ViewFormer: Exploring Spatiotemporal Modeling for Multi-View 3D Occupancy Perception via View-Guided Transformers

Jinke Li[1], Xiao He[1], Chonghua Zhou[2], Xiaoqiang Cheng[1], Yang Wen[1], and Dan Zhang[1]

[1] Uisee Foundation Research & Development, Beijing, China
[2] University of Science and Technology of China, China

Abstract. 3D occupancy, an advanced perception technology for driving scenarios, represents the entire scene without distinguishing between foreground and background by quantifying the physical space into a grid map. The widely adopted projection-first deformable attention, efficient in transforming image features into 3D representations, encounters challenges in aggregating multi-view features due to sensor deployment constraints. To address this issue, we propose our learning-first view attention mechanism for effective multi-view feature aggregation. Moreover, we showcase the scalability of our view attention across diverse multi-view 3D tasks, including map construction and 3D object detection. Leveraging the proposed view attention as well as an additional multi-frame streaming temporal attention, we introduce ViewFormer, a vision-centric transformer-based framework for spatiotemporal feature aggregation. To further explore occupancy-level flow representation, we present FlowOcc3D, a benchmark built on top of existing high-quality datasets. Qualitative and quantitative analyses on this benchmark reveal the potential to represent fine-grained dynamic scenes. Extensive experiments show that our approach significantly outperforms prior state-of-the-art methods. The codes are available at `https://github.com/ViewFormerOcc/ViewFormer-Occ`.

Keywords: 3D Occupancy · Occupancy flow · Multi-view Interaction · Spatiotemporal Modeling · Streaming video pipeline

1 Introduction

The vision-centric autonomous driving (AD) systems are attracting extensive attention in recent years, promoting the research domain perceiving the real 3D world from 2D images. 3D object detection is a traditional task, representing foreground objects with limited categories. As illustrated in Fig. 1(a), the pedestrian is detected as a 3D bounding box, while the uncommon suitcase is hardly defined in driving scenarios. Once such a suitcase appears on the road, an AD system relying only on object detection is unable to secure the driving safety. In contrast, the 3D occupancy representation, unifying the concept of foreground and background, and quantifying the entire 3D space into voxel-wise cells with

semantic labels, shows superior performance, where objects like suitcases can be effectively defined as category-agnostic occupancies as shown in Fig. 1(b). Beyond static occupancy, occupancy flow is crucial for representing dynamic scenes as shown in Fig. 1(d). In this paper, we introduce *ViewFormer*, a transformer-based framework designed to predict 3D occupancy and occupancy flow with multi-camera images as input.

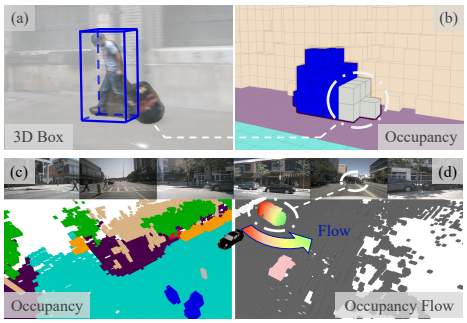

Fig. 1: Treating objects simply as 3D boxes lacks the sense of the background, such as the suitcase in (**a**). Defining 3D space as occupancies (**b**) and (**c**) is more effective in representing objects. Beyond static occupancy, occupancy flow is crucial to perceive dynamic scenes. In the case of a turning car in (**d**), different flow directions of occupancies can be clearly observed.

Typical vision-centric 3D perception frameworks can be decomposed into spatial interaction and temporal interaction components. The former is responsible for transforming image features into 3D space, while the latter, on one hand, enhances the 3D features, and on the other hand, models dynamic scenes to reason velocity information. Regarding spatial interaction, to reduce computational costs, sparse methods have been explored to transform multi-view image features into 3D space [18,27,39]. For instance, deformable attention [42], originally designed for monocular images, is extended to the multi-view field by BEVFormer [18] through projecting 3D reference points onto multiple images, which is referred to as the projection-first method in this paper. This method is widely used to aggregate multi-view features and predict 3D occupancy [13,19,31]. Nevertheless, we identify two predominant issues of this approach.

To gain a better understanding, let's review the projection-first method illustrated in Fig. 2(a), which projects the 3D reference point of a voxel query onto images first and then performs deformable attention [42]. However, a critical limitation arises when a 3D reference point, fixed during training, is projected outside the image size for a specific camera, the projection-first method no longer applies deformable attention to extracting features for this reference point. As a consequence, features from this camera are masked out throughout the entire dataset. Notably, this issue is widespread, as seen in scenarios like nuScenes [5,9], where numerous 3D reference points can only be projected onto a single camera due to sensor deployment. Additionally, as the deformable attention utilized in the method learns sample points on the image plane, the 2D sample area corresponding to a 3D object undergoes rapid variations with changes in depth due to perspective transformation, such scale inconsistency poses a convergence challenge to 3D perception tasks.

To address the aforementioned issues, we introduce our learning-first *view attention*, facilitating multi-view feature aggregation. As shown in Fig. 2(b), to

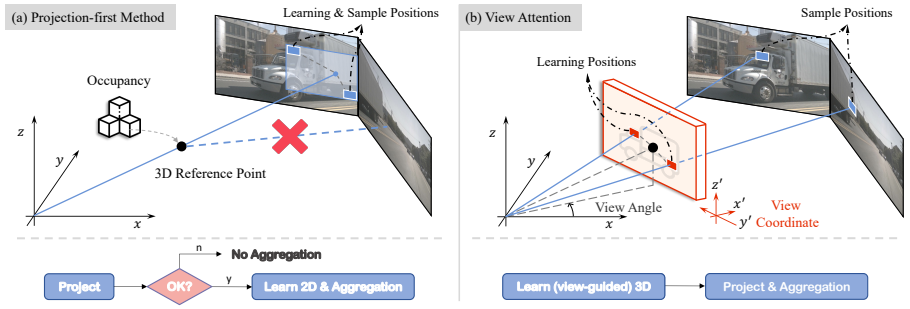

Fig. 2: Constrained by fixed reference points, the projection-first method (**a**) introduced in [18] fails to collect multi-view features. In contrast, our learning-first view attention (**b**) gathers features from multiple cameras more adequately.

overcome the constraint imposed by fixed reference points on feature collection, we adopt a strategy to learn local regions in 3D space for a given query. The corresponding 3D points of these regions are then projected onto multiple images for feature aggregation. Consequently, the extraction of features across cameras becomes a data-driven process. For higher efficiency and faster convergence, we define the learned 3D regions in a local view coordinate (VC) system, providing view guidance. These regions remain invariant as the query's view angle changes, introducing effective rotational invariance in perception around the vehicle. Furthermore, learning regions directly in 3D space preserves a consistent 3D spatial scale, avoiding the challenges correlated to perspective transformation.

Temporal modeling in computer vision often emphasizes the efficiency of leveraging video data [4, 24, 28]. The popular multi-camera temporal method BEVFormer [18] achieves temporal interaction solely with a single historical frame, leading to limited performance. Besides, in BEVFormer, employing a sliding window approach during training and switching to online video during inference brings inconsistency, and the use of windowed data also increases training time due to redundant inference of many frames. Therefore, we propose our streaming temporal attention mechanism with multi-frame interaction, in which the utilization of a streaming memory mechanism [26, 35] significantly reduces training time without additional inference latency.

In aspect of dataset, although prior work [31] delves into object-level occupancy flow by assigning the center velocity of an object to all its internal occupancies, the exploration of occupancy-level flow representation remains limited. As shown in Fig. 1(d), fine-grained occupancy flow provides more detailed information such as motion directions for different parts of a turning car. Furthermore, occupancy-level flow has the potential to represent objects whose shapes vary in motion for future research. Hence, we create our *FlowOcc3D* dataset, building upon nuScenes [5, 9] and Occ3D [34] datasets, featuring fine-grained occupancy-level flow annotations.

In summary, our contributions are four main aspects: 1) We identify limitations of the widely used projection-first method and propose our view attention to more effectively transform multi-view features into 3D space. We demonstrate its scalability across various multi-view 3D tasks, which can be new baselines for future multi-view 3D perception research. 2) We introduce ViewFormer, a vision-centric transformer-based framework that incorporates the novel view attention and multi-frame streaming temporal attention, enchancing spatiotemporal modeling for multi-view 3D perception. 3) We create FlowOcc3D, a high-quality occupancy-level flow benchmark. Qualitative and quantitative analyses are conducted to compare models trained on occupancy-flow and object-flow, revealing the potential of fine-grained representation in dynamic scenes. 4) ViewFormer achieves state-of-the-art performance across diverse benchmarks, surpassing previous methods by substantial margins.

2 Related Work

2D Image to 3D Space Transformation. To transform 2D image features to 3D space, bottom-up methods [17,27,29] rely on pixel-wise depth prediction, where the limited reconstruction density due to corresponding image resolution poses a challenge for the demanding dense 3D occupancy representation. In contrast, top-down transformer-based methods [18,23,39] are more flexible by allowing arbitrary predefined resolutions for 3D space and implementing feature transformation through query-to-feature interactions. For transformer-based approaches, sparse deformable attention [18] is particularly suitable to handle a large number of occupancy instances with low computational overhead. However, we find that extending the monocular deformable attention to multi-view tasks through the projection-first method [18], which is also widely adopted in [13,19,20,31], presents shortcomings, motivating us to explore our view attention to adequately collect image features for multi-view AD systems.
3D Scene Reconstruction. The occupancy network, introduced by Tesla [1], brings the concept of the occupancy grid map, which has long been utilized in the domains of robotic mapping and planning [8,30], into the AD field. Semantic scene completion [32] is similar to the goals of occupancy tasks discussed in this paper. MonoScene [6] leverages a U-Net architecture to infer dense 3D occupancy from a single image. VoxFormer [16] introduces depth estimation to guide voxel queries, and OccDepth [25] adopts stereo depth information to improve occupancy prediction. TPVFormer [13] proposes a tri-perspective view representation to aggregate image features. FB-OCC [19] constructs 3D features via forward-backward transformations. In addition, [31,38] contribute high-quality occupancy benchmarks to facilitate the research community. Although [31] assigns coarse object-level velocity directly to object occupancies, fine-grained occupancy-level flow representation still remain unexplored, by which we are motivated to present our occupancy-level flow dataset FlowOcc3D, as well as a feasible framework based on our view attention and streaming temporal attention for motion 3D occupancy prediction.

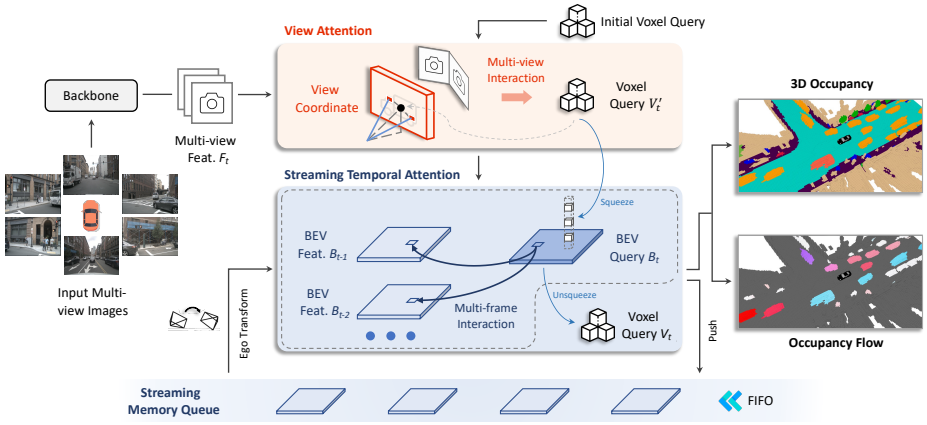

Fig. 3: ViewFormer pipeline. In our ViewFormer, the multi-view features F_t are first extracted from the multiple images via a backbone. Then we introduce the view attention specific for addressing the limitations of the existing projection-first method, allowing us to aggregate multi-view features for voxels V_t' more adequately. In our streaming temporal attention, we squeeze the voxel queries V_t' into the BEV queries B_t with concern of the computing complexity. Each BEV cell of B_t interacts with historical multi-frame BEV features stored in the streaming memory queue, where we utilize ego transformation to compensate ego motion. The voxels V_t obtained from unsqueezing the updated BEV features are subsequently fed into 3D occupancy and occupancy flow prediction. We push the updated BEV queries into the memory queue for subsequent temporal interaction in the video stream pipeline.

3 Methodology

In this paper, we propose a general framework, named ViewFormer, gathering spatiotemporal features from multi-view images adequately for unified prediction of 3D occupancy and occupancy flow, as shown in Fig. 3. Our approach stands out due to two key components: the view attention and streaming temporal attention, both of which compose a transformer encoder, wherein queries circulate in the form of voxels and BEV cells, ensuring efficient computation while extracting fine-grained representative 3D features.

Spatial Interaction. Following bird's-eye-view (BEV) perception methods [18, 23, 39], we extract multi-view image features F_t via an image backbone and then update queries through spatial interaction. The difference is that we directly use voxel queries V_t' instead of BEV queries to collect finer-grained 3D occupancy features, where a relatively small number of query channels is set to reduce computational effort for a large number of voxel queries. Serving as the core of spatial interaction, our view attention is introduced specific for transforming multi-view image features into 3D space, exceeding a simple extension of the 2D deformable attention commonly used in [13, 18, 19, 31].

Temporal Modeling. Inspired by the concept of streaming video methods [26, 35], we construct a streaming memory queue to dynamically store historical features spanning N frames during both the training and inference phases, which follows the first-in, first-out (FIFO) principle [35] for entry and exit. Considering the increased burden of storage and computation, we proceed with temporal modeling at the 2D BEV level. Specifically, the voxel queries V_t' are squeezed into BEV-level queries B_t along the z-axis, each BEV cell of B_t interacts with the historical multi-frame BEV features stored in the memory queue, where we utilize ego transformation to compensate ego motion. The voxels V_t obtained from unsqueezing the updated BEV queries are then fed into 3D occupancy and occupancy flow prediction. Meanwhile, we push the updated BEV queries into the memory queue for subsequent temporal interaction in the video stream pipeline.

3.1 View Attention

Faced with the challenge of 3D scenarios, we employ dense voxel queries ($V_t \in \mathbb{R}^{Z\times H\times W\times C_{\text{Voxel}}}$) as containers to facilitate fine-grained voxel feature extraction throughout the entire pipeline. Here Z, H, and W represent the 3D dimensions of the space, and C_{Voxel} denotes the feature channels. To effectively construct spatial 3D features with reasonable receptive fields in a vision-centric AD system, the keystone is to aggregate coherent 3D features from discrete multi-view images, for which we introduce our learning-first view attention mechanism to address the limitations of the existing projection-first method, as discussed Sec. 1.

Our view attention is implemented by abstracting an occupancy-related view coordinate (VC) system $\mathcal{T}$ as depicted in Fig. 2(b). Given multi-view features F_t, a voxel query $\boldsymbol{q} \in \mathbb{R}^{C_{\text{Voxel}}}$ and its corresponding fixed 3D reference point $\boldsymbol{p}$, view attention can be formulated as:

$$\text{ViewAttn}(\boldsymbol{q}, \boldsymbol{p}, F_t) = \sum_{m=1}^{M} W_m \sum_{k=1}^{K}\sum_{j=1}^{J} A_{mkj} W_m' F_t(\boldsymbol{p} + \Delta\boldsymbol{p}_{mk}^{\mathcal{T}}), \tag{1}$$

where M, K and J are the number of attention heads, sample points and cameras respectively, $W_m \in \mathbb{R}^{C_{\text{Voxel}}\times C_v}$, $W_m' \in \mathbb{R}^{C_v\times C_{\text{Voxel}}}$ are the learning weights ($C_v = C_{\text{Voxel}}/M$ by default), A_{mkj} is the normalized attention weight, $\Delta\boldsymbol{p}_{mk}^{\mathcal{T}}$ denotes a learnable sample point associated with query $\boldsymbol{q}$ in the VC system $\mathcal{T}$, and $F_t(\cdot)$ represents extracting features from F_t by projecting the learnable sample points onto multi-view images.

Now, let's delve into how we represent learnable sample points in the view-guided VC system $\mathcal{T}$. Given the coordinate (x, y, z) of a reference point $\boldsymbol{p}$ for a query $\boldsymbol{q}$, we define its view angle θ as the angle between its projection line on the x-y plane and the x-axis, as illustrated in Fig. 2(b). The VC system $\mathcal{T}$ of this query can be obtained by translating the origin of the ego-centric perception coordinate system to the reference point and rotating it around the z-axis by the view angle. The corresponding rotation matrix $\boldsymbol{R}(\theta)$ are calculated as:

$$\theta = \arctan2\left(y, x\right), \\ \boldsymbol{R}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

Through a transformation process, a learnable sample point $\Delta \boldsymbol{p}^{\mathcal{T}}$, initially defined in the local VC system $\mathcal{T}$, can be converted to be a 3D sample point $\boldsymbol{p}_{\mathrm{s}}$ in the perception coordinate system, which is denoted as:

$$\boldsymbol{p_s} = \boldsymbol{p} + \boldsymbol{R}(\theta) \cdot \Delta \boldsymbol{p}^{\mathcal{T}}. \tag{3}$$

Subsequently, we project all the 3D sample points associated with a query onto the multi-view images via camera projection matrices for multi-view spatial interaction, aggregating features from F_t into voxel queries V_t'.

3.2 Streaming Temporal Attention

Streaming Memory Queue. Drawing inspiration from the efficient 3D point cloud perception [14,15], where it is observed that fine-grained 3D feature representation is not necessarily required for 3D space, we recognize that combining the voxel level and the BEV level can significantly enhance computation efficiency and alleviate storage consumption. Therefore, we establish our BEV-level streaming memory queue $\boldsymbol{B} = \{B_i \in \mathbb{R}^{H \times W \times C_{\mathrm{BEV}}}, i = t-1, ..., t-N\}$ with $C_{\mathrm{BEV}} > C_{\mathrm{Voxel}}$, to dynamically store historical features spanning N frames during both the training and inference phases. The latest BEV features are pushed into the memory queue per frame for later BEV-level temporal interaction.

Multi-frame Temporal Interaction. We perform a multi-frame streaming temporal interaction. We observe that, compared to voxel queries V_t' directly interacting with all historical BEV features in the memory queue, implementing BEV-level temporal interaction not only reduces computational overhead but also improves accuracy. Specifically, we first compress the voxel queries V_t' into BEV queries $B_t \in \mathbb{R}^{H \times W \times C_{\mathrm{BEV}}}$ along the z-axis, of which each BEV cell interacts with all the memory BEV features by leveraging the deformable attention [42], and the updated BEV queries are then recovered to be the voxel queries V_t through a feed-forward function for final predictions. We attribute this improvement to the presence of a substantial number of empty voxels, the compressed BEV queries B_t lead to more pure information and thus enhance the temporal interaction.

To compensate ego motion, we apply feature warping by transforming all memory BEV features to the current frame as in [11, 17, 26]. Taking the last frame $t-1$ as an example, given the ego pose matrices of the last frame T_{t-1} and the current frame T_t, the transformation matrix T_{t-1}^{t} between two frames is calculated as follow:

$$T_{t-1}^{t} = T_{t}^{inv} \cdot \ T_{t-1}. \tag{4}$$

We align the last BEV features B_{t-1} to the current frame:

$$\tilde{B}_t = T_{t-1}^t \cdot \ B_{t-1}, \tag{5}$$

where $\tilde{B}_t$ is the aligned BEV features in the local coordinate system of the current ego pose. Then, we adopt deformable attention [42] to achieve interaction between the current BEV queries B_t and the aligned multi-frame BEV features $\tilde{\boldsymbol{B}}_{\boldsymbol{t}}$ for the final updated BEV queries. The multi-frame aggregation mechanism follows a multi-scale aggregation [42].

4 Optimization

4.1 Occupancy Flow Generation

Although OpenOcc [31] has already generated object-level flow annotations by assigning the center velocity of an object to all its internal occupancies, the occupancy-level flow representation still remains unexplored. As illustrated in Fig. 4, fine-grained occupancy flow is able to capture more accurate flow vectors for different parts of a turning vehicle, offering the potential for a more precise representation of dynamic scenes. This capability is beneficial for decision-making in an AD system. Hence, we build our FlowOcc3D dataset with occupancy-level flow annotations. Fig. 4(e) illustrates the generation process of flow annotations. As nuScenes [5, 9] does not provide explicit temporal association for occupancies themselves, we leverage the temporal association of objects to track their internal occupancies. Given the pose matrices O_t and O_{t-1} of an object in two frames, along with an occupancy center p_t within the object box at O_t, where O_t, O_{t-1} and p_t are all defined in the global coordinate system, by taking the box as a rigid body, we map p_t to obtain its historical position p_{t-1} in the temporal frame $t-1$, via $p_{t-1} = O_{t-1} \cdot O_t^{inv} \cdot p_t$. Then, the corresponding flow vector can be computed by $f = (p_t - p_{t-1})/\Delta t$, where Δt denotes the time interval.

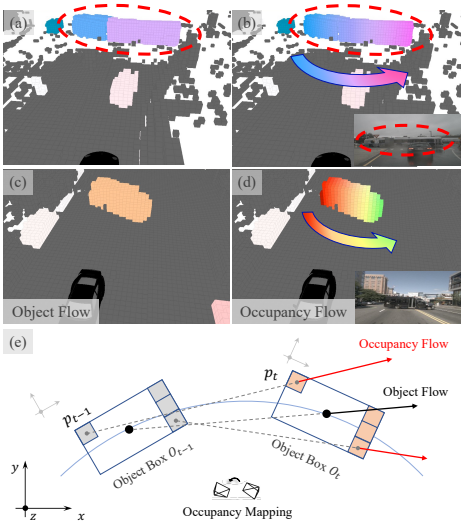

Fig. 4: Occupancy flow vs. object flow. Object flow assigns only a single flow vector to the entire object as in (**a**) and (**c**), while occupancy flow provides finer-grained flow vectors for all occupancy grids as in (**b**) and (**d**), where the color and brightness represent the flow direction and magnitude respectively.

4.2 Loss Function

Our final loss function is comprised of four parts, the focal loss [21] $\mathcal{L}_{focal}$ for supervising occupancy state, the cross-entropy loss $\mathcal{L}_{ce}$ as well as the Lovasz softmax loss [3] $\mathcal{L}_{ls}$ for semantic classification, and the L1 loss $\mathcal{L}_{l1}$ with λ to adjust loss weight for flow regression, which can be formulated as:

$$\mathcal{L} = \mathcal{L}_{focal} + \mathcal{L}_{ce} + \mathcal{L}_{ls} + \lambda \mathcal{L}_{l1}. \tag{6}$$

4.3 Implementation Details

Network. Following the experimental setup [18,31], we employ two backbones: ResNet50 (Res50) [10] initialized from ImageNet [7], and ResNet101 (Res101) [10] initialized from FCOS3D [36] for experiments in Tab. 1 and Tab. 2. In respect of experiments in Tab. 3 and the ablation section, we utilize InternImage-Tiny (InterT) [37] initialized from COCO [22] as image backbone. Our transformer encoder has 4 layers. For the detection space of the voxel queries V_t, we define the dimensions as: $H = 100$, $W = 100$, $Z = 8$, and $C_{\text{Voxel}} = 72$, while the channels of the BEV queries B_t are set to $C_{\text{BEV}} = 126$. We use $N = 4$ temporal frames for the streaming memory queue.
Training and Inference. By default, we train our ViewFormer with a streaming video approach [26,35] for 24 epochs. The learning rate is set to 2×10^{-4}, and the image size is 256×704. We also apply interpolation method on predictions to align with ground truth whose resolutions are inconsistent with our detection space. For the ease of comparison among different methods, we predict and evaluate BEV-level flow, and for visualization, we map the BEV-level flow to each voxel cell to recover occupancy flow. Inference FPS is measured with a single RTX 3090 GPU, while the training time is recorded using 8 A100 GPUs.

5 Evaluation

To evaluate the performance of our method comprehensively, we utilize the high-quality 3D occupancy benchmarks Occ3D [34] and OpenOcc [31]. Moreover, based on the nuScenes [5,9] and Occ3D [34] datasets, we build our FlowOcc3D with occupancy-level flow annotations to study the potential of fine-grained occupancy flow representation, on which we further conduct extensive experiments.

5.1 Datasets

Both of the Occ3D [34] and OpenOcc [31] occupancy datasets are derived from nuScenes [5,9], a substantial driving dataset that comprises videos of 1000 scenes equipped with multi-view cameras. It includes 34,149 annotated frames for all 700 training scenes and 150 validation scenes.
Benchmark Details. Occ3D [34] defines the detection space $\mathcal{V} = [-40\text{m}, 40\text{m}] \times [-40\text{m}, 40\text{m}] \times [-1\text{m}, 5.4\text{m}]$, in the ego vehicle coordinate system to generate

Table 1: 3D Occupancy Prediction on Occ3D benchmark. Our ViewFormer outperforms previous SOTAs by significant margins.

Method	Backbone	mIoU	others	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	driv. surf.	other flat	sidewalk	terrain	manmade	vegetation
MonoScene [6]	Res101	6.06	1.75	7.23	4.26	4.93	9.38	5.67	3.98	3.01	5.90	4.45	7.17	14.91	6.32	7.92	7.43	1.01	7.65
TPVFormer [13]	Res101	27.83	7.22	38.90	13.67	40.78	45.90	17.23	19.99	18.85	14.30	26.69	34.17	55.65	35.47	37.55	30.70	19.40	16.78
OccFormer [40]	Res101	21.93	5.94	30.29	12.32	34.40	39.17	14.44	16.45	17.22	9.27	13.90	26.36	50.99	30.96	34.66	22.73	6.76	6.97
BEVFormer [18]	Res101	26.88	5.85	37.83	17.87	40.44	42.43	7.36	23.88	21.81	20.98	22.38	30.70	55.35	28.36	36.00	28.06	20.04	17.69
CTF-Occ [34]	Res101	28.53	8.09	39.33	20.56	38.29	42.24	16.93	24.52	22.72	21.05	22.98	31.11	53.33	33.84	37.98	33.23	20.79	18.00
FB-OCC [19]	Res50	39.11	**13.57**	44.74	27.01	**45.41**	49.10	**25.15**	26.33	27.86	27.79	32.28	36.75	80.07	42.76	51.18	55.13	42.19	37.53
ViewFormer (ours)	Res50	**41.85**	12.94	**50.11**	**27.97**	44.61	**52.85**	22.38	**29.62**	**28.01**	**29.28**	**35.18**	**39.40**	**84.71**	**49.39**	**57.44**	**59.69**	**47.37**	**40.56**

Table 2: 3D Occupancy Prediction on OpenOcc benchmark. Our ViewFormer demonstrates significant performance improvements over previous SOTAs in terms of both the mIoU and IoU_{geo}.

Method	Backbone	mIoU	IoU_{geo}	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	driv. surf.	other flat	sidewalk	terrain	manmade	vegetation
TPVFormer [13]	Res101	23.67	37.47	27.95	12.75	33.24	**38.70**	12.41	17.84	11.65	8.49	**16.42**	26.47	47.88	25.43	30.62	30.18	15.51	23.12
OccNet [31]	Res101	26.98	41.08	**29.77**	16.89	34.16	37.35	15.58	**21.92**	**21.29**	**16.75**	16.37	26.23	50.74	27.93	31.98	33.24	20.80	30.68
ViewFormer (ours)	Res101	**27.37**	**41.86**	28.18	**17.96**	**34.49**	37.03	**16.00**	21.53	19.50	13.56	15.59	**26.52**	**51.48**	**31.81**	**34.73**	**34.77**	**22.20**	**32.62**
BEVDet4D [11]	Res50	9.85	18.27	13.56	0.00	13.04	26.98	0.61	1.20	6.76	0.93	1.93	12.63	27.23	11.09	13.64	12.04	6.42	9.56
BEVDepth [17]	Res50	11.88	23.45	15.15	0.02	20.75	27.05	1.10	2.01	9.69	1.45	1.91	14.31	31.92	7.88	17.08	16.27	8.76	14.75
BEVDet [12]	Res50	12.49	27.46	16.06	0.11	18.27	21.09	2.62	1.42	7.78	1.08	3.4	13.76	33.89	10.84	17.55	22.03	11.72	18.15
OccNet [31]	Res50	19.48	37.69	20.63	5.52	24.16	27.72	9.79	7.73	13.38	7.18	10.68	18.00	46.13	20.6	26.75	29.37	16.90	27.21
ViewFormer (ours)	Res50	**23.84**	**40.40**	**24.35**	**12.55**	**28.87**	**31.87**	**15.35**	**17.89**	**14.76**	**8.53**	**14.18**	**23.54**	**49.02**	**29.05**	**32.35**	**32.16**	**17.93**	**29.00**

occupancy data. The space $\mathcal{V}$ is voxelized with a resolution of $\Delta s = 0.4$m, producing $200 \times 200 \times 16$ voxel grids to represent the 3D environment. Furthermore, the dataset includes occupancy visibility masks for various sensors, facilitating performance improvement and evaluation in diverse tasks. To create our **FlowOcc3D** dataset, we attach the occupancy-level flow information to the occupancy cells of Occ3D [34] for each annotated frame. **OpenOcc** [31] defines the detection space $\mathcal{V} = [-50\text{m}, 50\text{m}] \times [-50\text{m}, 50\text{m}] \times [-5\text{m}, 3\text{m}]$ in LiDAR coordinate system, and produces $200 \times 200 \times 16$ voxel grids by voxelizing $\mathcal{V}$ with a resolution of $\Delta s = 0.5$m.

Evaluation Metrics. Occ3D utilizes the mean Intersection-over-Union (mIoU) across categories. OpenOcc uses both the mIoU and the single-class IoU_{geo} as metrics to evaluate the occupancy state. Besides these two metrics, we additionally compute the mean absolute velocity error (mAVE) across categories to evaluate occupancy flow on our FlowOcc3D benchmark.

5.2 Main Results

3D Occupancy on Occ3D. We compare our ViewFormer with previous state-of-the-art methods on the 3D occupancy task in Tab. 1, where the baselines BEVFormer [18], TPVFormer [13], CTF-Occ [40] and FB-OCC [19] adopt the projection-first spatial interaction method as analyzed in Sec. 1. In terms of training setup, we follow FB-OCC [19], the winner of 3D occupancy challenge of CVPR 2023 [31, 34], which includes 90-epoch training, the same pretrained weights, and depth supervision for image backbone. As shown in Tab. 1, in con-

Table 3: 3D Occupancy and Occupancy Flow Prediction on FlowOcc3D. Our ViewFormer achieves the best result among previous SOTAs. *: we add a flow head for flow prediction to BEVFormer [18] and FB-OCC [19] and retrain them on FlowOcc3D.

Method	Backbone	Flow Head	mIoU	IoU_{geo}	mAVE↓	others	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	driv. surf.	other flat	sidewalk	terrain	manmade	vegetation
BEVFormer* [18]	InternT	✓	33.61	67.41	0.695	8.44	40.80	12.57	37.70	45.76	18.44	12.14	22.52	21.25	25.97	29.40	80.92	38.19	48.56	52.58	40.98	35.07
FB-OCC* [19]	InternT	✓	37.36	69.73	0.433	10.95	40.08	23.14	42.87	47.17	**21.43**	23.84	27.24	24.50	31.54	37.36	80.31	42.26	50.14	55.44	39.98	36.84
ViewFormer (ours)	InternT	✓	**42.54**	**72.36**	**0.412**	**13.63**	**49.35**	**28.74**	**46.63**	**52.71**	21.04	**29.63**	**30.34**	**30.53**	**34.01**	**41.04**	**85.23**	**50.63**	**58.68**	**61.63**	**47.72**	**41.56**
BEVFormer [18]	InternT	✗	36.93	68.49	-	8.56	42.94	19.34	47.02	49.59	20.36	22.62	24.69	19.77	30.11	35.24	82.11	40.86	50.62	54.81	42.56	36.55
FB-OCC [19]	InternT	✗	38.69	69.95	-	11.4	41.42	24.27	46.01	49.38	**24.56**	27.06	28.09	25.61	32.23	38.46	80.97	42.99	50.95	56.15	40.55	37.61
ViewFormer (ours)	InternT	✗	**43.61**	**72.46**	-	**13.82**	**50.32**	**29.49**	**49.24**	**54.52**	24.34	**32.72**	**31.09**	**31.49**	**34.44**	**41.62**	**85.47**	**51.27**	**59.03**	**62.15**	**48.33**	**42.06**

trast to the strong baseline FB-OCC [19], which utilizes longer video sequences, our model achieves 2.74 mIoU improvement, proving its superiority.

3D Occupancy on OpenOcc. We conduct experiments on OpenOcc in Tab. 2, where we adopt the same configuration as OccNet [31], with an image size of 450 $\times$ 800. Our approach demonstrates significant performance improvements over the baselines. Especially, we surpass the state-of-the-art OccNet [31] by 4.36 on mIoU and 2.71 on IoU_{geo} when utilizing Res50 as the backbone.

Occupancy and Flow on FlowOcc3D. We validate our ViewFormer against previous state-of-the-art methods on the 3D occupancy and flow tasks with our FlowOcc3D in Tab. 3, training both the occupancy and flow heads simultaneously. The two selected baselines, BEVFormer [18] and FB-OCC [19], employ two typical temporal modeling approaches that can directly affect the flow prediction. The former utilizes a transformer-based single-frame interaction trained in a sliding-window fashion, while the latter adopts a CNN-based streaming video interaction from [26]. We add a flow head to the two methods and retrain them on our FlowOcc3D for 24 epochs with an image size of 256 $\times$ 704, matching our ViewFormer setup. The results in Tab. 3 show that our method surpasses the two baselines in both the occupancy and flow prediction. It can be observed that, under the 24-epoch setup, our model outperforms FB-OCC by 5.2 mIoU, 2.63 IoU_{geo} and 2.1% mAVE, highlighting the rapid convergence of our method. We also report the results without the flow head under the same settings.

5.3 Ablations and Analysis

View Attention. To validate the effectiveness of our view attention, we conduct experiments as presented in Tab. 4. Since this module serves as a spatial interaction and is not involved in temporal modeling, we temporarily disable the flow head. The findings are as below. 1) We replace our view attention module with the projection-first deformable attention as BEVformer [18], the results show that our learning-first view attention leads to an improvement of 1.22 mIoU and 0.43 IoU_{geo} under approximate computational complexity, revealing the limitation of the commonly used projection-first method for multi-view feature aggregation. 2) Learning sample points directly in the ego-centric perception coordinate system instead of the query-specific view coordinate (VC)

Table 4: Ablation study for the view attention. "Projection-first" denotes the extension of deformable attention used in the multi-view field as [18]. "w/o VC trans." denotes learning sample points directly in the ego-centric perception coordinate system without view angle rotation.

Method	mIoU	IoU_{geo}	car	truck	vege.
View Attention	**43.61**	**72.46**	**54.52**	**41.62**	**42.06**
Projection-first	42.39	72.03	53.16	40.58	40.99
w/o VC trans.	42.86	72.26	53.30	40.70	41.93

system results in performance degradation of -0.75 mIoU, which validates our motivation mentioned in Sec. 1 that the rotation invariance introduced by our view attention effectively reduces learning complexity, accelerates convergence, and enhances accuracy. Detailed convergence experiments can be found in supplementary materials.

Visualization Analysis. We visualize our view attention in Fig. 5. The green cross mark denotes a query's reference point, which can only be projected onto the left image. The learned 3D points are projected onto multi-view images represented by filled circles, with colors to indicate attention weights. In such scenarios, the projection-first method, like BEVFormer [18] that only gathers features from the image where the reference point can be projected, fails to collect features from the right image, as discussed in Sec. 1. In contrast, our method is competent to gather features from both images, allowing for constructing more semantically informative and robust features for this query.

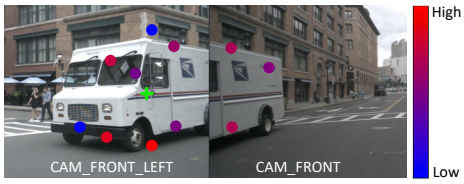

Fig. 5: Visualization on ViewAttn..

Application of View Attention. We also apply our view attention to other tasks to assess its scalability, including MapTR [20] for HD map construction and DETR3D [39] for 3D object detection, both of them collect multi-view image features in a similar way to the projection-first method analyzed in Sec. 1. We re-

Table 5: Results on multi-view map construction. We apply our view attention to MapTR [20].

Method	mAP	AP_{ped}	AP_{div}	AP_{bound}
MapTR	44.73	39.69	44.59	49.91
w/ ViewAttn	**50.55**	**45.85**	**52.82**	**52.98**

Table 6: Results on multi-view 3D object detection. We apply our view attention to DETR3D [39].

Method	mAP↑	NDS↑	mATE↓	mAVE↓
DETR3D	0.347	0.422	0.765	0.876
w/ ViewAttn	**0.388**	**0.441**	**0.712**	**0.874**

place their corresponding feature collection module with our view attention. The evaluation metrics are available in their respective literatures. We train MapTR for 24 epochs with an image size of 324×576, and Res50 as the backbone. Tab. 5 indicates our ViewAttn-MapTR achieves a substantial 5.82% improvement on mAP, demonstrating a remarkable increase in convergence speed. For training DETR3D, we use Res101-DCN as the backbone without CBGS [41]. Tab. 6 shows our ViewAttn-DETR3D also results in 4.1% and 1.9% improvements on mAP and NDS respectively. Unlike a considerable amount of recent researches mainly focusing on temporal modeling, and lacking the study of the fundamental spatial interaction for multi-camera 3D perception, our work reveals the limitations of the widely used projection-first method and highlights substantial research opportunities that remain unexplored.

Streaming Temporal Attention. We train both the occupancy head and the flow head simultaneously to evaluate our temproal modeling in Tab. 7. Overall, our streaming temporal attention improves the mIoU by 3.26 and produces more reasonable flow prediction. Notably, the "BEV-to-BEV" interaction method of our streaming temporal attention as depicted in Sec. 3.2 outperforms the "Voxel-to-BEV" interaction method by 0.92 mIoU, with less computational effort by reducing the number of interaction queries. Due to the presence of a substantial number of empty voxels, compressing voxel queries into BEV queries leads to more pure information and thus enhances the temporal interaction. Additionally, high mAVE indicates that the "Voxel-to-BEV" interaction method faces convergence issues in the flow task.

Length of the Memory Queue. We also study the influence of the queue length in Tab. 7, which shows that the performance improves as the queue length increases, proving the effectiveness of our multi-frame interaction mechanism compared to the single-frame interaction mechanism with $N = 1$ as BEVFormer [18]. The performance starts to plateau after $N = 3$ as in [35]. We use $N = 4$ in our framework, greater than which the improvement becomes ignorable. As presented in Tab. 7 under the "FPS" column, thanks to the efficient streaming memory mechanism, the additional time cost brought by multi-frame temporal interaction is negligible.

Table 7: Ablation study for the streaming temporal attention. "w/o TempAttn." denotes disabling the streaming temporal attention module.

Method	mIoU	IoU_{geo}	mAVE↓	FPS↑	car	truck	vege.
w/o TempAttn.	39.28	68.38	1.125	**4.2**	49.97	37.38	37.27
Voxel to BEV	41.62	71.53	1.104	3.9	52.16	40.59	40.68
BEV to BEV	**42.54**	**72.36**	**0.412**	4.1	**52.71**	41.04	**41.56**
Queue $N = 1$	41.67	71.18	0.426	4.1	52.56	40.36	40.28
Queue $N = 2$	42.28	71.95	0.421	4.1	52.53	40.75	41.11
Queue $N = 3$	42.42	72.18	0.415	4.1	52.70	**41.20**	41.48

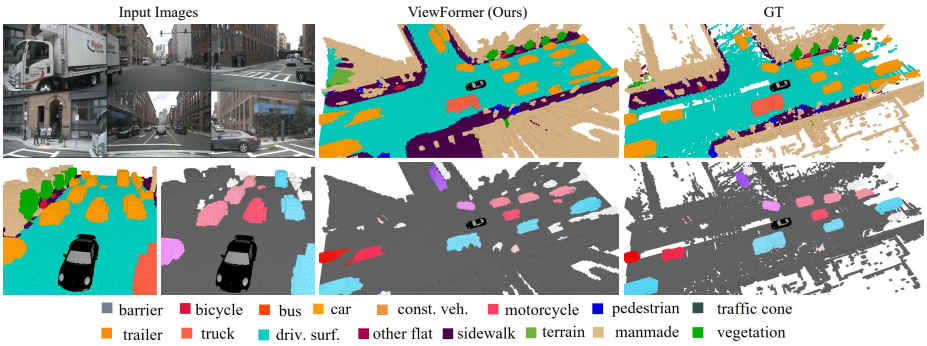

Fig. 6: Qualitative results of 3D occupancy and occupancy flow prediction. 3D occupancies are color-coded according to semantic categories. For flow visualization, we utilize color and brightness to represent the flow direction and magnitude respectively, following the convention of the optical flow fields [2, 33].

5.4 Qualitative Results

We present qualitative results of our ViewFormer in Fig. 6. An additional animated video is also provided in the supplementary material. Moreover, we demonstrate a qualitative comparison for flow predictions supervised by object-level flow (a) and occupancy-level flow (b) in Fig. 7, the model trained with our occupancy-level FlowOcc3D dataset achieves more reasonable results for a turning car, providing fine-grained motion information for AD systems.

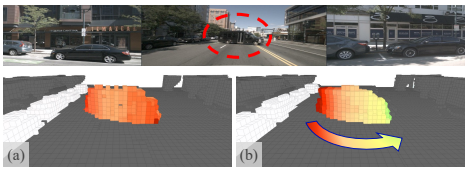

Fig. 7: Visualization on flow predictions. (a) Flow predictions supervised by object-level flow. (b) Flow predictions supervised by our FlowOcc3D.

6 Conclusion

We present the ViewFormer framework for 3D occupancy and occupancy flow prediction, featuring our proposed view attention that addresses the limitations of the existing projection-first spatial interaction method, as well as the streaming temporal attention designed for multi-frame temporal interaction. Furthermore, we build a novel occupancy-level flow benchmark FlowOcc3D to explore the potential of occupancy flow representation for dynamic scenes, which we also contribute to the research community. Our approach demonstrates significant advancements over previous state-of-the-art methods.

References

1. Tesla AI Day. https://www.youtube.com/watch?v=j0z4FweCy4M (2021)
2. Baker, S., Scharstein, D., Lewis, J.P., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. In: ICCV. pp. 1–8 (2007)
3. Berman, M., Triki, A.R., Blaschko, M.B.: The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In: CVPR. pp. 4413–4421 (2018)
4. Brazil, G., Pons-Moll, G., Liu, X., Schiele, B.: Kinematic 3d object detection in monocular video. In: ECCV. pp. 135–152 (2020)
5. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR. pp. 11618–11628 (2020)
6. Cao, A.Q., de Charette, R.: Monoscene: Monocular 3d semantic scene completion. In: CVPR. pp. 3991–4001 (2022)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255 (2009)
8. Dezert, J., Moras, J., Pannetier, B.: Environment perception using grid occupancy estimation with belief functions. In: FUSION. pp. 1070–1077 (2015)
9. Fong, W.K., Mohan, R., Hurtado, J.V., Zhou, L., Caesar, H., Beijbom, O., Valada, A.: Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. pp. 3795–3802 (2022)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
11. Huang, J., Huang, G.: Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. arXiv preprint arXiv:2203.17054 (2022)
12. Huang, J., Huang, G., Zhu, Z., Yun, Y., Du, D.: Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. arXiv preprint arXiv:2112.11790 (2021)
13. Huang, Y., Zheng, W., Zhang, Y., Zhou, J., Lu, J.: Tri-perspective view for vision-based 3d semantic occupancy prediction. In: CVPR. pp. 9223–9232 (2023)
14. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: CVPR. pp. 12697–12705 (2019)
15. Li, J., He, X., Wen, Y., Gao, Y., Cheng, X., Zhang, D.: Panoptic-phnet: Towards real-time and high-precision lidar panoptic segmentation via clustering pseudo heatmap. In: CVPR. pp. 11799–11808 (2022)
16. Li, Y., Yu, Z., Choy, C., Xiao, C., Alvarez, J.M., Fidler, S., Feng, C., Anandkumar, A.: Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. arXiv preprint arXiv:2302.12251 (2023)
17. Li, Y., Ge, Z., Yu, G., Yang, J., Wang, Z., Shi, Y., Sun, J., Li, Z.: Bevdepth: Acquisition of reliable depth for multi-view 3d object detection (*arXiv preprint arXiv:220610092*, 2022)
18. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In: ECCV. pp. 1–18 (2022)
19. Li, Z., Yu, Z., Wang, W., Anandkumar, A., Lu, T., Alvarez, J.M.: Fb-bev: Bev representation from forward-backward view transformations. In: ICCV. pp. 6919–6928 (2023)

20. Liao, B., Chen, S., Wang, X., Cheng, T., Zhang, Q., Liu, W., Huang, C.: Maptr: Structured modeling and learning for online vectorized hd map construction. In: ICLR (2023)
21. Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV. pp. 2999–3007 (2017)
22. Lin, T., Maire, M., Belongie, S.J., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: ECCV. pp. 740–755 (2014)
23. Liu, Y., Wang, T., Zhang, X., Sun, J.: PETR: Position embedding transformation for multi-view 3d object detection. In: ECCV. pp. 531–548 (2022)
24. Luo, W., Yang, B., Urtasun, R.: Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In: CVPR. pp. 3569–3577 (2018)
25. Miao, R., Liu, W., Chen, M., Gong, Z., Xu, W., Hu, C., Zhou, S.: Occdepth: A depth-aware method for 3d semantic scene completion. arXiv:2302.13540 (2023)
26. Park, J., Xu, C., Yang, S., Keutzer, K., Kitani, K., Tomizuka, M., Zhan, W.: Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. In: ICLR (2023)
27. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: ECCV. pp. 194–210 (2020)
28. Qi, C.R., Zhou, Y., Najibi, M., Sun, P., Vo, K., Deng, B., Anguelov, D.: Offboard 3d object detection from point cloud sequences. In: CVPR. pp. 6134–6144 (2021)
29. Reading, C., Harakeh, A., Chae, J., Waslander, S.L.: Categorical depth distribution network for monocular 3d object detection. In: CVPR. pp. 8555–8564 (2021)
30. Schreiber, M., Belagiannis, V., Gläser, C., Dietmayer, K.: Dynamic occupancy grid mapping with recurrent neural networks. In: ICRA. pp. 6717–6724 (2021)
31. Sima, C., Tong, W., Wang, T., Chen, L., Wu, S., Deng, H., Gu, Y., Lu, L., Luo, P., Lin, D., Li, H.: Scene as occupancy. In: ICCV. pp. 8406–8415 (2023)
32. Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.: Semantic scene completion from a single depth image. In: CVPR. pp. 1746–1754 (2017)
33. Teed, Z., Deng, J.: RAFT: recurrent all-pairs field transforms for optical flow. In: ECCV. pp. 402–419 (2020)
34. Tian, X., Jiang, T., Yun, L., Wang, Y., Wang, Y., Zhao, H.: Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. arXiv preprint arXiv:2304.14365 (2023)
35. Wang, S., Liu, Y., Wang, T., Li, Y., Zhang, X.: Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In: ICCV. pp. 11618–11628 (2023)
36. Wang, T., Zhu, X., Pang, J., Lin, D.: Fcos3d: Fully convolutional one-stage monocular 3d object detection. In: ICCV. pp. 913–922 (2021)
37. Wang, W., Dai, J., Chen, Z., Huang, Z., Li, Z., Zhu, X., Hu, X., Lu, T., Lu, L., Li, H., Wang, X., Qiao, Y.: Internimage: Exploring large-scale vision foundation models with deformable convolutions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 14408–14419 (June 2023)
38. Wang, X., Zhu, Z., Xu, W., Zhang, Y., Wei, Y., Chi, X., Ye, Y., Du, D., Lu, J., Wang, X.: Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. arXiv preprint arXiv:2303.03991 (2023)
39. Wang, Y., Guizilini, V., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: DETR3D: 3d object detection from multi-view images via 3d-to-2d queries. In: CoRL. pp. 180–191 (2021)

40. Zhang, Y., Zhu, Z., Du, D.: Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. arXiv preprint arXiv:2304.05316 (2023)
41. Zhu, B., Jiang, Z., Zhou, X., Li, Z., Yu, G.: Class-balanced grouping and sampling for point cloud 3d object detection. arXiv preprint arXiv:1908.09492 (2019)
42. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: Deformable transformers for end-to-end object detection. In: ICLR (2021)