Streaming Dense Voxel Representations for 3D Occupancy Prediction

Anonymous ECCV 2026 Submission

Paper ID #556

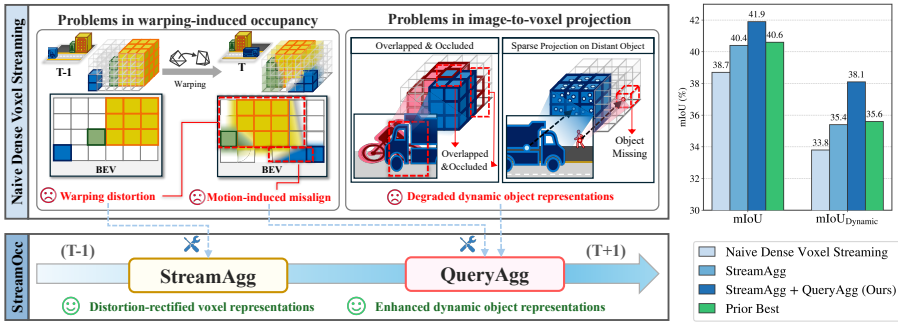
Abstract. In this paper, we explore dense voxel streaming for accurate and efficient 3D occupancy prediction. While dense voxel representations offer fine-grained spatial details and streaming paradigm enables efficient temporal processing, naively combining the two introduces key challenges: (i) warping-induced distortions caused by interpolation used for temporal alignment, and (ii) degraded dynamic object representations due to motion misalignment and detail loss in image-to-voxel projection. To address these, we propose **StreamOcc**, a novel framework that utilizes two aggregation strategies. Specifically, it first refines propagated voxel features to reduce warping artifacts before temporal accumulation, and then selectively injects instance-level query features encoding dynamic-object semantics into the corresponding occupied voxel regions, preserving temporally consistent modeling while strengthening dynamic object representations. Unlocking effective dense voxel streaming, StreamOcc achieves state-of-the-art performance on SurroundOcc-benchmark and Occ3D-nuScenes under real-time constraints, outperforming the prior best methods by $+$**1.3**/**2.5** and $+$**1.5**/**2.0 in (overall/dynamic object) mIoU**, respectively, while running at 83.3 ms per frame with only 2.8 GB of memory. Code will be available.

Keywords: 3D Occupancy Prediction · Dense Voxel Streaming · Autonomous Driving

1 Introduction

Vision-based 3D occupancy prediction has become a key perception task for autonomous driving, enabling dense and comprehensive scene understanding. Specifically, it aims to classify each voxel in 3D space into semantic categories, including static objects (e.g., sidewalks, drivable surfaces), dynamic objects (e.g., vehicles, pedestrians), and free space [31, 35, 38]. This requires an accurate and fine-grained understanding of dense 3D spatial semantics.

Recent methods often leverage the multi-frame fusion mechanism (i.e., jointly processing current and several past frames) to incorporate temporal context, using either dense voxel representations [7, 19–21, 26, 30, 34] or sparse representations [5, 6, 9, 11, 24, 32] (e.g., z-axis pooled features, sparse voxels or queries, and Gaussians). However, these methods suffer from an inherent accuracy–efficiency



(a) Naive Dense Voxel Streaming vs StreamOcc (ours) **(b)** Effectiveness of StreamOcc

Fig. 1: (a) Challenges of naive dense voxel streaming and the proposed StreamOcc framework, consisting of StreamAgg and QueryAgg to address them. **(b)** 3D occupancy prediction results on Occ3D-nuScenes [31], showing consistent improvements with StreamOcc components over naive streaming and the prior best real-time method [2].

trade-off. While dense voxel representations preserve fine-grained 3D spatial details, repeatedly processing dense historical features incurs high computational costs in terms of memory consumption (5–12GB) and inference latency (166–1,250ms), limiting practical deployment. In contrast, compressed or sparse representations attempt to improve efficiency, but this often comes at the cost of spatial fidelity due to their limited representational capacity.

To avoid repeatedly processing multiple frames, the streaming paradigm offers an efficient alternative by recurrently updating propagated features with current-frame features. While this approach has demonstrated strong temporal modeling performance in sparse prediction tasks, such as 3D object and map detection [3, 23, 37], its extension for 3D occupancy prediction remains non-trivial. Recent works have extended this streaming paradigm for 3D occupancy prediction, including GaussianWorld [43] and ViewFormer [15]. However, these approaches rely on sparse representation, such as Gaussian primitives or compressed representations, which inherently limit their capacity to accurately model dense 3D spatial semantics.

Motivated by these observations, we explore a dense voxel streaming framework that integrates streaming-based temporal efficiency with voxel-based fine-grained 3D representation. Yet, as illustrated in Fig. 1 (a)-top, naively applying this integration struggles to capture fine-grained spatial details of the scene due to two key challenges: **(i)** warping-induced distortions, which arise when aligning propagated voxel features from the previous timestep to the current ego-centric coordinate as grid values are resampled via interpolation. **(ii)** degraded dynamic object features, which are caused by motion-induced misalignment and information loss that inherently occurs when projecting image features into voxel space (e.g., sparse projections of distant objects, coarse-grid instance merging, and occlusion-induced feature truncation).

In response to these challenges, our proposed method (**StreamOcc**) adopts two aggregation strategies to enable effective dense voxel streaming. **First**, we

design Rectified Voxel Streaming Aggregation (StreamAgg) to enable temporally consistent streaming of dense voxel features. It warps voxel features from the previous timestep and refines propagated voxel features to mitigate distortion through geometry-aware adaptive residual correction and align voxel semantics with the current frame before recurrent fusion. **Second**, to improve dynamic object modeling, which is safety-critical in autonomous driving due to importance of interactions with dynamic agents, we introduce Query-guided Aggregation (QueryAgg). In this module, instance queries encode dynamic object semantics extracted from image space and are selectively injected into the corresponding occupied voxel regions, compensating for the limitations of voxel-only accumulation. Unlike prior strategies that re-aggregate image features across all voxels [19, 26, 30], our targeted aggregation focuses on dynamic objects, thereby enhancing their representations more effectively and efficiently.

Our contributions can be summarized as follows:

- We introduce StreamOcc, the first framework that integrates a streaming paradigm with dense voxel representations for 3D occupancy prediction, achieving both high spatial fidelity and computational efficiency.
- To mitigate warping-induced distortions and degraded dynamic object representations arising from streaming dense voxel features, we introduce Rectified Voxel Streaming Aggregation and Query-guided Aggregation; their effects are shown in Fig. 1(b).
- We demonstrate that StreamOcc achieves state-of-the-art performance on SurroundOcc-benchmark and Occ3D-nuScenes under real-time constraints. Notably, it runs at 83.3 ms latency with only 2.8 GB memory, striking a strong balance between accuracy and efficiency.

2 Related Work

2.1 3D Occupancy Prediction

3D occupancy prediction is a dense prediction task that assigns semantic labels to the entire scene-wise voxel space for comprehensive 3D scene understanding. Early methods extended BEV features into 3D voxel space to construct voxel-based representations [8,17,19], while subsequent approaches [20,31,35,38] further improved these voxel representations through multi-scale encoding and depth–semantic fusion. However, using only a single frame limits visibility in occluded regions and results in sparse feature projection, leading to incomplete scene reconstructions. To alleviate this limitation, multi-frame fusion methods [7, 21, 26, 30, 34] jointly process consecutive frames to improve temporal consistency and scene completeness, achieving strong performance. While multi-frame fusion improves temporal consistency and scene completeness, employing dense voxel representations in this setting introduces substantial computational overhead. In pursuit of higher efficiency, recent methods adopt sparse or compressed representations. TPVFormer [10] models 3D space using tri-plane

features, while SparseOcc [24], FastOcc [6], GSD-Occ [5], and OPUS [32] reduce computation through compressed features or sparse representations, including voxel- and query-based approaches. More recently, Gaussian-based methods [9,11,28,43] have emerged as efficiency-oriented alternatives. However, compressed and sparse representations often struggle to preserve fine-grained spatial fidelity or require a large number of primitives (e.g. Gaussians, query sets) to maintain semantic detail, leading to increased computational cost. Consequently, achieving high spatial fidelity without sacrificing efficiency remains a key challenge in 3D occupancy prediction.

2.2 Streaming in 3D Perception

To avoid the computational cost of processing multiple frames, the streaming paradigm, which recurrently updates propagated features with current features, has emerged as an efficient alternative. This design has demonstrated strong temporal modeling performance in sparse prediction tasks, such as 3D object or map detection [3, 13, 14, 23, 33, 37]. Motivated by these advantages, recent works have extended the streaming paradigm for 3D occupancy prediction using sparse representations, including GaussianWorld [43], which models the scene using Gaussian primitives, and ViewFormer [15], which adopts compressed representations (BEV feature). However, unlike sparse prediction tasks, which model only a subset of spatial locations and do not require maintaining a fully dense representation, 3D occupancy prediction requires voxel-wise semantic modeling across the entire scene. This fundamental difference makes sparse representations inherently insufficient for 3D occupancy prediction, as their limited representational capacity restricts accurate modeling of fine-grained 3D spatial semantics. Building upon this observation, we propose StreamOcc, a framework that effectively integrates the streaming paradigm with dense voxel representations for accurate and efficient 3D occupancy prediction.

3 Method

We present a 3D occupancy prediction framework that effectively integrates a streaming paradigm with dense voxel representations. As illustrated in Fig. 2, the framework consists of two main phases: Rectified Voxel Streaming Aggregation (StreamAgg; Sec. 3.1) and Query-guided Aggregation (QueryAgg; Sec. 3.2).

3.1 StreamAgg: Rectified Voxel Streaming Aggregation

In this section, we present Rectified Voxel Streaming Aggregation (StreamAgg), which recurrently accumulates voxel features while mitigating warping-induced distortions to preserve temporal consistency. StreamAgg comprises three components: (i) 2D-to-3D feature extraction, (ii) motion-aware warping for temporal alignment, and (iii) adaptive residual correction to suppress the distortions.

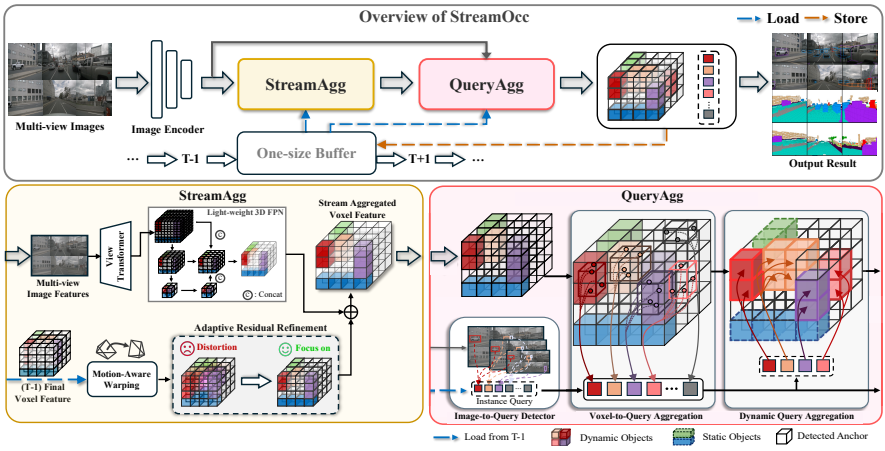


Fig. 2: Overview of StreamOcc. StreamOcc predicts the 3D occupancy state of each voxel in a streaming manner via two-stage aggregation. First, voxel features are recurrently accumulated over time through Rectified Voxel Streaming Aggregation (StreamAgg), which mitigates warping-induced distortions during temporal propagation. These temporally aggregated features are further refined using Query-guided Aggregation (QueryAgg), which leverages instance queries encoding fine-grained semantics to enhance dynamic object representations.

2D-to-3D View Transformation. Given a set of N multi-view images $\{\mathbf{I}_i^t\}$ for $i \in \{1, 2, \ldots, N\}$ at the current timestep t, we extract multi-scale 2D features $\mathbf{F}_i$ using a ResNet [4] with FPN [22]. Following prior works [5,7,26], we project the image features into a unified 3D voxel space, producing an initial voxel feature $\mathbf{V}_{\text{init}}^t \in \mathbb{R}^{C_{\text{init}} \times X \times Y \times Z}$, where X, Y, and Z denote the voxel grid dimensions. We then apply a lightweight 3D-FPN to aggregate multi-level voxel features and obtain the current downsampled voxel feature $\mathbf{V}_{\text{down}}^t \in \mathbb{R}^{C \times \frac{X}{2} \times \frac{Y}{2} \times \frac{Z}{2}}$.

Motion-aware Voxel Feature Warping. In streaming settings, propagated voxel features $\mathbf{V}_{\text{final}}^{t-1} \in \mathbb{R}^{C \times \frac{X}{2} \times \frac{Y}{2} \times \frac{Z}{2}}$ from the previous timestep (which have recurrently accumulated temporal context) must be aligned to the current ego-centric coordinates. Without such alignment, naive temporal accumulation leads to spatial inconsistencies, degrading dense voxel representations. To address this, we warp $\mathbf{V}_{\text{final}}^{t-1}$ according to the ego-motion between frames.

As the initial step of this warping process, we transform the spatial coordinates of the previous voxel grid $\mathbf{P}^e(t-1)$ into the current ego-centric frame:

$$\bar{\mathbf{P}}^e(t) = \mathcal{T}_{g \to e}^t \cdot \mathcal{T}_{e \to g}^{t-1} \cdot \mathbf{P}^e(t-1), \tag{1}$$

where $\mathcal{T}_{e \to g}^{t-1}$ maps coordinates from the past ego frame to the global frame, and $\mathcal{T}_{g \to e}^t$ maps them to the current ego-centric frame. We then obtain the warped voxel feature via trilinear interpolation $\texttt{Interp}(\cdot)$ to resample past voxel features onto the current voxel grid:

$$\mathbf{V}_{\text{warp}}^t = \texttt{Interp}(\mathbf{V}_{\text{final}}^{t-1}, \bar{\mathbf{P}}^e(t), \mathbf{P}^e(t-1)), \tag{2}$$

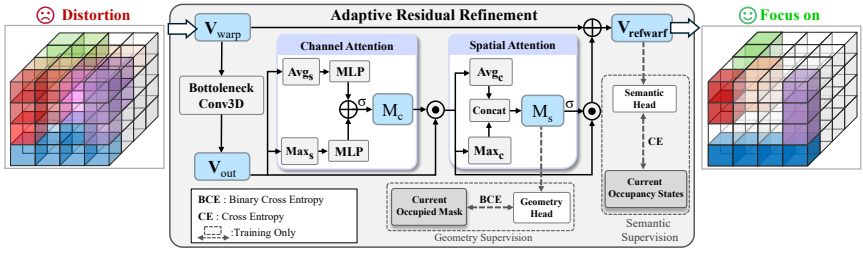


Fig. 3: Details of Adaptive Residual Refinement. This module rectifies warping-induced distortions via adaptive residual correction that focuses on informative features guided by the geometric context. The refined warped voxel representation is supervised to achieve semantic alignment with the current scene state for consistent streaming. Geometry and Semantic heads are used only during training.

Adaptive Residual Refinement. The warping process propagates contextual cues from past observations but inevitably introduces distortions due to interpolation-based voxel features resampling. These distortions often lead to erroneous values or blurred object boundaries, degrading feature quality and temporal consistency. To mitigate these effects and preserve spatial consistency over time, we introduce an Adaptive Residual Refinement module (Fig. 3).

Specifically, it adopts a bottleneck design [4] to efficiently process the warped feature $\mathbf{V}^t_{\text{warp}}$: it compresses the feature to $C/4$ channels, applies 3D convolutions, and expands it back to C channels to obtain a corrective representation $\mathbf{V}^t_{\text{out}}$. Since warping artifacts are concentrated around object boundaries rather than uniformly distributed across the voxel space, the residual update should be selectively weighted. Accordingly, we extend the Convolutional Block Attention Module [36] to the 3D domain and introduce explicit supervision for adaptive residual correction and semantic refinement. Under this design, 3D channel and spatial attention are applied as follows:

$$\begin{aligned}\mathbf{M}_c &= \sigma(\text{MLP}(\text{Avg}_s(\mathbf{V}^t_{\text{out}})) + \text{MLP}(\text{Max}_s(\mathbf{V}^t_{\text{out}}))),\\ \mathbf{M}_s &= \text{Conv3D}([\text{Avg}_c(\mathbf{M}_c \odot \mathbf{V}^t_{\text{out}}); \text{Max}_c(\mathbf{M}_c \odot \mathbf{V}^t_{\text{out}})]),\end{aligned} \tag{3}$$

where σ and $\odot$ denote the sigmoid function and element-wise multiplication, respectively, alongside spatial (Avg_s, Max_s) and channel (Avg_c, Max_c) pooling.

The spatial attention feature $\mathbf{M}_s$ serves as a soft gating signal for the adaptive residual update. To guide the gating mechanism to focus on meaningful features for the residual update, we supervise $\mathbf{M}_s$ with the ground-truth binary occupied mask during training **(Geometry Supervision)**. This supervision enables $\mathbf{M}_s$ to distinguish occupied regions from free space, allowing geometry-aware adaptive weighting of the residual update. Using these geometry-aware adaptive weights, we refine the warped voxel feature as follows:

$$\mathbf{V}^t_{\text{refwarp}} = \sigma(\mathbf{M}_s) \odot (\mathbf{M}_c \odot \mathbf{V}^t_{\text{out}}) + \mathbf{V}^t_{\text{warp}}. \tag{4}$$

While geometry-guided gating encourages the residual update to focus on meaningful regions, it does not explicitly specify which semantic cues the corrective feature should learn to rectify warping-induced distortions in $\mathbf{V}^t_{\text{warp}}$. To

address this, we introduce **Semantic Supervision** to enforce semantic consistency by training $\mathbf{V}^t_{\text{refwarp}}$ to predict the semantic occupancy of the current frame. This supervision guides the residual correction to encode semantically consistent features, suppressing warping artifacts and aligning the representation with the current scene state.

StreamAgg then outputs the aggregated voxel feature $\mathbf{V}^t_{\text{S.A}}$ by concatenating the refined warped feature with the current voxel feature and fusing them with a Conv1D block.

$$\mathbf{V}^t_{\text{S.A}} = \text{Conv}_{1D}\big([\mathbf{V}^t_{\text{refwarp}}; \mathbf{V}^t_{\text{down}}]\big). \tag{5}$$

3.2 QueryAgg: Query-guided Aggregation

Dynamic object features (e.g., vehicles, pedestrians) are particularly vulnerable to degradation due to motion misalignment and sparse image-to-voxel projection. (see Fig. 5) To address this limitation, we introduce Query-guided Aggregation (QueryAgg). QueryAgg recurrently updates instance queries from multi-level image features $\mathbf{F}_i$ using Sparse4Dv3 [23] in a streaming manner. These queries are then enhanced with 3D geometric cues from voxel features and selectively aggregated into the corresponding occupied voxel regions, enabling accurate and robust dynamic object representation.

Voxel-to-Query Aggregation. Instance queries obtained from the Image-to-Query Detector [23] encode rich semantics and provide coarse localization. However, due to depth ambiguity along the viewing ray, they often produce multiple false-positive queries across different depths (see Supplementary Material). In contrast, voxel representations provide reliable 3D geometric information, but lack fine-grained semantic cues for dynamic objects. This complementarity motivates the Voxel-to-Query Aggregation, which injects geometric information from $\mathbf{V}^t_{\text{S.A}}$ into instance queries to reduce depth ambiguity.

Specifically, we refine each query feature using deformable attention [40] to selectively sample informative regions in the voxel space. The refined query feature q^i_{ref} is updated as:

$$\begin{aligned} q^i_{\text{ref}} = q^i_{\text{img}} + \textstyle\sum_{h=1}^{H} W_h \Big[\sum_{o=1}^{O} \alpha_{iho} \cdot W'_h \cdot \\ \mathbf{V}^t_{\text{S.A}}\big(x_i + \Delta x_{iho}, y_i + \Delta y_{iho}, z_i + \Delta z_{iho}\big) \Big], \end{aligned} \tag{6}$$

where q^i_{img} denotes the query feature produced by Image-to-Query Detector, and(x_i, y_i, z_i) denotes the 3D center of the associated object, and H and O are the numbers of attention heads and sampling points, respectively. Each head predicts sampling offsets $(\Delta x_{iho}, \Delta y_{iho}, \Delta z_{iho})$ around the query center and aggregates voxel features from the StreamAgg representation $\mathbf{V}^t_{\text{S.A}}$, using attention weights α_{iho}. The learnable matrices W_h and W'_h project query and sampled voxel features into a shared embedding space. This aggregation injects geometric cues into instance queries, reducing depth ambiguity and improving spatial localization.

Dynamic Query Aggregation. As mentioned above, voxel-only feature accumulation yields degraded representations for dynamic objects; therefore, we introduce Dynamic Query Aggregation (DQA), which selectively injects instance-level query features into voxel regions corresponding to dynamic objects.

Specifically, DQA first filters high-scoring instance queries (see the following subsection) that capture dynamic objects and maps them to the voxel regions occupied by the corresponding objects. Query-to-voxel attention is then applied to aggregate instance-level features into the voxel space as follows (in the following, we omit the timestep index t for notational simplicity.):

$$\mathbf{q}^i = \mathbf{W}_Q \left(\mathbf{V}^i_{\text{S.A}} + \mathbf{p}_i\right), \quad \mathbf{k}^{ij} = \mathbf{W}_K q^{ij}_{\text{ref}}, \quad \mathbf{v}^{ij} = \mathbf{W}_V q^{ij}_{\text{ref}}, \tag{7}$$

$$\alpha^i = \text{softmax}\left(\frac{\mathbf{q}^{i\top}}{\sqrt{d}} \cdot \left[\mathbf{k}^{ij}\right]_{j\in\mathcal{N}^i}\right), \quad \mathbf{z}^i = \sum_{j\in\mathcal{N}^i} \alpha^{ij} \cdot \mathbf{v}^{ij}, \tag{8}$$

$$\mathbf{g}^i = \sigma\left(\mathbf{W}_{\text{gv}}\mathbf{V}^i_{\text{S.A}} + \mathbf{W}_{\text{gz}}\mathbf{z}^i\right), \tag{9}$$

where $\mathcal{N}^i$ denotes the set of instance queries whose predicted bounding boxes overlap with the i-th voxel grid cell, and q^{ij}_{ref} represents the feature of the j-th query associated with voxel i. Here, $\mathbf{q}^i$, $\mathbf{k}^{ij}$, and $\mathbf{v}^{ij}$ are obtained via linear projections $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$, respectively, and $\mathbf{p}_i$ denotes positional embedding. Attention vector α^i is computed over $\mathcal{N}^i$, and α^{ij} denotes its j-th element, which weights each query feature when forming the aggregated feature $\mathbf{z}^i$.

For voxels with no overlapping queries ($\mathcal{N}^i = \emptyset$), the features remain unchanged; for voxels with overlapping queries, the gating vector $\mathbf{g}^i$ determines how much of the aggregated instance feature $\mathbf{z}^i$ should be injected into voxel i:

$$\mathbf{V}^i_{\text{DQA}} = \begin{cases} \mathbf{V}^i_{\text{S.A}}, & \text{if } \mathcal{N}^i = \emptyset, \\ \mathbf{V}^i_{\text{S.A}} + \mathbf{g}^i \odot \mathbf{z}^i, & \text{otherwise.} \end{cases} \tag{10}$$

This selective update allows DQA to enhance dynamic-object voxels while preserving stable representations in static or empty regions. The updated voxel features $\mathbf{V}_{\text{DQA}}$ are further processed by a feed-forward network with normalization and residual connections, producing the final voxel representation $\mathbf{V}_{\text{final}}$, which is fed into the occupancy head for 3D occupancy prediction.

Query Selection for DQA. Reliable query selection is essential for effective Dynamic Query Aggregation, as naive selection based solely on IoU may introduce shortcut learning. In particular, IoU-only selection can map queries that happen to be close to the ground truth into voxel features regardless of detection quality, leading to implicit ground-truth leakage and degraded inference performance. Therefore, we design a training-time query selection strategy that combines confidence scores (0.3) with either IoU or geometric constraints to maintain generalization. The selection is scale-aware: we use an IoU-based criterion for large objects, while adopting a geometry-based criterion for small objects where IoU becomes unreliable, by enforcing consistency between predicted and ground-truth box geometry (e.g., center and size deviations); detailed thresholds and formulations are provided in the Supplementary Material.

During inference, the filtering strategy is simplified by selecting only instance queries with a confidence score above 0.3, consistent with the detection threshold.

3.3 Decoder for Occupancy Prediction

To predict the high-resolution 3D voxel occupancy, we upsample the final voxel feature $\mathbf{V}_{\text{final}}$ to obtain $\mathbf{V}_{\text{up}} \in \mathbb{R}^{C \times X \times Y \times Z}$, which is processed by a lightweight MLP-based decoder used in training and inference. To further enhance semantic representation learning without modifying the voxel features, we incorporate an Auxiliary Mask Decoder inspired by Co-DETR [41], adopting the multi-head transformer-based mask prediction design from the Semantic-aware Group Decoder in COTR [26]. During training, a group-wise one-to-many assignment associates each ground-truth mask with multiple queries, providing richer supervision. As this auxiliary decoder is used solely for feature learning, it is omitted at inference time to avoid unnecessary computation and maintain efficiency.

4 Experiments

4.1 Datasets

We evaluate our method on two benchmark datasets for 3D occupancy prediction: Occ3D-nuScenes [31] and SurroundOcc dataset [35]. Both datasets represent the scene as a voxel grid of size $200 \times 200 \times 16$, where each voxel is labeled as either *occupied* or *empty*, and occupied voxels are further categorized into 16 semantic classes such as `car` and `pedestrian`. In **Occ3D-nuScenes,** the voxel grid is defined in the ego coordinate, spanning $[-40\text{m}, 40\text{m}]$ along both X and Y, and $[-1\text{m}, 5.4\text{m}]$ along Z, discretized at a resolution of 0.4m. Occupied voxels are assigned 17 semantic classes (16 known categories + `others`). In **SurroundOcc,** the voxel grid is defined in the LiDAR coordinate, covering $[-50\text{m}, 50\text{m}]$ along X and Y, and $[-5\text{m}, 3\text{m}]$ along Z, with a voxel resolution of 0.5m.

4.2 Implementation and Evaluation Details

Following standard practice [2, 5, 7, 26], we use ResNet-50 [4] as the image backbone and resize input images to 256×704. The initial voxel feature $\mathbf{V}_{\text{init}}$ is represented as a $200 \times 200 \times 16$ voxel grid with 64 channels. We use 900 instance queries for processing dynamic objects, and the auxiliary mask decoder consists of 6 heads. We train with AdamW [25] for 24 epochs on Occ3D-nuScenes and 20 epochs on SurroundOcc, without CBGS [39], using a batch size of 8, gradient clipping, an initial learning rate of 2×10^{-4}, and a warmup ratio of 1/3 for the first 200 iterations. Inference latency and GPU memory are measured on a single NVIDIA A100 or RTX 4090 GPU. Further, we evaluate occupancy prediction using mean IoU (mIoU and mIoU_D over semantic classes and dynamic objects), and additionally report class-agnostic IoU to assess overall 3D geometry reconstruction by treating all occupied voxels as foreground. The Image-to-Query detector used in QueryAgg is evaluated using nuScenes Detection Score (NDS) [1] and mean Average Precision (mAP).

Table 1: Quantitative results on Occ3D-nuScenes. We report mIoU, mIoU_D (dynamic objects only), inference latency (measured on a NVIDIA A100 GPU), and memory consumption. Methods with near real-time latency (≈100 ms) are highlighted in gray. († denotes reproduced without CBGS [39] under the same setting as ours.)

Method	Backbone	Image Size	Visible Mask	mIoU ↑	mIoU_D ↑	Latency (ms)	Memory (MB)
BEVFormer [19]	ResNet-101	900 × 1600	✔	39.3	37.2	212.7	6,651
BEVDet4D [7]	ResNet-50	256 × 704	✔	39.2	32.8	1250.0	6,053
PanoOcc [34]	ResNet-101	864 × 1600	✔	41.6	37.3	333.3	11,991
GEOcc [30]	ResNet-50	256 × 704	✔	43.6	38.6	450.0	-
COTR [26]	ResNet-50	256 × 704	✔	44.5	39.5	1111.1	10,453
ALOcc† [2]	ResNet-50	256 × 704	✔	45.0	38.7	166.7	10,793
STCOcc [21]	ResNet-50	256 × 704	✔	44.6	40.0	212.8	7,739
SparseOcc [24]	ResNet-50	256 × 704	✘	30.9	28.2	56.5	6,883
OPUS [32]	ResNet-50	256 × 704	✘	35.6	32.6	75.7	6,735
FastOcc [6]	ResNet-101	320 × 800	✔	37.2	-	93.4	-
ProtoOcc [27]	ResNet-50	432 × 800	✔	37.8	32.1	105.0	-
FB-OCC [20]	ResNet-50	256 × 704	✔	39.1	34.3	97.1	9,632
ProtoOcc [12]	ResNet-50	256 × 704	✔	39.6	34.8	77.9	-
GSD-Occ [5]	ResNet-50	256 × 704	✔	39.4	35.1	50.0	4,759
ViewFormer† [15]	ResNet-50	256 × 704	✔	39.6	33.3	102.0	3,103
ALOcc-mini† [2]	ResNet-50	256 × 704	✔	40.6	35.6	33.1	2,577
StreamOcc (Ours)	ResNet-50	256 × 704	✔	**41.9**	**38.1**	83.3	2,788

Table 2: Results on the SurroundOcc dataset [35]. We report IoU, mIoU, mIoU_D, inference latency (Lat., ms on a NVIDIA 4090 GPU) and memory consumption (Mem., MB).

Method	IoU↑	mIoU↑	mIoU_D↑	Lat.	Mem.
TPVFormer [10]	30.9	17.1	13.4	320	5,100
SurroundOcc [35]	31.5	20.3	16.0	344	5,491
GaussianFormer [11]	29.8	19.1	16.7	372	6,229
GaussianFormer-2 [9]	31.7	20.8	18.3	451	4,535
QuadricFormer [42]	32.1	21.1	17.3	179	2,563
GaussianWorld [43]	33.0	21.9	19.0	228	7,030
StreamOcc (Ours)	**33.8**	**23.4**	**21.0**	**84**	2,788

Table 3: RayIoU [24] evaluation on Occ3D-nuScenes [31]. Comparison with real-time methods (≤100 ms), without visibility mask.

Method	RayIoU	*1m*	*2m*	*4m*
SpaseOcc [24]	35.1	29.1	35.8	40.3
FB-Occ [20]	35.6	-	-	-
GSD-Occ [5]	38.9	33.0	39.7	44.1
ODG [28]	39.2	-	-	-
ALOcc-mini [2]	39.3	32.9	40.1	44.8
OPUS [32]	40.3	33.7	41.1	46.0
StreamOcc (Ours)	**41.1**	**34.2**	**41.9**	**47.1**

4.3 Comparison with SOTA Methods

To comprehensively compare StreamOcc with prior works, we evaluate it on Occ3D-nuScenes [31] against multi-frame fusion approaches, additionally reporting RayIoU [24] for ray-wise consistency, and on the SurroundOcc benchmark [35] against Gaussian- and Quadric-based methods. Furthermore, since accurate modeling of dynamic objects is safety-critical in autonomous driving, we report mIoU_D to explicitly quantify performance on dynamic objects.

Comparison with SOTA Methods on Occ3D-nuScenes. We first evaluate StreamOcc on the Occ3D-nuScenes dataset [31] in Tab. 1. StreamOcc achieves 41.9 mIoU and 38.1 mIoU_D, the best accuracy among real-time methods. It outperforms the prior-best real-time method, ALOcc-mini [2], by +1.3 mIoU and +2.5 mIoU_D. StreamOcc also surpasses multi-frame fusion methods that use sparse or compressed representations, such as SparseOcc [24], FastOcc [6], OPUS [32], and GSD-Occ [5]. Compared to ViewFormer [15], which uses compressed multi-frame features propagated in a streaming manner, StreamOcc achieves higher accuracy (+2.3 mIoU and +4.8 mIoU_D) while operating faster.

Table 4: Ablation study on the Occ3D-nuScenes dataset. Evaluates the effect of each component of StreamOcc in terms of overall mIoU (mIoU_A), dynamic and static object mIoU (mIoU_D and mIoU_S, respectively), and per-class performance.

				mIoU ↑ (Dynamic Objects)										mIoU ↑ (Static Objects)					
Method	mIoU_A ↑	mIoU_D ↑	mIoU_S ↑	barrier	bicycle	bus	car	cons. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. surf.	other flat	sidewalk	terrain	manmade	vegetation
A. Base (Single-Frame)	36.8	32.6	48.5	44.1	22.4	44.9	50.0	21.2	23.9	25.7	25.4	32.0	36.3	78.9	41.0	48.5	51.4	38.1	32.9
B. StreamAgg	40.4	35.4	**53.4**	47.2	26.2	45.5	53.8	24.8	29.4	26.9	29.5	32.8	38.4	83.2	45.6	**54.0**	**57.6**	**42.7**	**37.5**
C. B + Detection Head	40.8	36.3	53.1	48.0	26.8	46.0	54.6	24.9	29.6	28.6	30.7	33.2	40.9	83.2	45.4	53.8	57.3	41.8	37.4
D. B + QueryAgg (Ours)	**41.9**	**38.1**	53.3	**50.8**	**29.6**	**48.5**	**56.1**	**25.4**	**31.4**	**30.0**	**33.0**	**34.6**	**41.9**	**83.3**	**45.9**	53.8	57.3	42.3	37.2

Compared to multi-frame dense voxel fusion methods, StreamOcc achieves stronger performance with significantly lower cost than PanoOcc [34], which is about 4× slower and uses 4.3× more memory. Although COTR [26] and the ALOcc [2] attain higher accuracy, they incur substantially higher overhead, being about 13× and 2× slower than StreamOcc while consuming 3.7× and 3.8× more memory, respectively. Such high latency and memory overhead make these approaches less suitable for on-vehicle deployment.

Comparison with SOTA Methods on SurrondOcc-benchmark. On the SurroundOcc [35] dataset (Tab. 2), StreamOcc likewise achieves the best performance across IoU, mIoU, and mIoU_D, outperforming Gaussian- and Quadric-based approaches [9, 11, 42]. Notably, it also surpasses the prior best method, GaussianWorld [43], by +1.5 mIoU and +2.0 mIoU_D, while requiring substantially lower cost, running 2.7× faster and using 2.5× less memory.

Comparison with SOTA Methods on RayIoU. We further evaluate methods using RayIoU [24] on Occ3D-nuScenes [31] (Tab. 3), which measures the consistency of occupancy prediction along viewing rays. StreamOcc achieves the highest overall RayIoU (41.1), consistently outperforming prior real-time approaches. Together, these results demonstrate that StreamOcc provides state-of-the-art accuracy while maintaining practical runtime and memory efficiency, making it suitable for real-world deployment.

4.4 Ablations

In this section, we conduct ablation studies on Occ3D-nuScenes to analyze the contribution of each module and validate our design choices. Inference latency and GPU memory are measured on one NVIDIA A100.

Effect of StreamOcc Components. Tab. 4 presents an ablation study on the Occ3D-nuScenes dataset [31]. Base model (A), using only a single-frame, achieves 36.8 mIoU. Adding StreamAgg (B), which effectively rectifies warping distortions to maintain temporal consistency while sequentially accumulating voxel features over time, we observe a substantial performance improvement to 40.4 mIoU. However, (B) still shows a large performance gap between static and dynamic objects (53.4 vs. 35.4), indicating that voxel-only accumulation struggles to capture the representations of dynamic objects. To address this limitation,

Table 5: Refinement ablation.

Method	mIoU↑	Lat.	Mem.
Naive Voxel Streaming	38.72	43.9	2,423
(+) Adaptive Residual Refinement	39.84	49.0	2,437
(+) Semantic Supervision	40.25	49.0	2,437
(+) Geometry Supervision	**40.37**	49.0	2,437

Table 6: QueryAgg ablation.

I2Q	V2Q	DQA	mIoU↑	NDS↑	mAP↑
			40.37	–	–
	✔		40.06	0.4704	0.3458
✔			40.42	0.4904	0.3516
✔	✔		40.78	0.4964	0.3620
✔	✔	✔	**41.90**	**0.5001**	**0.3681**

Table 7: Image-to-voxel aggregation ablation.

Model	mIoU_A ↑	mIoU_D ↑	mIoU_S ↑	Latency (ms)	Memory (MB)
StreamAgg only	40.37	35.36	53.42	49.0	2,437
StreamAgg + Spatial Cross-Attention	40.72	35.77	**53.59**	95.2	3,554
StreamAgg + QueryAgg	**41.90**	**38.12**	53.31	83.3	2,788

we add an auxiliary detection head in (C) for dynamic-object supervision. This yields only a marginal gain (+0.9 mIoU on dynamic objects), proving that indirect supervision lacks the capacity to enrich these representations. In contrast, our QueryAgg module (D) directly injects instance-level dynamic object features into the corresponding voxel regions, yielding a substantial gain of $+2.7$ mIoU_D over (B). Overall, the combination of StreamAgg and QueryAgg achieves the best performance (41.9 mIoU), strengthening dynamic object representations while maintaining prediction accuracy for static objects.

Ablation of Refinement Module. Tab. 5 analyzes the effect of our Adaptive Residual Refinement module and its supervision. Naive voxel streaming yields sub-optimal result because the propagated voxel features suffer from warping-induced distortions during temporal alignment. Adopting Adaptive Residual Refinement (w/o supervision) alleviates these artifacts, improving performance to 39.84 mIoU (+1.12 mIoU) with minimal overhead (only +5 ms latency and +14 MB memory). Semantic Supervision further enhances temporal semantic consistency, while Geometry Supervision guides the refinement to focus on informative features, achieving the best performance of 40.37 mIoU without additional computational cost. Overall, each component contributes complementary gains, and the full StreamAgg provides a stable and spatially coherent voxel representation.

How to Utilize a Detector for Joint Learning? Tab. 6 presents an ablation study on how QueryAgg components improve both detection and occupancy prediction. The StreamAgg-only setting serves as the baseline. Applying a query-based detector on voxel features (V2Q) for multi-task learning, as in [29, 34], is inefficient due to the large search space and may fail to detect objects that are weakly represented in voxel features. Using an Image-to-Query detector (I2Q) provides finer image-based detection, but yields limited gains without voxel-level guidance. Combining I2Q with V2Q improves this by coupling semantic cues from images with geometric details from voxel features to guide dynamic object localization, yet still lacks direct voxel interaction. The complete StreamOcc further introduces direct query–voxel interaction by injecting instance-level features into voxel features, complementing dynamic object representations. This interactive design enhances both detection and occupancy prediction, which are built upon distinct representations, achieving the highest NDS, mAP, and mIoU.

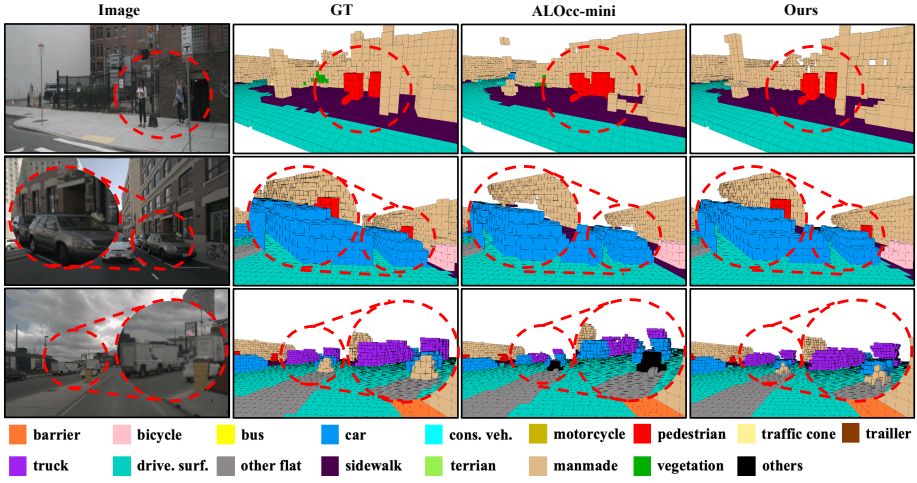


Fig. 4: Qualitative comparison with other SOTA method. StreamOcc enables more precise reconstruction of dynamic objects and more accurate semantic predictions for static objects than previous best real-time method, ALOcc-mini [2].

Effect of Query-guided Aggregation. Tab. 7 compares image-to-voxel aggregation methods for overcoming the limitations of voxel-based encoding. Prior works adopt Spatial Cross-Attention [26, 30, 38] to distribute image features across all voxel grids, providing slight improvements. However, spatial misalignment between voxel and image space often leads to hallucinated mappings [16], and the lack of explicit supervision on what to extract and where to place it results in suboptimal performance. In contrast, Query-guided Aggregation (QueryAgg) explicitly extracts dynamic object features and selectively integrates them into the voxel regions they occupy. This targeted aggregation strategy mitigates hallucinations, reduces unnecessary computation, and yields a +2.76 mIoU gain on dynamic objects and +1.53 overall, while maintaining high efficiency with 83.3 ms latency and 2,788 MB memory.

4.5 Qualitative Analysis

In Fig. 4, we compare StreamOcc with the previous state-of-the-art real-time method, ALOcc-mini [2]. StreamOcc produces predictions closer to the ground truth in the regions highlighted by red dotted circles. It reconstructs pedestrians more precisely, whereas ALOcc-mini produces overly smoothed pedestrian groups (Row 1), correctly captures even a partially occluded pedestrian that is missed by ALOcc-mini (Row 2), and better reconstructs distant moving vehicles with shapes closer to the ground truth, while also improving semantic predictions not only for dynamic objects but also for static objects (Row 3). Overall, these qualitative results support our quantitative findings that StreamOcc strengthens dynamic-object reconstruction while maintaining accurate semantics for static objects under real-time constraints.

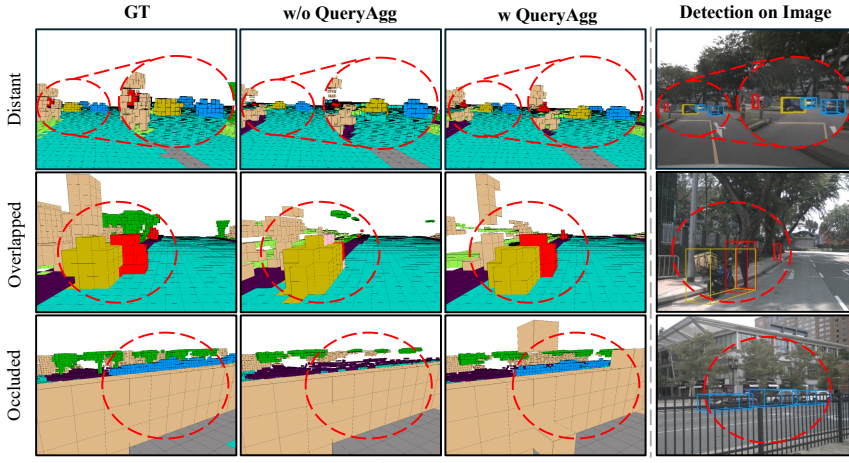


Fig. 5: Effect of QueryAgg. QueryAgg enables more accurate predictions for distant, overlapped, and occluded dynamic objects. The fourth column shows the detection outputs from the instance queries used in QueryAgg, and red dotted circles highlight the improved regions.

Additionally, Fig. 5 shows that QueryAgg effectively mitigates information loss induced by image-to-voxel projection, enhancing voxel representations of dynamic objects that appear small due to long-range distance, overlap with nearby instances, or are partially occluded. Compared to the model without QueryAgg, our method with QueryAgg yields predictions closer to the ground truth in these challenging scenarios. As illustrated in the rightmost column, the instance queries accurately detect dynamic objects, and QueryAgg performs targeted aggregation by injecting instance-level features from these dynamic queries into the voxel regions occupied by the corresponding objects. This design enables voxel features to retain a more precise representation of dynamic objects even under sparse projections for distant objects, severe overlap that causes feature mixing in coarse grids, and occlusions that lead to incomplete projections.

5 Conclusion

In this paper, we presented StreamOcc, a novel framework that utilizes dense voxel streaming for efficient and accurate 3D occupancy prediction. StreamOcc addresses the challenges of naive dense voxel streaming by introducing two aggregation strategies: StreamAgg, which enables temporally consistent streaming by mitigating warping-induced distortions in propagated voxel features, and QueryAgg, which enhances dynamic object representations by selectively injecting instance-level semantics into corresponding occupied voxel regions. Extensive experiments on Occ3D-nuScenes and SurroundOcc-benchmark demonstrate that StreamOcc achieves state-of-the-art performance, outperforming prior methods that utilize sparse representation while maintaining computational costs. Overall, StreamOcc unlocks dense voxel streaming as a practical paradigm and establishes a strong framework for future research in dense 3D scene understanding.

A Further Implementation Details

A.1 Lightweight 3D-FPN

Following BEVDepth [18], we lift image features into the 3D voxel space, obtaining $\mathbf{V}_{\text{init}} \in \mathbb{R}^{C\times X\times Y\times Z}$, which encodes both spatial and semantic information and serves as the foundation for 3D occupancy prediction. To improve computational efficiency while preserving essential structure, we apply a lightweight 3D-FPN that downsamples voxel features, avoiding excessively fine voxel grids that introduce unnecessary computation.

Specifically, $\mathbf{V}_{\text{init}}$ is passed through a Conv3D block to generate multi-scale voxel features: $\mathbf{V}_1 \in \mathbb{R}^{C_1\times \frac{X}{2}\times \frac{Y}{2}\times \frac{Z}{2}}$ and $\mathbf{V}_2 \in \mathbb{R}^{C_2\times \frac{X}{4}\times \frac{Y}{4}\times \frac{Z}{4}}$. These features, together with $\mathbf{V}_{\text{init}}$, are then uniformly resampled to a resolution of $\frac{X}{2} \times \frac{Y}{2} \times \frac{Z}{2}$ via trilinear interpolation. The interpolated volumes are concatenated and compressed using a Conv1D layer to obtain a compact representation $\mathbf{V}_{\text{down}} \in \mathbb{R}^{C\times \frac{X}{2}\times \frac{Y}{2}\times \frac{Z}{2}}$. This design substantially reduces computational overhead while maintaining spatial and semantic detail.

A.2 Query Selection for Dynamic Query Aggregation

Selecting reliable instance queries is critical for Dynamic Query Aggregation (DQA). Naively selecting queries solely based on IoU with ground-truth boxes may introduce shortcut learning, as queries that are spatially close to ground-truth can be mapped into voxel features even when their detection quality is poor. This may implicitly expose ground-truth information during training and degrade generalization at inference. To address this issue, we adopt a training-time query filtering strategy that combines confidence scores with IoU criteria, and geometric constraints. For sufficiently large objects, we apply an IoU-based rule and select queries that satisfy:

$$\text{IoU}(\hat{b}, b) > 0.4 \quad \text{and} \quad s > 0.3,$$

where $\hat{b}$ and b denote the predicted and ground-truth bounding boxes and s is the query confidence score.

For small objects, IoU becomes unstable because even minor deviations in center position or box size can significantly reduce IoU despite accurate detection. Therefore, we introduce a geometry-based criterion that measures the deviation between predicted and ground-truth box geometry. A query is selected if it satisfies:

$$(\sigma_c D_{\text{center}} + \sigma_b D_{\text{size}}) < 1.5 \quad \text{and} \quad s > 0.3,$$

where D_{center} denotes the center distance between $\hat{b}$ and b, and D_{size} measures the difference in box dimensions. The weighting coefficients are empirically set to $\sigma_c = 2.0$ and $\sigma_b = 1.0$. This criterion allows queries that capture object geometry accurately to be selected even when IoU is unreliable.

During inference, the filtering strategy is simplified by selecting only instance queries with confidence score $s > 0.3$, consistent with the detection threshold used in the detector.

A.3 Auxiliary Supervision Head for Refinement

To ensure that the Adaptive Residual Refinement module reliably learns to adjust voxel features, we incorporate two auxiliary supervision heads used only during training.

First, for **Geometry Supervision**, we upsample the spatial attention feature $\mathbf{M}_s$ to the original voxel resolution and apply the MLP-based decoder to predict a binary occupancy map $\mathbf{V}^t_{\text{geo}} \in \mathbb{R}^{1\times X\times Y\times Z}$. This prediction is supervised with the ground-truth occupied mask via binary cross-entropy loss, encouraging the refinement module to focus on informative features that are meaningful for reducing distortions. Additionally, we apply **Semantic Supervision** to the refined warped voxel feature $\mathbf{V}^t_{\text{refwarp}}$. Specifically, $\mathbf{V}^t_{\text{refwarp}}$ is decoded using the same upsampling and MLP-based decoder to predict the voxel semantic occupancy at the current timestep, $\mathbf{V}^t_{\text{sem}} \in \mathbb{R}^{Class\times X\times Y\times Z}$. This supervision explicitly guides the refinement module to correct warping-induced distortions and align the refined representation with the current scene state.

Since this refinement is applied to the propagated voxel features before they are combined with the current voxel features, it helps maintain temporal consistency and enables stable temporal accumulation across timesteps.

A.4 Loss Functions

To train our model, we combine multiple task-specific loss functions. For depth estimation, we adopt the depth loss $\mathcal{L}_{\text{depth}}$ from BEVDepth [18]. Voxel occupancy prediction is supervised using the cross-entropy-based occupancy loss $\mathcal{L}_{\text{occ}}$, applied to the predictions generated from the final voxel features V_{fin} via an MLP decoder. For the Image-to-Query Detector, we employ the detection loss $\mathcal{L}_{\text{det}}$ from Sparse4D v3 [23]. To further enhance voxel representations, we incorporate an Auxiliary Mask Decoder with the mask loss $\mathcal{L}_{\text{mask}}$.

In addition, the Semantic Head supervises the refined warped voxel features by predicting the current semantic occupancy distribution, which is optimized using a cross-entropy loss $\mathcal{L}_{\text{sem}}$. The Geometry Head predicts whether each voxel is occupied or empty and is supervised with a binary cross-entropy loss $\mathcal{L}_{\text{geo}}$.

The overall training objective $\mathcal{L}_{\text{total}}$ is defined as:

$$\begin{aligned}\mathcal{L}_{\text{total}} = \lambda_{\text{depth}}\cdot\mathcal{L}_{\text{depth}} + \lambda_{\text{occ}}\cdot\mathcal{L}_{\text{occ}} + \lambda_{\text{det}}\cdot\mathcal{L}_{\text{det}} \\ + \lambda_{\text{mask}}\cdot\mathcal{L}_{\text{mask}} + \lambda_{\text{sem}}\cdot\mathcal{L}_{\text{sem}} + \lambda_{\text{geo}}\cdot\mathcal{L}_{\text{geo}},\end{aligned} \tag{11}$$

where λ_{depth}, λ_{occ}, λ_{det}, λ_{mask}, λ_{sem}, and λ_{geo} denote the balancing weights for each loss term. In our experiments, these weights are empirically set to 0.05, 10.0, 0.2, 1.0, 10.0, and 10.0, respectively.

B More Experiments

B.1 Evaluation Metrics

We evaluate occupancy prediction using IoU and mIoU, and further adopt RayIoU as the primary evaluation metric following SparseOcc [24]. Unlike voxel-

level mIoU, RayIoU measures occupancy prediction along rays, jointly considering semantic correctness and depth accuracy. Specifically, a predicted ray is regarded as a true positive only when its semantic class matches the ground truth and the depth error falls within a predefined threshold (e.g., 1 m, 2 m, or 4 m). The metrics are defined as follows:

$$\text{mIoU/RayIoU} = \frac{1}{|C|} \sum_{i \in C} \frac{TP_i}{TP_i + FP_i + FN_i}, \tag{12}$$

$$\text{IoU} = \frac{TP_{c_0}}{TP_{c_0} + FP_{c_0} + FN_{c_0}}, \tag{13}$$

where TP_i, FP_i, and FN_i denote the numbers of true positives, false positives, and false negatives for class i, respectively, C is the set of semantic classes, and c_0 denotes the occupied class.

B.2 Ablation on Query Selection Strategy

To analyze the impact of the query selection strategy used in Dynamic Query Aggregation (DQA), we conduct an ablation study by varying the criteria used to select instance queries during training. In particular, we compare four strategies: selecting queries based only on IoU with ground-truth boxes, selecting based only on classification confidence, combining IoU and classification scores, and our full strategy that additionally incorporates geometry-based constraints for small objects.

As shown in Table 8, selecting queries based solely on IoU yields limited performance (39.9 mIoU), as it may select queries that are spatially close to ground truth but poorly detected. Using only classification confidence slightly improves performance (40.1 mIoU), but still lacks reliable spatial alignment. Combining IoU and classification scores significantly improves performance (41.3 mIoU) by filtering unreliable detections.

Our full query selection strategy further incorporates geometry-based constraints for small objects, addressing the instability of IoU for small bounding boxes. This strategy achieves the best performance of **41.9** mIoU, demonstrating that geometry-aware query filtering effectively improves the reliability of instance queries used in DQA.

Table 8: Ablation study on query selection strategy for Dynamic Query Aggregation.

IoU	Cls	Geo	mIoU ↑
✔			39.9
	✔		40.7
✔	✔		41.3
✔	✔	✔	**41.9**

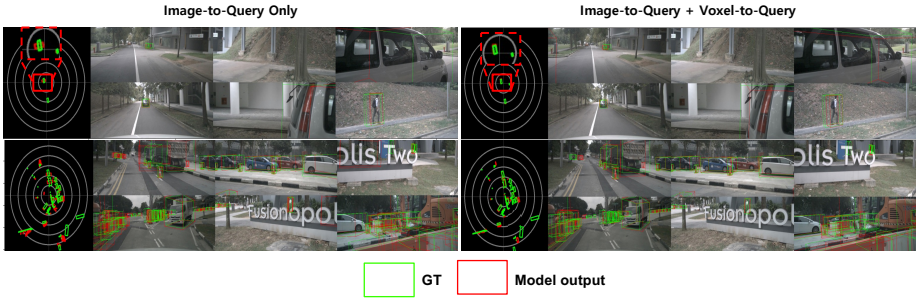


Fig. 6: Qualitative comparison of features used for detection. The left column shows detection results using only image features, whereas the right column additionally leverages voxel features that provide complementary geometric information. Incorporating voxel-level geometric cues reduces false positives arising along rays and enables more precise 3D detections.

B.3 More visualization

Effect of voxel features on detection. Figure 6 provides a qualitative comparison of the features used for detection. The left column shows detection results using only image features, whereas the right column additionally leverages voxel features that provide complementary geometric information. By incorporating voxel-level geometric cues, the detector reduces false positives arising along rays and produces more precise 3D detections.

Effect of QueryAgg on occupancy prediction. Figure 7 further demonstrates the effectiveness of StreamOcc across three representative scenarios by comparing the ground truth, results without QueryAgg, results with QueryAgg, and the detection outputs from the instance queries. In the first row, QueryAgg enhances the representation of distant pedestrians. In the second row, it helps separate nearby objects and more accurately capture their occupancy states. In the third row, QueryAgg enables the occupancy map to recover a truck that is partially occluded by a fence. Since the fence obstructs the truck in the image view, its semantic information is not reliably projected into 3D voxel space. By directly aggregating object-level information from instance queries, QueryAgg allows the truck to be more accurately represented in the occupancy map.

Additional qualitative results. Figures 8 to 11 present additional qualitative results in complex driving scenarios, further validating the robustness of the proposed framework.

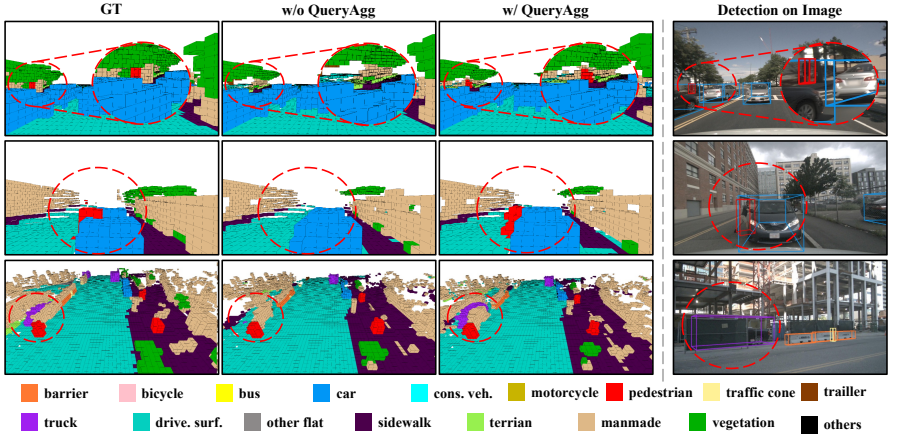


Fig. 7: Visualization results of StreamOcc across three distinct scenarios. Each row shows the ground-truth (GT), results without QueryAgg (w/o QueryAgg), with QueryAgg (w/ QueryAgg), and detection outputs from the instance queries.

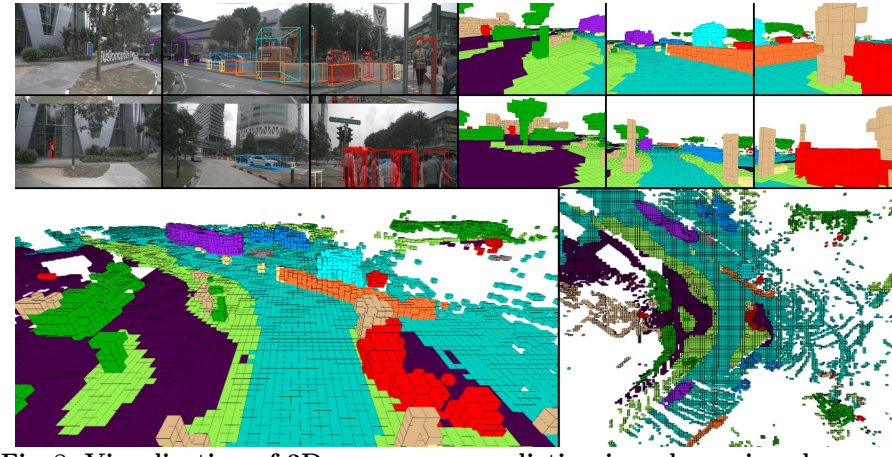


Fig. 8: Visualization of 3D occupancy prediction in a dynamic urban scene with construction and pedestrians. The top-left shows object detection on the images, the top-right presents occupancy prediction in the camera view, the bottom-left illustrates the top-front view, and the bottom-right depicts the top-down occupancy prediction results.

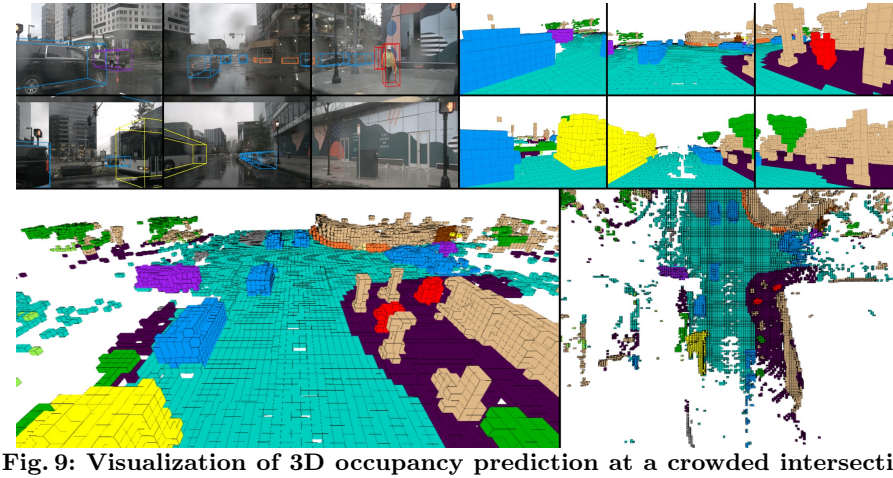
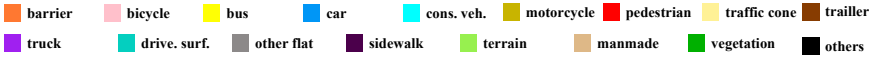


Fig. 9: Visualization of 3D occupancy prediction at a crowded intersection on a rainy day. The top-left shows object detection in the image space, the top-right presents occupancy prediction in the camera view, the bottom-left illustrates the top-front view, and the bottom-right depicts the top-down occupancy prediction results.



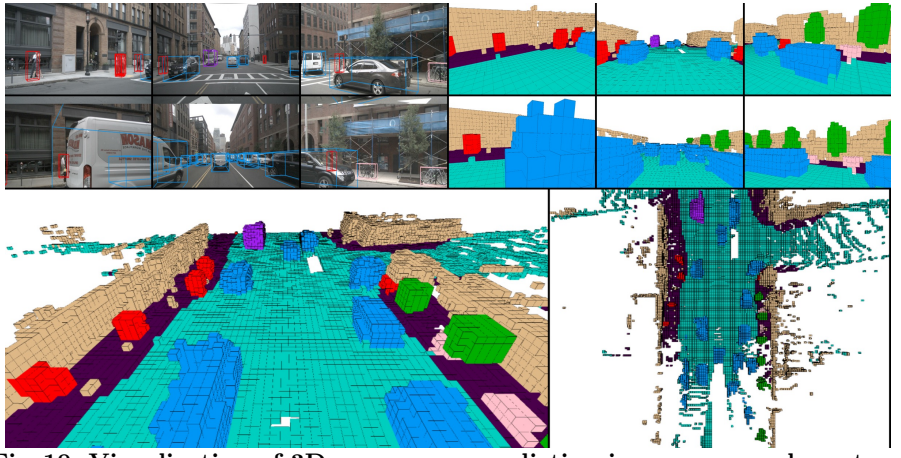


Fig. 10: Visualization of 3D occupancy prediction in a narrow urban street with parked and moving vehicles, pedestrians, and bicycles. The top-left shows object detection in the image space, the top-right presents occupancy prediction in the camera view, the bottom-left illustrates the top-front view, and the bottom-right depicts the top-down occupancy prediction results.

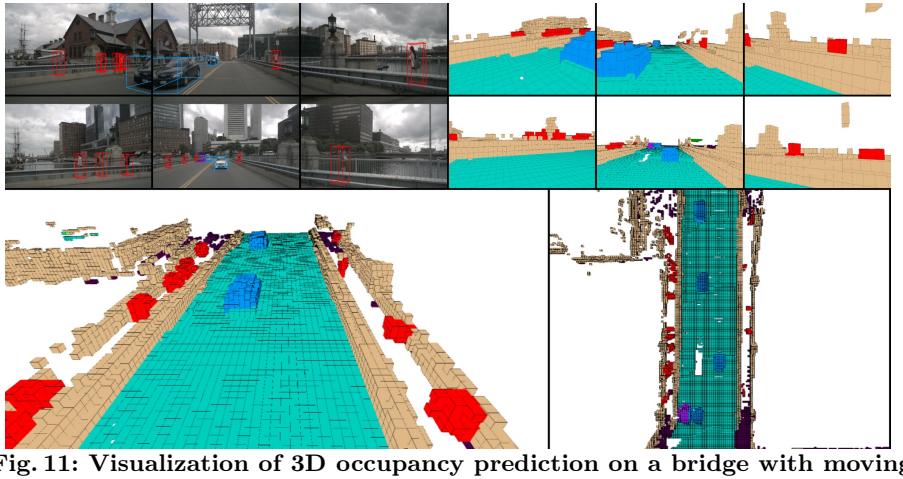
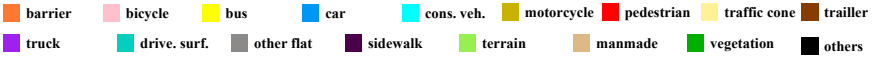


Fig. 11: Visualization of 3D occupancy prediction on a bridge with moving vehicles and pedestrians. The top-left shows object detection in the image space, the top-right presents occupancy prediction in the camera view, the bottom-left illustrates the top-front view, and the bottom-right depicts the top-down occupancy prediction results.



References

1. 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. 11621–11631 (2020)
2. Chen, D., Fang, J., Han, W., Cheng, X., Yin, J., Xu, C., Khan, F.S., Shen, J.: Alocc: Adaptive lifting-based 3d semantic occupancy and cost volume-based flow predictions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4156–4166 (2025)
3. Han, C., Yang, J., Sun, J., Ge, Z., Dong, R., Zhou, H., Mao, W., Peng, Y., Zhang, X.: Exploring recurrent long-term temporal fusion for multi-view 3d perception. IEEE Robotics and Automation Letters **9**(7), 6544–6551 (2024)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
5. He, Y., Chen, W., Xun, T., Tan, Y.: Real-time 3d occupancy prediction via geometric-semantic disentanglement. arXiv preprint arXiv:2407.13155 (2024)
6. Hou, J., Li, X., Guan, W., Zhang, G., Feng, D., Du, Y., Xue, X., Pu, J.: Fastocc: Accelerating 3d occupancy prediction by fusing the 2d bird's-eye view and perspective view. ICRA (2024)
7. Huang, J., Huang, G.: Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. arXiv preprint arXiv:2203.17054 (2022)
8. Huang, J., Huang, G., Zhu, Z., Ye, Y., Du, D.: Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. arXiv preprint arXiv:2112.11790 (2021)
9. Huang, Y., Thammatadatrakoon, A., Zheng, W., Zhang, Y., Du, D., Lu, J.: Gaussianformer-2: Probabilistic gaussian superposition for efficient 3d occupancy prediction. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 27477–27486 (2025)
10. Huang, Y., Zheng, W., Zhang, Y., Zhou, J., Lu, J.: Tri-perspective view for vision-based 3d semantic occupancy prediction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9223–9232 (2023)
11. Huang, Y., Zheng, W., Zhang, Y., Zhou, J., Lu, J.: Gaussianformer: Scene as gaussians for vision-based 3d semantic occupancy prediction. In: European Conference on Computer Vision. pp. 376–393. Springer (2024)
12. Kim, J., Kang, C., Lee, D., Choi, S., Choi, J.W.: Protoocc: Accurate, efficient 3d occupancy prediction using dual branch encoder-prototype query decoder. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 4284–4292 (2025)
13. Koh, J., Lee, Y., Kim, J., Lee, D., Choi, J.W.: Onlinebev: Recurrent temporal fusion in bird's eye view representations for multi-camera 3d perception. IEEE Transactions on Intelligent Transportation Systems (2025)
14. Li, C., Gu, Z., Chen, G., Huang, L., Zhang, W., Zhou, H.: Real-time stereo-based 3d object detection for streaming perception. Advances in Neural Information Processing Systems **37**, 115468–115490 (2024)
15. Li, J., He, X., Zhou, C., Cheng, X., Wen, Y., Zhang, D.: Viewformer: Exploring spatiotemporal modeling for multi-view 3d occupancy perception via view-guided transformers. In: European Conference on Computer Vision. pp. 90–106. Springer (2024)

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. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9087–9098 (2023)
17. Li, Y., Bao, H., Ge, Z., Yang, J., Sun, J., Li, Z.: Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo. In: AAAI. pp. 1486–1494 (2023)
18. 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. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 1477–1485 (2023)
19. 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)
20. Li, Z., Yu, Z., Austin, D., Fang, M., Lan, S., Kautz, J., Alvarez, J.M.: Fb-occ: 3d occupancy prediction based on forward-backward view transformation. arXiv preprint arXiv:2307.01492 (2023)
21. Liao, Z., Wei, P., Chen, S., Wang, H., Ren, Z.: Stcocc: Sparse spatial-temporal cascade renovation for 3d occupancy and scene flow prediction. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 1516–1526 (2025)
22. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
23. Lin, X., Pei, Z., Lin, T., Huang, L., Su, Z.: Sparse4d v3: Advancing end-to-end 3d detection and tracking. arXiv preprint arXiv:2311.11722 (2023)
24. Liu, H., Wang, H., Chen, Y., Yang, Z., Zeng, J., Chen, L., Wang, L.: Fully sparse 3d panoptic occupancy prediction. arXiv preprint arXiv:2312.17118 (2023)
25. Loshchilov, I.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
26. Ma, Q., Tan, X., Qu, Y., Ma, L., Zhang, Z., Xie, Y.: Cotr: Compact occupancy transformer for vision-based 3d occupancy prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19936–19945 (2024)
27. Oh, G., Kim, S., Ko, H., Chi, H.g., Kim, J., Lee, D., Ji, D., Choi, S., Jang, S., Kim, S.: 3d occupancy prediction with low-resolution queries via prototype-aware view transformation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 17134–17144 (2025)
28. Shi, Y., Zhu, Y., Han, S., Jeong, J., Ansari, A., Cai, H., Porikli, F.: Odg: Occupancy prediction using dual gaussians. arXiv preprint arXiv:2506.09417 (2025)
29. Sima, C., Tong, W., Wang, T., Chen, L., Wu, S., Deng, H., Gu, Y., Lu, L., Luo, P., Lin, D., et al.: Scene as occupancy. arXiv preprint arXiv:2306.02851 (2023)
30. Tan, X., Wu, W., Zhang, Z., Fan, C., Peng, Y., Zhang, Z., Xie, Y., Ma, L.: Geocc: Geometrically enhanced 3d occupancy network with implicit-explicit depth fusion and contextual self-supervision. arXiv preprint arXiv:2405.10591 (2024)
31. 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)
32. Wang, J., Liu, Z., Meng, Q., Yan, L., Wang, K., Yang, J., Liu, W., Hou, Q., Cheng, M.: Opus: occupancy prediction using a sparse set (2024)
33. Wang, S., Liu, Y., Wang, T., Li, Y., Zhang, X.: Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3621–3631 (2023)

34. Wang, Y., Chen, Y., Liao, X., Fan, L., Zhang, Z.: Panoocc: Unified occupancy representation for camera-based 3d panoptic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 17158–17168 (2024)
35. Wei, Y., Zhao, L., Zheng, W., Zhu, Z., Zhou, J., Lu, J.: Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 21729–21740 (2023)
36. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018)
37. Yuan, T., Liu, Y., Wang, Y., Wang, Y., Zhao, H.: Streammapnet: Streaming mapping network for vectorized online hd map construction. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 7356–7365 (2024)
38. Zhang, Y., Zhu, Z., Du, D.: Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9433–9443 (2023)
39. 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)
40. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)
41. Zong, Z., Song, G., Liu, Y.: Detrs with collaborative hybrid assignments training. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6748–6758 (2023)
42. Zuo, S., Zheng, W., Han, X., Yang, L., Pan, Y., Lu, J.: Quadricformer: Scene as superquadrics for 3d semantic occupancy prediction. arXiv preprint arXiv:2506.10977 (2025)
43. Zuo, S., Zheng, W., Huang, Y., Zhou, J., Lu, J.: Gaussianworld: Gaussian world model for streaming 3d occupancy prediction. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 6772–6781 (2025)