This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

STUR3D: Spatio-Temporal Unified Representation Learning for 3D Object Detection

Huijie Fan[1†], Pengrui Huang[1,2†], Qiang Wang[3*], Baojie Fan[4], Jiahua Dong[5], Liangqiong Qu[6]
[1]State Key Laboratory of Robotics and Intelligent Systems,
Shenyang Institute of Automation, Chinese Academy of Sciences
[2]Shenyang University of Chemical Technology
[3]Key Laboratory of Manufacturing Industrial Integrated Automation, Shenyang University
[4]Nanjing University of Posts and Telecommunications
[5]Mohamed bin Zayed University of Artificial Intelligence [6]The University of Hong Kong
{fanhuijie,huangpengrui}@sia.cn,wangqiang@syu.edu.cn,
{jobfbj,dongjiahua1995}@gmail.com,liangqqu@hku.hk

Abstract

Existing surrounding-view 3D object detectors initialize high-confidence queries using current 2D information, while leveraging historical 3D features as priors. However, such heavy reliance on 2D cues introduces spatio-temporal inconsistencies between 2D and 3D representations. Specifically, 2D cues lack sufficient spatial information, limiting 3D localization capability. Moreover, insufficient temporal interaction often leads to object omission under occlusion. To address these challenges, we propose STUR3D, a unified framework establishing spatio-temporal alignment between 2D and 3D perception. First, we project temporal 3D features to the 2D image plane, empowering the 2D detector to distill representations essential for 3D localization, harmonizing cross-dimensional information. Second, we inject temporal cues into 2D detection, fostering spatio-temporal reasoning, and ensuring robust 3D detection under dynamic scenes and occlusion. Additionally, we embed depth-aware geometric cues into features for 2D-to-3D lifting, mitigating inherent ambiguities. Extensive nuScenes experiments validate STUR3D, achieving SOTA on the `test` *set with 57.9% mAP and 64.6% NDS. Our code will be released at https://github.com/snowindog/STUR3D.*

1. Introduction

3D object detection aims to identify objects and estimate their 3D positions and dimensions. It is fundamental to autonomous driving, robotics, and navigation. Recently, surrounding-view 3D detection [6, 27] has shown great potential with camera-only 2D inputs at a lower computational cost than LiDAR-based methods. Existing methods [16, 17, 26, 27] typically adopt a query-driven detection framework following DETR3D [27]. These approaches generally initialize queries and propagate them across frames for temporal context. Although temporal query propagation improves performance, heavy reliance on sampling 2D image-plane features causes 3D detectors to struggle in complex scenarios where powerful 2D detectors succeed.

To address this, recent studies introduce a 2D-to-3D pipeline [7, 8, 25, 28], leveraging 2D detectors as reliable priors to refine or initialize 3D queries. Specifically,

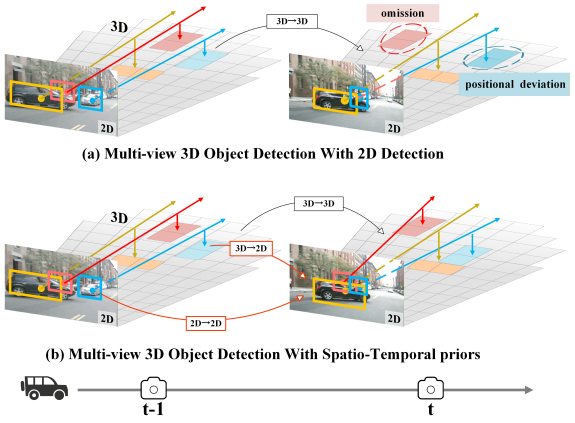


Figure 1. **Comparison of 3D detection frameworks.** (a) Existing methods suffer spatio-temporal discrepancy from over-reliance on 2D detection. (b) STUR3D enhances 2D–3D spatial representation consistency via comprehensive spatio-temporal interactions.

† Equal contribution.
* Corresponding author.

methods such as MV2D [28] project RoI 2D features into 3D space via camera parameters. However, this design typically assigns higher confidence to 3D queries initialized from 2D information, while treating historically propagated 3D cues with less emphasis. Consequently, these approaches inherit inherent limitations in both spatial and temporal modeling. From a spatial view, 3D space contains geometric dimensions including depth, scale, and orientation that 2D features do not explicitly model. Thus, 3D queries solely from 2D representations lack geometric grounding, causing spatial discrepancies and suboptimal localization after lifting. This issue is further exacerbated by the continued dominance of 2D cues during subsequent query refinement. From a temporal perspective, these methods underutilize the temporal dynamics encoded in historical 3D detections. When occluded, the model cannot effectively leverage accumulated 3D cues to maintain attention on the target [7], causing temporal inconsistencies and object omission, as shown in Fig. 1(a).

To address these limitations, we propose STUR3D, a unified framework that unifies 2D and 3D spatio-temporal representations through enriched cross-frame and cross-modal information exchange. First, we introduce the Spatio-Temporal Object Prior Propagator (STOPP), which performs both 2D-to-2D and 3D-to-2D temporal propagation. By reprojecting historical 2D and 3D detections onto the current image plane, STOPP generates explicit spatio-temporal priors that guide the 2D detector toward representation learning while suppressing noisy or inconsistent proposals. This enables the detector to retain attention on occluded objects, reducing false alarms via historical evidence. Second, we develop the Spatio-Temporal Geometry Encoder (STGE) to further inject temporal and geometric cues into the 2D detection head. Leveraging a depth-aware geometric attention mechanism, STGE fuses historical 2D and 3D detection features with current image features and filters out unreliable foreground hypotheses. This encoding reduces the representational gap between 2D and 3D feature spaces and facilitates the extraction of discriminative representations essential for accurate 3D localization in complex environments. Additionally, we introduce the Object-center Query Generator (OQG), which lifts refined 2D detections into 3D queries using depth-guided positional encoding. By infusing pseudo-depth cues, OQG enhances geometric grounding of 3D queries and mitigates the inherent spatial biases introduced by relying solely on appearance-based 2D features. Together, these form a unified spatio-temporal pipeline that harmonizes 2D and 3D representations, improving temporal coherence and increasing robustness in dynamic and occluded scenarios.

The contributions of our method can be summarized as:

- We present **STUR3D**, a spatio-temporal unified representation learning framework that aligns heterogeneous 2D and 3D modalities through cross-frame reprojection, addressing the spatial and temporal discrepancies that limit existing 2D-to-3D pipelines.
- We introduce three complementary modules, **STOPP**, **STGE**, and **OQG** that incorporate historical 2D and 3D priors, depth-aware geometric cues, and object-centered lifting, enabling cross-frame reasoning and producing geometry-consistent 3D queries.
- STUR3D achieves state-of-the-art performance on the nuScenes, obtaining **57.9% mAP** and **64.6% NDS** on the `test` set, and consistently surpasses strong query-based baselines across various backbones and input resolutions.

2. Related Work

2.1. Camera-based 3D Object Detection.

Surrounding-view detection frameworks perceive multiple images from different cameras and fully exploit the geometric correlations among views and camera poses to predict 3D bounding boxes in the world coordinate space. Based on different ways of feature representation and data processing, existing surrounding-view 3D detection methods can generally be divided into two categories: dense BEV-based methods [2, 6, 10–12, 20, 21, 33] and sparse query-based [5, 16, 17, 19, 27, 29] methods.

BEV-based methods. BEV-based methods construct a BEV feature map, and then perform downstream tasks through a 3D head. LSS series [6, 11, 21] lift surrounding-view features into 3D space and splats to BEV feature map. Several works [2, 10, 20, 35], improve object detection performance by propagating long-term historical features.

Sparse query-based methods. Sparse query-based methods [22, 27, 30] sample image features with queries and uses a decoder network to detect objects. The DETR3D series [19, 27] use a set of queries to implicitly transform features from 2D to 3D space. The PETR series [16, 17] explicitly inject 3D positional information into image features before decoding. StreamPETR [26] develops an efficient object-centric temporal fusion mechanism that propagates historical 3D decoder embeddings as contextual cues. 3DPPE series [5, 23, 29] introduces depth cues to enhance 3D object detection, with reliable supervision provided by LiDAR signals.

2.2. Visual Promp of 3D Object Detection.

Recent studies [7, 8, 22, 25, 28] have shown that introducing 2D detection as auxiliary supervision provides valuable cues for improving 3D object detection. Most methods first perform detection in the perspective view and then transform the results to initialize 3D queries. MV2D [28] learns query content from 2D detection results. Far3D [8] leverages 2D detection priors and depth estimation to generate adaptive 3D queries, improving long-range object percep-

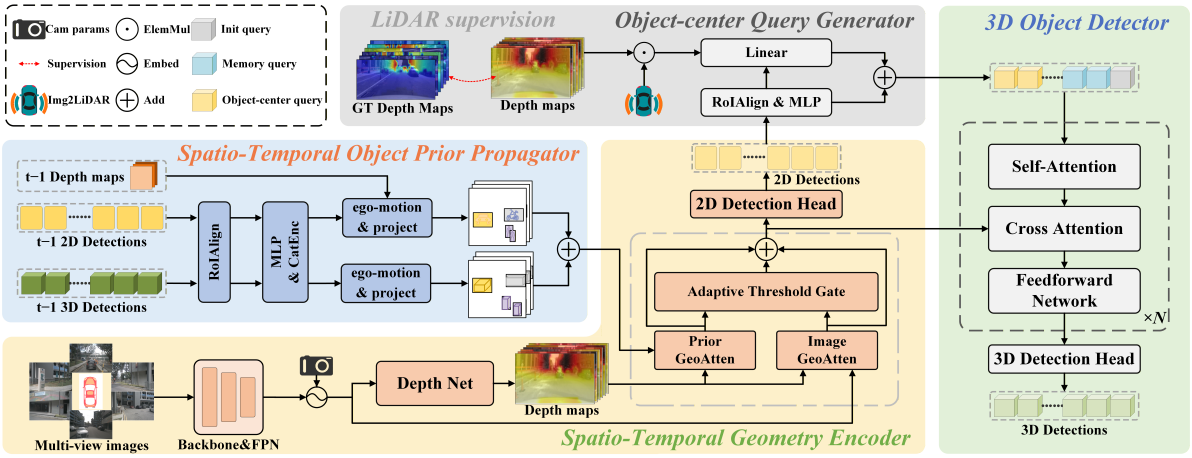


Figure 2. **Overall architecture of STUR3D.** STUR3D achieves temporal interactions from 2D-to-2D and 3D-to-2D via a Spatio-Temporal Object Prior Propagator (STOPP) and a Spatio-Temporal Geometry Encoder (STGE), and spatial interactions from pseudo-3D to 2D through the Object-center Query Generator, alleviating temporal and spatial misalignment caused by over-reliance on 2D detections.

tion. QAF2D [7] generates 3D query anchors from 2D detection results by associating sampled points within 2D bounding boxes with depth, orientation, and size candidates, and refines them through validity filtering for more reliable 3D query. DVPE [25] incorporates 2D RoI features into object-centric temporal modeling.

3. Method

3.1. Overview Architecture

The overview architecture of the proposed STUR3D is shown in Fig. 2. At the frame t, we utilize depth maps, 2D and 3D detections from $t-1$ frame as prior information and the surrounding-view images at t frame as the current information by using a backbone network [3, 9] and a FPN neck [13]. Our STUR3D framework comprises three core components: Spatio-Temporal Object Prior Propagator (STOPP), Spatio-Temporal Geometry Encoder (STGE), and Object-center Query Generator (OQG), followed by a 3D perception head. The STOPP module takes prior information as input to establish spatial alignment between 2D and 3D temporal information. Subsequently, the STGE module fuses these spatio-temporal priors with current image features under depth guidance, performing 2D object detection. The OQG module then refines generated pseudo-point clouds via alignment with 2D detection results, eliminating their inherent deviations and yielding more accurate 3D target localization. Finally, a 3D perception head generates the final 3D detection results.

3.2. Spatio-Temporal Object Prior Propagator

In surrounding-view temporal 3D object detection, detection results verified in previous frames are often geometrically reliable and semantically stable. Instead of relying on implicit feature propagation or global fusion, the core idea of STOPP is to explicitly reuse reliable historical detections and convert them into structured spatio-temporal object priors through temporal propagation, projection, and semantic injection, as shown in Fig. 3. This design enables the detector to focus on consistent foreground regions across time and suppress background noise. The STOPP module employs two parallel branches for 2D and 3D Object localization. The 2D branch uses depth information from the prior frame to perform temporal conversion, while the 3D branch reprojects the object localization onto the 2D plane, thereby enforcing cross-modal consistency and eliminating localization discrepancies. Finally, we feed the generated spatio-temporal object prior information into the following STGE module for the object detection of current frames.

Historical Object Prior Definition. At frame $t-1$, we define the high-confidence 2D and 3D detections as:

$$D^{2D}_{t-1} = \{(B^{2D}_{i,t-1}, c_{i,t-1}, f^{2D}_{i,t-1})\}^{M}_{i=1}, \quad (1)$$

$$D^{3D}_{t-1} = \{(B^{3D}_{j,t-1}, c_{j,t-1}, f^{3D}_{j,t-1})\}^{N}_{j=1}, \quad (2)$$

where M and N represent the number of boxes, $B^{2D}_{i,t-1} = (u_1, v_1, u_2, v_2)$ is a 2D box, where locations are constrained within certain image regions, $B^{3D}_{j,t-1} = (c_x, c_y, c_z, w, h, l, \cos\theta, \sin\theta, v_x, v_y)$ is a 3D box. Here c_x, c_y, c_z are center coordinates, w, h, l represent the weight, length and height, θ is the yaw angle of the 3D box, and v_x, v_y are velocities of the box. $c \in \mathbb{R}^{10}$ is the category label, and $f^{2D}_{i,t-1}, f^{3D}_{j,t-1} \in \mathbb{R}^{C\times y\times y}$ denote RoI-Aligned features of each box, where y represent the fixed-length.

Feature Extraction. Given surrounding-view feature maps $F_{t-1} = \{F^{v}_{t-1} \in \mathbb{R}^{C\times H_f\times W_f}\}^{Ncam}_{v=1}$ at frame $t-1$, where C, H_f and W_f represent the number of feature channels, height, and width of the feature maps. For the two

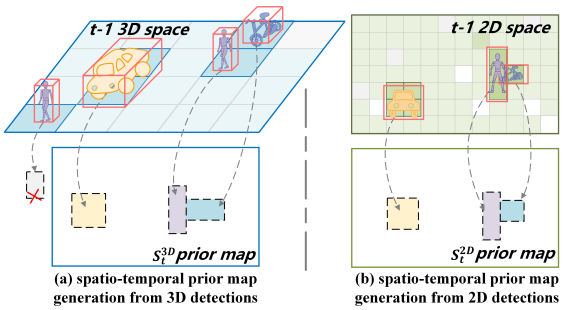


Figure 3. **Overview of the STOPP.** The STOPP injects spatial representations from 3D space and rich semantic context into the current 2D features through (a) 3D-to-2D and (b) 2D-2D temporal interactions, enhancing the consistency between 2D-based queries and 3D spatial representations. Meanwhile, historical detections guide the model to focus on potentially occluded targets.

branches, we first extract the 2D object features using a RoIAlign [4] operation:

$$f_{i,t-1}^{2D} = \text{RoIAlign}(F_{t-1}^{v}, B_{i,t-1}^{2D}). \tag{3}$$

For each 3D bounding box, we apply a geometric projection operator $\Pi(\cdot)$ to project it onto the image plane of each view, yielding the feature $f_{j,t-1}^{3D}$. This is formulated as:

$$f_{j,t-1}^{3D} = \text{RoIAlign}(F_{t-1}^{v}, \Pi(\mathbf{K}_{t-1}, B_{j,t-1}^{3D})), \tag{4}$$

where $\mathbf{K}_{t-1}$ represents the lidar2img projection matrix that combines the camera's intrinsic and extrinsic parameters.

Temporal Geometric Alignment and Projection. To maintain temporal consistency, historical 3D boxes and 2D boxes are aligned to the current frame using ego-motion and camera parameters. Given the ego-motion matrix $\mathbf{T}_{t-1\to t}$, each 3D bounding box from the previous frame is transformed and reprojected to current image views:

$$B_{j,t}^{3D|2D} = \Pi(\mathbf{K}_t, \mathbf{T}_{t-1\to t} \cdot B_{j,t-1}^{3D}). \tag{5}$$

For 2D boxes, we sample the depth values $d_{i,t-1}$ from the predicted depth map of the previous frame $D_{t-1} = \{D_{t-1}^{v} \in \mathbb{R}^{1\times H_f \times W_f}\}_{v=1}^{N_{cam}}$ at the center of each 2D box $(\bar{u}_i, \bar{v}_i)$. Then, we can construct the pseudo-3D center with the depth values, which are temporally propagated as:

$$B_{i,t}^{2D} = \Pi(\mathbf{K}_t, \mathbf{T}_{t-1\to t} \cdot \mathbf{K}_{t-1}^{-1} \cdot [\bar{u}_i, \bar{v}_i, 1] d_{i,t-1}). \tag{6}$$

All projected bounding boxes follow the same visibility principle, discarding those outside the field of view.

Prior Map Generation. To construct object semantic priors, we encode category and appearance features into compact object embeddings using an MLP:

$$e_o = \text{MLP}\big(f_{o,t-1} \oplus \text{CatEnc}(c_{o,t-1})\big), \tag{7}$$

where $f_{o,t-1}$ denotes the object feature of object o, CatEnc is Category Encoder, and $e_o \in \mathbb{R}^C$ represents the resulting object embedding. For each temporally aligned projected box $B_{o,t}$, we construct a spatial prior mask on the feature map of size $H_f \times W_f$:

$$M_o(i,j) = \begin{cases} 1, & (i,j) \in B_{o,t} \\ 0, & \text{otherwise} \end{cases}, \tag{8}$$

where (i, j) indicates the spatial position on the feature map. The object embedding e_o is injected into its corresponding spatial region and aggregated across all objects with normalization to handle overlapping projections:

$$S_t = \frac{\sum_{o=1}^{O} e_o \otimes M_o}{\max\Big(1, \sum_{o=1}^{O} M_o\Big)}, \tag{9}$$

where $\otimes$ represents outer product, and $S_t = \{S_t^v \in \mathbb{R}^{C\times H_f \times W_f}\}_{v=1}^{N_{\text{cam}}}$ represents a generic formulation for generating surrounding-view object semantic prior maps. This unified process Eq. (7, 8, 9) is applied to obtain two types of spatio-temporal object prior maps, S_t^{2D} and S_t^{3D}, corresponding to priors derived from 2D and 3D box features.

The proposed STOPP module has several merits. First, it suppresses background interference, enabling object localization. Second, it leverages temporal cues to mitigate occlusion effects. Third, it eliminates discrepancies between 2D and 3D detection, achieving cross-modal alignment.

3.3. Spatio-Temporal Geometry Encoder

After obtaining the spatio-temporal object prior maps from STOPP, we proceed to unify spatio-temporal reasoning across the 2D and 3D paradigms and guide the 2D detector to distill the representations required for 3D detection, as shown in Fig. 4. Given the image features $F_t = \{F_t^v \in \mathbb{R}^{C\times H_f \times W_f}\}_{v=1}^{N_{cam}}$, we first encode the intrinsic camera parameters through an MLP. Then, a lightweight depth estimation network DepthNet$(\cdot)$ [23] composed of residual and convolutional blocks predicts both depth regression and probabilistic depth distributions. Finally, we obtain per-pixel depth maps $D_t = \{D_t^v \in \mathbb{R}^{1\times H_f \times W_f}\}_{v=1}^{N_{cam}}$, supervised by LiDAR depth signals.

Inspired by Deformerv2 [32], each pixel (u, v) in the depth map possesses a unique 2D spatial coordinate, and its depth value encodes a relative distance relation to other pixels. Based on this observation, we generate two types of geometric priors: the image-plane prior and the depth-distance prior. By integrating these two priors, a geometric prior $\mathcal{X}$ is obtained, with the shape $H_f W_f \times H_f W_f$. We use a geometry-attention operation to inject these geometric priors into visual features:

$$\text{GeoAtten}(F, \mathcal{X}) = \big(\text{Softmax}(QK^{\top}) \odot \beta^{\mathcal{X}}\big)V, \tag{10}$$

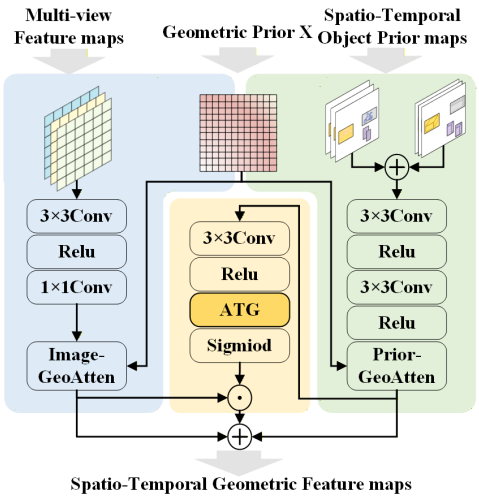


Figure 4. **Overview of the Spatio-Temporal Geometry Encoder.** Through a series of simple convolutional operations, gating mechanisms, and attention modules, STGE generates spatio-temporal geometric feature maps that improve localization and occlusion recovery.

where Q, K, V are query, key, and value matrices obtained via linear projections, $\odot$ denotes element-wise multiplication, and $\beta \in (0, 1)$ is a learnable decay rate controlling the influence of $\mathcal{X}$.

Following Eq. 10, the surrounding-view feature maps F_t and the combined spatio-temporal object prior maps S_t^{2D} and S_t^{3D} are processed independently. The feature maps F_t and the combined priors $S_t^{2D} + S_t^{3D}$ first undergo simple convolutional operations before being passed through their respective GeoAtten modules.

$$F_t' = \text{Image-GeoAtten}(F_t, \mathcal{X}), \tag{11}$$

$$S_t = \text{Prior-GeoAtten}(S_t^{2D} + S_t^{3D}, \mathcal{X}). \tag{12}$$

Then, we design an Adaptive Threshold Gate (ATG) to generate a spatial mask, which emphasizes high-confidence foreground regions. We define the final spatio-temporal geometric feature maps as follows:

$$F_{sp,t} = F_t' + F_t' \odot \sigma(\text{ATG}((S_t - \gamma)\tau) + S_t, \tag{13}$$

where $F_{sp,t} = \{F_{sp,t}^v \in \mathbb{R}^{C \times H_f \times W_f}\}_{v=1}^{N_{cam}}$ are the final spatio-temporal geometric feature maps, which can be used for 2D detection. $\sigma(\cdot)$ is the sigmoid function. The Adaptive Threshold Gate (ATG) uses two learnable parameters, γ and τ, where γ controls the threshold shift and τ adjusts the smoothness of the spatial mask.

The proposed STGE module enables the 2D detector to distill and directly supply the feature representations actually consumed by the 3D head, including the 3D cues that previously had to be relayed across frames via the 2D head, thereby narrowing the gap between 2D and 3D representations. Additionally, depth values enhance object boundary clarity, guiding the 2D detector toward more accurate localization.

3.4. Object-center Query Generator

The spatio-temporal geometric feature maps F_t are fed into the 2D detection head to produce updated 2D bounding boxes. Based on these refined 2D boxes, we generate object-centric 3D queries as the input for the 3D decoder, establishing an explicit bridge between 2D perception and 3D reasoning.

Through RoIAlign [4] and MLP, we obtain the feature embeddings $e_t^{2d} \in \mathbb{R}^C$ of each detection box. The $\mathbf{K}_t^{-1}$ is employed to generate coarse positional encoding $P^{e'} \in \mathbb{R}^{3\times M}$, where M represents the number of 2D boxes.

For each detection, we also extract the depth value beneath the bounding box center from the depth maps, concatenate the depth with the center coordinates to form pseudo-3D points, and then generate the 3D object-center locations $P^o \in \mathbb{R}^{3\times M}$ by transforming these coordinates through the $\mathbf{K}_t^{-1}$. The coarse positional encoding $P^{e'} \in \mathbb{R}^{3\times M}$ and 3D object-center locations $P^o \in \mathbb{R}^{3\times M}$ are fed into a $Linear$ layer to get offset that enriches the positional features representation:

$$P^e = Linear(concat(P^{e'}, P^o)) + P^{e'}, \tag{14}$$

where $P^e \in \mathbb{R}^{3\times M}$ are the final refined positional encoding, and its contributions are two-fold. First, reliable 2D detections provide corrective cues for pseudo-point clouds. Second, with the help of positional encoding, we generate refined 3D points, thereby suppressing error propagation to the 3D detection pipeline.

4. Experiments

4.1. Datasets and Metrics

We evaluate our method on the large-scale nuScenes dataset [1], a widely used benchmark for surrounding-view 3D object detection in autonomous driving. It contains 1,000 scenes divided into 700 for training, 150 for validation, and 150 for testing, each about 20 seconds long and annotated at 2 Hz. The dataset provides 32-beam LiDAR point clouds and six camera views covering 360 degrees. Following the official protocol, we report mAP and NDS, along with mATE, mASE, mAOE, mAVE, and mAAE for detailed analysis.

4.2. Implementation Details

We follow previous approaches [26] and use ResNet50 [3], ResNet101 [3], and V2-99 [9] as backbones. To supervise depth, LiDAR points are projected onto camera views to

Method	Backbone	Image Size	mAP↑	NDS↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
StreamPETR[26]	ResNet50	704×256	43.2	54.0	0.581	0.272	0.413	0.295	0.196
FreqPDE[24]	ResNet50	704×256	43.5	54.3	0.577	0.270	0.442	0.257	0.199
STUR3D(Ours)	ResNet50	704×256	**44.8**	**55.0**	0.558	0.275	0.431	0.257	0.208
QAF2D†[7]	ResNet50	704×256	46.0	56.1	0.573	0.263	0.387	0.276	0.191
DySS[31]†	ResNet50	704×256	46.2	56.2	0.537	0.268	0.338	0.239	0.177
GeoBEV[33]*†	ResNet50	704×256	43.0	54.6	0.533	0.275	0.419	0.298	0.214
OPEN†[5]	ResNet50	704×256	47.0	56.5	0.573	0.275	0.413	0.235	0.193
STUR3D(Ours)†	ResNet50	704×256	**48.6**	**57.9**	0.562	0.269	0.362	0.237	0.202
QAF2D[7]	V2-99	320×800	50.0	58.6	0.561	0.261	0.369	0.251	0.196
tgGBC[29]	V2-99	320×800	52.1	60.3	0.533	0.268	0.344	0.210	0.192
STUR3D(Ours)	V2-99	320×800	**53.0**	**61.2**	0.517	0.270	0.307	0.241	0.194
MV2D-T[28]†	ResNet101	1600×900	47.1	56.1	0.593	0.262	0.340	0.368	0.184
StreamPETR†[26]	ResNet101	1408×512	50.4	59.2	0.569	0.262	0.315	0.257	0.199
Sparse4Dv2[14] †§	ResNet101	1408×512	52.1	60.8	0.519	0.265	0.364	0.199	0.180
DVPE†[25]	ResNet101	1408×512	52.1	60.3	0.544	0.262	0.318	0.248	0.200
OPEN†[5]	ResNet101	1408×512	51.9	60.6	0.528	0.266	0.312	0.222	0.190
STUR3D(Ours)†	ResNet101	1408×512	**53.1**	**61.3**	0.509	0.267	0.313	0.224	0.201

Table 1. **Comparison with previous State-of-the-art multi-view 3D detectors on the nuScenes `val` set**. For fair comparisons, we reproduce the baseline method under the same settings as our method. † benefited from the perspective-view pre-training of nuImages.* indicates methods with CBGS training which will elongate 1 epoch into 4.5 epochs. § means to use a future frame.

Methods	Backbone	Image Size	mAP↑	NDS↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
StreamPETR[26]	V2-99	640×1600	55.0	63.6	0.479	0.239	0.317	0.241	0.119
QAF2D[7]	V2-99	640×1600	55.5	63.6	0.471	0.237	0.322	0.268	0.118
DVPE[25]	V2-99	640×1600	57.2	64.4	0.466	0.240	0.319	0.266	0.126
RayDN[15]	V2-99	640×1600	56.5	64.5	0.461	0.241	0.322	0.239	0.114
FreqPDE[24]	V2-99	640×1600	56.0	64.2	0.468	0.241	0.315	0.242	0.115
STUR3D(Ours)	V2-99	640×1600	**57.9**	**64.6**	0.449	0.259	0.355	0.243	0.124

Table 2. **Comparison on the nuScenes `test` set**. No other tricks(*e.g.*, CBGS, TTA) are used during training and test in our methods.

form sparse depth maps with a downscale $\times 16$, and DepthNet [23] is trained with masked regression loss on valid pixels. Depth prediction is supervised by LiDAR during training, but operates in a vision-only mode during inference. All models are trained on 4 NVIDIA GeForce RTX A6000 GPUs using AdamW [18] with a cosine annealing policy, a learning rate of 4×10^{-4}, and a batch size of 8. Our models are trained for 90 epochs on the nuScenes `val` set and 60 epochs on the `test` set. No CBGS [34], test time augmentation, or future frames are used. We retain the top-128 2D detections and top-256 3D detections per frame. In addition, we initialize the learnable decay parameter β and the gating parameters γ and τ to 0.5. For the geometric prior $\mathcal{X}$, we generate a single-scale tensor and follow the split set [32] of Eq. 10 for inference.

4.3. Comparison with State-of-the-art

We compare the proposed STUR3D with the existing state-of-the-art surrounding-view 3D object detectors on nuScenes `val` and `test` sets [1]. As shown in Tab. 1 for nuScenes `val` set, when adopting ResNet50 [3] backbone pre-trained on nuImages and 704×256 input image size, STUR3D achieves 48.6% mAP and 57.9% NDS performance, which outperforms the state-of-the-art method OPEN [5] by 1.6% mAP and 1.4% NDS, and outperforms our baseline StreamPETR [26] by 3.7% mAP and 2.9% NDS. STUR3D achieves 53.1% mAP and 61.3% NDS performance with ResNet101 [3] backbone pre-trained on nuImages and 1408×512 input image size. STUR3D outperforms the state-of-the-art method Sparse4Dv2 [14] by 1.0% mAP and 0.5% NDS, which establishes a new state-of-the-art method on the nuScenes `val` set. Our STUR3D has an obvious performance improvement over the baseline [26] by 2.7% mAP and 2.1% NDS.

As shown in Tab. 6, STUR3D achieves 52.7% mAP and 60.2% NDS using only camera inputs and without any LiDAR supervision during training. LiDAR is never used as input at inference time for any model, and the table compares only whether LiDAR signals are used as training supervision. In Tab. 2, we provide the detection results on nuScenes `test` set, and compare the proposed framework with previous state-of-the-art methods.

Exp.	STOPP	STGE	OQG	mAP↑	NDS↑
a	-	-	-	48.2	57.1
b	✓	-	-	48.9	58.1
c	-	✓	-	48.7	58.0
d	-	-	✓	50.2	59.2
e	✓	-	✓	52.1	60.5
f	-	✓	✓	51.7	60.2
g	✓	✓	✓	**53.0**	**61.2**

Table 3. **Ablation studies for each component in STUR3D on the nuScenes `val` set.** The STOPP, STGE and OQG represent the Spatio-Temporal Object Prior Propagator, Spatio-Temporal Geometry Encoder, and Object-center Query generator.

Method	STOPP	3D gen.	FPS↑	mAP↑	NDS↑
StreamPETR [26]	-	-	13.0	48.2	57.1
MV2D [28]	-	✓	6.4	47.2	56.0
QAF2D [7]	-	✓	3.9	49.8	58.8
OPEN [5]	-	-	10.3	52.1	61.2
DVPE [25]	-	✓	6.2	50.5	56.6
OPEN+STOPP	✓	-	8.7	52.4	60.7
DVPE+STOPP	✓	✓	5.6	50.8	56.9
STUR3D	✓	✓	7.8	**53.0**	**61.2**

Table 4. **Runtime and accuracy on the nuScenes `val` set.** "3D gen." indicates the 3D query generator, where STUR3D uses OQG.

#	#Frames	2D detections	3D detections	mAP↑	NDS↑
A	1	128	256	**53.0**	**61.2**
B	1	-	256	52.4	60.6
C	1	128	-	52.1	60.3
D	1	64	128	52.7	61.0
E	1	256	256	52.6	60.9
F	2	128	256	52.9	61.0
G	4	128	256	52.3	60.6

Table 5. **Ablation of detection and temporal caching frames in the STOPP.**

Method	LiDAR supervision	mAP↑	NDS↑
Linear	-	52.5	59.8
Linear	✓	52.7	60.7
MLP	-	51.7	60.1
MLP	✓	52.0	60.6
STGE	-	52.7	60.2
STGE	✓	**53.0**	**61.2**

Table 6. **Comparison of other depth encoding methods on the nuScenes `val` set.** ***LiDAR supervision*** indicates whether LiDAR signals are used as supervision during training; all models do not use LiDAR inputs during inference.

STUR3D achieves 57.9% mAP and 64.6% NDS performance with V2-99 backbone [9] and 640×1600 input image size. STUR3D outperforms our baseline StreamPETR [26] by 2.8% mAP and 1.0% NDS, and exceeds state-of-the-art method DVPE [25] by 0.6% mAP and 0.2% NDS.

4.4. Ablation Study

Effectiveness of each component. To evaluate the effectiveness of STUR3D, we conduct a series of ablation studies on the nuScenes validation set [1], using V2-99 [9] as the backbone and an input image size of 320×800. We retrain STUR3D for 24 epochs, and summarize the results in Tab. 3. We can observe progressive improvements with each introduced component. First, the proposed model only with the STOPP module (b) improves mAP by +0.7%, which demonstrates the impact of temporal prior propagation in stabilizing object representations. Incorporating STGE (c) further enhances mAP by +0.5% and NDS by +0.9%, emphasizing the importance of geometry-aware consistency across surrounding-view features. Introducing OQG (d) alone results in +2.0% mAP and +2.1% NDS gains, highlighting the importance of object-centric query generation with depth priors for stronger initialization. When combining STOPP and OQG (e), the system benefits from both temporal and geometric priors, achieving +3.9% mAP and +3.4% NDS over the baseline. Similarly, pairing STGE with OQG (f) further refines depth-guided geometry, improving the spatial grounding of 3D queries. Finally, the full STUR3D framework achieves 53.0% mAP and 61.2% NDS, surpassing the baseline by +4.8% mAP and +4.1% NDS, confirming that the integration of all proposed components leads to substantial improvements in surrounding-view 3D object detection.

Comparison of runtime and accuracy. Tab. 4 shows runtime and accuracy on nuScenes `val` set. We also mark whether a STOPP module is used and whether a 3D query generator is present, and STUR3D adopts OQG. STUR3D attains 53.0% mAP and 61.2% NDS at 7.8 FPS. Relative to MV2D [28], which runs at 6.4 FPS and yields 47.2% mAP and 56.0% NDS, and DVPE [25], which runs at 6.2 FPS and yields 50.5% mAP and 56.6% NDS, STUR3D offers comparable throughput while delivering sizable gains, namely +5.8% mAP and +5.2% NDS over MV2D and +2.5% mAP and +4.6% NDS over DVPE. Compared with QAF2D [7], which runs at 3.9 FPS, STUR3D is about 2× faster and more accurate. Although OPEN [5] achieves the highest throughput at 10.3 FPS, STUR3D provides the best mAP and matches OPEN on NDS at 61.2%, showing a favorable balance between accuracy and efficiency.

Ablations of the STOPP. The STOPP module demonstrates strong plug-and-play properties. As shown in Tab. 4, after integrating it into DVPE [25], the speed is 5.6 FPS, and mAP and NDS increase by +0.3% and +0.3%, respectively. After integrating it into OPEN [5], the total speed is 8.7 FPS, the mAP is 52.4%, and the NDS is 60.7%. These results indicate that the spatio-temporal prior prop-

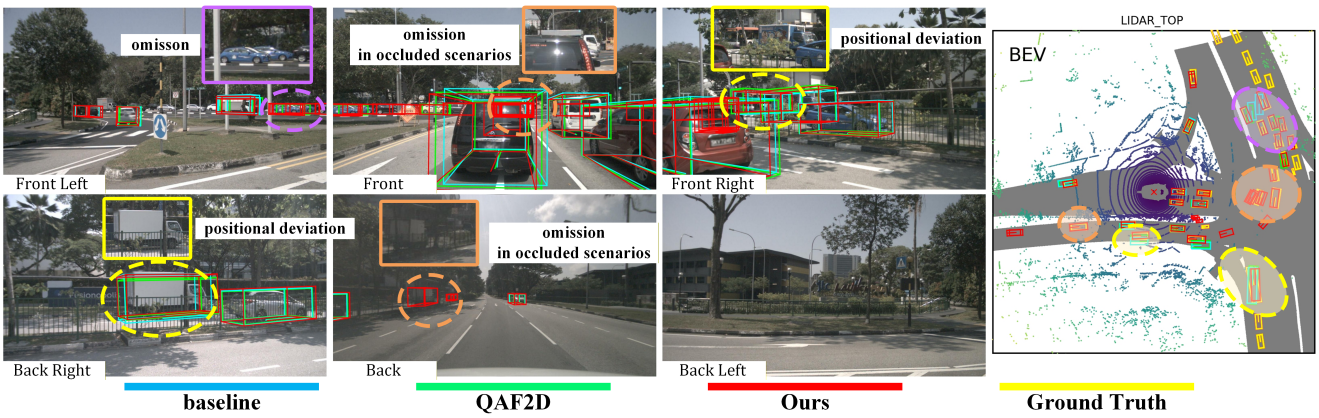


Figure 5. **Qualitative detection results on surrounding-view images and the BEV space on the nuScenes `val` set.** The 3D predicted bounding boxes are shown with different colors for each method on surrounding-view images. Blue represents the baseline, green for QAF2D, and red for our predictions. The yellow boxes on the BEV map represent the ground truth.

agation mechanism of STUR3D is a transferable module, and the overall design significantly improves detection accuracy and temporal consistency while ensuring inference efficiency.

As shown in Tab. 5, we analyze the influence of different detection inputs and the number of cached historical feature maps in the STOPP. Removing 2D or 3D detections (**B–C**) leads to noticeable drops in both mAP and NDS, indicating that the joint use of 2D and 3D priors is essential for accurate propagation. Increasing cached frames from one to two (**F**) maintains comparable accuracy, while extending to four (**G**) brings no further gain due to accumulated temporal noise.

Ablations on depth encoding methods. As shown in Tab. 6, the proposed STGE consistently outperforms both Linear and MLP encoders across all metrics, indicating stronger geometric feature modeling. With LiDAR supervision, STGE yields a notable gain in overall detection quality, NDS increases by +1.0%, reflecting improved 3D localization accuracy.

4.5. Visualization Results

From the visualization results on the nuScenes `val` set as shown in Fig. 5, we observe that STUR3D achieves clear improvements over both the baseline [26] and QAF2D [7] across various challenging scenarios. In the front-right and back-right images, STUR3D achieves significantly higher localization precision, with predicted bounding boxes more closely aligned with the ground truth in both the image and BEV spaces. In image front-left image, STUR3D demonstrates stronger detection capability, successfully identifying objects that are entirely missed by the baseline. In images front and back, STUR3D effectively recovers objects that are heavily occluded or partially hidden, where both the baseline and QAF2D fail. This improvement stems from the spatio-temporal propagation structure, which leverages information from previous frames to maintain temporal consistency and restore occluded targets. These results validate that STUR3D not only enhances detection completeness and localization accuracy but also exhibits superior robustness and stability in occluded environments.

5. Conclusion

In this paper, we introduced STUR3D, a spatio-temporal unified representation learning framework for camera-based 3D object detection. By propagating aligned spatio-temporal historical 3D and 2D detections to the image plane via STOPP, injecting geometry- and depth-aware cues with STGE, and lifting refined 2D detections into object-center 3D queries through OQG, STUR3D explicitly aligns heterogeneous 2D and 3D spatio-temporal representations, guiding the 2D detector to distill representations essential for 3D detection. Extensive experiments on the nuScenes benchmark demonstrate that STUR3D substantially outperforms strong query-based baselines across diverse backbones and input resolutions, achieving 57.9% mAP and 64.6% NDS on the test set while preserving a favorable accuracy-efficiency tradeoff and robustness to occlusion. Ablation studies further verify the contribution of each component, the plug-and-play property of STOPP, and the effectiveness of our depth-aware geometric modeling under both LiDAR-supervised and vision-only depth settings. Future work will focus on enhancing the efficiency and scalability of geometric attention mechanisms.

6. Acknowledgments

This work is supported by the National Natural Science Foundation of China (62273339, U24A20282).

References

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019. 5, 6, 7

[2] Chunrui Han, Jian-Yuan Sun, Zheng Ge, Jinrong Yang, Runpei Dong, Hongyu Zhou, Weixin Mao, Yuang Peng, and Xiangyu Zhang. Exploring recurrent long-term temporal fusion for multi-view 3d perception. *IEEE Robotics and Automation Letters*, 9:6544–6551, 2023. 2

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 3, 5, 6

[4] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross B. Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017. 4, 5

[5] Jinghua Hou, Tong Wang, Xiaoqing Ye, Zhe Liu, Xiao Tan, Errui Ding, Jingdong Wang, and Xiang Bai. Open: Object-wise position embedding for multi-view 3d object detection. In *ECCV*, 2024. 2, 6, 7

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

[7] Haoxuanye Ji, Pengpeng Liang, and Erkang Cheng. Enhancing 3d object detection with 2d detection-guided query anchors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21178–21187, 2024. 1, 2, 3, 6, 7, 8

[8] Xiaohui Jiang, Shuailin Li, Yingfei Liu, Shihao Wang, Fan Jia, Tiancai Wang, Lijin Han, and Xiangyu Zhang. Far3d: Expanding the horizon for surround-view 3d object detection. 2023. 1, 2

[9] Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, and Jongyoul Park. An energy and gpu-computation efficient backbone network for real-time object detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 752–760, 2019. 3, 5, 7

[10] Peidong Li, Wancheng Shen, Qihao Huang, and Dixiao Cui. Dualbev: Cnn is all you need in view transformation. *arXiv preprint arXiv:2403.05402*, 2024. 2

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

[12] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022. 2

[13] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017. 3

[14] Xuewu Lin, Tianwei Lin, Zixiang Pei, Lichao Huang, and Zhizhong Su. Sparse4d v2: Recurrent temporal fusion with sparse model, 2023. 6

[15] Feng Liu, Tengteng Huang, Qianjing Zhang, Haotian Yao, Chi Zhang, Fang Wan, Qixiang Ye, and Yanzhao Zhou. Ray denoising: Depth-aware hard negative sampling for multi-view 3d object detection. *arXiv preprint arXiv:2402.03634*, 2024. 6

[16] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. *arXiv preprint arXiv:2203.05625*, 2022. 1, 2

[17] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Qi Gao, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petrv2: A unified framework for 3d perception from multi-camera images. *arXiv preprint arXiv:2206.01256*, 2022. 1, 2

[18] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 6

[19] Zhipeng Luo, Changqing Zhou, Gongjie Zhang, and Shijian Lu. Detr4d: Direct multi-view 3d object detection with sparse attention. *arXiv preprint arXiv:2212.07849*, 2022. 2

[20] Jinhyung Park, Chenfeng Xu, Shijia Yang, Kurt Keutzer, Kris Kitani, Masayoshi Tomizuka, and Wei Zhan. Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. 2023. 2

[21] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Proceedings of the European Conference on Computer Vision*, 2020. 2

[22] Yining Shi, Jingyan Shen, Yifan Sun, Yunlong Wang, Jiaxin Li, Shiqi Sun, Kun Jiang, and Diange Yang. Srcn3d: Sparse r-cnn 3d surround-view camera object detection and tracking for autonomous driving. arXiv, 2022. 2

[23] Changyong Shu, Jiajun Deng, Fisher Yu, and Yifan Liu. 3dppe: 3d point positional encoding for transformer-based multi-camera 3d object detection. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3557–3566, 2023. 2, 4, 6

[24] Haisheng Su, Junjie Zhang, Feixiang Song, Sanping Zhou, Wei Wu, Nanning Zheng, and Junchi Yan. Freqpde: Rethinking positional depth embedding for multi-view 3d object detection transformers, 2025. 6

[25] Jiasen Wang, Zhenglin Li, Ke Sun, Xianyuan Liu, and Yang Zhou. Dvpe: Divided view position embedding for multi-view 3d object detection. *arXiv preprint arXiv:2407.16955*, 2024. 1, 2, 3, 6, 7

[26] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3598–3608, 2023. 1, 2, 5, 6, 7, 8

[27] Yue Wang, Vitor Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, , and Justin M. Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *The Conference on Robot Learning (CoRL)*, 2021. 1, 2

[28] Zitian Wang, Zehao Huang, Jiahui Fu, Naiyan Wang, and Si Liu. Object as query: Lifting any 2d object detector to 3d detection. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3768–3777, 2023. 1, 2, 6, 7

[29] Lizhen Xu, Xiuxiu Bai, Xiaojun Jia, Jianwu Fang, and Shanmin Pang. Accelerate 3d object detection models via zero-shot attention key pruning, 2025. 2, 6

[30] Ziteng Xue, Mingzhe Guo, Heng Fan, Shihui Zhang, and Zhipeng Zhang. Corrbev: Multi-view 3d object detection by correlation learning with multi-modal prototypes. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 27413–27423, 2025. 2

[31] Rajeev Yasarla, Shizhong Han, Hong Cai, and Fatih Porikli. Dyss: Dynamic queries and state-space learning for efficient 3d object detection from multi-camera videos. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2501–2510, 2025. 6

[32] Bo-Wen Yin, Jiao-Long Cao, Ming-Ming Cheng, and Qibin Hou. Dformerv2: Geometry self-attention for rgbd semantic segmentation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19345–19355, 2025. 4, 6

[33] Jinqing Zhang, Yanan Zhang, Yunlong Qi, Zehua Fu, Qingjie Liu, and Yunhong Wang. Geobev: Learning geometric bev representation for multi-view 3d object detection. *arXiv preprint arXiv:2409.01816*, 2024. 2, 6

[34] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection, 2019. 6

[35] Zhuofan Zong, Dongzhi Jiang, Guanglu Song, Zeyue Xue, Jingyong Su, Hongsheng Li, and Yu Liu. Temporal enhanced training of multi-view 3d object detector via historical object prediction. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3758–3767, 2023. 2