

STCOcc: Sparse Spatial-Temporal Cascade Renovation for 3D Occupancy and Scene Flow Prediction

Zhimin Liao, Ping Wei*, Shuaijia Chen, Haoxuan Wang, Ziyang Ren
National Key Laboratory of Human-Machine Hybrid Augmented Intelligence
Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University

liaozm@stu.xjtu.edu.cn, pingwei@xjtu.edu.cn

Abstract

3D occupancy and scene flow offer a detailed and dynamic representation of 3D scene. Recognizing the sparsity and complexity of 3D space, previous vision-centric methods have employed implicit learning-based approaches to model spatial and temporal information. However, these approaches struggle to capture local details and diminish the model's spatial discriminative ability. To address these challenges, we propose a novel explicit state-based modeling method designed to leverage the occupied state to renovate the 3D features. Specifically, we propose a sparse occlusion-aware attention mechanism, integrated with a cascade refinement strategy, which accurately renovates 3D features with the guidance of occupied state information. Additionally, we introduce a novel method for modeling long-term dynamic interactions, which reduces computational costs and preserves spatial information. Compared to the previous state-of-the-art methods, our efficient explicit renovation strategy not only delivers superior performance in terms of RayIoU and mAVE for occupancy and scene flow prediction but also markedly reduces GPU memory usage during training, bringing it down to 8.7GB. Our code is available on <https://github.com/lzzzzzm/STCOcc>

1. Introduction

Accurate perception of 3D surrounding scenes is indeed vital for autonomous systems. The goal of occupancy and scene flow prediction [1] is to segment the entire space into 3D voxels and to determine the semantic and flow information of each voxel. This capability is crucial for understanding the environment and making informed decisions, which is well-suited for downstream tasks in autonomous systems, such as mapping and planning [2, 4, 38, 40, 49].

Due to data sparsity and information redundancy in 3D

*Corresponding author.

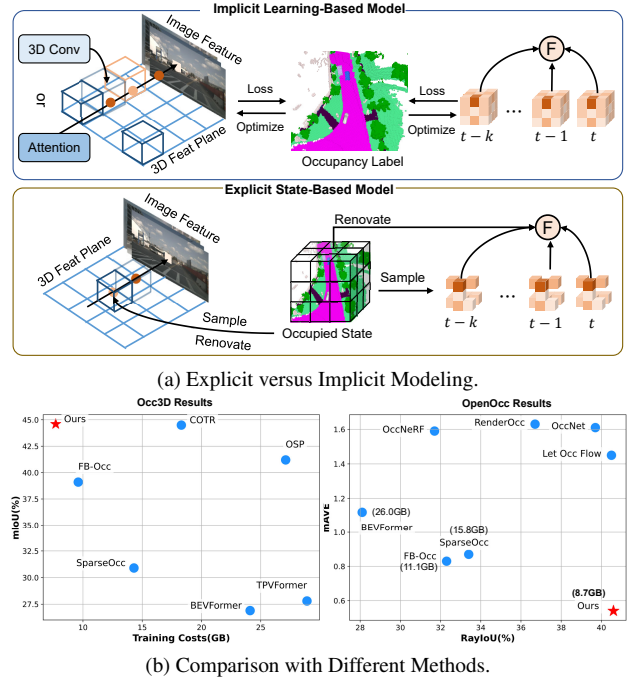


Figure 1. (a) **Explicit versus Implicit Modeling:** We propose a novel explicit state-based modeling approach that explicitly leverages the occupied state to maintain feature sparsity and model spatial details. (b) **Comparison with Different Methods:** Our approach achieves state-of-the-art performance of RayIoU and mAVE with lower training costs.

space, employing efficient and robust approaches for 3D feature processing is critical. Existing vision-centric methods [8, 9, 14–16, 21, 28, 30, 40, 44, 52] rely on implicit learning-based method for modeling spatial and temporal information, as shown in Fig. 1a. While these methods optimize 3D reconstruction and long-term temporal fusion through loss supervision, they face inherent limitations. Specifically, the exclusive reliance on loss-driven training hinders the model's ability to capture fine-grained spatial details and compromises the spatial discriminative capacity

of learned features, ultimately reducing the effectiveness for occupancy prediction tasks.

In this paper, we propose an explicit state-based modeling method that leverages the occupied state of 3D space to refine spatial and temporal feature representations, as shown in Fig. 1a. Our key insight stems from the inherent geometric correspondence between the occupied state and the 3D structure. Since the occupied state directly encodes the geometry of 3D space, it can serve as a robust prior to guide feature learning. This geometric alignment ensures that the feature space preserves structural fidelity, thereby simplifying the learning process and enhancing the discriminative power of the model.

We introduce STCOcc, a sparse Spatial-Temporal Cascade renovation framework tailored for occupancy and scene flow prediction. Within this framework, we present a *Spatial-Temporal Cascade Decoder* designed to renovate the 3D features both spatially and temporally, leveraging the occupied state of the 3D space. Specifically, at each stage, we employ a *Self-Recursive Occupancy Predictor* (SROP) to progressively refine the occupied state of the 3D space, thereby providing a more precise 3D geometric state for renovating the 3D features. Subsequently, we propose a sparse occlusion-aware attention mechanism to renovate the 3D features. Our attention mechanism differs from prior methods [15, 17, 19], which relied solely on depth information to renovate the 3D features. Instead, we utilize the occupied state in conjunction with bin depth information to accurately model the 3D spatial features. This approach provides details of local regions and makes the features more geometrically accurate. Furthermore, leveraging the accurate occupied state identified by SROP, we employ the *Occlusion-Aware Temporal Self-Attention* (OA-TSA) to model dynamic information using a recurrent strategy, supplying detailed short-term temporal information.

To efficiently integrate long-term temporal fusion, we propose a novel sparse-based method for temporal fusion modeling. It also avoids redundant information in historical data and preserves spatial information. Specifically, based on the occupied state, we sample non-empty and empty regions into long-term and short-term streams, respectively. Then, we incorporate the occupied state into both streams and employ a parallel strategy to fuse the temporal information within these two streams. This approach not only reduces computational costs but also retains the spatial information within the 3D space. Our contributions can be summarized as follows:

- We introduce an explicit state-based modeling approach designed to renovate the 3D features both spatially and temporally.
- We propose a sparse, occlusion-aware mechanism that provides more accurate geometric 3D features. Additionally, we propose a novel sparse-based method for mod-

eling long-term dynamic information. This approach not only reduces computational costs but also ensures spatial consistency.

- Our method achieves a RayIoU of 41.7% on Occ3D [40] and a RayIoU of 40.8% along with a mAVE of 0.44 for occupancy and scene flow prediction on OpenOcc [38], while also reducing the training memory usage to 8.7GB, as shown in Fig. 1b.

2. Related Work

2.1. Camera-based 3D Occupancy Prediction

Occupancy, as proposed by [29, 35], focuses on the continuous representation of 3D scenes. MonoScene [5] leverages monocular images for semantic scene completion, employing a 3D UNet to process voxel features. TPVFormer [9] lifts image features into 3D TPV space and expands them into voxel representations for 3D occupancy prediction. OccFormer [52] proposes a dual-path transformer for encoding the dense 3D volume features.

Considering the inherent sparsity of 3D scenes, recent methods [14, 18, 21, 28, 39, 46] optimize computational efficiency by processing only non-empty voxels using sparse convolutions or attention mechanisms. VoxFormer [14] utilizes a depth-based query proposal network to generate sparse query proposals for 3D-to-2D cross-attention. SGN [28] introduces a dense-sparse-dense framework that dynamically selects sparse seed voxels and employs hybrid guidance to enhance the convergence of semantic diffusion. Symphonize [10] reconstructs the 3D scene using instance queries. SparseOcc [21] proposes a fully sparse framework that focuses exclusively on non-empty regions. However, these sparse methods rely solely on the occupied state of 3D space to select the region of interest which ignores feature semantics or contextual relationships. Several methods [34, 46] leverage occupancy-based loss supervision to refine 3D features, improving the spatial fidelity of feature representations. However, their exclusive reliance on loss-driven optimization restricts their ability to model fine-grained 3D spatial structures.

2.2. Camera-based Temporal Modeling

Temporal modeling is essential for camera-based perception due to the inherent challenge of lacking depth information. When considering the modeling space, the methods can be divided into two main categories: image feature-based [19, 20, 22, 23, 43] and 3D feature-based [12, 13, 15, 33, 38, 42, 47, 51]. Image feature-based temporal modeling methods utilize multi-frame image features to provide dynamic information. For instance, PETR [22, 23] projects 3D points onto multi-frame image features to generate implicit 3D features for modeling temporal information. Sparse4D [19] creates 4D keypoints based on 3D an-

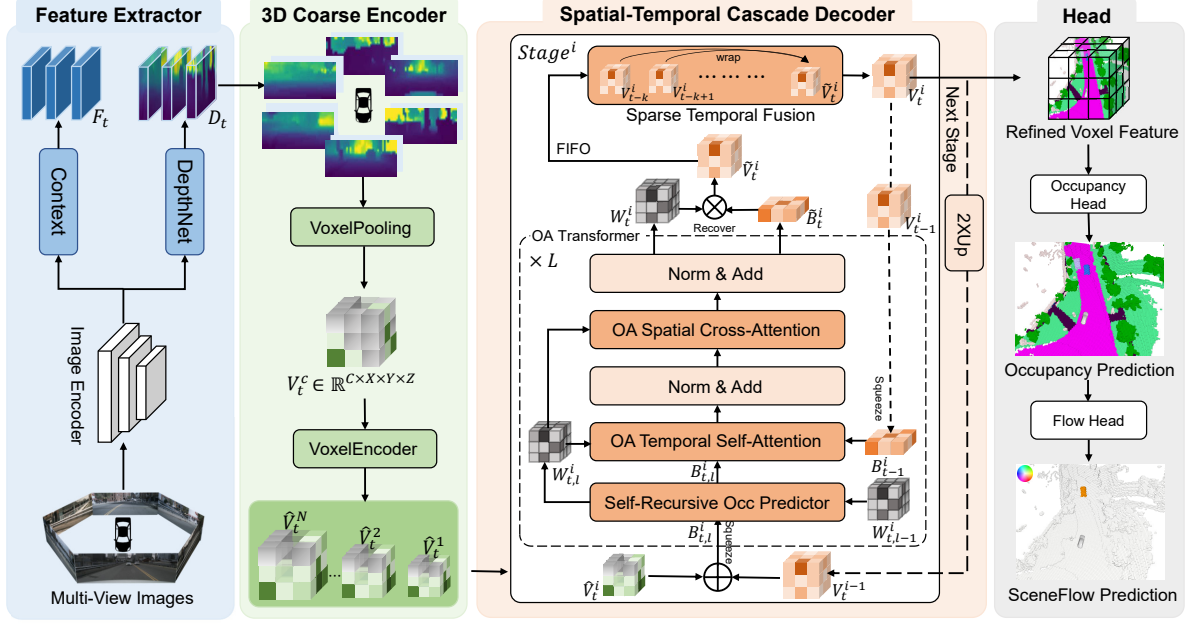


Figure 2. **The overall architecture of STCOcc.** The STCOcc framework is primarily composed of four integral modules: a feature extractor that captures image features and depth distribution, a 3D coarse encoder that generates multi-resolution coarse voxel features, a multi-stage spatial-temporal cascade decoder that incrementally renovates these coarse voxel features in both spatial and temporal dimensions, and a head module designed to leverage the refined voxel features for the prediction of 3D occupancy and scene flow.

chors and projects these points to aggregate features from multi-frame image features. SparseBEV [20] adaptively generates sampling points based on the query features.

On the other hand, 3D feature-based methods model temporal features in BEV or voxel space. BEVFormer [15] designs a temporal self-attention mechanism to recursively fuse BEV features. OccNet [38] extends this paradigm to voxel-based temporal self-attention to recursively fuse voxel feature. SOLOFusion [33] uses a parallel strategy to model BEV-level long-term information. We propose a novel sparse approach to model long-term 3D features.

3. Methods

3.1. Overall Architecture

An overview of STCOcc is presented in Fig. 2. Along the timestamp, we take the multi-view images as video sequence. At current frame t , multi-view images are first processed by the *Feature Extractor* to obtain image features and depth distribution. The *3D Coarse Encoder* uses the image features and depth distribution to create multi-resolution coarse voxel features via LSS-based transformation [12, 13, 36]. The *Spatial-Temporal Cascade Decoder* then progressively renovates these voxel features spatially and temporally, stage by stage. Finally, the *Head* module utilizes the refined voxel feature to predict occupancy and scene flow.

Feature Extractor. At each time t , the feature extractor initially uses an image backbone (e.g., ResNet [7]) to extract multi-view features $F_t = \{F_t^j \in \mathbb{R}^{C \times H \times W}\}_{j=1}^{N_c}$, where F_t^j represents the features of the j -th camera view at time t , and N_c is the number of cameras. Then, the depth network [12, 13] processes these image features to predict the bin depth distribution $D_t = \{D_t^j \in \mathbb{R}^{D_{bin} \times H \times W}\}_{j=1}^{N_c}$.

3D Coarse Encoder. The 3D coarse encoder adheres to the Lift and Splat framework as delineated in the LSS paradigm [12, 13, 36]. In the Lift phase, each pixel within the 2D image feature planes F_t is projected into the 3D voxel space guided by the predicted bin depth distribution D_t . Subsequently, the Splat phase consolidates the feature values of pixels falling within each voxel through voxel pooling [12, 13, 36], thereby constructing the coarse voxel feature $V_t^c \in \mathbb{R}^{C \times X \times Y \times Z}$. Subsequently, we engage a lightweight voxel encoder (e.g. ResNet3D-18 [7]) to produce multi-resolution coarse voxel features $V_t^i = \{\hat{V}_t^i \in \mathbb{R}^{C \times X_i \times Y_i \times Z_i} \mid i = 1, 2, \dots, N\}$, where $X_i = \frac{X}{2^{(N-i)}}$, $Y_i = \frac{Y}{2^{(N-i)}}$, $Z_i = \frac{Z}{2^{(N-i)}}$, and N is the number of processing stages.

3.2. Spatial-Temporal Cascade Decoder

To explicitly model the spatial and temporal information into features with the occupied state of 3D space, we introduce a spatial-temporal cascade decoder that renovates the 3D coarse voxel features V_t^c through a multi-stage process. As depicted in Fig. 2, the decoder comprises two pri-

mary components: Occlusion-Aware (OA) transformer layers, which accurately capture spatial and short-term temporal information, and a sparse-based temporal fusion module that employs a first-in, first-out (FIFO) memory sequence to encode long-term temporal information.

At each stage i , we refine voxel feature in BEV space rather than in voxel space. Initially, we fuse the coarse voxel feature \hat{V}_t^i with the refined voxel feature from the previous stage output V_t^{i-1} , and then project it into the BEV representation to obtain $B_{t,0}^i \in \mathbb{R}^{C \times X_i \times Y_i}$ as the input of OA transformer. We subsequently apply L layers *OA-Transformer*, which is analogous to the approach in [15], to refine $B_{t,0}^i$. This transformer layer includes three specialized modules: the *Self-Recursive Occupancy Predictor* (SROP), *OA Temporal Self-Attention* (OA-TSA), and *OA Spatial Cross-Attention* (OA-SCA). The SROP is designed to provide an accurate occupied state of the 3D space for each stage. The OA-TSA captures short-range temporal dynamics within the BEV space. The OA-SCA explicitly utilizes the occupied state to address the ambiguous projection problem [15, 17], which transfers geometric 2D feature information into the 3D space.

After the *OA-Transformer* processes the features, the refined BEV features \hat{B}_t^i are converted back to voxel form \hat{V}_t^i using the occupied weight W_t^i . Subsequently, we leverage the occupied state of the 3D space to guide a novel sparse-based approach for modeling long-term temporal information, thereby obtaining the output V_t^i at time t . V_t^i is upsampled by a factor of 2 for the next stage of refinement.

3.2.1. Self-Recursive Occupancy Predictor

To provide a more accurate representation of the occupied state of 3D space and to mitigate the one-off selection issues present in previous methods [14, 21, 28]. We draw inspiration from earlier studies [19, 45, 48] and design the self-recursive occupancy predictor. This predictor employs successive transformer layers to progressively refine the occupied state layer by layer. Specifically, at each layer l , it utilizes a simple Multilayer Perceptron (MLP) to recover the height of $B_{t,l}^i$ to voxel space and predict the occupancy weights $W_{t,l}^i \in \mathbb{R}^{X_i \times Y_i \times Z_i}$. The process of the self-recursive occupancy predictor can be described as follows:

$$W_{t,l}^i = f^i(B_{t,l}^i) + \alpha_l^i W_{t,l-1}^i, \quad (1)$$

The function $f^i(\cdot)$ corresponds to the occupancy predictor in stage i , which shares the same weights across different layers. The parameter α_l^i signifies the effect of layer $l-1$, a learnable parameter initialized to 0.5. At each stage, the initial $W_{t,0}^i$ is derived by upsampling the occupied weights from the previous stage, except that it is set to zeros in the first stage.

3.2.2. OA Temporal Self-Attention

The temporal modeling is crucial for representing the dynamic driving scene [15]. Given the historical BEV feature B_{t-1}^i (discussed in Sec. 3.2.4), we align it with the current feature $B_{t,l}^i$ via the motion of the ego-vehicle. To efficiently model the dynamic information, we propose Occlusion-Aware Temporal Self-Attention (OA-TSA) to focus the temporal modeling on the non-empty space. The OA-TSA is represented by:

$$TSA_{OA}(Q_{x,y}, \mathcal{B}, \bar{W}) = \sum_{b \in \mathcal{B}} \bar{w}_{x,y} \mathcal{F}_d(Q_{x,y}, b, \bar{w}_{x,y}), \quad (2)$$

where $Q_{x,y}$ denotes the BEV feature located at $p = (x, y)$ and $\mathcal{B} = \{B_{t,l}^i, B_{t-1}^i\}$. $\bar{W} \in \mathbb{R}^{X_i \times Y_i}$ is the average of $W_{t,l}^i$ along the z-axis, and \mathcal{F}_d signifies the deformable attention mechanism [53]. $\bar{w}_{x,y}$ represents the occupied weights \bar{W} at position p . Unlike the vanilla deformable attention [53], the offsets are predicted by the concatenation of \bar{W} and \mathcal{B} . By reweighting the TSA, we enable the model to focus more effectively on the dynamic information within the 3D space.

3.2.3. OA Spatial Cross-Attention

To explicitly utilize the occupied state to renovate the 3D features, we propose the Occupancy-Aware Spatial Cross-Attention (OA-SCA), which leverages the occupied state to enhance geometric features.

We first revisit the vanilla Spatial Cross-Attention (SCA) [15] as follows. As shown in Fig. 3, it samples N_{ref} 3D points $\mathbf{X} = \{\mathbf{x} = (x, y, z_h) | h = 1, 2, \dots, N_{ref}\}$ with different height for each $Q_{x,y}$, and projects these 3D points to 2D image feature planes F_t to obtain corresponding features. Formally, the SCA can be expressed as:

$$SCA(Q_{x,y}, F_t) = \sum_{\mathbf{x} \in \mathbf{X}} \mathcal{F}_d(Q_{x,y}, P(\mathbf{x}), F_t), \quad (3)$$

To simplify our initial analysis, we consider a scenario with a single camera, $P(\cdot)$ is the projection matrix that projects points from the 3D space onto the feature plane. The projection process can be mathematically represented as:

$$d \cdot \begin{bmatrix} u & v & 1 \end{bmatrix}^T = P \cdot \begin{bmatrix} x & y & z & 1 \end{bmatrix}^T, \quad (4)$$

where d denotes the depth of the point (u, v) on the 2D image plane. This 3D to 2D transformation introduces ambiguity, as different 3D points along the same projection ray map to identical 2D coordinates and are assigned the same features $F_{(u,v)}$, as illustrated in Fig. 3, where even the green points corresponding to empty areas receive the same feature.

To address this ambiguity, previous methods [17, 19] propose utilizing depth information to reweight the features of sampled points. However, these methods overlook the precision of the predicted depth and the state attributes of the sampled features. In contrast, our approach is inspired

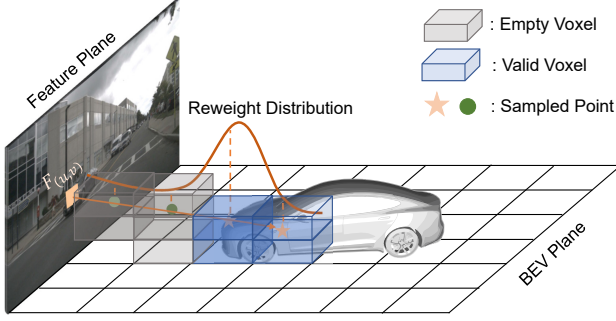


Figure 3. **Illustration of OA-SCA.** Due to the projection process, sampled points along the same ray in the feature plane are assigned identical features, even when they represent empty voxel space, as depicted by the green points. To address this limitation, our approach integrates depth and occupancy information to assign appropriate weights to the sampled points, thereby enhancing the differentiation of features along each ray.

by volume rendering techniques [27, 31], which allows us to renovate the features of sampled points more effectively. The volume rendering can be represented as:

$$\mathcal{C} = \sum_{n=1}^{N_r} \tau_n \cdot \sigma_n \cdot c_n, \quad (5)$$

where \mathcal{C} represents the expected value of light emitted by particles within the volume as a ray samples N_r points. σ_n denotes the density at each point, τ_n is the transmittance, and c_n is the corresponding color value. This physical function is analogous to the reverse processing of SCA, where \mathcal{C} corresponds to $F_{(u,v)}$ in the image plane, and c_n corresponds to the sampled point features, as illustrated in Fig. 3.

Based on above observation, our propose OA-SCA is designed to address the ambiguity inherent in vanilla SCA. Furthermore, to maintain the sparsity of the model, we employ probability sampling to select 3D reference points for refinement. The OA-SCA can be formulated as:

$$SCA_{OA}(Q_{x,y}, F_t) = \sum_{\mathbf{x} \in \mathbf{X}_s} \Omega_{\mathbf{x}} \mathcal{F}_d(Q_{x,y}, P(\mathbf{x}), F_t), \quad (6)$$

where $\mathbf{X}_s = \{\mathbf{x} \in \mathbf{X} \mid w_{\mathbf{x}} > u_{\mathbf{x}}\}$. During training, $u_{\mathbf{x}}$ follows a truncated normal distribution (0.5, 1). For stable inference, $u_{\mathbf{x}}$ is set to 0.5. This sampling approach allows our method to account for the uncertainty region during training. $w_{\mathbf{x}}$ is the reference point \mathbf{x} corresponding occupied weight. The reweighting parameter $\Omega_{\mathbf{x}} = w_{\mathbf{x}} \cdot \beta_{\mathbf{x}}$, for each 3D point \mathbf{x} , $\beta_{\mathbf{x}}$ can be calculated as:

$$\beta_{\mathbf{x}} = \exp \left(- \frac{\min(|d_r - (d'_r - \Delta d)|, |d_r - (d'_r + \Delta d)|)^2}{2\sigma^2} \right), \quad (7)$$

d_r and d'_r represent the depth attributes of the reference point \mathbf{x} respectively analogous to z and d in Eq. (4), they denote the distance of the sampled point from the ego-vehicle

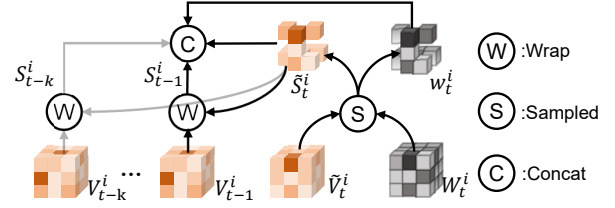


Figure 4. **Illustration of Sparse Temporal Fusion.** We implement temporal fusion using a parallel strategy in a sparse manner, focusing only on modeling the sampled features.

and the distance of the ray’s corresponding object from the ego-vehicle, respectively. It should be noted that d_r is derived from the predicted bin depth distribution D_t and is transformed into relative depth, while Δd represents the bin interval. The parameter σ serves as an adjustment factor for d_r and d'_r , enabling fine-tuning of the depth matching tolerance. By default, σ is set to 2, offering a balance between strict and lenient matching. This design takes into account both the attributes of bin depth distribution and the state of the sampled point, enabling a more accurate modeling of 3D spatial information.

3.2.4. Sparse Temporal Modeling

To integrate long-term historical information into the feature representation, at each stage i , we maintain a streaming history memory bank that adheres to the first-in, first-out rule to dynamically fuse information using a parallel strategy. Inspired by the SlowFast [6] and considering the redundancy in 3D space, we decouple the non-empty and empty regions into long-term and short-term streams, respectively. The long-term stream models the non-empty region to capture long-term dynamic information, while the short-term stream focuses on the empty region, modeling the overall 3D space in a short-term manner.

For clarity, we consider one stream as an example, as shown in Fig. 4. Given the current refined voxel feature \tilde{V}_t^i and the occupied weights W_t^i , we first apply a top-k sampling method to extract the seed feature $\tilde{S}_t^i \in \mathbb{R}^{C \times N_s}$ and corresponding occupied state $w_t^i \in \mathbb{R}^{C_s \times N_s}$, where C_s represents the embedding dimensions for occupied weights. Utilizing the position of S_t and the ego-pose transformation matrix T_t^{t-j} from frame t to frame $t-j$, we can retrieve the corresponding historical feature S_{t-j}^i . We then concatenate all the corresponding historically sampled features with w_t^i , and subsequently apply MLP to fuse the information along the channel dimension:

$$S_t^i = \text{MLP}([\tilde{S}_t^i, S_{t-1}^i, \dots, S_{t-k}^i, w_t^i]). \quad (8)$$

Finally, we add S_t^i to the corresponding position in \tilde{V}_t^i . By default, at each stage, we use T_i frames to model the long-term stream and $\frac{T_i}{2}$ frames to model the short-term stream. At each stage i , the fusion output V_t^i is appended to

Method	Backbone	Input Size	Epochs	RayIoU _{1m} (%)↑	RayIoU _{2m} (%)↑	RayIoU _{4m} (%)↑	RayIoU(%)↑	Memory(G)↓
BEVFormer* [15]	R101	1600 × 900	24	26.1	32.9	38.0	32.4	24.1
RenderOcc* [32]	Swin-B	1408 × 512	12	13.4	19.6	25.5	19.5	17.5
BEVDet-Occ* [8]	R50	704 × 256	90	23.6	30.0	35.1	29.6	10.2
FB-Occ* [17]	R50	704 × 256	90	26.7	34.1	39.7	33.5	9.6
SparseOcc (16f) [21]	R50	704 × 256	24	29.1	35.8	40.3	35.1	23.7
COTR [†] [26]	R50	704 × 256	24	36.3	41.7	45.1	41.0	18.3
OPUS-L [41]	R50	704 × 256	100	34.7	42.1	46.7	41.2	12.1
STCOcc (ours)	R50	704 × 256	36	36.2	42.7	46.4	41.7	7.7
STCOcc (ours)	R50	1408 × 512	36	36.9	42.8	46.7	42.1	8.9

Table 1. **Comparison of RayIoU (%) performance on the Occ3D-nus dataset.** * indicates models trained with camera mask, [†] denotes that official code was utilized to retrain the model.

Method	Backbone	Input Size	Memory(G)	mIoU(%)	others	barrier	bicycle	bus	car	cons. veh.	motor.	pedes.	tfc. cone	trailer	truck	drv. surf.	other flat	sidewalk	terrain	manmade	vegetation
BEVFormer [15]	R101	1600×900	24.1	26.9	5.9	37.8	17.9	40.4	42.4	7.4	23.9	21.8	21.0	22.4	30.7	55.4	28.4	36.0	28.1	20.0	17.7
CTF-Occ [40]	R101	1600×900	-	28.5	8.1	39.3	20.6	38.3	42.2	16.9	24.5	22.7	21.1	23.0	31.1	53.3	33.8	38.0	33.2	20.8	18.0
TPVFormer [9]	R101	1600×900	28.9	27.8	7.2	38.9	13.7	40.8	45.9	17.2	20.0	18.9	14.3	26.7	34.2	55.7	35.5	37.6	30.7	19.4	16.8
OSP* [37]	R101	1600×900	20.7	41.2	10.9	49.0	27.7	50.2	55.9	22.9	31.0	30.9	30.3	35.6	31.2	82.1	42.6	51.9	55.1	44.8	38.2
SparseOcc (8f) [21]	R50	704×256	14.3	30.9	10.6	39.2	20.2	32.9	43.3	19.4	23.8	23.4	29.3	21.4	29.3	67.7	36.3	44.6	40.9	22.0	21.9
FB-Occ* [16]	R50	704×256	9.6	39.1	13.6	44.7	27.0	45.4	49.1	25.2	26.3	27.9	27.8	32.3	36.8	80.1	42.8	51.2	55.1	42.2	37.5
ViewFormer* [11]	R50	704×256	-	41.9	12.9	50.1	27.9	44.6	52.9	22.4	29.6	28.0	29.2	35.2	39.4	84.7	49.4	57.4	59.7	47.4	40.6
COTR* [26]	R50	704×256	18.3	44.5	13.3	52.1	31.9	46.0	55.6	32.6	32.8	30.4	34.1	37.7	41.8	84.5	46.2	57.6	60.7	51.9	46.3
STCOcc* (ours)	R50	704×256	7.7	44.6	15.3	52.9	31.6	46.4	55.9	31.5	32.6	32.1	34.5	39.5	42.5	83.6	47.8	56.4	60.1	50.8	44.8
STCOcc* (ours)	R50	1408×512	8.9	45.0	15.2	52.3	32.2	50.5	56.5	31.7	33.9	33.4	33.8	38.9	44.9	83.9	47.4	57.1	60.1	50.6	42.7

Table 2. **Comparison of mIoU (%) performance on the Occ3d-nus dataset.** * indicates models trained with camera mask.

the memory queue, and a BEV representation B_t^i is stored for the subsequent frame of OA-TSA. This strategy provides a long-term perspective for recurrent temporal modeling and mitigates the gradient vanishing issue that is commonly encountered in previous recurrent temporal modeling methods [15].

3.3. Loss

We compute the occupancy loss for each stage, adopting the Scene-Class Affinity Loss (\mathcal{L}_{scal}) from MonoScene [5]. This loss is applied to both semantic and geometric predictions to ensure accurate scene understanding. Given the sparsity and class imbalance inherent in 3D scenes, we also utilize a weighted focal loss [16] combined with the Lovasz loss [3]. The loss for each stage i is formulated as follows:

$$\mathcal{L}_{occ}^i = \mathcal{L}_{scal}^{geo} + \mathcal{L}_{scal}^{sem} + \mathcal{L}_{focal} + \mathcal{L}_{lov}. \quad (9)$$

When considering the scene flow, we utilize the L1 loss \mathcal{L}_1 to supervise only the foreground voxel. The overall loss function is formulated as follow:

$$\mathcal{L} = \lambda_f \mathcal{L}_1 + \mathcal{L}_{depth} + \sum_{i=1}^N w_i \times \mathcal{L}_{occ}^i, \quad (10)$$

where \mathcal{L}_{depth} is the cross-entropy loss used to supervise the depth network. w_i is computed by $\frac{1}{2^{N-i}}$.

4. Experiments

4.1. Experimental Setup

Dataset. To evaluate the performance of our model, we utilize the Occ3D-nus dataset [40] for assessing 3D occupancy prediction and the OpenOcc dataset [38] for evaluating scene flow prediction. Both datasets are derived from the NuScenes dataset [4], encompassing 600 outdoor scenes for training, 150 for validation, and 150 for testing. Since the Occ3D-nus dataset does not provide scene flow labels, we have omitted the flow head when using this dataset. Additionally, since NuScenes does not provide depth labels, we follow previous methods [12, 13] by projecting LIDAR points onto the image plane to serve as depth labels.

Metric. The mean Intersection-over-Union (mIoU) is a prevalent metric for assessing occupancy prediction performance in the Occ3D-nus dataset. However, this metric only accounts for the visible area at the current moment and may not fully reflect the model’s completion capabilities [21]. Consequently, we also employ Ray-Based

Method	Sup.	Backbone	Input Size	RayIoU(%) \uparrow	mAVE \downarrow	Mem(G) \downarrow
OccNeRF-C [50]	C	R101	1600 \times 900	21.6	1.53	-
OccNeRF-L	L	R101	1600 \times 900	31.7	1.59	-
RenderOcc [32]	L	R101	1600 \times 900	36.7	1.63	-
Let Occ Flow [24]	C+L	R101	1408 \times 512	40.5	1.45	-
OccNet [38]	3D	R101	1600 \times 900	39.7	1.61	-
BEVFormer † [15]	3D	R50	1600 \times 900	28.1	1.12	26.0
FB-Occ † [16]	3D	R50	704 \times 256	32.3	0.83	11.1
SparseOcc † [21]	3D	R50	704 \times 256	33.4	0.87	15.8
STCOcc (ours)	3D	R50	704 \times 256	40.8	0.44	8.7

Table 3. **Comparison of RayIoU (%) and mAVE performance on the OpenOcc [38] dataset.** C and L denote Camera and Lidar supervision. † denotes that we utilize the official code and add the flow head to produce the results.

IoU (RayIoU) [21] to evaluate occupancy prediction performance. RayIoU is computed at three distance thresholds: 1, 2, and 4 meters. The final ranking metric is obtained by averaging the results across these thresholds. Additionally, we assess the performance of scene flow prediction using the mean absolute velocity error (mAVE) [38] across defined categories (e.g., car, truck). The mAVE is calculated for the set of true positives within a query ray threshold of 2 meters.

Implementation Details. We adopt the depth network from BEVStereo [12] and configure our model with three processing stages. Within each stage, the number of layers in the OA transformer is set to 2. The amount of historical information preserved for each stage is 16, 8, and 4, respectively. Unless specified otherwise, all models use the AdamW optimizer [25] with a global batch size of 16. Moreover, because our sampling strategy results in variable GPU memory costs, we report the maximum value in our results.

4.2. Main Results

Main Results on Occ3D-nus. As shown in Tab. 1 and Tab. 2. We compare our method with previous state-of-the-art methods on 3D occupancy task. Our methods achieves the state of the art performance 41.7% in RayIoU and 44.6% in mIoU, which is particularly noteworthy given the significantly lower training costs of 7.7GB, as opposed to the high training costs associated with COTR (18.3 GB) and OPUS-L (12.1 GB). Furthermore, since our spatial refinement process is dependent on the image size, we resize the input to 1408 \times 512 and achieve improved results in terms of RayIoU and mIoU.

Main Results on OpenOcc. As demonstrated in Tab. 3, we conducted experiments on the OpenOcc dataset to assess the performance of our model in terms of occupancy and scene flow. Our approach, which employs a smaller backbone (ResNet-50) and a reduced image input size (704 \times 256), achieves RayIoU scores of 40.8% and mAVE of 0.44.

Spatial		Temporal		Metric		
OA-SCA	SROP	STF	OA-TSA	RayIoU(%) \uparrow	mAVE \downarrow	Mem(G) \downarrow
				35.1	1.32	5.5
\checkmark				35.7	1.27	5.7
\checkmark	\checkmark			36.0	1.20	5.7
\checkmark	\checkmark	\checkmark		38.0	0.69	8.6
\checkmark	\checkmark	\checkmark	\checkmark	38.4	0.63	8.7

Table 4. **Ablation study on the each component.** SROP refers to the Self-Recursive Occupancy Predictor, OA-SCA refers to the Occupancy-Aware Spatial Cross Attention, OA-TSA refers to the Occlusion-Aware Temporal Self-Attention, and STF refers to the Sparse Temporal Fusion.

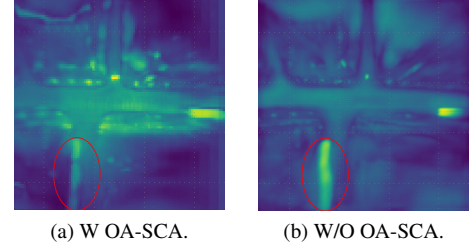


Figure 5. **Ablation on the OA-SCA module.** We visualize the features after refinement with and without the OA-SCA module.

These results surpass those of both OccNet [38] (which uses ResNet-101 with an input size of 1600 \times 900) and Let Occ Flow [24] (which also uses ResNet-101 with an input size of 1408 \times 512).

4.3. Ablation Study

To investigate the impact of various modules, we perform ablation experiments on the OpenOcc dataset [38]. It is important to note that the ablation experiments were conducted on half of the training dataset and then evaluated on the full validation set. Specifically, we used the first 300 sequences to constitute half of the training dataset.

The Effectiveness of Each Component. In Tab. 4, we demonstrate the effectiveness of each component in our model. For the baseline, we omit the OA-SCA and make the occupancy predictor independent in each stage, similar to previous one-off selection methods [21, 38]. This baseline achieves a RayIoU of 35.1% and a mAVE of 1.32 with a memory cost of 5.5GB. Integrating the OA-SCA into the baseline results in a 1.7% increase in RayIoU and a 3.7% increase in mAVE, with an additional memory cost of only 0.2GB. Introducing the SROP further enhances the model’s performance without incurring any additional memory costs. Utilizing long-term temporal fusion and the OA-TSA, we achieve a 6.3% increase in RayIoU and a 45.5% increase in mAVE, with a memory cost of only 1.9GB.

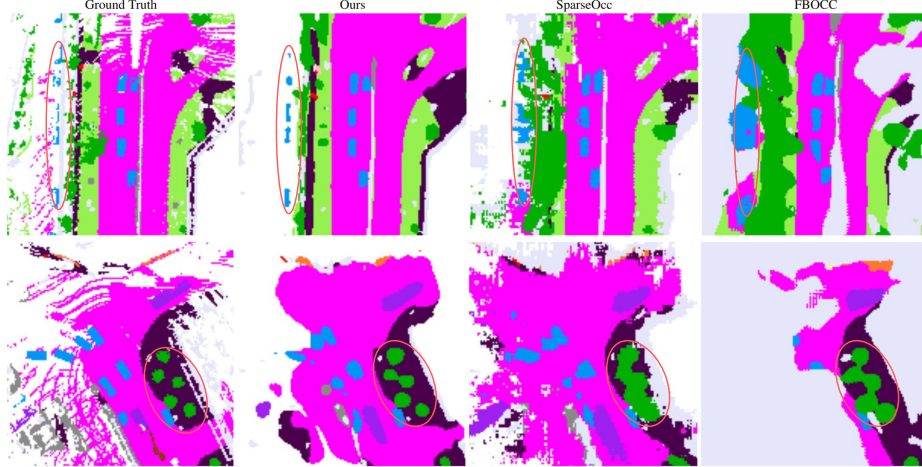


Figure 6. **Qualitative results on Occ3d-nus validation set.** As depicted in the red circle, our method delivers detailed predictions for objects such as cars and trucks, while also offering clear boundary delineations for structures like buildings and vegetation.

The Effectiveness of Sparse Temporal Modeling. In Tab. 5, we compare several representations in temporal modeling. The BEV modeling approach [33], while saving computation when modeling long-term history, sacrifices spatial information of the 3D space, resulting in poorer performance in occupancy prediction tasks. Our sparse modeling approach outperforms voxel-level modeling approach [16] in both effectiveness and computational cost. Furthermore, in Tab. 6, we compare our proposed Occupancy-Aware Temporal Self Attention (OA-TSA) with the vanilla Temporal Self Attention (TSA) [15]. Our method achieves superior performance in RayIoU and mAVE metrics, attributing this success to the guidance provided by occupied state modeling.

The Effectiveness of Occupancy-Aware Spatial Cross Attention. In Fig. 5, we compare the features refined by the OA-SCA module with those refined without it. It is evident that the OA-SCA module renovates the features, providing a more accurate geometric representation of the 3D scene, which is crucial for precise spatial modeling. Moreover, the OA-SCA module selectively enhances the foreground objects in the 3D scene, significantly improving the model’s discriminability. Furthermore, in Tab. 7, we compare the vanilla Spatial Cross Attention (SCA) [15] and the Depth-Aware Spatial Cross Attention (DA-SCA) proposed by FB-BEV [17] with our OA-SCA. It is observed that, due to inaccurate spatial modeling, neither SCA nor DA-SCA significantly improves performance. In contrast, Our explicit state-based modeling approach leverages the occupied state to accurately capture detailed spatial information.

4.4. Visualizations

In Fig. 6, we present the BEV visualizations on the Occ3D-nus validation set. In comparison to implicit learning-based

Representation	Frame	RayIoU(%) \uparrow	mAVE \downarrow	Mem(G) \downarrow
BEV [33]	8	36.8	0.81	6.7
Voxel [16]	8	37.2	0.73	7.8
Sparse (ours)	8	37.5	0.71	7.3
BEV	16	37.4	0.77	7.6
Voxel	16	38.0	0.64	9.8
Sparse (ours)	16	38.4	0.60	8.7

Table 5. **Ablation on the Sparse Temporal Fusion.** We compare the traditional representations of BEV and voxel to our sparse modeling approach across various frame numbers.

Method	RayIoU(%)	mAVE
W/O TSA [15]	37.9	0.69
TSA	38.0	0.67
OA-TSA	38.3	0.63

Table 6. **Ablation on the OA-TSA Module.**

Method	RayIoU(%)
W/O SCA [15]	37.6
SCA	37.5
DA-SCA [17]	37.7
OA-SCA	38.3

Table 7. **Ablation on the OA-SCA Module.**

approaches [16, 21], our explicit state-based method produces clearer boundaries for objects such as cars, buildings, and vegetation.

5. Conclusions

We propose an explicit state-based modeling approach to capture detailed geometric information in 3D space and integrate long-term dynamic information effectively. Our proposed STCOcc framework incorporates occlusion-aware mechanisms to enhance 3D features in both spatial and temporal aspects, thereby achieving better performance in 3D occupancy and flow prediction. The results demonstrate the efficacy of our paradigm, underscoring its strong potential for applications in downstream tasks.

Acknowledgement

This research was supported by the National Natural Science Foundation of China (No. U23B2060, No.62088102), and the Youth Innovation Team of Shaanxi Universities.

References

- [1] Tesla AI Day. <https://www.youtube.com/watch?v=j0z4FweCy4M>, 2021. 1
- [2] Jens Behley, Martin Garbade, Andres Milioto, Jan Quen-
zel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Se-
mantickitti: A dataset for semantic scene understanding of
lidar sequences. In *ICCV*, 2019. 1
- [3] Maxim Berman, Amal Rannen Triki, and Matthew B
Blaschko. The lovász-softmax loss: A tractable surrogate
for the optimization of the intersection-over-union measure
in neural networks. In *CVPR*, 2018. 6
- [4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora,
Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Gi-
ancarlo Baldan, and Oscar Beijbom. nuscenes: A multi-
modal dataset for autonomous driving. In *CVPR*, 2020. 1, 6
- [5] Anh-Quan Cao and Raoul De Charette. Monoscene: Mono-
cular 3d semantic scene completion. In *CVPR*, 2022. 2, 6
- [6] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and
Kaiming He. Slowfast networks for video recognition. In
ICCV, 2019. 5
- [7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.
Deep residual learning for image recognition. In *CVPR*,
2016. 3
- [8] Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Da-
long Du. Bevdet: High-performance multi-camera 3d object
detection in bird-eye-view. *arXiv:2112.11790*, 2021. 1, 6
- [9] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou,
and Jiwen Lu. Tri-perspective view for vision-based 3d se-
mantic occupancy prediction. In *CVPR*, 2023. 1, 2, 6
- [10] Haoyi Jiang, Tianheng Cheng, Naiyu Gao, Haoyang Zhang,
Tianwei Lin, Wenyu Liu, and Xinggang Wang. Sym-
phonize 3d semantic scene completion with contextual in-
stance queries. In *CVPR*, 2024. 2
- [11] Jinke Li, Xiao He, Chonghua Zhou, Xiaoqiang Cheng, Yang
Wen, and Dan Zhang. Viewformer: Exploring spatiotem-
poral modeling for multi-view 3d occupancy perception via
view-guided transformers. In *ECCV*, 2024. 6
- [12] Yinhao Li, Han Bao, Zheng Ge, Jinrong Yang, Jianjian Sun,
and Zeming Li. Bevstereo: Enhancing depth estimation
in multi-view 3d object detection with temporal stereo. In
AAAI, 2023. 2, 3, 6, 7
- [13] 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 detec-
tion. In *AAAI*, 2023. 2, 3, 6
- [14] Yiming Li, Zhiding Yu, Christopher Choy, Chaowei Xiao,
Jose M Alvarez, Sanja Fidler, Chen Feng, and Anima Anand-
kumar. Voxformer: Sparse voxel transformer for camera-
based 3d semantic scene completion. In *CVPR*, 2023. 1, 2, 4
- [15] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chong-
hao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer:
Learning bird’s-eye-view representation from multi-camera
images via spatiotemporal transformers. In *ECCV*, 2022. 2, 3, 4, 6, 7, 8
- [16] Zhiqi Li, Zhiding Yu, David Austin, Mingsheng Fang, Shiyi
Lan, Jan Kautz, and Jose M Alvarez. Fb-occ: 3d occupancy
prediction based on forward-backward view transformation.
arXiv:2307.01492, 2023. 1, 6, 7, 8
- [17] Zhiqi Li, Zhiding Yu, Wenhai Wang, Anima Anandkumar,
Tong Lu, and Jose M Alvarez. Fb-bev: Bev representa-
tion from forward-backward view transformations. In *ICCV*,
2023. 2, 4, 6, 8
- [18] Zhimin Liao and Ping Wei. Cascadeflow: 3d occupancy and
flow prediction with cascaded sparsity sampling refinement
framework. In *CVPR2024 Autonomous Grand Challenge
Track On Occupancy and Flow*, 2024. 2
- [19] Xuewu Lin, Tianwei Lin, Zixiang Pei, Lichao Huang, and
Zhizhong Su. Sparse4d: Multi-view 3d object detection with
sparse spatial-temporal fusion. *arXiv:2211.10581*, 2022. 2, 4
- [20] Haisong Liu, Yao Teng, Tao Lu, Haiguang Wang, and Limin
Wang. Sparsebev: High-performance sparse 3d object detec-
tion from multi-camera videos. In *ICCV*, 2023. 2, 3
- [21] Haisong Liu, Yang Chen, Haiguang Wang, Zetong Yang,
Tianyu Li, Jia Zeng, Li Chen, Hongyang Li, and Limin
Wang. Fully sparse 3d occupancy prediction. In *ECCV*,
2024. 1, 2, 4, 6, 7, 8
- [22] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun.
Petr: Position embedding transformation for multi-view 3d
object detection. In *ECCV*, 2022. 2
- [23] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Aqi Gao, Tian-
cai Wang, and Xiangyu Zhang. Petr v2: A unified framework
for 3d perception from multi-camera images. In *ICCV*, 2023.
2
- [24] Yili Liu, Linzhan Mou, Xuan Yu, Chenrui Han, Sitong Mao,
Rong Xiong, and Yue Wang. Let occ flow: Self-supervised
3d occupancy flow prediction. *arXiv:2407.07587*, 2024. 7
- [25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay
regularization. *arXiv:1711.05101*, 2017. 7
- [26] Qihang Ma, Xin Tan, Yanyun Qu, Lizhuang Ma, Zhizhong
Zhang, and Yuan Xie. Cotr: Compact occupancy transformer
for vision-based 3d occupancy prediction. In *CVPR*, 2024. 6
- [27] N. Max. Optical models for direct volume rendering. *TVCG*,
1995. 5
- [28] Jianbiao Mei, Yu Yang, Mengmeng Wang, Junyu Zhu, Xian-
grui Zhao, Jongwon Ra, Laijian Li, and Yong Liu. Camera-
based 3d semantic scene completion with sparse guidance
network. *IEEE Transactions on Image Processing*, 2024. 1, 2, 4
- [29] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Se-
bastian Nowozin, and Andreas Geiger. Occupancy networks:
Learning 3d reconstruction in function space. In *CVPR*,
2019. 2
- [30] Ruihang Miao, Weizhou Liu, Mingrui Chen, Zheng Gong,
Weixin Xu, Chen Hu, and Shuchang Zhou. Occdepth:
A depth-aware method for 3d semantic scene completion.
arXiv:2302.13540, 2023. 1

- [31] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 2021. 5
- [32] Mingjie Pan, Jiaming Liu, Renrui Zhang, Peixiang Huang, Xiaoqi Li, Li Liu, and Shanghang Zhang. Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision. *arXiv:2309.09502*, 2023. 6, 7
- [33] Jinhyung Park, Chenfeng Xu, Shijia Yang, Kurt Keutzer, Kris Kitani, Masayoshi Tomizuka, and Wei Zhan. Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. *arXiv:2210.02443*, 2022. 2, 3, 8
- [34] Liang Peng, Junkai Xu, Haoran Cheng, Zheng Yang, Xiaopei Wu, Wei Qian, Wenxiao Wang, Boxi Wu, and Deng Cai. Learning occupancy for monocular 3d object detection. In *CVPR*, 2024. 2
- [35] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *ECCV*, 2020. 2
- [36] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, 2020. 3
- [37] Yiang Shi, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Xinggang Wang. Occupancy as set of points. In *ECCV*, 2024. 6
- [38] Chonghao Sima, Wenwen Tong, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, and Hongyang Li. Scene as occupancy. In *ICCV*, 2023. 1, 2, 3, 6, 7
- [39] Pin Tang, Zhongdao Wang, Guoqing Wang, Jilai Zheng, Xiangxuan Ren, Bailan Feng, and Chao Ma. Sparseocc: Rethinking sparse latent representation for vision-based semantic occupancy prediction. In *CVPR*, 2024. 2
- [40] Xiaoyu Tian, Tao Jiang, Longfei Yun, Yucheng Mao, Huitong Yang, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. In *NeurIPS*, 2023. 1, 2, 6
- [41] Jiabao Wang, Zhaojiang Liu, Qiang Meng, Liujiang Yan, Ke Wang, Jie Yang, Wei Liu, Qibin Hou, and Ming-Ming Cheng. Opus: Occupancy prediction using a sparse set. In *NeurIPS*, 2024. 6
- [42] Ruihao Wang, Jian Qin, Kaiying Li, Yaochen Li, Dong Cao, and Jintao Xu. Bev-lanedet: An efficient 3d lane detection based on virtual camera via key-points. In *CVPR*, 2023. 2
- [43] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *CoRL*, 2022. 2
- [44] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *ICCV*, 2023. 1
- [45] Yiming Wu, Ruixiang Li, Zequn Qin, Xinhai Zhao, and Xi Li. Heightformer: Explicit height modeling without extra data for camera-only 3d object detection in bird's eye view. *IEEE Transactions on Image Processing*, 2024. 4
- [46] Junkai Xu, Liang Peng, Haoran Cheng, Linxuan Xia, Qi Zhou, Dan Deng, Wei Qian, Wenxiao Wang, and Deng Cai. Regulating intermediate 3d features for vision-centric autonomous driving. In *AAAI*, 2024. 2
- [47] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In *CVPR*, 2023. 2
- [48] Chenhongyi Yang, Tianwei Lin, Lichao Huang, and Elliot J Crowley. Widthformer: Toward efficient transformer-based bev view transformation. In *IROS*, 2024. 4
- [49] Zetong Yang, Li Chen, Yanan Sun, and Hongyang Li. Visual point cloud forecasting enables scalable autonomous driving. In *CVPR*, 2024. 1
- [50] Chubin Zhang, Juncheng Yan, Yi Wei, Jiaxin Li, Li Liu, Yansong Tang, Yueqi Duan, and Jiwen Lu. Occnerf: Self-supervised multi-camera occupancy prediction with neural radiance fields. *arXiv:2312.09243*, 2023. 7
- [51] Yunpeng Zhang, Zheng Zhu, Wenzhao Zheng, Junjie Huang, Guan Huang, Jie Zhou, and Jiwen Lu. Beverage: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. *arXiv:2205.09743*, 2022. 2
- [52] Yunpeng Zhang, Zheng Zhu, and Dalong Du. Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. In *ICCV*, 2023. 1, 2
- [53] 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. 4