

CramNet: Camera-Radar Fusion with Ray-Constrained Cross-Attention for Robust 3D Object Detection

Jyh-Jing Hwang, Henrik Kretzschmar, Joshua Manela, Sean Rafferty,
Nicholas Armstrong-Crews, Tiffany Chen, Dragomir Anguelov

Waymo

Abstract. Robust 3D object detection is critical for safe autonomous driving. Camera and radar sensors are synergistic as they capture complementary information and work well under different environmental conditions. Fusing camera and radar data is challenging, however, as each of the sensors lacks information along a perpendicular axis, that is, depth is unknown to camera and elevation is unknown to radar. We propose the camera-radar matching network CramNet, an efficient approach to fuse the sensor readings from camera and radar in a joint 3D space. To leverage radar range measurements for better camera depth predictions, we propose a novel ray-constrained cross-attention mechanism that resolves the ambiguity in the geometric correspondences between camera features and radar features. Our method supports training with sensor modality dropout, which leads to robust 3D object detection, even when a camera or radar sensor suddenly malfunctions on a vehicle. We demonstrate the effectiveness of our fusion approach through extensive experiments on the RADIATE dataset, one of the few large-scale datasets that provide radar radio frequency imagery. A camera-only variant of our method achieves competitive performance in monocular 3D object detection on the Waymo Open Dataset.

Keywords: Sensor fusion; cross attention; robust 3D object detection.

1 Introduction

3D object detection that is robust to different weather conditions and sensor failures is critical for safe autonomous driving. Fusion between camera and radar sensors stands out as they are both relatively resistant to various weather conditions [2] compared to the popular lidar sensor [3]. A fusion design that naturally accepts single-sensor failures (lidar, radar, or camera or radar) is thus desired and boosts safety in an autonomous driving system (Figure 1).

Most sensor fusion research has focused on fusion between lidar and another sensor [32,54,7,11,50,51,19,11,31,57,39] because lidar provides complete geometric information, i.e., azimuth, range, and elevation. Sparse correspondences between

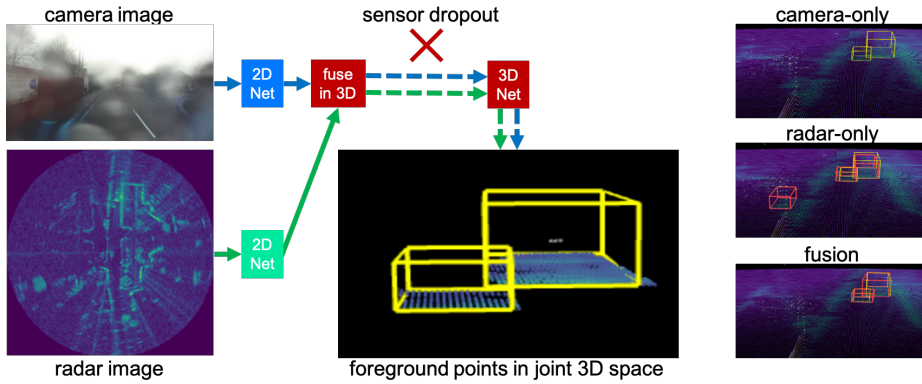


Fig. 1: Our approach takes as input a camera image (top left) and a radar RF image (bottom left). The model then predicts foreground segmentation for both native 2D representations before projecting the foreground points with features into a joint 3D space (middle bottom) for sensor fusion. Finally, the method runs sparse convolutions in the joint space for 3D object detection. The network architecture naturally supports training with sensor dropout. This allows the resulting model to cope with sensor failures at inference time as it can run on camera only and radar only input depending on which sensors are available.

lidar and another sensor is thus well defined, making lidar an ideal carrier for fusion. On the other hand, even though camera and radar sensors are lighter and cheaper, consume less power, and endure longer than lidar, camera-radar fusion is understudied. Camera-radar fusion is especially challenging as each sensor lacks information along one perpendicular axis: depth unknown for camera and elevation unknown for emerging imaging radar, as summarized in Table 1. Radar produces radio frequency (RF) imagery that encodes the environment approximately in the bird’s-eye view (BEV) with various noise patterns, an example shown in Figure 1. As a result, camera data (in perspective view) and radar data (in BEV) form many-to-many mappings and the exact matching is unclear from geometry alone.

To solve the matching problem, we consider three possible schemes for fusion: **(1) Perspective view primary** [32]: This scheme implies we trust the depth reasoning from the perspective view. One can project camera pixels to their 3D locations with depth estimates and find their vertical nearest neighbors of corresponding radar points. If depth is unknown, one can project a pixel along a ray in 3D and perform matching. **(2) Bird’s-eye view primary** [50]: This scheme implies we trust the elevation reasoning from the bird’s-eye view. However, since it’s difficult to predict elevation from radar imagery directly, one might borrow elevation information from the map. Hence, the inferred elevation for radar is sometimes inaccurate, resulting in rare usage unless LiDAR is available. **(3) Cross-view matching** [13]: This scheme implies we perform matching in a joint 3D space. For example, one can use supplementary information (map or camera depth estimation) to upgrade camera and radar 2D image pixels to 3D point clouds (with some uncertainty) and perform matching between point clouds

Sensor	Azimuth	Range	Elevation	Resistance to weather	3D detection literature
Camera	✓	x	✓	medium	abundant
Radar	✓	✓	x*	high	scarce
Lidar	✓	✓	✓	low	abundant

Table 1: Characteristics of major sensors commonly used for autonomous driving. Both camera and radar tend to be less affected by inclement weather compared to lidar scanners. However, whereas regular camera does not directly measure range, radar does not measure elevation. This poses a unique challenge for fusing camera and radar readings as the geometric correspondences between the two sensors are underconstrained. Overall, camera-radar fusion is still underexplored in the literature. *Although there exists radars with elevation, this paper focuses on planar radar which, at the moment, is more common for automotive radar.

directly. This is supposedly the most powerful scheme if we can properly handle uncertainties. Our architecture is designed to enable this matching scheme, hence we name it CramNet (Camera and RADar Matching Network).

Since the effectiveness of projecting into 3D space heavily relies on accurate camera depth estimates, we propose a ray-constrained cross-attention mechanism to leverage radar for better depth estimation. The idea is to match radar responses along each camera ray emitted from a pixel. The correct projection should be the locations where radar senses reflections. Our architecture is further designed to accept sensor failures naturally. As shown in Figure 1, the model is able to operate even when one of the modalities is corrupted during inference. To this end, we incorporate sensor dropout [7,52] in the point cloud fusion stage during training to boost the sensor robustness.

We summarize the contributions of this paper as follows:

1. We present a camera-radar fusion architecture for 3D object detection that is flexible enough to fall back to a single sensor modality in the event of a sensor failure.
2. We demonstrate that the sensor fusion model effectively leverages data from both sensors as the model outperforms both the camera-only and the radar-only variants significantly.
3. We propose a ray-constrained cross-attention mechanism that leverages the range measurements from radar to improve camera depth estimates, leading to improved detection performance.
4. We incorporate sensor dropout during training to further improve the accuracy and the robustness of camera-radar 3D object detection.
5. We demonstrate state-of-the-art radar-only and camera-radar detection performance on the RADIATE dataset [40] and competitive camera-only detection performance on the Waymo Open Dataset [47].

2 Related Work

Camera-based 3D object detection. Monocular camera 3D object detection is first approached by directly extending 2D detection architectures and

incorporating geometric relationships between the 2D perspective view and 3D space [6,27,4,43,23,44,8,16]. Utilizing pixel-wise depth maps as an additional input shows improved results, either for lifting detected boxes [26,42] or projecting image pixels into 3D point clouds [53,24,58,9,55] (also known as Pseudo-LiDAR [53]). More recently, another camp of methods emerge to be promising, i.e., projecting intermediate features into BEV grid features along the projection ray without explicitly forming 3D point clouds [36,46,34,18].

The BEV grid methods benefit from naturally expressing the 3D projection uncertainty along the depth dimension. However, these methods suffer from significantly increased compute requirements as the detection range expands. In contrast, we model the depth uncertainty through sampling along the projection ray and consulting radar features for more accurate range signals. This also enables the adoption of foreground extraction that allows a balanced trade-off between detection range and computation.

Radar-based 3D object detection. Frequency modulated continuous wave (FMCW) radar is usually presented by two kinds of data representations, i.e., radio frequency (RF) images and radar points. The RF images are generated from the raw radar signals using a series of fast Fourier transforms that encode a wide variety of sensing context whereas the radar points are derived from these RF images through a peak detection algorithm, such as Constant False Alarm Rate (CFAR) algorithm [35]. The downside of the radar points is that recall is imperfect and the contextual information of radar returns is lost, with only the range, azimuth and doppler information retained. As a result, radar points are not suitable for effective single modality object detection [38,33], which is why most works use this data format only to foster fusion [2,13,29,28]. On the other hand, the RF images maintain rich environmental context information and even complete object motion information to enable a deep learning model to understand the semantic meaning of a scene [25,40]. Our work is therefore built upon radar RF images and can produce reasonable 3D object detection predictions with radar-only inputs.

Sensor fusion for 3D object detection. Sensor fusion for 3D object detection has been studied extensively using lidar and camera. The reasons are twofold: 1) Lidar scans provide comprehensive representations in 3D for inferring correspondences between sensors, and 2) camera images contain more semantic information to further boost the recognition ability. Various directions have been explored, such as image detection in 2D before projecting into frustums [32,54], two-stage frameworks with object-centric modality fusion [7,11,17], image feature-based lidar point decoration [50,51], or multi-level fusion [19,11,31]. Since sparse correspondences between camera and lidar are well defined, fusion is mostly focused on integrating information rather than matching points from different sensors.

As a result, these fusion techniques are not directly applicable to camera-radar fusion where associations are underconstrained. Early work, Lim et al. [20], applies feature fusion directly between camera and radar features without any geometric considerations. Recently, more works tend to leverage camera models and geometry for association. For example, CenterFusion [28] creates camera

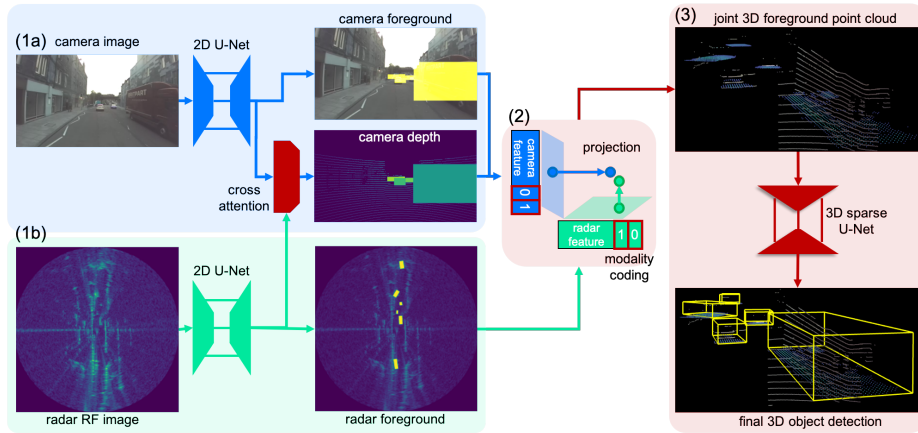


Fig. 2: Architecture overview. Our method can be partitioned into three stages: (1a) camera 2D foreground segmentation and depth estimation, (1b) radar 2D foreground segmentation, (2) projection from 2D to 3D and subsequent point cloud fusion, and (3) 3D foreground point cloud object detection. The cross-attention mechanism modifies the camera depth estimation by consulting radar features, as further illustrated in Figure 3. The modality coding module appends a camera or radar binary code to the features that are fed into the 3D stage, enabling sensor dropout and enhancing robustness. We depict the camera stream in blue, the radar stream in green, and the fused stream in red.

object proposal frustums to associate radar features and GRIF Net [13] projects 3D RoI to camera perspective and radar BEV to associate features. Our model, on the other hand, fuses camera-radar data in a joint 3D space with the flexibility to perform 3D detection with either single modality, leading to increased robustness.

3 CramNet for Robust 3D Object Detection

We describe the overall architecture for camera-radar fusion in Section 3.1. In Section 3.2, We then introduce a ray-constrained cross-attention mechanism to leverage radar for better camera 3D point localization. Finally, we propose sensor dropout that can be integrated seamlessly into the architecture in Section 3.3 to further improve the robustness of 3D object detection.

3.1 Overall Architecture

Our model architecture, in Figure 2, is inspired by Range Sparse Net (RSN) [48], which is an efficient two-stage lidar-based object detection framework. The RSN framework takes input of perspective range images, segments perspective foreground pixels, extracts 3D (BEV) features on foreground regions using sparse convolution [56], and performs CenterNet-style [60] detection. We adapt the framework for camera-radar fusion and the overall architecture can be partitioned

into three stages: (1) 2D foreground segmentation, (2) 2D to 3D projection and point cloud fusion, and (3) 3D foreground point cloud detection.

Stage 1: 2D foreground segmentation. The goal of this stage is to perform efficient foreground segmentation for native dense representations from two modalities. This allows us to restrict the expensive 3D operations to foreground points. The network takes as input a pair of camera images \mathbf{I}_C and radar RF images \mathbf{I}_R . We then employ two identical lightweight U-Nets [37] to extract 2D features and predict foreground segmentation masks for each modality, \mathbf{F}_C and \mathbf{F}_R , respectively. For camera image feature extraction, one can also adopt a more powerful, multi-scale feature extractor, such as a feature pyramid network [21]. The detailed design of the U-Net can be found in the supplementary.

To train such a segmentation network (for both camera and radar), we use the 2D projection of 3D bounding box labels as ground truth – a pixel belongs to the foreground class if it falls inside any of the projected 2D boxes. This might introduce some noise as background pixels sometimes fall within a box, but we find that this noise is insignificant in practice. We then apply a pixel-wise focal loss [22] to classify each pixel:

$$L_{\text{seg}} = \frac{-1}{N} \left(\sum_{i \in \mathbf{F}} (1 - p_i)^{\gamma_s} \log(p_i) + \sum_{i \in \mathbf{B}} p_i^{\gamma_s} \log(1 - p_i) \right), \quad (1)$$

where N is the total number of pixels, \mathbf{F} and \mathbf{B} are the sets of foreground and background pixels, and p_i is the model’s estimated probability of foreground for pixel i . The hyperparameter γ_s controls the penalty reduction. A pixel with foreground score higher than τ will be selected. Since the 3D stage can resolve false positives, whereas false negatives cannot be recovered, we typically set a low value for τ to attain high recall.

Stage 2: 2D to 3D projection and point cloud fusion. Once we obtain the foreground pixels, we project them into 3D for the following 3D stage. For the camera projection, we predict a depth value for each pixel from the same U-Net with additional convolutional layers. The depth ground truth is obtained by projecting lidar points to the camera view and overlaying them with depth values from projected ground truth 3D boxes. The use of depth from ground truth boxes is to enable 3D detection where lidar data alone is insufficient. This is especially true outside of lidar range, as well as when lidar points are deteriorated due to weather. We train the depth estimation using pixelwise L2 losses on valid regions, or L_{depth} . The camera projection relies on the camera model, i.e., the intrinsics and extrinsics, with depth to infer the 3D location of each pixel.

For radar projection, we use the radar model to transform radar BEV points to 3D using the sensor height as elevation. If map is available, the road elevation can be used to offset this value to handle non-planar scenes like hills.

There are several options to combine the camera and radar 3D point clouds. One plausible choice is to select one modality as a major sensor and gather features from the other modality. This is usually how researchers fuse lidar with other sensors [50]. However, the drawback is obvious: the major sensor is a single point of failure. Instead, we directly place two point clouds in a joint 3D

space. We align the feature dimensions of both modalities and append a modality code to the feature so that the 3D network can leverage the multi-modality information easily. The major benefit is to enable robust detection especially when one modality fails to perform.

Stage 3: 3D foreground point cloud detection. We apply dynamic voxelization [61] on the fused foreground point cloud, whose features are then encoded into sparse voxel features. A 2D or 3D sparse convolution network [10] (for pillar style [15], or 3D voxelization, respectively) is applied on the sparse voxels. The network details can be found in the supplementary.

We follow RSN [48] for CenterNet-style [60] 3D box regression. We calculate a ground truth objectness heatmap for every point $x \in \mathbb{R}^3$: $h(x) = \max\{\exp(-\frac{\|x-c\|-\|x_c-c\|}{\sigma^2}) \mid c \in C(x)\}$ where $C(x)$ is the set of centers of boxes containing x , x_c is the closest point to box center c , and σ is a constant. In other words, the objectness of a point is inversely related to its distance to the closest box center. We train the network to predict a heatmap using a focal loss [22]:

$$L_{\text{hm}} = \frac{-1}{N} \sum_x \left((1 - \tilde{h}(x))^{\gamma_h} \log(\tilde{h}(x)) \mathbb{1}(h > 1 - \epsilon_h) + (1 - h(x))_{\tilde{h}}^{\alpha_h} \tilde{h}(x)^{\gamma_h} \log(1 - \tilde{h}(x)) \mathbb{1}(h \leq 1 - \epsilon_h) \right), \quad (2)$$

where $\mathbb{1}(\cdot)$ is the indicator function, h and \tilde{h} are the ground truth and predicted heat map, $(1 - \epsilon_h)$ decides the threshold for ground truth objectness, and α_h and γ_h are hyperparameters in the focal loss.

The 3D boxes are parameterized as $\mathbf{b} = (b_x, b_y, b_z, l, w, h, \theta)$ where b_x, b_y, b_z are the offsets of a 3D box center relative to a voxel center, and l, w, h, θ are the length, width, height, and heading of a box. All the box parameters are trained with smooth L1 losses except for the heading that is trained with a bin loss [41]. An additional IoU loss [59] is employed for better accuracy. The box regression loss is as follows:

$$L_{\text{box}} = \frac{1}{B} \sum_i \left(L_{\text{SmoothL1}}(\mathbf{b}_i \setminus \theta_i - \tilde{\mathbf{b}}_i \setminus \tilde{\theta}_i) + L_{\text{bin}}(\theta_i, \tilde{\theta}_i) + L_{\text{IoU}_i} \right), \quad (3)$$

where B is the total number of boxes with ground truth heatmap value greater than a threshold τ_{hm} , and \mathbf{b}_i and $\tilde{\mathbf{b}}_i$ denote the ground truth and prediction for box \mathbf{b}_i , respectively; the same for θ_i . For more details on heatmap and box regression, we refer interested readers to RSN [48].

We train the fusion network end-to-end with losses summarized as:

$$L = \lambda_{\text{seg}} L_{\text{seg}} + \lambda_{\text{depth}} L_{\text{depth}} + \lambda_{\text{hm}} L_{\text{hm}} + L_{\text{box}}, \quad (4)$$

where λ_* are hyperparameters for the respective loss weighting.

3.2 Ray-Constrained Cross-Attention

It is widely known [30,53] that camera-based 3D object detection relies heavily on accurate depth estimation, either explicitly or implicitly through BEV

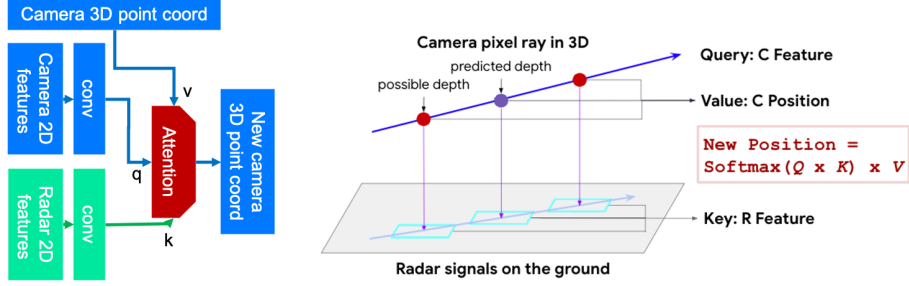


Fig. 3: The proposed ray-constrained cross-attention mechanism resolves the ambiguity in the geometric correspondences between camera features and radar features. Following the Transformer [49], we take camera features as queries and radar features as keys to transform 3D camera points as values.

representations. Luckily, for camera-radar fusion, we don’t have to rely solely on camera to infer depth as radar provides relatively accurate range estimates. To utilize the complementary sensing directions, we propose a ray-constrained cross attention mechanism to leverage radar for improving camera 3D point localization, illustrated in Figure 3.

Our observation is that an optimal 3D location for each foreground camera pixel usually accompanies a corresponding peak response from radar. Thus we propose to consult radar features along a camera 3D ray, emitted from each pixel, to rectify the camera 3D point location after projection. Since there is infinite possible locations, we perform sampling along the ray, centered at the initial depth estimation. The final 3D location is decided by matching between camera and radar features among these sampled locations.

We denote the projected camera 3D location from a depth estimate \tilde{d}_i for pixel i as $M(\tilde{d}_i)$. We sample s points farther and closer around the estimated location respectively, or $M(\tilde{d}_i \pm \epsilon \times k)$, where ϵ is a hyperparameter for depth error, and k ranges from 1 to s . We denote this set of 3D locations as $\tilde{\mathbf{M}}_i \in \mathbb{R}^{(2s+1) \times 3}$. We gather closest radar features for every sampled location, denoted as $\psi_{Ri} \in \mathbb{R}^{(2s+1) \times d}$. Likewise, we denote the camera feature for a pixel i as $\psi_{Ci} \in \mathbb{R}^{1 \times d}$. The final camera 3D point location $M_i \in \mathbb{R}^{1 \times 3}$ for pixel i can thus be obtained using a cross-attention formulation [49]:

$$M_i = \text{softmax}\left(\frac{\psi_{Ci}\psi_{Ri}^T}{\sqrt{d}}\right)\tilde{\mathbf{M}}_i. \quad (5)$$

To relate to the naming convention in attention [49], we use the camera feature ψ_{Ci} as a query, a set of radar features ψ_{Ri} as keys, and the sampled 3D locations $\tilde{\mathbf{M}}_i$ as values. Therefore, the final location is calculated by matching the query with the most active keys, associated with the respective values.

Notably, this design is computationally efficient. The time complexity is asymptotically proportional to $(N \times d \times s)$, where N is the number of (foreground) pixels. Since s is a small constant, this operation is as cheap as a conv layer.

3.3 Sensor Dropout

One appealing property of this architecture is the independence of each sensor. We can perform camera-only or radar-only 3D object detection with the same architecture when one modality is unavailable. This is desired in practice as one never knows when a sensor might be unavailable due to various situations, e.g., occlusions, weather, or sensor failure.

To enhance the model ability to handle sensor failures, we incorporate a sensor dropout mechanism [7,52] during training. With a probability P_{drop} , we randomly drop out the entire set of point features of camera ψ_C or radar ψ_R , or

$$\begin{aligned} \mathbf{X}_C &= \mathbb{1}(r_1 \geq P_{\text{drop}}) \mathbf{X}_C + \mathbb{1}(r_1 < P_{\text{drop}} \wedge r_2 \geq 0.5) \mathbf{0} \\ \mathbf{X}_R &= \mathbb{1}(r_1 \geq P_{\text{drop}}) \mathbf{X}_R + \mathbb{1}(r_1 < P_{\text{drop}} \wedge r_2 < 0.5) \mathbf{0} \end{aligned} \quad (6)$$

where $\mathbf{0}$ is a zero matrix and r_1 and r_2 are uniform random numbers in $[0, 1]$. Note that camera and radar features won't be dropped out at the same time. We use $p_{\text{drop}} = 0.2$ in our experiments.

The reason why we choose to mask out 3D point features instead of input data directly is that we can still train the cross-attention with proper 2D features normally. If radar sensor is corrupted during inference and produces noisy 2D features, it results in a uniform attention map inside cross-attention and little effect on 3D camera point locations.

4 Experiments

We present experiments on the RADIATE [40] and Waymo Open [47] datasets to verify the efficacy of our proposed CramNet model. We introduce the settings in Section 4.1 and 4.2. We include the main results, ablation studies, robustness tests, and visualization on the RADIATE dataset in Section 4.3, 4.4, 4.5, and 4.7, respectively. We also present our camera-only results on the Waymo Open dataset in Section 4.6. More ablation studies can be found in the supplementary.

4.1 Dataset and Evaluation

RADIATE dataset. We evaluate our method on the challenging RADIATE dataset [40]. This dataset features radar sensor data collected for scene understanding for safe autonomous driving in various weather conditions, including sunny, night, rainy, foggy, and snowy. The dataset includes 3 hours of annotated radar imagery with more than 200k labeled objects for 8 categories. These properties make the RADIATE dataset one of the few public datasets that contain high-resolution radar data along with a large number of ground truth labels for road actors. While the dataset provides high-quality radar data, the quality of its camera and LiDAR data is not comparable to that of other autonomous driving datasets, such as the Waymo Open Dataset [47]. This shortcoming, however, makes the evaluation of the robustness of our proposed sensor fusion algorithm even more compelling. In all of our experiments, we train the models on the

Method	Overall	Sunny (Parked)	Overcast (Motorway)	Sun/OC (Urban)	Night (Motorway)	Rain (Suburban)	Fog (Suburban)	Snow (Suburban)
Baseline [40]	46.55	79.72	44.23	35.45	64.29	31.96	51.22	8.14
CramNet-C*	23.66	67.98	6.50	23.43	2.24	17.69	9.50	0.12
CramNet-R	56.19	83.58	37.65	48.33	60.38	42.86	71.11	15.84
CramNet	62.07	96.68	50.49	52.25	79.56	57.90	85.26	8.89

Table 2: Main results evaluated in BEV AP (%) on the RADIATE dataset [40]. CramNet-C (*notes evaluation on camera/lidar-specific labels), CramNet-R, and CramNet denote our camera-only, radar-only, and fusion models, respectively. Our final model outperforms the baseline Faster R-CNN [40] by 16 percentage points, the camera-only variant by 38 points, and the radar-only variant by 6 points. These large gains validate the efficacy of our proposed sensor fusion model.

training set that contains both good and bad weather conditions, and we evaluate the resulting models on the standard validation set.

Evaluation. The (pseudo) 3D labels in the RADIATE dataset are 2D BEV labels with assumed heights for each category. We therefore report our 3D detection results in terms of BEV AP to align with the baselines, unless otherwise noted. We follow the proposed evaluation in the dataset and define the category “vehicle” to encompass the six categories “car”, “van”, “truck”, “bus”, “motorbike”, and “bicycle”. The final BEV/3D AP numbers are therefore weighted sums of the objects from these categories. For all radar and fusion experiments, we evaluate the performance on the region that is captured by both the cameras and the radar sensors, up to the radar range of 100 meters. For all camera-only experiments, we exclude labels that do not contain any LiDAR points. The motivation for this is that camera depth estimates beyond the LiDAR supervision (up to 70 meters) tend to be inaccurate.

4.2 Implementation Details

Hyperparameters. CramNet follows the implementation of RSN [48]. The sparse convolution implementation is also similar to [56]. The input camera and radar RF images are both normalized to be in $[0, 1]$. The foreground score cutoff is set to 0.15, the segmentation loss weight is set to 400, and the depth loss weight is set to 20. For cross-attention, we sample 1 point closer and farther around the predicted depth location, with 10% error. The voxelization region is $[-100\text{m}, 100\text{m}] \times [-100\text{m}, 100\text{m}] \times [-5\text{m}, 5\text{m}]$ with 0.2 meter voxel sizes. In the heatmap computation, σ_h is set to 1.0, the heatmap loss weight is set to 4 and threshold ϵ_h are set to 0.2. We use 12 bins in the heading bin loss for heading regression.

Training and inference. We train CramNet from scratch end-to-end using the Adam optimizer [14] on Tesla V100 GPUs. The models are trained with 5 batches on 8 GPUs. We use a cosine learning rate decay, where we set the initial learning rate to 0.006, with 1k warm-up steps starting at 0.003 and 50k steps in total. We use layer normalization [1] instead of batch normalization [12] in the 3D network for the number of foreground points varies among different scenes. We do not perform 2D data augmentation but adopt two 3D data augmentation strategies,

namely, random flipping along the x-axis and a global rotation around the z-axis, with a random angle from $[-\pi/4, \pi/4]$ on the selected foreground points.

4.3 Performance on the RADIATE dataset

We evaluate the performance of our method on the RADIATE dataset [40] and summarize the results in Table 2. We report the BEV AP at a 0.5 IoU threshold to align with the baseline proposed in the RADIATE dataset [40]. The baseline runs a Faster R-CNN detector with a ResNet-101 backbone on radar RF images.

Our radar-only variant, CramNet-R, outperforms the baseline by a large margin, ~ 10 percentage points in AP. Our two-stage framework effectively filters out radar noise in the segmentation stage to focus inference on the remaining radar signals in subsequent stages. Our camera-only variant, CramNet-C, performs the worst. Several factors may contribute to the poor performance. First, adverse weather affects the cameras more than the radar sensors, which is exacerbated by the lack of wipers mounted on the vehicles. Second, the effective range of the LiDAR sensors, which we use for camera depth supervision at training time, tends to drop from 70 meters in clear weather to about 40 meters in adverse weather, whereas we have labeled ground truth boxes within a range of 100 meters. Overall, we observe that the short sensing range and the sparsity of the points prevent the model from learning accurate camera depth estimation, resulting in poor camera-only 3D detection performance.

Our proposed fusion model, CramNet, equipped with ray-constrained cross attention and sensor dropout, outperforms the baseline BEV AP by 16 percentage points, the camera-only variant by 38 points, and the radar-only variant by 6 points. These large gains validate the efficacy of our proposed sensor fusion model. In the next sections, we study the performance of our method in more detail.

4.4 Ablation Study

Attention	Dropout	BEV AP	Radar Intensity Threshold	# of Points	BEV AP	Degradation
		56.19				
✓		60.20	None	-	60.20	-
	✓	61.23	0.25	70K	50.90	-15.45%
✓	✓	62.07	0.5	2K	17.81	-70.42%

Table 3: Ablation study on CramNet on the RADIATE dataset [40]. Left: The cross-attention and sensor dropout both improve over the vanilla fusion model by 4 to 5 points in AP. Putting them together yields the final fusion model with the best performance. Right: We simulate the radar sparse signals by setting the intensity thresholds to 0.25 or 0.5, resulting in $\sim 70\text{K}$ or 2K points, respectively. As a result, our model performance is degraded relatively by 15% to 70%. This confirms radar RF imagery contains critical information for 3D detection.

Dropout Location	BEV AP
Normal	57.00
Input	55.78
Point Cloud	58.64
Point Feature	61.23

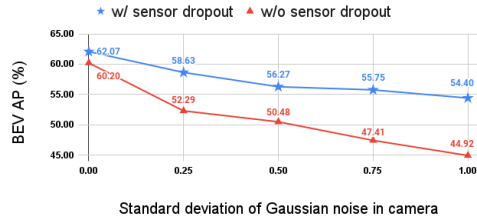


Fig. 4: Left: Analysis on different sensor dropout strategies. Masking out point features yields the best performance. Two possible benefits for this dropout location: 1) Reduce the 3D network reliance on features, which are disrupted the most given sensor noise. 2) Remain smooth training of 2D feature extractors and cross-attention. **Right:** Analysis on model performance degradation on corrupted data. We add varying degrees of Gaussian white noises to corrupt camera images and evaluate the performance. Our fusion model trained with sensor dropout greatly outperforms the one without by 2 to 10 percentage points in BEV AP. This demonstrates that sensor dropout can drastically enhance sensor robustness.

Effects of ray-constrained cross-attention and dropout. We experiment the fusion model with different settings to enable/disable ray-constrained cross-attention and sensor dropout mechanisms. The experimental results are summarized in Table 3 (left). The cross-attention and sensor dropout both improve over the vanilla fusion model by 4 to 5 points in BEV AP. Putting them together yields our final fusion model, achieving the performance of 62.07% AP.

Effects of sampling radar points. Most of the camera-radar fusion methods, such as GRIF Net [13] and CenterFusion [28], perform experiments on the nuScenes dataset [5] that contains only sparse radar points, at the scale of hundreds of points in a scene. The resulted radar-only model usually performs poorly, such as 25.5% AP reported in [13]. On the other hand, our model is specifically designed to perform either single modality effectively by taking as input the RF images instead of sparse points.

To quantitatively study how the sparsity of radar signals affects the performance, we filter RF images with varying intensity thresholds, as summarized in Table 3 (right). We set the intensity thresholds to 0.25 or 0.5, resulting in $\sim 70\text{K}$ or 2K points, respectively, which are already denser than sparse radar points available on nuScenes [5] or SeeingThroughFog [2] datasets. As a result, our model performance is degraded relatively by 15% to 70%. We conclude that radar RF imagery contains critical information for effective 3D object detection.

4.5 Detection Robustness

Detection robustness against sensor deterioration is critical for safe autonomous driving. In this section, we study the effects of our proposed sensor dropout with ablation study and corrupted sensor data.

Where to drop out sensor data? Dropout is a popular technique for training neural network models [45]. It is usually applied on a layer to randomly mask out neuron activations. We experiment on various places to drop out sensor data and summarize them in Table 4 (left). The ‘normal’ dropout applies the conventional dropout on point cloud features, regardless of which sensor the points are from. This conventional dropout does not provide benefits in either overall performance or in bad weather conditions. The ‘input’ dropout randomly masks out a sensor (radar or camera) entirely. The ‘point cloud’ dropout randomly masks out the 3D points from one sensor entirely. The ‘point feature’ dropout randomly masks out the initial point cloud features from one sensor entirely but leaves the point cloud positions intact. As the numbers dictate, masking out point features yields the best performance. Two possible benefits for this dropout location: (1) Reduce the 3D network reliance on features, which are disrupted the most due to sensor noise. (2) Remain training of 2D feature extractors and cross-attention. As such, we conclude dropping out sensor point features randomly is the most effective.

Sensor dropout improves robustness against input corruption. We study how the corruption of sensors will affect the performance with and without sensor dropout and summarize the experiments in Table 4 (right). For this purpose, we add random Gaussian white noise with varying standard deviation to corrupt the camera images to different degrees. We evaluate the fusion model on the corrupted data, with or without sensor dropout during training. The experimental results show that our fusion model trained with sensor dropout greatly outperforms the one without by 2 to 10 percentage points in BEV AP. This study demonstrates that sensor dropout can drastically enhance sensor robustness.

4.6 Camera-only CramNet on Waymo Open Dataset

Our radar-only and camera-radar fusion models perform strongly on the RADAR-ATE dataset [40]. However, the camera-only model suffers from the poor image quality and adverse weather conditions in the dataset. Since we do not have access to another public dataset that contains radar RF imagery, we evaluate our camera-only model performance on the Waymo Open Dataset [47], as summarized in Table 4. We report 3D AP/APH with 0.7 IoU threshold on the LEVEL_1 difficulty in Table 4. Our camera-only model, CramNet-C, achieves competitive performance. More details can be found in the supplementary.

4.7 Visualization

We present the visual comparisons between our camera-only, radar-only, and fusion models in Figure 5. Since the camera visibility is severely reduced due to either underexposure or adverse weather, the camera-only model tends to miss detection and the predicted localization tends to be inaccurate. In contrast, the radar-only model suffers from false positives due to lack of appearance features from RF images. Overall, our camera-radar fusion model combines the advantages from the two and produces the most accurate predictions.

Method	3D AP	0 - 30m	30 - 50m	50m - ∞	3D APH	0 - 30m	30 - 50m	50m - ∞
M3D-RPN [4]	0.35	1.12	0.18	0.02	0.34	1.10	0.18	0.02
CaDDN [34]	5.03	14.54	1.47	0.10	4.99	14.43	1.45	0.10
CramNet-C	4.14	15.46	1.20	0.15	4.10	15.31	1.19	0.13

Table 4: Camera-only 3D detection results on the Waymo Open Dataset [47] validation set on the vehicle class, evaluated in terms of 3D AP/APH at 0.7 IoU on the LEVEL_1 difficulty. Baseline numbers are from [34]. Our camera-only model, CramNet-C, achieves competitive performance among state-of-the-art.

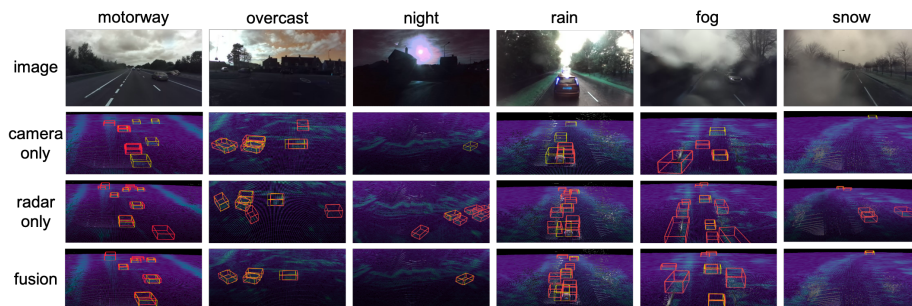


Fig. 5: Visual comparison between CramNet-C, CramNet-R, and CramNet from 6 scenarios. We visualize the predicted boxes in red and the ground truth boxes in yellow with projected radar and camera pixels. Whereas the camera-only model tends to miss detections and predict inaccurate localization, the radar-only model suffers from false positives. Our camera-radar fusion model combines the advantages of the two and produces the most accurate predictions.

5 Conclusion

We introduced a camera-radar sensor fusion approach for robust 3D object detection for autonomous driving. The method relies on a ray-constrained cross-attention mechanism to leverage the range measurements from radar to improve camera depth estimates. Training with sensor dropout allows the method to fall back to a single modality when one of the sensors malfunctions. We present experiments on the RADIATE dataset and the Waymo Open Dataset.

Limitations. Whereas a camera pixel corresponds to a ray, a (range, azimuth) radar reading corresponds to an arc in 3D space. Intersecting a camera ray and a radar arc yields their correspondence. We approximate the radar arc as a pillar, that is, we assume that the radar points are at the same elevation as the sensor. This assumption works well in practice when most objects are at a similar elevation as the sensor. We currently use the RF images in Cartesian coordinates, which may be suboptimal as the radar natively operates in polar coordinates. We will explore a polar convolutional network design and radar-specific spherical voxelization in future work.

References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
2. 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: CVPR (2020)
3. Bijelic, M., Gruber, T., Ritter, W.: A benchmark for lidar sensors in fog: Is detection breaking down? In: 2018 IEEE Intelligent Vehicles Symposium (IV) (2018)
4. Brazil, G., Liu, X.: M3d-rpn: Monocular 3d region proposal network for object detection. In: ICCV (2019)
5. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR (2020)
6. Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., Urtasun, R.: Monocular 3d object detection for autonomous driving. In: CVPR (2016)
7. Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-view 3d object detection network for autonomous driving. In: CVPR (2017)
8. Chen, Y., Tai, L., Sun, K., Li, M.: Monopair: Monocular 3d object detection using pairwise spatial relationships. In: CVPR (2020)
9. Ding, M., Huo, Y., Yi, H., Wang, Z., Shi, J., Lu, Z., Luo, P.: Learning depth-guided convolutions for monocular 3d object detection. In: CVPR workshops (2020)
10. Graham, B., van der Maaten, L.: Submanifold sparse convolutional networks. arXiv preprint arXiv:1706.01307 (2017)
11. Huang, T., Liu, Z., Chen, X., Bai, X.: Epnet: Enhancing point features with image semantics for 3d object detection. In: ECCV (2020)
12. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: ICML (2015)
13. Kim, Y., Choi, J.W., Kum, D.: Grif net: Gated region of interest fusion network for robust 3d object detection from radar point cloud and monocular image. In: IROS (2020)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
15. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: CVPR (2019)
16. Li, P., Zhao, H., Liu, P., Cao, F.: Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. In: ECCV (2020)
17. Li, Y., Yu, A.W., Meng, T., Caine, B., Ngiam, J., Peng, D., Shen, J., Lu, Y., Zhou, D., Le, Q.V., et al.: Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17182–17191 (2022)
18. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Yu, Q., Dai, J.: Bevformer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers. In: ECCV (2022)
19. Liang, M., Yang, B., Wang, S., Urtasun, R.: Deep continuous fusion for multi-sensor 3d object detection. In: ECCV (2018)
20. Lim, T.Y., Ansari, A., Major, B., Fontijne, D., Hamilton, M., Gowaikar, R., Subramanian, S.: Radar and camera early fusion for vehicle detection in advanced driver assistance systems. In: Machine Learning for Autonomous Driving Workshop at the 33rd Conference on Neural Information Processing Systems (2019)

21. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017)
22. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV (2017)
23. Liu, L., Wu, C., Lu, J., Xie, L., Zhou, J., Tian, Q.: Reinforced axial refinement network for monocular 3d object detection. In: ECCV (2020)
24. Ma, X., Liu, S., Xia, Z., Zhang, H., Zeng, X., Ouyang, W.: Rethinking pseudo-lidar representation. In: ECCV (2020)
25. Major, B., Fontijne, D., Ansari, A., Teja Sukhavasi, R., Gowaikar, R., Hamilton, M., Lee, S., Grzechnik, S., Subramanian, S.: Vehicle detection with automotive radar using deep learning on range-azimuth-doppler tensors. In: ICCV Workshops (2019)
26. Manhardt, F., Kehl, W., Gaidon, A.: Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In: CVPR (2019)
27. Mousavian, A., Anguelov, D., Flynn, J., Kosecka, J.: 3d bounding box estimation using deep learning and geometry. In: CVPR (2017)
28. Nabati, R., Qi, H.: 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)
29. Nobis, F., Shafiei, E., Karle, P., Betz, J., Lienkamp, M.: Radar voxel fusion for 3d object detection. Applied Sciences (2021)
30. Park, D., Ambrus, R., Guizilini, V., Li, J., Gaidon, A.: Is pseudo-lidar needed for monocular 3d object detection? In: ICCV (2021)
31. Piergiovanni, A., Casser, V., Ryoo, M.S., Angelova, A.: 4d-net for learned multi-modal alignment. In: ICCV (2021)
32. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3d object detection from rgb-d data. In: CVPR (2018)
33. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. NeurIPS (2017)
34. Reading, C., Harakeh, A., Chae, J., Waslander, S.L.: Categorical depth distribution network for monocular 3d object detection. In: CVPR (2021)
35. Richards, M.A., Scheer, J., Holm, W.A., Melvin, W.L.: Principles of modern radar (2010)
36. Roddick, T., Kendall, A., Cipolla, R.: Orthographic feature transform for monocular 3d object detection. arXiv preprint arXiv:1811.08188 (2018)
37. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015)
38. Schumann, O., Hahn, M., Dickmann, J., Wöhler, C.: Semantic segmentation on radar point clouds. In: 2018 21st International Conference on Information Fusion (FUSION) (2018)
39. Shah, M., Huang, Z., Laddha, A., Langford, M., Barber, B., Zhang, S., Vallespi-Gonzalez, C., Urtasun, R.: Liranet: End-to-end trajectory prediction using spatio-temporal radar fusion. In: CoRL (2020)
40. Sheeny, M., De Pellegrin, E., Mukherjee, S., Ahrabian, A., Wang, S., Wallace, A.: Radiate: A radar dataset for automotive perception in bad weather. In: ICRA (2021)
41. Shi, S., Wang, X., Li, H.: Pointcnn: 3d object proposal generation and detection from point cloud. In: CVPR (2019)
42. Shi, X., Chen, Z., Kim, T.K.: Distance-normalized unified representation for monocular 3d object detection. In: ECCV (2020)

43. Simonelli, A., Bulo, S.R., Porzi, L., López-Antequera, M., Kotschieder, P.: Disentangling monocular 3d object detection. In: ICCV (2019)
44. Simonelli, A., Bulo, S.R., Porzi, L., Ricci, E., Kotschieder, P.: Towards generalization across depth for monocular 3d object detection. In: ECCV (2020)
45. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* (2014)
46. Srivastava, S., Jurie, F., Sharma, G.: Learning 2d to 3d lifting for object detection in 3d for autonomous vehicles. In: IROS (2019)
47. Sun, P., Kretschmar, 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: CVPR (2020)
48. Sun, P., Wang, W., Chai, Y., Elsayed, G., Bewley, A., Zhang, X., Sminchisescu, C., Anguelov, D.: Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In: CVPR (2021)
49. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. *NeurIPS* (2017)
50. Vora, S., Lang, A.H., Helou, B., Beijbom, O.: Pointpainting: Sequential fusion for 3d object detection. In: CVPR (2020)
51. Wang, C., Ma, C., Zhu, M., Yang, X.: Pointaugmenting: Cross-modal augmentation for 3d object detection. In: CVPR (2021)
52. Wang, W., Tran, D., Feiszli, M.: What makes training multi-modal classification networks hard? In: CVPR (2020)
53. Wang, Y., Chao, W.L., Garg, D., Hariharan, B., Campbell, M., Weinberger, K.Q.: Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In: CVPR (2019)
54. Wang, Z., Jia, K.: Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. In: IROS (2019)
55. Weng, X., Kitani, K.: Monocular 3d object detection with pseudo-lidar point cloud. In: ICCV Workshops (2019)
56. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. *Sensors* (2018)
57. Yang, B., Guo, R., Liang, M., Casas, S., Urtasun, R.: Radarnet: Exploiting radar for robust perception of dynamic objects. In: ECCV (2020)
58. You, Y., Wang, Y., Chao, W.L., Garg, D., Pleiss, G., Hariharan, B., Campbell, M., Weinberger, K.Q.: Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. *arXiv preprint arXiv:1906.06310* (2019)
59. Zhou, D., Fang, J., Song, X., Guan, C., Yin, J., Dai, Y., Yang, R.: Iou loss for 2d/3d object detection. In: 2019 International Conference on 3D Vision (3DV) (2019)
60. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. *arXiv preprint arXiv:1904.07850* (2019)
61. Zhou, Y., Sun, P., Zhang, Y., Anguelov, D., Gao, J., Ouyang, T., Guo, J., Ngiam, J., Vasudevan, V.: End-to-end multi-view fusion for 3d object detection in lidar point clouds. In: CoRL (2020)

A Appendix

We propose an efficient camera-radar sensor fusion approach for robust 3D object detection for autonomous driving. The method uses a ray-constrained cross-attention mechanism to leverage the range measurements from radar to improve camera depth estimates, leading to improved detection performance. More importantly, the architecture is designed in a way that training with dropout allows the method to fall back to a single modality when one of the sensors malfunctions.

Here, we include more details on the following aspects:

1. We describe the experimental details and present more complete results on the Waymo Open Dataset in A.1.
2. We study the trade-off between latency and foreground thresholds in A.2.
3. We present ablation study on hyperparameters related to the fusion design, e.g., ray-constrained cross-attention and sensor dropout in A.3.
4. We document the detailed architecture of CramNet in A.4.

A.1 Experiment on Waymo Open Dataset

We use the same hyperparameters as on the RADIATE dataset [40] for training a camera-only CramNet on the Waymo Open Dataset [47]. We adopt a longer training procedure, i.e., 60k warm-up steps and 120k total steps, due to the larger size of the dataset. We align our setting with CaDDN [34] to train and evaluate our performance using the front camera. However, we train our model on a lower resolution (640, 960), than in CaDDN [34], (832, 1248).

We report the 3D AP/APH with 0.5 and 0.7 IoU threshold on the LEVEL_1 and LEVEL_2 difficulties in Table 5. We conclude that our camera-only model, CramNet-C, achieves competitive performance among the state-of-the-art models.

We notice that our model performs significantly better (+50% ~ 300%) in the long range region (50 m - ∞). This suggests the sparse operation in 3D after the 2D segmentation filtering can better handle the long range objects, even without implicitly or explicitly modeling depth uncertainty.

A.2 Ablation Study on Latency and Foreground Threshold

One of the important trade-off in our model hyperparameters is the foreground segmentation threshold. This threshold controls the density of foreground points passed from the 2D to 3D stage. Therefore, we expect the model to perform better with a lower threshold, with the trade-off of a higher latency.

We summarize this ablation study in Figure 6. Our reported performance in the main paper is at the 0.15 threshold with a latency of 46.4 ms. We observe a general trend of lower accuracy and lower latency when setting a lower threshold.

Difficulty	Method	3D AP	0 - 30m	30 - 50m	50m - ∞	3D APH	0 - 30m	30 - 50m	50m - ∞
Level 1 (IoU = 0.5)	M3D-RPN [4]	3.79	11.14	2.16	0.26	3.63	10.70	2.09	0.21
	CaDDN [34]	17.54	45.00	9.24	0.64	17.31	44.46	9.11	0.62
	CramNet-C	11.81	32.20	7.24	2.00	11.59	31.75	7.08	1.93
Level 2 (IoU = 0.5)	M3D-RPN [4]	3.61	11.12	2.12	0.24	3.46	10.67	2.04	0.20
	CaDDN [34]	16.51	44.87	8.99	0.58	16.28	44.33	8.86	0.55
	CramNet-C	10.64	30.29	6.56	1.76	10.44	29.86	6.42	1.69
Level 1 (IoU = 0.7)	M3D-RPN [4]	0.35	1.12	0.18	0.02	0.34	1.10	0.18	0.02
	CaDDN [34]	5.03	14.54	1.47	0.10	4.99	14.43	1.45	0.10
	CramNet-C	4.14	15.46	1.20	0.15	4.10	15.31	1.19	0.13
Level 2 (IoU = 0.7)	M3D-RPN [4]	0.33	1.12	0.18	0.02	0.33	1.10	0.17	0.02
	CaDDN [34]	4.49	14.50	1.42	0.09	4.45	14.38	1.41	0.09
	CramNet-C	3.72	14.53	1.09	0.13	3.68	14.38	1.07	0.13

Table 5: Camera-only 3D detection results on the Waymo Open Dataset [47] validation set on the vehicle class, evaluated in terms of 3D AP/APH at 0.5 or 0.7 IoU on the LEVEL.1 or LEVEL.2 difficulties. Baseline numbers are from [34]. Our camera-only model, CramNet-C, achieves competitive performance among state-of-the-art with the best long range detection.

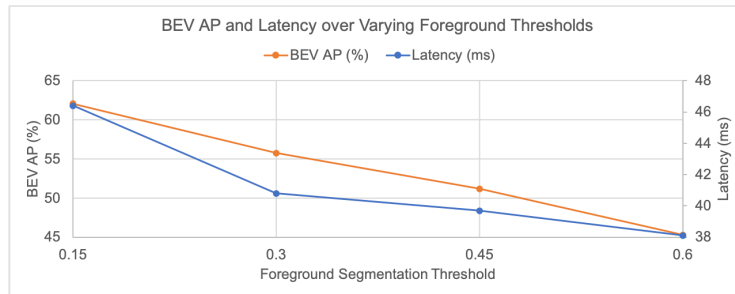


Fig. 6: Ablation study on foreground segmentation thresholds. The model accuracy in BEV AP and latency both decreases as the foreground segmentation threshold decreases.

A.3 Ablation Studies on Fusion Hyperparameters

We conduct ablation studies on the effect of hyperparameters w.r.t. the 3D detection performance in BEV AP, summarized in Figure 7. All in all, the ablation studies suggest the model is not too sensitive to hyperparameters.

For the ray constrained cross-attention, we notice the best error rate ($\epsilon = 0.1$) corresponds to the general depth errors. Also, we do not need many samples along the camera ray (from 3 to 5 sampled points) as each sample already covers a large region through feature extraction.

For the sensor dropout, the performance peaks at 0.2 dropout probability and decreases as the probability increases, indicating that too frequent dropout actually hurts the model.

For modality encoding, when removing the modality code, the BEV AP of CramNet degrades by 8.7 percentage points from 62.1% to 53.4%. This indicates that the modality encoding is critical for the model to distinguish and utilize features from different sensors.

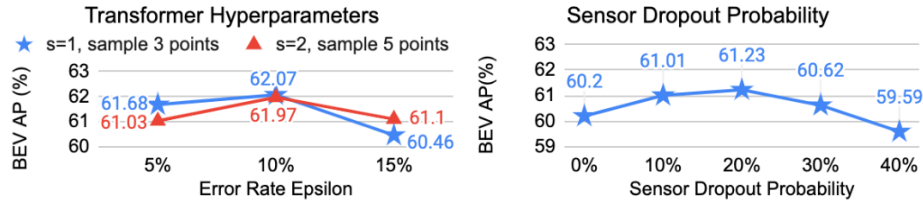


Fig. 7: **Left:** Ablation study on hyperparameters in ray-constrained cross-attention. **Right:** Ablation study on hyperparameters in sensor dropout. All in all, the ablation studies suggest the model is not too sensitive to hyperparameters.

A.4 Architecture Details

2D U-Net. A downsampling block $D(B_i, C_i)$ at level i contains B resnet blocks with C -dimensional outputs, with stride 2 in the first convolutional layer. Each upsampling block $U(B_i, C_i)$ at level i also contains B resnet blocks with C -dimensional outputs. The upsampling is performed by a 1×1 convolution layer followed by a bilinear interpolation layer. We connect the same level of the corresponding downsampling block and upsampling block to construct the U-Net.

After applying an initial 1×1 convolution layer on the input with 16-dimensional outputs, we construct a 2-D U-Net with hyperparameters specified in Table 6. We use the exact same 2D U-Net for both camera and radar inputs for simplicity. One can replace them with stronger feature extractor backbones.

$D(B_1, C_1)$	$D(B_2, C_2)$	$D(B_3, C_3)$	$D(B_4, C_4)$	$U(B_1, C_1)$	$U(B_2, C_2)$	$U(B_3, C_3)$	$U(B_4, C_4)$
(3, 16)	(3, 16)	(1, 64)	(0, 128)	(1, 16)	(1, 16)	(1, 64)	(1, 128)

Table 6: Detailed hyperparameters to construct a 2D U-Net.

3D Sparse U-Net. We reuse the notations as 2D U-Net for 3D U-Net. A residual block in 3D U-Net is replaced by a $3 \times 3(\times 3)$ sparse convolution layer before the residual connection and by two $3 \times 3(\times 3)$ submanifold sparse convolution layers within the residual block. Unlike the symmetric downsampling and upsampling blocks in the 2D U-Net, we employ 2 more downsampling blocks to output a lower resolution objectness heatmap. We summarize the hyperparameters used to construct a 3D sparse U-Net in Table 7.

$D(B_1, C_1)$	$D(B_2, C_2)$	$D(B_3, C_3)$	$D(B_4, C_4)$	$D(B_5, C_5)$	$U(B_1, C_1)$	$U(B_2, C_2)$	$U(B_3, C_3)$
(1, 96)	(2, 96)	(2, 96)	(1, 96)	(1, 96)	(0, 96)	(2, 96)	(2, 96)

Table 7: Detailed hyperparameters to construct a 3D Sparse U-Net.