SPSC: Sparse and Scalable Multi-Modal 3D Occupancy Prediction for Autonomous Driving

Qingju Guo[1], Shuang Li[2*], Binhui Xie[1], Jing Geng[1*], Wei Li[3]

[1]School of Computer Science & Technology, Beijing Institute of Technology, Beijing, China
[2]School of Artificial Intelligence, Beihang University, Beijing, China
[3]School of Intelligence Science and Technology, Nanjing University, SuZhou, China
{qingjuguo, binhuixie, janegeng}@bit.edu.cn, shuangliai@buaa.edu.cn, liweimcc@gmail.com

Abstract

3D semantic occupancy prediction offers a nuanced representation of the surrounding environment, which is crucial for ensuring the safety of autonomous driving. However, fine-grained scene representations inevitably result in cubic growth in data scale, which imposes substantial demands on model architecture and computational complexity, especially in high-resolution scenarios. Existing approaches for handling high-resolution scenes typically obtain fine-grained features by grid sampling on low-resolution feature map, resulting in limited sparsity and insufficient feature interaction. This paper presents a framework leveraging SParse representation and SCalable feature interaction to address the aforementioned challenges, called SPSC. Specifically, we maintain sparsity by progressively pruning unoccupied queries during the coarse-to-fine process, thereby reducing the scale of data that the model needs to handle. Subsequently, we introduce query serialization, which transforms queries into an ordered sequence while preserving their spatial structure. This enables fine-grained feature interaction while maintaining linear computational complexity and a larger receptive field. Without complex architectural designs, SPSC significantly outperforms SOTA approaches, enhances the mIoU by 12.0%, 11.0% and 4.8% on nuScenes-Occupancy dataset under the muli-modal, LiDAR and camera settings, respectively.

Introduction

Accurate perception of fine-grained 3D environmental information plays a pivotal role in autonomous driving systems, especially in scenarios when High Definition maps (Li et al. 2022; Qiao et al. 2023; Liu et al. 2024b) are unavailable. Traditional tasks such as 3D object detection (Pan et al. 2021; Yin, Zhou, and Krahenbuhl 2021; Liu et al. 2022; Wang et al. 2022) and Bird's Eye View (BEV)-based perception (Li et al. 2024b; Liu et al. 2023) employ coarse-grained representations of objects, which limits their capacity to comprehensively understand complex surrounding environments. Unlike conventional approaches, 3D semantic occupancy prediction assigns semantic categories to each voxel within the 3D space, enabling precise representation of irregularly-shaped objects, and has emerged as a major research direction in autonomous driving.

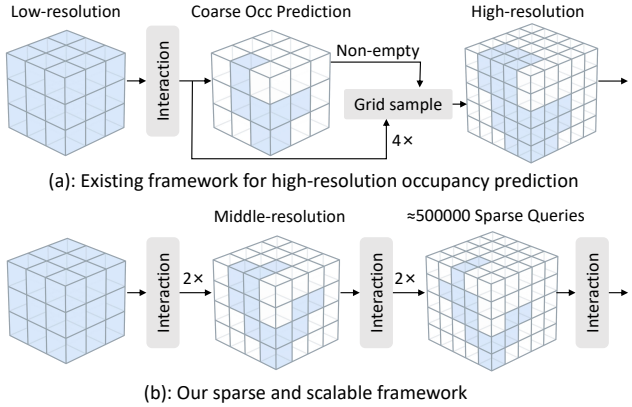

Figure 1: (a) Existing frameworks typically leverage coarse-grained predictions as guidance for grid sampling to directly acquire fine-grained features, resulting in limited sparsity and insufficient interaction between fine-grained features. (b) SPSC preserves query sparsity while enabling feature interactions across all granularities.

However, while fine-grained scene representations enhances perceptual capabilities, it simultaneously leads to a substantial increase in computational overhead. To achieve high-resolution occupancy prediction, we need to perform category predictions for roughly 10^7 voxels per frame, which is computationally infeasible given current hardware limitations. A practical approach is to utilize sparse scene representation. Due to the inherent sparsity of real-world environments, more than 90% voxels in the 3D space remain unoccupied. Inspired by this observation, existing methods have significantly reduced computational costs by progressively filtering out non-occupied voxels (Liu et al. 2024a) or modeling occupied voxels through a fixed number of queries (Wang et al. 2024b). However, due to their reliance on global self-attention mechanisms, these methods still incur substantial computational overhead in high-resolution scenarios ($\sim 5 \times 10^5$ non-empty voxels). Therefore, sparse scene representation is only a necessary approach for handling high-resolution scenarios. Under sparse scene representation, we still need to explore an efficient method for processing non-empty voxels.

*Corresponding Authors are Shuang Li and Jing Geng.

As illustrated in Fig. 1 (a), to reduce computational costs, existing mainstream frameworks (Wang et al. 2023; Pan, Wang, and Wang 2024; Wang et al. 2024a; Zhang, Ding, and Liu 2024b,a) first perform feature interaction at low resolution to produce coarse-grained occupancy predictions. They then upsample the predicted sparse, non-empty voxels and directly sample high-resolution features from the low-resolution feature maps using grid sampling. These methods primarily focus on detailed design of network structures at low resolution, while lacking feature interactions at high resolution. Although complex designs can improve performance, we believe that over-optimizing network structures may compromise generalization performance. Therefore, as illustrated in Fig. 1 (b), we focus more on sparse scene representation and high-resolution feature interactions rather than specific improvements to network structures.

To address the aforementioned issues, a framework with simple structure, based on sparse scene representation, and capable of efficiently handling large-scale feature interactions is desired. In this paper, we employ coarse-to-fine occupancy queries for 3D scene representation (**sparse**) and serialized query attention for large-scale feature interaction (**scalable**). For LiDAR, camera, and multimodal data, our approach consistently achieves state-of-the-art performance. More precisely, the proposed technique, SParse and SCalable multi-modal framework, named SPSC, begins with sampling extracted LiDAR and image features into low-resolution dense grids according to used modalities to form initial occupancy queries. Next, a coarse-to-fine decoder is applied to process the initial occupancy queries. Within each decoder layer, multi-modal features are sampled via query-guided sampling and fused with query features. The queries are then serialized into an ordered sequence using space-filling curves, enabling efficient large-scale feature interactions while preserving geometric information. Finally, the query resolution is upsampled by a factor of 2×, and unoccupied queries are pruned to maintain sparsity. Due to the efficiency of SPSC in handling large-scale data, we conducted experiments using multi-frame inputs, which further enhances model performance. The contribution of this paper can be summarized as follows:

- We propose a sparse and scalable multi-modal framework, which focuses on sparse scene representation and high-resolution feature interactions.
- We propose query serialization, which enables efficient large-scale query interactions while preserving the query's geometric information.
- SPSC demonstrates superior performance on the high-resolution nuScenes-Occupancy benchmark and the use of multi-frame inputs further boosts model performance. Thoughtful ablation studies further confirm the effectiveness of each component.

Related Work

Efficient 3D Occupancy Prediction

The massive scale of data has driven research into efficient 3D occupancy prediction. Existing approaches to achieving efficiency primarily focus on two strategies: one is reducing the scale of data to be processed, and the other is minimizing the computational complexity of the model.

Reducing the scale of data. There are two main approaches to reducing the scale of data: projection-based methods and sparse representation-based methods. For projection-based methods, either BEV (Yu et al. 2023; Lu et al. 2024; Yu et al. 2024; Shi et al. 2024) or TPV representations (Cui et al. 2024; Liang et al. 2024; Huang et al. 2023) are used to reduce the area processed by the model. These methods inevitably lose geometric information of objects due to the projection operations. For sparse representation-based methods, (Liu et al. 2024a) progressively filters out non-occupied voxels to maintain sparsity. (Wang et al. 2024b) only models occupied voxels through a fixed number of queries. However, even sparse representations involve a large number of voxels to be processed in high-resolution scenarios, which constrains the computational complexity of the model.

Minimizing the computational complexity. Existing mainstream frameworks (Wang et al. 2023; Pan, Wang, and Wang 2024; Wang et al. 2024a; Zhang, Ding, and Liu 2024b,a) directly obtain high-resolution features by grid sampling on low-resolution feature map, which lacks feature interaction. (Wang et al. 2024c) employs RWKV to achieve linear complexity, but it fails to fully leverage the intrinsic geometric information of the scene. (Lu et al. 2025) based on octree queries but still constrained by the octree structure itself. (Tang et al. 2024) utilizes sparse convolution to process fine-grained features, but its limited receptive field prevents it from achieving optimal performance. In this work, we address the aforementioned issues through sparse serialized queries. Our approach ensures linear complexity while achieving a larger receptive field and effectively leveraging the intrinsic geometric information of the scene.

Method

In this section, we present SPSC, a sparse and scalable framework that utilizes serialized occupancy queries to enable feature interaction at high resolutions. As show in Fig. 2, SPSC first extract multi-level LiDAR and image features and the transformed 3D image features are fused with LiDAR features to obtain the initial occupancy queries. Then, we transform the unordered set of queries into an ordered sequence through query serialization, enabling the use of attention-based blocks with linear complexity to achieve interactions between queries. Finally, a coarse-to-fine decoder is used to reconstruct the sparse geometry of the scene.

Query Initialization

Our 3D sparse representation is based on occupancy queries Q, where each query q_i consists of 3D spatial coordinates $p_i = [x_i, y_i, z_i]$ and a feature f_i, aiming to represent a non-empty voxel in the scene. We adopt a low-resolution dense grid (*e.g.* $64 \times 64 \times 5$) as initial occupancy queries and obtain query features by sampling the lowest-level LiDAR and image features into this dense 3D grid.

LiDAR branch. Since the 3D encoder we use is voxel-based, sparse LiDAR features can be directly mapped to the

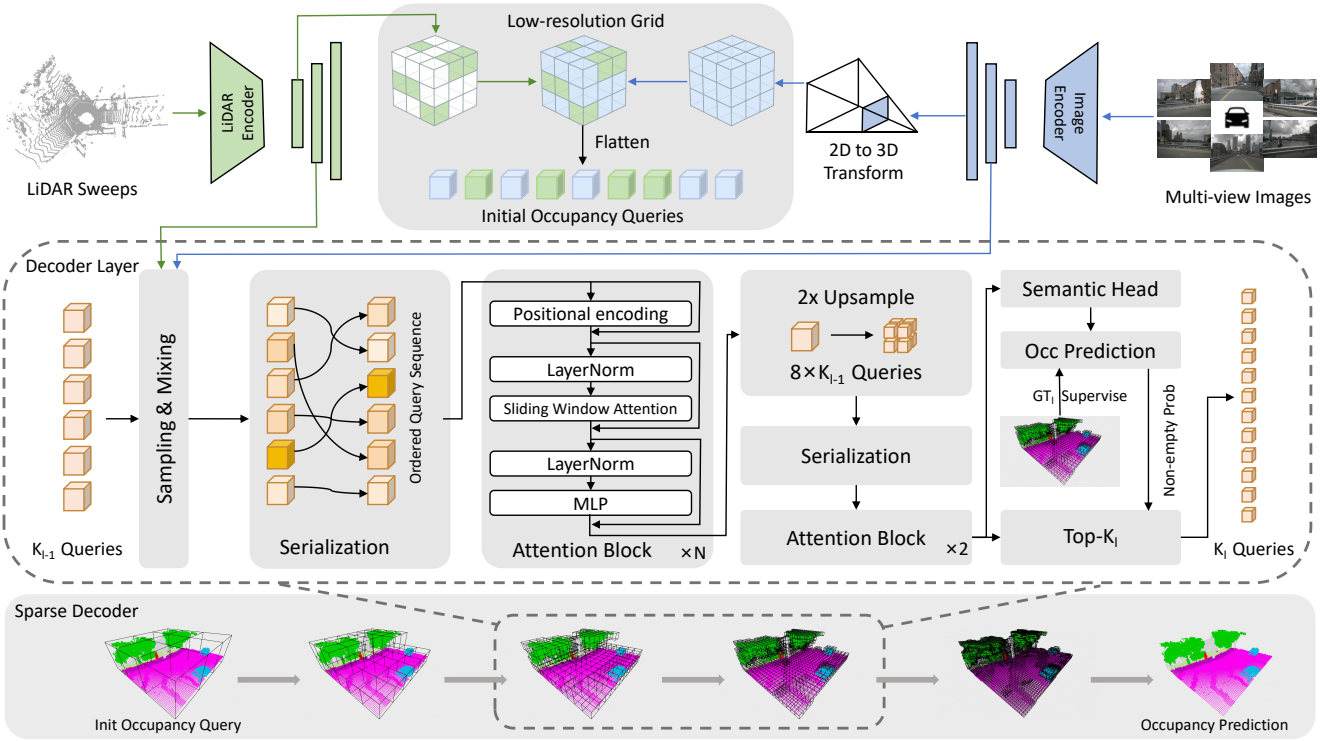

Figure 2: The overall architecture of the proposed SPSC framework.

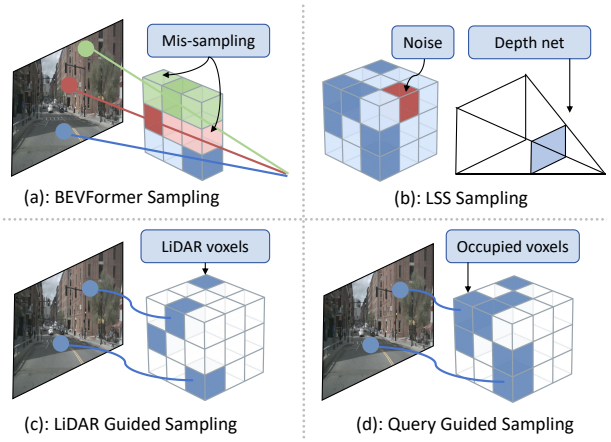

Figure 3: Comparison of different image feature sampling methods for query initialization.

dense 3D grid with the same resolution. Thus we derive LiDAR branch initial query features $F_{Q_0^L}$ as:

$$f^i_{Q_0^L} = \begin{cases} f^i_{L_0} & \text{if } p^i_{Q_0} \in L_0, \\ \{0\}^{1\times D_0} & \text{otherwise}, \end{cases} \tag{1}$$

where L_0 and D_0 represent the lowest-level LiDAR voxels and feature dimensions respectively.

Image branch. There are two mainstream image feature sampling methods. One approach (Huang et al. 2021; Li et al. 2023b,a) follows the lifting paradigm proposed in LSS (Philion and Fidler 2020), while the other (Liu et al. 2023, 2024a; Wang et al. 2024b) adopts the sampling point paradigm introduced in BEVFormer (Li et al. 2024b). As show in Fig. 3 (a), BEVFormer sampling maps all voxels along the same camera ray to the same location on the image, which results in non-occupied voxels along the ray also being sampled to image features. To reduce mis-sampling, we simply derive image branch initial query features $F_{Q_0^C}$ by LSS (Philion and Fidler 2020).

Multi-modal branch. As show in Fig. 3 (b), LSS (Philion and Fidler 2020) introduces additional parameters and computational overhead, and can introduce noise when depth estimation is inaccurate. Thus, for multi-modal branch, we use the LiDAR position information to guide image sampling (Fig. 3 (c)). We derive multi-modal branch initial query features $F_{Q_0^M}$ as:

$$f^i_{Q_0^C} = \begin{cases} \mathcal{G}(F_{C_0}, \mathcal{T}_{V\to I}(p^i_{Q_0})) & \text{if } p^i_{Q_0} \in L_0, \\ \{0\}^{1\times D_0} & \text{otherwise}, \end{cases} \tag{2}$$

$$F_{Q_0^M} = Linear([F_{Q_0^L}, F_{Q_0^C}]), \tag{3}$$

where $\mathcal{T}_{V\to I}$ transforms the voxel coordinates to the image coordinates, $\mathcal{G}$ is the grid sample function and $[\cdot,\cdot]$ is the channel-wise concatenation operation. In the following sparse decoder, since the queries provide denser positional information than LiDAR, we use the coordinates of the queries to guide image feature sampling (Fig. 3 (d)).

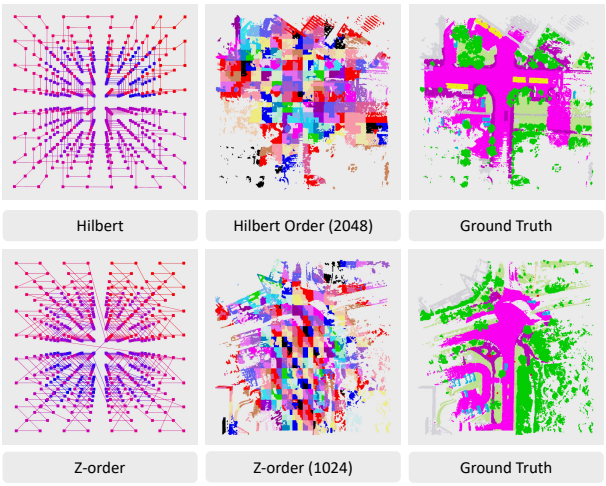

Figure 4: Visualization of query serialization. We show two types of space-filling curves (left), different number of serialized neighbouring queries represented by different color blocks (middle), and the semantic ground truth of the queries (right). Please zoom in for details.

Query Serialization

Although we use queries to achieve a sparse scene representation, the number of queries at high resolution remains too large to handle efficiently. Sparse convolution can handle such large-scale queries. However, the limited receptive field, combined with the scale of the queries restricting the number of layers that can be stacked, prevents sparse convolution from achieving optimal performance. Attention naturally has a larger receptive field, and many efficient attention mechanisms (Parmar et al. 2018; Beltagy, Peters, and Cohan 2020; Liu et al. 2021) have been proposed. However, these attention mechanisms typically reduce computational complexity by limiting the attention range. Therefore, an efficient way for determining the attention range based on the geometric information of the queries is desired.

Space-filling curves (Morton 1966; Hilbert 1935), widely used in database (Balkić, Šoštarić, and Horvat 2012), image compression (Liang et al. 2008) and point cloud processing (Wu et al. 2024), provide a possible solution. As show in Fig. 4 (left), space-filling curves can traverse the 3D grid space while maintaining a certain degree of spatial proximity. Typically, a space-filling curve is a bijective function $\psi : \mathbb{Z}^n \leftrightarrow \mathbb{Z}$ that maps a high-dimensional discrete space $\mathbb{Z}^n$ to a one-dimensional discrete space $\mathbb{Z}$. Here, we use three-dimensional spatial filling curves $\psi^3 : \mathbb{Z}^3 \leftrightarrow \mathbb{Z}$ to serialize the queries, which maps the coordinates of the queries to an integer that represents the position of the query on the one-dimensional curve. We obtain the query order by simply sorting these integers. The query serialization process can be formulated as follows:

$$Serialize(Q) = Q[argsort(\psi^3(P_Q))], \tag{4}$$

where $[\cdot]$ denotes the indexing operations.

Query serialization transforms unordered queries into an ordered sequence, so we can simply use a sliding window on the sequence to determine the attention range for each query. As show in Fig. 4 (middle), queries within the same window are also likely to be close in 3D space. Moreover, the entire serialization process only requires sorting the queries, which can be efficiently implemented on GPUs. Therefore, even though reduction from high to low dimensions unavoidably incurs some geometric information loss, we deem this trade-off necessary in pursuit of a larger receptive field and enhanced model scalability.

Sparse Decoder

Overall architecture. As show in Fig. 2, the sparse decoder employs a coarse-to-fine pipeline to decode the initial occupancy queries. In each layer, we first fuse the LiDAR and image features at the corresponding resolution with the current occupancy query features. The queries are then serialized and passed through several attention blocks for interaction. Next, each coarse-grained query is upsampled into 8 fine-grained queries. After additional attention interaction, a semantic head is used to predict the category of each query. The top-K queries with the highest occupancy probabilities are selected as the input for the next layer.

Attention block. We employ sliding window attention (Beltagy, Peters, and Cohan 2020) to process the serialized queries. By restricting the attention range to a fix-sized window, this approach achieves a computational complexity of $N \times W$, where N is the number of queries and W is the window size. Similar to (Chu et al. 2021; Wang 2023), we use a sparse convolution layer as the positional encoding:

$$PE(Q) = SPConv([F_Q, P_Q]), \tag{5}$$

where $[\cdot, \cdot]$ is channel-wise concatenation. The remaining design is identical to the original attention blocks.

Sampling & Mixing. For mixing LiDAR features, due to the potential erroneous removal of occupied queries earlier, LiDAR coordinates are not entirely contained within the query coordinates. To recover the erroneously filtered queries, we derive the mixed query feature F_{Q_l} as:

$$f^i_{Q_l} = \begin{cases} f^i_{L_l} + f^i_{Q_{l-1}} & \text{if } p^i_{Q_l} \in L_l \cap P_{Q_{l-1}}, \\ f^i_{L_l} & \text{if } p^i_{Q_l} \in L_l - P_{Q_{l-1}}, \\ f^i_{Q_{l-1}} & \text{otherwise}, \end{cases} \tag{6}$$

where $P_{Q_l} = L_l \cup P_{Q_{l-1}}$. For mixing image features, we first derive sampled image features $F_{Q^C_l}$ by query guided sampling $\mathcal{G}(F_{C_l}, \mathcal{T}_{V\to I}(p^i_{Q_l}))$ as described in §. Then, we mix $F_{Q^C_l}$ with query features F_{Q_l} as:

$$W = (\mathcal{A}([\mathcal{A}(F_{Q_l}), \mathcal{A}(F_{Q^C_l})])), \tag{7}$$

$$F_{Q_l} = \sigma(W) \odot F_{Q_l} + (1 - \sigma(W)) \odot F_{Q^C_l}, \tag{8}$$

where $\mathcal{A}$ is the attention block, σ is the $Sigmoid$ function and $\odot$ denotes element-wise product.

Other details. We use a linear layer to upscale the query channels to $8 \times D_l$ and distribute the enhanced features along the channel dimension to the upsampled queries. We always have $K_{l-1} < K_l < 8 \times K_{l-1}$ to maintain sparsity and a single linear layer is used as the semantic head to predict categories for the $8 \times K_{l-1}$ queries.

Method	Mod.	IoU	mIoU	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. surf.	other flat	sidewalk	terrain	manmade	vegetation
C-CONet (Wang et al. 2023)		20.1	12.8	13.2	8.1	15.4	17.2	6.3	11.2	10.0	8.3	4.7	12.1	31.4	18.8	18.7	16.3	4.8	8.2
C-OccMamba (Li et al. 2024a)		21.7	13.2	13.2	7.4	15.3	17.6	6.1	10.3	10.3	6.5	6.2	13.3	32.9	20.5	19.8	17.4	4.8	8.7
C-OccGen (Wang et al. 2024a)	C	23.4	14.5	15.5	9.1	15.3	19.2	7.3	**11.3**	11.8	8.9	5.9	13.7	34.8	**22.0**	21.8	**19.5**	**6.0**	**9.9**
C-SPSC		**24.3**	**15.2**	**18.9**	**9.6**	**19.4**	**21.3**	**8.5**	11.0	**13.9**	**9.4**	**6.6**	**14.0**	**34.8**	19.2	**22.4**	18.8	5.3	9.6
C-SPSC†		25.5	16.2	19.9	10.1	14.3	19.1	10.8	10.6	7.1	11.3	8.2	14.8	40.5	23.4	25.9	21.6	8.0	13.3
L-CONet (Wang et al. 2023)		30.9	15.8	17.5	5.2	13.3	18.1	7.8	5.4	9.6	5.6	13.2	13.6	34.9	21.5	22.4	21.7	19.2	23.5
L-OccGen (Wang et al. 2024a)		31.6	16.8	18.8	5.1	14.8	19.6	7.0	7.7	11.5	6.7	13.9	14.6	36.4	22.1	22.8	22.3	20.6	24.5
L-OccMamba (Li et al. 2024a)	L	36.4	22.7	26.8	11.3	20.8	26.1	**14.6**	16.3	20.3	14.0	17.5	20.3	**39.5**	24.9	25.9	**25.3**	**28.3**	**30.6**
L-SPSC		**36.6**	**25.2**	**30.9**	**17.3**	**23.5**	**30.0**	12.1	**24.9**	**34.3**	**17.9**	**18.1**	**22.8**	38.8	**26.5**	**26.8**	24.3	27.2	27.7
L-SPSC†		40.9	28.2	32.6	18.1	24.1	30.6	14.4	24.3	33.5	19.2	20.3	25.4	44.8	32.1	32.5	29.7	35.1	34.5
M-CONet (Wang et al. 2023)		29.5	20.1	23.3	13.3	21.2	24.3	15.3	15.9	18.0	13.3	15.3	20.7	33.2	21.0	22.5	21.5	19.6	23.2
CO-Occ (Pan, Wang, and Wang 2024)		30.6	21.9	26.5	16.8	22.3	27.0	10.1	20.9	20.7	14.5	16.4	21.6	36.9	23.5	25.5	23.7	20.5	23.5
OccGen (Wang et al. 2024a)		30.3	22.0	24.9	16.4	22.5	26.1	14.0	20.1	21.6	14.6	17.4	21.9	35.8	24.5	24.7	24.0	20.5	23.5
OccLoff (Zhang, Ding, and Liu 2024b)		31.4	22.9	26.7	17.2	22.6	26.9	16.4	22.6	24.7	16.4	16.3	22.0	37.5	22.3	25.3	23.9	21.4	24.2
OccMamba (Li et al. 2024a)	M	33.7	25.1	29.6	20.2	25.7	28.5	**16.7**	25.0	23.2	19.9	20.3	24.5	36.1	25.3	25.1	24.8	27.7	28.9
SPSC		**35.5**	**28.1**	**32.8**	**26.0**	**27.8**	**32.1**	16.6	**30.9**	**36.7**	**23.4**	**20.5**	**26.8**	**39.2**	**25.7**	**27.9**	**25.8**	**28.1**	**29.1**
SPSC†		40.9	30.1	34.0	24.0	26.7	32.1	18.3	29.3	35.4	24.7	20.5	28.1	43.5	30.4	32.3	30.5	36.1	35.7

Table 1: 3D semantic occupancy prediction results on nuScenes-Occupancy validation set. The C, L, M denotes **Camera-only**, **LiDAR-only** and **Multi-modal** methods, respectively. † denotes the model are trained using multiple historical frames as inputs. All mIoU scores are given in percentage (%). The best results are highlighted in **bold**.

Supervision. We compute losses for the output queries of each layer. Following CONet (Wang et al. 2023), we use cross-entropy loss $\mathcal{L}_{ce}$, lovasz-softmax loss (Berman, Triki, and Blaschko 2018) $\mathcal{L}_{ls}$, affinity losses (Cao and De Charette 2022) $\mathcal{L}_{scal}^{geo}$ and $\mathcal{L}_{scal}^{sem}$. Moreover, depth supervision (Li et al. 2023b) $\mathcal{L}_{d}$ is used to train a depth-aware LSS (Philion and Fidler 2020) in image branch. The overall training objective can be formulated as:

$$\mathcal{L}_{total} = \mathcal{L}_{ce} + \mathcal{L}_{ls} + \mathcal{L}_{scal}^{geo} + \mathcal{L}_{scal}^{sem} + \mathcal{L}_{d}, \tag{9}$$

We exclusively compute losses for top-K selected queries and the remaining queries are ignored.

Experiment

Experiment Setup

Dataset and metrics. We evaluate our proposed method using the nuScenes-Occupancy (Wang et al. 2023) dataset. The dataset consists of 28,130 training frames and 6,019 validation frames, each with dense semantic occupancy annotations. Voxel grids span a range of [-51.2m, 51.2m] along the X and Y axes and [-5m, 3m] along the Z axis, with a voxel resolution of [0.2m, 0.2m, 0.2m]. This results in an output volume of size 512 × 512 × 40. Each voxel is assigned one of 17 labels, including 16 semantic categories and 1 empty category. Following OpenOccupancy (Wang et al. 2023), we adopt Intersection over Union (IoU) as the geometric metric and mean IoU (mIoU) as the semantic metric.

Implementation details. Unless otherwise specified, we adopt the same experimental setup as in (Wang et al. 2023; Pan, Wang, and Wang 2024; Zhang, Ding, and Liu 2024b), to ensure a fair comparison. For single-frame experiments, a frame containing 6 images with a resolution of 1600×900 and 10 adjacent LiDAR sweeps are used as input. For mutiframe experiments, 4 image frames with a resolution of 704×256 (sampling interval of 2) and 10 LiDAR sweeps (sampling interval of 10) are used as input. We use ImageNet (Deng et al. 2009) pretrained ResNet50 (He et al. 2016) with FPN (Lin et al. 2017) as the imange encoder and Voxelnet (Zhou and Tuzel 2018) with Second (Yan, Mao, and Li 2018) FPN as the LiDAR encoder. The resolution of initial dense query grid is set to $64 \times 64 \times 5$ and the number of decoder layers is set to 3. The number of attention blocks is set to [8, 4, 2], top-K is set to [30,000, 80,000, 480,000] and the feature channel is configured as [256, 128, 64]. We employ two space-filling curves: Hilbert (Hilbert 1935) and Z-order (Morton 1966) and randomly select one during each serialization operation. The sliding window size in the attention block is fixed at 1024. The models are trained using PYTorch (Paszke et al. 2019) for 15 epochs on 8 NVIDIA 4090 GPUs, with a batch size of 8.

Main Results

As show in Tab. 1, we conduct a quantitative comparison with existing camera-based, LiDAR-based, and multi-modal 3D occupancy prediction methods on nuScenes-Occupancy val set. It is evident that SPSC achieves significant improvements compared to all the existing approaches. Compared with the current SOTA method, SPSC achieves a remarkable boost of 0.7%, 2.5%, and 3.0% mIoU for camera-only, LiDAR-only, and multi-modal benchmarks. This substantiates the efficacy of SPSC in high-resolution 3D occupancy prediction. Notably, SPSC demonstrates more pronounced

Method	Params	Latency	Modality	IoU	mIoU
BEVFormer	0M	2.2ms	C	22.7	13.9
			M	33.8	26.8
LSS	29.7M	79.2ms	C	24.3	15.2
			M	34.8	27.6
LiDAR Guided	0M	1.5ms	M	35.5	28.1
Optimized	0M	1.7ms	M	36.9	28.4

Table 2: Comparison of different treatments on query initialization. "Optimized" indicates that occupancy ground truth are utilized to guide the sampling of the initial features.

improvements in small-volume objects (*e.g.* bicycle, motorcycle, pedestrian, traffic cone), which validates that the interaction of high-resolution features can provide a more nuanced understanding of the scene. Moreover, by leveraging the capability of `SPSC` to efficiently process large-scale data, we further enhance the model's performance by incorporating information from multiple historical frames.

Ablations and Analysis

Query initialization. Experimentally, we find that different query initialization methods significantly impact the results. As show in Tab. 2, we present a detailed comparison of different image feature sampling methods used for query initialization. We employ occupancy ground truth to guide the sampling of image features, sampling only at the locations of occupied voxels, which serves as the upper bound for these sampling methods. BEVFormer sampling (Li et al. 2024b) suffers from performance degradation due to missampling, while LSS sampling (Philion and Fidler 2020) introduces additional parameters and inference latency. In contrast, LiDAR-guided sampling achieves near-optimal performance without introducing additional overhead.

The scale of feature interaction. We progressively reduce the scale of feature interaction by gradually decreasing the number of decoder layers, and use the same occupancy head as (Wang et al. 2023; Pan, Wang, and Wang 2024) to upsample low-resolution features to high resolution ($512 \times 512 \times 40$) through grid sampling. As shown in Tab. 3, the model performance improves significantly as the scale of feature interaction increases, which demonstrates the importance of fine-grained feature interaction. We also provide results for some small-volume categories. Compared to the overall results, the improvement for small-volume objects is more pronounced, indicating that fine-grained feature interaction enables the model to achieve a more detailed understanding of the scene. Notably, compared to M-CONet (Wang et al. 2023) with the same scale of feature interaction ($128 \times 128 \times 10$), our method significantly improves the mIoU from 20.1 to 24.3. This improvement is primarily attributed to the larger receptive field of attention mechanisms compared to sparse convolutions.

Serialization patterns. In Tab. 4, we compare the inference latency and 3D occupancy prediction results across different serialization patterns. Each decoder layer needs to serialize the initial K_l queries as well as the upsampled $8 \times K_l$ queries. Therefore, the serialization query scales for L_0, L_1, and L_2 are 184,320, 270,000, and 720,000, respectively. Z-order curve (Morton 1966) is computationally efficient, while Hilbert curve (Hilbert 1935) better preserves spatial locality. As a result, Z-order curve achieves lower inference latency, whereas Hilbert curve delivers higher performance. Our experiments also demonstrate that combining multiple serialization patterns can yield stronger performance.

Interaction scale	bic.	motor.	ped.	tra.	IoU	mIoU
$64 \times 64 \times 5$	15.3	20.1	25.3	13.9	27.5	20.4
$128 \times 128 \times 10$	21.8	25.4	30.8	19.2	31.4	24.3
$256 \times 256 \times 20$	24.3	28.4	35.2	21.1	33.7	26.5
$512 \times 512 \times 40$	26.0	30.9	36.7	23.4	**35.5**	**28.1**

Table 3: Comparison of different feature interaction scales. Bic., motor., ped. and tra. denote bicycle, motorcycle, pedestrian and traffic cone, respectively

Patterns	Latency(ms)			IoU	mIoU
	L_0	L_1	L_2		
Z	0.9	1.1	1.2	34.5	27.4
H	4.9	6.2	24.5	34.7	27.5
$Z + H$(fix)	2.9	3.6	12.9	35.1	27.8
$Z + H$(random)	5.3	6.9	24.9	**35.5**	**28.1**

Table 4: Analysis of serialization patterns. "fix" denotes randomly selecting one pattern per frame. "random" indicates randomly selecting one pattern for each layer. L denotes all serialization operations within a single decoder layer.

Top-K queries. In Tab. 5, we investigate the impact of query sparsity on the final results. The number of queries per layer varies proportionally, and we only provide the number of queries for the final layer. We found that the optimal number of queries is around 480k, which accounts for only about 4.6% of the dense voxels. It is worth noting that while further increasing the number of queries can improve scene completion performance, the introduction of more non-occupied queries leads to a decline in the final results. Remarkably, increasing the query scale by 80k only incurs an additional computational overhead of approximately 10 milliseconds, which demonstrates the efficiency and scalability of our serialization-based feature interaction.

Window size. We explore the impact of receptive field on the final results by adjusting the size of the sliding window in Tab. 6. During the process of gradually increasing the sliding window size, the model's performance significantly improves and reaches its optimal value at 1024. Surprisingly, when the receptive field is expanded to 2048, the model's performance actually declines. As show in Fig. 4 (middle), a window size of 1024 is already sufficient to encompass most objects, while a larger window size may cause the model to overlook some smaller objects, thereby leading to a decline in model performance. We also conduct experiments using sparse convolutions with a kernel size of 3 to replace the attention blocks. Compared to the attention mechanism, its performance is suboptimal due to the limited receptive field.

Model efficiency. In Tab. 7, we present the model's parameter count, GPU memory usage during training and infer-

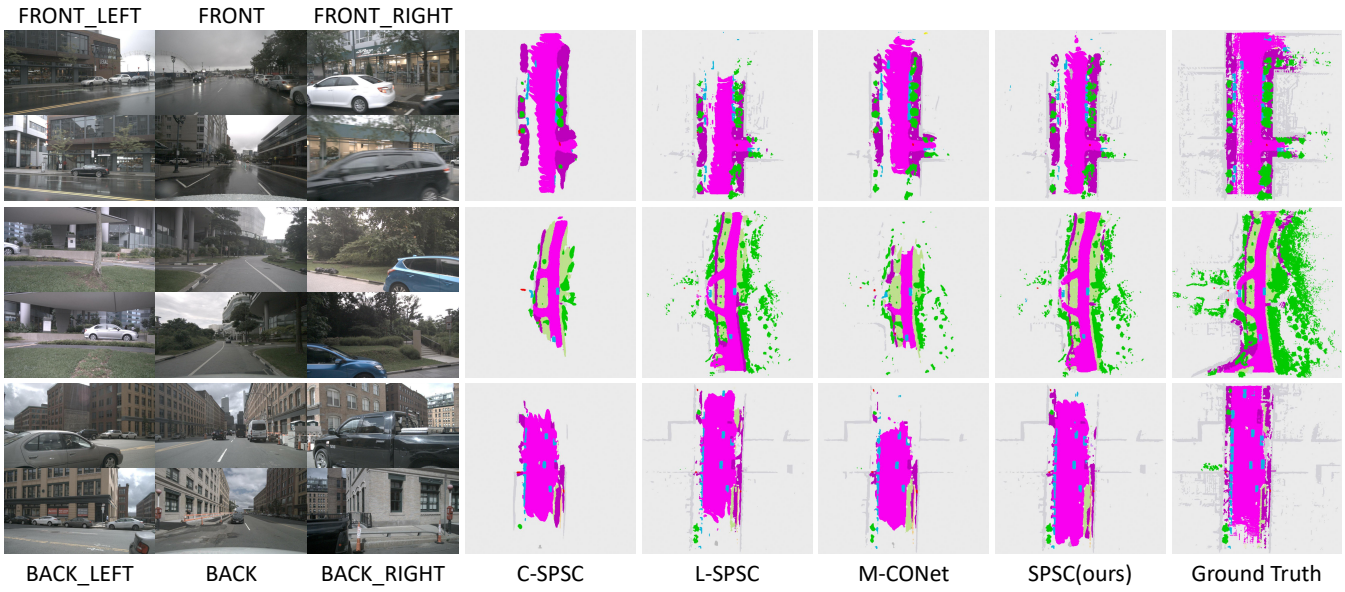

Figure 5: Qualitative Results of SPSC and M-CONet (Wang et al. 2023) on nuScenes-Occupancy. SPSC achieves promising scene completion results using LiDAR data alone, and further enhances model performance by leveraging multimodal data.

Top-K queries	160k	240k	320k	400k	480k	560k
Latency(ms)	298	306	316	327	340	355
Memory(G)	2.9	3.1	3.1	3.2	3.3	3.4
IoU	28.3	31.4	33.8	34.8	35.5	**35.8**
mIoU	25.2	26.4	27.2	27.6	**28.1**	27.8

Table 5: Analysis of Top-K queries. Memory and latency represent GPU memory usage and model inference time, respectively. All tests are conducted on a single 4090 GPU.

Window size	sp-conv	32	128	512	1024	2048
IoU	30.3	31.6	32.8	34.3	**35.5**	35.4
mIoU	24.3	26.8	27.1	27.8	**28.1**	28.0

Table 6: Analysis of window size. Sp-conv denotes the use of sparse convolution-implemented basic residual blocks to replace the attention blocks (He et al. 2016), while keeping the remaining structure unchanged.

ence, inference latency, and final occupancy prediction results. SPSC achieves significant performance improvements with fewer parameters, highlighting the importance of fine-grained feature interaction. The efficiency of query serialization enables SPSC to maintain low inference latency while performing feature interactions at a larger scale.

Qualitative Results. Fig. 5 provides a visual comparison of 3D semantic occupancy predictions between SPSC and the baseline method M-CONet (Wang et al. 2023) on nuScenes-Occupancy val set. Our LiDAR-based approach demonstrates superior scene completion capabilities, while the introduction of the camera modality further enhances the model's semantic understanding of the scene, particularly for foreground objects. These results highlight the effectiveness of SPSC in delivering more accurate and refined 3D occupancy predictions.

Method	Param.	Mem.T	Mem.I	Latency	IoU	mIoU
C-CONet	114.4M	22.0G	4.5G	353ms	20.1	12.8
C-SPSC	81.3M	23.9G	3.9G	319ms	24.3	15.2
L-CONet	65.6M	8.5G	3.7G	244ms	30.9	15.8
L-SPSC	39.3M	19.3G	2.9G	266ms	36.6	25.2
M-CONet	125.4M	24.0G	4.8G	387ms	29.5	20.1
SPSC	65.1M	21.9G	3.3G	340ms	35.5	28.1

Table 7: Model efficiency. Param. denote the trainable parameters of the model. Mem.T, and Mem.I denote the GPU memory usage during training and inference, respectively.

Conclusion

In this work, we present SPSC, a SParse and SCalable multi-modal framework, to address the challenges arising from the inherent sparsity of real-world scenes and the massive scale of high-resolution data in 3D semantic occupancy prediction. SPSC achieves solid performance with adequate inference latency, demonstrating its effectiveness.

Limitations & future work. The coarse-to-fine architecture implies that queries erroneously discarded at the coarse level cannot be recovered in subsequent processing stages. In the future, we will focus on introducing compensation mechanisms to prevent the accumulation of errors. Moreover, a fixed receptive field is not optimal for real-world scenes where object volumes vary significantly, and adaptive receptive fields remain to be explored in future work.

Acknowledgements

This paper was supported by the National Natural Science Foundation of China (No. 62376026, 62388101, 42571343) and Beijing Nova Program (No. 20230484296).

References

Balkić, Z.; Šoštarić, D.; and Horvat, G. 2012. GeoHash and UUID identifier for multi-agent systems. In *Agent and Multi-Agent Systems. Technologies and Applications: 6th KES International Conference, KES-AMSTA 2012, Dubrovnik, Croatia, June 25-27, 2012. Proceedings 6*, 290–298. Springer.

Beltagy, I.; Peters, M. E.; and Cohan, A. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Berman, M.; Triki, A. R.; and Blaschko, M. B. 2018. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *CVPR*, 4413–4421.

Cao, A.-Q.; and De Charette, R. 2022. Monoscene: Monocular 3d semantic scene completion. In *CVPR*, 3991–4001.

Chu, X.; Tian, Z.; Zhang, B.; Wang, X.; and Shen, C. 2021. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*.

Cui, Y.; Li, Z.; Wang, J.; and Fang, Z. 2024. LOMA: Language-assisted Semantic Occupancy Network via Triplane Mamba. *arXiv preprint arXiv:2412.08388*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255. Ieee.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hilbert, D. 1935. Über die stetige Abbildung einer Linie auf ein Flächenstück. *Dritter Band: Analysis· Grundlagen der Mathematik· Physik Verschiedenes: Nebst Einer Lebensgeschichte*, 1–2.

Huang, J.; Huang, G.; Zhu, Z.; Ye, Y.; and Du, D. 2021. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*.

Huang, Y.; Zheng, W.; Zhang, Y.; Zhou, J.; and Lu, J. 2023. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *CVPR*, 9223–9232.

Li, H.; Hou, Y.; Xing, X.; Sun, X.; and Zhang, Y. 2024a. OccMamba: Semantic Occupancy Prediction with State Space Models. *arXiv preprint arXiv:2408.09859*.

Li, Q.; Wang, Y.; Wang, Y.; and Zhao, H. 2022. Hdmapnet: An online hd map construction and evaluation framework. In *ICRA*, 4628–4634. IEEE.

Li, Y.; Bao, H.; Ge, Z.; Yang, J.; Sun, J.; and Li, Z. 2023a. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo. In *AAAI*, volume 37, 1486–1494.

Li, Y.; Ge, Z.; Yu, G.; Yang, J.; Wang, Z.; Shi, Y.; Sun, J.; and Li, Z. 2023b. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *AAAI*, volume 37, 1477–1485.

Li, Z.; Wang, W.; Li, H.; Xie, E.; Sima, C.; Lu, T.; Yu, Q.; and Dai, J. 2024b. Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *TPAMI*.

Liang, J.; Yin, H.; Qi, X.; Park, J. J.; Sun, M.; Madhivanan, R.; and Manocha, D. 2024. ET-Former: Efficient Triplane Deformable Attention for 3D Semantic Scene Completion From Monocular Camera. *arXiv preprint arXiv:2410.11019*.

Liang, J.-Y.; Chen, C.-S.; Huang, C.-H.; and Liu, L. 2008. Lossless compression of medical images using Hilbert space-filling curves. *Computerized Medical Imaging and Graphics*, 32(3): 174–182.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature pyramid networks for object detection. In *CVPR*, 2117–2125.

Liu, H.; Chen, Y.; Wang, H.; Yang, Z.; Li, T.; Zeng, J.; Chen, L.; Li, H.; and Wang, L. 2024a. Fully sparse 3d occupancy prediction. In *ECCV*, 54–71. Springer.

Liu, H.; Teng, Y.; Lu, T.; Wang, H.; and Wang, L. 2023. Sparsebev: High-performance sparse 3d object detection from multi-camera videos. In *ICCV*, 18580–18590.

Liu, X.; Wang, S.; Li, W.; Yang, R.; Chen, J.; and Zhu, J. 2024b. Mgmap: Mask-guided learning for online vectorized hd map construction. In *CVPR*, 14812–14821.

Liu, Y.; Wang, T.; Zhang, X.; and Sun, J. 2022. Petr: Position embedding transformation for multi-view 3d object detection. In *ECCV*, 531–548. Springer.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 10012–10022.

Lu, M.; Huang, Y.; Liu, J.; Huang, X.; Li, D.; Peng, J.; Tian, L.; and Barsoum, E. 2024. Fast Occupancy Network. *arXiv preprint arXiv:2412.07163*.

Lu, Y.; Zhu, X.; Wang, T.; and Ma, Y. 2025. Octreeocc: Efficient and multi-granularity occupancy prediction using octree queries. *Advances in Neural Information Processing Systems*, 37: 79618–79641.

Morton, G. M. 1966. A computer oriented geodetic data base and a new technique in file sequencing. *physics of plasmas*.

Pan, J.; Wang, Z.; and Wang, L. 2024. Co-occ: Coupling explicit feature fusion with volume rendering regularization for multi-modal 3d semantic occupancy prediction. *IEEE Robotics and Automation Letters*.

Pan, X.; Xia, Z.; Song, S.; Li, L. E.; and Huang, G. 2021. 3d object detection with pointformer. In *CVPR*, 7463–7472.

Parmar, N.; Vaswani, A.; Uszkoreit, J.; Kaiser, L.; Shazeer, N.; Ku, A.; and Tran, D. 2018. Image transformer. In *ICML*, 4055–4064. PMLR.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Philion, J.; and Fidler, S. 2020. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, 194–210. Springer.

Qiao, L.; Ding, W.; Qiu, X.; and Zhang, C. 2023. End-to-end vectorized hd-map construction with piecewise bezier curve. In *CVPR*, 13218–13228.

Shi, Y.; Jiang, K.; Wang, K.; Qian, K.; Wang, Y.; Li, J.; Wen, T.; Yang, M.; Xu, Y.; and Yang, D. 2024. Effocc: A minimal baseline for efficient fusion-based 3d occupancy network. *arXiv preprint arXiv:2406.07042*.

Tang, P.; Wang, Z.; Wang, G.; Zheng, J.; Ren, X.; Feng, B.; and Ma, C. 2024. Sparseocc: Rethinking sparse latent representation for vision-based semantic occupancy prediction. In *CVPR*, 15035–15044.

Wang, G.; Wang, Z.; Tang, P.; Zheng, J.; Ren, X.; Feng, B.; and Ma, C. 2024a. Occgen: Generative multi-modal 3d occupancy prediction for autonomous driving. In *ECCV*, 95–112. Springer.

Wang, J.; Liu, Z.; Meng, Q.; Yan, L.; Wang, K.; Yang, J.; Liu, W.; Hou, Q.; and Cheng, M. 2024b. Opus: occupancy prediction using a sparse set. In *NeurIPS*.

Wang, J.; Yin, W.; Long, X.; Zhang, X.; Xing, Z.; Guo, X.; and Zhang, Q. 2024c. OccRWKV: Rethinking Efficient 3D Semantic Occupancy Prediction with Linear Complexity. *arXiv preprint arXiv:2409.19987*.

Wang, P.-S. 2023. Octformer: Octree-based transformers for 3d point clouds. *TOG*, 42(4): 1–11.

Wang, X.; Zhu, Z.; Xu, W.; Zhang, Y.; Wei, Y.; Chi, X.; Ye, Y.; Du, D.; Lu, J.; and Wang, X. 2023. Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. In *ICCV*, 17850–17859.

Wang, Y.; Guizilini, V. C.; Zhang, T.; Wang, Y.; Zhao, H.; and Solomon, J. 2022. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, 180–191. PMLR.

Wu, X.; Jiang, L.; Wang, P.-S.; Liu, Z.; Liu, X.; Qiao, Y.; Ouyang, W.; He, T.; and Zhao, H. 2024. Point transformer v3: Simpler faster stronger. In *CVPR*, 4840–4851.

Yan, Y.; Mao, Y.; and Li, B. 2018. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10): 3337.

Yin, T.; Zhou, X.; and Krahenbuhl, P. 2021. Center-based 3d object detection and tracking. In *CVPR*, 11784–11793.

Yu, Z.; Shu, C.; Deng, J.; Lu, K.; Liu, Z.; Yu, J.; Yang, D.; Li, H.; and Chen, Y. 2023. Flashocc: Fast and memory-efficient occupancy prediction via channel-to-height plugin. *arXiv preprint arXiv:2311.12058*.

Yu, Z.; Shu, C.; Sun, Q.; Bian, Y.; Wei, X.; Yu, J.; Liu, Z.; Yang, D.; Li, H.; and Chen, Y. 2024. Panoptic-flashocc: An efficient baseline to marry semantic occupancy with panoptic via instance center. *arXiv preprint arXiv:2406.10527*.

Zhang, J.; Ding, Y.; and Liu, Z. 2024a. Occfusion: Depth estimation free multi-sensor fusion for 3d occupancy prediction. In *ACCV*, 3587–3604.

Zhang, J.; Ding, Y.; and Liu, Z. 2024b. OccLoff: Learning Optimized Feature Fusion for 3D Occupancy Prediction. *arXiv preprint arXiv:2411.03696*.

Zhou, Y.; and Tuzel, O. 2018. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, 4490–4499.