BEVFusion-R: Efficient and Deployment-Ready Camera-Radar Fusion

Kevin Shao
MIT
kshao23@mit.edu

Zhijian Liu
MIT
zhijian@mit.edu

Haotian Tang
MIT
kentang@mit.edu

Xuanyao Chen
Fudan University
xuanyaochen19@fudan.edu.cn

Song Han
MIT
songhan@mit.edu

Abstract

As a critical component to realizing widespread autonomous driving, 3D perception systems have come to be heavily studied in the community. However, many solutions overlook practical considerations – cost, speed, and real-world deployment. In this paper, we propose ***BEVFusion-R****, a camera-radar fusion framework that is a strong candidate for practical real-time 3D object detection. By fusing features from each input modality in the shared bird's eye view space, we capture both semantic and geometric information from each. Moreover, by carefully designing each module with both performance and acceleration in mind, BEVFusion-R achieves a 2.1% NDS improvement on nuScenes over the previous state-of-the-art with a 4.5× measured speedup. It runs on an edge GPU at real-time speeds.*

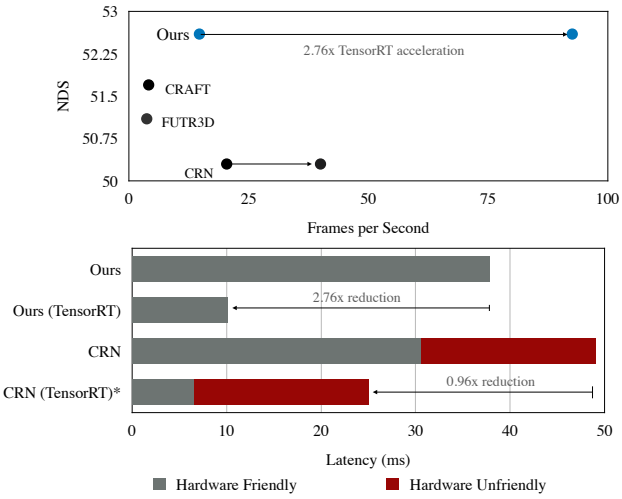

Figure 1. While outperforming competing works, our BEVFusion-R is optimization-friendly, allowing it to enjoy significantly more hardware acceleration. See Section 4 for more details.

1. Introduction

The task of 3D object detection has become increasingly important over the past few years, particularly in the context of autonomous driving applications. Large multi-modal datasets [1, 3, 28] have been instrumental in driving research and advancing accuracy. However, there has been relatively less emphasis on the practical deployment of these models, which requires accounting for multiple factors simultaneously. A practical detector should be both *accurate* and *fast*, while also being *cost-effective* and *deployable*.

Although LiDAR-based 3D perception [9, 22, 37] has consistently outperformed other modalities, camera-based approaches have also been extensively studied due to their cost-effectiveness [13, 15, 16, 21, 24, 32, 35]. However, camera-based detection suffers from poor distant object localization due to the ill-posed depth estimation problem. Moreover, cameras are sensitive to lighting conditions, and their performance degrades severely in nighttime conditions, which limits their usefulness in real-world scenarios [8, 30]. Conversely, LiDAR points provide very accurate depth, but the sparsity of points at large distances can make object recognition very challenging. Therefore, research in sensor fusion techniques [11, 31, 33, 36, 39] has been motivated by the complementary features of these modalities.

Recently, radar has garnered notable interest as a cost-efficient alternative to LiDAR. Like LiDAR, radar data has characteristics that are largely complementary to cameras. However, radar sensors also have advantageous characteristics as well. First, radar sensors are much cheaper than LiDAR. Despite the cost of LiDAR's decreasing nearly ten-fold in recent years, a sensor still costs thousands of dollars, whereas a radar sensor costs a few hundred. Second, radar uses a lower frequency than LiDAR, allowing it to be more robust in rainy or foggy conditions. Finally, radar data offers accurate velocity estimation, whereas LiDAR data depends on temporal fusion to infer it.

In this paper, we propose to use camera-radar fusion as a prime candidate to satisfy all these desiderata. By performing the fusion in the shared bird's-eye view (BEV) space, we retain both semantic and geometric information from the input modalities, which allows the model to exploit the complementary information from each. Furthermore, we design the radar encoder and radar-guided view transformation in an optimization-friendly manner. In this way, our model,

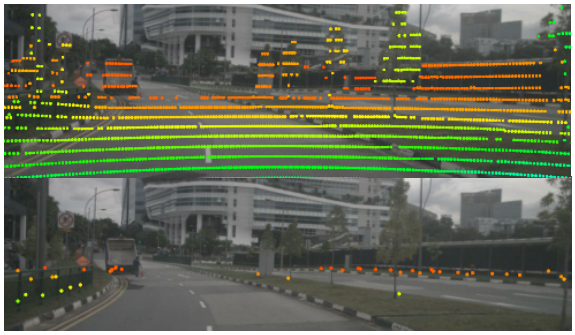

Figure 2. Radar returns are far sparser than LiDAR point clouds.

with TensorRT acceleration, runs 4.5× faster than competing methods, while achieving 2.1% better NDS. Together, these techniques allow us to deploy the model on an edge device and realize real-time latency.

2. Radar

Similar to LiDAR, radar captures data in the form of a 3D point cloud. As depicted in Figure 2, radar data is usually sparser and noisier compared to LiDAR data. In fact, radar data can be over 100× sparser than 32-beam LiDAR data, and more than 5× sparser than 1-beam LiDAR data. Nonetheless, a surface-level comparison of the two modalities overlooks their inherent differences and the unique advantages of radar. For instance, radar has superior coverage of distant objects and enables accurate velocity estimation, which are both very crucial for 3D perception.

Object Coverage. As illustrated in Table 1, while radar returns are sparser than LiDAR, they offer superior object coverage than sparse LiDAR and even comparable coverage to dense LiDAR at distances greater than 50 meters. Here, an object is considered covered if at least one radar or LiDAR point falls within its corresponding 3D cuboid. Maintaining good object coverage at long ranges is crucial, particularly since camera-based localization struggles disproportionately at these distances. Thus, having excellent object coverage at long range is a desirable quality in camera-radar fusion.

Velocity Estimation. Accurate velocity estimation is crucial for 3D perception since it offers important information for motion prediction. Almost all LiDAR perception models employ multi-frame stacking to enable temporal reasoning. In contrast, radar data can directly measure the tangential velocity component of a moving target. In Figure 3, we analyze the correlation between the velocity obtained from radar returns and the ground truth velocity of each covered object. When considering only moving objects, we find a strong linear correlation of $r^2 = 0.77$, which provides a robust basis for accurate velocity estimation using camera-radar fusion.

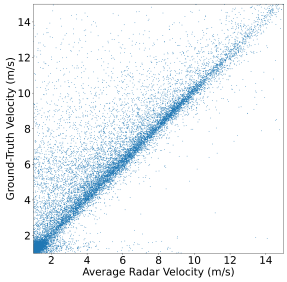

Figure 3. The velocity from radar exhibits a linear correlation with the actual velocity.

	<10m	10-25m	25-50m	>50m	All
Objects	2.8	9.3	9.9	3.5	25.5
Radar	51%	60%	65%	28%	57%
LiDAR (1b)	46%	56%	44%	9%	44%
LiDAR (32b)	100%	100%	100%	32%	90%

Table 1. Despite its sparsity, radar data provides superior object coverage compared to sparse LiDAR data and is even comparable to dense LiDAR data at long range.

3. Method

Following many recent works [5, 8, 18], we perform the camera-radar fusion in the bird's-eye view (BEV) space. We first extract image and radar features with encoders and then project the image features onto BEV using a radar-guided view transformer. Finally, we fuse the multi-sensory features using a BEV encoder, and generate the final prediction using a detection head. Figure 4 illustrates our BEVFusion-R.

Image Encoder. We encode multi-view RGB images with ResNet [4] and fuse multi-scale feature maps with FPN [14]. Despite achieving higher accuracy, more advanced image encoders, like Swin Transformer [17], tend to be less efficient.

Radar Encoder. Most existing camera-radar fusion approaches [8] use variants of PointNet++ [27] as their radar encoders. However, they are not very efficient or hardware-friendly [19] due to their costly neighbor query and gathering operations. In this paper, we adopt PointPillars [10] as our radar encoder, which uses PointNet [26] within each pillar to extract features. We increase the size of the pillar to accommodate the additional noise of the radar returns. Despite its simplicity, it achieves comparable performance to strong sparse convolution-based encoders [29, 34, 38] while being 7× faster (see Table 3 for detailed results).

Radar-Guided View Transformer. Since radar features are in BEV and image features are in the perspective view, we follow LSS/BEVDet [6, 25] to transform image features into BEV. We first predict the depth distribution of each pixel and then "splat" each pixel into 3D space based on the depth prediction. Some recent papers [12] have identified that the depth quality is a significant performance bottleneck since estimating depth from a single image is an inherently ill-posed problem. Given that radar returns provide sparse, noisy but relatively accurate spatial information, we have re-designed the view transformer to be *radar-guided*. Specifically, we first height-expand the radar point cloud [20] and project radar points onto the image plane. Next, we feed the

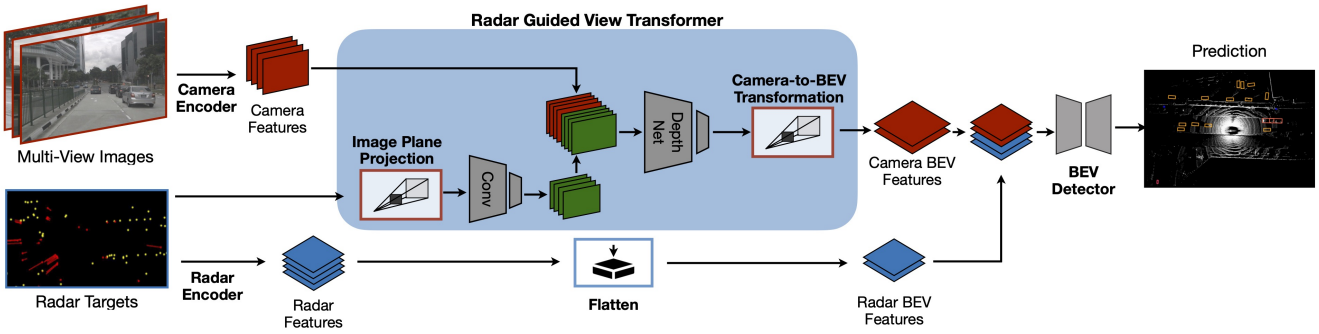

Figure 4. **BEVFusion-R** is a highly efficient multi-modal 3D perception model that leverages information from radar and camera sensors within the shared bird's-eye-view (BEV) space. In contrast to BEVFusion [18], our approach incorporates a radar-guided view transformer, which utilizes precise depth information from 3D radar to explicitly enhance camera-to-BEV projection.

radar point features and one-hot encoded radar depth into a lightweight CNN. Finally, we input the resulting radar features with the image features to the depth estimation module.

BEV Encoder. Once both image and radar features are in BEV, we proceed to fuse them using a BEV encoder following BEVFusion [18]. Previous works have used deformable attention or deformable convolution in the BEV encoder to address spatial misalignment between modalities. However, these operators have an irregular data access pattern, making them challenging to accelerate and deploy on hardware. Instead, we observe that standard convolutions can also increase the receptive field and improve performance without requiring these hardware-unfriendly operations.

Deployment. As our target hardware is NVIDIA GPUs, we deploy our model using `TensorRT`* for the best system optimizations. Almost all operators in our model are directly deployable using `TensorRT`, since we avoid using nearest neighbor query (in radar encoder) or deformable convolution (in BEV encoder). For unsupported operators, such as pillar scatter (in radar encoder) and BEV pooling (in view transformer), we develop custom `ONNX` and `TensorRT` plugins for each of them. As a result, our entire model is deployment-ready and can be run in an end-to-end manner.

4. Experiments

Setup. We conduct all our experiments on nuScenes [1], which is currently the largest publicly available multimodal dataset that includes radar data. We evaluate our model and baselines using two primary metrics, NDS and mAP, on the validation set. We measure the latency numbers on a single NVIDIA RTX 3090 GPU, in FP16 with a batch size of 1.

Main Results. We compare BEVFusion-R with state-of-the-art single-frame camera-radar fusion detectors, as summarized in Table 2. Our method not only achieves the highest NDS score but also significantly reduces the inference latency of CRN [8] by **4.5**×, thanks to our deployment-friendly design. Moreover, BEVFusion-R operates at a remarkable speed of 24 frames per second (FPS) on an NVIDIA Jetson AGX Orin, enabling real-time 3D perception at the edge. We anticipate that BEVFusion-R can further leverage the benefits of multi-frame camera inputs, as observed in [8].

Ablation Studies. In order to validate the design choices of our fusion detection model, we carry out several ablation studies on the Nuscenes *val* set.

To assess the impact of radar encoder architecture, we conducted an ablation study presented in Table 3. We compared VoxelNet [34] (VN) and PointPillars (PP) [10], which are commonly employed as 3D backbones in LiDAR-only and camera-LiDAR 3D object detectors. VoxelNet generally exhibits superior accuracy at the cost of increased latency. However, this accuracy advantage diminishes rapidly when dealing with sparse 3D data, such as a single beam LiDAR or radar. We therefore choose PointPillars as our radar feature encoder in BEVFusion-R, which is **7**× faster than VoxelNet and delivers almost the same accuracy.

Table 4 shows the impact that accurate depth estimation can bring to a fusion detection model. As observed by [12], depth supervision can improve the overall performance significantly. Moreover, allowing the depth estimation module to access radar data allows it to more accurately predict the depth, again leading to better results.

In addition, we emphasize the significance of aggregating multiple radar frames, as depicted in Figure 5. Motivated by Figure 3, the velocity estimation performance improves by 25.7% when we increase the radar input from one frame to ten frames. Moreover, considering that radar provides accurate 3D localization for a substantial portion of objects (as demonstrated in Table 1), incorporating more radar frames also leads to a 7.2% reduction in translation error.

Latency and Real Time Deployment. We conduct an in-depth analysis of the latency breakdown in BEVFusion-R,

*https://github.com/NVIDIA/TensorRT

	Modality	Image Size	Latency (ms)	NDS	mAP	mATE	mASE	mAOE	mAAE	mAVE
MVFusion [30]	C+R	900×1600	–	45.5	38.0	0.675	0.258	0.372	0.198	0.394
CenterFusion [23]	C+R	448×800	219	45.3	33.2	0.649	0.263	0.535	0.142	0.540
FUTR-3D [2]	C+R	900×1600	271	51.1	39.9	0.647	0.270	0.365	0.189	0.413
CRAFT [7]	C+R	448×800	244	51.7	41.1	0.494	0.276	0.454	0.176	0.486
RCBEV4D [40]	C+R	256×704	–	49.7	38.1	0.526	0.262	0.445	0.185	0.465
CRN [8]	C+R	256×704	49.0	50.3	42.9	0.519	–	0.577	–	0.520
BEVFusion-R (Ours)	C+R	256×704	**10.8**	**52.4**	42.8	0.523	0.273	0.542	0.185	0.379

Table 2. Results for single-frame 3D Object Detection on the nuScenes validation set. Our method achieves a state-of-the-art nuScenes detection score (NDS) while maintaining a 92 FPS speed, which is 4.5× faster than the best available baseline.

	NDS			Latency (ms)		
	VN	PP	Δ	VN	PP	Δ
32B LiDAR	64.8	60.2	+4.6	118.0	88.0	+30.0
1B LiDAR + C	53.2	50.5	+2.7	37.7	3.7	+34.0
Radar + C	48.7	48.6	+0.1	16.7	2.4	+14.3

Table 3. Ablation study on the choice of encoder for the point cloud. VN denotes VoxelNet. PP denotes PointPillars. Despite worsening latency in all cases, using VoxelNet, a 3D based encoder, will improve performance in the 32-beam and 1-beam LiDAR settings. However, for radar inputs, it performs no better than PointPillars.

Method	mAP	NDS
No depth supervision	38.9	51.0
With depth supervision	40.7	51.6
With depth supervision + Radar input	41.4	52.2

Table 4. Ablation study on the View Transformer. Not only does explicit depth supervision enable stronger performance, but using input from other modalities further improves the accuracy.

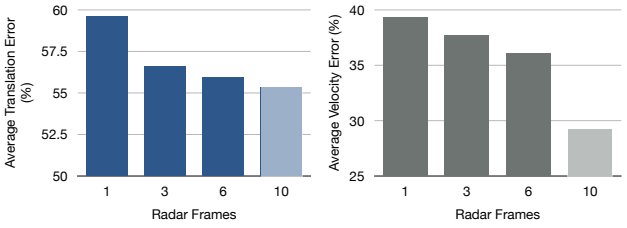

Figure 5. Ablation study on the number of radar frames included. As the number of radar points increases, both object localization (translation error) and velocity estimation (velocity error) improve. 10 frames is not a valid setting as nuScenes detection rules only allow for using up to 6 past frames.

shown in Figure 6. We compare our model with CRN [8], which is recognized as the fastest available camera-radar fusion 3D object detector. Our BEVFusion-R is executed end-to-end using TensorRT. For CRN, as the source code is unavailable, we estimate its deployment latency by running

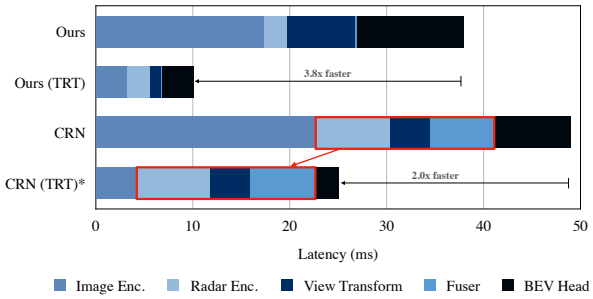

Figure 6. Acceleration of our model via TensorRT. Despite having similar PyTorch latency as CRN [8], our model is entirely hardware friendly, whereas the red sections in CRN are not. As such, we observe a much more dramatic speedup.

all TensorRT-compatible operators with the TensorRT backend while executing the remaining modules (*e.g.* PointNet++ radar encoder, multi-modal fuser based on deformable attention) in PyTorch. Consequently, our model benefits more from the faster TensorRT backend (**3.8**× *vs.* 2.0× faster) compared to CRN. In addition, BEVFusion-R achieves 24.4 FPS on Orin, whereas CRN could only run at 10.7 FPS. It is noteworthy that the nuScenes radar sensor operates at 13 FPS. Consequently, BEVFusion-R still maintains real-time performance on the edge, while CRN could not.

5. Conclusion

In this work, we developed BEVFusion-R, an efficient and performant camera-radar fusion model tailored for practical real-time autonomous driving. By performing sensor fusion in bird's-eye view space with the help of a radar-guided view transformer, we allow the model to exploit the complementary characteristics of the sensors. Moreover, by considering hardware optimization from the beginning and performing end-to-end deployment and acceleration, our model can simultaneously achieve state-of-the-art accuracy while running 4.5× faster than previous methods. Capable of real-time latency on an edge device, we hope that our method will inspire future research on lightweight camera-radar fusion for 3D perception.

References

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A Multimodal Dataset for Autonomous Driving. In *CVPR*, 2020. 1, 3

[2] Xuanyao Chen, Tianyuan Zhang, Yue Wang, Yilun Wang, and Hang Zhao. FUTR3D: A Unified Sensor Fusion Framework for 3D Detection. *arXiv*, 2022. 4

[3] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 1

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 2

[5] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022. 2

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

[7] Youngseok Kim, Sanmin Kim, Jun Won Choi, and Dongsuk Kum. Craft: Camera-radar 3d object detection with spatio-contextual fusion transformer, 2022. 4

[8] Youngseok Kim, Sanmin Kim, Juyeb Shin, Jun Won Choi, and Dongsuk Kum. Crn: Camera radar net for accurate, robust, efficient 3d perception, 2023. 1, 2, 3, 4

[9] Junho Koh, Junhyung Lee, Youngwoo Lee, Jaekyum Kim, and Jun Won Choi. Mgtanet: Encoding sequential lidar points using long short-term motion-guided temporal attention for 3d object detection. *arXiv preprint arXiv:2212.00442*, 2022. 1

[10] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, and Jiong Yang. PointPillars: Fast Encoders for Object Detection from Point Clouds. In *CVPR*, 2019. 2, 3

[11] Yanwei Li, Yilun Chen, Xiaojuan Qi, Zeming Li, Jian Sun, and Jiaya Jia. Unifying Voxel-based Representation with Transformer for 3D Object Detection. In *NeurIPS*, 2022. 1

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

[13] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. BEVFormer: Learning Bird's-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers. In *ECCV*, 2022. 1

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *CVPR*, 2017. 2

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

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

[17] 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. 2

[18] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation. In *ICRA*, 2023. 2, 3

[19] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-Voxel CNN for Efficient 3D Deep Learning. In *NeurIPS*, 2019. 2

[20] Chen-Chou Lo and Patrick Vandewalle. Depth estimation from monocular images and sparse radar using deep ordinal regression network. In *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, sep 2021. 2

[21] Jiachen Lu, Zheyuan Zhou, Xiatian Zhu, Hang Xu, and Li Zhang. Learning ego 3d representation as ray tracing. In *European Conference on Computer Vision*, 2022. 1

[22] Tao Lu, Xiang Ding, Haisong Liu, Gangshan Wu, and Limin Wang. Link: Linear kernel for lidar-based 3d perception, 2023. 1

[23] Ramin Nabati and Hairong Qi. Centerfusion: Center-based radar and camera fusion for 3d object detection. *arXiv preprint arXiv:2011.04841*, 2020. 4

[24] Jinhyung Park, Chenfeng Xu, Shijia Yang, Kurt Keutzer, Kris Kitani, Masayoshi Tomizuka, and Wei Zhan. Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. 2023. 1

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

[26] Charles Ruizhongtai Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *CVPR*, 2017. 2

[27] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *NeurIPS*, 2017. 2

[28] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *CVPR*, 2020. 1

[29] Haotian Tang, Zhijian Liu, Xiuyu Li, Yujun Lin, and Song Han. TorchSparse: Efficient Point Cloud Inference Engine. In *MLSys*, 2022. 2

[30] Zizhang Wu, Guilian Chen, Yuanzhu Gan, Lei Wang, and Jian Pu. Mvfusion: Multi-view 3d object detection with semantic-aligned radar and camera fusion, 2023. 1, 4

[31] Enze Xie, Zhiding Yu, Daquan Zhou, Jonah Philion, Anima Anandkumar, Sanja Fidler, Ping Luo, and Jose M Alvarez. Mˆ 2bev: Multi-camera joint 3d detection and segmentation with unified birds-eye view representation. *arXiv preprint arXiv:2204.05088*, 2022. 1

[32] Kaixin Xiong, Shi Gong, Xiaoqing Ye, Xiao Tan, Ji Wan, Errui Ding, Jingdong Wang, and Xiang Bai. Cape: Camera view position embedding for multi-view 3d object detection. 2023. 1

[33] Junjie Yan, Yingfei Liu, Jianjian Sun, Fan Jia, Shuailin Li, Tiancai Wang, and Xiangyu Zhang. Cross modal transformer via coordinates encoding for 3d object dectection. *arXiv preprint arXiv:2301.01283*, 2023. 1

[34] Yan Yan, Yuxing Mao, and Bo Li. SECOND: Sparsely Embedded Convolutional Detection. *Sensors*, 2018. 2, 3

[35] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, Jie Zhou, and Jifeng Dai. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision, 2022. 1

[36] Zeyu Yang, Jiaqi Chen, Zhenwei Miao, Wei Li, Xiatian Zhu, and Li Zhang. Deepinteraction: 3d object detection via modality interaction. In *NeurIPS*, 2022. 1

[37] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3D Object Detection and Tracking. In *CVPR*, 2021. 1

[38] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-Point++ Submission to the Waymo Real-time 3D Detection Challenge. Technical report, 2022. 2

[39] Yunpeng Zhang, Zheng Zhu, Wenzhao Zheng, Junjie Huang, Guan Huang, Jie Zhou, and Jiwen Lu. Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving, 2022. 1

[40] Taohua Zhou, Junjie Chen, Yining Shi, Kun Jiang, Mengmeng Yang, and Diange Yang. Bridging the view disparity between radar and camera features for multi-modal fusion 3d object detection. *IEEE Transactions on Intelligent Vehicles*, 8(2):1523–1535, 2023. 4