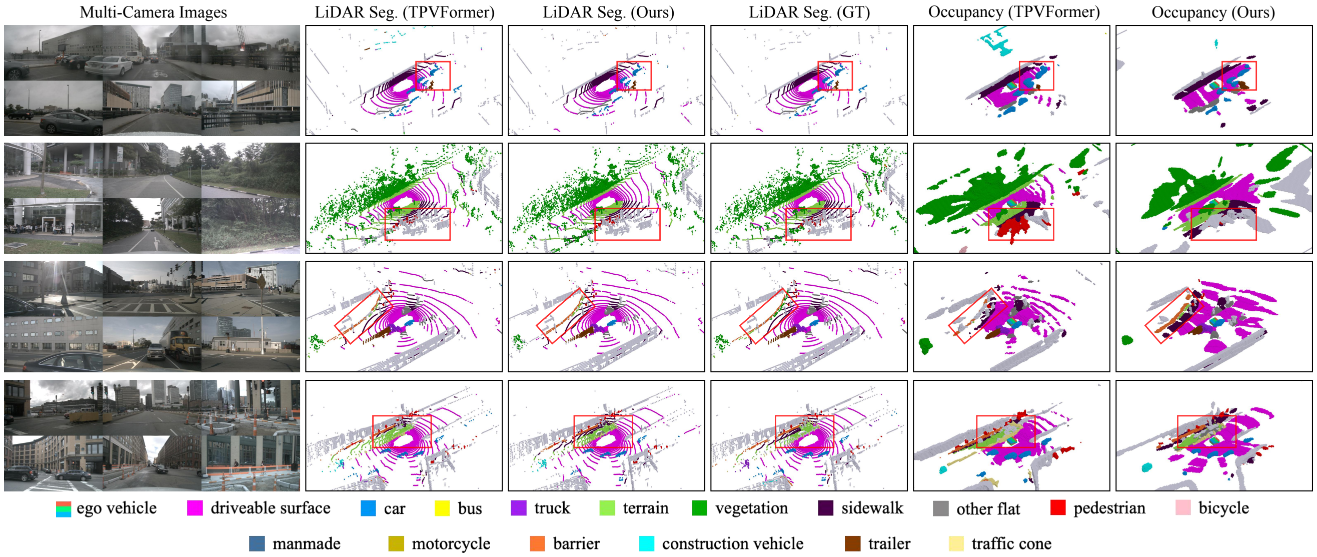

Figure 5: **Qualitative results on nuScenes validation set.** The leftmost column shows the input surrounding images, the following three columns visualize the LiDAR segmentation from TPVFormer [20], our method, and the annotation. The final two columns visualize the predicted 3D semantic occupancy from TPVFormer and our method.

more important. When it is disabled, the best performance is achieved at the 9th epoch, despite the total training schedule of 30 epochs.

4.6. Analysis

Class-Guided Sampling. Since the 3D feature volume contains a vast number of positions to supervise, a more effective sampling method is required to enable efficient training. As shown in Fig. 3, the proposed class-guided sampling can greatly improve the supervision signals for rare classes. Quantitatively, the class-wise comparison between uniform sampling and our class-guided sampling is presented in Tab. 9. Despite minor degradation in larger classes including road, sidewalk, and parking, the class-guided sampling demonstrates a remarkable boost in fewer classes, such as truck, person, bicyclist, and traffic sign. The different patterns from different sampling methods also offer an approach for the model ensemble.

4.7. Qualitative Results

Semantic Scene Completion. In Fig. 4, we visualize the predicted results of semantic scene completion on SemanticKITTI validation set from MonoScene [4] and our proposed OccFormer. Compared with MonoScene, our method can better understand the scene-level semantic layout and hallucinate the invisible regions. Also, OccFormer is good at recovering the object structures and reasoning about the interactions among neighbouring semantic classes. For example, the predicted buildings (in golden yellow) are more complete and located properly with the surrounding vegetation (in dark green), while MonoScene can generate the entangled results.

LiDAR Segmentation and 3D Semantic Occupancy. We visualize the predictions for LiDAR segmentation and 3D semantic occupancy in Fig. 5. Note that TPVFormer generates the required outputs with two separately trained models, while our method uses one single model. Nonetheless, OccFormer still achieves more accurate results on LiDAR segmentation. More importantly, the predicted 3D semantic occupancy from OccFormer is more contiguous, complete, and realistic than TPVFormer. For example, the predicted driveable surface is more contiguous and the foreground objects like cars and traffic cones have more accurate structures.

5. Conclusion

In this paper, we have presented OccFormer, a dual-path transformer network for camera-based 3D semantic occupancy prediction. To effectively process the camera-generated 3D voxel features, we have proposed the dual-path transformer block, which efficiently captures the fine-grained details and scene-level layouts with the local and global pathways. Also, we have been the first to employ mask classification models for 3D semantic occupancy prediction. Given the inherent sparsity and class imbalance, the proposed preserve-pooling and class-guided sampling have significantly improved the performance. OccFormer has achieved state-of-the-art performance for semantic scene completion on SemanticKITTI test set and for camera-based LiDAR segmentation on nuScenes test set.

References

[1] Adil Kaan Akan and Fatma Güney. Stretchbev: Stretching future instance prediction spatially and temporally. In *ECCV*, 2022. 1, 2

[2] 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, 5, 8

[3] 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. 1, 2, 6

[4] Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion. In *CVPR*, 2022. 1, 2, 5, 6, 8, 9

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2017. 3

[7] Qi Chen, Sourabh Vora, and Oscar Beijbom. Polarstream: Streaming object detection and segmentation with polar pillars. *NeurIPS*, 2021. 6

[8] 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*, 2020. 1, 2, 5

[9] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, 2022. 2, 4, 6, 8

[10] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *NeurIPS*, 2021. 4, 5

[11] Ran Cheng, Christopher Agia, Yuan Ren, Xinhai Li, and Liu Bingbing. S3cnet: A sparse semantic scene completion network for lidar point clouds. In *CoRL*, 2021. 1, 2

[12] Ran Cheng, Ryan Razani, Ehsan Taghavi, Enxu Li, and Bingbing Liu. 2-s3net: Attentive feature fusion with adaptive feature selection for sparse semantic segmentation network. In *CVPR*, 2021. 6

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 1

[14] 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. *RA-L*, 2022. 6

[15] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 1, 5

[16] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *CVPR*, 2019. 1

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 6

[18] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. Fiery: future instance prediction in bird's-eye view from surround monocular cameras. In *ICCV*, 2021. 1, 2

[19] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 1, 2, 3, 6

[20] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. *arXiv preprint arXiv:2302.07817*, 2023. 2, 5, 6, 7, 9

[21] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *CVPR*, 2020. 5

[22] H. W. Kuhn and Bryn Yaw. The hungarian method for the assignment problem. *Naval Res. Logist. Quart*, 1955. 5

[23] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019. 1

[24] Youngwan Lee, Jonghee Kim, Jeffrey Willette, and Sung Ju Hwang. Mpvit: Multi-path vision transformer for dense prediction. In *CVPR*, 2022. 3

[25] Jie Li, Kai Han, Peng Wang, Yu Liu, and Xia Yuan. Anisotropic convolutional networks for 3d semantic scene completion. In *CVPR*, 2020. 1, 2, 5

[26] Jie Li, Yu Liu, Dong Gong, Qinfeng Shi, Xia Yuan, Chunxia Zhao, and Ian Reid. Rgbd based dimensional decomposition residual network for 3d semantic scene completion. In *CVPR*, 2019. 2

[27] Shijie Li, Xieyuanli Chen, Yun Liu, Dengxin Dai, Cyrill Stachniss, and Juergen Gall. Multi-scale interaction for real-time lidar data segmentation on an embedded platform. *RA-L*, 2021. 6

[28] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. *arXiv preprint arXiv:2206.10092*, 2022. 1, 2, 3, 5, 6

[29] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *ECCV*, 2022. 1

[30] 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. *arXiv preprint arXiv:2203.17270*, 2022. 1, 2, 3

[31] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 6, 7

[32] Venice Erin Liong, Thi Ngoc Tho Nguyen, Sergi Widjaja, Dhananjai Sharma, and Zhuang Jie Chong. Amvnet: Assertion-based multi-view fusion network for lidar semantic segmentation. *arXiv preprint arXiv:2012.04934*, 2020. 6

[33] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. *arXiv preprint arXiv:2203.05625*, 2022. 1

[34] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 1, 3

[35] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *CVPR*, 2022. 6, 7

[36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 6

[37] Lang Peng, Zhirong Chen, Zhangjie Fu, Pengpeng Liang, and Erkang Cheng. Bevsegformer: Bird's eye view semantic segmentation from arbitrary camera rigs. *arXiv preprint arXiv:2203.04050*, 2022. 1, 2

[38] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, 2020. 1, 2, 3

[39] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *CVPR*, 2018. 1

[40] Cody Reading, Ali Harakeh, Julia Chae, and Steven L Waslander. Categorical depth distribution network for monocular 3d object detection. In *CVPR*, 2021. 2

[41] Christoph B Rist, David Emmerichs, Markus Enzweiler, and Dariu M Gavrila. Semantic scene completion using local deep implicit functions on lidar data. *TPAMI*, 2021. 2

[42] Thomas Roddick and Roberto Cipolla. Predicting semantic map representations from images using pyramid occupancy networks. In *CVPR*, 2020. 1, 2

[43] Luis Roldao, Raoul de Charette, and Anne Verroust-Blondet. Lmscnet: Lightweight multiscale 3d semantic completion. In *3DV*. IEEE, 2020. 1, 2, 5

[44] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, 2019. 1

[45] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, 2017. 2

[46] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020. 1

[47] Haotian Tang, Zhijian Liu, Shengyu Zhao, Yujun Lin, Ji Lin, Hanrui Wang, and Song Han. Searching efficient 3d architectures with sparse point-voxel convolution. In *ECCV*, 2020. 6

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[49] Yufei Xu, Qiming Zhang, Jing Zhang, and Dacheng Tao. Vitae: Vision transformer advanced by exploring intrinsic inductive bias. *NeurIPS*, 2021. 3

[50] 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*, 2021. 1, 2, 5, 6

[51] Dongqiangzi Ye, Zixiang Zhou, Weijia Chen, Yufei Xie, Yu Wang, Panqu Wang, and Hassan Foroosh. Lidarmultinet: Towards a unified multi-task network for lidar perception. *arXiv preprint arXiv:2209.09385*, 2022. 6

[52] Maosheng Ye, Rui Wan, Shuangjie Xu, Tongyi Cao, and Qifeng Chen. Drinet++: Efficient voxel-as-point point cloud segmentation. *arXiv preprint arXiv:2111.08318*, 2021. 6

[53] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *CVPR*, 2021. 1

[54] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *ICCV*, 2021. 1

[55] Jiahui Zhang, Hao Zhao, Anbang Yao, Yurong Chen, Li Zhang, and Hongen Liao. Efficient semantic scene completion network with spatial group convolution. In *ECCV*, 2018. 2

[56] Qiming Zhang, Yufei Xu, Jing Zhang, and Dacheng Tao. Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond. *IJCV*, 2023. 3

[57] 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. 6

[58] Yunpeng Zhang, Zheng Zhu, Wenzhao Zheng, Junjie Huang, Guan Huang, Jie Zhou, and Jiwen Lu. Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. *arXiv preprint arXiv:2205.09743*, 2022. 1, 2, 6

[59] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, 2021. 1

[60] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *CVPR*, 2022. 1, 2

[61] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 2, 4

[62] 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. 1, 6