2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

GaussianCaR: Gaussian Splatting for Efficient Camera-Radar Fusion

Santiago Montiel-Marín[1], Miguel Antunes-García[1], Fabio Sánchez-García[1], Angel Llamazares[1], Holger Caesar[2], and Luis M. Bergasa[1]

***Abstract*— Robust and accurate perception of dynamic objects and map elements is crucial for autonomous vehicles performing safe navigation in complex traffic scenarios. While vision-only methods have become the de facto standard due to their technical advances, they can benefit from effective and cost-efficient fusion with radar measurements. In this work, we advance fusion methods by repurposing Gaussian Splatting as an efficient *universal view transformer* that bridges the view disparity gap, mapping both image pixels and radar points into a common Bird's-Eye View (BEV) representation. Our main contribution is GaussianCaR, an end-to-end network for BEV segmentation that, unlike prior BEV fusion methods, leverages Gaussian Splatting to map raw sensor information into latent features for efficient camera-radar fusion. Our architecture combines multi-scale fusion with a transformer decoder to efficiently extract BEV features. Experimental results demonstrate that our approach achieves performance on par with, or even surpassing, the state-of-the-art on BEV segmentation tasks (57.3%, 82.9%, 50.1% IoU for vehicles, roads, and lane dividers) on the nuScenes dataset, while maintaining a 3.2× faster inference runtime. Code and project page are available online.**

I. Introduction

Developing robust and accurate perception models within an efficient framework is a cornerstone to enabling autonomous vehicles to achieve reliable scene understanding. Effectively interpreting dynamic objects and static map elements is a step towards ensuring safe navigation in complex environments such as traffic scenarios. Early perception solutions relied on LiDAR measurements [1], [2] due to the high precision of their 3D geometric information, despite the high cost and sensitivity to adverse weather conditions. With the advent of Deep Learning-based *projectors* or *view transformation* modules, which map image pixels into 3D space, vision-centric solutions [3]–[6] emerged as the dominant paradigm, offering a cost-effective path to large-scale deployment of perception models in the automotive domain. However, while cameras provide rich and dense semantic information, they lack motion cues and precise geometric accuracy, leading to depth and scale ambiguities, as well as localization errors. In contrast, radar provides a sparse yet accurate point cloud with position and velocity measurements, making it a suitable complementary sensor to cameras. Fusing cameras and radar signals enables a robust and cost-effective perception framework, suitable for mass-scale deployment in autonomous systems.

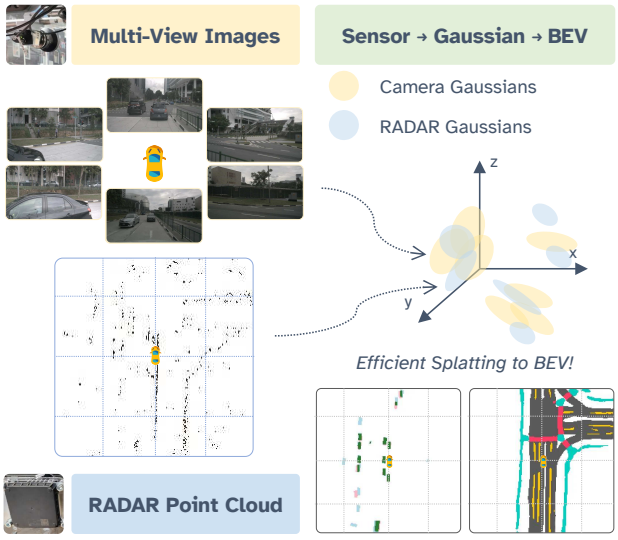

Fig. 1. We propose **GaussianCaR**, a novel method for efficient camera-radar fusion. We envision sensor fusion as a **modality** → **Gaussians** → **BEV** transformation, achieving competitive accuracy with significantly fast inference times for BEV segmentation tasks.

In this work, we tackle the key problem of fusing camera and radar modalities to produce a dense and robust BEV latent representation within a simple yet efficient framework for BEV perception tasks, focusing on vehicle and map segmentation. Since the introduction of BEVFusion [7], sensor fusion through BEV latent representations has become standard practice. The main challenge to fuse multiple modalities with different input representations lies in bridging the *view disparity*. On the one hand, camera data is represented as images, which naturally lack depth, scale, and motion information. To mitigate this issue, recent literature identifies two trends for performing an image-to-BEV transformation. Depth or forward-based approaches [3], [4], [8] estimate a depth distribution along rays passing through each pixel, but feature projection is limited to the physical distribution of grid cells. Projection or backward-based transformations aim to pull image features to a volume through simple interpolation [6] or costly attention-based learning [5]. On the other hand, radar point clouds must also be transformed into BEV representations, mainly through voxelization or pillarization. Mapping a radar measurement to a grid cell is straightforward; however, each point is assigned to a single voxel or pillar, without taking into account the uncertainties

[1] Department of Electronics. University of Alcalá, Spain.
[2] Department of Cognitive Robotics. Delft University of Technology, The Netherlands.
This work has been supported by projects PID2021-126623OB-I00 and PID2024-161576OB-I00, funded by MCIN/AEI/10.13039/501100011033 and co-funded by the European Regional Development Fund (ERDF, "A way of making Europe"), by project PLEC2023-010343 (INARTRANS 4.0) funded by MCIN/AEI/10.13039/501100011033, and by the R&D program TEC-2024/TEC-62 (iRoboCity2030-CM) and ELLIS Unit Madrid, granted by the Community of Madrid.

in the measurements. This process results in highly sparse latent BEV representations due to the limited number of points in radar point clouds and the relatively large working grid size. Recently, 3D Gaussian Splatting (GS) [9] has emerged as an efficient and powerful technique for 3D scene reconstruction, representing a scene as a set of learnable Gaussians that can be differentiably rasterized into plane-like representations. Inspired by this, we envision BEV sensor fusion as a **modality/view** $\rightarrow$ **Gaussians** $\rightarrow$ **BEV** transformation, leveraging GS as a *universal view transformer* for all modalities. This approach enables the unified sensor fusion of diverse inputs (pixels and points) with dense feature propagation and uncertainty awareness.

The main contribution of this paper is a robust, simple, and efficient sensor fusion framework for camera and radar data, leveraging Gaussian Splatting as a universal view transformer. We propose **GaussianCaR**, a novel method for BEV segmentation that uses two modality-specific encoders – Pixels-to-Gaussians and Points-to-Gaussians – to lift features from each sensor space into a unified sparse 3D space, enabling multi-modal fusion. To the best of our knowledge, we are the first model dedicated to BEV segmentation fusing camera and radar data within a Gaussian-based framework. Finally, we perform a multi-stage transformer-based fusion and decoding process to produce the desired BEV outputs. Extensive evaluation on the nuScenes dataset [10] demonstrates that our approach achieves state-of-the-art (SOTA) performance on dense BEV perception tasks while maintaining efficient inference time and memory usage.

In summary, we make three key claims: (i) **GaussianCaR** scores on par with, or even surpasses, SOTA methods in dense BEV perception tasks, such as vehicle, drivable surface, and lane segmentation; (ii) our Pixels-to-Gaussians and Points-to-Gaussians modules efficiently lift modality features to BEV space, enabling effective multi-modal sensor fusion; and (iii) the method is fast and efficient in terms of inference time, making it suitable for deployment. These claims are supported by the results and evaluations presented in this manuscript.

II. Related Work

In this section, we review related works in three areas: camera-based BEV perception, camera-radar fusion for BEV perception, and the use of Gaussian Splatting in robotics.

Camera-based BEV Perception. Vision-centric solutions for perception tasks were fundamentally limited by the ill-posed nature of monocular depth estimation in the camera perspective view. The foundational work LSS [3] proposed a shift from perspective view to local camera frustum space via differentiable *feature lifting* by predicting depth distribution and features per image. Lifted features from multiple views are aggregated into a unified BEV space. BEVDepth [8] improved depth estimation by incorporating cross-modal supervision from sparse LiDAR measurements. The BEVDet series [4], [11] further enhanced performance and introduced efficient view transformation techniques.

Other methods rely on projection or learning-based approaches performing view transformation via a learned component or an attention mechanism. SimpleBEV [6] employs bilinear sampling to populate local camera frustums. BEVFormer [5] introduces a 2D-to-3D attention mechanism to associate a set of BEV queries with image features. Hybrid approaches, such as FB-Occ [12] and BEVNeXt [13], combine geometrical and learning-based transformations within a unified framework.

In this work, we implement a Pixels-to-Gaussians encoder that lifts camera features to BEV space via differentiable Gaussian rasterization, expanding geometrical-based view transformations with a coarse-to-fine strategy for accurate spatial positioning of Gaussians in metric space.

Camera-Radar Fusion for BEV Perception. Camera and radar sensors exhibit complementary strengths and weaknesses. Camera provides dense, high-resolution semantic information, while radar delivers sparse but reliable spatial and motion cues, especially under adverse conditions. Fusing both modalities enables more accurate and robust scene understanding.

Early fusion methods operated in the perspective view by projecting points onto the image plane, as seen in CenterFusion [14], RADIANT [15], and CRAFT [16]. With the emergence of BEVFusion [7], dense fusion in a unified BEV space became the dominant paradigm for combining images and radar point clouds. SimpleBEV [6] incorporated radar in BEV space via an occupancy map. BEVCar [17] uses radar points to guide bilinear sampling of image features, enabling multi-level fusion. Methods such as CRN [18] and CRT-Fusion [19] perform fusion in the camera frustum view using fused frustum volumes or attention-based mechanisms to align modalities.

We propose performing fusion in BEV space through a two-step transformation, **modality** $\rightarrow$ **Gaussians** $\rightarrow$ **BEV**. To this end, we encode radar data with our Points-to-Gaussians module, which converts radar measurements to Gaussians to effectively capture and propagate uncertainty.

Gaussian Splatting in Robotics. Gaussian Splatting [9] has rapidly become a foundation for scene reconstruction and neural rendering. It represents 3D environments as sets of learnable anisotropic Gaussian primitives, each parametrized by position, scale, rotation, opacity, and feature vector. This formulation enables efficient, fully differentiable forward rendering and continuous geometric representation.

The ability to capture the environment with an efficient, continuous, and differentiable geometric representation makes GS a promising approach for robotics and perception applications. In the SLAM domain, OpenGS-SLAM [20] performs open-set segmentation and indoor scene reconstruction from an RGB-D stream as input, while WildGS-SLAM [21] reconstructs 3D Gaussian maps for static scenes, handling dynamic objects to avoid scene blurring. In dense BEV perception, GaussianLSS [22] and GaussianBeV [23] introduce camera-only architectures with differentiable Gaussian rendering to lift image features into BEV space and perform semantic segmentation in an end-to-end fashion.

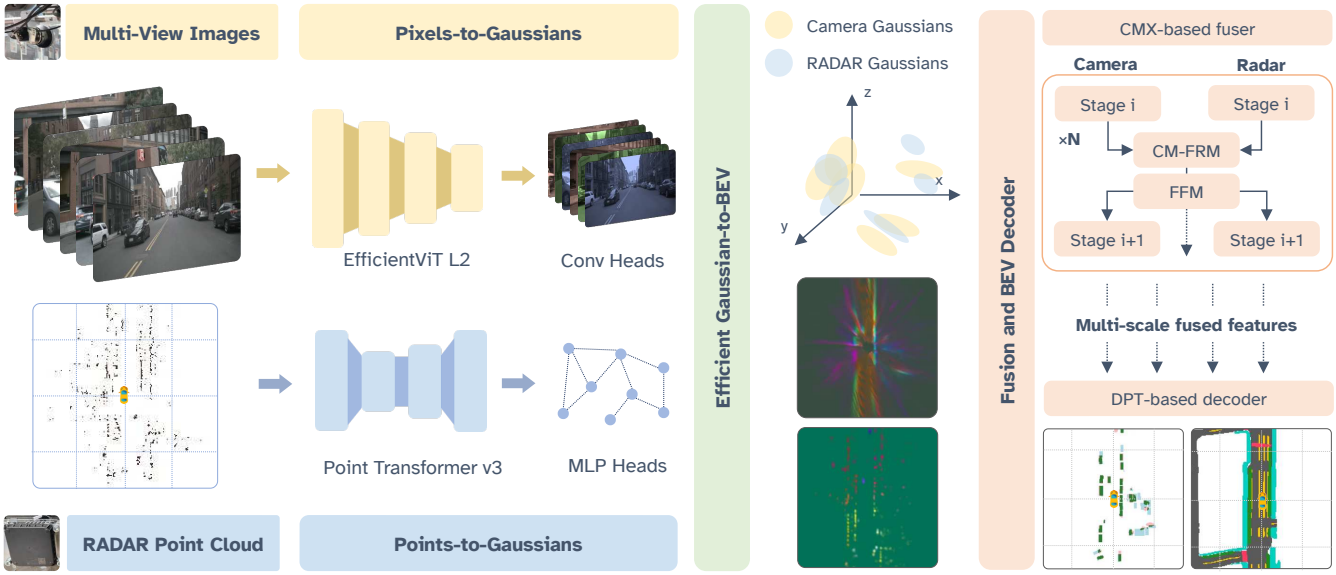

Fig. 2. **Main diagram of our proposal, GaussianCaR.** Given multi-view camera images and radar point clouds, we leverage Gaussian Splatting as a *universal view transformer* and formulate sensor fusion as **modality** → **Gaussians** → **BEV** transformation. The model predicts BEV segmentation maps for dynamic vehicles and map elements. We employ two feature encoding branches: Pixels-to-Gaussians for camera features and Points-to-Gaussians for radar point clouds. Features are splatted and fused in BEV space using a CMX-based fuser, and decoded via a DPT decoder.

Building upon this paradigm, we repurpose Gaussian Splatting as a *universal view transformer*, mapping input modalities to BEV latent representations through a set of 3D Gaussian primitives, and extend the approach from camera to camera-radar data. To the best of our knowledge, this is the first cost-effective and efficient Gaussian-based framework for camera-radar sensor fusion applied to BEV segmentation tasks, paving the way for large-scale deployment.

III. Methodology

A. Task Definition and Overview

The primary objective of **GaussianCaR**, described in Fig. 2, is to predict BEV segmentation maps of relevant road entities for autonomous navigation, such as vehicles or drivable surfaces, leveraging Gaussian Splatting techniques to fuse multi-view cameras and radar sensors.

Given as inputs: (a) images from a multi-camera system with N_c views, $I \in \mathbb{R}^{N_c \times 3 \times H \times W}$; (b) a radar point cloud with N_r points, and F_r dimensional features, $R \in \mathbb{R}^{N_r \times F_r}$; and (c) the intrinsic and extrinsic calibration matrices between the vehicle sensors. From these inputs, GaussianCaR produces a BEV segmentation map, $S \in \mathbb{R}^{C \times H_{BEV} \times W_{BEV}}$, where C is the number of semantic classes and H_{BEV}, W_{BEV} define the BEV resolution. Our approach leverages two modality-specific encoders: for camera, Pixels-to-Gaussians (Sec. III-B), and for radar, Points-to-Gaussians (Sec. III-C). Our modality-based fusion and BEV decoding is described in Sec. III-D. Our training objectives are defined in Sec. III-E.

B. Pixels to Gaussians

For our Pixels-to-Gaussians encoder (in Fig. 4), we build upon the foundations in [22], [23]. We process multi-camera images, I, and extract x1/8 low-resolution feature maps, F, using EfficientViT-L2 [24], a lightweight, transformer-based backbone, and a neck for feature aggregation.

To lift image features from pixel space to 3D, we map from pixels to Gaussians, producing $|\mathcal{G}| = N_c \cdot H_{low} \cdot W_{low}$ Gaussians. A series of convolutional heads is applied to the low-resolution feature maps to predict the physical and semantic properties of each Gaussian, $\mathcal{G}_i$, including position, p_i, size, s_i, orientation, R_i, opacity, α_i, and features, f_i.

We estimate the geometrical position or mean of each Gaussian, p_i, using a coarse-to-fine strategy. In the coarse stage, a depth head predicts a probability distribution along the optical ray for each pixel, where depth is uniformly discretized into B bins between $[d_{min}, d_{max}]$. This produces a tensor $F_{dep} \in \mathbb{R}^{|\mathcal{G}| \times B}$, containing per-Gaussian depth classification logits. A coarse position is calculated via probability-weighted sum over the depth bins and projected to 3D space using the camera intrinsic and extrinsic matrices. In the fine stage, an offset head refines the final 3D position in metric space, $F_{off} \in \mathbb{R}^{|\mathcal{G}| \times 3}$, enabling the Gaussian to deviate from the set of discrete bin centers and achieve higher precision. Final per-Gaussian position (shown in Fig. 3.a-b) is determined as:

$$\mathbf{p}_i = \mathcal{P}(\mathbf{u}_i, \hat{d}_i(F_{dep_i})) + F_{off_i} \tag{1}$$

where $\hat{d}_i$ is the predicted bin center along the optical ray, and $\mathcal{P}(\mathbf{u}_i, \dots)$ is the back-projection of a pixel $\mathbf{u}_i$ to the world using intrinsic and extrinsic matrices.

The size, s_i, and orientation, R_i, of each Gaussian are derived from the predicted depth distribution and camera geometry. From the probability distribution in the coarse estimation, we compute the standard deviations around the mean position. These deviations are assembled into a covariance matrix that encodes spatial uncertainty. The

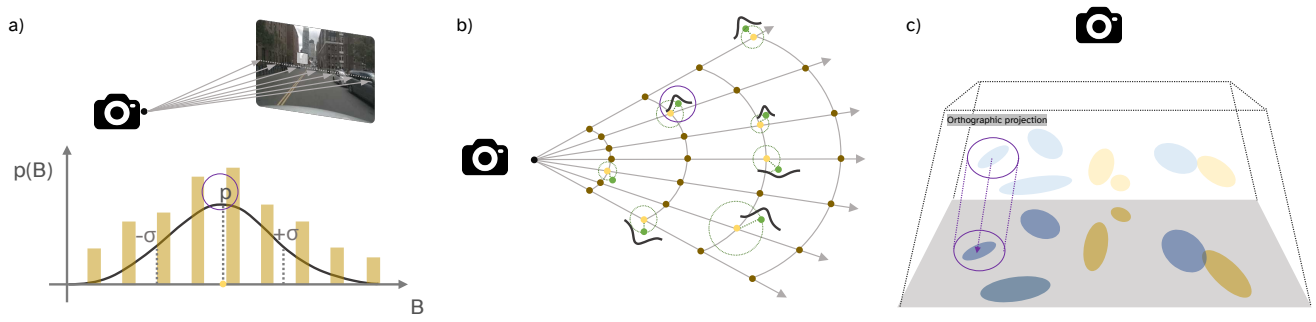

Fig. 3. **Gaussian modeling process.** In (a), we present the process of extracting a Gaussian from a discrete probability distribution; in (b), we depict the behavior of the offset head, displacing the final Gaussian position from the original set of candidates; in (c), we illustrate the Gaussian rasterization process, projecting Gaussians from 3D space to BEV space via orthographic projection.

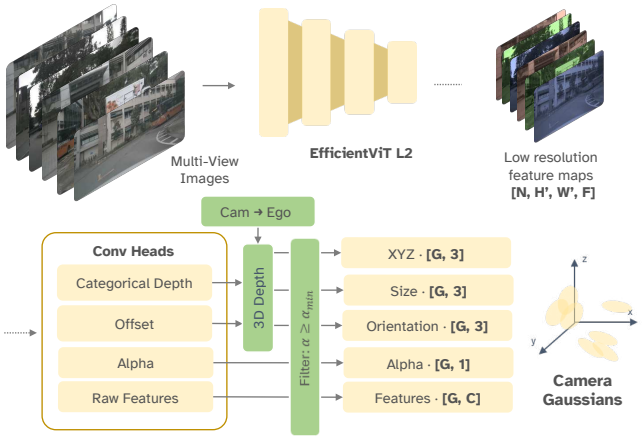

Fig. 4. Our **Pixels-to-Gaussians** extracts low-resolution feature maps using an EfficientViT backbone and a neck. A set of convolutional heads predicts $\mathcal{G}_c$ Gaussians. To position the Gaussians in 3D space, camera intrinsic and extrinsic matrices are used.

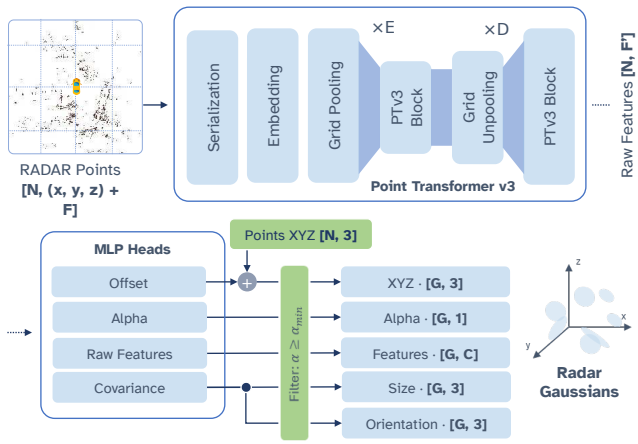

Fig. 5. Our proposed **Points-to-Gaussians** module processes radar point clouds using a lightweight PTv3, composed of $\mathcal{E}$ encoder and $\mathcal{D}$ decoder blocks. A set of MLP heads then predicts $\mathcal{G}_r$ Gaussians, each parameterized by geometric and semantic attributes.

covariance is then scaled by an error tolerance coefficient, $k = 0.5$, which controls the effective spread of the Gaussian. Finally, the eigenvalues of the scaled covariance determine the size along each principal axis, while the eigenvectors define the orientation in 3D space.

Lastly, each Gaussian is assigned with an opacity parameter $\alpha_i \in [0, 1]$, predicted by a convolutional head followed by a sigmoid activation, yielding a tensor $F_{opa} \in \mathbb{R}^{|\mathcal{G}| \times 1}$. This parameter regulates the influence of each Gaussian during differentiable rendering. Furthermore, we empirically set a minimum threshold $\alpha_{min} = 0.01$, allowing Gaussians with negligible contribution to be discarded.

C. Points to Gaussians

To extract features from radar point clouds, we employ a lightweight variant of Point Transformer v3 (PTv3) [25], depicted in Fig. 5. The raw, unstructured point cloud is serialized and transformed into multiple ordered representations using space-filling curves and neighbor mapping. From these representations, non-overlapping patches are constructed to capture local neighborhoods. An efficient inter-patch attention mechanism is then applied, enabling both spatial and global context modeling at the per-point level. The overall architecture follows a UNet-like design and outputs point-wise feature embeddings with rich semantic information.

On top of these embeddings, we attach a set of MLP heads to predict the physical attributes and semantic properties of each point. Similar to the coarse-to-fine position estimation in the camera branch, we predict only a metric offset head, as the initial point positions are already known. The opacity attribute is estimated in the same manner as in the camera branch. For size and orientation, we predict a compact representation of the covariance matrix:

$$R_{cov_i} \in \mathbb{R}^6 = [xx\ xy\ xz\ yy\ yz\ zz]$$

where eigenvalues are enforced to be positive via a softplus activation. Further implementation details for PTv3 can be found in [26].

D. Modality-based Fusion and BEV Decoding

To complete the **modality** → **Gaussian** → **BEV** cycle, we splat each set of learned $|f_i|$-dimensional Gaussian representations (with $|f_i| = 128$ in our experiments) to BEV using differentiable Gaussian rasterization through an

TABLE I
BEV VEHICLE SEGMENTATION ON THE NUSCENES VALIDATION SET

Method	Code	Cam Enc	Radar Enc	IoU (↑)
Camera-only				
BEVFormer [5]	✓	RN-101	-	43.2
GaussianLSS [22]	✓	RN-101	-	46.1
SimpleBEV [6]	✓	RN-101	-	47.4
PointBeV [29]	✓	EN-b4	-	47.8
GaussianBeV [23]	✗	EN-b4	-	50.3
Camera-radar				
SimpleBEV++‡ [6]	✓	RN-101	PFE+Conv	52.7
SimpleBEV [6]	✓	RN-101	Conv	55.7
BEVCar‡ [17]	✓	DINOv2/B +Adapter	PFE+Conv	58.4
CRN◇ [18]	✗	RN-50	SECOND	58.8
BEVGuide [30]	✗	EN-b4	SECOND	**59.2**
GaussianCaR (*ours*)	✓	EViT-L2	PTv3	57.3

We report the results from [17], [23]. Evaluation done with image resolution $(448, 800)$ (or ‡$(448, 896)$, if needed) and applying visibility filtering. ◇CRN uses 4 input frames (3 past and 1 current) at inference time. ✗ GaussianBeV, CRN, and BEVGuide do not release code for BEV segmentation. Best is marked in **bold** and second best is underlined.

TABLE II
BEV MAP SEGMENTATION ON THE NUSCENES VALIDATION SET

Method	Driv. Area (↑)	Lane Div. (↑)
Camera-only		
LSS [31]	72.9	20.0
BEVFormer [5]	80.1	25.7
GaussianBeV [32]	82.6	47.4
Camera-radar		
BEVGuide [33]	76.7	44.2
Simple-BEV++‡ [6]	81.2	40.4
BEVCar‡ [17]	**83.3**	45.3
GaussianCaR (*ours*)	82.9	**50.1**

We report the results from [17], [23]. Evaluation done with image resolution $448, 800$ (or ‡$(448, 896)$, if needed) and applying visibility filtering. Visibility filtering does not apply to map evaluation. Best is marked in **bold** and second best is underlined.

orthographic projection and alpha-blending:

$$\mathbf{F} = \sum_{i \in \mathcal{N}} f_i \, \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j) \tag{2}$$

where f_i is the feature vector of each Gaussian, and F is the computed per-pixel feature after blending, shown in Fig. 3-c.

Inspired by [27], we adopt a four-stage, multi-scale feature fusion strategy, depicted in Fig. 2. Each stage receives feature maps from both modalities and consists of Cross-Modal Feature Rectification (CM-FRM) and Feature Fusion Modules (FFM), producing fused representations that serve as inputs for a DPT-based decoder [28] to generate the final BEV representation. The output of the first fusion stage is connected to an auxiliary head, while the output of the decoder feeds the final segmentation head.

E. Training Losses

We train our model end-to-end using two semantic segmentation loss terms, a main loss L_{sem} and an auxiliary loss L_{sem}^{aux}. It is defined as:

$$L = L_{sem} + L_{sem}^{aux} \tag{3}$$

For each component, we adopt a combo loss, comprising a binary cross-entropy L_{bce} and Dice loss L_{dice}, and additional centerness L_{ctr} and offset L_{off} components for regularization, each balanced by its respective weight λ_i.

$$\begin{aligned} L_{sem} = L_{sem}^{aux} = \lambda_{bce} \cdot L_{bce} + \lambda_{dice} \cdot L_{dice} \\ + \lambda_{ctr} \cdot L_{ctr} + \lambda_{off} \cdot L_{off} \end{aligned} \tag{4}$$

While L_{sem} is applied to the final BEV prediction and L_{sem}^{aux} is attached to the output of the first feature fusion stage to provide early supervision, both losses are computed using an identical definition, following Eq. 4.

IV. EXPERIMENTAL EVALUATION

The main focus of this work is to enable robust and efficient fusion of camera and radar data for BEV perception tasks, leveraging Gaussian Splatting as *universal view transformer*. We present our experiments to demonstrate the capabilities of our method and support our key claims, showing that our approach achieves performance on par with, or even surpassing, SOTA methods in BEV segmentation tasks, while maintaining fast and efficient inference runtimes. Finally, we validate our design choices through an ablation study that highlights the effectiveness of our proposal.

A. Experimental Settings

We present our experimental setup, detailing the dataset, evaluation metrics, and implementation specifics.

Dataset and Metrics: We train and evaluate our model on the nuScenes [10] dataset, the only large-scale multimodal dataset that includes synchronized data from 6 surround-view cameras, 5 automotive radars, and a 32-beam LiDAR, as well as high-quality annotations for 3D objects and map surfaces. The dataset contains 1,000 20-second driving scenes, split into 700 training, 150 validation, and 150 test sequences.

We quantify the performance of our network in the tasks of BEV vehicle and map segmentation using the Intersection over Union (IoU) or Jaccard index metric, defined as:

$$IoU(\hat{y}, y) = \frac{|\hat{y} \cap y|}{|\hat{y} \cup y|} = \frac{\sum_{H,W} \hat{y} \cdot y}{\sum_{H,W} (\hat{y} + y - \hat{y} \cdot y)} \tag{5}$$

where $\hat{y}_i \in \{0, 1\}$ is the confidence-thresholded prediction, and $y_i \in \{0, 1\}$ is the ground-truth label.

Implementation Details: We train our model for 40 epochs in a distributed setup consisting of 4x NVIDIA A100 80 GB GPUs, using a DDP strategy and gradient accumulation for an effective batch size of 16. We use the AdamW optimizer and a linear annealing scheduler with warmup. Maximum learning rate is $lr_{max} = 3e^{-4}$ and linearly decreases to $lr_{end} = 0$. Weight decay is set to $w_d = 1e^{-7}$. Gaussian splatting rasterizers use a version

TABLE III
ABLATION STUDY

Method	IoU (↑)	ms (↓)	FPS (↑)
Baseline: GaussianLSS [22]	46.1	53.9	18.6
Image Encoding Branch			
+ EffViT L2	47.3	56.6	17.8
+ Offset Head	47.7	56.9	17.6
+ Early auxiliary loss	47.8	56.9	17.6
+ Dice loss	48.0	56.9	17.6
Radar Encoding Branch			
+ PTv3 w./ scatter (XYZ)	55.0	86.1	11.6
+ PTv3 w./ scatter (*all variables*)	56.1	86.9	11.5
+ PTv3 w./ Gaussians (*all variables*)	56.9	83.7	12.0
Fusion and BEV Decoding			
+ DPT-based decoder	57.1	78.2	12.8
+ CMX-based fuser	57.3	75.6	13.2

All experiments are run on an NVIDIA RTX 4090 with an image resolution of $(448, 800)$ for the task of vehicle segmentation, utilizing visibility filtering.

of `diff-gaussian-rasterization` library from [22]. The architecture and codebase are implemented in PyTorch 2.4.1 and Lightning.

Images are processed in half-scale $H, W = (448, 800)$. We apply image data augmentation, such as random horizontal flip, zoom-in/out, and rotations, with camera intrinsic matrices being updated consistently. We accumulate 7 radar sweeps and preprocess all variables in the point cloud. We apply data augmentation in BEV space, following [29].

Gaussians are rasterized to a BEV grid space with a perception range of 100 m in both x-y directions (from -50 to 50 m) with 0.5 m of resolution, resulting in a grid of 200×200 cells.

B. Quantitative Results

In this section, we compare **GaussianCaR** with our camera-only baseline, GaussianLSS [22], and SOTA approaches across two BEV segmentation tasks: vehicle and map. We support the claim that we achieve competitive performance, on par with the current SOTA or even surpassing it.

Vehicle segmentation. We evaluate our model and multiple SOTA methods for BEV vehicle segmentation in Tab. I. To ensure a fair comparison, we follow [17], [23] and evaluate the task applying vehicle visibility filtering (at least 40%) and image resolution $(448, 800)$.

We first evaluate against our vision-based baseline, GaussianLSS [22], as well as leading methods including BEVFormer [5], SimpleBEV [6], PointBeV [29], and GaussianBeV [23]. Our fusion-based approach outperforms them, achieving a +7.0 IoU over the prior SOTA, and demonstrating the added value of radar data in vision-centric approaches. Next, we compare against fusion methods: SimpleBEV [6], BEVCar, SimpleBEV++ [17], CRN [18], and BEVGuide [30]. Note that CRN requires 4 input frames at inference time and, as BEVGuide, do not release code, complicating direct comparison. Our method outperforms SimpleBEV (+1.6 IoU) and performs competitively with the strongest fusion-based approaches, with only a -1.1 IoU gap relative to BEVCar.

TABLE IV
RUNTIME ANALYSIS

Method	Veh. IoU (↑)	ms (↓)	FPS (↑)
Simple-BEV† [6]	55.7	**57.6**	**17.4**
Simple-BEV++‡ [17]	52.7	211.3	4.7
BEVCar‡ [17]	**58.4**	245.6	4.1
GaussianCaR† *(vehicle)*	<u>57.3</u>	<u>75.6</u>	<u>13.2</u>
GaussianCaR† *(map)*	–	81.1	12.3

Inference time of a forward pass measured on an NVIDIA RTX 4090 with image resolution †: $(448, 800)$, or ‡: $(448, 896)$. Best is marked in **bold** and second best is <u>underlined</u>.

Map segmentation. For this task, we aim to segment all relevant road elements such as: drivable area, lane boundaries, road and lane dividers, pedestrian crossings, walkways and carpark areas. Following [17], [23], we report metrics for the drivable area and lane boundaries and evaluate the task with image resolution $(448, 800)$ in Tab. II.

Our approach surpasses all camera-only baselines in drivable area and lane boundary segmentation, achieving improvements of +0.3 IoU and +2.7 IoU, respectively, over GaussianBeV. When compared to fusion-based methods, our method matches the top-performing approach, BEVCar, in drivable area segmentation, while substantially outperforming it in lane boundary segmentation, with a margin of +4.8 IoU.

C. Ablation Study

We conduct an incremental ablation study (Tab. III) to assess our proposed framework, GaussianCaR, which extends the camera-only baseline GaussianLSS [22], evaluating all methods on vehicle segmentation at $(448, 800)$ resolution.

For our baseline, we report an IoU score of 46.1 with a runtime of 18.6 Hz. We introduce EfficientViT L2 as a stronger image backbone to ensure a fair comparison with the current SOTA of camera-radar fusion methods, and the metric offset head. For supervision, we introduce an early guidance loss and a Dice loss component for both segmentation losses, leading to an improvement of $+1.9$ IoU. The combination of these components constitutes our Pixels-to-Gaussians module.

We add a radar branch based on a lightweight PTv3 with 7 accumulated sweeps. Encoding radar positions with a scatter-to-BEV mechanism yields 55.0 IoU, a +7.0 gain over vision-only. Incorporating all radar variables further improves performance by +1.1 IoU. We then introduce Gaussians as a view transformation, allowing features to offset and diffuse locally; together, these components form our Points-to-Gaussians module, reaching 56.9 IoU. Lastly, since we fuse two modalities, we adopted the multi-scale gating fusion and BEV decoder from [34] for the previous experiments. We propose to improve this stage by incorporating our CMX-based fusion and DPT-based decoder, yielding +0.4 IoU at 13.2 Hz, confirming SOTA-level accuracy with high efficiency.

To evaluate the effect of accumulated radar sweeps, we ablate the model using 1, 4, and 7 sweeps, obtaining IoU

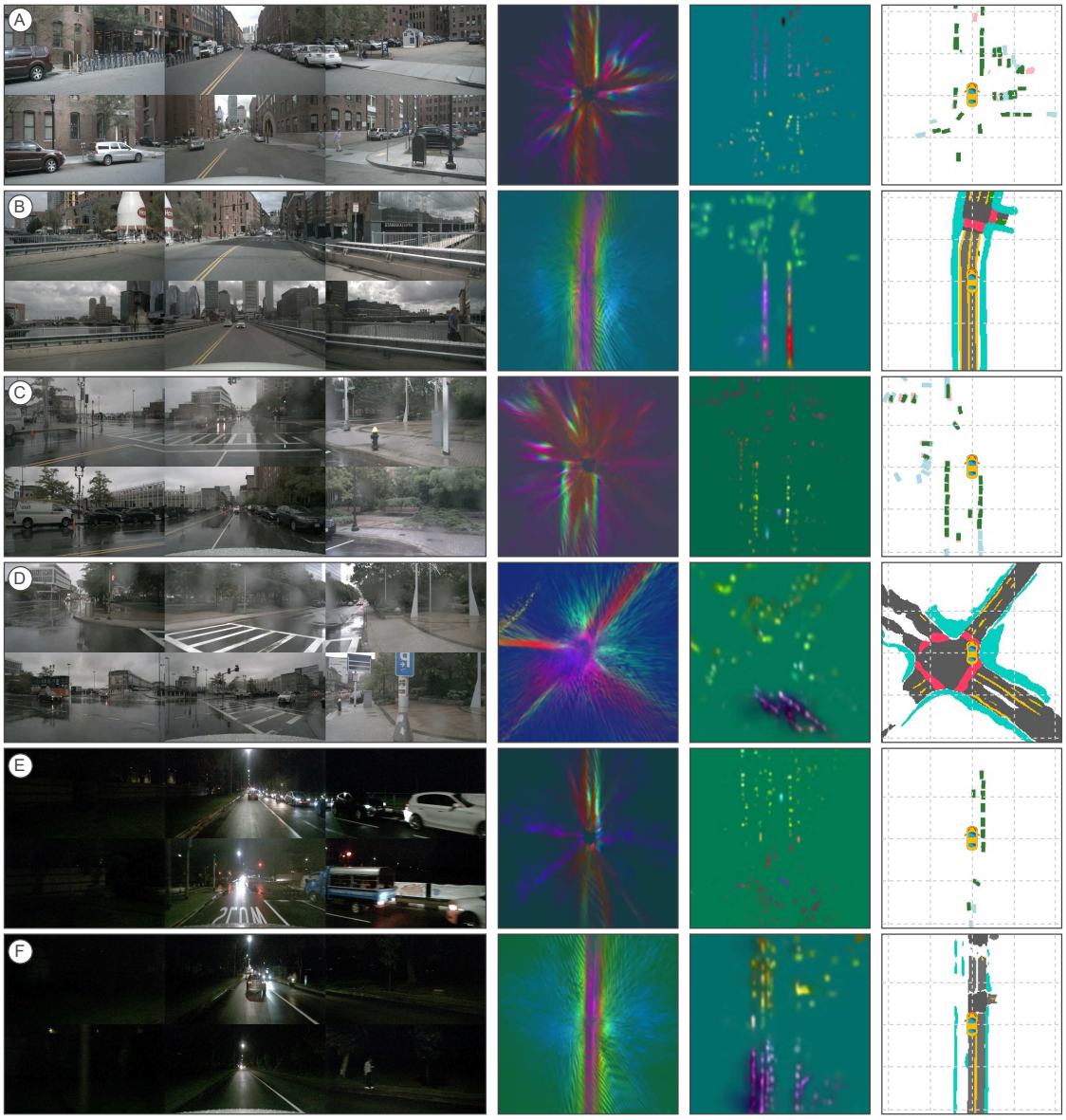

Fig. 6. **Qualitative results on the nuScenes validation set.** Each row shows, from left to right: multi-view camera images, PCA camera latent features, PCA radar latent features, and predictions. For vehicle segmentation, we report an error map where correctness is indicated by color: correct, missing, and incorrect. For map segmentation, we report classes by color: drivable area, lane and road dividers, pedestrian crossings, walkway and carpark areas.

scores of 54.9, 56.2, and 57.3, respectively, which confirms the positive impact of incorporating additional radar sweeps.

D. Runtime Analysis

To support the claim that our approach is robust and efficient for camera-radar fusion, we report the forward pass runtimes of our proposed **GaussianCaR** alongside other SOTA methods in Tab. IV. All measurements were conducted on an NVIDIA RTX 4090 GPU using a batch size of 1 with FP32 precision, and image resolution $(448, 800)$.

A key observation from our method is that, while efficient, the rasterization module scales linearly with the number of Gaussians, making the convergence count a key determinant of runtime. Convergence typically occurs at $\sim$14k Gaussians for the vehicle segmentation task, and $\sim$24k for the map. Consequently, the runtime of our method exhibits slight task-dependent variations. We report a mean runtime of **75.6 ms** and **81.1 ms** on vehicle and map segmentation, respectively.

We conduct a fair comparison with two SOTA methods, SimpleBEV++ and BEVCar, both employing similar image backbones. Their inference times are 211.3 and 245.6 ms, respectively. Our method achieves performance on par with BEVCar while preserving the efficiency of SimpleBEV, resulting in a **3.2× faster runtime** compared to BEVCar.

E. Qualitative Results

Fig. 6 shows qualitative results on nuScenes. Each row corresponds to a scene, displaying multi-view inputs, PCA-projected camera and radar features, and predictions. Scenes 6.a–b represent daytime urban traffic, 6.c–d a rainy four-way intersection, and 6.e–f nighttime driving.

V. Conclusion

In this paper, we propose a novel framework for simple, yet robust and efficient camera–radar fusion in perception applications, demonstrating strong performance in both accuracy and inference speed. Our method leverages Gaussian Splatting to reframe sensor fusion in latent space as a **modality → Gaussians → BEV** process. We implement and evaluate our approach on BEV segmentation tasks using the nuScenes dataset, achieving performance on par with the state of the art, and even surpassing it in lane divider segmentation. Furthermore, we achieve a 3.2× faster runtime compared to BEVCar, delivering both top-tier performance and fast inference. These promising results also open several avenues for future research, including alternative backbones that exploit the modality-to-Gaussian transformation cycle or novel fusion mechanisms in the Gaussian intermediate space.

References

[1] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 697–12 705.

[2] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 784–11 793.

[3] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D," in *European Conference on Computer Vision*, 2020, pp. 194–210.

[4] J. Huang, G. Huang, Z. Zhu, Y. Ye, and D. Du, "BEVDet: High-performance multi-camera 3D object detection in bird-eye-view," *arXiv preprint arXiv:2112.11790*, 2021.

[5] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *European Conference on Computer Vision*, 2022.

[6] A. W. Harley, Z. Fang, J. Li, R. Ambrus, and K. Fragkiadaki, "Simple-BEV: What really matters for multi-sensor BEV perception?" in *IEEE International Conference on Robotics and Automation*, 2023, pp. 2759–2765.

[7] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. L. Rus, and S. Han, "Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *2023 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2023, pp. 2774–2781.

[8] Y. Li, Z. Ge, G. Yu, J. Yang, Z. Wang, Y. Shi, J. Sun, and Z. Li, "Bevdepth: Acquisition of reliable depth for multi-view 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 2, 2023, pp. 1477–1485.

[9] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[10] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A multimodal dataset for autonomous driving," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 618–11 628.

[11] J. Huang and G. Huang, "Bevpoolv2: A cutting-edge implementation of bevdet toward deployment," *arXiv preprint arXiv:2211.17111*, 2022.

[12] Z. Li, Z. Yu, D. Austin, M. Fang, S. Lan, J. Kautz, and J. M. Alvarez, "Fb-occ: 3d occupancy prediction based on forward-backward view transformation," *arXiv preprint arXiv:2307.01492*, 2023.

[13] Z. Li, S. Lan, J. M. Alvarez, and Z. Wu, "Bevnext: Reviving dense bev frameworks for 3d object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 20 113–20 123.

[14] R. Nabati and H. Qi, "Centerfusion: Center-based radar and camera fusion for 3d object detection," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 1527–1536.

[15] Y. Long, A. Kumar, D. Morris, X. Liu, M. Castro, and P. Chakravarty, "Radiant: Radar-image association network for 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 2, 2023, pp. 1808–1816.

[16] Y. Kim, S. Kim, J. W. Choi, and D. Kum, "Craft: Camera-radar 3d object detection with spatio-contextual fusion transformer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 1, 2023, pp. 1160–1168.

[17] J. Schramm, N. Vödisch, K. Petek, B. R. Kiran, S. Yogamani, W. Burgard, and A. Valada, "Bevcar: Camera-radar fusion for bev map and object segmentation," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 1435–1442.

[18] Y. Kim, J. Shin, S. Kim, I.-J. Lee, J. W. Choi, and D. Kum, "CRN: Camera radar net for accurate, robust, efficient 3D perception," in *International Conference on Computer Vision*, 2023, pp. 17 569–17 580.

[19] J. Kim, M. Seong, and J. W. Choi, "Crt-fusion: Camera, radar, temporal fusion using motion information for 3d object detection," *arXiv preprint arXiv:2411.03013*, 2024.

[20] D. Yang, Y. Gao, X. Wang, Y. Yue, Y. Yang, and M. Fu, "Opengs-slam: Open-set dense semantic slam with 3d gaussian splatting for object-level scene understanding," *arXiv preprint arXiv:2503.01646*, 2025.

[21] J. Zheng, Z. Zhu, V. Bieri, M. Pollefeys, S. Peng, and I. Armeni, "Wildgs-slam: Monocular gaussian splatting slam in dynamic environments," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 11 461–11 471.

[22] S.-W. Lu, Y.-H. Tsai, and Y.-T. Chen, "Toward real-world bev perception: Depth uncertainty estimation via gaussian splatting," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 17 124–17 133.

[23] F. Chabot, N. Granger, and G. Lapouge, "Gaussianbev: 3d gaussian representation meets perception models for bev segmentation," in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2025, pp. 2250–2259.

[24] H. Cai, J. Li, M. Hu, C. Gan, and S. Han, "Efficientvit: Lightweight multi-scale attention for high-resolution dense prediction," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 17 302–17 313.

[25] X. Wu, L. Jiang, P.-S. Wang, Z. Liu, X. Liu, Y. Qiao, W. Ouyang, T. He, and H. Zhao, "Point transformer v3: Simpler faster stronger," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 4840–4851.

[26] P. Wolters, J. Gilg, T. Teepe, F. Herzog, F. Fent, and G. Rigoll, "Sparc: Sparse radar-camera fusion for 3d object detection," *arXiv preprint arXiv:2411.19860*, 2024.

[27] J. Zhang, H. Liu, K. Yang, X. Hu, R. Liu, and R. Stiefelhagen, "Cmx: Cross-modal fusion for rgb-x semantic segmentation with transformers," *IEEE Transactions on intelligent transportation systems*, vol. 24, no. 12, pp. 14 679–14 694, 2023.

[28] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 179–12 188.

[29] L. Chambon, E. Zablocki, M. Chen, F. Bartoccioni, P. Pérez, and M. Cord, "Pointbev: A sparse approach for bev predictions," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 15 195–15 204.

[30] Y. Man, L.-Y. Gui, and Y.-X. Wang, "Bev-guided multi-modality fusion for driving perception," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 960–21 969.

[31] J. Philion and S. Fidler, "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*. Springer, 2020, pp. 194–210.

[32] A. Hu, Z. Murez, N. Mohan, S. Dudas, J. Hawke, V. Badrinarayanan, R. Cipolla, and A. Kendall, "Fiery: Future instance prediction in bird's-eye view from surround monocular cameras," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 273–15 282.

[33] Y. Man, L.-Y. Gui, and Y.-X. Wang, "BEV-guided multi-modality fusion for driving perception," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 960–21 969.

[34] S. Montiel-Marín, A. Llamazares, M. Antunes-García, F. Sánchez-García, and L. M. Bergasa, "Car1: A multi-modal baseline for bev vehicle via camera-radar fusion," *arXiv preprint arXiv:2025.10139*, 2025.