# BEV-Guided Multi-Modality Fusion for Driving Perception

Yunze Man
UIUC
yunzem2@illinois.edu

Liang-Yan Gui
UIUC
lgui@illinois.edu

Yu-Xiong Wang
UIUC
yxw@illinois.edu

## Abstract

*Integrating multiple sensors and addressing diverse tasks in an end-to-end algorithm are challenging yet critical topics for autonomous driving. To this end, we introduce BEVGuide, a novel Bird's Eye-View (BEV) representation learning framework, representing the first attempt to unify a wide range of sensors under direct BEV guidance in an end-to-end fashion. Our architecture accepts input from a diverse sensor pool, including but not limited to Camera, Lidar and Radar sensors, and extracts BEV feature embeddings using a versatile and general transformer backbone. We design a BEV-guided multi-sensor attention block to take queries from BEV embeddings and learn the BEV representation from sensor-specific features. BEVGuide is efficient due to its lightweight backbone design and highly flexible as it supports almost any input sensor configurations. Extensive experiments demonstrate that our framework achieves exceptional performance in BEV perception tasks with a diverse sensor set. Project page is at https://yunzeman.github.io/BEVGuide.*

## 1. Introduction

The recent research in Bird's Eye-View (BEV) perception and multi-sensor fusion has stimulated rapid progress for autonomous driving. The BEV coordinates naturally unify various downstream object-level and scene-level perception tasks, while joint learning with multiple sensors minimizes uncertainty, resulting in more robust and accurate predictions. However, existing work still exhibits fundamental limitations. On the one hand, fusion strategies often necessitate explicit space transformations, which can be ill-posed and prone to errors. On the other hand, existing techniques utilizing BEV representations rely on ad-hoc designs and support a limited set of sensors (*i.e.*, cameras and Lidar). These constraints impede the evolution of a more general and flexible multi-sensor architecture for BEV 3D perception, which inspires the design of our work.

More specifically, as different sensors always lie in different coordinate systems, prior approaches usually transform features of each sensor into the same space prior to fusion. For example, some work prioritizes one sensor over another [1, 42, 43]. However, such fusion architectures tend to be inflexible and heavily reliant on the presence of the primary sensor – should the primary sensor be unavailable or malfunction, the entire pipeline collapses. Alternatively, other work transforms all sensors into the same space (3D or BEV space) using provided or estimated geometric constraints [10, 22, 25]. Such methods usually require an explicit depth estimation from camera images, which is susceptible to errors due to the ill-posed nature of the image modality. Moreover, errors that arise during the transformation process may propagate into subsequent feature fusion stages, ultimately impacting downstream tasks. Our approach seeks to streamline this process by employing the BEV space to directly guide the fusion of multiple sensor feature maps within their *native spaces*.

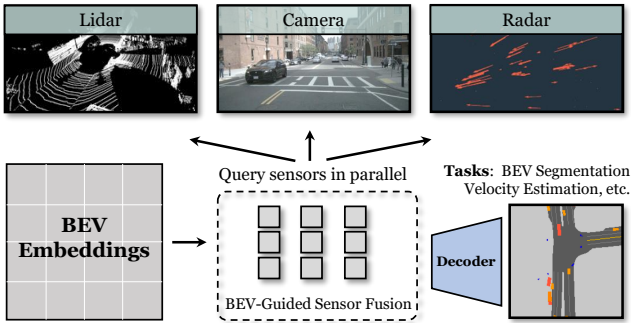


Figure 1. BEVGuide takes input from a sensor combination and learns BEV feature representation using a portable and general BEV-guided multi-sensor attention module. In principle, BEVGuide is able to take a wide variety of sensors and perform any BEV perception task.

Simultaneously, in addition to camera and Lidar sensors which bring about rich semantic information and 3D features respectively, we emphasize the integration of **Radar sensors**, which deliver unique velocity information and robust signals in extreme weather conditions but have received considerably less attention in research compared with other sensing modalities. Among the limited literature that involves Radar learning, some work focuses on utilizing the velocity measurement for prediction [39, 45], while

others treat Radar points as an additional 3D information source to aid detection and tracking [7, 10, 31]. Our method aims to accomplish perception tasks in the BEV space and analyze the synergy among all three types of sensors.

BEVGuide is a *general and flexible multi-modality fusion* framework designed for BEV perception. A paradigm of it is shown in Figure 1. It accommodates any potential sensor configurations within the sensor pool, including but not limited to camera, Lidar, and Radar sensors. For any new sensor, the core fusion block simply requires a sensor-specific feature embedding, which can be obtained from any backbone encoder, whether pretrained or yet to be trained. The fusion block consists of a BEV-guided sensor-agnostic attention module. We split the BEV space into small patches with position-aware embeddings, through which the model queries and fuses all sensor features to generate a unified BEV representation. By employing positional encoding to encapsulate geometric constraints, we avoid error-prone explicit feature space transformations, enabling the model to focus on positions of interest across sensors. Designed in this manner, the core fusion module is *modality-agnostic* and can potentially support any sensor configurations in real-world applications. We evaluate our model in BEV scene segmentation and velocity estimation tasks, where BEVGuide achieves leading results across various sensor configurations. Moreover, We observe that BEVGuide exhibits great robustness in different weather and lighting conditions, facilitated by the inclusion of different sensors.

The main contributions of this paper are as follows. (1) We propose BEVGuide, a comprehensive and versatile multi-modality fusion architecture designed for BEV perception. (2) We underscore the significance of Radar sensors in velocity flow estimation and BEV perception tasks in general, offering an insightful analysis in comparison with camera and Lidar sensors. (3) We present a map-guided multi-sensor cross-attention learning module that is general, sensor-agnostic, and easily extensible. (4) BEVGuide achieves state-of-the-art performance in various sensor configurations for BEV scene segmentation and velocity flow estimation tasks. And in principle, BEVGuide is compatible with a wide range of other BEV perception tasks.

## 2. Related Work

**Camera-Lidar Fusion for 3D Perception.** Considerable research has examined leveraging signals from multiple modalities, especially images and point clouds, for 3D perception tasks. Some work prioritizes one sensor over the other [1, 34, 42, 43], where it first extracts embeddings of the main sensor, and then augments the embeddings by transforming the auxiliary sensor features into the main feature space. Frustum PointNet [36] generates 2D bounding boxes and uses them to guide 3D detection with point clouds. Another line of work [1, 9, 19, 21, 29, 42] extracts point cloud features first, and then fuses 2D RGB features with them. Recent work [3, 18, 44, 47, 49] starts to explore deep feature-level fusion between point and image modalities. Ye *et al.* [46] propose to use feature alignment between point clouds and images to improve monocular 3D object detection. BEVFusion [25] extracts features for Lidar and camera sensors and projects them in the BEV space before fusing them for 3D detection and segmentation tasks. Our method differs from such work, as we leverage a unique BEV-guided architecture to simplify the fusion and allow more flexible sensor combination settings, including the important Radar sensor type.

**Radar for Autonomous Driving.** Despite being sparser and less accurate in angular direction than Lidar, Radar has several appealing properties such as its low cost, robustness to extreme weather, and radial velocity information from the Doppler effect. This has motivated a recent uprising trend to utilize Radar information for autonomous driving, where a lot of work introduces Radar to perception tasks along with camera or Lidar sensors [7, 8, 10, 11, 13, 14, 28, 31, 45]. Some classical tracking based approaches [8, 11, 13] perform Radar fusion using filtering techniques such as Kalman Filters. More recently, several data-driven deep learning approaches are proposed. Radar and cameras are fused for 3D object detection in [7, 31]. Joint 3D detection and velocity estimation is performed in [28] using the raw Range-Azimuth-Doppler tensor. Lidar and Radar are fused in [39, 45] to perform speed estimation and trajectory estimation. Simple-BEV [14] fuses Lidar and cameras for BEV vehicle segmentation, and FUTR3D [10] fuses camera, Lidar, and Radar for more robust 3D object detection. In contrast to these, our method proposes to use Radar velocity in an *end-to-end* manner together with its positional information to benefit a diverse spectrum of driving tasks from BEV scene segmentation to velocity estimation.

**3D Scene Understanding in BEV Frame.** Inferring 3D scenes from the BEV perspective has recently received a large amount of interest due to its practicality and effectiveness. MonoLayout [30] estimates the layout of urban driving scenes from images in the BEV frame and uses an adversarial loss to enhance the learning of hidden objects. Can *et al.* [5, 6] propose to employ graphical representation and temporal aggregation for better inference of the driving scenarios using on-board cameras. In the meanwhile, the BEV perspective also enables the efficient fusion of multiple sensor modalities for scene analysis [16, 33]. Recently, using BEV representation to merge images from multiple camera sensors has become a popular approach in autonomous driving 3D perception. Following the monocular feature projection proposed by Orthographic Feature Transform (OFT) [38], Pyramid Occupancy Networks [37]

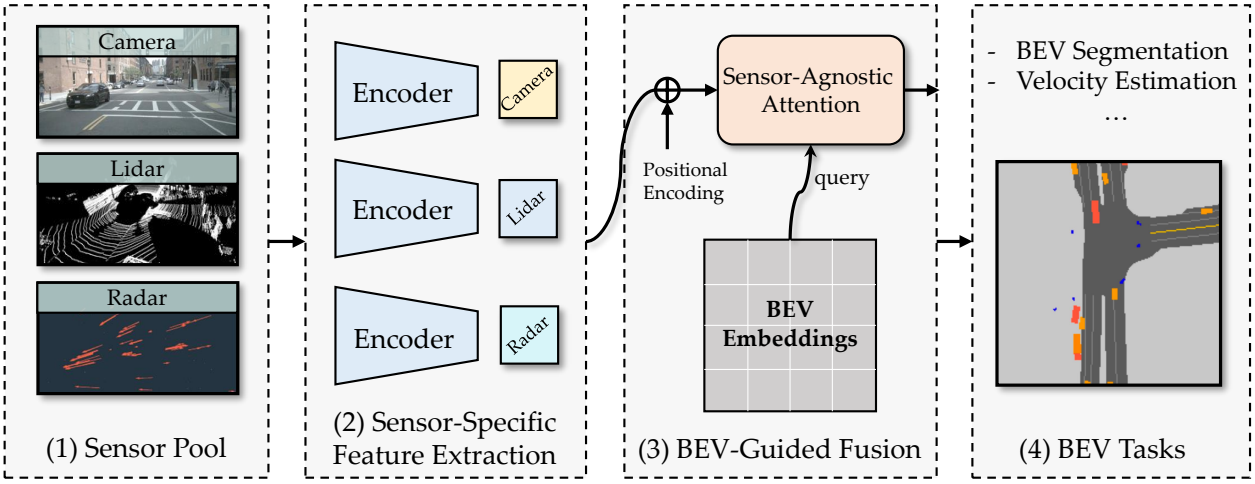


Figure 2. **BEVGuide Paradigm.** Our proposed method is able to work with any collection of sensors and fuse the feature representation using a BEV-guided transformer backbone. (1) BEVGuide takes input from a sensor pool and (2) extracts sensor-specific features, (3) then a BEV-guided multi-sensor fusion model takes queries from BEV embeddings and learns BEV features using a sensor-agnostic attention module, (4) and the learned BEV representation is decoded for final perception tasks.

employ transformer architectures to better convert images into BEV frames. Alternatively, Lift-Splat-Shoot [35] disentangles feature learning and depth inference by learning a depth distribution over pixels to convert camera image features into BEV. Cross-view-transformer [48] leverages transformer to learn BEV segmentation from the multi-view images. Our method stands out from existing models for (1) a more unified architecture that allows the use of more diverse sensor settings including camera, Lidar, and Radar, and (2) a BEV-guided multi-sensor attention mechanism to learn from the modalities in an adaptive manner.

## 3. Approach

BEVGuide considers general and flexible *multi-sensor fusion* (*i.e.,* multi-view cameras, Lidar, and Radar sensors) for *BEV scene perception* (*i.e.,* semantic segmentation and velocity estimation). We provide an overview of our framework in Figure 2. Aiming at working with a wide range of sensor modalities, we first introduce our extendable feature extraction module, including a sensor pool and a group of sensor-specific encoders (Sec 3.1). We transform the sensory features in different coordinates into a unified BEV space using a *BEV-guided multi-sensor attention* module, together with positional encoding (Sec 3.2). Finally, we use the learned BEV feature embedding to conduct BEV scene perception tasks (Sec 3.3).

### 3.1. Flexible Sensor Encoders

Different autonomous robots and vehicles have a diverse collection of available sensors, and therefore, a robust learning architecture should be able to adapt to different modality configurations without too much effort. In this light, we define a sensor pool, which consists of all available sensor inputs for our model – For instance, images from the cameras, point clouds from the Lidar, and points with velocities from the Radar. For each type of the sensor input, a sensor-specific encoder extracts its feature embedding. The encoders can either be fixed or be trained and finetuned end-to-end. The multiple trained embeddings will be used in the subsequent multi-sensor attention module to extract BEV-related information from BEV queries.

**Camera.** To demonstrate the generalization ability of our multi-sensor attention module, we leverage a very common and lightweight convolutional backbone to generate image features from multi-view cameras. The feature embeddings are in the image coordinate as opposed to the BEV space, and we also have camera matrices ready for the subsequent feature transformation.

**Lidar.** We apply pointpillars [20], a lightweight voxel-based encoder for the Lidar point clouds. The resulting feature embeddings are flattened along the height dimension perpendicular to the ground plane, and as such are represented in the BEV coordinates.

**Radar.** Unlike camera and Lidar sensors with plenty of well-explored feature encoders for 3D tasks, Radar sensors do not have a widely accepted architecture for feature extraction. LiRaNet [39] and FUTR3D [10] adopt nearest neighbor search and multi-layer perception (MLP) to extract features from the points, CenterFusion [32] applies pointpillars architecture, and Simple-BEV [14] uses projection and convolution-based encoders. Motivated by these methods, BEVGuide adopts a simple encoder. We first project the Radar points onto the BEV grids, treating Radar

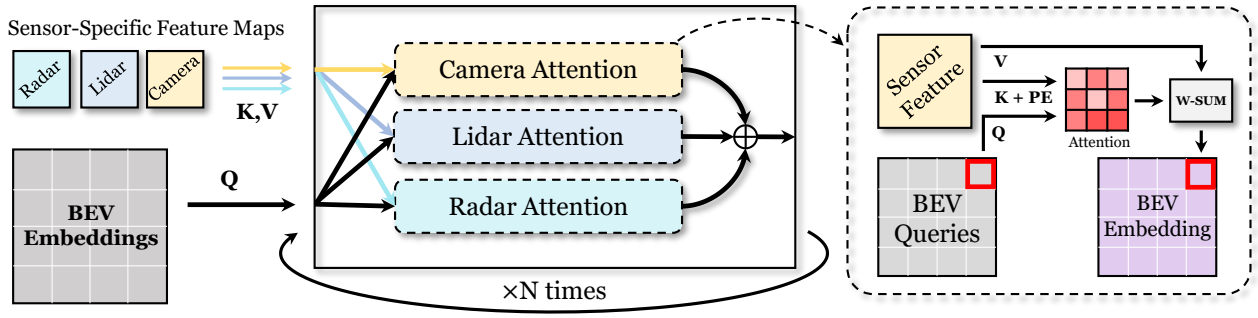


Figure 3. **Overview of our BEV-guided multi-sensor attention. Left:** the module takes queries from BEV embeddings, and takes keys and values from sensor-specific feature maps. We use parallel sensor-agnostic attention modules to extract features from multiple sensors. **Right:** the detailed sensor-agnostic attention module. The BEV queries learn an attention map and extract the final BEV representation from the sensory feature maps. PE and W-SUM stand for positional encoding and weighted summation, respectively. K, V and Q represent keys, values, and queries, respectively.

velocity, intensity, and other indicators as feature channels, and then use two convolutional layers to learn the Radar representation in the BEV coordinates.

We have designed this extendable encoder module such that it can easily accommodate to more or fewer sensors in the sensor pool by adding or removing some sensor-specific encoders without impacting the subsequent multi-sensor attention architecture.

### 3.2. BEV-Guided Multi-Sensor Attention

Sensor features exist in different coordinates. For instance, images are in the 2D camera view, Lidar/Radar points are often in the 3D view, with Radar data sometimes being in the frequency-space view. One of the most critical challenges in sensor fusion is to transform different features into the same view. We adopt BEV coordinates as the unified representation space and use BEV queries to fetch features from sensor-specific feature maps with an attention mechanism. We provide an overview of the multi-sensor attention module in Figure 3.

We partition the BEV map into $H \times W$ 2D patches, each of which represents a certain area of region. For each region, we use a learnable D-dimensional embedding to encode its positional information, which we call a *BEV query*. Given $M$ sensor modalities, each regional embedding queries the sensor-specific feature maps using $M$ separate attention modules to acquire the feature embedding of that region before aggregation. We make each BEV query to attend all locations of a sensor feature map with the help of multiple sensor-agnostic attention modules, which can be arranged in sequence (cascade) or in parallel. With all sensory features from the attention blocks sharing the same BEV coordinates, we fuse them together with an element-wise operator (*i.e.,* addition, concatenation).

**Sensor-Agnostic Attention Block.** Despite the number of different sensors as input into the model, all the attention modules share the same architecture which is internally sensor-agnostic. An illustration of the sensor-agnostic attention block is shown on the right of Figure 3.

Given a sensor feature embedding $f \in \mathbb{R}^{h \times w \times d}$ and a BEV map embedding $b \in \mathbb{R}^{H \times W \times D}$, the attention module treats $b = \{q^{(1)}, q^{(2)}, \cdots\}$ as $H \times W$ independent queries, $f$ as keys and values (with the positional encoding), and generates a BEV feature map $\phi$ of dimension $(H, W, C)$ from the feature map $f$. For each query $q^{(i)}$ in $b$, the model obtains a weight map $\lambda^{(i)}$ for the feature embedding by calculating the dot product similarity between the query and keys. Then the BEV feature of the location represented by the query $q^{(i)}$ is generated by a weighted sum from weight map $\lambda^{(i)}$ and values $f$. The module difference between different sensors lies only in the feature dimension and the construction of positional encoding. Hence, as long as a proper sensor-specific encoder is provided, BEVGuide can take any sensory input without making major changes to the BEV-guided sensor-agnostic attention module.

**Geometry-Aware Positional Embedding.** When calculating the weight map $\lambda^{(i)}$ for the feature embedding, each query scans over the entire feature map. It thus needs guidance about where to focus more and where less. We use positional encoding to provide soft geometric correspondence between BEV query positions and feature map positions. Usually presented in different views, sensor coordinates and BEV coordinates are connected by a transformation matrix in the Special Euclidean Group in 3D (SE(3) transformations matrices). For *image features*, given the intrinsic matrix $K$ and extrinsic matrix $M$, we have the equation between image location $x^{(\mathrm{im})}$ and world (BEV) location $x^{(\mathrm{w})}$,

$$x^{(\mathrm{im})} \simeq KMx^{(\mathrm{w})}, \tag{1}$$

where $\simeq$ represents equality up to a scale ambiguity due to the unknown depth. Then, we construct the image positional embedding as $\mathrm{e}^{(\mathrm{im})} = M^{-1}K^{-1}x^{(\mathrm{im})}$ and BEV query positional embedding directly as $\mathrm{e}^{(\mathrm{w})} = x^{(\mathrm{w})}$. in this way, we introduce the soft geometric correspondence by

giving higher cosine similarity during the attention calculation. For Lidar and Radar sensors, their features are learned on the BEV coordinate which is the same as the queries, so we use the sine function positional encoding [41].

### 3.3. BEV Scene Perception Tasks.

After $N$ sensor-agnostic modules to extract and fuse features for all sensors, we get the final multi-sensor BEV feature representation and use convolution-based decoder heads to output our final objectives. In principle, our model can be applied to and address all BEV perception-related tasks. We use the BEV scene segmentation task to demonstrate the overall perception capability of the surrounding environment, and the BEV velocity flow estimation task to demonstrate the advantage of instantaneous speed measurement from the Radar sensor. We leave other tasks as the future work. The whole BEVGuide pipeline is end-to-end trainable. The training details are provided in Sec. 4.

## 4. Experiments

We evaluate BEVGuide with different sensor combinations on vehicle and road BEV semantic segmentation, velocity estimation, and 3D detection tasks on the large-scale autonomous driving dataset nuScenes [4].

**Dataset.** The nuScenes [4] dataset is a large-scale outdoor dataset collected over a variety of weather and time-of-day conditions. It has 40,157 annotated samples, each of which consists of 6 camera images, 5 Radar point clouds of different views covering approximately the full surrounding angles of the ego-vehicle, and a 360° 32-beam Lidar scan. We use the official nuScenes training/validation split, which contains 28,130 samples in the training set, and 6,019 samples in the validation set. We generate the ground-truth BEV semantic and velocity labels of $(200, 200)$ resolution from the map annotation, bounding boxes, and sensor calibration matrices provided by the dataset.

**Evaluation.** We use a 100m$\times$100m region around the ego-vehicle with 50cm resolution for the BEV map-view evaluation [14,35,48]. For BEV map-view semantic segmentation task, we use the Intersection-over-Union (IoU) score between the prediction and the ground-truth annotation on the vehicle, drivable area, and lane classes as the performance measure. For a fair comparison with prior approaches, we conduct binary segmentation for each class separately and select the highest IoU across different thresholds [25,48].

For the velocity estimation task, we formulate it as a BEV velocity flow semantic task similar to the occupancy flow [27], which we call *BEV velocity flow estimation*. To generate the ground-truth annotations, we first compute the velocity of the moving vehicles in each frame with the bounding box and timestamp information. We compensate the velocity of the ego-vehicle and project the 3D bounding boxes onto the BEV map. For pixels inside a bounding box of a moving object, we assign the velocity of the object to the pixels, and do this for all pixels on the $(200, 200)$ BEV map, which is an alternative to object-wise AVE [4,45]. We call this metric pixel-wise Average Velocity Error (p-AVE) to measure the velocity estimation performance, which is computed as the $l_2$ velocity error averaged over all pixels classified as objects (vehicles).

**Model.** We use EfficientNet [35,40,48] pretrained on ImageNet [12] as our image backbone encoder. We use pointpillars as our Lidar backbone [20], and use the projection based Radar backbone as described in Sec. 3. We downsample the camera images to 28$\times$60, 1/8 of the input size. The Lidar and Radar feature embeddings are both interpolated to 200$\times$200 size in the BEV frame. We use 4-head attention blocks with embedding of 64 channels. The decoder is composed of three 2$\times$ bilinear-upsample layers, each followed by a convolution layer to obtain the final output map of the desired size.

We train our model with focal loss [23] for semantic segmentation and $\ell_2$ loss for velocity estimation task. We optimize the model with AdamW [26], learning rate of 4e-3, and weight decay of 1e-7. The model is trained on a 4-A100 machine with batch size of 4 for 40 epochs. Results for 3D detection is shown in supplementary.

**Baselines.** We compare our method with the state-of-the-art BEV scene semantic segmentation work. We also compare with state-of-the-art BEV 3D object detection baselines, which are marked with $^\ddagger$ in Table 1. For detection baselines, we approximate their segmentation results by taking their pretrained models and project the estimated bounding boxes onto the BEV coordinates to calculate the IoU with the ground-truth BEV segmentation on the vehicle class. This allows us to compare with a wide range of BEV perception baselines. For Camera + Radar fusion, we compare with CenterFusion [32], FUTR3D [10], and Simple-BEV [14]. For Camera + Lidar fusion, we compare with Pointpainting [42], Simple-BEV [14], X-Align [3], and BEVFusion [25]. For the joint fusion of all three sensors, although FUTR3D [10] is the only related work that is designed to support all three sensors, it does not report the results on 3-sensor fusion. Hence, we compare with a simplified version of our model where we take the sensor-specific feature maps, project them onto the BEV coordinates, and directly fuse them by concatenation. We also include some state-of-the-art camera-only BEV segmentation baselines in Table 1 for additional comparison, including OFT [38], LSS [35], FIERY [17], and CVT [48]. More results on different feature backbones (EfficientNet [40], ResNet-101 [15], and Swin-Transformer [24]) can be found in the supplementary.

Table 1. BEVGuide achieves state-of-the-art BEV semantic segmentation performance on the nuScenes validation set for all types of sensor combination. ↑ indicates that higher value is better. 'C,' 'R,' 'L' represent Camera, Radar, and Lidar modalities, respectively. ‡: In addition to the BEV segmentation work, we also compare with some of the best 3D detection approaches. *: our re-implementation.

| Method | Modality | Vehicles ↑ | Drivable Area ↑ | Lane ↑ | mIoU ↑ |
|---|---|---|---|---|---|
| OFT [38] | C | 30.1 | 72.2 | 16.9 | 39.7 |
| Lift-Splat [35] | C | 32.1 | 74.1 | 18.8 | 41.7 |
| FIERY [17] | C | 35.8 | - | - | - |
| CVT [48] | C | 36.0 | 74.3 | 29.4 | 46.6 |
| CenterFusion‡ [32] | C+R | 46.5 | - | - | - |
| FUTR3D‡ [10] | C+R | 46.6 | - | - | - |
| Simple-BEV [14] | C+R | 55.7 | - | - | - |
| **BEVGuide** (Ours) | C+R | 59.2 | 76.7 | 44.2 | 60.0 |
| Pointpainting* [42] | C+L | 60.2 | 75.9 | 41.9 | 59.3 |
| Simple-BEV [14] | C+L | 60.8 | - | - | - |
| BEVFusion [25] | C+L | - | 85.5 | 53.7 | - |
| X-Align [3] | C+L | - | 86.8 | **58.2** | - |
| **BEVGuide** (Ours) | C+L | 76.1 | 86.3 | 56.1 | 72.8 |
| BEVGuide-Simple (Ours) | C+R+L | 76.8 | 81.5 | 45.1 | 67.8 |
| **BEVGuide** (Ours) | C+R+L | **79.0** | **86.9** | 56.2 | **74.0** |

### 4.1. BEV Semantic Segmentation

We first experiment on the nuScenes BEV semantic segmentation task, where BEVGuide achieves superior performance for all sensor combinations involving camera, Lidar, and Radar. Note that some of the existing models [14, 42] are trained for the 3D object detection task, so we transform their results into BEV segmentation by projecting the predicted 3D bounding boxes onto the BEV map frame.

**Camera + Radar.** Radar possesses the great advantage over cameras in providing 3D information at an affordable expense, but it is also sparser than Lidar points. As shown in Table 1, BEVGuide achieves state-of-the-art results in the camera + Radar fusion scenario. Compared with Simple-BEV [14], BEVGuide achieves **3.5**% improvement in vehicle class segmentation. Since CenterFusion [32], Simple-BEV [14], and FUTR3D [10] do not perform scene-level segmentation, we also include some strong camera-only BEV map segmentation baselines [17, 35, 38, 48]. BEVGuide compares favorably against CVT [48] with **13.4**% higher mIoU in scene semantic segmentation. As a result, it demonstrates that despite the sparsity and inaccurate 3D measurement of Radar, BEVGuide is able to exploit the additional sensor and help cameras get a more thorough understanding of the surrounding environment.

**Camera + Lidar.** This is the most commonly researched sensor combination for multi-modality fusion in recent work. We compare BEVGuide with strong BEV segmentation models [3, 14, 25, 42]. As shown in Table 1, BEVGuide achieves leading results in the BEV segmentation task, with the highest IoU in vehicle class and on-par map components performance with BEVFusion [25] and X-Align [3]. Notice that we adopt a *more light-weight* architecture, including an EfficientNet backbone for camera feature extraction compared with the heavier Swin-Transformer in prior work. Hence, our work achieves **24** Frame-per-second (FPS) inference time on an NVIDIA Tesla V100 GPU, compared with 9 FPS for BEVFusion. We argue that the efficiency gain brings great advantages of BEVGuide in reality, and in the meanwhile, using stronger backbones can further improve our performance with a tradeoff in runtime.

Table 2. BEVGuide achieves leading BEV velocity flow estimation performance. ↓ indicates that lower value is better. 'C,' 'R,' 'L' represent Camera, Radar, and Lidar modalities, respectively.

| Method | C | L | R | p-AVE ↓ |
|---|---|---|---|---|
| CVT [48] | ✓ | | | 2.13 |
| Pointpainting [42] | ✓ | ✓ | | 1.90 |
| BEVGuide (Ours) | ✓ | ✓ | | 1.63 |
| **BEVGuide** (Ours) | ✓ | ✓ | ✓ | **0.81** |

**Camera + Radar + Lidar.** Very little existing work explores the joint learning of the three sensor types, despite the existence of supported large public driving datasets [2, 4]. The only recent approach that supports all three modalities is FUTR3D [10], yet the paper does not release the results or the pretrained checkpoint of the three-sensor joint model. We compare with an alternative of BEVGuide, where we replace the BEV-guided multi-sensor attention module with a simple projection- and concatenation-based

Table 3. BEVGuide is robust under different weather and lighting conditions, because of the multi-sensor joint learning. We measure the performance of vehicle class segmentation and velocity estimation error p-AVE. 'C,' 'R,' and L represent camera, Radar, and Lidar modalities, respectively. *Abs Diff.* stands for absolute difference.

| | | IoU ↑ | | | p-AVE ↓ | | | IoU ↑ | | | p-AVE ↓ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Modality | Day | Night | Abs Diff. | Day | Night | Abs Diff. | Sunny | Rainy | Abs Diff. | Sunny | Rainy | Abs Diff. |
| CVT [48] | C | 40.4 | 18.8 | 21.6 | 1.99 | 2.71 | 0.72 | 37.3 | 28.1 | 9.2 | 2.03 | 2.59 | 0.56 |
| BEVGuide | C+L | 76.7 | 58.8 | 17.9 | 1.57 | 1.98 | 0.38 | 77.0 | 69.9 | 7.1 | 1.55 | 2.06 | 0.51 |
| BEVGuide | C+L+R | **79.5** | **64.2** | **15.3** | **0.80** | **0.86** | **0.06** | **80.7** | **74.6** | **6.1** | **0.79** | **0.87** | **0.08** |

fusion module. As shown in Table 1, our full model achieves leading performance for all classes in BEV segmentation. The gain in vehicle perception is **2.9**% higher than our camera + Lidar fusion alternative, **19.8**% higher than our camera + Radar fusion, and **43.0**% higher than the camera-only backbone.

### 4.2. BEV Velocity Flow Estimation

In addition to the sparse and cost-effective nature, the Radar sensor can measure the velocity of the objects using the Doppler effect, which is a unique feature that camera and Lidar sensors do not possess. To evaluate the usefulness of the velocity measurement, we further conduct the experiment of BEV velocity estimation. For each pixel in the BEV velocity map, if it lies inside a moving object, then its velocity is defined as the velocity of that object. In this way, we can "project" the object-level velocity vector into the BEV map. As shown in Table 2, significant improvement is observed when additional Radar sensor measurements are taken into account by BEVGuide. Our full model achieves **50.3**% better average velocity error than the camera + Lidar alternative. This marks the importance of the Radar sensor in providing instantaneous speed measurement and validates that BEVGuide extracts meaningful features from the Radar data despite them being sparse and noisy.

### 4.3. Ablation and Analysis

We further test BEVGuide over different weather and lighting conditions and in the challenging sensor failure scenarios, and also analyze our proposed modules and various other design choices.

**Weather and Lighting Effects.** Another advantage of multi-sensor joint learning is the increased robustness in different environmental conditions. In Table 3, we investigate the performance of BEVGuide and the baseline camera model CVT [48] under different weather and lighting conditions. The low-lighting condition poses great challenges for camera-only perception models, because camera relies purely on ambient light as opposed to Lidar and Radar sensors that have active imaging mechanism. BEVGuide not only performs best on both Day and Night scenarios, but also successfully closes the gap between two different lighting conditions by leveraging multi-sensor complementary information (Abs Diff decreases from **21.6** to **15.3** for IoU and from **0.72** to **0.06** for p-AVE). Meanwhile, perception in rainy weather is challenging for the Lidar sensor due to significant sensor noises. BEVGuide also closes the gap between sunny and rainy perception performance with the help of camera and Radar sensors.

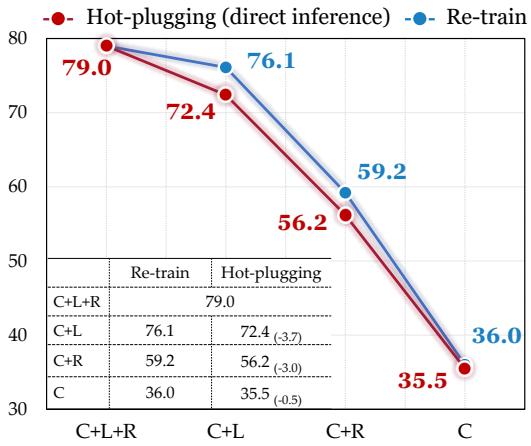


Figure 4. BEVGuide performs great in the challenging sensor failure cases (*hot-plugging*) without re-training the model.

**Robustness to Sensor Failures.** One of the main motivations of this paper is to design a flexible architecture that can adapt to sensor failures, glitches, or missing cases. Here, we present a pilot study of *sensor hot-plugging*, where we train BEVGuide with camera, Lidar, and Radar sensors, but only input fewer types of sensors to the model for inference *without re-training the model*. To simulate the sensor failure cases, we simply replace the original sensor-specific feature map with all-zero tensors. As shown in Figure 4, BEVGuide achieves decent results, compared with alternative models that are trained on the available type of sensors. In all three simulated sensor failure settings, BEVGuide drops no more than 5.1% in vehicle segmentation IoU. We also notice that with a small amount of finetuning (less than 5% of the original training time), BEVGuide reaches back the best performance. This experiment demonstrates the generalization capability and robustness of BEVGuide,

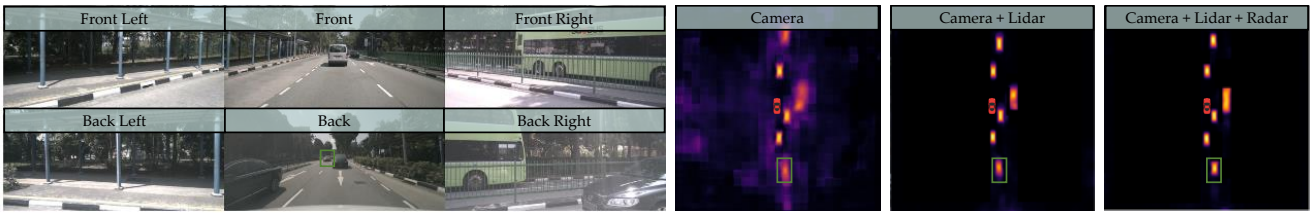


Figure 5. Qualitative results of BEVGuide on vehicle segmentation. Lidar helps the model better locate objects in general, and Radar further improves the perception of distant objects. The bright (yellow) region means high probability of being vehicle and vice versa.

Table 4. Ablation study validates that our various design choices improve the final performance. Default settings are marked in gray.

(a) Object Distance (IoU↑)

| | 0-20m | 20-35m | 35-50m |
|---|---|---|---|
| C | 52.7 | 33.1 | 19.2 |
| C+L | 84.4 | 72.8 | 59.7 |
| C+L+R | **84.5** | **74.1** | **62.0** |

(b) Cross-Attention Strategy

| | IoU↑ | p-AVE↓ |
|---|---|---|
| None | 76.8 | 1.01 |
| Series | 78.2 | 0.84 |
| Parallel | **79.0** | **0.81** |

(c) Aggregated Radar Frames

| | IoU↑ | p-AVE↓ |
|---|---|---|
| 1 frame ($0s$) | 78.1 | 1.18 |
| 4 frames ($0.25s$) | 78.8 | 0.93 |
| 7 frames ($0.5s$) | **79.0** | **0.81** |

(d) Size of BEV Grids

| | IoU↑ | p-AVE↓ |
|---|---|---|
| $10 \times 10$ (10m×10m) | 68.6 | 1.08 |
| $15 \times 15$ (7m×7m) | 77.3 | 0.90 |
| $25 \times 25$ (4m×4m) | 79.0 | **0.81** |
| $50 \times 50$ (2m×2m) | **79.8** | 0.90 |

(e) Positional Encodings (PE)

| | IoU↑ | p-AVE↓ |
|---|---|---|
| None (direct projection) | 65.1 | 0.94 |
| Above + Camera PE | 76.8 | 1.01 |
| Above + Lidar PE | 77.5 | 1.03 |
| Above + Radar PE | **79.0** | **0.81** |

which are important for real-world driving challenges like sensor hot-swapping and sensor failures.

**Ablation Studies.** In Table 4, we present ablation experiments to validate our proposed modules and design choices. For BEV semantic segmentation task, we only report the vehicle class, and we train the model for fewer epochs for faster convergence. In Table 4a, we observe that BEVGuide brings larger improvements to the fewer- or uni-sensor models for objects which are more distant from the ego-vehicle. In Table 4b, we observe that the BEV-guided sensor-agnostic attention module improves the simple feature map concatenation by 2.2% IoU in the segmentation task and 0.2 p-AVE in the velocity estimation task. We also find that the parallel arrangement of attention modules is more favorable against the series (cascade) arrangement, where we put camera, Lidar, and Radar attention modules in a row and learn embeddings one by one. We argue that this is because the parallel arrangement removes the human prior on sensor ordering, such that the BEV queries will learn the attention of all sensor feature maps equally. In Table 4c, we also find that similar to that in the Lidar sensor, aggregating multiple frames of Radar points also helps improve the two BEV perception tasks, especially for the velocity flow estimation task because it relies mostly on the Radar sensor. We also notice in Table 4d that the results of two tasks improve as the size of BEV grids increases. However, because the multi-sensor attention module grows quadratically with the grid size, we use the $25\times25$ grid size by default. Table 4e demonstrates that the positional encoding strategies we use in BEVGuide is useful in guiding the attention module to adaptively learn the geometric correspondence between different views to the BEV coordinates.

**Qualitative Results.** Figure 5 shows qualitative results on BEV vehicle segmentation. We observe that the perception quality of the camera model deteriorates quickly as the distance to the ego-vehicle increases. The Lidar sensor greatly helps the camera-only model to locate objects on the BEV plane with its accurate 3D points. And the Radar sensor further improves the perception of distant objects (*i.e.*, the vehicle in the green box) with its longer working range. This observation is consistent with what we find in Table 4a.

## 5. Conclusion

In this paper, we proposed BEVGuide – a unified sensor fusion architecture – to estimate scene representation in the BEV frame. To achieve this, we design modularized sensor-specific encoders to extract features from a diverse sensor pool, and propose a BEV-guided sensor-agnostic attention module to learn the BEV scene representation from the feature maps. Results on large-scale datasets with a wild spectrum of sensor configurations demonstrated the effectiveness of our BEVGuide, which marks a significant step towards efficient and robust 3D scene perception.

**Acknowledgement.** This work was supported in part by NSF Grant 2106825, NIFA Award 2020-67021-32799, the Jump ARCHES endowment, the NCSA Fellows program, the IBM-Illinois Discovery Accelerator Institute, the Illinois-Insper Partnership, and the Amazon Research Award. This work used NVIDIA GPUs at NCSA Delta through allocation CIS220014 from the ACCESS program.

## References


[1] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3D object detection with transformers. In *CVPR*, 2022. 1, 2

[2] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The Oxford radar robotcar dataset: A radar extension to the Oxford robotcar dataset. In *ICRA*, 2020. 6

[3] Shubhankar Borse, Marvin Klingner, Varun Ravi Kumar, Hong Cai, Abdulaziz Almuzairee, Senthil Yogamani, and Fatih Porikli. X-Align: Cross-Modal Cross-View Alignment for Bird's-Eye-View Segmentation. In *WACV*, 2023. 2, 5, 6

[4] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, and Qiang Xu. nuScenes: A Multimodal Dataset for Autonomous Driving. In *CVPR*, 2020. 5, 6

[5] Yigit Baran Can, Alexander Liniger, Danda Pani Paudel, and Luc Van Gool. Structured Bird's-Eye-View Traffic Scene Understanding from Onboard Images. In *ICCV*, 2021. 2

[6] Yigit Baran Can, Alexander Liniger, Ozan Unal, Danda Pani Paudel, and Luc Van Gool. Understanding Birds-Eye View of Road Semantics using an Onboard Camera. *RA-L*, 2022. 2

[7] Simon Chadwick, Will Maddern, and Paul Newman. Distant vehicle detection using radar and vision. In *ICRA*, 2019. 2

[8] Ricardo Omar Chavez-Garcia and Olivier Aycard. Multiple sensor fusion and classification for moving object detection and tracking. *IEEE Transactions on Intelligent Transportation Systems*, 2015. 2

[9] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3D object detection network for autonomous driving. In *CVPR*, 2017. 2

[10] Xuanyao Chen, Tianyuan Zhang, Yue Wang, Yilun Wang, and Hang Zhao. Futr3D: A unified sensor fusion framework for 3D detection. *arXiv preprint arXiv:2203.10642*, 2022. 1, 2, 3, 5, 6

[11] Hyunggi Cho, Young-Woo Seo, BVK Vijaya Kumar, and Ragunathan Raj Rajkumar. A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In *ICRA*, 2014. 2

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[13] Daniel Göhring, Miao Wang, Michael Schnürmacher, and Tinosch Ganjineh. Radar/lidar sensor fusion for car-following on highways. In *The 5th International Conference on Automation, Robotics and Applications*, 2011. 2

[14] Adam W Harley, Zhaoyuan Fang, Jie Li, Rares Ambrus, and Katerina Fragkiadaki. Simple-BEV: What Really Matters for Multi-Sensor BEV Perception? In *ICRA*, 2023. 2, 3, 5, 6

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[16] Noureldin Hendy, Cooper Sloan, Feng Tian, Pengfei Duan, Nick Charchut, Yuesong Xie, Chuang Wang, and James Philbin. Fishing net: Future inference of semantic heatmaps in grids. In *CVPR*, 2020. 2

[17] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. FIERY: Future Instance Prediction in Bird's-Eye View From Surround Monocular Cameras. In *ICCV*, 2021. 5, 6

[18] Tengteng Huang, Zhe Liu, Xiwu Chen, and Xiang Bai. Epnet: Enhancing point features with image semantics for 3D object detection. In *ECCV*, 2020. 2

[19] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3D proposal generation and object detection from view aggregation. In *IROS*, 2018. 2

[20] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019. 3, 5

[21] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3D object detection. In *CVPR*, 2019. 2

[22] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3D object detection. In *ECCV*, 2018. 1

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 5

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 5

[25] Zhijian Liu, Haotian Tang, Alexander Amini, Xingyu Yang, Huizi Mao, Daniela Rus, and Song Han. BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation. In *ICRA*, 2023. 1, 2, 5, 6

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *ICLR*, 2017. 5

[27] Reza Mahjourian, Jinkyu Kim, Yuning Chai, Mingxing Tan, Ben Sapp, and Dragomir Anguelov. Occupancy flow fields for motion forecasting in autonomous driving. *RA-L*, 2022. 5

[28] Bence Major, Daniel Fontijne, Amin Ansari, Ravi Teja Sukhavasi, Radhika Gowaikar, Michael Hamilton, Sean Lee, Slawomir Grzechnik, and Sundar Subramanian. Vehicle detection with automotive radar using deep learning on range-azimuth-doppler tensors. In *ICCVW*, 2019. 2

[29] Yunze Man, Xinshuo Weng, Prasanna Kumar Sivakumar, Matthew O'Toole, and Kris M Kitani. Multi-Echo LiDAR for 3D Object Detection. In *ICCV*, 2021. 2

[30] Kaustubh Mani, Swapnil Daga, Shubhika Garg, Sai Shankar Narasimhan, Madhava Krishna, and Krishna Murthy Jatavallabhula. Monolayout: Amodal scene layout from a single image. In *WACV*, 2020. 2

[31] Michael Meyer and Georg Kuschk. Deep learning based 3d object detection for automotive radar and camera. In *European Radar Conference (EuRAD)*, 2019. 2

[32] Ramin Nabati and Hairong Qi. Centerfusion: Center-based radar and camera fusion for 3D object detection. In *WACV*, 2021. 3, 5, 6

[33] Bowen Pan, Jiankai Sun, Ho Yin Tiga Leung, Alex Andonian, and Bolei Zhou. Cross-view semantic segmentation for sensing surroundings. *RA-L*, 2020. 2

[34] Jinhyung Park, Xinshuo Weng, Yunze Man, and Kris Kitani. Multi-Modality Task Cascade for 3D Object Detection. In *BMVC*, 2021. 2

[35] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D. In *ECCV*, 2020. 3, 5, 6

[36] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3D object detection from rgb-d data. In *CVPR*, 2018. 2

[37] Thomas Roddick and Roberto Cipolla. Predicting semantic map representations from images using pyramid occupancy networks. In *CVPR*, 2020. 2

[38] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3D object detection. *arXiv preprint arXiv:1811.08188*, 2018. 2, 5, 6

[39] Meet Shah, Zhiling Huang, Ankit Laddha, Matthew Langford, Blake Barber, sida zhang, Carlos Vallespi-Gonzalez, and Raquel Urtasun. LiRaNet: End-to-End Trajectory Prediction using Spatio-Temporal Radar Fusion. In *CoRL*, 2021. 1, 2, 3

[40] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 5

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 5

[42] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3D object detection. In *CVPR*, 2020. 1, 2, 5, 6

[43] Zhixin Wang and Kui Jia. Frustum ConvNet: Sliding Frustums to Aggregate Local Point-Wise Features for Amodal. In *IROS*, 2019. 1, 2

[44] Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris M Kitani. GNN3DMOT: Graph neural network for 3D multi-object tracking with 2D-3D multi-feature learning. In *CVPR*, 2020. 2

[45] Bin Yang, Runsheng Guo, Ming Liang, Sergio Casas, and Raquel Urtasun. Radarnet: Exploiting radar for robust perception of dynamic objects. In *ECCV*, 2020. 1, 2, 5

[46] Xiaoqing Ye, Liang Du, Yifeng Shi, Yingying Li, Xiao Tan, Jianfeng Feng, Errui Ding, and Shilei Wen. Monocular 3D object detection via feature domain adaptation. In *ECCV*, 2020. 2

[47] Jin Hyeok Yoo, Yeocheol Kim, Ji Song Kim, and Jun Won Choi. 3D-CVF: Generating Joint Camera and LiDAR Features Using Cross-View Spatial Feature Fusion for 3D Object Detection. In *ECCV*, 2020. 2

[48] Brady Zhou and Philipp Krähenbühl. Cross-view Transformers for real-time Map-view Semantic Segmentation. In *CVPR*, 2022. 3, 5, 6, 7

[49] Ming Zhu, Chao Ma, Pan Ji, and Xiaokang Yang. Cross-Modality 3D Object Detection. In *WACV*, 2021. 2