Exploiting Temporal Relations on Radar Perception for Autonomous Driving

Li, Peizhao; Pu, Wang; Berntorp, Karl; Liu, Hongwu

TR2022-069 July 20, 2022

Abstract

We consider the object recognition problem in autonomous driving using automotive radar sensors. Comparing to Lidar sensors, radar is cost-effective and robust in all-weather conditions for perception in autonomous driving. However, radar signals suffer from low angular resolution and precision in recognizing surrounding objects. To enhance the capacity of automotive radar, in this work, we exploit the temporal information from successive ego-centric bird-eye-view radar image frames for radar object recognition. We leverage the consistency of an object's existence and attributes (size, orientation, etc.), and propose a temporal relational layer to explicitly model the relations between objects within successive radar images. In both object detection and multiple object tracking, we show the superiority of our method compared to several baseline approaches

IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2022

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Exploiting Temporal Relations on Radar Perception for Autonomous Driving

Peizhao Li[1]*, Pu Wang[2], Karl Berntorp[2], Hongfu Liu[1]
[1]Brandeis University, [2]Mitsubishi Electric Research Laboratories
{peizhaoli,hongfuliu}@brandeis.edu, {pwang,berntorp}@merl.com

Abstract

We consider the object recognition problem in autonomous driving using automotive radar sensors. Comparing to Lidar sensors, radar is cost-effective and robust in all-weather conditions for perception in autonomous driving. However, radar signals suffer from low angular resolution and precision in recognizing surrounding objects. To enhance the capacity of automotive radar, in this work, we exploit the temporal information from successive ego-centric bird-eye-view radar image frames for radar object recognition. We leverage the consistency of an object's existence and attributes (size, orientation, etc.*), and propose a temporal relational layer to explicitly model the relations between objects within successive radar images. In both object detection and multiple object tracking, we show the superiority of our method compared to several baseline approaches.*

1. Introduction

Autonomous driving utilizes sensing technology for robust dynamic object perception, and sequentially uses the perception for reliable and safe vehicle decision-making [34]. Among various perception sensors, camera and Lidar are the two dominant ones exploited for surrounding object recognition. The camera provides semantically rich visual features of traffic scenarios, while Lidar provides high-resolution point clouds that can capture the reflection from objects. Compared with camera and Lidar, radar enjoys the following unique advantages when applied in automotive applications. Primarily operating at 77 GHz, radar transmits electromagnetic waves at a millimeter wavelength to estimate the range, velocity, and angle of objects. At such a wavelength, it can penetrate or diffract around tiny particles in conditions such as rain, fog, snow, and dust, and offer long-range perception in these adverse weather conditions [35]. In contrast, laser sent by Lidar at a much shorter wavelength may bounce off these tiny particles, which leads to a significantly reduced operating range. Compared with the camera, radar is also resilient to light conditions, *e.g.*, night and sun glare. Furthermore, radar offers a cost-effective and reliable option to complement other sensors. For the cost of Lidar, according to an aggressive estimate by Luminar, is expected to be the range of \$500 - \$1000 [1]. In contrast, automotive radar is expected to be less than \$100 in 2022 [8]. However, as a disadvantage of radar-assisted automotive perception, a high angular resolution in the azimuth and elevation domains are indispensable. In recent open-access automotive radar datasets, an azimuth resolution of 1° becomes available, while the elevation resolution is still lagging behind. With 1° azimuth resolution, semantic features for objects in a short range, *e.g.*, corners and shapes, can be observed, while an object at far distances can still be blurred due to the cross-range resolution. In summary, the capability of localizing and identifying objects for radar is still falling behind from full-level autonomous driving.

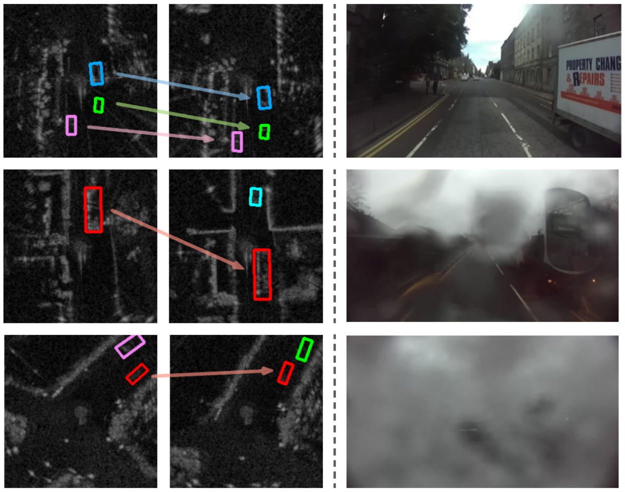


Figure 1. Showcasing of two successive radar images and the corresponding camera recording from *Radiate* dataset [20]. From top to bottom, we display examples in the normal, foggy, and snowy weather. The bounding boxes are the ground-truth annotations of objects where its color implies the object ID. The plotted arrows show the consistency of the object's appearance and attributes within a short time period, *e.g.*, length, width, and orientation.

Some recent efforts have been taken to leverage and enhance automotive radar for object recognition from an algorithmic perspective. [14] proposes a deep-learning approach using range-azimuth-doppler measurement. [16] detects ob-

*Work done during the internship at MERL jects via synchronous radar and Lidar signals. Similarly, [12, 30] exploit the multi-modal sensing fusion. Besides deep learning, Bayesian learning has also attempted to solve extended object tracking with radar point clouds [28, 31]. The above works mainly focus on multi-modal sensing fusion for robust perception [12, 16, 30]. Differently, in this paper, we take our attempt to enhance the perception only using radar information, which requires fewer perception resources and avoids a complicated synchronized process for signals among multi-modal sensors.

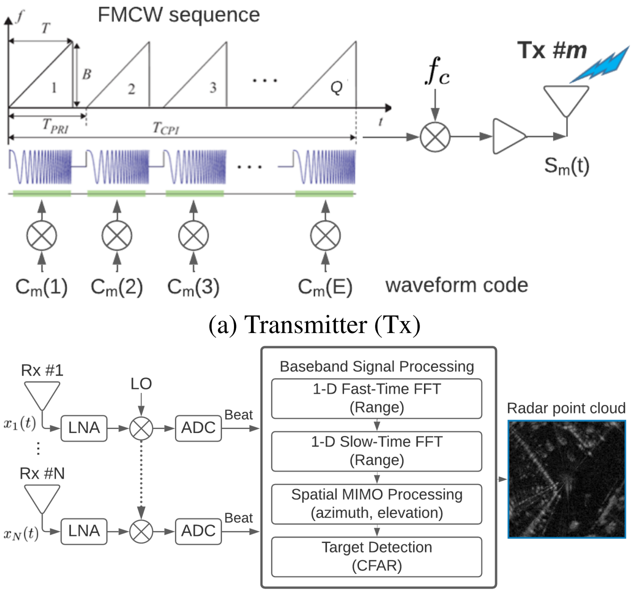
In this paper, we consider ego-centric bird-eye-view radar point clouds presented in a Cartesian frame, where pixel values indicate the strength of reflections. We develop an approach to enhance radar perception using temporal information. Based on the observation in Fig. 1, we assume that the same objects detected by radar within successive frames are consistent and share almost the same attributes, such as the object's existence, length, orientation, *etc*. As a result, the detection at one frame can be facilitated by a previous/future frame through object-level correlations. To compensate for the blurriness and low angular resolution raised by radar sensors, we involve temporality and incorporate customized temporal relational layers to explicitly handle the object-level relations across successive frames. The temporal relational layer takes feature vectors at the potential object's centers and conducts a temporal as well as a self-attention over the object features which are wrapped with their locality. Colloquially, this layer links temporally similar objects and transmits their representations, and is akin to feature smoothing. Hence, temporal relational layers could insert the inductive bias from object temporal consistency. Afterward, the object heatmap (indicating the center of objects) and relevant attributes are inferred upon the updated feature representation from temporal relational layers.

In this work, we consider the object recognition problem using radar in autonomous driving, which is a crucial alternative sensing technology that owes unique advantages. We underline major contributions of our work as follows:

- We facilitate the radar perception with additional temporal information to compensate for the blurriness and low angular resolution raised by radar sensors.
- We design a customized temporal relational layer, where the networks are inserted with an inductive bias that the same object in successive frames should share consistent appearance and attributes.
- We evaluate our method in object detection and multiple object tracking on *Radiate* dataset. With the comprehensive comparison to baseline methods, we show the consistent improvements brought by our method.

2. Radar Perception: Background

Automotive radar dominantly uses frequency modulated continuous waveform (FMCW) to detect objects and generate point clouds over multiple physical domains. As shown in Fig. 2 (a), it transmits a sequence of FMCW pulses through one of its M transmitting antennas:



(a) Transmitter (Tx)

(b) Receiver (Rx)

Figure 2. FMCW-based automotive radar.

$$s_m(t) = \sum_{q=0}^{Q-1} c_m(q) s_p\left(t - nT_{\text{PRI}}\right) e^{j2\pi f_c t}, \qquad (1)$$

where m and q are the indices for transmitting antenna and pulse, T_{PRI} is pulse repetition interval, f_c is the carrier frequency (e.g., 79 GHz), and $s_p(t)$ is baseband FMCW waveform (shown as the sinusoids in Fig. 2 (a)).

An object at a range of R_0 with a radial velocity v_t and a far-field spatial angle (*i.e*. azimuth, elevation, or both) induces amplitude attenuation and phase modulation to the received FMCW signal at each of N receiver RF chains (including the low noise amplifier (LNA), local oscillator (LO), and analog-to-digital converter (ADC)) of Fig. 2 (b). The induced modulation from the target is captured by the baseband signal processing block (including fast Fourier transforms (FFTs) over range, Doppler, and spatial domains) in Fig. 2 (b). All these processes lead to a multi-dimensional spectrum. With the constant false alarm rate (CFAR) detection step that compares the spectrum with an adaptive threshold, radar point clouds are generated in the range, Doppler, azimuth, and elevation domains [3, 10, 25].

Considering the computing and cost constraints, automotive radar manufactures may define the radar point clouds in a subset of the full four dimensions. For instance, traditional automotive radar generates detection points in the range-Doppler domain, whereas some produce the points in the range-Doppler-azimuth plane [17]. In *Radiate* dataset [20] considered in this paper, the radar point cloud is defined in the range-azimuth plane with a 360° field view. The resulting polar-coordinate point cloud is further transformed into an ego-centric Cartesian coordinate system, then a standard voxelization can convert the point cloud into an image.

3. Radar Perception with Temporality

We present our framework in Fig. 3. Corresponding to Fig. 3 from top to bottom, in the subsequent sections, we introduce the temporal feature extraction from two successive frames, the temporal relational layers, the learning method, followed by the extension to multiple object tracking.

Notation We clarify the following notations. θ denotes the learnable parameters in neural networks, and for simplification, we unify the notations of parameters with θ for all modules. We use a bracket following a three-dimensional matrix to represent the feature gathering process at certain coordinates. Consider a feature representation $Z \in \mathbb{R}^{C\times H\times W}$ with C, H, and W represent channel, height, and width, respectively. Let P represent a coordinate (x, y) or a set of two-dimensional coordinates $\{(x, y)\}_K$ with cardinality equal to K and $x, y \in \mathbb{R}$. $Z[P]$ means taking the feature at a coordinate system indicated by P along width and height dimensions, with the returned features in $\mathbb{R}^C$ or $\mathbb{R}^{K\times C}$.

3.1. Temporal Feature Extraction

Denote a single radar frame as $I \in \mathbb{R}^{1\times H\times W}$. We concatenate two successive radar images: a current frame and its previous frame, along the channel dimension to involve temporal information at the input level. The channel-concatenated temporal input image for the current and previous frames can be respectively written as I_{c+p} and $I_{p+c} \in \mathbb{R}^{2\times H\times W}$. The order of 'current' c and 'previous' p in the subscript indicates the feature-concatenating order of these two frames. We obtain the feature representations for the two frames by forwarding the formulated inputs through a backbone neural network $\mathcal{F}_\theta(\cdot)$:

$$Z_c := \mathcal{F}_\theta(I_{c+p}), \;\; Z_p := \mathcal{F}_\theta(I_{p+c}). \tag{2}$$

The backbone network $\mathcal{F}_\theta(\cdot)$ is built in standard deep convolutional neural networks (*e.g*., ResNet), and model parameters are shared for processing two inputs I_{p+c} and I_{c+p}.

To jointly involve high-level semantics and low-level finer details in feature representations, we build skip connections between features at different scales in neural networks. Specifically, for one skip connection, we up-sample the pooled feature from a deep layer to align its size with the feature from previous shallow layers via bilinear interpolation. A list of operations including convolution, non-linear activation, and batch normalization are afterward applied to the up-sampled feature. Next, the up-sampled features are concatenated with those from shallow layers along the channel dimension. Three skip connections are inserted into the networks to drive the features embrace semantics at four different levels. The final feature representation from the backbone neural networks are resulted in $Z_c, Z_p \in \mathbb{R}^{C\times \frac{H}{s}\times\frac{W}{s}}$, where s is the down-sampling ratio over the spatial dimension. We add an illustrative figure in Appendix A.

3.2. Modeling Object Temporal Relations

We design a temporal relational layer to model the correlation and consistency between potential objects in successive frames. The temporal relational layer receives multiple feature vectors from the two frames with each vector representing a potential object in a radar image. We apply a filtering module $\mathcal{G}_\theta^{\text{pre-hm}} : \mathbb{R}^{C\times \frac{H}{s}\times\frac{W}{s}} \rightarrow \mathbb{R}^{1\times \frac{H}{s}\times\frac{W}{s}}$ on features Z_c and Z_p to select top K potential object features for the relational modeling. The set of coordinates P_c for potential objects in Z_c is obtained via the following equation:

$$P_c := \{(x, y) \mid \mathcal{G}_\theta^{\text{pre-hm}}(Z_c)_{xy} \geq [\mathcal{G}_\theta^{\text{pre-hm}}(Z_c)]_K\}, \tag{3}$$

where $[\mathcal{G}_\theta^{\text{pre-hm}}(Z_c)]_K$ is the K-th largest value in $\mathcal{G}_\theta^{\text{pre-hm}}(Z_c)$ over the spatial space $\frac{H}{s}\times\frac{W}{s}$, and the subscript xy denotes taking value at coordinate (x, y). Clearly, the cardinality of P_c is $|P_c| = K$. By substituting Z_p into Eq. (3), P_p for Z_p can be obtained similarly. We do not include features from all coordinates into the temporal relational layer due to that the computational complexity of the subsequent attention mechanism grows quadratically towards the value K.

By taking the coordinate sets P_c and P_p into feature representations, we have the selective feature matrix as:

$$\mathbf{H}_c := Z_c[P_c], \; \mathbf{H}_p := Z_p[P_p]. \tag{4}$$

Sequentially, let $\mathbf{H}_{c+p} := \left[\mathbf{H}_c, \mathbf{H}_p\right]^\top \in \mathbb{R}^{2K\times C}$ denote the matrix concatenation of top-K selected features in the two frames that forms the input to the temporal relational layer.

We supplement the positional encoding into feature vectors before passing $\mathbf{H}_{c+p}$ into the temporal relational layer. The reason is that Convolutional neural networks do not encompass absolute positional information into output feature representation since CNNs enjoy the translational invariance property. However, the position is crucial in object temporal relations because objects at a certain spatial distance in two successive frames are more likely to be associated and would share similar object's attributes. The spatial distance between the same object is conditional on the frame rate and vehicle's motion, and can be learned through a data-driven approach. Denote $\mathbf{H}_{c+p}^{\text{pos}} \in \mathbb{R}^{2K\times(C+D_{\text{pos}})}$ as the feature supplemented by the positional encoding via feature concatenation, where D_{pos} is the dimension of positional encoding. Positional encoding is projected from the normalized 2D coordinate (x, y) that takes values in $[0, 1]$ via linear mappings.

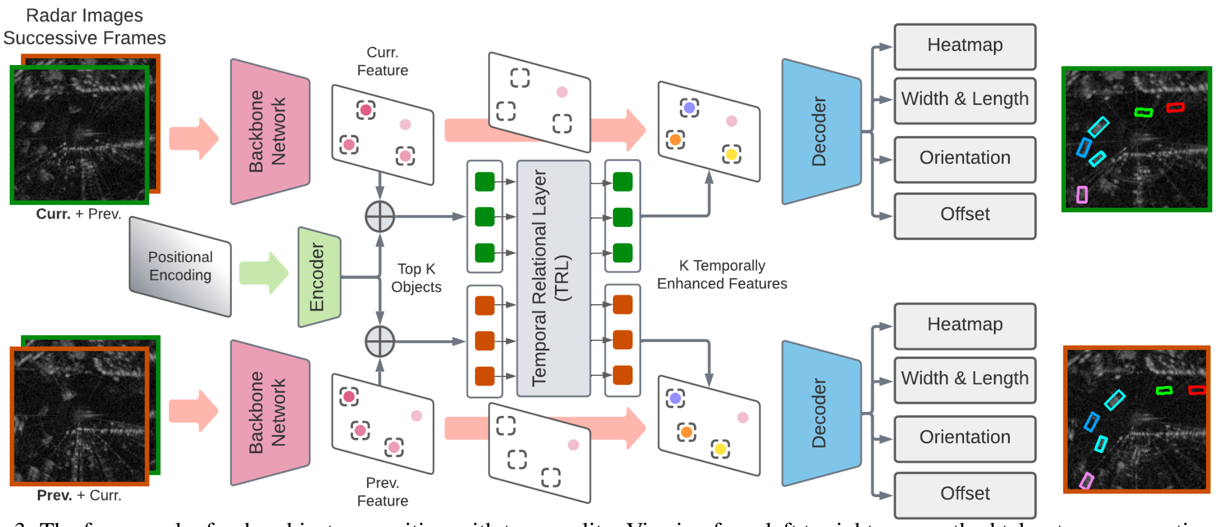


Figure 3. The framework of radar object recognition with temporality. Viewing from left to right, our method takes two consecutive radar frames and extracts the temporal feature from each frame. Then, we select features that could be potential objects and learn the temporal consistency between them. Finally, several regression objectives are conducted upon the updated features for training.

Having the formulations above, we have our main operation for modeling the relations across frames. For a single l-th temporal relational layer, we use a superscript l to denote the input feature and $l+1$ to denote the output feature:

$$\mathbf{H}^{l+1}_{c+p} = \text{softmax}\left(\frac{\mathbf{M} + q(\mathbf{H}^{l,\text{pos}}_{c+p})k(\mathbf{H}^{l,\text{pos}}_{c+p})^\top}{\sqrt{d}}\right) v(\mathbf{H}^{l}_{c+p}), \tag{5}$$

where $q(\cdot)$, $k(\cdot)$, and $v(\cdot)$ are linear transformation layers applied to features and are referred as, respectively, query, keys, and values. d is the dimension of query and keys and is used to scale the dot product between them. The masking matrix $\mathbf{M} \in \mathbb{R}^{2K\times 2K}$ is defined as:

$$\mathbf{M} := \sigma \cdot \left(\begin{bmatrix} \mathbf{1}_{K,K}, \mathbf{0}_{K,K} \\ \mathbf{0}_{K,K}, \mathbf{1}_{K,K} \end{bmatrix} - \mathbb{1}_{2K} \right), \tag{6}$$

where $\mathbf{1}_{K,K}$ is the all-one matrix with size $K \times K$, $\mathbf{0}_{K,K}$ is the all-zero matrix with size $K \times K$, $\mathbb{1}_{2K}$ is the identity matrix of size $2K$, and σ is a negative constant which is set to $-(1\text{e}{+}10)$ in our implementation to guarantee a near-zero value in the output through softmax. The diagonal matrices of $\mathbf{1}_{K,K}$ disable the attention between features from the same frame, while the off-diagonal matrices of $\mathbf{0}_{K,K}$ allow the cross-frame attention. Also, the identity matrix $\mathbb{1}_{2K}$ unlocks the object self-attention. The logic behind self-attention is that the same object co-occurrence cannot always be guaranteed in successive frames since an object can move out of the scope, thereby self-attention is desirable when an object is missing in only one frame. Noticeably, the positional encoding is only attached to keys and query but not to values, so the output feature does not involve locality. Other technical details follows the design of Transformer [24], and here we omit the detailed descriptions for simplification.

After executing the object temporal attention across frames in Eq. (5), we sequentially apply a feed-forward function that consists of two linear layers, layer normalization, and shortcut on features. The relational modeling is built with multiple temporal relational layers with the identical design. At the end, we split the updated features $\mathbf{H}^{l+1}_{c}$ and $\mathbf{H}^{l+1}_{p}$ from $\mathbf{H}^{l+1}_{c+p}$ and refill the feature vector to Z_c and Z_p in the corresponding spatial coordinates from P_c and P_p. Regressions in the next subsection are conducted on top of the refilled feature representations.

Discussion The above feature operations share some similarities with Transformer [24]. Transformer is designed for language representation learning, intending to map the words into a similar latent representation if two words are sharing correlations among the training corpus, including the co-existence, word positions, and semantics. The multi-head attention operations in the stacked architecture can be understood as smoothing over the feature of semantically similar words [4, 6, 11]. In our context, the feature of objects with an identical ID in successive frames should be correlated and share a similar latent representation. This is particularly crucial since the latent representation store all object-relevant attributes and will be used for the subsequent decoding purpose, as elaborated in Section 3.3. The smoothing over two feature vectors of the same object in successive frames satisfies our basic temporal consistency assumption, and can enhance the detection when the object information is partially lost in one frame due to the blurriness from radar.

3.3. Learning

We pick the object's center coordinates from the heatmap, and learn its attributes (*i.e.* the width, length, orientation, and center coordinate offset) from feature representations through regression.

Heatmap To localize objects, the 2D coordinate of a peak value in the heatmap is considered as the center of an object. The heatmap is obtained by a module $\mathcal{G}_\theta^{\text{hm}} : \mathbb{R}^{C\times\frac{H}{s}\times\frac{W}{s}} \to \mathbb{R}^{1\times\frac{H}{s}\times\frac{W}{s}}$ followed by a sigmoid function. We generate the ground-truth heatmap by placing the 2D radial basis function (RBF) kernel on the center of every ground-truth object, while the parameter σ in the RBF kernel is set proportional to the object's width and length. Considering the sparsity of objects in radar images, we use focal loss [13] to balance the regression of ground-truth centers and background, and drive the predicted heatmap to approximate the ground-truth heatmap. Let h_i and $\hat{h}_i$ denote the ground-truth and predicted value at i-th coordinate, N the total number of values in the heatmap, we express the focal loss as:

$$\begin{aligned} L_h := -\frac{1}{N}\sum_i \big(&\mathbb{1}_{h_i=1}(1-\hat{h}_i)^\alpha \log(\hat{h}_i) \\ &+ \mathbb{1}_{h_i\neq 1}(1-h_i)^\beta \hat{h}_i^\alpha \log(1-\hat{h}_i)\big), \end{aligned} \tag{7}$$

where α and β are hyper-parameters and are chosen empirically with 2 and 4, respectively, following the prior work [32]. The same loss function is conducted for $\mathcal{G}_\theta^{\text{pre-hm}}$ to rectify the feature selection of the relational modeling. During inference, a threshold is set on the heatmap to distinguish the object center from backgrounds. Non-maximum suppression is applied to avoid excessive bounding boxes.

Width & Length We predict the width and length of an oriented bounding box from the feature vector positioned at the center coordinate in the feature map through another regression head $\mathcal{G}_\theta^{\text{b}} : \mathbb{R}^C \to \mathbb{R}^2$. Let P_{gt}^k denote the coordinate (x, y) of the center of k-th ground-truth object, b^k the ground-truth vector containing width and length of k-th object, and Z a unified notation for Z_c and Z_p. We have:

$$L_{\text{b}} := \frac{1}{N}\sum_{k=1}^{N} \text{Smooth}_{L_1}\left(\|\mathcal{G}_\theta^{\text{b}}(Z[P_{\text{gt}}^k]) - b^k\|\right), \tag{8}$$

where the L_1 smooth loss is defined as:

$$\text{Smooth}_{L_1}(x) := \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise.} \end{cases} \tag{9}$$

Orientation All vehicles are presented with an orientation in the bird-eye-view image. An angle range in $[0°, 360°)$ can be measured by the deviation between the object's orientation and the boresight direction of the ego vehicle. We regress the sine and cosine values of the angle ϑ via $\mathcal{G}_\theta^{\text{r}} : \mathbb{R}^C \to \mathbb{R}^2$:

$$L_{\text{r}} := \frac{1}{N}\sum_{k=1}^{N} \text{Smooth}_{L_1}(\|\mathcal{G}_\theta^{\text{r}}(Z[P_{\text{gt}}^k]) - (\sin(\vartheta), \cos(\vartheta))\|). \tag{10}$$

During the inference stage, the orientation can be predicted by $\sin(\hat{\vartheta})$ and $\cos(\hat{\vartheta}))$ via $\arctan(\sin(\hat{\vartheta})/\cos(\hat{\vartheta}))$.

Offset Down sampling in the backbone networks could incur a center coordinate shift for every object. The center coordinates in the heatmap are integers while the true coordinates are likely to be off the heatmap grids due to the spatial down sampling. To compensate for the shift, we calculate a ground-truth offset for the k-th object as:

$$o^k := \left(\frac{c_x^k}{s} - \left[\frac{c_x^k}{s}\right], \frac{c_y^k}{s} - \left[\frac{c_y^k}{s}\right]\right), \tag{11}$$

where c_x^k and c_y^k is the k-th center coordinate, s is the down sampling ratio, and the bracket $[\cdot]$ is the rounding operation to an integer. Having $\mathcal{G}_\theta^{\text{o}} : \mathbb{R}^C \to \mathbb{R}^2$, the regression for center positional offset can be similarly expressed as:

$$L_{\text{o}} := \frac{1}{N}\sum_{k=1}^{N} \text{Smooth}_{L_1}(\|\mathcal{G}_\theta^{\text{o}}(Z[P_{\text{gt}}^k]) - o^k\|). \tag{12}$$

Training All above regression functions compose the final training objective by a linear combination:

$$\min_\theta \; L := L_h + L_b + L_r + L_o. \tag{13}$$

We omit the balanced factors for each term for simplification.

For each training step, our training procedure calculates the loss L and does the backward for both the current and previous frame simultaneously. Standing at the current frame, objects in the current frame receives information from the past for object recognition. On the other hand, from the previous frame perspective, objects utilize the temporal information from the immediate future frame. Therefore, the optimization can be viewed as a *bi-directional* backward-forward training towards two successive frames. For now, we do not extend the current framework to multiple frames, since an intermediate frame do not have a proper concatenated order of input images for temporal feature extraction (neither from past to future or nor from future to past) and would reduce the training efficiency.

3.4. Extending to Multiple Object Tracking

Our framework can be easily extended to online multiple object tracking by adapting a similar tracking procedure as in [36]. For multiple object tracking, we add a regression head to the center feature vector to predict a 2D moving offset between the center of an object holding the same tracking ID in current and previous frames. We simply use Euclidean distance to accomplish the association in tracking decoding. We defer a detailed illustration and algorithm for Multiple Object Tracking to Appendix B.

4. Experiment

4.1. Experimental Setup

Dataset We use the radar dataset *Radiate* [20] in our experiments for the following reasons: (1) it contains high-

Table 1. Experimental results of object detection on *Radiate* dataset. TRL is the abbreviation of 'temporal relational layer.'

	Split: train good weather			Split: train good and bad weather		
	mAP@0.3	mAP@0.5	mAP@0.7	mAP@0.3	mAP@0.5	mAP@0.7
RetinaNet-OBB-ResNet18	52.50± 1.81	37.83± 1.82	8.46± 0.61	49.44± 1.32	31.57± 1.54	6.97± 1.24
RetinaNet-OBB-ResNet34	50.79± 3.10	35.61± 3.35	7.67± 1.71	48.09± 3.85	31.10± 3.37	6.93± 1.60
RetinaNet-OBB-ResNet34-T.	52.52± 4.68	37.30± 3.35	8.75± 1.50	42.95± 3.46	24.50± 3.72	3.98± 1.55
CenterPoint-OBB-EfficientNetB4	61.15± 1.23	51.43± 1.45	20.31± 1.73	54.97± 2.59	42.37± 2.14	13.15± 0.98
CenterPoint-OBB-ResNet18	58.69± 3.09	49.41± 2.94	19.02± 1.80	55.83± 3.28	44.48± 3.19	14.43± 2.56
CenterPoint-OBB-ResNet34	59.42± 1.92	50.17± 1.91	18.93± 1.46	53.92± 3.44	42.81± 3.04	13.43± 1.92
BBAVectors-ResNet18	59.38± 3.47	50.53± 2.07	19.72± 1.10	56.84± 3.45	45.43± 2.87	15.07± 1.76
BBAVectors-ResNet34	60.88± 1.79	51.26± 1.99	19.86± 1.36	55.87± 2.90	44.61± 2.57	14.67± 1.45
Ours-EfficientNetB4-w/o TRL	60.77± 0.97	50.93± 1.27	20.31± 1.73	54.97± 2.59	42.37± 2.14	13.15± 0.98
Ours-EfficientNetB4-w. TRL	61.59± 1.54	50.98± 1.52	17.91± 1.48	55.28± 2.32	43.05± 2.63	13.48± 2.01
Ours-ResNet18-w/o TRL	57.48± 4.82	47.90± 4.77	16.85± 2.98	55.64± 2.32	44.48± 2.76	15.10± 1.68
Ours-ResNet18-w. TRL	62.79± 2.01	53.11± 1.96	20.57± 1.47	**58.87**± 3.31	**46.42**± 3.24	**15.59**± 2.31
Ours-ResNet34-w/o TRL	60.98± 1.89	49.98± 2.28	18.89± 1.46	57.21± 3.76	45.93± 3.52	15.51± 2.71
Ours-ResNet34-w. TRL	**63.63**± 2.08	**54.00**± 2.16	**21.08**± 1.66	56.18± 4.27	43.98± 3.75	14.35± 2.15

Table 2. Comparison on object detection to [20]. Results of [20] are directly copied from the original paper.

split: train good weather	mAP@0.5
FasterRCNN-ResNet50 [20]	45.31
FasterRCNN-ResNet101 [20]	45.84
Ours-ResNet18-w. TRL	48.02
Ours-ResNet34-w. TRL	**48.66**

resolution radar images; (2) it provides well-annotated oriented bounding boxes with tracking IDs for objects; and (3) it records various real driving scenarios in adverse weather. *Radiate* is consist of video sequences recorded in adverse weather including sun, night, rain, fog, and snow. The driving scenarios vary from the motorway to the urban. The data format radar images generated from point clouds, where pixel values indicate the strength of radar signal reflections. *Radiate* adopts mechanically scanning Navtech CTS350-X radar, providing 360° high-resolution range-azimuth images at 4 Hz. Currently, the radar does not afford doppler or velocity information. The whole dataset has in total 61 sequences and we follow the official 3 splits: train in good weather (31 sequences, 22383 frames, only in good weather, sunny or overcast), train good and bad weather (12 sequences, 9749 frames, both good and bad weather conditions), and test (18 sequences, 11305 frames, all kinds of weather conditions). We separately train models on the former two training sets and evaluate on the test set. Numerical results from both two splits are reported. We also comprehensively review other public radar datasets and discuss why currently they are not feasible for our experiments in Section 5.

Baseline We implement several detectors, which have been well demonstrated in visual object detection for comparison. These detectors include: Faster-RCNN [18], RetinaNet [13], CenterPoint [37], and BBAVectors [32]. The comparison is conducted with different backbone networks [7, 22]. Traditional detectors are not designed for oriented objects. To make them fit the oriented object detection, we manually add an extra dimension on anchors or regression to predict the angle of the object's orientation. We denote the adaptation as 'OBB' (oriented bounding box) by the end of detector's names in Table 1. To highlight the benefit from temporal modeling, we add the temporal input to baselines where 'T.' indicates the input with two successive frames and 'Ours-w/o TRL' is architecturally equivalence to the CenterPoint model with temporal input. For multiple object tracking, we include CenterTrack [36] on oriented objects that use the same tracking heuristics with us for comparison.

Implementation We follow [20] and exclude pedestrians and groups of pedestrians from detection and tracking targets since only very few reflections are observed in these two kinds of objects. We also do not distinguish the object categories like [20] because there is no significant difference between vehicle categories presented by radar signals (*e.g.*, truck and bus). Regarding the computation, operations related to oriented rectangles like the calculation of the overlapping of oriented bounding boxes are conducted in CPU using DOTA benchmark toolkit [27], while the rest part on deep neural networks is running on a single RTX 3090. For all numerical results in Table 1, we apply a center crop with size 256×256 upon input images and exclude the targets outside this scope. This helps us to conduct comprehensive evaluations using our computational resource and numbers are averaged over 10 random seeds. For results in Table 2 and 3, we keep the original resolution with size 1152×1152 to make a fair comparison to the results from [20]. We set the gap of frames between two successive frames to 3 for detection and 1 for tracking, the position dimension D_p to

64, the number of temporal relational layers to 2, the batch size to 64 for cropped images with a gradient accumulation to every 2 steps, the learning rate to 5e-4 and weight decay to 1e-2 for Adam optimizer with five training epochs.

We adopt mean Average Precision (mAP) with Intersection over Union (IoU) at 0.3, 0.5, and 0.7 for the evaluation of oriented object detection. For multiple object tracking, we adopt the series of MOT metrics [15] including MOTA, MOTP, IDSW, Frag., MT and PT, but defer the descriptions to Appendix B due to the page limitation.

4.2. Result and Analysis

Detection We report detection results in Table 1 and 2. Our method consistently achieves better results on both two training splits among different levels of IoU thresholds. Besides, the margin between the performance with or without tempporal relational layers further confirms the contribution from modeling the temporal object consistence in successive frames. Regarding the two training splits, intuitively, adding more weather conditions into training could enhance the robustness of detection and tracking, since the testing set contains various weather. However, for radar, there is no significant difference in the presentation of data among diverse weather. The margin between two training splits mainly comes from the margin of the number of training samples. Regarding the difference in image size, there is a slight performance drop when involving a larger scope for detection. The drop comes from the cross-range resolution, where further objects might suffer from a heavier blurriness.

Tracking We report results on multiple object tracking in Table 3, where our methods achieve better performance comparing to baseline. For the baseline method, CenterTrack also considers the temporal information by adding the heatmap of the previous frame and the previous image into input during the inference stage. They use the ground-truth heatmap for training and the predicted heatmap for inference. This kind of learning can work well for RGB video tracking since the detection is mostly accurate. However, the detection on radar cannot achieve such accuracy so far, and therefore breaking the alignment of the heatmap in training and inference. The tracking performance with or without temporal relational layers highlights the effectiveness of modeling temporal object-level relations.

Visualization We present visualization results in Fig. 4 on both object detection and multiple object tracking, and more visualizations are attached in Appendix C. We observe many predictions hit the annotations with a slight shift. Except the correct predictions, it is noticeable that our model brings some false positive predictions. However, when looking into these false positives, with a high probability, they will be a cluster of reflections inside the box that can be viewed as a ghost object. This may be the main reason for creating these false positives. Meanwhile, our model miss some objects in the outer space. The reflections of missed objects are drowning in the reflections of static surroundings due to the low angular resolution. How to enhance the detection on ghost objects and blurriness would be an interesting problem.

We add an experiment in Appendix D to analyze the best amount of selective features in temporal relational layers. The empirical results guide the heuristic setting of K.

5. Related Work

Radar Perception in Autonomous Driving There is an increasing attention on the adoption of radar in autonomous driving. We review some recent work from both algorithmic and radar resource perspectives. The work [14] proposes a deep-learning approach for automotive radar object detection using range-azimuth-doppler measurement. [16] focus on sensor fusion and propose a method to incorporate synchronous radar and Lidar signals for object detection. [12, 30] also exploit the multi-modal sensing fusion in autonomous driving. Besides deep learning, Bayesian learning has also been used for extended object tracking using radar [28, 31]. Our work only leverages radar signals but enhances the recognition with the temporal consistency on objects, which has not been explored by previous works. We defer a short review of current radar dataset in Appendix E.

Detection with Temporality Consecutive video frames could provide spatial-temporal cues for object recognition. [26] leverage a feature bank that extends the time horizon for spatial-temporal action localization. [21] and [2] insert the object-level association from short or long temporal dependency into Faster-RCNN [18] to capture the spatial-temporal information in object detection. Other techniques such as video pixel flow or 3D convolutions [29, 38, 39] are applied for visually rich video sequences but too heavy and not efficient for radar images. Our work shares the same philosophy that using spatial-temporal object-level correlation along the time horizon. However, all studies mentioned above are focusing on RGB video data but not design for oriented objects. The object's size and scale may not be consistent if an object is approaching or leaving the scope of the camera. Differently, we put our emphasis on radar data in autonomous driving, where the bird-eye-view point cloud-based images provide significant object property comparing to RBG video data. We design an anchor-free one-stage detector with temporality, which is efficient and does not have to tackle the pre-defined anchor parameters. The center-based detector is suitable for the bird-eye-view presentation since there is no object overlap from this view, hence the central feature is fully exposed to represent an object. Moreover, we do not explore the long-range dependency but restrict the consistency in only one successive frame, since vehicles can move out of the scope if the timescale is too long and consequently no more temporal relation is available.

Table 3. Experimental results of multiple object tracking on *Radiate* dataset. TRL is the abbreviation of 'temporal relational layer.'

split: train good weather	MOTA↑	MOTP↑	IDSW↓	Frag.↓	MT↑	PT↑
CenterTrack-ResNet18	0.1301	0.7026	873	920	269	254
CenterTrack-ResNet34	0.1455	0.7005	802	831	282	279
Ours-ResNet-18-w/o TRL	0.3293	0.7135	513	593	151	324
Ours-ResNet-18-w. TRL	0.3359	**0.7349**	**349**	**498**	**145**	330
Ours-ResNet-34-w/o TRL	0.3569	0.7080	557	640	179	362
Ours-ResNet-34-w. TRL	**0.3791**	0.7188	474	527	219	**332**

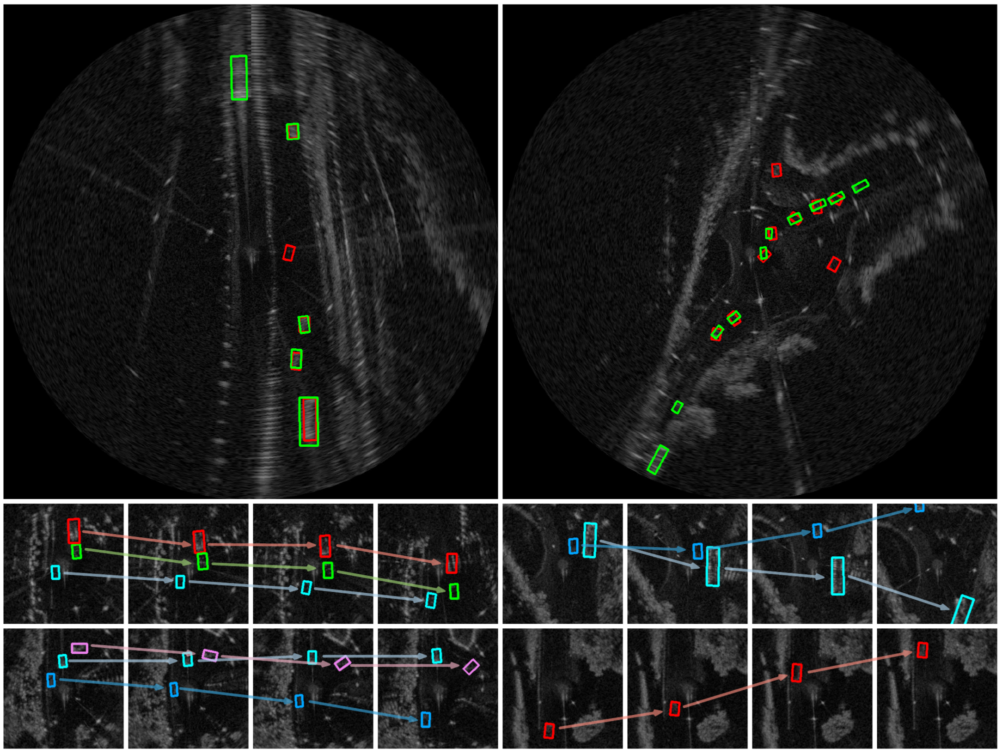


Figure 4. Visualizations on radar perception on *Radiate* dataset. The upper two figures show the object detection while the lower four sets of successive visualizations show multiple object tracking. In detection, green bounding boxes are ground-truth annotations, while red are model predictions. In multiple object tracking, bounding boxes are model predictions, colors indicate the object IDs, and plotted arrows show the moving of objects. Regarding the figure source, the left detection figure is from night-1-4, while the right one is from rain-4-0. From left to right and top to bottom, the tracking sequences are from city-7-0, rain-4-0, fog-6-0, and junction-1-10.

Multiple Object Tracking A well-established paradigm for visual multiple object tracking [15] is tracking-by-detection [9, 19, 23]. The detected object bounding boxes are provided by an external detector, then data association techniques based on object appearance or motion are applied to detection to associate identical objects among candidates in multiple consecutive frames. Recent developments in multiple object tracking convert detectors into tracking algorithms to jointly detect and track objects [5, 33, 36]. We follow the simple tracking rule that is purely based on the cost of euclidean distance [33, 36] to extend our framework to multiple object tracking. Differently, [33, 36] only stack frames at multiple time steps as input, while our networks explicitly consider the object-level consistency.

6. Conclusion

We studied the object recognition problem using radar in autonomous driving. We facilitated the radar perception with temporality from video frames based on the assumption that the same object within successive frames should be consistent and share almost the same attributes. We designed a framework inserted with temporal relational layers to explicitly model the object-level consistency. We showed the effectiveness of our method by experiments in object detection and multiple object tracking.

Acknowledgement The authors would like to thank Petros T. Boufounos, Toshiaki Koike-Akino, Hassan Mansour, and Philip V. Orlik for their helpful discussion.

References

[1] Alan Ohnsman. Luminar Surges On Plan To Supply Laser Sensors For Nvidia's Self-Driving Car Platform, 2021. 1

[2] Sara Beery, Guanhang Wu, Vivek Rathod, Ronny Votel, and Jonathan Huang. Context r-cnn: Long term temporal context for per-camera object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13075–13085, 2020. 7

[3] I. Bilik, O. Longman, S. Villeval, and J. Tabrikian. The rise of radar for autonomous vehicles: Signal processing solutions and future research directions. *IEEE Signal Processing Magazine*, 36(5):20–31, Sep. 2019. 2

[4] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. *arXiv preprint arXiv:2103.03404*, 2021. 4

[5] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3038–3046, 2017. 8

[6] Chengyue Gong, Dilin Wang, Meng Li, Vikas Chandra, and Qiang Liu. Improve vision transformers training by suppressing over-smoothing. *arXiv preprint arXiv:2104.12753*, 2021. 4

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[8] Jessie Lin and Hana Hu. Digitimes Research: 79GHz to replace 24GHz for automotive millimeter-wave radar sensors, 2017. 1

[9] Xiaolong Jiang, Peizhao Li, Yanjing Li, and Xiantong Zhen. Graph neural based end-to-end data association framework for online multiple-object tracking. *arXiv preprint arXiv:1907.05315*, 2019. 8

[10] J. Li and P. Stoica. *MIMO Radar Signal Processing*. John Wiley & Sons, 2008. 2

[11] Peizhao Li, Jiuxiang Gu, Jason Kuen, Vlad I. Morariu, Handong Zhao, Rajiv Jain, Varun Manjunatha, and Hongfu Liu. Selfdoc: Self-supervised document representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5652–5660, June 2021. 4

[12] Teck-Yian Lim, Amin Ansari, Bence Major, Daniel Fontijne, Michael Hamilton, Radhika Gowaikar, and Sundar Subramanian. Radar and camera early fusion for vehicle detection in advanced driver assistance systems. In *Machine Learning for Autonomous Driving Workshop at the 33rd Conference on Neural Information Processing Systems*, 2019. 2, 7

[13] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017. 5, 6

[14] Bence Major, Daniel Fontijne, Amin Ansari, Ravi Teja Sukhavasi, Radhika Gowaikar, Michael Hamilton, Sean Lee, Slawomir Grzechnik, and Sundar Subramanian. Vehicle detection with automotive radar using deep learning on range-azimuth-doppler tensors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019. 1, 7

[15] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016. 7, 8

[16] Kun Qian, Shilin Zhu, Xinyu Zhang, and Li Erran 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*, pages 444–453, 2021. 1, 2, 7

[17] Karthik Ramasubramanian and Brian Ginsburg. AWR1243 sensor: Highly integrated 76–81-GHz radar front-end for emerging ADAS applications. In *Texas Instruments Technical Report*, pages 1–12, 2017. 2

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28:91–99, 2015. 6, 7

[19] Samuel Schulter, Paul Vernaza, Wongun Choi, and Manmohan Chandraker. Deep network flow for multi-object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6951–6960, 2017. 8

[20] Marcel Sheeny, Emanuele De Pellegrin, Saptarshi Mukherjee, Alireza Ahrabian, Sen Wang, and Andrew Wallace. Radiate: A radar dataset for automotive perception. *arXiv preprint arXiv:2010.09076*, 2020. 1, 2, 5, 6

[21] Mykhailo Shvets, Wei Liu, and Alexander C Berg. Leveraging long-range temporal relationships between proposals for video object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9756–9764, 2019. 7

[22] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 6

[23] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3539–3548, 2017. 8

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 4

[25] Pu Wang, Petros Boufounos, Hassan Mansour, and Philip V. Orlik. Slow-time MIMO-FMCW automotive radar detection with imperfect waveform separation. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8634–8638, 2020. 2

[26] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 284–293, 2019. 7

[27] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3974–3983, 2018. 6

[28] Yuxuan Xia, Pu Wang, Karl Berntorp, Lennart Svensson, Karl Granström, Hassan Mansour, Petros Boufounos, and Philip V Orlik. Learning-based extended object tracking using hierarchical truncation measurement model with automotive radar. *IEEE Journal of Selected Topics in Signal Processing*, 15(4):1013–1029, 2021. 2, 7

[29] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision*, pages 305–321, 2018. 7

[30] Bin Yang, Runsheng Guo, Ming Liang, Sergio Casas, and Raquel Urtasun. Radarnet: Exploiting radar for robust perception of dynamic objects. In *European Conference on Computer Vision*, pages 496–512, 2020. 2, 7

[31] Gang Yao, Perry Wang, Karl Berntorp, Hassan Mansour, P Boufounos, and Philip V Orlik. Extended object tracking with automotive radar using b-spline chained ellipses model. In *2021 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8408–8412, 2021. 2, 7

[32] Jingru Yi, Pengxiang Wu, Bo Liu, Qiaoying Huang, Hui Qu, and Dimitris Metaxas. Oriented object detection in aerial images with box boundary-aware vectors. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2150–2159, 2021. 5, 6

[33] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11784–11793, 2021. 8

[34] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469, 2020. 1

[35] Shuqing Zeng and James N. Nickolaou. Automotive radar. In Gregory L. Charvat, editor, *Small and Short-Range Radar Systems*, chapter 9. CRC Press, Inc., 2014. 1

[36] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *European Conference on Computer Vision*, pages 474–490, 2020. 5, 6, 8

[37] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 6

[38] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 408–417, 2017. 7

[39] Xizhou Zhu, Yuwen Xiong, Jifeng Dai, Lu Yuan, and Yichen Wei. Deep feature flow for video recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2349–2358, 2017. 7