Noname manuscript No.
(will be inserted by the editor)

GraphBEV++: Multi-Modal Feature Alignment for Autonomous Driving

Ziying Song[1] · **Hongyu Pan**[2] · **Lin Liu**[1] · **Shaoqing Xu**[3] · **Lei Yang**[4] · **Caiyan Jia**[1*] · **Yadan Luo**[5]

Received: date / Accepted: date

Abstract Feature misalignment in BEV perception is a critical yet often overlooked challenge in autonomous driving, especially under calibration uncertainties between LiDAR and camera sensors. To address this issue, we propose a robust multi-modal fusion framework, **GraphBEV++**, which systematically mitigates projection-induced misalignment. The framework consists of two key modules: LocalAlign-v2 and GlobalAlign-v2. LocalAlign-v2 introduces neighborhood-aware depth features via graph matching to correct local misalignment. It supports both LSS-based and query-based BEV representations, making it compatible with BEVFusion and BEVFormer architectures for consistent cross-paradigm alignment. GlobalAlign-v2 encompasses two variants: Deformable and Diffusion. The Deformable variant addresses global misalignment in LSS-based multi-modal BEV by explicitly learning cross-modal feature offsets. In contrast, the Diffusion variant targets implicit misalignment in query-based BEV by injecting noise to simulate misalignment and employing a denoising process to recover aligned features. Experimental results show that GraphBEV++ achieves state-of-the-art performance under misalignment noise on nuScenes and Waymo subset, improves long-range detection on Argoverse2, and generalizes effectively to the 3D occupancy prediction task, consistently improving occupancy estimation accuracy and robustness under both clean and noisy settings. Furthermore, GraphBEV++ effectively alleviates misalignment issues in end-to-end autonomous driving. Compared with five baselines (UniAD, VAD, FusionAD, MomAD, and WoTE), it demonstrates superior performance in both open-loop (nuScenes) and closed-loop (Bench2Drive and NAVSIM) evaluations across perception, prediction, and planning tasks.

Keywords Autonomous Driving · Multi-Modal Fusion · Feature Alignment · Bird's-Eye View.

1 Introduction

In autonomous driving systems, multi-modal fusion plays a critical role. Different sensors provide complementary perceptual capabilities: cameras offer rich semantic and texture information that is valuable for recognizing fine-grained features such as traffic signs, lane markings, and pedestrians, while LiDAR provides accurate 3D geometric information that enables reliable estimation of object distances and spatial structures. Relying on a single modality often leads to perception blind spots or degraded performance—for example, cameras may fail under low-light or adverse weather conditions, whereas LiDAR suffers from sparsity and lacks semantic richness. Therefore, effectively fusing multi-modal data to leverage the strengths of each modality is essential for enhancing the robustness, accuracy, and safety of perception systems, and serves as a foundational technology for achieving high-level autonomous driving [151].

Multi-modal fusion has evolved from early point-level [129,147,50,106,148,177] and feature-level [139,19,2,18,135,131] methods to the currently prevalent methods of Bird's-Eye View (BEV) fusion like BEVFusion [94,108]. Although current BEV methods are

*Corresponding Author: Caiyan Jia
[1]Beijing Key Laboratory of Traffic Data Mining and Embodied Intelligence, School of Computer Science and Technology, Beijing Jiaotong University Email: {songziying, cyjia}@bjtu.edu.cn
[2]Horizon Robotics
[3]University of Macau, [4]Tsinghua University
[5]The University of Queensland

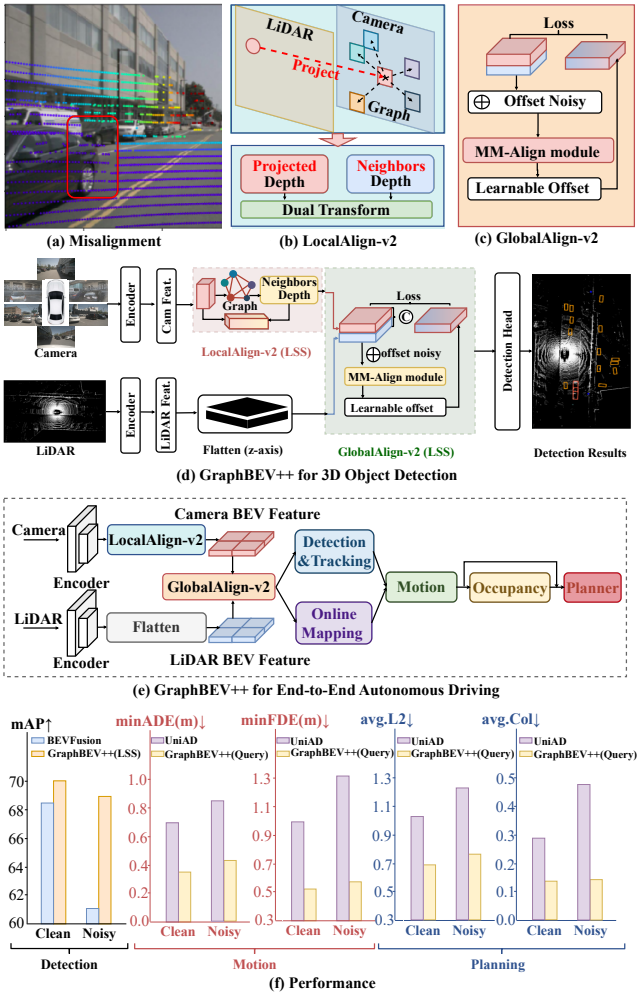

Fig. 1 (a) Feature misalignment often arises from projection matrix errors between LiDAR and camera, leading to inaccurate depth and distorted spatial relationships when projecting LiDAR points onto the image. **(b) LocalAlign-v2** addresses local feature misalignment by encoding the projected neighborhood depth features through learnable representations. **(c)** GlobalAlign-v2 mitigates global BEV misalignment via learned offsets (Deformable) or diffusion-based feature alignment.**(d)** The GraphBEV++ framework for 3D object detection effectively resolves BEV feature misalignment in both types of BEV representations: LSS-based (as in BEVFusion) and Query-based (as in BEVFormer). **(e)** GraphBEV++ systematically investigates the impact of feature misalignment on end-to-end autonomous driving and effectively mitigates it through principled BEV alignment mechanisms. **(f)** Empirical results demonstrate that GraphBEV++ effectively mitigates the impact of feature misalignment in both 3D object detection and end-to-end autonomous driving tasks.

effective on clean datasets like nuScenes [6], their performance deteriorate on misaligned data. This performance decline is primarily due to calibration errors between LiDAR and camera, exacerbated by factors like road vibrations [179]. Feature misalignment presents a major challenge in practical multi-modal fusion

Table 1 Correspondence Between LocalAlign-v2 / GlobalAlign-v2, Tasks, and Module Variants.

Version	Tasks	Modules
GraphBEV++ (LSS)	3D Object Detection BEV Segmentation	LocalAlign-v2 (LSS) GlobalAlign-v2 (Deformable)
GraphBEV++ (Query)	3D Object Detection End-to-End Autonomous Driving	LocalAlign-v2 (Query) GlobalAlign-v2 (Diffusion)

tasks, especially when integrating LiDAR and camera data [179,28], as illustrated in Figure 1. Such misalignment [179,28,133,35,34,127,3,30,119,37] arises from a range of factors, including: (1) inaccuracies in the extrinsic calibration matrix between LiDAR and camera sensors; (2) temporal desynchronization between asynchronous sensing modalities, such as spinning LiDAR and event-based cameras; (3) calibration noise induced by mechanical vibrations or sensor mounting instability; (4) discrepancies in the fields of view across different sensors; and (5) modality-specific sensitivity to external factors such as lighting and adverse weather conditions. These challenges collectively hinder reliable feature fusion and substantially degrade downstream perception performance. For instance, adverse weather such as rain or snow can lead to signal scattering and reflection in LiDAR, resulting in sparse or noisy point clouds. This, in turn, increases the depth estimation error—especially for distant or boundary objects—which severely affects the LiDAR-to-Camera projection quality.

Most feature-level multi-modal methods [86,19,1,171,189] employ the Cross Attention operation to query features of a specific modality, circumventing the need for projection matrices. A few feature-level multi-modal methods [178,135,131,18,81,183,80,71] have sought to mitigate these errors through the use of projection offsets or neighboring projections. A few BEV-based methods, such as ObjectFusion [8], eliminate the camera-to-BEV transformation during fusion to align object-centric features across different modalities. MetaBEV [39] utilizes the Cross Deformable Attention for feature misalignment, but overlooks depth estimation errors in view transformation and aligns features only during LiDAR and camera BEV fusion.

Currently, mainstream BEV representations can be broadly categorized into two types: Lift-Splat-Shoot (LSS)-based methods and Query-based methods [133]. BEVFusion [108,94] and BEVFormer [91] respectively exemplify these two prominent paradigms. Both approaches face distinct forms of **feature misalignment** issues when fusing camera and LiDAR data into the BEV space to enhance detection performance. BEVFusion [108,94] unites camera and LiDAR data in BEV space to enhance detection, but overlooks **feature mis-**

alignment in real-world applications. It is primarily evident in two aspects. 1) BEVFusion [108] transforms multi-image features into a unified BEV representation using BEVDepth's [84] explicit depth supervision from LiDAR-to-camera. Although this LiDAR-to-camera strategy offers more reliable depth than LSS [120], it overlooks the misalignment between LiDAR and camera in real-world scenarios, leading to **local misalignment**. 2) In the LiDAR-camera BEV fusion, the misalignment of BEV features due to depth inaccuracies is overlooked by directly concatenating representations and applying basic convolution, as described in BEVFusion [108], resulting in **global misalignment**. Moreover, query-based BEV representation methods such as BEVFormer [91] also suffer from feature misalignment issues. When projecting 3D query points onto the image plane, these methods are highly sensitive to extrinsic calibration errors and depth estimation inaccuracies. Such projection deviations result in discrepancies between the sampled image features and the actual object locations, leading to local misalignment and degraded BEV feature representations.

It is important to note that the local and global misalignment discussed in this paper correspond to two different but complementary error granularities in BEV perception. Specifically, **local misalignment** refers to feature-level correspondence errors caused by inaccurate geometric projection, including depth estimation errors, LiDAR-camera calibration noise, and query projection deviations. These errors mainly affect local neighborhood consistency during image-to-BEV transformation and feature sampling. In contrast, **global misalignment** refers to representation-level inconsistencies after BEV construction, where camera and LiDAR BEV features exhibit spatial shifts, structural distortions, or semantic discrepancies due to the accumulation of local projection errors. In practice, both types of misalignment often coexist and form a hierarchical error propagation process: local projection errors introduced during BEV construction may gradually accumulate and eventually lead to global inconsistencies in fused BEV representations. Therefore, the two alignment modules in GraphBEV++ are designed to operate at different stages and address different error granularities. LocalAlign-v2 focuses on improving local geometric correspondence during BEV feature generation, while GlobalAlign-v2 further aligns heterogeneous BEV representations at the fusion stage. As a result, the two modules are complementary rather than conflicting, jointly providing robust feature alignment under real-world misalignment scenarios.

To address the aforementioned feature misalignment challenges, we propose a robust fusion framework, GraphBEV++, which enables reliable 3D object detection and end-to-end autonomous driving, especially under misalignment scenarios. Specifically, to handle local misalignment in the camera-to-BEV transformation, we introduce LocalAlign-v2 module with two variants: LocalAlign-v2 (LSS) and LocalAlign-v2 (Query). LocalAlign-v2 (LSS) operates within the view transformation step of the BEVFusion camera branch, where it leverages LiDAR-to-camera explicit depth supervision and utilizes a graph neural network to extract neighborhood-aware depth features. In contrast, LocalAlign-v2 (Query) is built upon BEVFormer [91] and addresses projection errors of reference points onto the image plane by aligning 2D-3D correspondences during feature sampling. To further address global misalignment during BEV-level fusion of LiDAR and camera features, we introduce GlobalAlign-v2 module. This module encodes both the projected LiDAR-to-camera depth and the neighborhood depth through dual-stream encoding to generate a reliable depth representation. It then dynamically estimates spatial offsets to align heterogeneous BEV features from the two modalities. Overall, the corresponding baselines, tasks, and datasets for LocalAlign-v2 and GlobalAlign-v2 are summarized in Table 1.

We evaluate GraphBEV++ on the nuScenes [6], Waymo subset [141], and Argoverse 2 [159] benchmark for 3D object detection. Experimental results show that GraphBEV++ achieves state-of-the-art performance in clean settings and improves mAP by 8.3% over BEVFusion under the misaligned noise setting proposed by Dong et al. [28]. In addition to perception tasks, we also investigate feature misalignment in end-to-end autonomous driving, where projection errors not only compromise perception quality but also jeopardize the stability and safety of the entire driving system. We conduct comprehensive evaluations on the nuScenes dataset under extrinsic perturbation settings, using UniAD [49], FusionAD [175], VAD [62], MomAD [130], and WoTE [85] as baselines. Furthermore, we perform closed-loop evaluations on Bench2Drive [60] and NAVSIM [25], demonstrating the effectiveness and generalizability of our multi-modal fusion strategy in both perception and planning pipelines. We believe that addressing misalignment issues is crucial for mitigating the detrimental impact on multi-modal end-to-end autonomous driving systems.

In summary, the main contributions of this study are as follows.

1. We propose a robust multi-modal fusion framework, named **GraphBEV++**, to address feature misalignment arising from projection errors between different sensors. Our framework extends beyond 3D

object detection—its original focus—towards more comprehensive autonomous driving tasks, including end-to-end prediction and planning.
2. By thoroughly analyzing the root causes of feature misalignment, we design **LocalAlign-v2** to mitigate local misalignment caused by imprecise depth estimation, and **GlobalAlign-v2** to correct global misalignment between different BEV features.
3. Extensive experiments validate the effectiveness of **GraphBEV++**, demonstrating competitive performance on nuScenes, Waymo subset and Argoverse2 dataset, and at the close-loop settings for the datasets Bench2Drive and NAVSIM. Notably, GraphBEV++ effectively mitigates feature misalignment on the nuScenes dataset for both 3D object detection and end-to-end autonomous driving tasks, improving long-range detection performance on Argoverse2.

This work's preliminary version of GraphBEV [137] was presented at ECCV 2024. Compared to our previous conference version [137], this paper introduces the following substantial improvements in both methodology and task scope.

1. **Module-wise Upgrades.** We extend the original GraphBEV by proposing four alignment variants: LocalAlign-v2 (LSS), LocalAlign-v2 (Query), GlobalAlign-v2 (Deformable), and GlobalAlign-v2 (Diffusion). Specifically, LocalAlign-v2 generalizes the neighbor-based alignment strategy to support both LSS- and query-based BEV paradigms, thereby addressing query-feature misalignment and enabling flexible neighbor retrieval across architectures like BEVDepth and BEVFormer. GlobalAlign-v2 incorporates deformable attention and a diffusion-based denoising mechanism, which simulates BEV noise and refines feature alignment through reverse diffusion, demonstrating improved robustness.
2. **Task-wise Extension.** Beyond 3D object detection and segmentation, GraphBEV++ is extended to support a full-stack end-to-end autonomous driving pipeline, covering perception, mapping, prediction, and planning. We introduce modality-specific alignment strategies for LiDAR-camera and radar-camera pairs, accounting for their inherent disparities in resolution, semantics, and spatial coverage. This design emphasizes the critical role of alignment in downstream driving decisions.
3. **Comprehensive Evaluation.** Compared with the six tasks reported in the original GraphBEV, GraphBEV++ includes 19 benchmarks across diverse tasks: 3D object detection, multi-object tracking, online mapping, occupancy prediction, motion forecasting, and planning trajectory prediction. We introduce radar-camera fusion, expand evaluations to Waymo and Argoverse datasets, and investigate misalignment effects in BEVFormer. For end-to-end driving, both open-loop (nuScenes) and closed-loop scenarios (NAVSIM, Bench2Drive) are covered. These extensive evaluations validate the robustness and generalizability of GraphBEV++, while offering valuable resources to the research community for benchmarking multi-modal alignment and planning systems.

2 Related Work

2.1 LiDAR-based 3D Object Detection

LiDAR-based 3D object detection methods can be categorized into three primary types based on point cloud representation: Point-based, Voxel-based, and PV-based (Point-Voxel). Point-based methods [90,121, 122,123,126] extend PointNet's [122,123] principle, directly processing raw point clouds with stacked Multi-Layer Perceptrons (MLPs) to extract point features. VoxelNet [194] is innovated by partitioning raw point clouds into uniform voxel grids. Voxel-based methods [194,26,167,16,150] typically convert point clouds into voxels and apply 3D sparse convolutions for voxel feature extraction. In addition, PointPillars [74] converts irregular raw point clouds into pillars and encodes them on a 2D backbone, achieving a very high FPS. Some Voxel-based methods [42,112,31] further exploit Transformers [146] post-voxelization to capture long-range voxel relationships. PV-based methods [136,109, 113,125,174] combine voxel and point-based strategies and extract features from point clouds' diverse representations using both approaches, achieving higher accuracy albeit with increased computational demand.

2.2 Camera-based 3D Object Detection

Camera-based 3D object detection methods have gained increasing attention in academia and industry, mainly due to the significantly lower cost of camera sensors compared to LiDAR [133]. Early methods [5,164,128] have focused on augmenting 2D object detectors with additional 3D bounding box regression heads. Current camera-based methods have rapidly evolved since LSS [120] introduced the concept of unifying multi-view information onto a BEV through "Lift and splat". LSS-based methods [120,84,

82,117,170,169] like BEVDepth [84] extract 2D features from multi-view images and provide effective depth supervision via LiDAR-to-camera projections before unifying multi-view features onto the BEV. Subsequent works [82,117] have introduced multi-view stereo techniques to improve depth estimation accuracy and achieve SOTA (state-of-the-art) performance. Additionally, inspired by the success of transformer-based architectures such as DETR [10] and Deformable DETR [196] in 2D detection, transformer-based detectors have emerged for 3D object detection. Following DETR3D [155], some methods design a set of object queries [64,103,104] or BEV grid queries [91,168], then perform view transformation through cross-attention between queries and image features.

2.3 Multi-modal 3D Object Detection

Multi-modal 3D object detection refers to using data features from different sensors and integrating these features to achieve complementarity, thus enabling the detection of 3D objects. Previous multi-modal methods can be coarsely classified into three types, i.e., point-level, feature-level, and BEV-based methods. Point-level methods [129,147,50,106,148,177] and feature-level methods [139,19,2,18,135,131,187] typically leverage image features to augment LiDAR points or 3D object proposals. BEV-based methods [108,94,39,8,132] efficiently unify the representations of LiDAR and camera into BEV space. Although BEVFusion [108,94] achieve high performance, they are typically tested on clean datasets like nuScenes [6], overlooking real-world complexities, especially **feature misalignment**, which hampers their applications.

2.4 3D Occupancy Prediction

3D occupancy prediction has emerged as a promising paradigm for holistic scene understanding in autonomous driving, as it provides a unified representation of geometry and semantics in 3D space [9,191,52,190]. Existing methods have explored diverse scene representations for occupancy estimation, ranging from dense voxel grids [9,87,190] to more efficient structures such as Tri-Perspective View (TPV) [52], 3D Gaussians [53], point-based representations [199], and fully sparse occupancy frameworks [100]. In parallel, large-scale benchmarks and comprehensive occupancy frameworks have been developed to advance occupancy perception in real-world autonomous driving scenarios [145,154,38]. *Gan et al.[38] present a comprehensive framework for 3D occupancy estimation, which reveals several key components for 3D occupancy estimation, such as network design, optimization, and evaluation. SparseOcc [100] introduces a fully sparse occupancy framework for efficient 3D scene understanding. These advances have significantly improved the accuracy and efficiency of 3D scene representation and understanding.* Despite these successes, feature misalignment remains an underexplored challenge in occupancy prediction. Since occupancy estimation relies on multi-view feature projection and aggregation, inaccurate geometric alignment may directly affect occupancy quality. Although GraphBEV++ is designed for multi-modal 3D object detection, the proposed LocalAlign-v2 and GlobalAlign-v2 modules provide a general alignment mechanism that could potentially benefit occupancy prediction tasks. Exploring robust feature alignment for occupancy estimation is an interesting direction for future research.

2.5 End-to-end Autonomous Driving

Recent works in autonomous driving have shifted towards exploring end-to-end tasks [12]. These works [49,175,142,62,14,58,173] are now designed to execute integrated tasks encompassing perception, prediction, and planning, spanning the entire process from scene perception to ego-planning. The advantage of end-to-end autonomous driving lies in its ability to provide interpretable intermediate results and avoids local optima by considering the system holistically, leading to significant breakthroughs in planning tasks [12]. However, most studies rely on a single modality (especially cameras) and ignore the impact of extrinsic calibration changes causing feature misalignment, which can disrupt end-to-end tasks. To address this, we propose the multi-modal end-to-end framework GraphBEV++ that improves feature alignment through graph matching, enhancing its robustness in real-world scenarios and offering significant potential for the future deployment of end-to-end autonomous driving systems.

3 Method

To address the challenge of feature misalignment in autonomous driving tasks [108,94,49], we propose a robust multi-modal fusion framework, GraphBEV++, designed for both 3D object detection and end-to-end autonomous driving. GraphBEV++ is highly extensible and supports both Lift-Splat-Shoot (LSS)-based and query-based BEV representations, effectively mitigating feature misalignment across modalities. The

overall architecture for end-to-end autonomous driving is illustrated in Figure 2. GraphBEV++ takes inputs from multiple sensors, including LiDAR and cameras, and extracts modality-specific features using dedicated encoders. We design the LocalAlign-v2 module to transform image features into the BEV space, alleviating local misalignment caused by projection errors between LiDAR and camera in conventional BEV fusion frameworks [108,94]. Additionally, we introduce the GlobalAlign-v2 module to further correct global misalignment between LiDAR and camera BEV features during the fusion process.

3.1 LocalAlign-v2

LocalAlign-v2 is designed to address local misalignment caused by projection discrepancies between heterogeneous sensors. It is compatible with various BEV representations, including both LSS-based and query-based paradigms. In the following, we present the details of **LocalAlign-v2 (LSS)** and **LocalAlign-v2 (Query)**, as shown in Figure 3 and Figure 4, respectively.

3.1.1 LocalAlign-v2 (LSS)

To facilitate the transformation of camera features into BEV features, the LSS-based methods like BEVFusion [108,94] leverage LiDAR-to-camera to provide projected depth, thereby enabling the fusion of depth and image features. In the process of camera-to-BEV, the methods like BEVFusion [108,94] operate under the assumption that the depth information provided by LiDAR-to-camera projection is accurate and reliable. However, they overlook the complexities inherent in real-world scenarios, where most of projection matrices between LiDAR and cameras are calibrated manually. Such calibration inevitably introduces projection errors, leading to **depth misalignment**—where the depths of surrounding neighbors are projected as the pixel's depth. This depth misalignment results in inaccuracies within the depth features, causing **local misalignment** during multi-view transformation into BEV representations. It underscores the challenges of ensuring precise depth estimation within BEVFusion [108] and highlights the importance of robust methods to address projection errors.

Therefore, we propose a LocalAlign-v2 (LSS) module to address local misalignment, with its pipeline depicted in Figure 3. Specifically, LiDAR-to-camera provides projected depth, defined as $D_S \in \mathbb{R}^{B_S \times N_C \times 1 \times H \times W}$, where B_S represents the batch size, N_C denotes the number of multi-views (six in the case of nuScenes), and H and W are the height and the width

Algorithm 1: Graph for Finding Neighbors

Input:
The indices of the projected pixels $M_{\text{Coords}} \in \mathbb{R}^{N_P \times 2}$.
Hyper-parameter: Number of neighbors $K_{\text{graph}} = 8$.
Output: Neighbors $M_{K_{\text{Coords}}} \in \mathbb{R}^{N_P \times K_{\text{graph}} \times 2}$.

```
while LocalAlign-v2 do
    Function KD-Tree (M_Coords, M_Coords_i, K_graph)
        Compute the Euclidean distance between
          M_Coords_i and M_Coords
        Indices = argsort(distances)
        return M_Coords[1 : K_graph]
    for i = 1 ... N_P do
        Neighbors = KD-Tree(M_Coords, M_Coords_i,
          K_graph)
        M_Coords_i = Neighbors
    end
end
```

of images, respectively. The projection from LiDAR-to-camera maps 3D point clouds onto an image plane, from which we can obtain the indices of the projected pixels, defined as $M_{\text{Coords}} \in \mathbb{R}^{N_P \times 2}$, where N_P refers to the number of points projected onto pixels, and 2 represents the pixel coordinates (u, v) as illustrated below.

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = h\mathcal{K} \begin{bmatrix} R\ T \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix}, \tag{1}$$

where P_x, P_y, P_z denote the LiDAR point's 3D location, (u, v) denotes the corresponding 2D location, and z_c represents the depth of its projection on the image plane, $\mathcal{K}$ denotes the camera intrinsic parameter, R and T denote the rotation and the translation of the LiDAR with respect to the camera reference system, and h denotes the scale factor due to down-sampling.

We employ the KD-Tree algorithm to obtain the indices of the projected pixels' neighbors, defined as $M_{K_{\text{Coords}}} \in \mathbb{R}^{N_P \times K_{\text{graph}} \times 2}$, where K_{graph} denotes the number of neighbors for each projected pixel. The process is outlined in Algorithm (1). It is worth noting that we simplify the process of the KD-Tree algorithm, and the code can be referred to scipy[1]. Then, we obtain the surrounding neighbor depth $D_K \in \mathbb{R}^{B_S \times N_C \times K_{\text{graph}} \times H \times W}$ by indexing D_S with M_{Coords}. Then, D_S and D_K simultaneously enter the Dual Transform module for deep feature encoding. The shapes of D_S and D_K are respectively modified to $[B_S \times N_C, 1, H, W]$ and $[B_S \times N_C, K_{\text{graph}}, H, W]$ before being fed into the Dual Transform module. This module comprises straightforward components, including con-

[1] https://github.com/minrk/scipy-1/blob/master/scipy/spatial/ckdtree.c

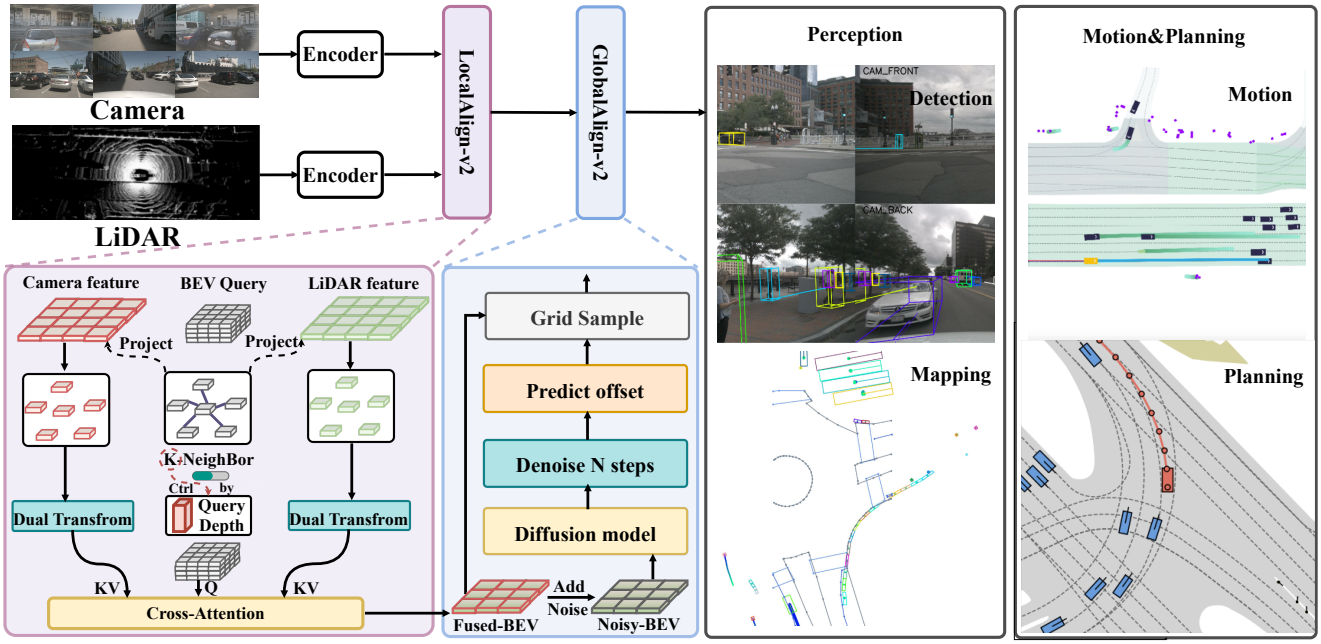

Fig. 2 The overview of **GraphBEV++** within an end-to-end autonomous driving framework. We primarily demonstrate our method based on the multi-modal end-to-end baseline FusionAD, aiming to mitigate the impact of BEV feature misalignment on downstream autonomous driving tasks.

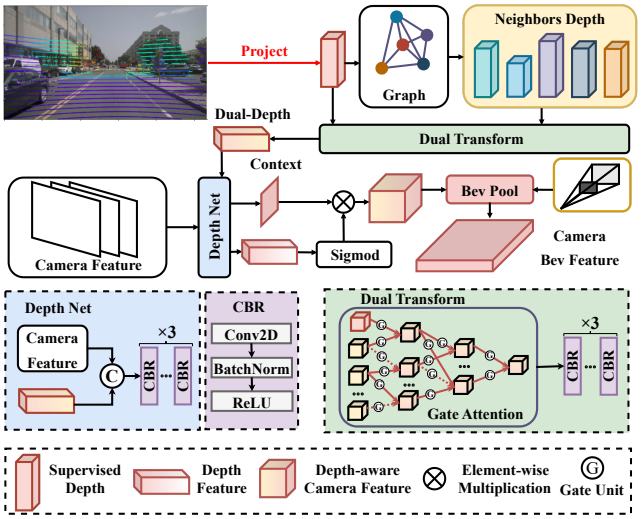

Fig. 3 Overview of the **LocalAlign-v2 (LSS)** pipeline. The LocalAlign-v2 (LSS) module mitigates local misalignment in LSS-based BEV representations by enhancing the camera-to-BEV transformation. It incorporates neighboring depth features obtained via KD-Tree–based nearest-neighbor search to refine LiDAR-to-camera projections.

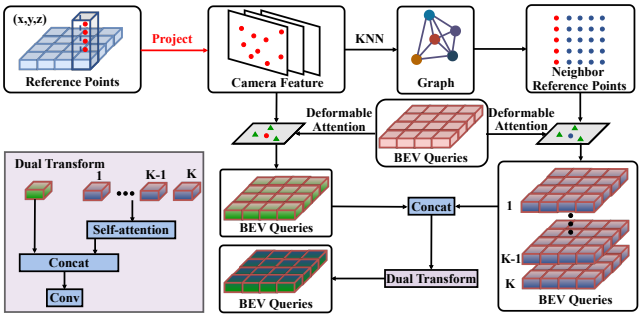

Fig. 4 Overview of the **LocalAlign-v2 (Query)** pipeline. The LocalAlign-v2 (Query) module addresses projection-induced local misalignment in query-based BEV representations. It refines the image-to-BEV transformation by leveraging adjacent BEV query features, with neighborhood relations established via KD-Tree.

volutional layers, Batch Normalization, and ReLU activations, as illustrated in Figure 3. The outcome of this process is the Dual Depth feature, denoted as D_{SK}, and its shape is $[B_S \times N_C, C_{\mathrm{SK}}, \frac{H}{8}, \frac{W}{8}]$. The camera Encoder outputs multi-scale image features from the FPN, including $F_{\mathrm{Cam.}} \in \mathbb{R}^{(B_S \times N_C) \times C_{\mathrm{Cam}} \times \frac{H}{8} \times \frac{W}{8}}$ for richer semantic information and another at a reduced resolution of $\frac{H}{16}, \frac{W}{16}$. We opt to utilize the feature with the resolution $\frac{H}{8}, \frac{W}{8}$ due to its more comprehensive semantic content.

We then design DepthNet model as illustrated in the right upper corner of Figure 3, where $F_{\mathrm{Cam.}}$ and D_{SK} are fed into DepthNet to fuse depth features with multi-view camera features. Initially, $F_{\mathrm{Cam.}}$ and D_{SK} are concatenated, followed by processing through three sets of CBR-module (see Figure 3) compromising a 2D convolution layer with Batch Normalization and ReLU activations. This results in the generation of a new depth-aware camera feature, denoted as $F_{\mathrm{DC}} \in \mathbb{R}^{(B_S \times N_C) \times C_{\mathrm{DC}} \times \frac{H}{8} \times \frac{W}{8}}$. Subsequently, F_{DC} is split along

the C_{DC} dimension into two new features: a novel depth feature, define as $\hat{F_D} \in \mathbb{R}^{(B_S \times N_C) \times \hat{C_D} \times \frac{H}{8} \times \frac{W}{8}}$ and a novel image context feature, define as $\hat{F_C} \in \mathbb{R}^{(B_S \times N_C) \times \hat{C_C} \times \frac{H}{8} \times \frac{W}{8}}$. It's important to note that $C_{\mathrm{DC}} = \hat{C_C} + \hat{C_D}$, indicating the division of the combined feature space into distinct depth and image feature components. Subsequently, $\hat{F_D}$ is subjected to a softmax operation and then multiplied with $\hat{F_C}$, resulting in a novel image feature with depth information, represented as $\hat{F}_{\mathrm{DC}} \in \mathbb{R}^{(B_S \times N_C) \times \hat{C_C} \times \hat{C_D} \times \frac{H}{8} \times \frac{W}{8}}$. Finally, adopting operations consistent with LSS [120] and BEVDepth [84], we utilize pre-generated 3D spatial coordinates and $\hat{F_{DC}}$ with BEV Pooling to output the camera BEV feature, thereby completing the camera-to-BEV transformation, and finally outputs the camera BEV feature, define as $F_B^C \in \mathbb{R}^{B_S \times \hat{C_C} \times H_B \times W_B}$.

3.1.2 LocalAlign-v2 (Query)

Following our discussion of how LocalAlign-v2 (LSS) addresses feature misalignment in LSS-based BEV representations such as BEVDepth and BEVFusion, we now turn to LocalAlign-v2 (Query), which focuses on aligning BEV features in query-based architectures like BEVFormer. Although the underlying mechanisms for BEV generation differ between these two paradigms, the core idea of mitigating feature misalignment through neighborhood-aware alignment remains consistent.

As illustrated in Figure 4, BEVFormer first samples 3D query points $P_{3D}^{loc} \in \mathbb{R}^{B_S \times N_P \times 3}$ and initializes them as BEV queries $Q_{BEV} \in \mathbb{R}^{B_S \times N_P \times C}$. These query points are projected onto the corresponding images to obtain their 2D coordinates $P_{2D}^{loc} \in \mathbb{R}^{B_S \times N_P \times 2}$, which are then used in deformable attention to extract image features for BEV construction. However, due to inevitable projection matrix errors in real-world settings, these 2D coordinates are often inaccurate, leading to misalignment between the sampled image features and their true semantic counterparts.

To address this issue, LocalAlign-v2 (Query) introduces a neighborhood-based correction mechanism. For each BEV query, it identifies K neighboring BEV queries $Q_{BEV}^{neighbor} \in \mathbb{R}^{B_S \times K \times N_P \times C}$ based on proximity in the image plane. These neighbors, along with the original BEV query, are fed into a dual-transform module that leverages a self-attention mechanism to adaptively fuse information from the local neighborhood. This process enhances the robustness and alignment of BEV features by compensating for projection-induced errors.

The significance of LocalAlign-v2 (Query) lies in its ability to generalize feature alignment to the query-based BEV paradigm, which is inherently more sensitive to calibration errors due to its reliance on reference point projections. By introducing a principled and learnable neighborhood-based correction mechanism, GraphBEV++ (Query) effectively extends our framework's applicability from dense BEV fusion methods to sparse, query-driven architectures. This ensures that GraphBEV++ not only maintains compatibility with a broader class of BEV-based perception systems but also enhances their resilience to real-world misalignment, thereby improving downstream tasks such as detection, prediction, and planning in end-to-end autonomous driving.

3.1.3 Adaptive KNN for Efficient LocalAlign-v2

Feature misalignment across modalities varies with the spatial scale and location of objects. We observe that large, nearby objects are generally more robust to local misalignment, whereas small or distant objects are more sensitive to projection-induced noise. The original GraphBEV adopts a fixed number of neighbors K in the LocalAlign module, which fails to account for such discrepancies. Applying a uniform K across all points leads to redundant computation for well-aligned regions and may result in overfitting uninformative areas.

To address this limitation, we introduce an **Adaptive KNN** mechanism that dynamically determines the number of neighbors K_i for each projected point. This allows LocalAlign-v2 to focus more effectively on regions prone to misalignment while improving computational efficiency.

During training, where ground-truth 3D bounding boxes are available, we use object size as a proxy for alignment difficulty. For each projected point i, we compute its associated object's scale size_i as the average of the bounding box's length, width, and height. The number of neighbors K_i is then determined by:

$$K_i = \mathrm{clip}\left(\frac{\gamma}{\mathrm{size}_i + \epsilon},\ K_{\min},\ K_{\max}\right) \tag{2}$$

where γ is a scaling factor, ϵ prevents division by zero, and $\mathrm{clip}(\cdot)$ constrains K_i to a reasonable range (e.g., $[4, 16]$). This rule ensures smaller or thinner objects receive more neighbors for better alignment, while large, well-perceived objects require fewer.

During inference, ground-truth labels are unavailable. Instead, we use the depth value $z_{c,i}$ of each projected point, obtained via LiDAR-to-camera projection, as an indirect indicator of alignment difficulty. The neighbor count is then computed as:

$$K_i = \mathrm{clip}\left(K_{\min} + \alpha \cdot \log(1 + z_{c,i}),\ K_{\min},\ K_{\max}\right) \tag{3}$$

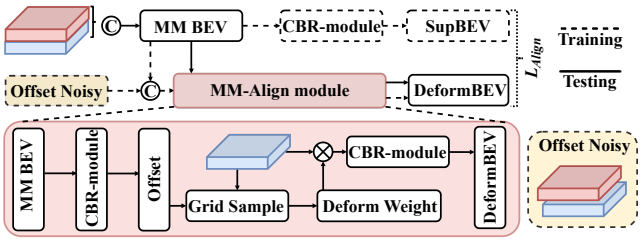

Fig. 5 The overview of **GlobalAlign-v2 (Deformable)** pipeline. The GlobalAlign-v2 (Deformable) module addresses the issue of LSS-based multi-modal BEV feature misalignment. During training, we add offset noise to simulate the global misalignment problem in camera and LiDAR BEV features. It is supervised through a simple CBR-module to learn the offsets of camera BEV features. We do not introduce noise during testing and employ learnable offsets for forward inference.

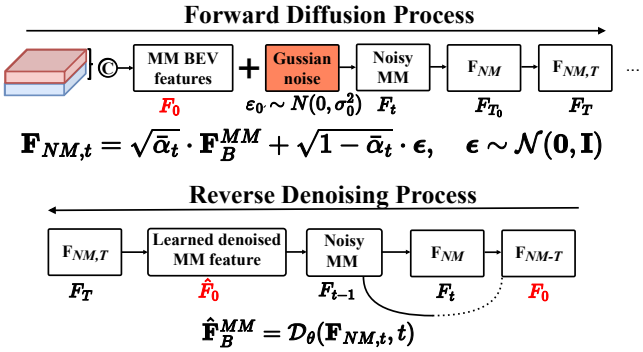

Fig. 6 The overview of GlobalAlign-v2 (Diffusion) pipeline. The GlobalAlign-v2 (Diffusion) module tackles the challenge of global misalignment in query-based multi-modal BEV representations. By treating global misalignment as a form of diffusion noise, the model progressively denoises noisy BEV features to recover robust and well-aligned representations.

Here, α controls the sensitivity of K_i to depth. This **depth-aware adaptation** increases the receptive field for distant or uncertain points, improving robustness without incurring significant overhead.

Overall, the Adaptive KNN strategy offers three key benefits. First, it improves efficiency by avoiding unnecessary computation for well-aligned regions. Second, it enhances robustness by allocating more context to points prone to misalignment. Third, it introduces no additional network parameters and can be seamlessly integrated into standard KNN-based neighbor search pipelines. This approach is also **backbone-agnostic**, making it compatible with both LSS-based and query-based BEV representations within the LocalAlign-v2 framework.

3.2 GlobalAlign-v2

GlobalAlign-v2 addresses the problem of global feature misalignment in multi-modal BEV fusion. It comprises two main variants: **GlobalAlign-v2 (Deformable)**, as illustrated in Figure 5, and **GlobalAlign-v2 (Diffusion)**, as shown in Figure 6. The Deformable variant is designed for explicit multi-modal BEV representations (e.g., BEVFusion), utilizing deformable convolutions for feature alignment. In contrast, the Diffusion variant targets implicit BEV representations (e.g., BEVFormer), and employs a multi-step diffusion-based denoising process to achieve more robust global alignment.

3.2.1 GlobalAlign-v2 (Deformable)

In the real world, feature misalignment is inevitable due to calibration matrix discrepancies between LiDAR and camera sensors. Although LocalAlign-v2 module mitigates the local misalignment issue in the camera-to-BEV process, deviations may still exist in camera-BEV features. During LiDAR-camera BEV fusion, despite being in the same spatial domain, inaccuracies in depth lead to global misalignment from the view transformer and overlooking of global offsets between LiDAR and camera BEV features. To tackle the global misalignment issue described above, we introduce the GlobalAlign-v2 (Deformable) module, employing learnable offsets to achieve the alignment of global multi-modal BEV features. As shown in Figure 5, we use clean datasets such as nuScenes [6] for training, which exhibit minimal deviations that can be considered negligible. The supervised information is derived from the features obtained after the fusion and convolution of LiDAR and camera BEV features. During training, we introduce global offset noise and employ learnable offsets. In the LiDAR branch, the LiDAR feature is flattened along the Z-axis to form the LiDAR BEV feature, defined as $F_B^L \in \mathbb{R}^{B_S \times \hat{C}_L \times H_B \times W_B}$. Initially, we concatenate F_B^L and F_B^C to obtain a fused BEV feature, denoted as $F_B^{MM} \in \mathbb{R}^{B \times (\hat{C}_C + \hat{C}_L) \times H_B \times W_B}$. Subsequently, F_B^{MM} undergoes a convolution operation, resulting in a new fused feature, denoted as $\hat{F}_B \in \mathbb{R}^{B_S \times \hat{C}_L \times H_B \times W_B}$. Notably, $\hat{F}_B$ will be utilized as a supervision signal during the training process.

As shown in Figure 5, we introduce random offset noise to the camera dimension of F_B^{MM} to obtain a new noisy feature $F_N^{MM} \in \mathbb{R}^{B_S \times (\hat{C}_C + \hat{C}_L) \times H_B \times W_B}$, simulating the global misalignment issue originating from camera BEV features. Notably, the LiDAR BEV feature is directly flattened, thus more accurate. Then, F_N^{MM} is input into the MM-Align module for global offset learning. F_N^{MM} is processed through the CBR-module with basic convolution operations to learn offsets, defined as

$F^O \in \mathbb{R}^{B_S \times 2 \times H_B \times W_B}$, where 2 corresponds to the offset coordinates (u, v). Subsequently, LiDAR BEV features F_B^L and F^O undergo grid sampling to generate new deform weights, defined as $F_W^D \in \mathbb{R}^{B_S \times \hat{C_L} \times H_B \times W_B}$. The purpose of grid sampling is to utilize offsets for spatial transformation of LiDAR BEV feature F_B^L, with learnable shifts dynamically adjusting to capture spatial dependencies more flexibly than standard convolution operations. Afterward, F_W^D is multiplied by LiDAR BEV features F_B^L to dynamically adjust features, followed by standard convolution operations through the CBR-module, culminating in the output Deform BEV defined as $F_B^D \in \mathbb{R}^{B_S \times \hat{C_L} \times H_B \times W_B}$. Finally, during training, we supervise F_B^D using $\hat{F}_B$ previously mentioned, and employ the L_{Align} for supervision as follows.

$$\mathcal{L}_{\text{Align}} = \frac{1}{N_B} \sum_{i=1}^{N_B} (\hat{F_B}_i - {F_B^D}_i)^2, \tag{4}$$

where $N_B = B_S \times H_B \times W_B$ represents the total number of elements, and $\hat{F}_{Bi}$ and ${F_B^D}_i$ denote the value of the ith element in $\hat{F}_B$ and F_B^D, respectively. This formula calculates the mean of the squared differences between corresponding positions in the two feature maps, serving as the loss.

3.2.2 GlobalAlign-v2 (Diffusion)

Query-based multi-modal BEV representations are inherently implicit, rendering it infeasible to apply GlobalAlign-v2 (Deformable), as depicted in Figure 6, for explicit global misalignment correction. To overcome this limitation, we introduce GlobalAlign-v2 (Diffusion), a diffusion-based framework specifically designed to address global misalignment in query-based BEV representations. Unlike traditional offset learning approaches with fixed noise injection, the proposed method leverages a progressive and learnable denoising process, enabling more effective modeling of the complex spatial misalignment patterns inherent to multi-modal fusion. Unlike conventional image-generation diffusion models, GlobalAlign-v2 (Diffusion) performs lightweight feature-level alignment on compact BEV representations. We adopt $T = 4$ throughout all experiments to achieve a favorable trade-off between robustness and efficiency.

It is worth noting that GlobalAlign-v2 (Diffusion) shares the same alignment objective as GlobalAlign-v2 (Deformable): both aim to estimate and compensate for global feature misalignment through spatial correction. The key difference lies in how the correction is performed. GlobalAlign-v2 (Deformable) adopts a one-shot alignment strategy that directly predicts spatial offsets from the current BEV representation, which is effective when misalignment can be explicitly observed in dense BEV features. However, query-based BEV representations encode spatial information implicitly within latent query embeddings, making global misalignment difficult to directly model using explicit offset prediction. Simply stacking multiple deformable alignment layers only increases network depth while still relying on deterministic offset estimation. In contrast, GlobalAlign-v2 (Diffusion) reformulates global alignment as a progressive denoising process. By gradually injecting and removing misalignment noise, the model performs multi-step offset refinement rather than a single offset prediction. Therefore, the proposed diffusion framework can be viewed as a generalized iterative alignment mechanism that extends deformable alignment to implicit query-based BEV representations.

As shown in Figure 6, we inject random offset noise into the camera BEV features to simulate global misalignment, denoted as:

$$F_{N,0}^{MM} = F_B^{MM} + \epsilon_0, \quad \epsilon_0 \sim \mathcal{N}(0, \sigma_0^2), \tag{5}$$

where F_B^{MM} is the clean fused BEV feature, and ϵ_0 represents initial Gaussian noise with variance σ_0^2.

We then apply a forward diffusion process over T discrete time steps, gradually adding noise according to a predefined schedule $\{\beta_t\}_{t=1}^T$, producing a series of noisy features $\{F_{N,t}^{MM}\}_{t=1}^T$. This process models the progressive corruption of BEV features by global misalignment noise.

The core of our strategy lies in training a neural network $\mathcal{D}_\theta$ to learn the reverse denoising process:

$$\hat{F}_B^{MM} = \mathcal{D}_\theta(F_{N,t}^{MM}, t), \tag{6}$$

which estimates the clean fused BEV feature $\hat{F}_B^{MM}$ from a noisy input $F_{N,t}^{MM}$ at any diffusion step t. Here, $\mathcal{D}_\theta$ adopts a similar architecture to the original MM-Align module of GlobalAlign-v2 (Deformable) but is enhanced to condition on the diffusion timestep and progressively remove noise.

At each reverse step, the denoised feature $\hat{F}_B^{MM}$ is used to predict spatial offsets F_t^O, which in turn guide the grid sampling operation applied on LiDAR BEV features F_B^L to obtain deformable aligned features:

$$F_{W,t}^D = \text{GridSample}(F_B^L, F_t^O), \quad F_{B,t}^D = F_{W,t}^D \odot F_B^L, \tag{7}$$

where $\odot$ denotes element-wise multiplication. This iterative correction allows the model to refine global alignment in a noise-adaptive manner.

We optimize the diffusion model parameters θ by minimizing a combination of reconstruction loss and denoising score matching loss across all diffusion steps:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{t,F_B^{MM},\epsilon_t} \left\| F_B^{MM} - \mathcal{D}_\theta(F_{N,t}^{MM}, t) \right\|^2 . \quad (8)$$

By integrating diffusion-based denoising into the global alignment module, our approach models complex and varying global misalignments in a progressive manner, leading to more accurate multi-modal fusion.

Conceptually, the proposed diffusion framework differs from a deeper or recurrent alignment network in that the denoising trajectory explicitly models the distribution of global misalignment noise. Rather than repeatedly applying the same alignment operator, the model learns to recover clean BEV representations from progressively corrupted states, enabling adaptive correction under different levels of global misalignment. Consequently, diffusion-based alignment provides a more robust solution for query-based BEV representations, where misalignment is implicitly encoded and difficult to correct using conventional deformable alignment alone.

3.3 3D Object Detection (3DOD)

To comprehensively validate the effectiveness of our method in addressing feature misalignment in 3D object detection, we design and evaluate our approach across multiple multi-modal fusion scenarios, including both **LiDAR-Camera** and **Radar-Camera** combinations.

LiDAR-Camera Fusion. We consider two mainstream BEV construction paradigms:

- **LSS-based**: We integrate our alignment modules into two representative LSS-based frameworks: *UniTR* [149] and *BEVFusion* [108].
- **Query-based**: We apply our method to *BEVFormer-M* [92], which adopts implicit BEV queries for BEV feature construction.

Radar-Camera Fusion. We further validate the generality of our approach by incorporating it into radar-camera fusion. Specifically, we adopt the **LSS-based** framework *HVDetFusion* [75] as the baseline for evaluation.Unlike LiDAR, Radar measurements are sparse and contain limited elevation information. Nevertheless, LocalAlign-v2 relies on local geometric neighborhood consistency rather than dense depth estimation. Therefore, the same KD-Tree-based neighborhood construction can be directly applied to projected Radar observations, enabling robust alignment under Radar-Camera fusion.

3.4 3D Occupancy Prediction (3DOP)

To further investigate the applicability of feature alignment beyond 3D object detection, we extend GraphBEV++ to the **3D occupancy estimation** task, where feature misalignment remains a critical challenge due to the reliance on multi-view geometric projection and feature aggregation.

We conduct experiments based on *GaussianFormer-T* [53], a representative query-based occupancy framework. Since GaussianFormer adopts a BEVFormer-style query paradigm for occupancy prediction, we integrate our **GraphBEV++ (Query)** variant into the framework. This setting enables us to evaluate whether the proposed LocalAlign-v2 and GlobalAlign-v2 modules can effectively improve feature alignment quality and enhance occupancy perception under complex driving scenarios.

3.5 End-to-End Autonomous Driving (E2E-AD)

Feature misalignment permeates the entirety of autonomous driving tasks, adversely affecting not only perception performance but also influencing downstream components such as motion prediction and planning. We are the first article dedicated to resolving misalignment issues within the end-to-end framework. To further demonstrate the scalability and generalizability of our approach, we extend GraphBEV++—originally designed for 3D object detection—into a unified end-to-end autonomous driving framework. This extension simultaneously enhances the performance of scene perception, motion forecasting, and ego-planning modules.

Our method is applicable to a wide range of BEV-based end-to-end driving systems, including *UniAD* [49], *FusionAD* [175], *VAD* [62], *MomAD* [130], and *WoTE* [85]. Specifically, for vision-only BEV frameworks such as *UniAD* [49], *VAD* [62], and *MomAD* [130], we adopt the LocalAlign-v2 (Query) module to mitigate local misalignment. For multi-modal BEV systems like *FusionAD* [175] and *WoTE* [85], we employ both LocalAlign-v2 (Query) and the proposed GlobalAlign-v2 (Diffusion) to address both local and global misalignments across modalities.

In summary, the alignment strategy introduced in GraphBEV++ exhibits strong extensibility and robustness, making it well-suited for a diverse set of perception and planning tasks within modern end-to-end autonomous driving pipelines.

4 Experiments

4.1 Datasets

Bench2Drive (E2E-AD). We conduct training and evaluation of GraphBEV++ on **Bench2Drive** [60], a closed-loop evaluation protocol based on the CARLA Leaderboard 2.0 [29] for E2E-AD. It provides a base training set of 1000 clips, with 950 used for training and 50 for open-loop validation. Each clip captures approximately 150 meters of continuous driving in a specific traffic scenario. For closed-loop evaluation, we use the official 220 routes, covering 44 interactive scenarios with 5 routes each.

NAVSIM (E2E-AD). We conduct training and evaluation of GraphBEV++ on **NAVSIM** [25] dataset. NAVSIM is a real-world, planning-oriented dataset that builds upon OpenScene [23], a compact redistribution of nuPlan [7], the largest publicly available annotated driving dataset. It leverages eight cameras to achieve a full 360° field of view, along with a merged LiDAR point cloud derived from five sensors. Annotations are provided at 2 Hz and include both HD maps and object bounding boxes. The dataset is specifically designed to emphasize challenging driving scenarios involving dynamic changes in driving intentions, while deliberately excluding trivial cases such as stationary scenes or constant-speed cruising.

NuScenes (E2E-AD&3DOD&3DOP). We conduct extensive open-loop experiments on the **nuScenes** dataset [6], which consists of 1000 driving scenes (700 for training, 150 for validation and 150 for test). Each scene lasts 20 seconds and includes around 40 keyframes annotated at 2 Hz. Each sample contains six images from surround-view cameras (covering 360° FOV), and point clouds from both LiDAR and radar sensors.

NuScenes-C (E2E-AD&3DOD). NuScenes-C [28] is a corrupted benchmark derived from the nuScenes validation set, introducing various types of noise to assess the robustness of planning models. It includes 27 corruption types applied at 5 severity levels. To evaluate robustness under adverse weather conditions, we select three representative weather corruptions — Rain, Snow, and Fog — as our test scenarios. In addition, to validate the robustness of **feature alignment**, we have followed Ref. [28] to simulate misalignment caused by LiDAR and camera projection errors. It is worth noting that Ref. [28] only adds noise to the validation dataset, rather than the train and test datasets. For the 3D detection task, we use noise severity levels from 1 to 5 and report the mean values. In the end-to-end autonomous driving tasks, we use noise severity levels from 1 to 10, with mean values reported.

Argoverse 2 (3DOD). We further conduct long-range experiments on the recently released Argoverse 2 (AV2) dataset [159] to demonstrate the superiority of our GraphBEV++ in long-range detection. AV2 has a large-scale data, and it contains 1000 sequences in total, 700 for training, 150 for validation, and 150 for testing. In addition to average precision (AP), AV2 adopts a composite score as an evaluation metric, which takes both AP and localization errors into account. The perception range in AV2 is 200 meters (cover area of 400m × 400m). We test on the AV2 dataset due to its extensive range of long-distance scenarios, where the issue of feature misalignment intensifies with increasing distance.

Waymo and Waymo-C (3DOD). Waymo Open dataset (WOD) [141] is a well-known benchmark for large-scale outdoor 3D perception, comprising 1150 scenes which are divided into 798 scenes for training, 202 scenes for validation, and 150 scenes for testing. Each scene includes about 200 frames, covering a perception range of 150m × 150m. Similar to nuScenes-C, Waymo-C [28] is constructed by applying all 27 corruptions to the Waymo validation set with 5 severities.

4.2 Evaluation Metrics

Bench2Drive (E2E-AD). The **Bench2Drive** [60] includes five metrics for closed-loop evaluation: Driving Score (DS), Success Rate (SR), Efficiency, Comfortness, and Multi-Ability. The Success Rate quantifies the proportion of routes successfully completed within the allotted time. The Driving Score follows CARLA [11], incorporating both route completion status and violation penalties, where infractions reduce the score via discount factors. Efficiency and Comfortness are used to measure the speed performance and comfort of the autonomous driving system during the driving process, respectively. Multi-Ability measures 5 advanced skills, including 'Merging, Overtaking, Emergency Brake, Give Way, and Traffic Sign', independently for urban driving.

NAVSIM (E2E-AD). NAVSIM [25] benchmarks planning performance using nonreactive simulations and closed-loop metrics for comprehensive evaluation. In this study, we employ the proposed PDM score (PDMS) [25], which is a weighted combination of several sub-scores: no at-fault collisions (NC), drivable area compliance (DAC), time-to-collision (TTC), comfort (Comf.), and ego progress (EP).

NuScenes and NuScenes-C (3DOD). For object detection on the nuScenes [6] and NuScenes-C [28] dataset, evaluation metrics include mAP and the nuScenes detection score (NDS). mAP is calculated by averaging over the distance thresholds of 0.5m, 1m, 2m,

and 4m across all categories. NDS is a weighted average of mAP and five other true positive metrics that measure translation, scaling, orientation, velocity, and attribute errors.

NuScenes and NuScenes-C (E2E-AD). For the metrics of perception tasks, we use mAP and NDS to evaluate the detection tasks, adopt Average Multi-object Tracking Accuracy (AMOTA) and Average Multi-object Tracking Precision (AMOTP) to evaluate the tracking tasks, use intersection-over-union (IoU) to evaluate the mapping tasks. To evaluate the prediction and planning tasks, we adopt conventional metrics, including End-to-end Prediction Accuracy (EPA), Average Displacement Error (ADE), Final Displacement Error (FDE), and Miss Rate (MR) to evaluate the performance of motion prediction. For future occupancy prediction, we use the metrics Future Video Panoptic Quality (VPQ) and IoU for near (30 × 30m) and far (100 × 100m) range, following FIERY [45]. For planning evaluation, we adopt the commonly used L2 displacement error (L2) and collision rate as the primary metrics.

Argoverse 2 (3DOD). For object detection on the Argoverse2 [159] dataset, mAP is adopted as the evaluation metric. We test on the AV2 dataset due to its extensive range of long-distance scenarios, where the issue of feature misalignment intensifies with increasing distance.

Waymo and Waymo-C (3DOD). For evaluation metrics, WOD [141] employs 3D mean Average Precision (mAP) and mAP weighted by heading accuracy (mAPH). Each object is divided into two difficulty levels: L1 is for objects detected with more than five points and L2 is for those at least one point. For Waymo-C, the official evaluation metrics are mAP and mAPH by taking the heading accuracy into consideration. We similarly calculate the corruption robustness and relative corruption error on Waymo-C.

NuScenes (3D Occupancy Prediction). For 3D semantic occupancy prediction, we use the intersection-over-union (IoU) of occupied voxels, regardless of their semantic categories, as the evaluation metric for the scene completion (SC) task. For the semantic scene completion (SSC) task, we adopt the mean IoU (mIoU) over all semantic classes:

$$\text{IoU} = \frac{TP}{TP + FP + FN}, \tag{9}$$

$$\text{mIoU} = \frac{1}{C} \sum_{i=1}^{C} \frac{TP_i}{TP_i + FP_i + FN_i}, \tag{10}$$

where TP, FP, and FN denote the numbers of true positives, false positives, and false negatives, respectively, and C is the total number of semantic classes.

4.3 Implementation Details

We implement GraphBEV++ within the PyTorch [118], built upon the open-source BEVFusion [108] and OpenPCDet [144]. For the LiDAR branch, feature encoding is performed using SECOND [167] to obtain LiDAR BEV features, with voxel dimensions set to [0.075m, 0.075m, 0.2m] and point cloud ranges specified as [-54m, -54m, -5m, 54m, 54m, 3m] across the X, Y, and Z axes, respectively. The camera branch employs a Swin Transformer [107] as the backbone, integrating Heads of numbers 3, 6, 12, 24, and utilizing FPN [43] to fuse multi-scale feature maps. The resolution of input images is adjusted and cropped to 256 × 704. In the LSS [120] configuration, frustum ranges are set with X coordinates [-54m, 54m, 0.3m], Y coordinates [-54m, 54m, 0.3m], Z coordinates [-10m, 10m, 20m], and depth range [1m, 60m, 0.5m].

For occupancy prediction, the spatial range is set to $[-50\,\text{m}, 50\,\text{m}]$ along the X- and Y-axes and $[-5\,\text{m}, 3\,\text{m}]$ along the Z-axis. The final occupancy volume has a resolution of $200 \times 200 \times 16$ with a voxel size of 0.5 m. The input image resolution is 1600×900. The network adopts a hierarchical architecture with $M = 4$ levels, where skip connections are not applied at Level 0. For the nuScenes dataset, we employ ResNet101-DCN [44, 24] initialized with FCOS3D [153] pretrained weights as the image backbone. The features from Stages 1–3 are processed by an FPN [97] to generate multi-scale image features. The numbers of 2D–3D spatial attention layers are set to 1, 3, and 6 for the three levels, respectively.

During training, we employ data augmentation for ten epochs, including random flips, rotations (within the range $[-\frac{\pi}{4}, \frac{\pi}{4}]$), translations (with std=0.5), and scaling in [0.9, 1.1] for LiDAR data enhancement. We use CBGS [195] to resample the training data. Additionally, we use random rotation in $[-5.4°, 5.4°]$ and random resizing in [0.38, 0.55] to augment the images. The Adam optimizer [73] is used with a one-cycle learning rate policy, setting the maximum learning rate to 0.001 and weight decay to 0.01. The batch size is 24, and our training is conducted on 8 NVIDIA GeForce RTX 3090 24G GPUs. During inference, we remove Test Time Augmentation (TTA) data augmentation, and the batch size is set to 1 on an A100 GPU. All latency measurements are taken on the same workstation with an A100 GPU.

4.4 Comparisons with State-of-the-Art Methods

4.4.1 3D Object Detection

NuScenes (val). We first compare GraphBEV++ with recent SOTA methods on the nuScenes validation set for 3D object detection, as shown in Table 2. GraphBEV++ achieves competitive performance under both LSS-based and query-based BEV paradigms. Specifically, GraphBEV++ (LSS) obtains 70.7

Among the multi-modal methods, point-level approaches such as PointPainting [147], PointAugmenting [148], and MVP [177] directly augment LiDAR points with image features, but they cannot explicitly handle feature misalignment in the BEV space. Feature-level methods, including GraphAlign [135], AutoAlignV2 [18], TransFusion [1], and DeepInteraction [171], alleviate cross-modal misalignment before BEV construction, but they do not directly address the misalignment introduced during the camera-to-BEV transformation. In contrast, GraphBEV++ explicitly models both local projection-induced misalignment and global BEV-level misalignment, leading to more robust multi-modal fusion.

It is also noteworthy that GraphBEV++ shows clear advantages on several challenging object categories. For example, GraphBEV++ (Query) achieves the best performance on Car, Truck, Barrier, Motor., and Ped., while GraphBEV++ (LSS) achieves the best result on T.C. These improvements indicate that feature alignment is particularly beneficial for categories that are sensitive to geometric projection errors and local feature inconsistency. Furthermore, compared with BEVFormer-M, GraphBEV++ (Query) improves mAP from 71.2% to 71.4% and NDS from 73.2% to 73.4%, confirming the compatibility of the proposed method with query-based BEV architectures.

NuScenes (test). As shown in Table 2, GraphBEV++ (LSS) also achieves strong performance on the nuScenes test set. Compared with BEVFusion-MIT [108], GraphBEV++ improves mAP from 70.2% to 72.0% and NDS from 72.9% to 74.0%, yielding gains of +1.8% mAP and +1.1% NDS. Compared with the conference version GraphBEV, GraphBEV++ improves mAP by +0.3% and NDS by +0.4%, demonstrating that the upgraded LocalAlign-v2 and GlobalAlign-v2 modules further enhance the detection capability.

Although the nuScenes test set contains relatively clean sensor calibration, feature misalignment can still occur due to projection noise, calibration deviation, and depth estimation errors, as illustrated in Fig. 1(a). By incorporating neighborhood-aware depth features through KD-Tree search, GraphBEV++ effectively improves the robustness of camera-to-BEV transformation. In particular, GraphBEV++ achieves the best results on Car, C.V., Ped., and T.C., and obtains second-best performance on Bus and Trailer. These gains are especially meaningful for small or structurally complex objects, whose projected image features are more sensitive to slight LiDAR-camera misalignment. Overall, the test-set results verify that GraphBEV++ provides a robust and generalizable feature alignment solution for BEV-based multi-modal 3D object detection.

NuScenes-C. As shown in Table 3, we conduct comparative experiments under the nuScenes-C misalignment noise setting, including SparseFusion, BEVFusion, BEVFormer, and GraphBEV. BEVFusion suffers a performance drop of 11.2% in mAP and 8.2% in NDS when transitioning from the clean to the noisy setting. Similarly, BEVFormer exhibits a drop of 10.8% in mAP and 9.2% in NDS, highlighting the challenge of feature misalignment. Moreover, GraphBEV++ (LSS) and GraphBEV++ (Query) further reduce the performance degradation to 2.0% and 3.2% mAP, respectively. These results verify that GraphBEV++ not only achieves better performance under clean conditions, but also maintains high robustness in noisy, real-world scenarios. Although GraphBEV++ (LSS) shows a slightly larger relative mAP drop than GraphBEV, it consistently achieves higher performance under noisy conditions, improving mAP from 69.1 to 69.3 and NDS from 72.0 to 72.3. These results indicate that GraphBEV++ enhances the performance ceiling while preserving strong robustness against feature misalignment.

As shown in Figure 7, we introduce varying levels of misalignment noise into the BEV perception module to evaluate its impact on end-to-end autonomous driving performance. Both FusionAD and UniAD are tested under two training settings: using clean data and employing misalignment-based data augmentation (RoarNet). While the augmented models show improved robustness under severe noise conditions, their performance on clean test data noticeably degrades. This reveals a trade-off: random data augmentation helps models generalize under noise but lacks the capacity to accurately simulate diverse real-world misalignment scenarios. In contrast, our GraphBEV++ addresses the misalignment problem at the algorithmic level, leading to a more principled and effective solution.

NuScenes (Radar and Camera). As shown in Table 4, we compare GraphBEV++ with existing radar-camera fusion methods under both clean and noisy settings. Compared with HVDetFusion [75], GraphBEV++ improves mAP from 45.1% to 45.9% under clean settings and from 40.1% to 45.0% under noisy settings. Notably, the improvement under noisy condi-

Table 2 Comparison with the SOTA methods on the nuScenes validation and test set. 'C.V.', 'Motor.', 'Ped.' and 'T.C.' are short for construction vehicles, motorcycles, pedestrians, and traffic cones. The Modality column: 'L' denotes LiDAR-only input, while 'LC' indicates the use of both LiDAR and camera data. † means using TTA (test-time augmentation). * denotes the re-implementation. The best results are highlighted in **bold**, and the second-best results are indicated with underlining.

Method	Venue	Modality	mAP	NDS	Car	Truck	C.V.	Bus	Trailer	Barrier	Motor.	Bike	Ped.	T.C.
Performances on validation set														
SparseBEV[15]	TPAMI'26	C	43.2	54.5										
Refine3D[76]	AAAI'26	C	53.9	62.4										
DGFSD[186]	MM'25	L	66.8	71.3	87.9	63.4	26.4	77.9	47.2	69.8	75.8	56.1	88.1	75.7
TransFusion-L [1]	CVPR'22	L	65.1	70.1	86.5	59.6	25.4	74.4	42.2	74.1	72.1	56.0	86.6	74.1
FSDv2[32]	TPAMI'24	L	64.7	70.4										
DepthFusion[55]	TMM'26	LC	71.2	**74.0**										
HD-Fusion[68]	TII'26	LC	70.5	72.9	88.8	60.5	**35.4**	**80.7**	**48.1**	68.1	79.8	**69.7**	90.3	83.3
ObjectFusion [8]	ICCV'23	LC	69.8	72.3	89.7	65.6	31.8	77.7	42.8	75.2	79.4	65.0	89.3	81.1
GraphBEV [137]	ECCV'24	LC	70.1	72.9	89.9	64.7	31.1	76.0	43.8	76.0	80.1	67.5	89.2	82.2
BEVFormer-M [92]	TPAMI'25	LC	71.2	73.2										
BEVFormer-M* [92]	TPAMI'25	LC	70.9	73.0	90.4	65.7	32.1	77.1	45.0	77.5	81.2	68.0	89.7	82.2
GraphBEV++ (LSS)	-	LC	70.7	73.2	90.5	65.3	31.8	76.4	44.4	76.6	80.8	68.1	89.7	**83.4**
GraphBEV++ (Query)	-	LC	**71.4**	73.4	**91.6**	**66.2**	32.3	77.4	45.5	**77.8**	**81.9**	68.4	**90.4**	82.5
Performances on test set														
OcRFDet[54]	ICCV'25	C	57.2	64.8										
PointPillars [74]	CVPR'19	L	40.1	55.0	76.0	31.0	11.3	32.1	36.6	56.4	34.2	14.0	64.0	45.6
FSDv2[32]	TPAMI'24	L	66.2	71.7	83.7	51.6	**66.4**	59.1	32.5	87.1	71.4	51.7	80.3	78.7
CenterPoint [176]†	CVPR'21	L	60.3	67.3	85.2	53.5	20.0	63.6	56.0	71.1	59.5	30.7	84.6	78.4
PointPainting [147]	CVPR'20	LC	46.4	58.1	77.9	35.8	15.8	36.2	37.3	60.2	41.5	24.1	73.3	62.4
PointAugmenting [148]†	CVPR'21	LC	66.8	71.0	87.5	57.3	28.0	65.2	60.7	72.6	74.3	50.9	87.9	83.6
MVP [177]	NeurIPS'21	LC	66.4	70.5	86.8	58.5	26.1	67.4	57.3	74.8	70.0	49.3	89.1	85.0
GraphAlign [135]	ICCV'23	LC	66.5	70.6	87.6	57.7	26.1	66.2	57.8	74.1	72.5	49.0	87.2	86.3
AutoAlignV2 [18]	ECCV'22	LC	68.4	72.4	87.0	59.0	33.1	69.3	59.3	-	72.9	52.1	87.6	-
TransFusion [1]	CVPR'22	LC	68.9	71.7	87.1	60.0	33.1	68.3	60.8	78.1	73.6	52.9	88.4	86.7
DeepInteraction [171]	NeurIPS'22	LC	70.8	73.4	87.9	60.2	37.5	70.8	63.8	80.4	75.4	54.5	90.3	87.0
BEVFusion-PKU [94]	NeurIPS'22	LC	69.2	71.8	88.1	60.9	34.4	69.3	62.1	78.2	72.2	52.2	89.2	85.2
ObjectFusion [8]	ICCV'23	LC	71.0	73.3	89.4	59.0	40.5	71.8	63.1	76.6	78.1	53.2	90.7	87.7
MV2DFusion[156]	TPAMI'25	LC	**74.5**	**76.7**										
MSMDFusion [65]	CVPR'23	LC	71.5	74.0	88.4	61.0	35.2	71.4	64.2	**80.7**	76.9	58.3	90.6	88.1
SparseFusion [163]	ICCV'23	LC	72.0	73.8	88.0	60.2	38.7	72.0	64.9	79.2	78.5	59.8	90.9	87.9
CMT [166]	ICCV'23	LC	72.0	73.8	88.0	**63.3**	37.3	**75.4**	**65.4**	78.2	**79.1**	**60.6**	87.9	84.7
BEVFusion-MIT [108]	ICRA'23	LC	70.2	72.9	88.6	60.1	39.3	69.8	63.8	80.0	74.1	51.0	89.2	86.5
DepthFusion[55]	TMM'26	LC	70.9	74.2										
GraphBEV [137]	ECCV'24	LC	71.7	73.6	89.2	60.0	40.8	72.1	64.5	80.1	76.8	53.3	90.9	88.9
GraphBEV++ (LSS)	-	LC	72.0	74.0	**89.5**	60.5	41.3	72.4	64.9	80.3	77.1	53.6	**91.1**	**89.3**

Table 3 Comparison with SOTAs on the **nuScenes (Clean)** and **nuScenes-C (Noisy)** dataset. Green denotes the relative performance drop from the clean to the noisy settings. All latency measurements are conducted on the same workstation with an A100 GPU.

Method	Setting	mAP	NDS
SparseFusion [163]	Clean	70.4	72.8
	Noisy	64.7-8.1%	67.1-7.8%
BEVFusion-MIT [108]	Clean	68.5	71.4
	Noisy	60.8-11.2%	65.7-8.2%
BEVFormer-M [92]	Clean	70.9	73.0
	Noisy	63.2-10.8%	66.3-9.2%
GraphBEV [137]	Clean	70.1	72.9
	Noisy	69.1-1.4%	72.0-1.2%
GraphBEV++ (LSS) [108]	Clean	70.7	73.2
	Noisy	69.3-2.0%	72.3-1.2%
GraphBEV++ (Query) [108]	Clean	71.4	73.4
	Noisy	69.1-3.2%	71.2-3.0%

Table 4 Comparison with SOTA methods on the nuScenes [6] validation set under clean and noise misalignment settings. 'RC' indicates the use of both Radar and camera data.

Methods	Venue	Modality	mAP (clean)	mAP (noise)
StreamPETR[152]	ICCV'23	C	45.0	40.8
RayFormer[22]	MM'24	C	45.9	41.4
RCBEVDet[99]	CVPR'24	RC	45.3	41.2
CRN[72]	ICCV'23	RC	49.0	43.8
HyDRa[160]	ICRA'25	RC	49.4	44.1
SparseInteraction[165]	MM'24	RC	45.8	-
HVDetFusion [75]	ArXiv'23	RC	45.1	40.1
GraphBEV++ (LSS)	-	RC	45.9+0.8	45.0+4.9
RaCFormer [21]	CVPR'25	RC	54.1	50.6
GraphBEV++ (LSS)	-	RC	54.8+0.7	54.2+3.6

tions (+4.9%) is significantly larger than that under clean conditions (+0.8%), demonstrating the robustness of the proposed alignment strategy against sensor misalignment. Furthermore, GraphBEV++ achieves competitive performance compared with recent radar-camera fusion methods, such as CRN [72] and HyDRa [160]. When integrated into the state-of-the-art RaCFormer [21], GraphBEV++ further improves mAP from 54.1% to 54.8% under clean settings and from 50.6% to 54.2% under noisy settings. These results indicate that the proposed LocalAlign-v2 generalizes well to sparse radar features and can consistently improve the robustness of different radar-camera fusion architectures.

Table 5 Comparison with UniTR (LSS) [149] on vehicle results under **Waymo** (clean) and **Waymo-C** (noise misalignment) settings (10% Waymo Training Data).

Methods	Modality	mAP/mAPH (clean) L1	mAP/mAPH (clean) L2	mAP/mAPH (noise) L1	mAP/mAPH (noise) L2
UniTR (LSS) [149]	LC	44.86/40.41	37.58/33.69	38.88/34.24	32.44/29.95
GraphBEV++ (LSS)	LC	45.73/41.17	38.51/35.14	43.50/38.97	36.48/33.08
		+0.87/+0.76	*+0.93/+1.45*	*+4.62/+4.73*	*+4.04/+3.13*

Table 6 Comparison with prior methods on **Argoverse2** validation set. Metrics: mAP (%)↑ for the overall results, AP (%)↑ for each category. * denotes result from paper [17]. † denotes result re-implentment.

Method	Venue	mAP	Veh.	Bus	Ped.	Stop.	Box.	Boll.	C-B.	M.-list	MPC.	M.-cycle	Bicycle	A-B.	School.	Truck.	C-C.	V-T.	Sign	Large.	Str.	Bic.-list
PETR[103]	ECCV'22	17.6																				
Sparse4Dv2[98]	Aexiv'23	18.9																				
StreamPETR[152]	ICCV'23	20.3																				
Far3D[63]	AAA'24	24.4																				
CenterPoint* [176]	CVPR'21	22.0	67.6	38.9	46.5	16.9	37.4	40.1	32.2	28.6	27.4	33.4	24.5	8.7	25.8	22.6	29.5	22.4	6.3	3.9	0.5	20.1
FSD* [33]	NeurPS'22	28.2	68.1	40.9	59.0	29.0	38.5	41.8	42.6	39.7	26.2	49.0	38.6	20.4	30.5	14.8	41.2	26.9	11.9	5.9	13.8	33.4
FSDv2 [33]	TPAMI'24	37.6	77.0	47.6	70.5	43.6	41.5	53.9	58.5	56.8	39.0	60.7	49.4	28.4	41.9	30.2	44.9	33.4	16.6	7.3	32.5	45.9
VoxelNeXt* [17]	CVPR'23	30.0	71.7	39.2	63.1	39.2	40.0	52.5	63.7	42.2	34.9	42.7	40.1	20.1	25.2	16.9	45.7	22.3	15.8	5.9	9.8	33.5
HEDNet[185]	NeurPS'23	37.1	78.2	47.7	67.6	46.4	45.9	56.9	67.0	48.7	46.5	58.2	47.5	23.3	40.9	21.6	46.8	27.9	20.6	6.9	27.2	38.7
SAFDNet[184]	CVPR'24	39.7	78.5	49.4	70.7	51.5	44.7	65.7	72.3	54.3	49.7	60.8	50.0	31.3	44.9	24.7	55.4	31.4	22.1	7.1	31.1	42.7
LION-Mamba[105]	NeurPS'24	41.5	75.1	43.6	73.9	53.9	45.1	66.4	74.7	61.3	48.7	65.1	56.2	21.7	42.7	25.3	58.4	28.9	23.6	8.3	49.5	47.3
UniMamba[66]	CVPR'25	42.0	78.9	47.9	74.3	51.8	46.8	67.8	76.9	55.8	51.7	62.8	52.4	30.2	44.6	24.6	59.4	32.2	23.2	6.7	41.5	48.5
GeoFormer[67]	ICCV'25	41.7	77.4	50.7	73.7																	
FSHNet[102]	CVPR'25	40.2	75.1	47.1	50.3	76.2	54.9	46.0	62.1	28.5	45.8	29.1	25.4	64.8	64.1	48.9	44.5	61.3	26.0	44.1	23.6	32.5
M3Net[180]	Arxiv'23	40.9	74.9	47.8	57.4	77.1	55.3	48.5	63.9	29.9	47.5	28.2	25.4	67.0	64.2	46.8	43.8	59.5	24.0	41.5	21.8	30.5
PGDC[79]	CVPR'26	41.7	76.2	49.0	58.5	78.1	56.8	49.8	65.1	31.2	48.5	31.0	26.9	68.1	65.8	50.3	46.0	62.5	27.5	45.8	25.0	33.8
BEVFusion† [108]	ICRA'23	43.1	83.1	48.1	64.2	48.2	54.5	58.1	65.3	45.1	39.8	49.3	48.1	33.1	38.1	28.5	54.1	37.4	34.6	31.2	34.4	38.1
GraphBEV [137]	ECCV'24	46.1	85.2	51.3	67.3	52.5	58.7	60.9	67.4	49.9	40.1	52.9	52.4	35.8	42.9	31.5	57.8	41.5	36.7	35.9	37.6	42.6
GraphBEV++ (LSS)	-	**46.7**	**85.9**	**51.7**	**67.8**	**53.0**	**59.1**	**61.3**	**67.8**	**50.3**	**40.6**	**53.3**	**52.8**	**36.3**	**43.4**	**32.0**	**58.2**	**41.9**	**37.2**	**36.4**	**38.1**	**43.0**

Table 7 Comparisons with the SOTA methods on BEV map segmentation on **nuScenes** validation set. The Modality column: 'L' denotes LiDAR-only input, while 'LC' indicates the use of both LiDAR and camera data.

Method	Venue	Drivable	Ped. Cross.	Walkway	Stop Line	Carpark	Divider	Mean
LSS[120]	ECCV'20	75.4	38.8	46.3	30.3	39.1	36.5	44.4
CVT[193]	CVPR'22	74.3	36.8	39.9	25.8	35.0	29.4	40.2
M2BEV[162]	ArXiv'22	77.2	-	-	-	-	40.5	-
MapPrior[197]	ICCV'23	81.7	54.6	58.3	46.7	53.3	45.1	56.7
X-Align[4]	WACV'23	82.4	55.6	59.3	49.6	53.8	47.4	58.0
MetaBEV[39]	ICCV'23	83.3	56.7	61.4	50.8	55.5	48.0	59.3
DDP[56]	ICCV'23	83.6	58.3	61.6	52.4	51.4	49.2	59.4
RGC[13]	WACV'24	81.7	57.1	60.5	51.7	53.8	53.5	59.7
BridgeTA[70]	ArXiv'25	83.3	58.6	62.9	53.6	56.6	50.1	60.8
PointPillars [74]	CVPR'19	72.0	43.1	53.1	29.7	27.7	37.5	43.8
CenterPoint [176]	CVPR'21	75.6	48.4	57.5	36.5	31.7	41.9	48.6
PointPainting [147]	CVPR'20	75.9	48.5	57.1	36.9	34.5	41.9	49.1
MVP [177]	NeurIPS'21	76.1	48.7	57.0	36.9	33.0	42.2	49.0
MapFusion[41]	IF'25	88.9	69.6	74.0	63.0	56.5	61.5	68.9
NRSeg[78]	TIP'26	59.1	16.9	21.9	12.1	16.8	21.0	24.6
BEVFusion [108]	ICRA'23	85.5	60.5	67.6	52.0	57.0	53.7	62.7
GraphBEV [137]	ECCV'24	86.3	60.9	69.1	53.1	57.5	53.1	63.3
GraphBEV++ (LSS)	-	86.8	61.5	69.7	53.6	58.1	53.1	63.8

Waymo and Waymo-C. As shown in Table 5, GraphBEV++ consistently surpasses UniTR [149] across all evaluation metrics under both clean and noisy settings on the Waymo-C benchmark (10% training data). In the noisy scenario, it achieves substantial improvements in L2 mAP and mAPH (+4.69%/+4.73%), demonstrating superior robustness to sensor misalignment noise.

Argoverse 2. As shown in Table 6, GraphBEV++ achieves the best overall performance with 46.7% mAP, outperforming the previous GraphBEV by +0.6% mAP and surpassing all existing LiDAR-only and multi-modal methods. These results demonstrate the effectiveness of the proposed LocalAlign-v2 and GlobalAlign-v2 modules in improving feature alignment and object localization. Furthermore, GraphBEV++ consistently improves performance across almost all categories. In particular, noticeable gains are observed on small, sparse, and long-tail categories, such as Bicyclist (43.0%), Motorcycle (53.3%), School Bus (52.8%), Vehicle Trailer (37.2%), and Large Vehicle (36.4%). Since these categories are more sensitive to feature misalignment and localization errors, the improvements indicate that GraphBEV++ can better preserve fine-grained geometric details during multi-modal fusion.

Table 8 3D semantic occupancy prediction results on nuScenes validation set. Since GaussianFormer follows a BEVFormer-style query-based architecture, we adopt GraphBEV++ (Query) in the comparison.

Method	Venue	IoU	mIoU	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. suf.	other flat	sidewalk	terrain	manmade	vegetation
MonoScene [9]	CVPR'22	23.96	7.31	4.03	0.35	8.00	8.04	2.90	0.28	1.16	0.67	4.01	4.35	27.72	5.20	15.13	11.29	9.03	14.86
Atlas [114]	ECCV'20	28.66	15.00	10.64	5.68	19.66	24.94	8.90	8.84	6.47	3.28	10.42	16.21	34.86	15.46	21.89	20.95	11.21	20.54
BEVFormer [91]	ECCV'22	30.50	16.75	14.22	6.58	23.46	28.28	8.66	10.77	6.64	4.05	11.20	17.78	37.28	18.00	22.88	22.17	13.80	22.21
TPVFormer [52]	CVPR'23	11.51	11.66	16.14	7.17	22.63	17.13	8.83	11.39	10.46	8.23	9.43	17.02	8.07	13.64	13.85	10.34	4.90	7.37
OccFormer [190]	ICCV'23	31.39	19.03	18.65	10.41	23.92	30.29	10.31	14.19	13.59	10.13	12.49	20.77	38.78	19.79	24.19	22.21	13.48	21.35
GaussianFormer [53]	ECCV'24	29.83	19.10	19.52	11.26	26.11	29.78	10.47	13.83	12.58	8.67	12.74	21.57	39.63	23.28	24.46	22.99	9.59	19.12
SurroundOcc [157]	CVPR'23	31.49	20.30	20.59	11.68	28.06	30.86	10.70	15.14	14.09	12.06	14.38	22.26	37.29	23.70	24.49	22.77	14.89	21.86
QuadricFormer[198]	NeurIPS'26	31.22	20.12	19.58	13.11	27.27	29.64	11.25	16.26	12.65	9.15	12.51	21.24	40.20	24.34	25.69	24.24	12.95	21.86
GaussianFormer-2[51]	CVPR'25	31.74	20.82	21.39	13.44	28.49	30.82	10.92	15.84	13.55	10.53	14.04	22.92	40.61	24.36	26.08	24.27	13.83	21.98
Gau-Occ[110]	CVPR'26	44.30	32.70	33.10	16.50	41.10	43.90	21.90	26.60	31.10	23.10	24.50	34.00	49.00	26.30	32.90	33.40	39.30	45.70
GaussianWorld[200]	CVPR'25	33.40	22.13	21.38	14.12	27.71	31.84	13.66	17.43	13.66	11.46	15.09	23.94	42.98	24.86	28.84	26.74	15.69	24.74
GaussianFormer-T[53]	ECCV'24	31.34	20.42	20.82	12.07	26.89	30.94	10.52	16.48	13.15	10.46	12.90	21.79	41.13	24.22	26.29	24.89	12.80	21.45
GraphBEV++	-	32.47	21.90	21.66	12.83	27.70	31.75	11.39	17.15	13.89	11.16	13.80	22.74	41.90	24.98	27.12	25.84	13.48	22.23
Performances on Noisy setting																			
GaussianFormer-T[53]	ECCV'24	27.86	17.63	17.42	8.38	22.91	27.48	7.56	12.03	9.81	6.87	8.72	18.02	37.18	20.44	22.67	21.18	9.22	17.74
GraphBEV++	-	29.41	19.37	18.74	10.43	24.62	28.39	8.71	13.92	11.28	8.74	10.61	19.58	38.71	21.56	24.03	22.77	10.92	19.31

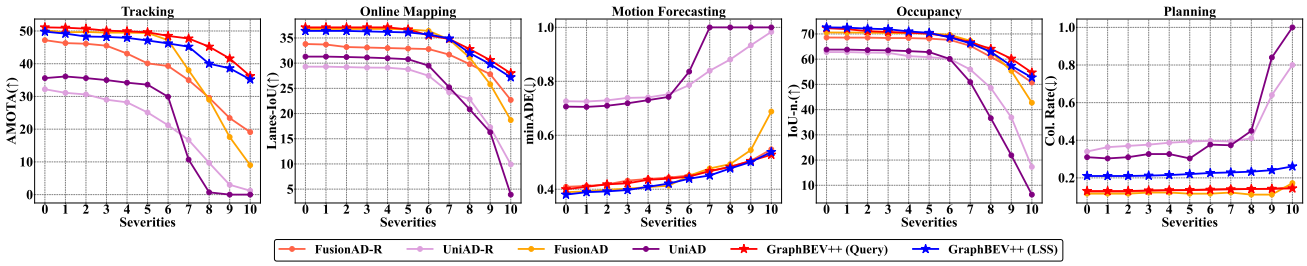

Fig. 7 End-to-end model performance with respect to the severity of misalignment noise in autonomous driving tasks. In autonomous driving tasks, we followed the alignment noise augmentation method by Dong et al. [28], introducing noise of varying severities to the projection matrices used in both the multi-modal approach, FusionAD [175], and the vision-based approach, UniAD [49]. The severity scale ranges from 0, indicating no noise (Clean), to 1 for slight noise, with increasing levels up to 10 for severe noise. As the noise level increased, we observed a significant misalignment impact on UniAD [49], followed by FusionAD [175]. Our proposed multi-modal end-to-end autonomous driving framework, GraphBEV++ (LSS) and GraphBEV++ (Query), leverages the advantage of neighbor alignment and performs better under misaligned conditions. Note that 'FusionAD-R' and 'UniAD-R' denote the variants of FusionAD and UniAD, respectively, trained with the RoarNet-based data augmentation strategy [127].

Table 9 The results of **multi-object tracking** in end-to-end autonomous driving (E2E-AD) tasks on nuScenes[6] dataset.

Method	Venue	AMOTA (%) ↑	AMOTP (m) ↓
ViP3D [40]	CVPR'23	21.7	1.625
QD3DT [46]	TPAMI'22	24.2	1.518
MUTR3D [188]	CVPR'22	29.4	1.498
DEFT [11]	Arxiv'21	20.1	-
DQTrack [88]	ICCV'23	36.7	1.351
STAR-Track [27]	RAI'23	37.9	1.358
CC-3DT [36]	Arxiv'22	42.9	1.257
PF-Track [116]	CVPR'23	40.8	1.343
SparseDrive [143]	ICRA'25	38.6	1.254
BridgeAD [182]	CVPR'25	39.8	1.232
MomAD[130]	CVPR'25	39.1	1.243
UC-Track [161]	TITS'26	43.8	1.290
UniAD [49]	CVPR'23	35.9	1.320
FusionAD [175]	Arxiv'23	50.1	1.065
GraphBEV++ (LSS)	-	49.8	1.082
GraphBEV++ (Query)	-	51.1	1.022

Table 10 The results of **online mapping** in end-to-end autonomous driving tasks on nuScenes[6] dataset.

Method	Venue	IoU-Lanes (%) ↑	IoU-Drivable (%) ↑
VPN [115]	RAL'20	18.0	76.0
LSS [120]	ECCV'20	18.3	73.9
BEVFormer [91]	ECCV'22	23.9	77.5
FusionAD [175]	Arxiv'23	36.7	73.1
UniAD [49]	CVPR'23	31.3	69.1
ParaDrive[158]	CVPR'24	71.0	33.0
DriveTransformer[61]	ICLR'25	77.0	39.0
GraphBEV++ (LSS)	-	36.4	73.9
GraphBEV++ (Query)	-	37.1	73.4

The strong performance on the Argoverse2 benchmark further validates the generalization ability of GraphBEV++ under large-scale and long-tail autonomous driving scenarios.

Table 11 The results of **motion forecasting** in end-to-end autonomous driving tasks.

Method	Venue	minADE (m) ↓	minFDE (m) ↓	MR (%) ↓	EPA (%) ↑
UniAD [49]	CVPR'23	0.71	1.02	15.1	45.6
VAD [62]	ICCV'23	0.68	0.88	8.3	-
FusionAD [175]	Arxiv'23	0.39	0.62	8.6	62.6
SparseDrive[142]	ICRA'25	0.62	0.99	13.6	48.2
SparseWorld	Arxiv'26	0.78	1.03	11.3	61.9
MomAD[130]	CVPR'25	0.61	0.98	13.7	49.9
CoT-Drive[96]	TAI'25	1.56	3.49	52.0	
GraphBEV++ (LSS)	-	0.40	0.59	8.5	**64.7**
GraphBEV++ (Query)	-	**0.38**	**0.52**	**7.7**	64.5

Table 12 The results of **planning** in end-to-end autonomous driving tasks on nuScenes [6] dataset. As Ref. [93] states, we **deactivate** the **ego status** information for a fair comparison.

Method	Venue	L2 (m)↓				Col. Rate (%)↓			
		1s	2s	3s	Avg.	1s	2s	3s	Avg.
NMP [181]	CVPR'19	-	-	2.31	-	-	-	1.92	-
SA-NMP [181]	CVPR'19	-	-	2.05	-	-	-	1.59	-
FF [47]	CVPR'21	0.55	1.20	2.54	1.43	0.66	0.17	1.07	0.43
EO [69]	ECCV'22	0.67	1.36	2.78	1.60	0.04	0.09	0.88	0.33
ST-P3 [48]	ECCV'22	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71
GPT-Driver [111]	ArXiv'23	0.27	0.74	1.52	0.84	0.07	0.15	1.10	0.44
UniAD [49]	CVPR'23	0.48	0.96	1.65	1.03	0.05	0.17	0.71	0.31
FusionAD [175]	ArXiv'23	0.02	0.08	0.27	0.81	0.05	0.17	0.71	0.12
VAD [62]	ICCV'23	0.41	0.70	1.05	0.72	0.11	0.24	0.42	0.26
DiffusionDrive [95]	CVPR'25	0.29	0.58	0.96	0.61	0.02	0.05	0.22	0.09
DIVER [134]	Arxiv'26	-	-	-	-	0.01	0.05	0.15	0.07
FocalAD [140]	Aut. Inno.'26	0.27	0.57	0.96	0.60	0.00	0.04	0.24	0.09
GuideFlow [101]	CVPR'26	-	-	-	-	0.00	0.02	0.18	0.07
SparseDrive [142]	ICRA'25	0.30	0.58	0.96	0.61	0.01	0.05	0.23	0.10
LAW [83]	ICLR'25	0.26	0.57	1.01	0.61	0.14	0.21	0.54	0.30
GenAD [192]	ECCV'24	0.28	0.49	0.78	0.52	0.08	0.14	0.34	0.19
Drive-OccWorld [191]	ECCV'24	0.25	0.44	0.72	0.47	0.03	0.08	0.22	0.11
SSR [77]	ICLR'25	0.19	0.36	0.62	0.39	0.10	0.10	0.24	0.15
MomAD [130]	CVPR'25	0.31	0.57	0.91	0.60	0.01	0.05	0.22	0.09
DriveWorld-VLA [57]	ICML'26	0.28	0.58	0.99	0.61	0.00	0.10	0.38	0.16
GraphBEV++ (LSS)	-	0.33	0.66	1.02	0.67	0.06	0.21	0.36	0.21
GraphBEV++ (Query)	-	0.40	0.67	1.04	0.70	0.03	0.07	0.29	0.13

Table 13 Open-loop and Closed-loop results on **Bench2Drive** [60] (V0.0.3) under base training set. * denotes expert feature distillation. † denotes the re-implementation. 'mmt' refers to the multi-mode trajectory variant.

Method	Venue	Open-loop Metric	Closed-loop Metric			
		Avg. L2 ↓	DS ↑	SR (%) ↑	Effi ↑	Comf ↑
ThinkTwice* [59]	CVPR'23	0.95	62.44	31.23	69.33	16.22
DriveAdapter* [58]	ICCV'23	1.01	64.22	33.08	70.22	16.01
VAD [62]	ICCV'23	0.91	42.35	15.00	157.94	46.01
DriveDPO [124]	NeurIPS'25	-	62.02	30.62	-	-
Raw2Drive [172]	NeurIPS'25	-	71.36	50.24	-	-
DriveTrans* [61]	ICLR'25	0.62	63.46	35.01	100.64	20.78
MomAD [130]	CVPR'25	0.85	45.35	17.44	162.09	49.34
WoTE* [85]	ICCV'25	-	61.71	31.36	-	-
ThinkTwice$_{mmt}$*† [59]	CVPR'23	0.93	63.34	33.23	71.56	18.32
GraphBEV++ (Query)	-	0.82	45.71	17.92	162.82	50.14

4.4.2 BEV Map Segmentation

To evaluate 3D object detection, we also assess the generalization capability in the BEV Map Segmentation (semantic segmentation) tasks on the nuScenes validation set, as shown in Table 7. Following the same training strategy as the baseline BEVFusion, we have conducted evaluations within the [-50m, 50m]×[-50m, 50m] region around an ego car for each frame. We then report Intersection-over-Union (IoU) scores for drivable area, pedestrian crossing, walkway, stop line, car park, and divider. Significant improvements are observed for drivable area, pedestrian crossing, walkway, stop line, and car park, with only a minor decrease for divider. Overall, our GraphBEV++ demonstrates not only significant performance in 3D object detection but also strong generalization capability in BEV Map Segmentation.

Table 14 Comparison on planning-oriented NAVSIM [25] navtest split with **Closed-Loop** metrics.

Method	Venue	Input	NC↑	DAC↑	TTC↑	Comf.↑	EP↑	PDMS↑
UniAD [49]	CVPR'23	C	97.8	91.9	92.9	100	78.8	83.4
LTF [20]	TPAMI'22	C	97.4	92.8	92.4	100	79.0	83.8
PARA-Drive [158]	CVPR'24	C	97.9	92.4	93.0	99.8	79.3	84.0
VADv2 [14]	ICLR'26	LC	97.2	89.1	91.6	100	76.0	80.9
Hydra-MDP [89]	Arxiv'24	LC	98.3	96.0	94.6	100	78.7	86.5
DiffusionDrive [95]	CVPR'25	LC	98.2	96.2	94.7	100	82.2	88.1
WoTE [85]	ICCV'25	LC	98.5	96.8	81.9	94.9	99.9	88.3
GraphBEV++ (Query)	-	LC	98.7	97.1	82.0	95.1	99.9	88.7

Table 15 Efficiency comparison with SOTA multi-modal 3D detection methods. FPS is reported or estimated on an A100 GPU.

Method	Modality	Model Size	FLOPs	FPS
RoboFusion-L [138]	LC	97.54M	1.23T	3.1
DeepInteraction [171]	LC	57.82M	775.2G	4.9
TransFusion [1]	LC	36.96M	612.6G	6.0
BEVFusion [108]	LC	40.81M	506.4G	7.5
GraphBEV [137]	LC	47.80M	535.7G	7.2
BEVFormer-S-C [92]	LC	68.70M	1303.5.7G	5.3
GraphBEV++ (LSS)	LC	48.12M	457.6G	7.1
GraphBEV++ (Query)	LC	69.76M	1357.6G	4.9

Table 16 Roles of different modules in GraphBEV++ (LSS) for feature alignment on nuScenes (clean) and nuScenes-C (noisy) dataset. Green denotes the relative performance drop from the clean to the noisy settings. '+L (LSS)' indicates the addition of only the LocalAlign-v2 (LSS) module, and '+G (Deformable)' indicates only the GlobalAlign-v2 (Deformable) module. GraphBEV++ (LSS) denotes the addition of both LocalAlign-v2 (LSS) and GlobalAlign-v2 (Deformable) modules. All latency measurements are conducted on the same workstation with an A100 GPU.

	Method	mAP	NDS	Latency(ms)
Clean	TransFusion [1]	67.3	71.2	164.6
	Baseline [108]	68.5	71.4	133.2
	+L (LSS)	69.8 *+1.3*	72.6 *+1.2*	135.1 *+2.9*
	+G (Deformable)	68.9 *+0.4*	71.7 *+0.3*	138.1 *+4.9*
	GraphBEV++ (LSS)	70.7 *+1.6*	73.2 *+1.5*	140.9 *+7.7*
Noisy	TransFusion [1]	66.4 *-1.3%*	70.6 *-0.8%*	164.6
	Baseline [108]	60.8 *-11.2%*	65.7 *-8.2%*	132.9
	+L (LSS)	67.4 *-3.4%* *+6.6*	70.4 *-3.0%* *+4.7*	135.8 *+2.9*
	+G (Deformable)	63.1 *-9.6%* *+2.3*	67.2 *-6.3%* *+1.5*	137.9 *+5.0*
	GraphBEV++ (LSS)	69.3 *-2.0%* *+8.5*	72.3 *-1.2%* *+6.6*	141.2 *+8.3*

Table 17 Roles of different modules in GraphBEV++ (Query) for feature alignment on nuScenes-C (noisy) dataset. '+L (Query)' indicates the addition of only the LocalAlign-v2 (Query) module, and '+G (Diffusion)' indicates only the GlobalAlign-v2 (Diffusion) module. GraphBEV++ (Query) denotes the addition of both LocalAlign-v2 (LSS) and GlobalAlign-v2 (Diffusion) modules. All latency measurements are conducted on the same workstation with an A100 GPU.

Method	mAP	NDS
BEVFormer-M [92]	63.2	66.3
+L (Query)	66.3 *+3.1*	69.0 *+2.7*
+G (Diffusion)	66.2 *+3.0*	68.1 *+1.8*
GraphBEV++ (Query)	69.1 *+5.9*	71.2 *+4.9*

Table 18 Effect of diffusion steps T on accuracy and inference speed under the nuScenes-C noisy setting.

T	mAP	NDS	FPS
1	67.8	69.6	5.21
2	68.5	70.3	5.06
4	69.1	71.2	4.90
8	69.2	71.3	4.57

Table 19 Evaluation of depth prediction on the **nuScenes** validation set. Following BEVDepth [84], 'soft' and 'hard' refer to Gaussian and one-hot depth randomization, respectively. 'learned' denotes GraphBEV++ (LSS).

D^{pred}	mAP↑	mATE↓	NDS↑
random soft	46.1	68.3	48.9
random hard	41.3	74.6	43.2
learned	70.7	27.6	73.2

4.4.3 3D Occupancy Prediction

To further validate the generality of GraphBEV++, we extend our feature alignment framework to the 3D semantic occupancy prediction task by integrating the GraphBEV++ (Query) variant into GaussianFormer-T. As shown in Table 8, GraphBEV++ consistently outperforms the baseline under both clean and noisy settings. Specifically, GraphBEV++ improves IoU/mIoU from 31.34/20.42 to 32.47/21.90 under the clean setting and from 27.86/17.63 to 29.41/19.37 under the noisy setting. Moreover, the performance degradation from clean to noisy conditions is reduced from 3.48 to 3.06 IoU and from 2.79 to 2.53 mIoU. These results demonstrate that the proposed LocalAlign-v2 and GlobalAlign-v2 modules effectively improve multi-view feature alignment and can generalize beyond 3D object detection to robust 3D occupancy prediction.

4.4.4 End-to-End Autonomous Driving

To evaluate end-to-end autonomous driving, we follow UniAD [49], reporting the performance of each task (perception, prediction, and planning) sequentially. We compare the performance of our GraphBEV++ (Query) and GraphBEV++ (LSS) with SOTA methods on the nuScenes validation set.

Perception Results. For multi-object tracking in Table 9, our GraphBEV++ (Query) achieves SOTA performance compared to end-to-end autonomous driving methods like UniAD [49], SparseDrive [142], and FusionAD [175]. As a multi-modal end-to-end approach, our GraphBEV++ (Query) achieves 51.1% AMOTA(%)↑ and 1.022m AMOTP(m)↓, significantly surpassing the baseline UniAD (35.9 % AMOTA and 1.320m AMOTP) as well as SparseDrive (38.6% AMOTA and 1.254m AMOTP) and FusionAD (50.1% AMOTA and 1.065m AMOTP). For online mapping in Table 10, our GraphBEV++ (Query) performs well on segmenting lanes (**+5.8 IoU(%)** compared to UniAD [49]), which is crucial for downstream agentroad interaction in the motion module. It is noteworthy that another version of our method, GraphBEV++ (LSS), performs similarly to GraphBEV++ (Query) on the aforementioned tasks. We have shown significant performance improvements in the perception module, attributed to our effective utilization of LiDAR, which offers new prospects for multi-modal end-to-end systems.

Prediction Results. As shown in Table 11, GraphBEV++ achieves the best overall motion forecasting performance. GraphBEV++ (Query) obtains the lowest minADE (0.38,m), minFDE (0.52,m), and MR (7.7%), outperforming recent methods such as SparseDrive and MomAD. In addition, GraphBEV++ achieves the highest EPA score of 64.7%, indicating superior trajectory quality and prediction reliability. Compared with GraphBEV++ (LSS), the query-based variant further improves forecasting accuracy, demonstrating the effectiveness of the proposed alignment strategy for query-based BEV representations. These results verify that accurate multi-modal feature alignment benefits not only perception but also downstream motion forecasting.

Planning Results (nuScenes). In the planning results shown in Table 12, our GraphBEV++ (Query) and GraphBEV++ (LSS) achieve superior performance compared to end-to-end autonomous driving methods such as ST-P3 [48], GPT-Driver [111], VAD [62], and UniAD [49]. GraphBEV++ (Query) reaches SOTA results in collision rate, while GraphBEV++ (LSS) achieves SOTA results in L2 error. By leveraging multi-modal information from LiDAR and camera data, our

Table 20 The performance of end-to-end methods (UniAD [49], FusionAD [175], GraphBEV++) under nuScene (clean) and nuScenes-C (misalignment) conditions. "↑" indicates better performance with higher metrics, while "↓" indicates better performance with lower metrics.

Method	Setting	Tracking		Mapping		Motion Forecasting			Occupancy				Planning	
		AMOTA ↑	AMOTP ↓	IoU-Lanes ↑	IoU-Drivable ↑	minADE ↓	minFDE ↓	MR ↓	IoU-n. ↑	IoU-f. ↑	VPQ-n. ↑	VPQ-f. ↑	avg.L2↓	avg.Col↓
FusionAD [175]	Clean	50.2	1.059	36.8	73.2	0.38	0.61	8.4	70.5	51.0	64.9	50.3	0.70	0.11
	Noisy	38.8	1.245	33.0	68.3	0.46	0.72	9.5	65.5	42.6	60.2	41.7	0.77	0.12
UniAD [49]	Clean	35.6	1.343	31.3	69.1	0.70	1.01	14.9	55.2	34.2	63.8	40.5	1.04	0.29
	Noisy	21.5	1.566	25.1	61.6	0.84	1.32	26.1	49.2	26.4	38.1	21.5	1.23	0.48
GraphBEV++ (LSS)	Clean	49.8	1.082	34.1	73.9	0.40	0.59	8.5	72.6	52.4	66.5	51.4	0.67	0.21
	Noisy	44.5	1.102	31.4	69.1	0.44	0.63	8.8	66.9	48.1	62.5	48.2	0.73	0.22
GraphBEV++ (Query)	Clean	51.1	1.022	37.1	73.4	0.38	0.52	7.7	72.3	52.2	66.1	51.5	0.70	0.13
	Noisy	47.0	1.124	34.6	69.6	0.45	0.57	8.4	66.5	47.8	63.2	47.9	0.76	0.14

Table 21 Effect of the hyperparameters K_{graph} for feature misalignment. We analyze the effect of hyperparameter K_{graph} in LocalAlign-v2 module for feature alignment under noisy misalignment settings on the nuScenes validation set. 'LT(ms)' represents latency. All latency measurements are conducted on the same workstation with an A100 GPU.

Baseline [108]			$K_{\text{graph}} = 5$			$K_{\text{graph}} = 8$			$K_{\text{graph}} = 12$			$K_{\text{graph}} = 16$			$K_{\text{graph}} = 25$		
mAP	NDS	LT	mAP	NDS	LT	mAP	NDS	LT	mAP	NDS	LT	mAP	NDS	LT	mAP	NDS	LT
60.8	65.7	**132.9**	67.1	70.9	138.2	**70.1**	**72.9**	140.9	69.8	72.2	143.4	68.8	70.5	145.3	67.1	69.9	150.0

Table 22 Unified ablation study of LocalAlign-v2 (Query) and GlobalAlign-v2 (Diffusion) on UniAD [49] across multiple autonomous driving tasks, including tracking, mapping, motion forecasting, occupancy prediction, and planning. "↑" indicates better performance with higher metrics, while "↓" indicates better performance with lower metrics.

LocalAlign-v2 (Query)	GlobalAlign-v2 (Diffusion)	Tracking		Mapping		Motion Forecasting			Occupancy				Planning	
		AMOTA ↑	AMOTP ↓	IoU-Lanes ↑	IoU-Drivable ↑	minADE ↓	minFDE ↓	MR ↓	IoU-n. ↑	IoU-f. ↑	VPQ-n. ↑	VPQ-f. ↑	avg.L2↓	avg.Col↓
		21.5	1.566	25.1	61.6	0.84	1.32	26.1	49.2	26.4	38.1	21.5	1.23	0.48
✓		40.3	1.208	31.4	69.1	0.56	0.81	11.7	61.4	41.5	63.1	44.3	0.87	0.26
	✓	31.9	1.337	28.7	65.9	0.63	0.93	15.4	58.1	37.6	54.9	37.2	0.95	0.33
✓	✓	44.5	1.102	31.4	69.1	0.44	0.63	8.8	66.9	48.1	62.5	48.2	0.73	0.22

GraphBEV++ (Query) reduces the planning L2 error and collision rate by 32.0% and 58.1%, respectively, compared to UniAD [49], based on average values for the planning horizon.

Planning Results (Bench2Drive). As shown in Table 13, we conduct experiments on the Bench2Drive closed-loop benchmark, using the MomAD (VAD version) as the baseline. Our GraphBEV++ achieves consistent improvements on this benchmark, demonstrating its potential for deployment in closed-loop autonomous driving systems. It is worth noting that the feature alignment in this closed-loop setting is ideal; this experiment primarily serves to validate the practical applicability of our method.

Planning Results (NAVSIM). As shown in Table 14, we conduct end-to-end evaluations on the closed-loop NAVSIM benchmark, using WoTE [85] as the baseline. Compared methods include UniAD, LTF, PARA-Drive, VADv2, Hydra-MDP, and DiffusionDrive. Notably, WoTE achieves a PDMS score of 88.3, while our GraphBEV++ (Query) reaches 88.7, indicating that our method also brings performance gains on NAVSIM. This demonstrates the effectiveness and generalizability of GraphBEV++ in closed-loop end-to-end driving scenarios.

4.5 Ablation Study

4.5.1 Efficiency Analysis of GraphBEV++

Table 15 compares the efficiency of GraphBEV++ with representative multi-modal perception methods. GraphBEV++ (LSS) achieves a favorable accuracy–efficiency trade-off, reducing FLOPs from 535.7G to 457.6G compared with GraphBEV while maintaining a similar inference speed (7.2 FPS vs. 7.1 FPS). Moreover, it is substantially more efficient than RoboFusion-L and DeepInteraction. These results demonstrate that the proposed alignment modules improve robustness and feature alignment quality with only marginal computational overhead.

4.5.2 Roles of Different Modules in GraphBEV++

To analyze the impact of misalignment, we conduct comparative experiments between our GraphBEV++ (LSS) and BEVFusion [108]. It is noteworthy that in Table 16, we introduce misalignment into the nuScenes validation set, rather than in the training and testing sets, following Ref. [28]. We train on the clean nuScenes [6] training dataset and evaluate performance under both clean and noisy misalignment conditions.

Table 23 Robustness to weather conditions, different ego distances, different sizes on nuScenes [6] clean validation set. It is important to note that the evaluation metric is mAP (%).

Method	Different Weather Conditions				Different Ego Distances			Different Object Sizes		
	Sunny	Rainy	Day	Night	Near	Middle	Far	Small	Moderate	Large
Baseline [108]	68.2	69.9	68.5	42.8	79.4	64.9	40.0	50.3	58.7	64.0
GraphBEV++ (LSS)	70.7	70.4	69.8	45.2	79.9	65.6	42.2	55.5	59.5	64.6
	+2.5	*+0.5*	*+1.3*	*+2.4*	*+0.5*	*+0.7*	*+2.2*	*+5.2*	*+0.8*	*+0.6*

In the clean setting, GraphBEV++ (LSS) outperforms BEVFusion [108] significantly, while in the noisy setting, the performance improvement is substantial. Furthermore, it is evident that BEVFusion [108] exhibits a significant decrease in metrics such as mAP and NDS when transitioning from clean to noisy conditions, whereas GraphBEV++ (LSS) demonstrates a small decrease in performance metrics. Notably, adding the LocalAlign-v2 (LSS) or GlobalAlign-v2 (Deformable) module to BEVFusion [108] has minimal impact on latency compared to BEVFusion [108] alone, and the latency is lower than that of TransFusion [1]. When only the LocalAlign-v2 (LSS) module is added to BEVFusion [108] and the KD-Tree algorithm is used to build proximity relationships, significant enhancements are observed in both clean and noisy misalignment settings by fusing projected depth with neighbor depth to prevent feature misalignment. Adding only the GlobalAlign-v2 (Deformable) module to BEVFusion [108] also leads to noticeable improvements. Particularly, the simultaneous addition of the LocalAlign-v2 (LSS) and GlobalAlign-v2 (Deformable) modules exhibits strong performance in both clean and noisy settings.

We further evaluate the effectiveness of **LocalAlign-v2 (Query)** and **GlobalAlign-v2 (Diffusion)** on the BEVFormer-M baseline. As shown in Table 17, the baseline achieves 63.2 mAP and 66.3 NDS on the nuScenes-C benchmark. Incorporating LocalAlign-v2 (Query) improves performance by 3.1% mAP and 2.7% NDS, while GlobalAlign-v2 (Diffusion) leads to a gain of 3.0% mAP and 1.8% NDS. The full GraphBEV++ (Query) model achieves 69.1 mAP and 71.2 NDS, outperforming the baseline by 5.9% and 4.9% respectively. These results validate the effectiveness of our method in addressing feature misalignment in query-based BEV frameworks.

Table 18 presents the trade-off between diffusion steps, detection performance, and inference speed. Increasing T consistently improves alignment quality, with mAP rising from 67.8 to 69.1 and NDS from 69.6 to 71.2 when T increases from 1 to 4. Nevertheless, the performance gain becomes saturated beyond $T = 4$, while the inference speed decreases from 4.90 FPS to 4.57 FPS. Therefore, a lightweight diffusion configuration with $T = 4$ is adopted throughout the paper, providing an effective balance between alignment performance and runtime efficiency.

4.5.3 Effect of Depth Noise on GraphBEV++

As shown in Table 19, we follow the noise injection protocol of BEVDepth [84], applying both Gaussian noise and one-hot random noise (which directly replaces depth values) to simulate degraded depth inputs. These perturbations significantly impact our method, as GraphBEV++ relies heavily on accurate depth information. Our approach is specifically designed to address the misalignment issues caused by inaccurate point cloud projection in BEVDepth. The variant labeled as *learned* refers to our GraphBEV++ framework, in which depth information is not only derived from projected LiDAR points but also encoded through a learnable depth feature representation. Overall, our method demonstrates strong robustness against depth noise.

4.5.4 Effect of GraphBEV++ in End-to-End Autonomous Driving for Feature Misalignment

As shown in Table 20, we analyze the robustness of GraphBEV++ (LSS), GraphBEV++ (Query), UniAD [49], and FusionAD [175] in end-to-end tasks under varying severity levels of misalignment conditions. The misalignment issue directly impacts the performance of models in various end-to-end autonomous driving tasks (Tracking, Mapping, Motion Forecasting, Occupancy and Planning), which highlights the importance of addressing misalignment to maintain performance in end-to-end autonomous driving tasks. Meanwhile, compared to mono modal method UniAD [49], multi-modal methods FusionAD and GraphBEV++ (Query) are more robust when facing the issues of misalignment. This is due to the effective utilization of the complementary information from different modalities in multi-modal methods. However, although multimodal methods are robust to misalignment issues, they are still affected by modal misalignment. Notably, by utilizing neighborhood information as guidance and correction for feature alignment, GraphBEV++ (Query) and GraphBEV++ (LSS) effectively achieve modal alignment from both local and global perspectives. This

enables GraphBEV++ to significantly outperform existing methods like UniAD [49] and FusionAD [175] under feature misalignment conditions. Noteworthy, by utilizing neighborhood information as a bridge for alignment between different modalities and achieving modality alignment from a global perspective, GraphBEV++ significantly outperforms the SOTA methods UniAD [49] and FusionAD [175] under varying levels of misalignment severity. Overall, our GraphBEV++ (LSS) and GraphBEV++ (Query) improve performance in misalignment scenarios through accurate feature alignment.

4.5.5 Effect of the Hyperparameters K_{graph} for Feature Misalignment

As shown in Table 21, to analyze the impact of the hyperparameter K_{graph} in the LocalAlign-v2 module on feature misalignment, we have studied its effects under noisy misalignment settings on the nuScenes validation set. K_{graph}, which is the number of nearest depths for LiDAR-to-camera projected depth in the LocalAlign-v2 module, influences the expressive capability of neighboring depth features. It is observed that our GraphBEV++ achieves optimal overall performance when K_{graph} is set to 8. Therefore, selecting an appropriate value for K_{graph} is crucial and may vary across different datasets. Furthermore, despite significant fluctuations in mAP resulting from changes in K_{graph}, the overall performance still surpasses that of BEVFusion.

4.5.6 Effect of LocalAlign-v2 and GlobalAlign-v2 Across Multiple Autonomous Driving Tasks

As shown in Table 22, we conduct a unified ablation study on UniAD to investigate the contributions of LocalAlign-v2 (Query) and GlobalAlign-v2 (Diffusion) across multiple autonomous driving tasks, including tracking, mapping, motion forecasting, occupancy prediction, and planning. It can be observed that both alignment modules consistently improve performance over the UniAD baseline. Specifically, introducing only LocalAlign-v2 yields substantial gains across all tasks, improving AMOTA from 21.5 to 40.3, IoU-Lanes from 25.1 to 31.4, and reducing the planning error (avg.L2) from 1.23 to 0.87. In comparison, GlobalAlign-v2 alone also provides consistent improvements, increasing AMOTA to 31.9 and reducing avg.L2 to 0.95. These results demonstrate that both local and global alignment errors negatively affect the entire autonomous driving pipeline. Furthermore, combining LocalAlign-v2 and GlobalAlign-v2 achieves the best overall performance on nearly all metrics. Compared with the UniAD baseline, the complete GraphBEV++ improves AMOTA by +23.0, IoU-f by +21.7, and VPQ-f by +26.7, while reducing minADE from 0.84 to 0.44, minFDE from 1.32 to 0.63, and avg.Col from 0.48 to 0.22. These results indicate that the two modules are highly complementary, where LocalAlign-v2 primarily addresses local feature correspondence while GlobalAlign-v2 further mitigates global feature misalignment. Their combination provides the most effective alignment solution and consistently benefits all downstream autonomous driving tasks.

4.6 Robustness Study

4.6.1 Robustness to Weather Conditions

Adverse weather amplifies sensor degradation and calibration uncertainty, thereby intensifying feature misalignment between LiDAR and camera modalities. Therefore, evaluating performance under different weather conditions serves as an indirect yet effective way to assess the model's ability to address feature misalignment. As shown in Table 23, we present a robustness analysis of our GraphBEV++ with respect to different weather conditions. Various weather conditions influence 3D object detection tasks. Following the approach of BEVFusion [108], we partition the scenes in the validation set into sunny, rainy, day, and night conditions. We outperform BEVFusion [108] under different weather conditions, especially in night scenes. Overall, our GraphBEV++ improves performance in sunny weather through accurate feature alignment and enhances performance in adverse weather conditions.

4.6.2 Robustness to Ego Distances and Object Sizes

As shown in Table 23, we analyze the impact of different ego distances and object sizes on the performance of GraphBEV++. We categorize annotation and prediction ego distances into three groups: Near (0-20m), Middle (20-30m), and Far (¿30m), and summarize the size distributions for each category, defining three equal-proportion size levels: Small, Moderate, and Large. It is evident that GraphBEV++ demonstrates significant performance improvements for distant and small objects. Compared to BEVFusion [108], our GraphBEV++ consistently enhances performance across all ego distances and object sizes, further narrowing the performance gaps. Overall, our GraphBEV++ exhibits great robustness to changes in ego distances and object sizes.

5 Conclusion

In this work, we propose GraphBEV++, a robust fusion framework designed to address the feature misalignment problem in autonomous driving. Our framework is applicable to both 3D object detection and end-to-end autonomous driving tasks. To tackle local feature misalignment caused by inaccurately projected depth from LiDAR, we introduce the LocalAlign-v2 module, which comes in two variants: LSS-based and Query-based. This design not only mitigates the misalignment issues inherent in LSS-dependent methods such as BEVFusion but also addresses the projection inaccuracies of 3D reference points in methods like BEVFormer, by incorporating neighbor-aware depth features via graph matching. To further handle global-level misalignment between LiDAR and camera BEV features during fusion, we propose the GlobalAlign-v2 module, featuring two variants: Deformable-based and Diffusion-based. These variants resolve global misalignment through learned spatial offsets and diffusion-based feature alignment, respectively. Overall, GraphBEV++ provides a unified and principled solution to both local and global misalignment, significantly enhancing the robustness of multi-modal BEV perception under real-world deployment scenarios.

Acknowledgments

This work was supported by the National Key R&D Program of China (2018AAA0100302) and the Fundamental Research Funds for the Central Universities (2024XKRC024).

Data Availability

The datasets generated and/or analyzed during this study are publicly available in the original repositories: nuScenes [6] https://www.nuscenes.org and Argoverse 2 [159] https://www.argoverse.org/av2.html.

References

1. Bai, X., Hu, Z., Zhu, X., Huang, Q., Chen, Y., Fu, H., Tai, C.L.: Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1090–1099 (2022)
2. Bi, J., Wei, H., Zhang, G., Yang, K., Song, Z.: Dyfusion: Cross-attention 3d object detection with dynamic fusion. IEEE Latin America Transactions **22**(2), 106–112 (2024)
3. Bijelic, M., Gruber, T., Mannan, F., Kraus, F., Ritter, W., Dietmayer, K., Heide, F.: Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11682–11692 (2020)
4. Borse, S., Klingner, M., Kumar, V.R., Cai, H., Almuzairee, A., Yogamani, S., Porikli, F.: X-align: Cross-modal cross-view alignment for bird's-eye-view segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3287–3297 (2023)
5. Brazil, G., Liu, X.: M3d-rpn: Monocular 3d region proposal network for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9287–9296 (2019)
6. 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: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 11621–11631 (2020)
7. Caesar, H., Kabzan, J., Tan, K.S., Fong, W.K., Wolff, E., Lang, A., Fletcher, L., Beijbom, O., Omari, S.: nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. arXiv preprint arXiv:2106.11810 (2021)
8. Cai, Q., Pan, Y., Yao, T., Ngo, C.W., Mei, T.: Objectfusion: Multi-modal 3d object detection with object-centric fusion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 18067–18076 (2023)
9. Cao, A.Q., De Charette, R.: Monoscene: Monocular 3d semantic scene completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3991–4001 (2022)
10. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision, pp. 213–229. Springer (2020)
11. Chaabane, M., Zhang, P., Beveridge, J.R., O'Hara, S.: Deft: Detection embeddings for tracking. arXiv preprint arXiv:2102.02267 (2021)
12. Chen, L., Wu, P., Chitta, K., Jaeger, B., Geiger, A., Li, H.: End-to-end autonomous driving: Challenges and frontiers. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
13. Chen, Q., Qi, X.: Residual graph convolutional network for bird's-eye-view semantic segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 3324–3331 (2024)
14. Chen, S., Jiang, B., Gao, H., Liao, B., Xu, Q., Zhang, Q., Huang, C., Liu, W., Wang, X.: Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. arXiv preprint arXiv:2402.13243 (2024)
15. Chen, Y., Liu, H., Wang, L.: Sparsebev: A fully sparse framework for multi-view 3d object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (2026)
16. Chen, Y., Liu, J., Zhang, X., Qi, X., Jia, J.: Largekernel3d: Scaling up kernels in 3d sparse cnns. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13488–13498 (2023)
17. Chen, Y., Liu, J., Zhang, X., Qi, X., Jia, J.: VoxelNeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking. In: Proceedings of the IEEE/CVF Con-

ference on Computer Vision and Pattern Recognition (2023)
18. Chen, Z., Li, Z., Zhang, S., Fang, L., Jiang, Q., Zhao, F.: Deformable feature aggregation for dynamic multi-modal 3d object detection. In: S. Avidan, G. Brostow, M. Cissé, G.M. Farinella, T. Hassner (eds.) Computer Vision – ECCV 2022, pp. 628–644. Springer Nature Switzerland, Cham (2022)
19. Chen, Z., Li, Z., Zhang, S., Fang, L., Jiang, Q., Zhao, F., Zhou, B., Zhao, H.: Autoalign: Pixel-instance feature aggregation for multi-modal 3d object detection. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (2022). DOI 10.24963/ijcai.2022/116
20. Chitta, K., Prakash, A., Jaeger, B., Yu, Z., Renz, K., Geiger, A.: Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(11), 12878–12895 (2023). DOI 10.1109/TPAMI.2022.3200245
21. Chu, X., Deng, J., You, G., Duan, Y., Li, H., Zhang, Y.: Racformer: Towards high-quality 3d object detection via query-based radar-camera fusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 17081–17091 (2025)
22. Chu, X., Deng, J., You, G., Duan, Y., Li, Y., Zhang, Y.: Rayformer: Improving query-based multi-camera 3d object detection via ray-centric strategies. In: Proceedings of the 32nd ACM International Conference on Multimedia, pp. 4620–4629 (2024)
23. Contributors, O.: Openscene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving (2023)
24. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp. 764–773 (2017)
25. Dauner, D., Hallgarten, M., Li, T., Weng, X., Huang, Z., Yang, Z., Li, H., Gilitschenski, I., Ivanovic, B., Pavone, M., et al.: Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. Advances in Neural Information Processing Systems **37**, 28706–28719 (2024)
26. Deng, J., Shi, S., Li, P., Zhou, W., Zhang, Y., Li, H.: Voxel r-cnn: Towards high performance voxel-based 3d object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 1201–1209 (2021)
27. Doll, S., Hanselmann, N., Schneider, L., Schulz, R., Enzweiler, M., Lensch, H.P.: Star-track: Latent motion models for end-to-end 3d object tracking with adaptive spatio-temporal appearance representations. IEEE Robotics and Automation Letters **9**(2), 1326–1333 (2023)
28. Dong, Y., Kang, C., Zhang, J., Zhu, Z., Wang, Y., Yang, X., Su, H., Wei, X., Zhu, J.: Benchmarking robustness of 3d object detection to common corruptions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1022–1032 (2023)
29. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: S. Levine, V. Vanhoucke, K. Goldberg (eds.) Proceedings of the 1st Annual Conference on Robot Learning, *Proceedings of Machine Learning Research*, vol. 78, pp. 1–16. PMLR (2017). URL https://proceedings.mlr.press/v78/dosovitskiy17a.html
30. Drews, F., Feng, D., Faion, F., Rosenbaum, L., Ulrich, M., Gläser, C.: Deepfusion: A robust and modular 3d object detector for lidars, cameras and radars. In: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 560–567. IEEE (2022)
31. Fan, L., Pang, Z., Zhang, T., Wang, Y.X., Zhao, H., Wang, F., Wang, N., Zhang, Z.: Embracing single stride 3d object detector with sparse transformer. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022). DOI 10.1109/cvpr52688.2022.00827
32. Fan, L., Wang, F., Wang, N., Zhang, Z.: Fsd v2: Improving fully sparse 3d object detection with virtual voxels. IEEE Transactions on Pattern Analysis and Machine Intelligence **47**(2), 1279–1292 (2024)
33. Fan, L., Wang, F., Wang, N., Zhang, Z.X.: Fully sparse 3d object detection. Advances in Neural Information Processing Systems **35**, 351–363 (2022)
34. Feng, D., Cao, Y., Rosenbaum, L., Timm, F., Dietmayer, K.: Leveraging uncertainties for deep multi-modal object detection in autonomous driving. In: 2020 IEEE Intelligent Vehicles Symposium (IV), pp. 877–884. IEEE (2020)
35. Feng, D., Haase-Schütz, C., Rosenbaum, L., Hertlein, H., Glaeser, C., Timm, F., Wiesbeck, W., Dietmayer, K.: Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. IEEE Transactions on Intelligent Transportation Systems **22**(3), 1341–1360 (2020)
36. Fischer, T., Yang, Y.H., Kumar, S., Sun, M., Yu, F.: Cc-3dt: Panoramic 3d object tracking via cross-camera fusion. arXiv preprint arXiv:2212.01247 (2022)
37. Frossard, D., Da Suo, S., Casas, S., Tu, J., Urtasun, R.: Strobe: Streaming object detection from lidar packets. In: Conference on Robot Learning, pp. 1174–1183. PMLR (2021)
38. Gan, W., Mo, N., Xu, H., Yokoya, N.: A comprehensive framework for 3d occupancy estimation in autonomous driving. IEEE Transactions on Intelligent Vehicles (2024)
39. Ge, C., Chen, J., Xie, E., Wang, Z., Hong, L., Lu, H., Li, Z., Luo, P.: Metabev: Solving sensor failures for 3d detection and map segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 8721–8731 (2023)
40. Gu, J., Hu, C., Zhang, T., Chen, X., Wang, Y., Wang, Y., Zhao, H.: Vip3d: End-to-end visual trajectory prediction via 3d agent queries. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5496–5506 (2023)
41. Hao, X., Diao, Y., Wei, M., Yang, Y., Hao, P., Yin, R., Zhang, H., Li, W., Zhao, S., Liu, Y.: Mapfusion: A novel bev feature fusion network for multi-modal map construction. Information Fusion **119**, 103018 (2025)
42. He, C., Li, R., Li, S., Zhang, L.: Voxel set transformer: A set-to-set approach to 3d object detection from point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8417–8427 (2022)
43. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp. 2961–2969 (2017)
44. 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)

45. Hu, A., Murez, Z., Mohan, N., Dudas, S., Hawke, J., Badrinarayanan, V., Cipolla, R., Kendall, A.: Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 15273–15282 (2021)
46. Hu, H.N., Yang, Y.H., Fischer, T., Darrell, T., Yu, F., Sun, M.: Monocular quasi-dense 3d object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(2), 1992–2008 (2022)
47. Hu, P., Huang, A., Dolan, J., Held, D., Ramanan, D.: Safe local motion planning with self-supervised freespace forecasting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12732–12741 (2021)
48. Hu, S., Chen, L., Wu, P., Li, H., Yan, J., Tao, D.: St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In: European Conference on Computer Vision, pp. 533–549. Springer (2022)
49. Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., Lu, L., Jia, X., Liu, Q., Dai, J., Qiao, Y., Li, H.: Planning-oriented autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 17853–17862 (2023)
50. Huang, T., Liu, Z., Chen, X., Bai, X.: Epnet: Enhancing point features with image semantics for 3d object detection. In: European Conference on Computer Vision, pp. 35–52. Springer (2020)
51. 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)
52. 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)
53. 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)
54. Ji, M., Zhang, S., Yang, J.: Ocrfdet: Object-centric radiance fields for multi-view 3d object detection in autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 24933–24942 (2025)
55. Ji, M., Zhang, S., Yang, J.: Depthfusion: Depth-aware hybrid feature fusion for lidar-camera 3d object detection. IEEE Transactions on Multimedia (2026)
56. Ji, Y., Chen, Z., Xie, E., Hong, L., Liu, X., Liu, Z., Lu, T., Li, Z., Luo, P.: Ddp: Diffusion model for dense visual prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 21741–21752 (2023)
57. Jia, F., Liu, L., Song, Z., Jia, C., Ye, H., Hao, X., Chen, L., et al.: Driveworld-vla: Unified latent-space world modeling with vision-language-action for autonomous driving. arXiv preprint arXiv:2602.06521 (2026)
58. Jia, X., Gao, Y., Chen, L., Yan, J., Liu, P.L., Li, H.: Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 7953–7963 (2023)
59. Jia, X., Wu, P., Chen, L., Xie, J., He, C., Yan, J., Li, H.: Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 21983–21994 (2023)
60. Jia, X., Yang, Z., Li, Q., Zhang, Z., Yan, J.: Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. Advances in Neural Information Processing Systems **37**, 819–844 (2024)
61. Jia, X., You, J., Zhang, Z., Yan, J.: Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. arXiv preprint arXiv:2503.07656 (2025)
62. Jiang, B., Chen, S., Xu, Q., Liao, B., Chen, J., Zhou, H., Zhang, Q., Liu, W., Huang, C., Wang, X.: Vad: Vectorized scene representation for efficient autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8340–8350 (2023)
63. Jiang, X., Li, S., Liu, Y., Wang, S., Jia, F., Wang, T., Han, L., Zhang, X.: Far3d: Expanding the horizon for surround-view 3d object detection. In: Proceedings of the AAAI conference on artificial intelligence, vol. 38, pp. 2561–2569 (2024)
64. Jiang, Y., Zhang, L., Miao, Z., Zhu, X., Gao, J., Hu, W., Jiang, Y.G.: Polarformer: Multi-camera 3d object detection with polar transformer. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 1042–1050 (2023)
65. Jiao, Y., Jie, Z., Chen, S., Chen, J., Ma, L., Jiang, Y.G.: Msmdfusion: Fusing lidar and camera at multiple scales with multi-depth seeds for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 21643–21652 (2023)
66. Jin, X., Su, H., Liu, K., Ma, C., Wu, W., Hui, F., Yan, J.: Unimamba: Unified spatial-channel representation learning with group-efficient mamba for lidar-based 3d object detection. In: Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 1407–1417 (2025)
67. Jin, X., Su, H., Ma, C., Liu, K., Wu, W., Hui, F., Yan, J.: Geoformer: Geometry point encoder for 3d object detection with graph-based transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 26879–26889 (2025)
68. Jing, W., Chen, X., Nie, S., Jiao, Z., Ma, J.: Hd-fusion: Hierarchical dynamic fusion of lidar-camera for robust 3-d object detection. IEEE Transactions on Industrial Informatics (2026)
69. Khurana, T., Hu, P., Dave, A., Ziglar, J., Held, D., Ramanan, D.: Differentiable raycasting for self-supervised occupancy forecasting. In: European Conference on Computer Vision, pp. 353–369. Springer (2022)
70. Kim, B., Woo, S., Heo, S., Kim, E.: Bridgeta: Bridging the representation gap in knowledge distillation via teacher assistant for bird's eye view map segmentation. arXiv preprint arXiv:2508.09599 (2025)
71. Kim, Y., Park, K., Kim, M., Kum, D., Choi, J.W.: 3d dual-fusion: Dual-domain dual-query camera-lidar fusion for 3d object detection. arXiv preprint arXiv:2211.13529 (2022)
72. Kim, Y., Shin, J., Kim, S., Lee, I.J., Choi, J.W., Kum, D.: Crn: Camera radar net for accurate, robust, efficient 3d perception. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 17615–17626 (2023)
73. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

74. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 12697–12705 (2019)
75. Lei, K., Chen, Z., Jia, S., Zhang, X.: Hvdetfusion: A simple and robust camera-radar fusion framework. arXiv preprint arXiv:2307.11323 (2023)
76. Li, F., Lu, J., Xu, Y., Wu, C., Xu, T., Xiang, Z., Niu, Y.: Refine3d: Scene-adaptive reference point refinement for sparse 3d object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 40, pp. 6073–6081 (2026)
77. Li, P., Cui, D.: Navigation-guided sparse scene representation for end-to-end autonomous driving. In: International Conference on Learning Representations, vol. 2025, pp. 29118–29134 (2025)
78. Li, S., Teng, F., Cao, Y., Yang, K., Li, Z., Wang, Y.: Nrseg: Noise-resilient learning for bev semantic segmentation via driving world models. IEEE Transactions on Image Processing (2026)
79. Li, X., Hu, Z., Xiao, X., Kong, B.: Look before you fuse: 2d-guided cross-modal alignment for robust 3d detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11589–11598 (2026)
80. Li, X., Ma, T., Hou, Y., Shi, B., Yang, Y., Liu, Y., Wu, X., Chen, Q., Li, Y., Qiao, Y., He, L.: Logonet: Towards accurate 3d object detection with local-to-global cross-modal fusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 17524–17534 (2023)
81. Li, X., Shi, B., Hou, Y., Wu, X., Ma, T., Li, Y., He, L.: Homogeneous multi-modal feature fusion and interaction for 3d object detection. In: European Conference on Computer Vision, pp. 691–707. Springer (2022)
82. 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. Proceedings of the AAAI Conference on Artificial Intelligence **37**(2), 1486–1494 (2023). DOI 10.1609/aaai.v37i2.25234. URL https://ojs.aaai.org/index.php/AAAI/article/view/25234
83. Li, Y., Fan, L., He, J., Wang, Y., Chen, Y., Zhang, Z., Tan, T.: Enhancing end-to-end autonomous driving with latent world model. In: International Conference on Learning Representations, vol. 2025, pp. 42942–42959 (2025)
84. 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)
85. Li, Y., Wang, Y., Liu, Y., He, J., Fan, L., Zhang, Z.: End-to-end driving with online trajectory evaluation via bev world model. arXiv preprint arXiv:2504.01941 (2025)
86. Li, Y., Yu, A.W., Meng, T., Caine, B., Ngiam, J., Peng, D., Shen, J., Lu, Y., Zhou, D., Le, Q.V., Yuille, A., Tan, M.: Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 17182–17191 (2022)
87. 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)
88. Li, Y., Yu, Z., Philion, J., Anandkumar, A., Fidler, S., Jia, J., Alvarez, J.: End-to-end 3d tracking with decoupled queries. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 18302–18311 (2023)
89. Li, Z., Li, K., Wang, S., Lan, S., Yu, Z., Ji, Y., Li, Z., Zhu, Z., Kautz, J., Wu, Z., et al.: Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. arXiv preprint arXiv:2406.06978 (2024)
90. Li, Z., Wang, F., Wang, N.: Lidar r-cnn: An efficient and universal 3d object detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7546–7555 (2021)
91. 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: European conference on computer vision, pp. 1–18. Springer (2022)
92. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Yu, Q., Dai, J.: Bevformer: Learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. IEEE Transactions on Pattern Analysis and Machine Intelligence **47**(3), 2020–2036 (2025). DOI 10.1109/TPAMI.2024.3515454
93. Li, Z., Yu, Z., Lan, S., Li, J., Kautz, J., Lu, T., Alvarez, J.M.: Is ego status all you need for open-loop end-to-end autonomous driving? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14864–14873 (2024)
94. Liang, T., Xie, H., Yu, K., Xia, Z., Lin, Z., Wang, Y., Tang, T., Wang, B., Tang, Z.: Bevfusion: A simple and robust lidar-camera fusion framework. Advances in Neural Information Processing Systems **35**, 10421–10434 (2022)
95. Liao, B., Chen, S., Yin, H., Jiang, B., Wang, C., Yan, S., Zhang, X., Li, X., Zhang, Y., Zhang, Q., et al.: Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. arXiv preprint arXiv:2411.15139 (2024)
96. Liao, H., Kong, H., Wang, B., Wang, C., Wang, K.Y., He, Z., Xu, C., Li, Z.: Cot-drive: Efficient motion forecasting for autonomous driving with llms and chain-of-thought prompting. IEEE Transactions on Artificial Intelligence (2), 625–641 (2025)
97. 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)
98. Lin, X., Lin, T., Pei, Z., Huang, L., Su, Z.: Sparse4d v2: Recurrent temporal fusion with sparse model. arXiv preprint arXiv:2305.14018 (2023)
99. Lin, Z., Liu, Z., Xia, Z., Wang, X., Wang, Y., Qi, S., Dong, Y., Dong, N., Zhang, L., Zhu, C.: Rcbevdet: Radar-camera fusion in bird's eye view for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14928–14937 (2024)
100. Liu, H., Chen, Y., Wang, H., Yang, Z., Li, T., Zeng, J., Chen, L., Li, H., Wang, L.: Fully sparse 3d occupancy prediction. In: European Conference on Computer Vision, pp. 54–71. Springer (2024)
101. Liu, L., Jia, C., Yu, G., Song, Z., Li, J., Jia, F., Wu, P., Hao, X., Luo, Y.: Guideflow: Constraint-guided flow

matching for planning in end-to-end autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3719–3728 (2026)
102. Liu, S., Cui, M., Li, B., Liang, Q., Hong, T., Huang, K., Shan, Y.: Fshnet: Fully sparse hybrid network for 3d object detection. In: Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 8900–8909 (2025)
103. Liu, Y., Wang, T., Zhang, X., Sun, J.: Petr: Position embedding transformation for multi-view 3d object detection. In: European Conference on Computer Vision, pp. 531–548. Springer (2022)
104. Liu, Y., Yan, J., Jia, F., Li, S., Gao, A., Wang, T., Zhang, X.: Petrv2: A unified framework for 3d perception from multi-camera images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3262–3272 (2023)
105. Liu, Z., Hou, J., Wang, X., Ye, X., Wang, J., Zhao, H., Bai, X.: Lion: Linear group rnn for 3d object detection in point clouds. Advances in Neural Information Processing Systems **37**, 13601–13626 (2024)
106. Liu, Z., Huang, T., Li, B., Chen, X., Wang, X., Bai, X.: Epnet++: Cascade bi-directional fusion for multi-modal 3d object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(7), 8324–8341 (2023). DOI 10.1109/TPAMI.2022.3228806
107. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 10012–10022 (2021)
108. Liu, Z., Tang, H., Amini, A., Yang, X., Mao, H., Rus, D.L., Han, S.: Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation pp. 2774–2781 (2023)
109. Liu, Z., Tang, H., Lin, Y., Han, S.: Point-voxel cnn for efficient 3d deep learning. Advances in Neural Information Processing Systems **32** (2019)
110. Lv, C., Li, Y., Yang, H., Wang, Y.: Gau-occ: Geometry-completed gaussians for multi-modal 3d occupancy prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14198–14207 (2026)
111. Mao, J., Qian, Y., Zhao, H., Wang, Y.: Gpt-driver: Learning to drive with gpt. arXiv preprint arXiv:2310.01415 (2023)
112. Mao, J., Xue, Y., Niu, M., Bai, H., Feng, J., Liang, X., Xu, H., Xu, C.: Voxel transformer for 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3164–3173 (2021)
113. Miao, Z., Chen, J., Pan, H., Zhang, R., Liu, K., Hao, P., Zhu, J., Wang, Y., Zhan, X.: Pvgnet: A bottom-up one-stage 3d object detector with integrated multi-level features. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021). DOI 10.1109/cvpr46437.2021.00329
114. Murez, Z., Van As, T., Bartolozzi, J., Sinha, A., Badrinarayanan, V., Rabinovich, A.: Atlas: End-to-end 3d scene reconstruction from posed images. In: European conference on computer vision, pp. 414–431. Springer (2020)
115. Pan, B., Sun, J., Leung, H.Y.T., Andonian, A., Zhou, B.: Cross-view semantic segmentation for sensing surroundings. IEEE Robotics and Automation Letters **5**(3), 4867–4873 (2020)
116. Pang, Z., Li, J., Tokmakov, P., Chen, D., Zagoruyko, S., Wang, Y.X.: Standing between past and future: Spatio-temporal modeling for multi-camera 3d multi-object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 17928–17938 (2023)
117. Park, J., Xu, C., Yang, S., Keutzer, K., Kitani, K.M., Tomizuka, M., Zhan, W.: Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. In: The Eleventh International Conference on Learning Representations (2023). URL https://openreview.net/forum?id=H3HcEJA2Um
118. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
119. Pfeuffer, A., Dietmayer, K.: Optimal sensor data fusion architecture for object detection in adverse weather conditions. In: 2018 21st International Conference on Information Fusion (FUSION), pp. 1–8. IEEE (2018)
120. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16, pp. 194–210. Springer (2020)
121. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3d object detection from rgb-d data. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 918–927 (2018)
122. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 652–660 (2017)
123. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems **30** (2017)
124. Shang, S., Chen, Y., Wang, Y., Li, Y., Zhang, Z.: Drivedpo: Policy learning via safety dpo for end-to-end autonomous driving. arXiv preprint arXiv:2509.17940 (2025)
125. Shi, S., Jiang, L., Deng, J., Wang, Z., Guo, C., Shi, J., Wang, X., Li, H.: Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. International Journal of Computer Vision **131**(2), 531–551 (2023)
126. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 770–779 (2019)
127. Shin, K., Kwon, Y.P., Tomizuka, M.: Roarnet: A robust 3d object detection based on region approximation refinement. In: 2019 IEEE intelligent vehicles symposium (IV), pp. 2510–2515. IEEE (2019)
128. Simonelli, A., Bulo, S.R., Porzi, L., López-Antequera, M., Kontschieder, P.: Disentangling monocular 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1991–1999 (2019)
129. Sindagi, V.A., Zhou, Y., Tuzel, O.: Mvx-net: Multimodal voxelnet for 3d object detection. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 7276–7282. IEEE (2019)
130. Song, Z., Jia, C., Liu, L., Pan, H., Zhang, Y., Wang, J., Zhang, X., Xu, S., Yang, L., Luo, Y.: Don't shake

the wheel: Momentum-aware planning in end-to-end autonomous driving. In: Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 22432–22441 (2025)
131. Song, Z., Jia, C., Yang, L., Wei, H., Liu, L.: Graphalign++: An accurate feature alignment by graph matching for multi-modal 3d object detection. IEEE Transactions on Circuits and Systems for Video Technology (2023)
132. Song, Z., Jia, F., Pan, H., Luo, Y., Jia, C., Zhang, G., Liu, L., Ji, Y., Yang, L., Wang, L.: Contrastalign: Toward robust bev feature alignment via contrastive learning for multi-modal 3d object detection. arXiv preprint arXiv:2405.16873 (2024)
133. Song, Z., Liu, L., Jia, F., Luo, Y., Jia, C., Zhang, G., Yang, L., Wang, L.: Robustness-aware 3d object detection in autonomous driving: A review and outlook. IEEE Transactions on Intelligent Transportation Systems pp. 1–30 (2024). DOI 10.1109/TITS.2024.3439557
134. Song, Z., Liu, L., Pan, H., Liao, B., Guo, M., Yang, L., Zhang, Y., Xu, S., Jia, C., Luo, Y.: Breaking imitation bottlenecks: Reinforced diffusion powers diverse trajectory generation. arXiv preprint arXiv:2507.04049 (2025)
135. Song, Z., Wei, H., Bai, L., Yang, L., Jia, C.: Graphalign: Enhancing accurate feature alignment by graph matching for multi-modal 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3358–3369 (2023)
136. Song, Z., Wei, H., Jia, C., Xia, Y., Li, X., Zhang, C.: Vp-net: Voxels as points for 3d object detection. IEEE Transactions on Geoscience and Remote Sensing (2023)
137. Song, Z., Yang, L., Xu, S., Liu, L., Xu, D., Jia, C., Jia, F., Wang, L.: Graphbev: Towards robust bev feature alignment for multi-modal 3d object detection. arXiv preprint arXiv:2403.11848 (2024)
138. Song, Z., Zhang, G., Liu, L., Yang, L., Xu, S., Jia, C., Jia, F., Wang, L.: Robofusion: Towards robust multi-modal 3d object detection via sam. arXiv preprint arXiv:2401.03907 (2024)
139. Song, Z., Zhang, G., Xie, J., Liu, L., Jia, C., Xu, S., Wang, Z.: Voxelnextfusion: A simple, unified, and effective voxel fusion framework for multimodal 3-d object detection. IEEE Transactions on Geoscience and Remote Sensing **61**, 1–12 (2023). DOI 10.1109/TGRS.2023.3331893
140. Sun, B., Zhang, B., Lu, J., Feng, X., Shang, J., Cao, R., Zheng, M., Wang, C., Yang, S., Cao, Y., et al.: Focalad: Local motion planning for end-to-end autonomous driving: B. sun et al. Automotive Innovation pp. 1–11 (2026)
141. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 2446–2454 (2020)
142. Sun, W., Lin, X., Shi, Y., Zhang, C., Wu, H., Zheng, S.: Sparsedrive: End-to-end autonomous driving via sparse scene representation. arXiv preprint arXiv:2405.19620 (2024)
143. Sun, W., Lin, X., Shi, Y., Zhang, C., Wu, H., Zheng, S.: Sparsedrive: End-to-end autonomous driving via sparse scene representation. In: 2025 IEEE International Conference on Robotics and Automation (ICRA), pp. 8795–8801. IEEE (2025)
144. Team, O.D.: Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet (2020)
145. Tian, X., Jiang, T., Yun, L., Mao, Y., Yang, H., Wang, Y., Wang, Y., Zhao, H.: Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. Advances in Neural Information Processing Systems **36**, 64318–64330 (2023)
146. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
147. Vora, S., Lang, A.H., Helou, B., Beijbom, O.: Pointpainting: Sequential fusion for 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 4604–4612 (2020)
148. Wang, C., Ma, C., Zhu, M., Yang, X.: Pointaugmenting: Cross-modal augmentation for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 11794–11803 (2021)
149. Wang, H., Tang, H., Shi, S., Li, A., Li, Z., Schiele, B., Wang, L.: Unitr: A unified and efficient multi-modal transformer for bird's-eye-view representation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 6792–6802 (2023)
150. Wang, L., Song, Z., Zhang, X., Wang, C., Zhang, G., Zhu, L., Li, J., Liu, H.: Sat-gcn: Self-attention graph convolutional network-based 3d object detection for autonomous driving. Knowledge-Based Systems **259**, 110080 (2023)
151. Wang, L., Zhang, X., Song, Z., Bi, J., Zhang, G., Wei, H., Tang, L., Yang, L., Li, J., Jia, C., et al.: Multi-modal 3d object detection in autonomous driving: A survey and taxonomy. IEEE Transactions on Intelligent Vehicles (2023)
152. 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)
153. Wang, T., Zhu, X., Pang, J., Lin, D.: Fcos3d: Fully convolutional one-stage monocular 3d object detection. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 913–922 (2021)
154. Wang, X., Zhu, Z., Xu, W., Zhang, Y., Wei, Y., Chi, X., Ye, Y., Du, D., Lu, J., Wang, X.: Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 17850–17859 (2023)
155. Wang, Y., Guizilini, V.C., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In: Conference on Robot Learning, pp. 180–191. PMLR (2022)
156. Wang, Z., Huang, Z., Gao, Y., Wang, N., Liu, S.: Mv2dfusion: Leveraging modality-specific object semantics for multi-modal 3d detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025)
157. 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)
158. Weng, X., Ivanovic, B., Wang, Y., Wang, Y., Pavone, M.: Para-drive: Parallelized architecture for real-time autonomous driving. In: Proceedings of the IEEE/CVF

Conference on Computer Vision and Pattern Recognition, pp. 15449–15458 (2024)
159. Wilson, B., Qi, W., Agarwal, T., Lambert, J., Singh, J., Khandelwal, S., Pan, B., Kumar, R., Hartnett, A., Pontes, J.K., Ramanan, D., Carr, P., Hays, J.: Argoverse 2: Next Generation Datasets for Self-Driving Perception and Forecasting. In: Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021) (2021)
160. Wolters, P., Gilg, J., Teepe, T., Herzog, F., Laouichi, A., Hofmann, M., Rigoll, G.: Unleashing hydra: Hybrid fusion, depth consistency and radar for unified 3d perception. In: 2025 IEEE International Conference on Robotics and Automation (ICRA), pp. 7467–7474. IEEE (2025)
161. Wu, D., Peng, J., Yu, S., Xu, K., Chen, Z., Ma, C.: Uctrack: Uncertainty-aware and task-coupled 3-d multi-object tracking. IEEE Transactions on Intelligent Transportation Systems (2026)
162. Xie, E., Yu, Z., Zhou, D., Philion, J., Anandkumar, A., Fidler, S., Luo, P., Alvarez, J.M.: M2bev: Multi-camera joint 3d detection and segmentation with unified birds-eye view representation. arXiv preprint arXiv:2204.05088 (2022)
163. Xie, Y., Xu, C., Rakotosaona, M.J., Rim, P., Tombari, F., Keutzer, K., Tomizuka, M., Zhan, W.: Sparsefusion: Fusing multi-modal sparse representations for multi-sensor 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 17591–17602 (2023)
164. Xu, B., Chen, Z.: Multi-level fusion based 3d object detection from monocular images. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2345–2353 (2018)
165. Xu, S., Jiang, S., Li, F., Liu, L., Song, Z., Yang, B., Yang, Z.x.: Sparseinteraction: Sparse semantic guidance for radar and camera 3d object detection. In: Proceedings of the 32nd ACM International Conference on Multimedia, pp. 9224–9233 (2024)
166. Yan, J., Liu, Y., Sun, J., Jia, F., Li, S., Wang, T., Zhang, X.: Cross modal transformer: Towards fast and robust 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 18268–18278 (2023)
167. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
168. Yang, C., Chen, Y., Tian, H., Tao, C., Zhu, X., Zhang, Z., Huang, G., Li, H., Qiao, Y., Lu, L., et al.: Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 17830–17839 (2023)
169. Yang, L., Tang, T., Li, J., Chen, P., Yuan, K., Wang, L., Huang, Y., Zhang, X., Yu, K.: Bevheight++: Toward robust visual centric 3d object detection. arXiv preprint arXiv:2309.16179 (2023)
170. Yang, L., Yu, K., Tang, T., Li, J., Yuan, K., Wang, L., Zhang, X., Chen, P.: Bevheight: A robust framework for vision-based roadside 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 21611–21620 (2023)
171. Yang, Z., Chen, J., Miao, Z., Li, W., Zhu, X., Zhang, L.: Deepinteraction: 3d object detection via modality interaction. In: S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, A. Oh (eds.) Advances in Neural Information Processing Systems, vol. 35, pp. 1992–2005. Curran Associates, Inc. (2022). URL https://proceedings.neurips.cc/paper_files/paper/2022/file/0d18ab3b5fabfa6fe47c62e711af02f0-Paper-Conference.pdf
172. Yang, Z., Jia, X., Li, Q., Yang, X., Yao, M., Yan, J.: Raw2drive: Reinforcement learning with aligned world models for end-to-end autonomous driving (in carla v2). arXiv preprint arXiv:2505.16394 (2025)
173. Yang, Z., Song, N., Li, W., Zhu, X., Zhang, L., Torr, P.H.: Deepinteraction++: Multi-modality interaction for autonomous driving. arXiv preprint arXiv:2408.05075 (2024)
174. Yang, Z., Sun, Y., Liu, S., Shen, X., Jia, J.: Std: Sparse-to-dense 3d object detector for point cloud. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 1951–1960 (2019)
175. Ye, T., Jing, W., Hu, C., Huang, S., Gao, L., Li, F., Wang, J., Guo, K., Xiao, W., Mao, W., Zheng, H., Li, K., Chen, J., Yu, K.: Fusionad: Multi-modality fusion for prediction and planning tasks of autonomous driving (2023). URL https://arxiv.org/abs/2308.01006
176. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021). DOI 10.1109/cvpr46437.2021.01161
177. Yin, T., Zhou, X., Krähenbühl, P.: Multimodal virtual point 3d detection. Advances in Neural Information Processing Systems **34**, 16494–16507 (2021)
178. Yoo, J.H., Kim, Y., Kim, J., Choi, J.W.: 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16, pp. 720–736. Springer (2020)
179. Yu, K., Tao, T., Xie, H., Lin, Z., Liang, T., Wang, B., Chen, P., Hao, D., Wang, Y., Liang, X.: Benchmarking the robustness of lidar-camera fusion for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 3188–3198 (2023)
180. Yuan, Y., Gao, P., Tan, X.: M3net: Multilevel, mixed and multistage attention network for salient object detection. arXiv preprint arXiv:2309.08365 (2023)
181. Zeng, W., Luo, W., Suo, S., Sadat, A., Yang, B., Casas, S., Urtasun, R.: End-to-end interpretable neural motion planner. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8660–8669 (2019)
182. Zhang, B., Song, N., Jin, X., Zhang, L.: Bridging past and future: End-to-end autonomous driving with historical prediction and planning. In: Proceedings of the Computer Vision and Pattern Recognition Conference, pp. 6854–6863 (2025)
183. Zhang, C., Wang, H., Cai, Y., Chen, L., Li, Y., Sotelo, M.A., Li, Z.: Robust-fusionnet: Deep multimodal sensor fusion for 3-d object detection under severe weather conditions. IEEE Transactions on Instrumentation and Measurement **71**, 1–13 (2022)
184. Zhang, G., Chen, J., Gao, G., Li, J., Liu, S., Hu, X.: Safdnet: A simple and effective network for fully sparse 3d object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 14477–14486 (2024)
185. Zhang, G., Junnan, C., Gao, G., Li, J., Hu, X.: Hednet: A hierarchical encoder-decoder network for 3d object

detection in point clouds. Advances in Neural Information Processing Systems **36**, 53076–53089 (2023)
186. Zhang, G., Ou, Z., Xue, K., Sun, J., Zhu, Y., Yao, S., Shen, Y., Song, M.: Dgfsd: Bridging the gap between dense and sparse for fully sparse 3d object detection. In: Proceedings of the 33rd ACM International Conference on Multimedia, pp. 4669–4678 (2025)
187. Zhang, G., Xie, J., Liu, L., Wang, Z., Yang, K., Song, Z.: Urformer: Unified representation lidar-camera 3d object detection with transformer. In: Chinese Conference on Pattern Recognition and Computer Vision (PRCV), pp. 401–413. Springer (2023)
188. Zhang, T., Chen, X., Wang, Y., Wang, Y., Zhao, H.: Mutr3d: A multi-camera tracking framework via 3d-to-2d queries. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4537–4546 (2022)
189. Zhang, Y., Chen, J., Huang, D.: Cat-det: Contrastively augmented transformer for multi-modal 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 908–917 (2022)
190. 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)
191. Zheng, W., Chen, W., Huang, Y., Zhang, B., Duan, Y., Lu, J.: Occworld: Learning a 3d occupancy world model for autonomous driving. In: European conference on computer vision, pp. 55–72. Springer (2024)
192. Zheng, W., Song, R., Guo, X., Zhang, C., Chen, L.: Genad: Generative end-to-end autonomous driving. In: European Conference on Computer Vision, pp. 87–104. Springer (2024)
193. Zhou, B., Krähenbühl, P.: Cross-view transformers for real-time map-view semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 13760–13769 (2022)
194. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4490–4499 (2018)
195. 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)
196. 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)
197. Zhu, X., Zyrianov, V., Liu, Z., Wang, S.: Mapprior: Bird's-eye view map layout estimation with generative models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 8228–8239 (2023)
198. Zuo, S., Zheng, W., Han, X., Yang, L., Lu, J., et al.: Quadricformer: Scene as superquadrics for 3d semantic occupancy prediction. Advances in Neural Information Processing Systems **38**, 47779–47801 (2026)
199. Zuo, S., Zheng, W., Huang, Y., Zhou, J., Lu, J.: Pointocc: Cylindrical tri-perspective view for point-based 3d semantic occupancy prediction. arXiv preprint arXiv:2308.16896 (2023)
200. 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)