2024 IEEE International Conference on Robotics and Automation (ICRA)
May 13-17, 2024. Yokohama, Japan

LiRaFusion: Deep Adaptive LiDAR-Radar Fusion for 3D Object Detection

Jingyu Song, Lingjun Zhao, and Katherine A. Skinner

2024 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-8457-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICRA57147.2024.10611436

***Abstract*— We propose LiRaFusion to tackle LiDAR-radar fusion for 3D object detection to fill the performance gap of existing LiDAR-radar detectors. To improve the feature extraction capabilities from these two modalities, we design an early fusion module for joint voxel feature encoding, and a middle fusion module to adaptively fuse feature maps via a gated network. We perform extensive evaluation on nuScenes to demonstrate that LiRaFusion leverages the complementary information of LiDAR and radar effectively and achieves notable improvement over existing methods.**

I. Introduction

Autonomous vehicles (AVs) are expected to accurately perceive the surrounding environment to enable effective and safe planning and control across a variety of scenarios and environmental conditions [1]–[5]. An important part of the perception task is to precisely localize the objects in the surrounding environment. A common representation for these objects is a set of 3D bounding boxes that have locations, sizes and classes [6], [7]. Despite various combinations of sensor configurations on AVs, many object detection algorithms rely on LiDAR and cameras due to their dense returns [1], [2], [8]–[11].

Still, LiDAR systems and cameras are sensitive to varying weather and lighting conditions, so AVs can suffer significantly from downgraded perceptual capability in these scenarios. To tackle this problem, recent research has focused on leveraging radar systems, which have automotive-grade design that ensures robust performance under various conditions [12]–[15]. Additional benefits of radars include their low cost, long detection range and Doppler effect information (i.e., velocity of captured targets). Therefore, it is of great significance to design a model that could effectively leverage radar for 3D object detection [2], [13], [14].

Existing detectors with radars can be categorized into single-modality methods [16] and fusion-based methods [13], [15]. Recent works [17], [18] have achieved impressive detection accuracy when fusing LiDAR and radars on the Oxford Radar RobotCar dataset [19], which has high-resolution radar data. However, this dataset uses a spinning radar, which lacks Doppler information and has increased cost [12]. Among the popular datasets for 3D object detection [6], [8], [12], [20], nuScenes [7] stands out because it is large-scale and has a complete sensor suite including radars. nuScenes represents radar data as object lists, which is a common representation that could also be interpreted as a very sparse point cloud with additional feature attributes from radar on-board signal processing [12], [13]. The common challenges with this dataset are the sparsity and noise of the radar data. Consequently, single-modality radar detectors fail to achieve reliable performance. Some fusion-based detectors [21], [22] suffer from downgraded performance when adding radar to LiDAR-only or LiDAR-camera fusion detectors, while some detectors [13], [15], [23] have to enforce hard constraints such as limiting detection to specific classes or limiting detection range to achieve improvement in performance. Our work seeks to fill the gap in the current literature by improving the design of fusion architecture for LiDAR and radar by leveraging their shared point cloud representation.

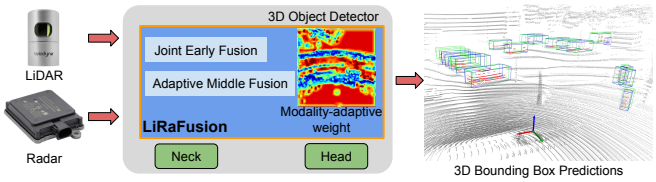

Fig. 1: We propose LiRaFusion to efficiently leverage the complementary information of LiDAR and radar for 3D object detection.

In this work, we propose a novel LiDAR-radar fusion detector, LiRaFusion (Fig. 1). Our main contributions are: (i) a novel joint feature extractor for effective LiDAR-radar fusion, (ii) the first introduction of the adaptive gated network into LiDAR-radar fusion for object detection with novel improvement considering the bird's-eye-view (BEV) feature space, and (iii) extensive evaluation on open source datasets and detectors that demonstrate improvement over existing LiDAR-radar fusion methods. As most existing detectors follow the backbone-neck-head design [10], [21], [24], LiRaFusion can be directly integrated into existing methods by serving as the backbone to enable more modality configurations, which is validated by extending it to LiDAR-camera-radar fusion. Code will be made available on the project website.[1]

II. Related Work

A. Radar Datasets for Autonomous Driving

Data is the key component for enabling development of learning-based object detectors. However, radar data was rarely available in early public autonomous driving datasets [6], [20], [25]. Recently, more datasets with radar data have become accessible to researchers. In these datasets, radar data usually has two representations. The most common representation is the point cloud, in which each point represents an object in the object list output by many off-the-shelf radars with on-board processing algorithms such

This work is supported by the Ford Motor Company via the Ford-UM Alliance under award N028603.

J. Song, L. Zhao, and K. Skinner are with the Department of Robotics, University of Michigan, Ann Arbor, MI, USA jingyuso,lingjunz,kskin@umich.edu.

[1] https://github.com/Song-Jingyu/LiRaFusion

as Constant False Alarm Rate (CFAR) [12], [13]. This representation is available on datasets such as nuScenes [7], aiMotive [26], and Zendar [27]. There are also datasets that directly use the raw data from radars [19], [28]. The raw data has denser information but the lack of CFAR processing leads to increased noise. The spinning radar configuration in [19] loses Doppler information (e.g., velocities of the captured targets), which is important to understand the scene. Another challenge brought by this representation is the difficulty of annotating the data [12]. In this work we leverage the nuScenes dataset [7] because of its full coverage on sensors and driving scenarios, accurate annotations, and popularity, which allows us to compare this work to many existing works. The proposed fusion architecture can be transferred to other datasets that have similar point cloud representation.

B. Multi-modality 3D Object Detection

3D object detection is an important part of the perception system for AVs [1]. The main objective is to assign a class label and a 3D bounding box for each detected object in the scene [8]. Common sensors used for 3D object detection include LiDAR and cameras. Though several single-modality object detectors with LiDAR [24] or cameras [29] achieve impressive results on KITTI [6] or nuScenes [7] benchmarks, multi-modality object detectors have recently shown promise in leveraging complementary information to improve robustness and accuracy [2], [11], [22], [24], [30]–[32]. LiDAR-camera fusion is the most common configuration. However, these two sensors have shared drawbacks (e.g., sparse information at long range, lacking velocity estimation) that could be compensated by radars that are commonly deployed on AVs [12], [13]. Radar-only detectors usually fail to overcome the data sparsity to perform comparably to camera- or LiDAR-based detectors [16], [33].

One recent trend is to fuse radar with one or several other sensors. Existing fusion configurations include camera-radar (CR), LiDAR-radar (LR) and LiDAR-camera-radar (LCR) [13]–[15], [21], [34]–[36]. As the main focus of this work is fusing LiDAR and radar in the shared point cloud representation, we mainly compare our methods with FUTR3D [21] and EZFusion [15] because they are the most recent state-of-the-art detectors supporting LR fusion on the nuScenes [7] dataset. In FUTR3D [21], though LR and LCR configurations are supported, they suffer from downgraded performance when compared with LiDAR-only (LO) or LC fusion. We argue the failure could come from the simple MLP-style feature extractor for radars and lack of joint feature fusion before sampling features for query points. Our proposed method aims to address these limitations with the proposed adaptive fusion framework. EZFusion [15] is built on CenterPoint [24] by adding radar feature projection for the LiDAR points. In EZFusion [15], though its LR fusion shows improvement over its LO configuration (equivalent to CenterPoint [24]), its partial moving class setting, which uses only the 7 moving classes out of 10 classes in the nuScenes benchmark, has more limited application for practical deployment since the missed static classes are also vital for keeping AVs safe. Our proposed method achieves further improvement over EZFusion under the same partial classes setting and is demonstrated on the complete class setting, which has stronger potential in application since it can account for both static and moving object classes.

C. Gated Network for Sensor Fusion

Fusing information from different modalities requires sophisticated architecture design and there are multiple prior works that addressed this challenge [37]. Among them, projection, addition and concatenation are common practices [11], [14], [15], [22]. Though these methods demonstrate improvement on multi-modality fusion, they are not learning-based and lack adaptivity. To account for this issue, researchers have turned to gated networks when different feature maps are fused. This process is also named as the mixture of experts because the backbones used for different modalities are considered different expert networks. This method is first introduced in [38], in which the expert network is defined as a domain-specific neural network to process a single sensing modality, and the gating network is a weighting neural network that selects useful features among the outputs of the expert networks. This idea has been leveraged by the perception community as more works have focused on using different modalities [2], [39]–[41]. For instance, in 3D object detection, 3D-CVF [39] proposes an LC fusion network using a gated network. Extensive evaluation is conducted on the KITTI [6] and nuScenes [7] datasets, which demonstrates the effectiveness of the gated network.

The success of fusing LiDAR and camera modalities through the gated network motivates us to adapt this method for the LR sensor configuration. The gated network is able to learn adaptive weights for different expert networks so the model can learn to be robust to noise from individual experts. We find this feature is of great significance in the LR fusion since the radars are more noisy, which can degrade the performance of multi-modal detectors if the information is not properly handled. In our proposed work, we extend the original gated network design [39], [42] by making it channel-wise so that each channel of the BEV map has an adaptive weight. According to our best knowledge, LiRaFusion is the first to introduce the gated network into LR perception.

III. Method

The goal of our method, LiRaFusion, is to achieve more effective feature extraction and fusion for LiDAR and radar data for 3D object detection (Fig. 2). The inputs to LiRaFusion are a LiDAR point cloud and a radar point cloud. One stream stacks these two point clouds as the input to the proposed early fusion module. The early fusion module processes the denser point cloud with the proposed joint feature encoder and a common sparse 3D convolution encoder. Its output is then fed into a common LiDAR backbone to obtain the feature map. In this work, we use the VoxelNet following [21], [24]. The other stream uses the PointPillars [43] backbone to process the radar points, taking advantage

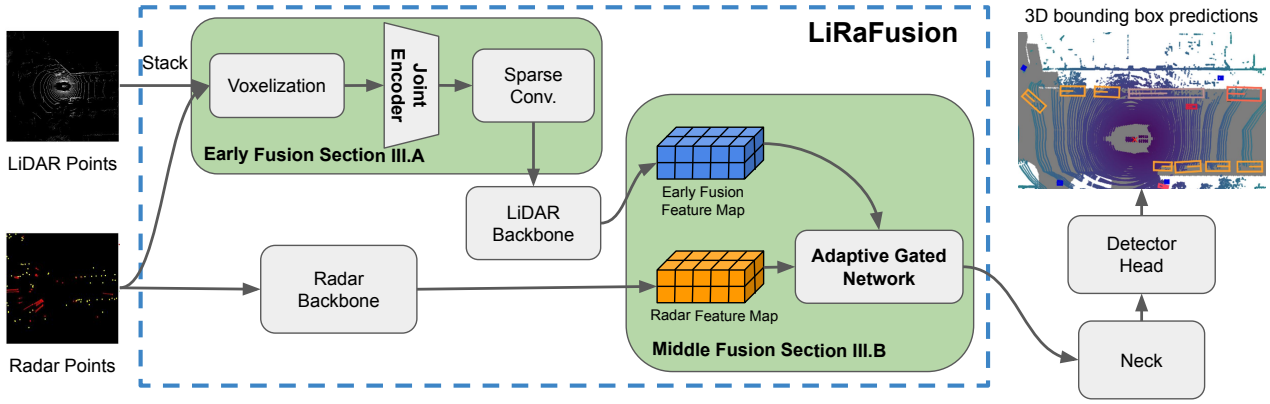

Fig. 2: Overview of the architecture of LiRaFusion. Our main contributions, shown as bold text, mainly include a joint voxel feature encoder to extract per-voxel features from the stacked point cloud, and a gated network to learn weights for each input feature map to fuse them adaptively.

of the pillars since the height measurement for radar points are noisy [12], [13]. The output is a radar feature map. The output feature maps from these two streams can be considered as two experts, which are further fused with the proposed gated network in the middle fusion module. The middle fusion module learns the adaptive weights for these two feature maps and then concatenates the weighted feature maps together. The concatenated feature map is passed into the Feature Pyramid Network (FPN) [44] neck and the detector head to generate predictions. Our main contributions are the novel architectures for the early fusion and middle fusion modules. LiRaFusion is an enhanced backbone for LR feature extraction so it can be extended to LCR configuration as well. Technical details of each module are discussed in the following subsections.

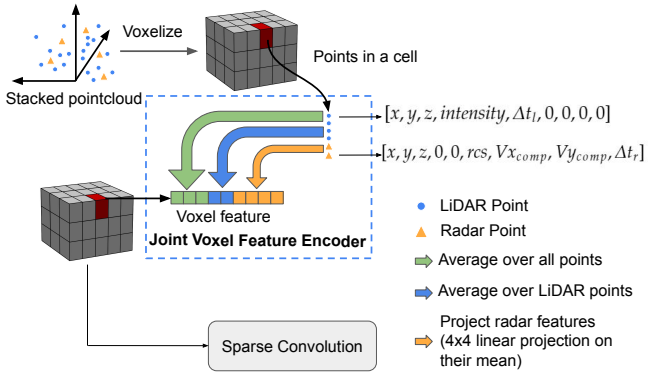

Fig. 3: The network architecture early fusion module. We stack the loaded LiDAR and radar points by zero-padding them to the same number of dimensions before feeding into the proposed joint voxel feature encoder.

A. Early Fusion

We design an early fusion block to fuse the LiDAR and radar points at an early stage to extract features for each voxel cell, as shown in Fig. 3. Unlike [13], [23], [26], which ignore features such as LiDAR intensity, Radar Cross-section (RCS), and velocity, we keep these features since LiDAR intensity and RCS are helpful to classify objects, and velocity information is important to distinguish static or dynamic objects and predict the velocity and rotation. For the LiDAR input, we keep the point intensity and the captured time offset (Δt_l) from the current frame as we accumulate multiple sweeps. For the radar points, we keep the RCS, compensated velocities ($V_{x_{\text{comp}}}, V_{y_{\text{comp}}}$) and the time offset ($\Delta t_r$). We use zero-padding to match the dimension of these points to merge them. After voxelization, we use the proposed joint voxel feature encoder to extract features for each voxel cell. We follow the simplified voxel feature encoder of VoxelNet [10] in MMDetection3D [20] to set the number of the voxel feature dimensions to be the same as the input points. The first 3 feature dimensions represent the location of the centroid of this cell, which is computed by taking the mean locations of all the points in this voxel cell. The following 2 feature dimensions correspond to features from LiDAR so we average over all LiDAR points. The last 4 feature dimensions correspond to the radar features. We pass the mean of the radar features to a 4×4 linear layer to enable to network to learn an appropriate way to handle the radar features. We only process non-empty voxel cells. For cells that do not have radar points, which is common due to sparsity, we leave the last 4 dimensions of these cells as zero. After obtaining the voxel features, we apply standard sparse convolution and further process its output with a standard LiDAR backbone, VoxelNet [10]. To simplify the terminology, we refer to the output of the early fusion stream as the LiDAR feature map in the following sections. Though radar data has already been fused when encoding the voxel feature, due to the sparsity of radar data, most information in the encoded voxel features is from LiDAR. We further fuse it with the radar feature map at a higher level with the proposed middle fusion module.

B. Middle Fusion

In order to perform adaptive sensor fusion on the feature maps from different modalities, we refer to [39], [42] for designing the gated network. To the best of our knowledge, we are the first to bring the adaptive gated network to the field of LR fusion for 3D object detection. We improve the existing gated network by enabling it to adaptively learn

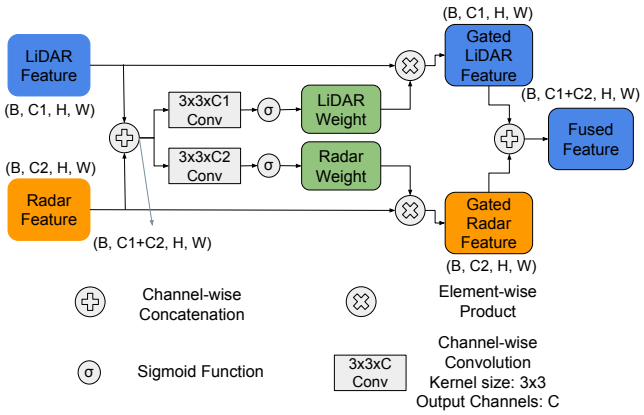

Fig. 4: The network architecture for the middle fusion module. In this module, by applying a channel-wise convolution and a sigmoid function to the concatenated LiDAR-radar feature map, the network generates adaptive weights for LiDAR and radar separately. Then the input LiDAR and radar feature maps are element-wise multiplied with the weights before being concatenated as a fused LiDAR-radar feature map.

the weight over the channel dimension. Specifically, the generated adaptive weight for the input feature maps are currently in the shape $B \times C \times H \times W$ instead of the previous channel-constant style in the shape $B \times 1 \times H \times W$. This change enables the gated network to weight and extract the LiDAR and radar features in a more flexible way. Intuitively, the feature maps are all in bird's-eye-view resulting from being flattened over the z axis. By proposing a channel-specific weight, we improve the capability of the network to exploit the spatial knowledge from the input experts.

The design of the adaptive gated network is shown in Fig. 4. The input LiDAR feature map $(B \times C_1 \times H \times W)$ and input radar feature map $(B \times C_2 \times H \times W)$ are first concatenated in the channel dimension. Then the concatenated feature map is passed to a convolution block that is followed by a sigmoid function. Notably, the output feature dimension of the convolution blocks for LiDAR and radar modalities is set to match the dimension of the input feature maps (C_1 for LiDAR and C_2 for radar). The learned adaptive weights are applied to the original input feature maps through an element-wise product operation. The obtained gated LiDAR and radar features are further concatenated together along the feature dimension as a fused feature map. The resulting fused feature map has the shape $B \times (C_1 + C_2) \times H \times W$ and serves as the output of the middle fusion block.

IV. Experiments

A. Experiment Design

As mentioned in Section II-A, we evaluate our method on the nuScenes dataset [7]. We follow the official split that has 700 scenes for training and 150 scenes for validation. It only has annotations for the key-frames (samples), but also provides non-key-frames (sweeps) without annotations. We follow common practices in [15], [20], [21] to load multiple sweeps into a current sample frame to increase the data density, while appending a time difference channel for data from each sweep as additional temporal information.

We use the mean-Average-Precision (mAP) and nuScenes Detection Score (NDS) [7] as the main evaluation metrics. We compare LiRaFusion with the existing LR and LCR detectors. In addition, we group predictions to break down the improvement of fusing radars. We also report runtime and additional TP (True Positive) metrics from nuScenes [7]. Ablation studies are conducted to validate our model design.

As mentioned previously, many existing detectors [10], [20], [21], [24] follow the backbone-neck-head design. Therefore, LiRaFusion can be integrated by replacing their backbones with LiRaFusion and keeping the same neck and head. As the baselines we choose – FUTR3D [21] and EZFusion [15] – are initially proposed to work with different heads, we group our experiments based on the detector heads. Inspired by [45], [46], FUTR3D [21] uses a transformer-based head, which is referred to as TransHead. EZFusion [15] uses the same head as CenterPoint [24], which is referred to as CenterHead. We choose the LO and LR configurations for FUTR3D as the baselines, denoted as FUTR3D-LO and FUTR3D-LR. We implement the LR fusion strategy in EZFusion with both heads, which is named as EZFusion-LR*, where * represents our re-implementation. The other group of experiments with CenterHead [24] focuses on 7 moving classes out of 10 complete classes in the nuScenes official benchmark for consistency with results reported in EZFusion [15]. We re-train the original CenterPoint [24] with 7 classes and name it as CenterPoint-7. Since FUTR3D-LO and CenterPoint are the state-of-the-art LO detectors, we include them in our comparison to demonstrate the improvement obtained with fusing radar data. We implemented LiRaFusion based on the MMDetection3D framework [20]. More implementation and training details can be found on the project website.

B. Results and Comparison

We perform a comparison of our method with several state-of-the-art LR fusion networks with CenterHead [24] and TransHead [21] separately on the nuScenes dataset [7]. Table I shows the results of all models trained with TransHead [21]. We can see that FUTR3D-LR performs worse than FUTR3D-LO, which proves that ineffective design of LR fusion strategy could actually harm performance. The re-implemented EZFusion with TransHead (EZFusion-LR*) fails to achieve further improvement over FUTR3D-LO. Our model, LiRaFusion, achieves the best results on the nuScenes validation set in terms of NDS and mAP. It also shows impressive improvement on certain classes such as car and pedestrian, the top two most frequent classes, which are critical for AVs to detect in the scene in order to operate effectively and safely.

Table II shows the results for models trained with CenterHead using the 7 moving classes for consistency with EZFusion [15]. Similarly to EZFusion, we re-train CenterPoint with the 7-class setting as a baseline LO detector and name it as CenterPoint-7. Though EZFusion-LR* achieves considerable improvement over CenterPoint-7, there is a small gap between EZFusion-LR* and the reported results

TABLE I: Results with TransHead [21] evaluated on nuScenes *val* set. EZFusion-LR* represents our re-implementation of [15]. All values are percentages. No model ensemble or test-time augmentation is used.

Method	Sensor	NDS ↑	AP (Average Precision) ↑										
			mean (mAP)	Car	Ped	Bicycle	Bus	Barrier	TC	Truck	Trailer	Moto	CV
FUTR3D-LO	LO	65.74	59.39	84.3	81.4	49.0	65.4	**62.4**	64.2	53.5	**41.8**	66.4	25.5
FUTR3D-LR	LR	65.37	58.08	83.8	81.2	**49.8**	65.4	60.4	60.6	51.0	41.0	65.1	22.6
EZFusion-LR*	LR	65.77	59.24	84.6	81.7	47.3	69.1	62.0	65.7	52.2	39.7	66.9	23.3
LiRaFusion (ours)	LR	**66.69**	**60.11**	**85.6**	**82.2**	46.9	**69.6**	61.2	**66.0**	**54.0**	40.7	**68.1**	**26.7**

Abbreviations: pedestrian (Ped), traffic cone (TC), motorcycle (Moto), and construction vehicle (CV).

TABLE II: Results with CenterHead [24] evaluated on nuScenes *val* set. We train these networks with 7 moving classes. EZFusion-LR stands for the results in its original paper [15]. * represents our re-implementation. In this experiment group, test-time augmentation was used to keep consistency with EZFusion [15]. No model ensemble is used.

Method	Sensor	NDS ↑	AP (Average Precision) ↑							
			mean (mAP)	Car	Ped	Bicycle	Bus	Truck	Trailer	Moto
CenterPoint-7	LO	69.41	61.38	86.0	79.1	40.3	68.7	**57.7**	37.1	60.7
EZFusion-LR	LR	N/A	63.21	N/A	N/A	N/A	N/A	N/A	N/A	N/A
EZFusion-LR*	LR	69.85	61.85	86.0	79.1	43.6	70.3	57.4	35.6	60.8
LiRaFusion (ours)	LR	**72.16**	**65.18**	**86.8**	**79.4**	**54.8**	**70.4**	57.5	**38.7**	**68.7**

TABLE III: Comparison with LR and LCR detectors on nuScenes *val* (**top**) and *test* (**bottom**) set. The reported LiRaFusion results are using TransHead [21]. No test time augmentation or model ensemble is used. Results for other models are the reported results in the papers so some entries are missing. All values are percentages.

Method	Sensor	NDS	mAP	AP (car)	AP (Moto)
RadarNet [13]	LR	N/A	N/A	84.3	53.7
RVF-Net [14]	LR	N/A	N/A	54.18	N/A
RVF-Net [14]	LCR	N/A	N/A	54.86	N/A
DeepFusion [22]	LCR	N/A	N/A	83.5	N/A
LiRaFusion (ours)	LR	66.7	60.1	85.6	68.1
Sparse-PointNet [23]	LCR	N/A	48.9	71	36
Frustum PointNet [47]	LCR	N/A	36.6	48	41
LiRaFusion (ours)	LR	66.2	59.5	84.7	63.3

TABLE IV: Runtime and True Positive (TP) metrics for experiments evaluated with TransHead [21]. Runtime is measured using frames per second (FPS). Lower is better across all metrics.

Method	Runtime	mATE	mASE	mAOE	mAVE	mAAE
FUTR3D-LO	6.0 FPS	**0.342**	**0.265**	0.321	0.276	0.193
FUTR3D-LR	5.6 FPS	0.347	0.267	0.308	0.256	0.189
EZFusion-LR*	5.9 FPS	**0.342**	0.266	0.315	0.267	0.196
LiRaFusion (ours)	5.5 FPS	0.346	0.267	**0.298**	**0.240**	**0.186**

Abbreviations: mean Average Translation Error (mATE), mean Average Scale Error (mASE), mean Average Orientation Error (mAOE), mean Average Velocity Error (mAVE), mean Average Attribute Error (mAAE).

in EZFusion [15] so we include both of them in the table. LiRaFusion is the top performer in terms of NDS and mAP. Similar to its performance with TransHead [21], LiRaFusion achieves impressive improvement in terms of AP over almost all classes.

In addition to the comparison with the most recent state-of-the-art LR detectors [15], [21] shown above, we compare LiRaFusion (with TransHead) with other LR or LCR detectors for a complete overview. Table III demonstrates that LiRaFusion has consistent improvement over existing LR and LCR detectors. It is worth mentioning that many detectors in Table III have to enforce a partial class setting, or use the extra camera modality, while LiRaFusion trains on the complete class setting with the LR sensor configuration and outperforms all the LR and LCR detectors.

We additionally report the model runtime and the True Positive (TP) metrics defined in [7] evaluated with the complete class setting in Table IV. The table shows LiRaFusion is comparable with other baselines in runtime, which is measured on the same desktop with an RTX A6000 GPU. We can also see LiRaFusion is comparable with the best method in terms of translation and scale error, while achieving the lowest error in estimating orientation, velocity and attribute. We argue the significant reduction in orientation and velocity error comes from the effective fusion of the Doppler information from radars.

We also group predictions based on object distances from the ego-vehicle and weather conditions to further break down the performance boost, as this directly represents the improvement from radars. Table V shows the performance breakdown based on the distance. We find the performance gain increases with increasing distance. This finding matches the expectation that radars complement LiDAR by providing additional information at distant locations where the LiDAR returns become sparser. Table VI shows that LiRaFusion achieves notable improvement over the LO baselines in rainy scenes, where LiDAR is generally believed to have reduced detection capability [13], [36]. This finding validates the importance of fusing radars and leveraging their robustness across different weather conditions.

C. Qualitative Results

Figure 5 shows qualitative comparison of LiRaFusion and FUTR3D-LO by presenting the predictions along with the ground truth bounding boxes. These results show that the radar sensor contributes several measurement points (shown in magenta) for a car that was previously missed by FUTR3D-LO, which only uses LiDAR data. The radar and LiDAR points are used effectively by LiRaFusion and

TABLE V: Performance by object distance of experiments with TransHead [21]. We report the mAP separately in 3 ranges: $0m - 20m$, $20m - 30m$ and $30m - 50m$. With increasing object distance, LiRaFusion demonstrates higher gain over FUTR3D-LO as radars complement LiDAR at far locations where LiDAR suffers from data sparsity. All values are percentages.

Method	mAP		
	$[0m, 20m]$	$[20m, 30m]$	$[30m, 50m]$
FUTR3D-LO	73.86	55.2	29.93
LiRaFusion (ours)	74.14 ↑ 0.28	55.88 ↑ 0.68	31.73 ↑ 1.8

TABLE VI: Performance by weather conditions of experiments with TransHead [21]. We report results of two weather conditions: Sunny and Rainy. The grouping strategy is based on the official scene description entries in nuScenes [7]. There are in total 968 rainy frames out of 6019 frames. All values are percentages.

Method	NDS		mAP	
	Sunny	Rainy	Sunny	Rainy
FUTR3D-LO	65.55	65.39	59.28	57.32
LiRaFusion (ours)	66.42 ↑ 0.87	67.15 ↑ 1.76	59.93 ↑ 0.65	59.35 ↑ 2.03

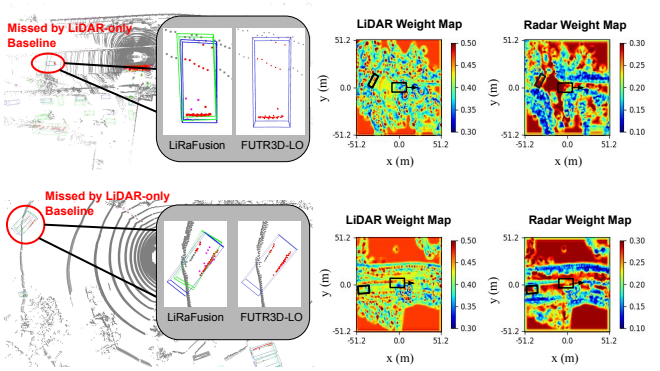

Fig. 5: Example bounding box predictions and corresponding weight maps. We present two frames in which LiRaFusion correctly detects a car (highlighted with a red circle) that is missed by the baseline LO detector. We also show a zoomed-in view in which we label radar points in magenta, and LiDAR points in gray or red (if they reside in a bounding box). We show ground truth bounding boxes in blue and predictions in green. In the visualization of weight maps, the black bounding box with arrow denotes the ego-vehicle. Boxes without an arrow denote the highlighted missed car object. Best viewed in color and zoomed-in.

the prediction by LiRaFusion aligns well with the ground truth.

We also show the corresponding adaptive weights on LiDAR and radar feature maps in Fig. 5. We see that the learned weight on the LiDAR feature map is generally higher than that in the radar feature map, which meets the expectation that LiDAR is the preferred sensor for object detection in terms of the density and geometric accuracy. We also notice that in the radar weight map, locations with larger distance to the ego-vehicle have relatively larger weight. This finding matches our intuition that incorporating radar data provides more information for long-range detection capabilities [12], [13], [48]. Basically, the smaller radar weight at the near locations means that the detector learns to be more dependent on LiDAR at places where it has dense returns, while trusting radar returns at farther locations where LiDAR data has reduced density. Note that the radar weight within the black bounding box in Fig. 5 is relatively large, which means that the proposed gated network is capable of learning to use the expert feature maps adaptively to let the LiDAR and the radar complement each other in a more effective way.

TABLE VII: Ablation study of fusion module.

Method	NDS ↑	mAP ↑	mAVE ↓	AP (car) ↑
FUTR3D-LO	65.74	59.39	0.276	84.3
LiRaFusion-early	65.95	59.11	0.263	85.4
LiRaFusion-middle	66.59	59.45	0.241	85.3
LiRaFusion	**66.69**	**60.11**	**0.240**	**85.6**

D. Ablation Studies

We study the contribution of each fusion module to the overall performance. We denote LiRaFusion-early as the model with the early fusion module only and LiRaFusion-middle as the model with the middle fusion module only. We report the results of these models in addition to the baseline FUTR3D-LO. As shown in Table VII, both LiRaFusion-early and LiRaFusion-middle achieve improvement over the baseline in most metrics. When the early fusion and middle fusion modules are combined together, the performance of the combined model is further enhanced.

TABLE VIII: Ablation study of gated network design.

Method	NDS ↑	mAP ↑
LiRaFusion-middle (Channel-constant)	66.01	58.84
LiRaFusion-middle (Channel-specific)	**66.59**	**59.45**

When designing the adaptive gated network used in the middle fusion module, we improve over the existing network design in [39], [42] that has a constant weight (referred as channel-constant) for all features at one location in the bird's-eye-view (BEV) feature map. Since the z dimension and original feature dimension are merged together to form the BEV feature map, a specific weight for each feature dimension could help to utilize the spatial knowledge. Inspired by this, we propose a channel-specific weight map in the gated network. As shown in Table VIII, the proposed channel-specific gated network outperforms the original network design, which validates the effectiveness of our improvement over the original gated network in [39], [42].

E. LiDAR-Camera-Radar Fusion

We explore the potential of LiRaFusion to fuse LiDAR, camera, and radar for 3D object detection as they are the common sensing modalities on modern AVs. Since most object detectors follow the backbone-neck-head paradigm, LiRaFusion can be applied to many LiDAR-camera (LC) detectors by replacing the LiDAR backbone to enable LCR fusion. FUTR3D [21] supports LC configuration (FUTR3D-LC) and is one of the state-of-the-art LC detectors. By replacing its LiDAR backbone to LiRaFusion, we implemented the LCR model referred as LiRaFusion-LCR. We directly compare LiRaFusion with FUTR3D-LC to evaluate the scalability of LiRaFusion. Table IX shows that LiRaFusion-LCR achieves further improvement over FUTR3D-LC. The results also demonstrate that radars can complement the LC configuration, which reinforces the importance of fusing radar data in modern object detectors. As the main focus of this project is on LR fusion, we leave more experiments on LCR fusion for future work, and hope our work can inspire more research on fusing radars with other sensors to improve perceptual capabilities of AVs.

TABLE IX: Results with LCR fusion and TransHead [21].

Method	NDS ↑	mAP ↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
FUTR3D-LC	68.0	64.2	0.350	**0.259**	0.304	0.305	0.193
LiRaFusion-LR	66.69	60.11	0.346	0.267	**0.298**	**0.240**	0.186
LiRaFusion-LCR	**68.65**	**64.76**	**0.345**	**0.259**	0.309	0.276	**0.181**

Abbreviations: mean Average Translation Error (mATE), mean Average Scale Error (mASE), mean Average Orientation Error (mAOE), mean Average Velocity Error (mAVE), mean Average Attribute Error (mAAE).

V. Conclusion

We have proposed a novel LiDAR-radar fusion network, LiRaFusion, to facilitate cross-modality feature extraction for 3D object detection. We design a joint voxel feature encoder to extract voxel feature encoding in an early stage. We propose an adaptive gated network to further fuse the feature maps from LiDAR and radar by learning modality-adaptive weight maps. Experimental results show that LiRaFusion achieves consistent improvement over existing LiDAR-radar detectors on the nuScenes benchmark. Future work includes applying LiRaFusion to existing LiDAR-camera detectors to further improve over existing LCR detectors, and also extending LiRaFusion to other scene understanding tasks.

REFERENCES

[1] J. Mao, S. Shi, X. Wang, and H. Li, "3d object detection for autonomous driving: a review and new outlooks," *arXiv preprint arXiv:2206.09474*, 2022.

[2] Y. Wang, Q. Mao, H. Zhu, J. Deng, Y. Zhang, J. Ji, H. Li, and Y. Zhang, "Multi-modal 3d object detection in autonomous driving: a survey," *International Journal of Computer Vision*, pp. 1–31, 2023.

[3] J. Wilson, J. Song, Y. Fu, A. Zhang, A. Capodieci, P. Jayakumar, K. Barton, and M. Ghaffari, "Motionsc: Data set and network for real-time semantic mapping in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8439–8446, 2022.

[4] J. Wilson, Y. Fu, A. Zhang, J. Song, A. Capodieci, P. Jayakumar, K. Barton, and M. Ghaffari, "Convolutional bayesian kernel inference for 3d semantic mapping," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 8364–8370.

[5] Z. Xu, G. P. Kontoudis, and K. G. Vamvoudakis, "Online and robust intermittent motion planning in dynamic and changing environments," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[6] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[7] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[8] R. Qian, X. Lai, and X. Li, "3d object detection for autonomous driving: a survey," *Pattern Recognition*, p. 108796, 2022.

[9] A. Zhang, C. Eranki, C. Zhang, J.-H. Park, R. Hong, P. Kalyani, L. Kalyanaraman, A. Gamare, A. Bagad, M. Esteva *et al.*, "Towards robust robot 3d perception in urban environments: The ut campus object dataset," *arXiv preprint arXiv:2309.13549*, 2023.

[10] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[11] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "Pointpainting: Sequential fusion for 3d object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[12] Y. Zhou, L. Liu, H. Zhao, M. López-Benítez, L. Yu, and Y. Yue, "Towards deep radar perception for autonomous driving: Datasets, methods, and challenges," *Sensors*, vol. 22, no. 11, p. 4208, 2022.

[13] B. Yang, R. Guo, M. Liang, S. Casas, and R. Urtasun, "Radarnet: Exploiting radar for robust perception of dynamic objects," in *Computer Vision – ECCV 2020*. Springer, 2020, pp. 496–512.

[14] F. Nobis, E. Shafiei, P. Karle, J. Betz, and M. Lienkamp, "Radar voxel fusion for 3d object detection," *Applied Sciences*, vol. 11, no. 12, p. 5598, 2021.

[15] Y. Li, J. Deng, Y. Zhang, J. Ji, H. Li, and Y. Zhang, "ezfusion: A close look at the integration of lidar, millimeter-wave radar, and camera for accurate 3d object detection and tracking," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 182–11 189, 2022.

[16] M. Ulrich, S. Braun, D. Köhler, D. Niederlöhner, F. Faion, C. Gläser, and H. Blume, "Improved orientation estimation and detection with hybrid object detection networks for automotive radar," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2022, pp. 111–117.

[17] K. Qian, S. Zhu, X. Zhang, and L. E. Li, "Robust multimodal vehicle detection in foggy weather using complementary lidar and radar signals," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[18] Y.-J. Li, J. Park, M. O'Toole, and K. Kitani, "Modality-agnostic learning for radar-lidar fusion in vehicle detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.

[19] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner, "The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6433–6438.

[20] M. Contributors, "MMDetection3D: OpenMMLab next-generation platform for general 3D object detection," https://github.com/open-mmlab/mmdetection3d, 2020.

[21] X. Chen, T. Zhang, Y. Wang, Y. Wang, and H. Zhao, "Futr3d: A unified sensor fusion framework for 3d detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2023.

[22] F. Drews, D. Feng, F. Faion, L. Rosenbaum, M. Ulrich, and C. Gläser, "Deepfusion: A robust and modular 3d object detector for lidars, cameras and radars," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 560–567.

[23] L. Wang and B. Goldluecke, "Sparse-pointnet: See further in autonomous vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7049–7056, 2021.

[24] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[25] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[26] T. Matuszka, I. Barton, Á. Butykai, P. Hajas, D. Kiss, D. Kovács, S. Kunsági-Máté, P. Lengyel, G. Németh, L. Pető, D. Ribli, D. Szeghy, S. Vajna, and B. V. Varga, "aimotive dataset: A multimodal dataset for robust autonomous driving with long-range perception," in *International Conference on Learning Representations 2023 Workshop on Scene Representations for Autonomous Driving*, 2023.

[27] M. Mostajabi, C. M. Wang, D. Ranjan, and G. Hsyu, "High-resolution radar dataset for semi-supervised learning of dynamic objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2020.

[28] M. Sheeny, E. De Pellegrin, S. Mukherjee, A. Ahrabian, S. Wang, and A. Wallace, "Radiate: A radar dataset for automotive perception in bad weather," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 1–7.

[29] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *Computer Vision – ECCV 2022*. Springer, 2022, pp. 1–18.

[30] V. A. Sindagi, Y. Zhou, and O. Tuzel, "Mvx-net: Multimodal voxelnet for 3d object detection," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7276–7282.

[31] D. Xu, D. Anguelov, and A. Jain, "Pointfusion: Deep sensor fusion for 3d bounding box estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[32] S. Pang, D. Morris, and H. Radha, "Clocs: Camera-lidar object candidates fusion for 3d object detection," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 386–10 393.

[33] Y. Wang, Z. Jiang, Y. Li, J.-N. Hwang, G. Xing, and H. Liu, "Rodnet: A real-time radar object detection network cross-supervised by camera-radar fused object 3d localization," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 4, pp. 954–967, 2021.

[34] R. Nabati and H. Qi, "Centerfusion: Center-based radar and camera fusion for 3d object detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2021.

[35] V. John and S. Mita, "Rvnet: Deep sensor fusion of monocular camera and radar for image-based obstacle detection in challenging environments," in *Pacific-Rim Symposium on Image and Video Technology*. Springer, 2019, pp. 351–364.

[36] M. Bijelic, T. Gruber, F. Mannan, F. Kraus, W. Ritter, K. Dietmayer, and F. Heide, "Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[37] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1341–1360, 2020.

[38] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991.

[39] J. H. Yoo, Y. Kim, J. Kim, and J. W. Choi, "3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection," in *Computer Vision – ECCV 2020*. Springer, 2020, pp. 720–736.

[40] S. Zhao, M. Gong, H. Fu, and D. Tao, "Adaptive context-aware multi-

modal network for depth completion," *IEEE Transactions on Image Processing*, vol. 30, pp. 5264–5276, 2021.

[41] D. Qiao and F. Zulkernine, "Adaptive feature fusion for cooperative perception using lidar point clouds," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2023.

[42] J. Kim, J. Koh, Y. Kim, J. Choi, Y. Hwang, and J. W. Choi, "Robust deep multi-modal learning based on gated information fusion network," in *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part IV*. Springer, 2019, pp. 90–106.

[43] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[44] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[45] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision – ECCV 2020*. Springer, 2020, pp. 213–229.

[46] Y. Wang, V. Guizilini, T. Zhang, Y. Wang, H. Zhao, , and J. M. Solomon, "Detr3d: 3d object detection from multi-view images via 3d-to-2d queries," in *The Conference on Robot Learning (CoRL)*, 2021.

[47] L. Wang, T. Chen, C. Anklam, and B. Goldluecke, "High dimensional frustum pointnet for 3d object detection from camera, lidar, and radar," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1621–1628.

[48] M. I. Skolnik, *Radar handbook*. McGraw-Hill Education, 2008.