EZFusion: A Close Look at the Integration of LiDAR, Millimeter-Wave Radar, and Camera for Accurate 3D Object Detection and Tracking

Yao Li, Jiajun Deng, Yu Zhang, Jianmin Ji, Houqiang Li, *Fellow, IEEE*, and Yanyong Zhang, *Fellow, IEEE*

***Abstract*—A recent trend is to combine multiple sensors (*i.e.*, cameras, LiDARs and millimeter-wave Radars) to achieve robust multi-modal perception for autonomous systems such as self-driving vehicles. Although quite a few sensor fusion algorithms have been proposed, some of which are top-ranked on various leaderboards, a systematic study on how to integrate these three types of sensors to develop effective multi-modal 3D object detection and tracking is still missing. Towards this end, we first study the strengths and weaknesses of each data modality carefully, and then compare several different fusion strategies to maximize their utility. Finally, based upon the lessons learnt, we propose a simple yet effective multi-modal 3D object detection and tracking framework (namely EZFusion). As demonstrated by extensive experiments on the nuScenes dataset, without fancy network modules, our proposed EZFusion makes remarkable improvements over the LiDAR-only baseline, and achieves comparable performance with the state-of-the-art fusion-based methods.**

***Index Terms*—Object detection, segmentation and categorization, sensor fusion, visual tracking.**

I. Introduction

In THE past few years, object detection and tracking [1]–[6] has attracted a great deal of research interest in autonomous systems such as self-driving cars. Compared to the 2D counterpart, 3D detection and tracking can leverage the accurate location information to generate high-level 3D features and trajectories, which can help autonomous vehicles (AVs) make better navigation decisions and thus improve the overall driving safety. A recent trend in 3D detection and tracking is to consider multiple data sources that may complement each other. For examples, methods in [7], [8] improve the 3D object detection accuracy by fusing camera images and LiDAR point clouds. 3D multi-object tracking schemes in [9], [10] propose to further leverage camera images to extract more discriminating re-identification embedding features, when its primary data modality, LiDAR, does not offer desired color or texture information. Methods in [11], [12] fuse millimeter-wave Radar (referred to as Radar in this letter) point clouds with camera images to obtain more reliable depth information. Combining the advantages of different sensors, these multi-modal methods can considerably improve the detection and tracking performances.

However, how to efficiently integrate these sensors has largely remained a hit-or-miss process. Several important questions remain unanswered in this space. To name a few, how much does each type of sensor data contribute to the detection and tracking performances? How should the fusion among the three modalities be carried out? Should one fuse LiDAR and camera data first and then add Radar, or in a different order? Should one employ point-wise fusion between LiDAR and Radar points or convert them to the bird's eye view (BEV) plane and perform fusion there?

To answer these questions, in this work, we take a close look at the integration of these three types of sensor data by conducting a systematic comparison. In particular, we focus our scope on the two important design questions: 1) what to fuse, and 2) how to fuse. For the what-to-fuse question, we first provide a comparison between the LiDAR, Radar, and camera sensors from several perspectives. And then we study several different fusion input configurations: LiDAR only, LiDAR-camera fusion, LiDAR-Radar fusion, and LiDAR-Radar-camera fusion, and quantify their detection and tracking performance carefully. While the three-way fusion (LiDAR-Radar-camera) provides the best performance as expected, each modality contributes to different metrics in a different scale. Extensive experiments reveal unique effects of different modalities with their own physical properties on detection and tracking system. For the how-to-fuse question, we compare three fusion strategies with all three types of sensor data, and identify the most efficient fusion strategy–first painting each LiDAR point with corresponding image features, and then combining the LiDAR-camera features based on painted points and the Radar features on the BEV plane.

After exploring the above design choices, we put together a pared-down 3D detection and tracking framework that integrates all three modalities, which we refer to as EZFusion.

Manuscript received 3 April 2022; accepted 27 June 2022. Date of publication 25 July 2022; date of current version 26 August 2022. This letter was recommended for publication by Associate Editor B. Noack and Editor E. Marchand upon evaluation of the reviewers' comments. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0100500, and in part by Anhui Province Development and Reform Commission 2020 New Energy Vehicle Industry Innovation Development Project and 2021 New Energy and Intelligent Connected Vehicle Innovation Project *(Corresponding author: Yanyong Zhang.)*

Yao Li, Yu Zhang, Jianmin Ji, and Yanyong Zhang are with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230052, China (e-mail: zkdly@mail.ustc.edu.cn; yuzhang@ustc.edu.cn; jianmin@ustc.edu.cn; yanyongz@ustc.edu.cn).

Jiajun Deng and Houqiang Li are with the Department of Electric Engineering and Information Science, University of Science and Technology of China, Hefei 230052, China (e-mail: djiajun1206@gmail.com; lihq@ustc.edu.cn).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2022.3193465, provided by the authors.

Digital Object Identifier 10.1109/LRA.2022.3193465

EZFusion only contains the bare minimum network modules to demonstrate the effectiveness of our fusion framework. More sophisticated modules can be easily added later to obtain further improved results. Even with such a simple pipeline, our EZFusion has an improvement of 4.7% in mAP, and 6.7% in AMOTA compared to the LiDAR-only baseline on the nuScenes validation set [13]. Our results are comparable to state-of-the-art fusion methods, while our framework don't need much bulkier operations.

II. Related Work

Fusion-based 3D Object Detection: 3D object detection based on multi-modal fusion has received some attention recently, in which the fusion of LiDAR and camera data is the most common. The mainstream fusion methods could be classified as point-wise, BEV-based and ROI-level manners. PointPainting [7] and EPNet [14] fuse image segmentation scores or other image features with LiDAR points in a point-wise manner. 3D-CVF [8] explores a deep architecture for fusing camera and LiDAR data both on the BEV plane and at the region of interest (ROI) level. Pointaugmenting [15] achieves an excellent detection performance only in a BEV-based fusion manner with cross-LiDAR-camera data augmentation. To explore more dynamic information in scenes, some work also considers the Radar data. RadarNet [16] fuses LiDAR and Radar data in both early and late fusion manners for better detection of dynamic objects. MVDNet [17] proposes a two-stage deep fusion method between LiDAR and Radar, it shows robust performance though in adverse weather conditions.

Fusion-based 3D Object Tracking. Most of 3D object tracking work follows the "tracking by detection" paradigm – in the first stage, the 3D detector predicts the locations and categories of the objects of interest; in the second stage, the tracking module associates objects in consecutive frames. Such as AB3D [6] is a typical baseline in this way, which performs the data association with Hungarian algorithm based on detection results. Several multi-modal based 3D tracking methods in this manner have been proposed recently. JRMOT [9] fuses 3D point cloud features and 2D image features, providing multiple affinity matrices for data association. GNN3DMOT [18] proposes a feature interaction mechanism with the Graph Neural Network to make the features more discriminative in data association, utilizing both 2D and 3D features in consecutive frames. Alphatrack [10] appends an additional image-based appearance branch for more robust tracking performances. Centerfusion [11] and cftrack [12] fuse Radar and camera data for both 3D object detection and tracking.

The above methods provide successful point solutions for dual-modal fusion based detection and tracking. In this work, we set out to look for a versatile and systematic framework that can serve as a base for the fusion of LiDAR, Radar and camera images for improved 3D detection and tracking.

III. Study of Different Fusion Inputs and Fusion Strategies

In this section, we carefully study two important questions in designing a multi-modal fusion module for both detection and tracking: 1) what to fuse, and 2) how to fuse. For each question, we compare multiple options and report their performances on the nuScenes dataset [13]. Section III-A first provides a comparison of LiDAR, camera and Radar sensor parameters. Then with Section III-C for detection and Section III-D for tracking, we answer the question of "what to fuse". Finally, Section III-E is to answer the question of "how to fuse" by comparing several different fusion strategies.

A. Qualitative Comparison of LiDAR, Camera and Radar Sensors

Before investigating the fusion options, we first provide a qualitative comparison of LiDAR, camera and Radar sensors in Table I, by extracting the information from the nuScenes dataset.

First of all, the LiDAR currently offers the most precise 3D measurements among the three sensors, also the most expensive in both cost and power. LiDAR has a much higher angular resolution compared to Radar – the nuScenes one delivers a real-time 360° horizontal field of view with a rotating head design, which results in $\sim$35,000 points per frame. The main drawback is that the LiDAR measurement performance degrades in certain weather conditions such as haze, rain, snow, sand and dust. Due to its ranging accuracy, we consider the LiDAR the base sensor for this study.

Cameras are much more affordable and power-efficient. Even with its lack of depth information, we argue that cameras remain indispensable for autonomous driving as they provide essential semantic information. The camera in nuScenes delivers 12 frames per second at 2 MP resolution. Most autonomous vehicles rely on camera images for important information such as drive-able area detection, lane detection, etc. Therefore, it is rather convenient to fuse camera images with point clouds since images are already processed in the system.

Radar (radio detection and ranging) detects the distance, angle, and radial velocity of objects using radio waves. Because the penetration ability of the radio waves in millimetre band is strong, millimeter-wave Radar (depicted by Radar in this letter) is robust to adverse weather. It has been deployed in AVs for a long time due to low cost and robustness. The raw data in the form of time-frequency signal of Radar is processed by clustering mostly, and then followed by tracking on the clusters. We focus on the data form of Point Targets (Radar clusters) for a trade-off between information richness and low noise. The output data form of Radar on nuScenes are Point Targets. Therefore, the amount of Radar data is orders of magnitude smaller than the other two. As a result, fusing such sparse Radar data does not add much computation or power.

To summarize, we take the viewpoint that the fusion of these three modalities is very worthwhile. There is no one sensor that can function well alone in all weather under all the settings. Multi-modal fusion is important for enhancing the safety of autonomous driving.

B. Experiment Setup

This section briefly overview the dataset and metrics exploited in our study. The implementation details of the involved models

TABLE I
A COMPARISON OF LiDAR, CAMERA AND RADAR SENSOR PARAMETERS ON THE NUSCENES DATASET

Sensors	Capture frequency & Output	Amount of data per frame	Detection range & Accuracy	Angular resolution & Velocity accuracy	Power & Price*	Environment factors
LiDAR	20Hz, & 3D LiDAR Data Points	∼35,000 points	Up to 100 m & Up to ±2 cm	0.1° ∼ 0.4° & -	12 W typ. & ∼$11924.46	scattered by particles
camera	12Hz, & Image Pixels, JPEG compressed	1600 px × 1200 px	-	-	2.1 W typ. & ∼$521.70	limited by lighting
Radar	13Hz, & Point Targets (clusters)	∼200 targets*	Up to 250 m & ±0.40 m/far, ±0.10 m/near	1.6°/far, 3.2°∼12.3°/near & ±0.1 km/h	6.6 W typ. & ∼$596.22	robust to whether

Price* means the price is obtained from mainland china, which is converted to USD. There are 6×Radars (to cover 360° field of view in horizontal) and 1×LiDAR on nuscenes, so 200 targets* represents the amount of radar point targets obtained from 6×Radars

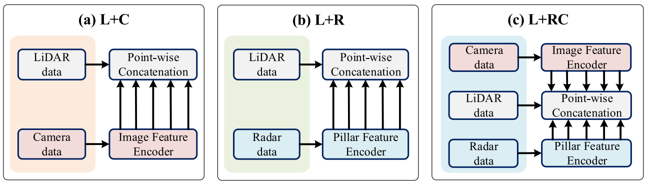

Fig. 1. Pipelines with different fusion inputs: (a) LiDAR-camera fusion (*L+C*), (b) LiDAR-Radar fusion (*L+R*), (c) LiDAR-Radar-camera fusion (*L+RC*). Point-wise fusion is adopted in all cases. The Pillar feature encoder and Image Feature Encoder are Pillar feature Net [1] and DLA-34 [24] respectively.

TABLE II
DETECTION RESULTS WITH DIFFERENT FUSION INPUTS ON NUSCENES VALIDATION SET: *LO*, *L+C*, *L+R*, AND *L+RC*

Input Modality	mAP (%) ↑	mAVE (m/s) ↓	mATE (m) ↓	FPS(Hz) ↑
LO	61.27	0.303	0.253	**1.28**
L+C	64.78	0.282	**0.239**	1.00
L+R	63.21	0.267	0.251	1.22
L+RC	**65.50**	**0.261**	0.248	0.82

The results of mAVE and mATE are given for a more comprehensive comparison and also follow the detection evaluation methods of nuscenes. We test the FPS of different models on the GeForce RTX 3090 GPU and the batch size is set to be 1.

are elaborated in Section V. All of the models follow the same training strategy.

Our experiments are conducted on the nuScenes dataset [13], which contains rich driving scenarios recorded by multiple sensors (6×cameras, 1×LiDAR and 5×Radars). There are 23 object classes annotated on nuScenes dataset, we choose 7 typical moving classes among them, *i.e.*, bicycle, bus, car, motorcycle, pedestrian, trailer and truck, for more challenging in our experiments. Unfortunately, we cannot conduct extensive experiments on KITTI [19] or Waymo [20] because these two datasets do not provide the Radar data. Our experiments are based on the stat-of-the-art 3D LiDAR detector CenterPoint [21] on nuScenes. Because most detectors follow the "Backbone-Neck-Head" paradigm like CenterPoint, the design of our fusion framework can be applied to most 3D detectors.

Several standard metrics are exploited, including NDS (nuScenes detection score, with respect to all the detection metrics of nuScenes comprehensively), AP (average detection precision), AVE (average absolute velocity error in m/s), ATE (average translation error in meters), AMOTA (average multi object tracking accuracy), AMOTP (average multi object tracking precision), MOTA (multi object tracking accuracy), MOTP (multi object tracking precision), and IDS (number of identity switches). The most important metrics in detection and tracking are mAP ('m' means multi-class, mAP evaluates both location and classification performances of the detection model) and AMOTA (this measure combines all three error sources: false positives, missed targets and identity switches of the tracking model), respectively. More details of these metrics can be found in [6], [13], [22], [23].

C. Comparison of Fusion Inputs for 3D Detection

In this section, we first compare several different fusion modalities for 3D detection task (corresponding network architectures shown in Fig. 1(a)–(c)). Here, we use the LiDAR modality as the baseline since it provides the most reliable 3D detection performance when used alone among the three. Meanwhile, to guarantee the generalization of our experiments, we choose the point-wise fusion manner which does not rely on the components in detection so that we can quantify the effect of different modalities independent of the detector. Specifically, We consider the following configurations:

1) *L+C*, in which we have both LiDAR point clouds and camera images as fusion input (network architecture shown in Fig. 1(a)). We project LiDAR points onto the 2D image plane to sample corresponding image features, and then concatenate each LiDAR point with the corresponding image features in a point-wise manner, aka, 'painting' the points [7]. The concatenated features are then fed to the 3D detector. We use CenterPoint [21] retrained with the configuration of 7 typical moving classes as the 3D base detector without additional statement in our study;
2) *L+R*, in which we have both LiDAR point clouds and Radar point clouds as fusion input (network architecture shown in Fig. 1(b)). In this case, we encode Radar points in a pillar-based manner [1] to generate Radar BEV feature maps. Then, we project LiDAR points onto the Radar BEV feature maps to find the corresponding Radar features. After that, we concatenate each LiDAR point with the corresponding Radar features in the same point-wise manner;
3) *L+RC*, in which we fuse all three data sources (network architecture shown in Fig. 1(c)). For fair comparison, we process camera and Radar data in the same way as above, and concatenate them with LiDAR points in the same point-wise manner.

Table II compares the above three inputs configurations against the LiDAR-only (*LO*) baseline:

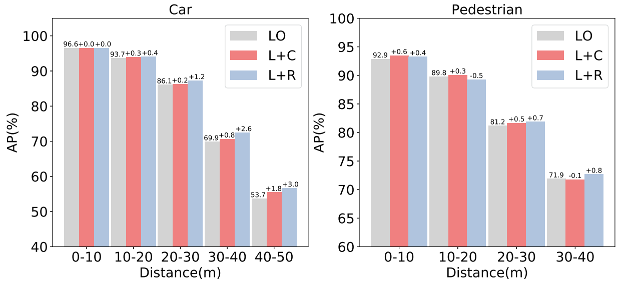

Fig. 2. AP results with respect to different distance ranges on nuScenes validation set for *LO*, *L+R*, *L+C*. Radar is more effective for longer distances. Note that pedestrians participate in evaluation within the range of 40 m on nuScenes.

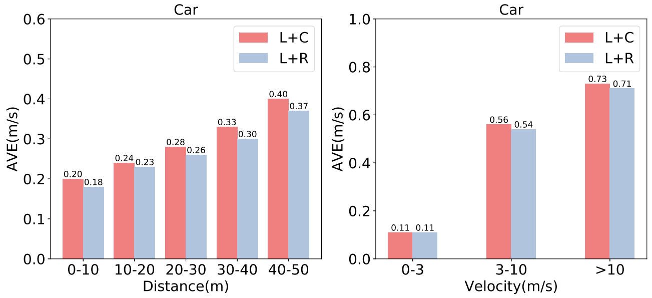

Fig. 3. AVE results of *L+C* and *L+R* in different distance and velocity ranges on nuScenes validation set.

mAP: (average precision of multi-class) The *L+C* fusion, or *L+R* fusion, can both outperform the *LO* baseline. Between these two, *L+C* fusion achieves better mAP results thanks to the richer semantic information in camera images compared with sparse Radar points. The result of fusing all three data sources (*L+RC*) yields the highest mAP result, demonstrating Radar and cameras each have complementary advantages. Specifically, Radar has a longer detection range but lacks of semantic information, while the camera data contains richer semantic information without depth measurement.

To further verify this, we show the AP results at different distance ranges for *LO*, *L+C*, and *L+R* in Fig. 2. Among all the classes, nuScenes has the most cars and pedestrians, which are thus used here. We observe that in relatively short distance ranges ($< 20\,m$), the two fusion methods only improve the AP marginally for cars; Radar may even hurt the AP for pedestrians. When the distance becomes larger ($> 20\,m$), Radar's improvement on AP becomes more pronounced and *L+R* fusion delivers the best AP value among the three, for both cars and pedestrians. This clearly demonstrates the effectiveness of Radar at long distances.

mAVE: (average absolute velocity error of multi-class) It is well understood that Radar can provide effective velocity estimations by the Doppler effect. Unsurprisingly, the *L+R* fusion can reduce mAVE(m/s) more effectively than *L+C* fusion. Furthermore, to compare the abilities of velocity estimations of those modalities that contain dynamic measurement information with those that do not, we choose two fusion approaches *L+R* and *L+C* (cameras provide static information from color measurements) in different distance and velocity ranges and show the results in Fig. 3. From the left figure, we observe that in all distance ranges, *L+R* gives lower AVE than *L+C*. From the right figure, we see that for slow cars($< 3\,m/s$), the AVE of *L+C* is comparable with *L+R*. For faster-moving cars ($> 3\,m/s$), *L+R* is clearly better. Notably, though Radar provides radial velocity measurements, not tangential velocity, the final velocity estimation can still be enhanced by a suitable network transformation.

mATE: (average translation error of multi-class) We find that the location estimation with *L+C* fusion is better than *L+R* because Radar has considerable positional noises, which inherently causes the object's Radar points to appear outside of the actual bounding box. We show several examples in Fig. 4.

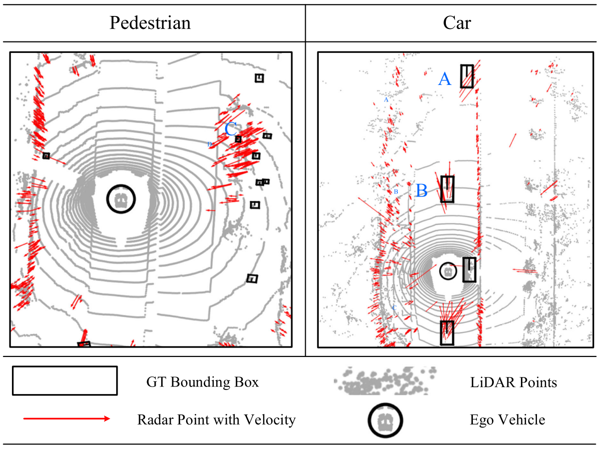

Fig. 4. Two example scenes in bird's eye view of cars and pedestrians on nuScenes dataset with both LiDAR and Radar points. For cars A and B (right figure), the radar points (red arrow tails) of objects appear outside of the actual bounding box. Pedestrians have very few Radar points because of small sizes. The pedestrian C (left figure) suffers from severe noise interference from multiple nearby radar points.

To summarize, LiDAR provides accurate 3D position measurement, but lacks of velocity information. Radar provides dynamic velocity measurements, has a longer detection range, yet suffers from position noises. Camera images contain rich semantic information for classification, especially for small objects like pedestrians which have sparse LiDAR and Radar point clouds. Combining these complementary sensors, we can achieve a stronger detection baseline.

D. *Comparison of Fusion Inputs for 3D Tracking*

Next, we compare the same set of point-wise fusion input configurations for their tracking performance: 1) *L+C* fusion, 2) *L+R* fusion, and 3) *L+RC* fusion. Here, we implement two common position based affinity and re-ID based affinity in tracking. The tracking pipelines are shown in Fig. 5. The re-ID head is embedded in the detector following common practice [25], [26].

In this set of experiments, we again take the CenterPoint [21] as the LiDAR-only baseline *LO* detector. For the re-ID based tracking, we train the re-ID head independently of the detector by softmax loss as in [26]. In the association phase, we adopt $cosine$ similarities between re-ID features or position offsets

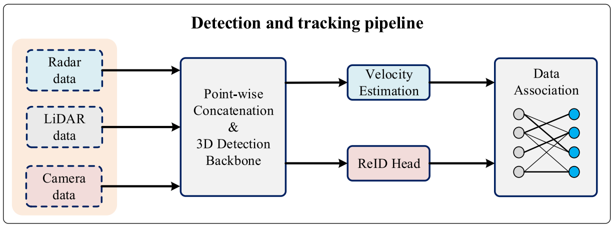
Detection and tracking pipeline
Radar data
LiDAR data
Camera data
Point-wise Concatenation & 3D Detection Backbone
Velocity Estimation
ReID Head
Data Association

Fig. 5. Detection and tracking pipelines with different fusion inputs: *L+C*, *L+R*, and *L+RC*. Point-wise fusion is adopted in all cases.

TABLE III
DETECTION AND TRACKING RESULTS ON NUSCENES VALIDATION SET WHEN ONLY USING POSITION AFFINITY IN DATA ASSOCIATION

Input Modality	Detection Metrics	Tracking Metrics			
	mAP(%) ↑	AMOTA(%) ↑	FP↓	FN↓	IDS↓
LO	61.27	64.50	12795	21247	601
L+C	64.78	68.30	**11790**	21278	**566**
L+R	63.21	67.70	13637	19258	572
L+RC	**65.50**	**69.20**	12825	**18985**	581

TABLE IV
DETECTION AND TRACKING RESULTS ON NUSCENES VALIDATION SET WHEN ONLY USING RE-ID FEATURE AFFINITY IN DATA ASSOCIATION

Input Modality	Detection Metrics	Tracking Metrics			
	mAP(%) ↑	AMOTA(%) ↑	FP↓	FN↓	IDS↓
LO	61.27	48.40	12150	32602	5429
L+C	64.88	55.80	10812	30392	4964
L+R	62.06	50.20	**10509**	31660	8784
*L+R**	61.74	49.70	11397	31530	8863
L+RC	**65.50**	**59.40**	11503	**27307**	**4055**

*R** represents radar data sans radar cross-section (RCS).

as the affinity matrix in greedy matching. In order to carefully quantify the impact of fusion inputs on each branch, we perform the position based tracking and re-ID based tracking separately and summarize the results in Tables III and IV respectively.

Position Offset: Table III shows the position based tracking results for different fusion inputs. Here, the tracking results are largely dependent on the detection results (i.e., AMOTA increases with mAP), and we thus have the same observations as in detection: 1) *L+C* fares better than *L+R* as images contain much more classification information than Radar points as AMOTA is related to classification scores more closely; 2) *L+R* improves the tracking performance thanks to more true positives at far distances, as shown by a lower FN; 3) fusing the three modalities *L+RC* together can combine the advantages of the three, which gives the best AMOTA results.

Re-ID: Table IV shows the re-ID based tracking results for different fusion inputs. Since LiDAR provides better position estimations than re-ID features (due to lack of information such as color, texture, velocity, etc), we expect greater benefits from multi-modal fusion in re-ID based tracking than in position-based tracking. We summarize our observations:

- Benefit of LiDAR-camera fusion. Compared to the *LO* baseline, *L+C* can increase the AMOTA value by 7.4%. This improvement does not only come from better mAP, but we also notice a lower IDS with *L+C*, suggesting the association becomes more accurate. This is well expected as images contain rich color or texture information which plays an important role in extracting the re-ID feature [25], [26].
- Effect of LiDAR-Radar fusion. *L+R* improves the AMOTA value by 1.8%. This is mainly due to the growth of mAP. The IDS actually becomes worse in *L+R*, because Radar point clouds are sparse and lack appearance details. In addition, we consider a slightly different fusion input, *L+R**, in which we drop the Radar cross-section (RCS) element in the Radar input. The RCS element of an object is related its size, surface material reflectivity, and directionality of the Radar reflection caused by the object's shape and orientation, which we believe have a bearing on both re-ID and detection tasks – e.g., different classes have different materials; same objects across frames have similar sizes, orientations and shapes. Unsurprisingly, *L+R** has worse mAP than *L+R* due to worse classification score prediction with higher FP. The IDS value shows the same trend.
- Fusion of all three modalities. *L+RC* performs the best among all the input configuration, with 11% AMOTA improvement compared to the *LO* baseline.
- Comparison of position based and re-ID based association affinity. Results in Tables III and IV show that the re-ID based association fares poorer than the position based association when fusing the same modalities. This is because the location and velocity estimations can be so accurate with the high precision sensor LiDAR that the positional association becomes dominant.

E. Comparison of Fusion Strategies for 3D Detection and Tracking

After comparing different fusion inputs, we next compare different fusion strategies to answer the question "how to fuse". Specifically, we consider all three input modalities, and with this configuration we evaluate different fusion strategies.

We compare the following fusion strategies when we perform a 3-way LiDAR-Radar-Camera fusion: 1) a point-wise concatenation among LiDAR points, Radar pillar features and camera image features (referred to as the point-wise 3-way fusion). 2) a point-wise concatenation between LiDAR points and camera image features, then followed by a BEV concatenation with Radar pillar features (referred to as the point-BEV-based 3-way fusion). 3) a BEV-based concatenation among LiDAR features, Radar pillar features and camera image features (referred to as the BEV-based 3-way fusion). Specially, we transform the camera features in perspective view to BEV by an extra 3D backbone [15]. The three network architectures are shown in Fig. 6. In all options, we first choose the point-wise projection between LiDAR and camera data because it has been shown effective to *paint* the LiDAR points with corresponding image features in a point-wise fashion as in [7], [14]. The main difference in these three options stems from the fusion stages among the LiDAR, Radar and camera data.

We follow the same training paradigm and use the same architecture for feature encoding in every fusion strategy. Specifically, we encode the Radar data using a pillar feature net [1] and encode

TABLE V
DETECTION AND TRACKING (POSITION OFFSET BASED) RESULTS ON THE NUSCENES VALIDATION SET FOR DIFFERENT FUSION STRATEGIES

input	strategy	Detection Metrics				Tracking Metrics	
		NDS(%) ↑	mAP(%) ↑	mAVE(m/s) ↓	FPS(Hz) ↑	AMOTA(%) ↑	IDS↓
LO	-	68.44	61.27	0.303	**1.28**	64.50	601
L+RC	point-wise	71.81	65.50	0.261	0.82	69.20	581
L+RC	point-BEV-based	**72.31**	**65.85**	**0.241**	0.77	68.90	547
L+RC	BEV-based	71.90	65.24	0.249	0.62	**70.40**	**541**

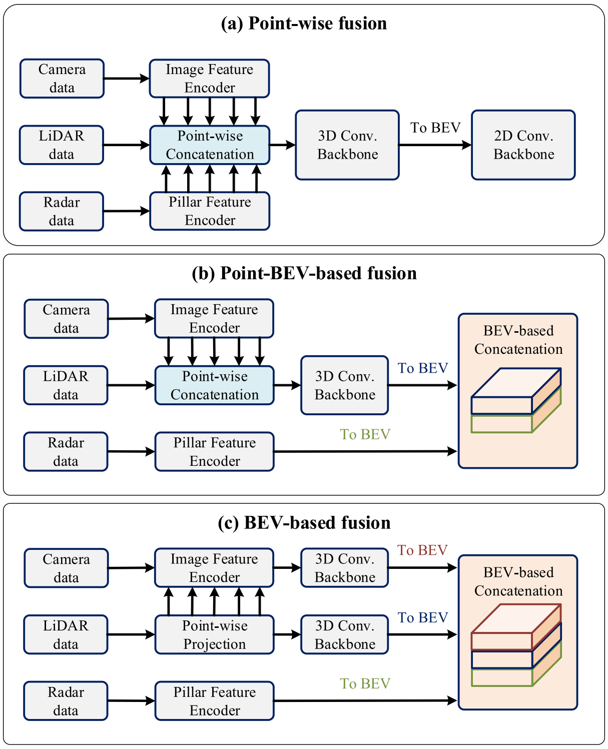

Fig. 6. Pipelines for different fusion strategies: (a) point-wise *L+RC* 3-way fusion, (b) point-BEV-based *L+RC* 3-way fusion, and (c) BEV-based *L+RC* 3-way fusion.

the image data with DLA-34 [24]. The affinity matrices used in tracking are all based on the position offset calculation manners.

Compared with other strategies, the point-BEV-based fusion between the painted LiDAR points and the Radar points gives best detection results (in terms of NDS, mAP, and mAVE). Its AMOTA is slightly worse (by 0.31%) than point-wise fusion. We observe that the performance is not better even though the Radar features are processed by the 3D backbone additionally in the point-wise 3-way fusion. A likely explanation is that the Radar data does not benefit much from the 3D convolution network due to the inaccurate height measurement.

Surprisingly, the BEV-based 3-way fusion only outperforms the others in tracking with best AMOTA and IDS, but not in detection. Its detection performance is similar to the other approaches though the image data are processed by an extra 3D backbone with lowest FPS. Due to sparsity, LiDAR points projected on the image plane can't cover the dense image features fully, causing the loss of image information. That's the main weakness of the point-wise projection, resulting in a performance bottleneck. More work can be done to map the image features to BEV efficiently in the future.

In summary, among the competitors, the point-BEV-based fusion strategy makes the best trade-off between accuracy and computation costs.

IV. EZFusion: A BARE-MINIMUM 3D DETECTION/TRACKING PIPELINE THAT INTEGRATES LIDAR, RADARS AND CAMERAS

Based on the observations in Section III, we put together a pared-down 3D object detection and tracking framework EZFusion, shown in Fig. 7, which combines complementary LiDAR, Radar and camera data simultaneously. EZFusion is similar to the pipeline for point-BEV-based LiDAR-Radar-camera 3-way fusion discussed in Section III-E (the figure in Fig. 6(b)). Below we explain its main modules in more details:

Point-Wise LiDAR-Camera Fusion: We use pre-trained DLA-34 [24] as the image feature extractor. For each LiDAR point (x, y, z), we perform a homogeneous transformation followed by a projection to obtain the corresponding 2D camera coordinates I_{uv} as well as the image feature f_{img}. Then we append the feature with the original LiDAR point to obtain the painted point (x, y, z, r, t, f_{img}), where r is the reflectance ratio, and t is the relative timestamp (i.e., the time difference between the current sweep and other sweeps so that the network can utilize the time feature instead of real timestamps) of multiple LiDAR sweeps. Next we follow the voxel-based pipeline [2] to process the painted LiDAR points. With a flattening operation, we can obtain the BEV feature map $M_{LC} \in \mathbb{R}^{H\times W\times C}$.

Radar Feature Extraction and BEV Fusion: We merge multiple frames of Radar point clouds to compensate for the sparsity and positional measurement uncertainty of Radar. We denote each Radar point in the merged Radar point clouds as $(x, y, z, t, rcs, vx_{comp}, vy_{comp})$, where t is the relative timestamp (similar time delta in LiDAR above) of multiple Radar frames, rcs is Radar cross-section, vx_{comp}, vy_{comp} are the radial velocities in m/s compensated by the ego motion. Due to ego-motion, we need to transform the position and velocity of Radar points from other frames to the reference frame before merging. Due to the lack of accurate z information, we encode Radar points by Pillar Feature Net [1] to generate the BEV pseudo image $M_R \in \mathbb{R}^{H\times W\times C'}$. Finally, we concatenate M_{LC} and M_R in the BEV perspective followed by a 2D RPN network in CenterPoint to generate the fused feature map $M_{LRC} \in \mathbb{R}^{H\times W\times C''}$.

Tracking with Joint Position and Re-ID Association: We append a re-ID head parallel to the regression head in the detection task. From the fused feature map M_{LRC} of LiDAR, Radar and camera data, we can extract more discriminative re-ID features and estimate more accurate velocities for tracking.

Similar to the regression head network in [21], our re-ID head also consists of a deformable convolution and 3×3

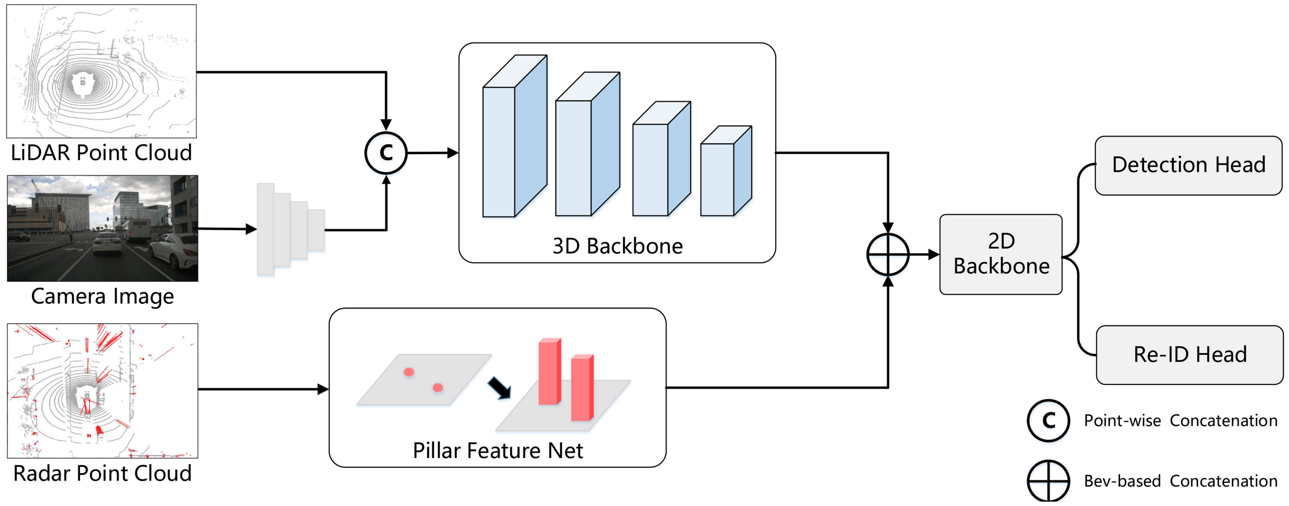

Fig. 7. *Overview of the* EZFusion *pipeline:* Our model encodes Radar points with the Pillar Feature Net. The LiDAR points and image features are fused in a point-wise manner, then followed by a BEV concatenation with Radar pillar features. The fused features are fed into multi branches for detection and re-ID tasks.

convolution with 128 kernels. Having a similar structure as the regression head and sharing the same range BEV fused feature map M_{LRC}, the re-ID feature is easy to align with regression features in detection. Like fairMOT [26], we treat the training process as a classification task. In the training process, we map the instance-aware appearance feature embedding from re-ID head to the class distribution vector s_j, and get the one-hot vector y_j according to the instance ID label from the ground truth. In our setup, to make ID labels of instances in the same frame not adjacent, we also map the ID labels to new ID labels by a non-adjacent list of integers. The mapping is one-to-one such that the instances of class distribution vector can be more discriminating in the training process. We compute the softmax loss as follows:

$$L_{soft} = -\sum_{j=1}^{J} y_j \log s_j, \tag{1}$$

where J is the number of identity classes.

Finally, we compute the affinity matrix by re-ID feature $cosine$ similarities and position offsets compensated by velocities between objects in adjacent frames. Then we associate these objects by a greedy matching algorithm.

V. Implementation and Results

Implementation: We implement our EZFusion on PyTorch [27] using the open-sourced MMDetection3D [28] framework. We test it on the nuScenes dataset [13]. We follow the nuScenes original split: 700 scenes for training and 150 scenes for validation. In our EZFusion, the resolution of image feature is 400×225, and the size of Radar pillar feature map is $180 \times 180 \times 32$. We adopt CenterPoint [21] as 3D detector, where the size of LiDAR voxels is $0.075\,\text{m} \times 0.075\,\text{m} \times 0.2\,\text{m}$, and the size of Radar pillars is $0.6\,\text{m} \times 0.6\,\text{m} \times 8.0\,\text{m}$.

In the training process, we apply GT-sampling, random flipping, global rotation and scaling for data augmentation. We simultaneously perform the same global augmentation operations on LiDAR points and Radar points. When projecting LiDAR points on image feature maps, we transform LiDAR points to their original states by reverse transformation like Moca [29]. Furthermore, in GT-sampling, we cut the image from mask (RLE encoding form) generated by an instance segmentation model HTC [30], then we paste the instance mask image on raw images in the order of the object's depth. We train our detection model on GeForce RTX 3090 GPU with 20 epochs at a learning rate of 0.001. The re-ID branch is trained independent of the detection part with another 10 epochs. The reason is that the classification and re-ID tasks have inconsistent training goals. We also follow the CBGS [31] to adopt the class-balanced grouping strategy.

TABLE VI
Detection and Tracking Results on NuScenes Validation Set for EZFusion

Input Modality	Tracking Association	Det. Metrics	Tracking Metrics		
		mAP(%) ↑	AMOTA(%) ↑	AMOTP(m) ↓	IDS↓
LO	position	61.3	64.5	**0.584**	601
L+RC	position	**66.0**	71.2	0.590	569
L+RC	position+re-ID	**66.0**	**71.3**	0.595	**557**

Results and Discussion: We summarize the detection and tracking results of our EZFusion on nuScenes validation set in Table VI. Compared to the baseline with LiDAR-only input, our EZFusion has an improvement of 4.7% in mAP, and 6.7% in AMOTA. Our performance is comparable to the top-ranked methods on the nuScenes 3D tracking leaderboard. For example, our EZFusion performs better than EagerMOT [32] that performs an roi-level late fusion in detection and a two-stage data association procedure in tracking, which can only use object-level results from bulky 2D and 3D detectors. In particular, we put together a minimal and practical fusion framework, that by itself competes nicely with the best fusion-based detection/tracking systems on nuScenes that include fancy designs. When using such a framework in practice, one can simply replace network blocks to those that fit their needs, while still maintaining the performance.

VI. Conclusion

In this letter, we set out to look for an efficient and robust 3D detection and tracking framework that can combine the advantages of LiDAR, Radar and camera images. Such a framework should be simple to put together, easy to extend, and able to deliver state of the art performance. Towards this goal, we first investigate the advantages and disadvantages of each data modality, and examine how to cleverly combine the three modalities for the maximal effect. Through extensive experiments, we find that it is the most effective to first paint the LiDAR points with corresponding image features, and then to fuse the painted LiDAR points with Radar points on the BEV plane. Following this guideline, we put together EZFusion, which, with the most basic network modules, can perform comparably with state-of-the-art fusion methods.

References

[1] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12697–12705.

[2] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4490–4499.

[3] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, "Voxel R-CNN: Towards high performance voxel-based 3D object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 1201–1209.

[4] J. Deng, Y. Pan, T. Yao, W. Zhou, H. Li, and T. Mei, "Single shot video object detector," *IEEE Trans. Multimedia*, vol. 23, pp. 846–858, Apr. 2020.

[5] J. Deng, W. Zhou, Y. Zhang, and H. Li, "From multi-view to hollow-3D: Hallucinated hollow-3D R-CNN for 3D object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 12, pp. 4722–4734, Dec. 2021.

[6] X. Weng, J. Wang, D. Held, and K. Kitani, "3D multi-object tracking: A baseline and new evaluation metrics," in *Proc. Int. Conf. Intell. Robots Syst.*, 2020, pp. 10359–10366.

[7] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "PointPainting: Sequential fusion for 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 4604–4612.

[8] 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 *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 720–736.

[9] A. Shenoi et al., "JRMOT: A real-time 3D multi-object tracker and a new large-scale dataset," in *Proc. Int. Conf. Intell. Robots Syst.*, 2020, pp. 10335–10342.

[10] Y. Zeng, C. Ma, M. Zhu, Z. Fan, and X. Yang, "Cross-modal 3D object detection and tracking for auto-driving," in *Proc. Int. Conf. Intell. Robots Syst.*, 2021, pp. 3850–3857.

[11] R. Nabati and H. Qi, "CenterFusion: Center-based radar and camera fusion for 3D object detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2021, pp. 1527–1536.

[12] R. Nabati, L. Harris, and H. Qi, "CFTrack: Center-based radar and camera fusion for 3D multi-object tracking," in *Proc. IEEE Intell. Veh. Symp.*, 2021, pp. 243–248.

[13] H. Caesar et al., "nuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11621–11631.

[14] T. Huang, Z. Liu, X. Chen, and X. Bai, "EPNet: Enhancing point features with image semantics for 3D object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 35–52.

[15] C. Wang, C. Ma, M. Zhu, and X. Yang, "PointAugmenting: Cross-modal augmentation for 3D object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11794–11803.

[16] B. Yang, R. Guo, M. Liang, S. Casas, and R. Urtasun, "RadarNet: Exploiting radar for robust perception of dynamic objects," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 496–512.

[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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 444–453.

[18] X. Weng, Y. Wang, Y. Man, and K. M. Kitani, "GNN3DMOT: Graph neural network for 3D multi-object tracking with 2D-3D multi-feature learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6499–6508.

[19] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[20] P. Sun et al., "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2446–2454.

[21] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3D object detection and tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 11784–11793.

[22] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The clear mot metrics," *EURASIP J. Image Video Process.*, vol. 2008, pp. 1–10, 2008.

[23] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 17–35.

[24] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2403–2412.

[25] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 107–122.

[26] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the fairness of detection and re-identification in multiple object tracking," *Int. J. Comput. Vis.*, vol. 129, no. 11, pp. 3069–3087, 2021.

[27] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, Art. no. 721.

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

[29] W. Zhang, Z. Wang, and C. Change Loy, "Multi-modality cut and paste for 3D object detection," 2020.

[30] K. Chen et al., "Hybrid task cascade for instance segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4974–4983.

[31] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-balanced grouping and sampling for point cloud 3D object detection," 2019, *arXiv:1908.09492*.

[32] A. Kim, A. Ošep, and L. Leal-Taixé, "EagerMOT: 3D multi-object tracking via sensor fusion," in *Proc. IEEE Int. Conf. Robot. Automat*, 2021, pp. 11315–11321.