SIRA: Scalable Inter-frame Relation and Association for Radar Perception

Yataka, Ryoma; Wang, Pu; Boufounos, Petros T.; Takahashi, Ryuhei

TR2024-041 April 26, 2024

Abstract

Conventional radar feature extraction faces limitations due to low spatial resolution, noise, multipath reflection, the presence of ghost targets, and motion blur. Such limitations can be exacerbated by nonlinear object motion, particularly from an ego-centric viewpoint. It becomes evident that to address these challenges, the key lies in exploiting temporal feature relation over an extended horizon and en- forcing spatial motion consistence for effective association. To this end, this paper proposes SIRA (Scalable Inter-frame Relation and Association) with two designs. First, inspired by Swin Transformer, we introduce extended temporal relation, generalizing the existing temporal relation layer from two consecutive frames to multiple inter-frames with tem- porally regrouped window attention for scalability. Second, we propose motion consistency track with the concept of a pseudo-tracklet generated from observational data for bet- ter trajectory prediction and subsequent object association. Our approach achieves 58.11 mAP@0.5 for oriented object detection and 47.79 MOTA for multiple object tracking on the Radiate dataset, surpassing previous state-of-the-art by a margin of +4.11 mAP@0.5 and +9.94 MOTA, respectively.

IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2024

SIRA: Scalable Inter-frame Relation and Association for Radar Perception

Ryoma Yataka[1,2], Pu Wang[1], Petros Boufounos[1], Ryuhei Takahashi[2]
[1]Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA
[2]Information Technology R&D Center, Mitsubishi Electric Corporation, Kanagawa 247-8501, Japan
{yataka,pwang,petrosb}@merl.com, Takahashi.Ryuhei@ab.MitsubishiElectric.co.jp

Abstract

Conventional radar feature extraction faces limitations due to low spatial resolution, noise, multipath reflection, the presence of ghost targets, and motion blur. Such limitations can be exacerbated by nonlinear object motion, particularly from an ego-centric viewpoint. It becomes evident that to address these challenges, the key lies in exploiting temporal feature relation over an extended horizon and enforcing spatial motion consistence for effective association. To this end, this paper proposes SIRA (Scalable Inter-frame Relation and Association) with two designs. First, inspired by Swin Transformer, we introduce extended temporal relation, generalizing the existing temporal relation layer from two consecutive frames to multiple inter-frames with temporally regrouped window attention for scalability. Second, we propose motion consistency track with the concept of a pseudo-tracklet generated from observational data for better trajectory prediction and subsequent object association. Our approach achieves 58.11 *mAP*@0.5 *for oriented object detection and* 47.79 *MOTA for multiple object tracking on the Radiate dataset, surpassing previous state-of-the-art by a margin of* $+4.11$ *mAP*@0.5 *and* $+9.94$ *MOTA, respectively.*

1. Introduction

Automotive perception involves the interpretation of the external driving environment and internal vehicle cabin conditions with an array of perception sensors to achieve robust safety and driving autonomy [40]. Compared to optical camera and lidar sensors, radar is cost-effective, friendly to sensor maintenance and calibration, and has distinct advantages in providing long-range perception capabilities in adverse weather and lighting conditions [59].

Nevertheless, a notable limitation of radar-based automotive perception is its low spatial resolution in the azimuth and elevation domains, and its inherent noise, including multipath reflection,ghost targets and motion blur. As a result, its ability to detect and track objects lags behind the requirements for fully autonomous driving capabilities. Recently, standalone radar-only perception has been investigated in [1, 14, 27, 28, 38, 39, 60]. Li et al. [27] proposed a framework called TempoRadar to study temporal attention to features from 2 ego-centric bird-eye-view (BEV) radar frames. It has shown promising performance gains when evaluated on the large-scale open *Radiate* [47] dataset.

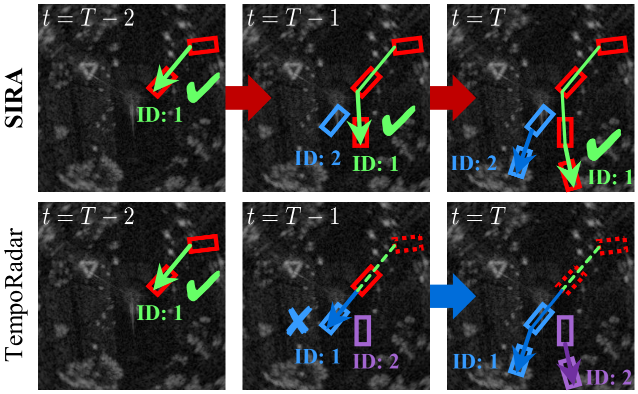


Figure 1. Conventional radar perception pipelines such as TempoRadar [27] (Bottom Row) rely on a limited number (one or two) of frames and the limited time horizon may lead to incorrect feature-level and object-level association (e.g., $t = T - 1$) and propagate to subsequent frames (e.g., $t = T$). In contrast, SIRA (Top Row) accounts for joint spatio-temporal consistency over an extended temporal horizon (e.g., all 3 frames here), allowing for more accurate association in nonlinear motion scenarios even in an ego-centric viewpoint.

However, such limitations can be exacerbated by nonlinear object motion, particularly from an ego-centric BEV. In particular, low frame rates result in significant influence from the nonlinearity of object motion, leading to frequent tracking errors. Conventional radar perception pipelines such as TempoRadar enables prediction based on information from the previous frame, but in the case of objects with fast and nonlinear motion within radar frames, such information is inadequate (Bottom of Fig. 1). Although applying Kalman filter (KF [24])-based algorithms [4, 8, 12, 62], is possible, radar perception is difficult to relate accurately due to a complex combination of factors, including the effects of high-speed nonlinear motion dynamics and the lack of detailed appearance features due to low resolution. To address these limitations and improve radar perception for object detection and tracking, we propose a framework called ***scalable inter-frame relation & association (SIRA)***. SIRA consists of two modules: extended temporal relation (ETR) and motion consistency track (MCTrack). The contributions of this study are as follows:

- We introduce ETR, generalizing the existing temporal relation layer from two consecutive frames to multiple inter-frames with temporally regrouped window attention for scalability. It emphasizes the temporal consistency of moving objects by enabling accurate detection while maintaining computational efficiency over long time horizon. This can facilitate easy detection through consistent correlations across multiple frames at the object level.
- We designed MCTrack based on the concept of pseudo-tracklets, which are generated by using a learnable module to predict the arbitral nonlinear motion of an object between multiple frames, and the association caused by these pseudo-tracklets enhances spatial consistency during inference. Thus, MCTrack enables more reliable position predictions, even in scenarios with fast-moving objects and low frame rates.
- We propose ***SIRA*** that adopts a loss function for the end-to-end learning of these two modules, achieving stable predictions that capture the spatio-temporal consistency of nonlinear moving objects.
- We evaluate SIRA on *Radiate* [47], a BEV radar dataset. Our approach achieves 58.11 mAP@0.5 for oriented object detection and 47.79 MOTA for multiple object tracking on the *Radiate* dataset, surpassing previous state-of-the-art by a margin of $+4.11$ mAP@0.5 and $+9.94$ MOTA, respectively.

2. Related Work for Radar Perception

Automotive radar predominantly employs a frequency-modulated continuous waveform (FMCW) for object detection, generating point clouds. The fundamental of FMCW is explained in Appendix 18. In addition, we defer a short review of recent visual tracking in Appendix 6.

Detection by Radar: For automotive perception, radar-assisted multimodal approaches were proposed [10, 29, 34, 42, 51, 55]. Compared with multimodals, standalone radar-only perception has been studied in [1, 13, 14, 27, 28, 38, 39, 60]. A multi-view feature fusion method was proposed in [14] to combine features from range-Doppler, range-angle, and angle-Doppler radar heatmaps for object classification. As opposed to single-frame radar feature extraction, Li et al. [27] proposed TempoRadar with 2 frames.

Mutiple Object Tracking by Radar: Object tracking with radar has seen several proposals depending on the sparsity or density of the radar points obtained for each object [40]. For sparse radar detection points, model-based tracking algorithms have been explored in the context of extended object tracking (EOT) [16]. They use Bayesian filtering [3, 6, 17, 25, 37, 49, 53] to model the spatial distribution of radar detection points across the vehicle's range and predict and update the extended states such as position and velocity. Moreover, to address the nonlinearity problem due to objects deviating from constant linear motion, algorithms such as extended KF [48] and unscented KF [23] have been proposed to handle nonlinear motion using first- and third-order Taylor approximations. However, these still rely on approximating the Gaussian prior distribution assumed by the KF, making modeling challenging for movements where the next position is determined by human intent, such as in vehicles. Particle filter [18] addresses nonlinear motion using a sampling-based posterior estimation, which requires exponential computation. For high-density radar detection points, following [58, 65], TempoRadar extended the achieved strong tracking performance through learning. Our proposed framework extends KF-based methods and learning-based approaches by assuming high-density radar detection points. It explicitly considers strong object-level consistency by using multiple frames to capture the nonlinear motion of objects.

3. Scalable Inter-frame Relation & Association

An overview of the SIRA framework is illustrated in Fig. 2 with two main modules: 1) ETR and 2) MCTrack. ETR focuses on the temporal consistency, while MCTrack captures the spatial motion consistency, ensuring the continuity and accuracy of object detection and tracking at the output.

3.1. Preliminary

Encoder: Radar perception pipelines employ an encoder to transform the radar frame $I_t \in \mathbb{R}^{1\times H\times W}$ into high-level features and accentuate the position of objects.

$$\mathbf{Z}_t := \mathcal{F}_\theta\left(I_t\right) \in \mathbb{R}^{C\times\frac{H}{s}\times\frac{W}{s}}, \tag{1}$$

where C, H, W, and s represent the number of channels, height, width, and downsampling ratio over the spatial dimension, respectively. $\mathcal{F}_\theta\left(\cdot\right)$ is encoder such as ResNet [19] with parameters θ. By denoting multiple T radar frames as $\mathbf{I} = \{I_t\}_{t=1}^{T} \in \mathbb{R}^{T\times H\times W}$, we can obtain informative features $\mathbf{Z}_t = \mathcal{F}_\theta\left(\mathbf{I}\right)$.

Decoder: The decoder estimates the bounding boxes from the features. To localize objects, the two-dimensional (2D) center coordinates (x_t, y_t) of the top-K peak values $\widehat{c}_t$

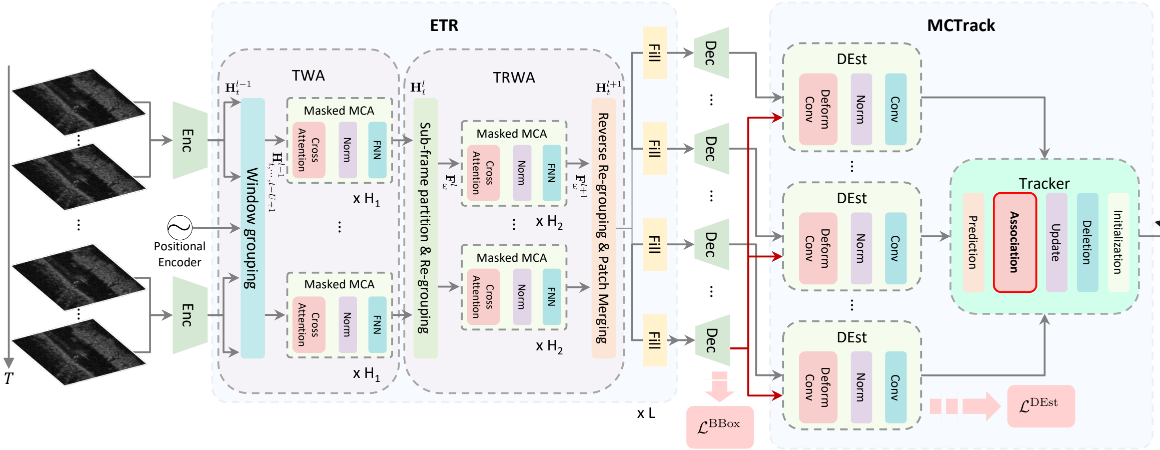


Figure 2. The architecture of SIRA with two modules: 1) extended temporal relation (ETR) capturing the temporal feature consistency while maintaining computational efficiency, and 2) motion consistency track (MCTrack) estimating pseudo-direction of objects during training and establishing pseudo-tracklets for better association in inference. The detection loss $\mathcal{L}_t^{\text{BBox}}$ and pseudo-direction loss $\mathcal{L}^{\text{DEst}}$ are used to train the pipeline end-to-end for object detection and tracking.

in the heatmap, corresponding width $\widehat{w}_t$ and length $\widehat{h}_t$, orientation $\widehat{\vartheta}_t$, and 2D offsets $(\widehat{o}_{x,t}, \widehat{o}_{y,t})$ are predicted as the output bounding box of an object with decoder heads $\mathcal{G}_\theta$ as:

$$\left(x_t, y_t, \widehat{w}_t, \widehat{h}_t, \widehat{\vartheta}_t, \widehat{o}_{x,t}, \widehat{o}_{y,t}, \widehat{c}_t\right)^\top = \mathcal{G}_\theta\left(\mathbf{Z}_t\right). \quad (2)$$

One such decoder is the one used in CenterPoint [64].

Exploiting Temporality: For radar perception, it is necessary enhance the feature extraction utilizing additional properties from the temporal domain. One straightforward way is to stack multiple frames as the input to the encoder, i.e., $\mathbf{Z}_t = \mathcal{F}_\theta\left(\mathbf{I}\right)$. To exploiting the feature-level temporal relation, TempoRadar [27] introduces a temporal relation layer (TRL) that selects top-K features $\mathbf{H}_t \in \mathbb{R}^{C\times K}$ from $\mathbf{Z}_t := \mathcal{F}_\theta\left(I_{t,t-1}\right)$ and $\mathbf{H}_{t-1} \in \mathbb{R}^{C\times K}$ from $\mathbf{Z}_{t-1} := \mathcal{F}_\theta\left(I_{t-1,t}\right)$, where $I_{t-1,t}$ concatenates two consecutive radar frames along the channel dimension in the order of $(t-1, t)$ with the following feature selector $\mathcal{S}_K$:

$$\mathbf{H}_t = \mathcal{S}_K\left(\mathbf{Z}_t\right), \quad t = \{t-1, t\}. \quad (3)$$

By concatenating the $2K$ selected features as $\mathbf{H}_{t,t-1} = \left[\mathbf{H}_t, \mathbf{H}_{t-1}\right]^\top$, TRL further computes masked multi-head cross-attention (MCA) as

$$\mathcal{A}\left(\mathbf{V}, \mathbf{X}\right) := \text{softmax}\left(\frac{\mathbf{M} + q\left(\mathbf{X}\right)k\left(\mathbf{X}\right)^\top}{\sqrt{d}}\right)v\left(\mathbf{V}\right) \quad (4)$$

where $\mathbf{V} = \mathbf{H}_{t,t-1}$, $\mathbf{X} = \mathbf{H}_{t,t-1}^{\text{pos}}$ is the concatenated feature $\mathbf{H}_{t,t-1}$ supplemented by the positional encoding, $\{q\left(\cdot\right), k\left(\cdot\right), v\left(\cdot\right)\}$ are query/keys/values, and d is the query/key dimension. The masking matrix $\mathbf{M}$ is designed to turn off the attention between features from the same frame and allow for only cross-frame feature attention to ensure temporal feature consistency.

These enhanced features are refilled back to $\mathbf{Z}_t$ and $\mathbf{Z}_{t-1}$ at corresponding spatial coordinates and fed to the decoder for object detection and tracking. Refer to Appendix 8 for the top-K feature selector $\mathcal{S}_K$ and the design of $\mathbf{M}$.

3.2. ETR: Extended Temporal Relation

The ETR module borrows the concept of shifted window attention in Swin Transformer [31] but in a deformable temporal fashion. It generalizes the TRL over a longer time horizon of consecutive frames with a scalable complexity. In the following, we introduce the two main blocks: temporal window attention (TWA) and temporally regrouped window attention (TRWA) of ETR shown in Fig. 2.

Temporal Window Attention: The l-th TWA layer expands the TRL from 2 consecutive frames to a temporal window of $U \geq 2$ frames and computes masked MCA within each window. In Fig. 3, we group $U = 4$ consecutive frames into one temporal window (in dash boxes) and we have 4 windows for $T = 16$ frames.

For each temporal window $\{t, t-1, \cdots, t-U+1\}$, we cyclically shift the frame indices and concatenate the U shifted radar frames along the channel dimension for the backbone feature extraction, i.e.,

$$\begin{aligned}\mathbf{Z}_t &:= \mathcal{F}_\theta\left(I_{t,t-1,\cdots,t-U+1}\right), \\ \mathbf{Z}_{t-1} &:= \mathcal{F}_\theta\left(I_{t-1,t-2,\cdots,t-U+1,t}\right), \cdots, \\ \mathbf{Z}_{t-U+1} &:= \mathcal{F}_\theta\left(I_{t-U+1,t,t-1,\cdots,t-U+2}\right). \end{aligned} \quad (5)$$

It is easy to see that, when $U = 2$, this reduces to the TRL. We then follow the same top-K feature selector as the TempoRadar (refer to Appendix 8)

$$\mathbf{H}_t = \mathcal{S}_K(\mathbf{Z}_t), \quad t = \{t, t-1, \cdots, t-U+1\}. \qquad (6)$$

By concatenating features from the temporal window of U frames, we have $\mathbf{H}^{l-1}_{t,\cdots,t-U+1} = [\mathbf{H}^{l-1}_t, \cdots, \mathbf{H}^{l-1}_{t-U+1}]^\top$, where the superindex denotes the layer index in the ETR model and $\mathbf{H}^0_t$ takes $\mathbf{H}_t$ of (6) as input for the first layer. We apply the masked MCA of (4) H_1 times to $\mathbf{H}^{l-1}_{t,\cdots,t-U+1}$ with a masking matrix $\mathbf{M}$ of dim $UK \times UK$ for cross-frame feature attention within each window. Collecting from all windows, the TWA block obtains the features $\mathbf{H}^l_t, \cdots, \mathbf{H}^l_{t-T+1}$ from all T frames at its output.

Temporally Regrouped Window Attention: To allow for cross-window attention, we regroup the subset features from different windows in a deformable temporal order. First, we partition the K features of each frame into Ω sub-frame patches with a stride S. Each sub-frame patch consists of M features. As shown in Fig. 3, one choice for a non-overlapping sub-frame partition is $M = K/2$ and $S = K/2$ (assuming K is even) where each frame is partitioned into $\Omega = 2$ sub-frame patches, as illustrated in two contrasting colors for each frame in Fig. 3. Alternatively, we may choose $S < M$ for overlapping partition. The resulting sub-frame patches of frame t are defined as $\mathbf{H}^l_t[\omega] \in \mathbb{R}^{C\times M}, \omega = 1, \cdots, \Omega$. For more discussion of patch size, refer to Appendix 11.

The sub-frame patches are regrouped into a new set of windows in a deformable temporal order for cross-window attention. For the newly regrouped window, the features are aggregated as

$$\mathbf{F}^l_t(\omega) := \left\{\mathbf{H}^l_t[\omega], \mathbf{H}^l_{t-U}[\omega], \cdots, \mathbf{H}^l_{t-T+U}[\omega]\right\}^\top, \quad (7)$$

As illustrated in the top right portion of Fig. 3, the regrouping operation extracts one sub-frame patch from each window and results in $U = 4$ patches and $UM = UK/2$ features in each new window. Subsequently, we apply the masked MCAs of (4) H_2 times over the aggregated feature $\mathbf{F}^l_t(\omega)$ in each new window with an affordable cross-window attention complexity of $TM/U \times TM/U$.

The cross-window attentive features are re-grouped in the reverse manner to construct the K features of each frame according to the temporal (t) and patch (ω) indices. In the case of overlapping patch partitioning, i.e., $S < M$, a patch merging operation $\mathcal{M}$ is necessary to merge the features $\mathbf{H}^{l+1}_t = \mathcal{M}\{\mathbf{H}^{l+1}_t[1], \cdots, \mathbf{H}^{l+1}_t[\Omega]\}$ at the overlapping positions. Patch merging operations (mean, sum and max) will be examined in Section 4.3. The TRWA block outputs $\mathbf{H}^{l+1}_t, \cdots, \mathbf{H}^{l+1}_{t-T+1}$ for all T frames, sharing the same dimension as the input $\mathbf{H}^l_t, \cdots, \mathbf{H}^l_{t-T+1}$.

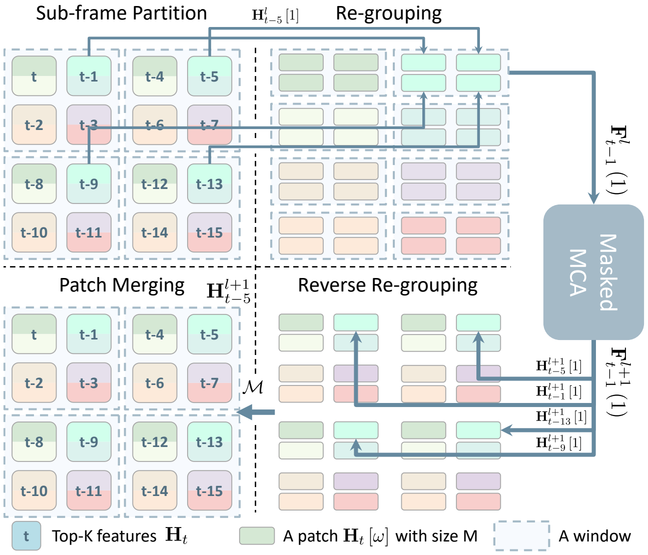


Figure 3. The TRWA block of the ETR module. Each frame is partitioned into sub-frame patches (in two contrasting colors of each frame in Top Left) and these patches are regrouped into new windows (Top Right) in a deformable temporal order (arrow lines). Masked multi-head cross-attention (MCA) is applied to new regrouped windows for scalable cross-window attention.

Stacking as a Stage: We can stack the TWA and TRWA blocks as one stage and repeat the stage L times. In between stages, the output of TRWA block serves the input to the TWA block in the next stage. Finally, we put these features $\mathbf{H}^{l+1}_t, \cdots, \mathbf{H}^{l+1}_{t-T+1}$ back to $\{\mathbf{Z}_t, \cdots, \mathbf{Z}_{t-T+1}\}$ at corresponding spatial coordinates. The effect of L will be examined in Section 4.3.

Complexity Analysis: For a given T, K, and the number of stages L, the computational complexity expressions for TempoRadar [27] and the ETR module are shown below

$$\text{TempoRadar: } (TK)^2 L \qquad (8)$$

$$\text{ETR: } (\text{TWA} + \text{TRWA})\, L = K^2TUL + MT^2KL/U \quad (9)$$

where U is the number of frames in one temporal window in the TWA block and M is the number of features for each sub-frame patch in the TRWA block. Note that, if $U = T$ and $M = K$, ETR reduces to the TWA module only, resulting in a full-size attention like TempoRadar. In this case, the ETR complexity in (9) reduces to that of TempoRadar in (8). Appendix 13 provides numerical comparison of the complexity in several settings.

3.3. MCTrack: Motion Consistency Track

As shown in Fig. 2, MCTrack takes the temporally enhanced features $\{\mathbf{Z}_t\}$ from the ETR output, and applies the decoding heads on each $\mathbf{Z}_t$ for bounding box estimation. To further exploit motion consistency, we introduce two MC

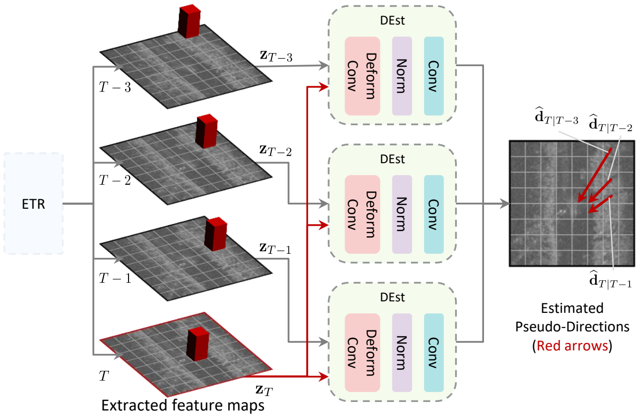


Figure 4. Direction Estimation (DEst) decoder head. Each DEst head takes a pair of 2 frames $\mathbf{Z}_T$ and $\mathbf{Z}_{T-\tau}$, and estimates the pseudo-direction $\widehat{\mathbf{d}}_{T|T-\tau}$ (arrow lines in red).

modules: one for training and one for inference, for improved detection and tracking performance.

Motion Consistency for Training: We introduce the concept of **pseudo-direction** to improve motion consistency during training. Pseudo-directions are vectors that directly predict the current object position from each of the previous frames, using a decoder head with learnable parameters. It is used to iteratively refine object positions between frames during learning and the pseudo-direction loss contributes to the overall training loss in Section 3.4.

To compute the τ-step pseudo-direction $\widehat{\mathbf{d}}_{T|T-\tau}$[1] from the past frame $T-\tau$ to frame T, we design a specific decoder head $\mathcal{G}_{\theta}^{\text{DEst}}(\cdot)$: direction estimation (DEst) with learnable parameters θ in Fig. 4,

$$\widehat{\mathbf{d}}_{T|T-\tau} = \mathcal{G}_{\theta}^{\text{DEst}}\left(\mathbf{Z}_T, \mathbf{Z}_{T-\tau}\right)\left[\mathbf{p}_{\mathbf{z}_T}\right] \in \mathbb{R}^2, \qquad (10)$$

where $\mathbf{Z}_T$ and $\mathbf{Z}_{T-\tau}$ are temporally enhanced features at frame T and $T-\tau$, $\mathbf{p}_{\mathbf{z}_T}$ is a two-dimensional coordinate, and $\tau = 1, 2, \cdots, T-1$. Fig. 4 shows the DEst head architecture, comprising the deformable convolution [9], normalization, and convolution layers. The deformable convolution is particularly used to capture features of objects that have undergone significant displacement across τ frames.

The estimated vectors represent the positional differences of objects across τ frames. It is essential to address scenarios where objects move significantly within just one frame due to low frame rates and ego-vehicle motions.

Motion Consistency for Inference: In inference, we use a KF-based tracker such as OC-SORT [8] to enforce motion consistency. As shown in Fig. 2, the tracker consists of a number of steps with the most crucial one in Association.

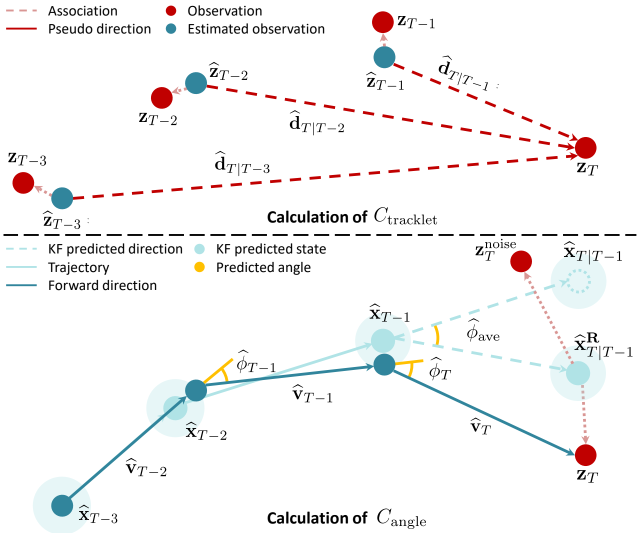

Association
Pseudo direction
Observation
Estimated observation
$\mathbf{z}_{T-1}$
$\widehat{\mathbf{z}}_{T-1}$
$\widehat{\mathbf{d}}_{T|T-1}$
$\widehat{\mathbf{z}}_{T-2}$
$\mathbf{z}_{T-2}$
$\widehat{\mathbf{d}}_{T|T-2}$
$\mathbf{z}_{T-3}$
$\widehat{\mathbf{d}}_{T|T-3}$
$\widehat{\mathbf{z}}_{T-3}$
$\mathbf{z}_T$
Calculation of C_{tracklet}
KF predicted direction
Trajectory
Forward direction
KF predicted state
Predicted angle
$\mathbf{z}_T^{\text{noise}}$
$\widehat{\mathbf{x}}_{T|T-1}$
$\widehat{\mathbf{x}}_{T-1}$
$\widehat{\phi}_{\text{ave}}$
$\widehat{\mathbf{x}}^{\mathbf{R}}_{T|T-1}$
$\widehat{\phi}_{T-1}$
$\widehat{\phi}_T$
$\widehat{\mathbf{v}}_{T-1}$
$\widehat{\mathbf{v}}_T$
$\widehat{\mathbf{x}}_{T-2}$
$\widehat{\mathbf{v}}_{T-2}$
$\widehat{\mathbf{x}}_{T-3}$
$\mathbf{z}_T$
Calculation of C_{angle}

Figure 5. The calculation of similarity metrics C^{angle} and C^{tracklet} in MCTrack at inference. A pseudo-tracklet $\{\{\widehat{\mathbf{z}}_t\}_{t=1}^{T}, \{\widehat{\mathbf{v}}_t\}_{t=2}^{T}\}$ is constructed with $\widehat{\mathbf{d}}_{T|T-\tau}$ estimated with DEst, and is used for association: (Top) rotating a state $\mathbf{x}_{T|T-1}$ to be more correlate the observation $\mathbf{z}_T$, (Bottom) directly correlating the observations $\mathbf{z}_t$ with $\widehat{\mathbf{z}}_t$.

To this end, we further introduce the concept of **pseudo-tracklet**[2], constructed from the above pseudo-direction estimation. A pseudo-tracklet consists of a pair of vectors: $\{\{\widehat{\mathbf{z}}_t\}_{t=1}^{T}, \{\widehat{\mathbf{v}}_t\}_{t=2}^{T}\}$. $\widehat{\mathbf{z}}_t$ is an estimated observation with pseudo-direction $\widehat{\mathbf{d}}_{T|T-\tau}$ and $\mathbf{z}_T$ (Top of Fig. 5), and $\widehat{\mathbf{v}}_t$ is the forward direction linking between the estimated observations (Bottom of Fig. 5).

The pseudo-tracklet can only be calculated from observations that are independent of the state of KF, and explicitly contains information about the movement of the object from the past to the present. We use this pseudo-tracklet to design the similarity metric in the association:

$$C^{\text{MCTrack}} = \lambda C^{\text{angle}} + (1-\lambda) C^{\text{tracklet}}, \qquad (11)$$

$$C^{\text{tracklet}} = \frac{1}{T-1} \sum_{\tau=1}^{T-1} \text{GIoU}\left(B_{\mathbf{z}_{T-\tau}}, B_{\widehat{\mathbf{z}}_{T-\tau}}\right), \qquad (12)$$

$$C^{\text{angle}} = \text{GIoU}\left(B_{\mathbf{z}_T}, B_{\widehat{\mathbf{x}}^{\mathbf{R}}_{T|T-1}}\right), \qquad (13)$$

where λ is the weighting coefficient, B represents the BBox with subscripts, and GIoU [46] denotes the similarity determined based on the distance between two BBoxes. In other words, C^{tracklet} and C^{angle} represent the similarity between the similarity between the pseudo-tracklet and the trajectory of the KF, and the current observation $\mathbf{z}_T$ and the rotated state $\widehat{\mathbf{x}}^{\mathbf{R}}_{T|T-1}$ of the KF, respectively.

[1] With slightly abused notation, we use T to denote not only the number of frames, but also current frame index in this section.

[2] A tracklet is essentially an aggregation of a small number of consecutive sensor reports processed by a sensor level tracker [11]. We use the tracklet as a short trajectory from a set of observations.

As shown in top of Fig. 5, C^{tracklet} directly correlates the observations $\mathbf{z}_t$s of the KF trajectory with the estimated observations $\widehat{\mathbf{z}}_t$ with the pseudo-direction. This approach, unlike the conventional method of correlating with only one observation value in the current frame, is more robust to motion. The effectiveness of using both C^{tracklet} and C^{angle} is reported in Section 4.3. Refer to Algorithm 1 in Appendix 11 for the pseudo-code of SIRA in inference.

In addition, as shown in bottom of Fig. 5 which represents the calculation of C^{angle}, the predicted state $\widehat{\mathbf{x}}_{T|T-1}$ with KF from the previous state $\widehat{\mathbf{x}}_{T-1}$ is rotated with a rotation matrix $\mathbf{R}$ of angle ϕ_{ave}. It can be calculated as $\mathbf{p}_{\widehat{\mathbf{x}}^{\mathbf{R}}_{T|T-1}} = \mathbf{R}(\mathbf{p}_{\widehat{\mathbf{x}}_{T|T-1}} - \mathbf{p}_{\widehat{\mathbf{x}}_{T-1}}) + \mathbf{p}_{\widehat{\mathbf{x}}_{T-1}}$, where the angle $\widehat{\phi}_{\text{ave}}$ can be calculated as $\widehat{\phi}_{\text{ave}} = \frac{1}{T-2}\sum_{\rho=0}^{T-3}\widehat{\phi}_{T-\rho}$ such that $\widehat{\phi}_{T-\rho} = \cos^{-1}\frac{(\widehat{\mathbf{v}}_{T-\rho}\cdot\widehat{\mathbf{v}}_{T-\rho-1})}{\|\widehat{\mathbf{v}}_{T-\rho}\|\|\widehat{\mathbf{v}}_{T-\rho-1}\|}$. By using this rotated state $\widehat{\mathbf{x}}^{\mathbf{R}}_{T|T-1}$, we can avoid a high correlation between the predicted state assuming linear motion and the incorrect observation $\mathbf{z}^{\text{noise}}_T$.

Our approach exploits the proposition that the temporally enhanced features across multiple frames from ETR allows for more robust estimation of the pseudo-direction $\widehat{\mathbf{d}}_{T|T-\tau}$ from past frame $T-\tau$ to current frame T, compared with conventional single-frame based approaches.

3.4. Learning

A loss function is constructed not only to acquire conventional detection capabilities, but also to provide a clear guideline to enhance tracking performance. It consists of two components: a loss between the predicted and the ground truth BBox ($\mathcal{L}^{\text{BBox}}_t$), and a loss of the pseudo-direction in which an object has moved between frames and the actual movement direction ($\mathcal{L}^{\text{DEst}}_t$), as shown in Fig. 2.

$$\mathcal{L}_\theta := \sum_{t=1}^{T}\left(\mathcal{L}^{\text{DEst}}_t + \mathcal{L}^{\text{BBox}}_t\right). \tag{14}$$

For each training step, our training procedure calculates $\mathcal{L}_\theta$ and does the backward for both $t = 1$ to $t = T$ and $t = T$ to $t = 1$ simultaneously. Therefore, optimization $\min_\theta \mathcal{L}_\theta$ can be viewed as a bidirectional backward-forward training through T frames. For more clear trainig procedure, refer to Fig. 8 in Appendix 11.

Oriented Bounding Box Loss: We pick the object's center coordinates from the heatmap, and learn its attributes from feature representations through regression. Regression functions, which are heatmap loss $\mathcal{L}^{\text{h}}_t$, width & Length loss $\mathcal{L}^{\text{b}}_t$, orientation loss $\mathcal{L}^{\text{r}}_t$, and offset loss $\mathcal{L}^{\text{o}}_t$, compose the training objective by a linear combination:

$$\mathcal{L}^{\text{BBox}}_t = \frac{1}{N_{\text{gt}}}\sum_{k=1}^{N_{\text{gt}}}\left(\mathcal{L}^{\text{b}}_{t,k} + \mathcal{L}^{\text{r}}_{t,k} + \mathcal{L}^{\text{o}}_{t,k}\right) - \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}^{\text{h}}_{t,i}, \tag{15}$$

where N denotes the total number of pixels in the heatmap and N_{gt} is the total number of ground truth bounding boxes. Refer to Appendix 9 for mathematical definition of each loss component.

Pseudo-Direction Estimation Loss: $\mathcal{L}^{\text{DEst}}$ represents a pseudo-direction estimation loss:

$$\mathcal{L}^{\text{DEst}}_t = \frac{1}{N_{\text{gt}}}\sum_{k=1}^{N_{\text{gt}}}\mathcal{L}^{\text{DEst}}_{t,k}, \tag{16}$$

$$\mathcal{L}^{\text{DEst}}_{t,k} = \frac{1}{T-1}\sum_{\tau=1}^{T}\begin{cases} S_{L_1}\left(\left\|\widehat{\mathbf{d}}_{t|\tau} - \mathbf{d}^{\text{gt}}_{t|\tau}\right\|\right) & \tau \neq t \\ 0 & \tau = t \end{cases}, \tag{17}$$

where $\widehat{\mathbf{d}}_{t|\tau} = \mathcal{G}^{\text{DEst}}_\theta(\mathbf{Z}_t, \mathbf{Z}_\tau)\left[\mathbf{p}^{\text{gt}}_{t,k}\right]$ denotes a two-dimensional direction from a position of time τ to a position of time t as mentioned in Section 3.3. $\mathbf{p}^{\text{gt}}_{t,k}$ denotes the coordinate $(x_{t,k}, y_{t,k})$ of the center of k-th ground truth object and $S_{L_1}(\cdot)$ is a smooth L_1 loss [15]. $\mathbf{d}^{\text{gt}}_{t|\tau} = \mathbf{p}^{\text{gt}}_{t,k} - \mathbf{p}^{\text{gt}}_{\tau,k}$ denotes the ground truth direction, which can be calculated from the difference between the coordinates of the k-th object. This loss improves the consistency of the detection positions between frames, which impacts both the detection and the tracking performance.

4. Experiments

4.1. Experimental Setup

Due to page limitations, more details on experimental settings are shown in Appendix 12.

Dataset: We use the automotive radar dataset: *Radiate* [47] in our experiments, the same as TempoRadar in [27]. The reasons to use this dataset are that it contains high-resolution radar images, provides well-annotated oriented bounding boxes with tracking IDs for objects, and records various real driving scenarios in adverse weather, please refer to Appendix 7 for more details of the reasons. *Radiate* consists of video sequences recorded in adverse weathers, including sun, night, rain, fog and snow. We follow the official 3 splits: "train in good weather" (22383 frames, only in good weather, sunny or overcast), "train good & bad weather" (9749 frames, both good & bad weather conditions), and "test" (11305 frames, all kinds of weather conditions).

Implementation: Our baseline detectors include: 1) RetinaNet [30], 2) CenterPoint [64], 3) BBAVectors [57], 4) TempoRadar [27] (referred to as TR in all results). We also implemented 5) a Sequential TempoRadar (SeTR) that

Table 1. Experimental results of object detection on *Radiate*. The number following the model name indicates the # of layers in the Resnet, and the number in parentheses indicates the # of frames T.

	Train good weather		Train good & bad weather	
	mAP@0.3	mAP@0.5	mAP@0.3	mAP@0.5
RetinaNet-18 (1)	52.50±1.81	37.83±1.82	49.44±1.32	31.57±1.54
CenterPoint-18 (1)	58.69±3.09	49.41±2.94	55.83±3.28	44.48±3.19
BBAVectors-18 (1)	59.38±3.47	50.53±2.07	56.84±3.45	45.43±2.87
TR-18 (2)	62.79±2.01	53.11±1.96	58.87±3.31	46.42±3.24
TR-18 (4)	66.37±1.62	53.23±1.67	65.10±1.67	52.47±1.21
SeTR-18 (4)	65.97±2.03	55.79±2.12	64.62±1.79	51.78±1.81
SIRA-18 (4)	**67.28**±1.47	**56.98**±1.35	**65.37**±1.76	**52.88**±1.60
RetinaNet-34 (1)	50.79±3.10	35.61±3.35	48.09±3.85	31.10±3.37
CenterPoint-34 (1)	59.42±1.92	50.17±1.91	53.92±3.44	42.81±3.04
BBAVectors-34 (1)	60.88±1.79	51.26±1.99	55.87±2.90	44.61±2.57
TR-34 (2)	63.63±2.08	54.00±2.16	56.18±4.27	43.98±3.75
TR-34 (4)	67.48±0.94	57.01±1.03	64.60±2.08	51.99±1.94
SeTR-34 (4)	67.30±1.80	56.61±1.83	65.51±1.52	52.43±1.51
SIRA-34 (4)	**68.68**±1.12	**58.11**±1.40	**66.14**±0.83	**53.79**±1.14

stacks self-attention for two consecutive frames and sequentially connects them through T frames. We defer the description of the SeTR to Appendix 10. We use ResNet-18 and ResNet-34 for the backbone feature extraction.

For MOT, we implemented several trackers that have been well demonstrated in this task for comparison. These trackers include the following: CenterTrack [65] and OC-SORT [8]. For the results of CenterTrack with TempoRadar and ResNet, we copied directly from the paper [27] except for TempoRadar with 34 layers. And for the KF-based method, we use the specific parameters and show the parameters in Appendix 17. We follow [47] 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. For all numerical results, we apply a center crop with size 256 × 256 upon input images and exclude the targets outside this scope. We additionaly report the detection results with the full size (1152 × 1152) images in Appendix 15.

Metrics: We adopt the mean average precision (mAP) with intersection over union (IoU) at 0.3, 0.5, and 0.7 (reported in Appendix 15) to evaluate detection performance. The numbers are averaged over 10 random seeds. For MOT, we adopt MOTA [35] and IDF1 [32] as the main metrics. MOTA focuses more on the detection performance, while IDF1 reflects on the performance of association and identity preservation. Other metrics [35] such as ID switch (IDs), fragmentation (frag), MT, and PT are also reported. Definitions of these MOT metrics are included in Appendix 14.

4.2. Main Results

Detection: We report the detection results in Table 1. The benefits of exploiting longer temporal relation for radar object detection are evident in improvements of about +3 mAP@0.3 and about +2.5 mAP@0.5 from single frame of RetinaNet, CenterPoint, BBAVectors to two frames of the TempoRadar, and further more of about +5 mAP@0.3 and about +4 mAP@0.5 from two frames to four frames of the best among TempoRadar, SeTR, and SIRA. In both training splits, our SIRA consistently outperforms TempoRadar and its simple extension SeTR with 4 radar frames. The improvement margin is more significant in the "good & bad weather" training split when ResNet34 is the backbone network. We report the effectiveness of increasing the number of frames in Appendix 15.

Table 2. Experimental results of MOT on *Radiate*.

Train good weather	MOTA↑	IDF1↑	IDs↓	Frag.↓	MT↑	PT↑
ResNet-18 (1) CenterTrack	13.01	-	873	920	269	254
ResNet-34 (1) CenterTrack	14.55	-	802	831	282	279
TR-18 (2) CenterTrack	33.59	-	349	498	145	330
TR-34 (2) CenterTrack	37.85	39.90	457	511	108	246
TR-34 (2) OC-SORT	40.74	45.01	**151**	**291**	124	172
TR-18 (4) CenterTrack	42.77	44.91	519	520	244	**336**
TR-34 (4) CenterTrack	43.64	44.17	503	538	197	326
TR-34 (4) OC-SORT	44.01	44.27	354	497	194	333
SeTR-18 (4) CenterTrack	42.11	50.33	658	561	261	317
SeTR-34 (4) CenterTrack	44.57	48.72	875	602	348	299
SeTR-34 (4) OC-SORT	40.16	28.20	775	689	**370**	305
ETR-34 (4) CenterTrack	46.06	50.81	1832	613	345	305
ETR-34 (4) OC-SORT	47.11	50.04	540	481	343	313
SIRA-34 (4) CenterTrack*	47.30	50.16	1249	566	354	300
SIRA-34 (4) OC-SORT	**47.79**	**51.13**	523	488	342	314

* C^{tracklet} is only used for association since this is not based on SORT.

Tracking: Table 2 illustrates the results of MOT. Similar conclusions can be made by observing the improvement margins in almost all metrics by using more radar frames. If we narrow down to the case of 4 frames and with CenterTrack as the tracker, SIRA-34 shows a significant improvement of +3.66 over TR-34 and +2.83 over SeTR-34 in MOTA. The combination of SIRA+OC-SORT can further improve the MOT by another +0.49 over SIRA+CenterTrack.

Compared with ETR (without $\mathcal{L}_t^{\text{DEst}}$ for training), SIRA shows consistent improvement in both MOTA and IDF1, highlighting the effectiveness of modeling consistency in object movement. For other metrics such as Frag., MT, and PT, SIRA shows fluctuating but close-to-the-best performance. Full results, including the effectiveness of increasing the number of frames and other indicators, are reported in Appendix 15 due to paper space limitations.

Visualization: We show the visualization results in Fig. 6. Each set of figures represents ground truth in the upper row and predictions in the lower row. It is observed that many of the predictions are made at approximately the same position

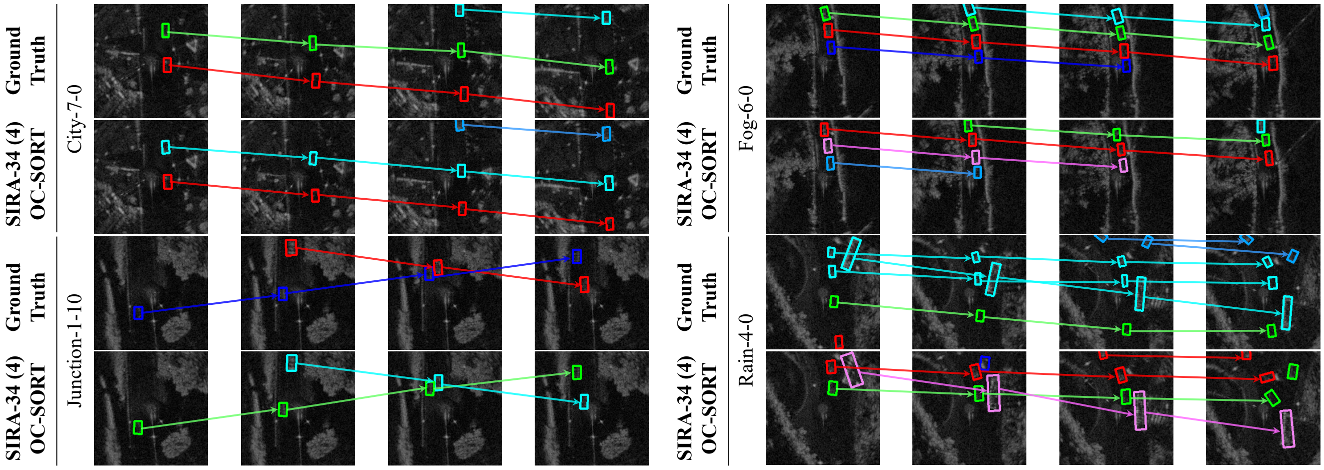


Figure 6. Visualizations on radar perception on *Radiate*. 4 sets of MOT results are shown in radar sequences of city-7-0, fog-6-0, junction-1-10 and rain-4-0. Each set contains 4 frames. Bounding boxes are ground truth or object detection from SIRA. Colors indicate object IDs and plotted arrows show the motion of detected objects.

$\mathcal{M}$	mAP@0.3	mAP@0.5
Mean	65.15±2.20	55.06±2.07
Sum	65.76±2.15	55.55±1.59
Max	**67.67**±1.18	**56.47**±1.54

(a) **Operations** $\mathcal{M}$. Using the Max operation works the best.

H_1	H_2	mAP@0.3	mAP@0.5
1	1	66.95±1.47	56.65±2.38
2	1	67.59±0.83	57.59±0.84
1	2	68.36±0.94	**58.46**±0.91
2	2	**68.68**±1.12	58.11±1.40

(b) **# of MCAs**. A larger H_2 contributes more than a larger H_1.

L	mAP@0.3	mAP@0.5
1	68.68±1.12	58.11±1.40
2	68.68±0.83	58.24±1.19
3	69.12±1.32	**58.28**±1.34
4	**69.16**±1.06	58.26±1.27

(c) **# of Stages**. More stages slightly improves the detection.

C^{tracklet}	C^{angle}	MOTA↑	IDF1↑
-	-	47.11	50.04
✓	-	47.11	50.02
-	✓	47.00	50.05
✓	✓	**47.79**	**51.13**

(d) **Associations** C. Using both C^{tracklet} and C^{angle} works the best.

Table 3. **SIRA ablation experiments** on *Radiate*. If not specified, we used SIRA-34 (4) trained on train good weathter and followed the experimental settings for other parameters. The best performance is marked in gray.

as the annotations. Furthermore, correct predictions are observed for complex motions, including nonlinear motions. More visualizations are included in Appendix 16 with more comparison to other baseline methods.

4.3. Ablation Study

Patch Merging Operator: In the context of patch merging within ETR, it is essential to merge feature vectors from overlapping positions. Multiple merging operations, including Mean, Sum and Max, can be considered. In the experiment, we use ETR-34 (4) as the model. Table 3a shows the detection performance. It is seen that the Max operation works best as the Mean and Sum operations may change the temporally enhanced features. We use the Max operation as the default.

Number of Masked MCA (H_1 and H_2): We investigated the effect of the number of masked MCA H_1 in TWA and H_2 of TRWA. The result in Table 3b shows that larger H improves the detection performance. More masked MCAs $H_2 = 2$ in the TRWA contributes to bigger improvement margin than using more masked MCAs $H_1 = 2$. We set $H_1 = 2$ and $H_2 = 2$ as the default.

Number of Stages (L): We investigated the effect of the number of stages L of ETR. Table 3c evaluates the detection performance when L varies from only 1 to 4. Stacking more ETR stages slightly improves the detection performance.

Association in MCTrack: In Table 3d, the ablation study on association reveals that using both C^{tracklet} and C^{angle} leads to improved tracking performance. These facts indicate that SIRA enforces the spatio-temporal consistency and can be effective to deal with nonlinear object motion across consecutive frames. See Appendix 15 for detailed evaluation results on the performance of Pseudo-Direction estimation and on the differences in λ.

5. Conclusion

We overcame the limitations of radar for effective object detection and tracking in automotive perception by introducing the SIRA framework, which includes ETR and MCTrack. SIRA exploits joint spatio-temporal consistency across multiple frames and enables reliable predictions despite low frame rates and nonlinear motion. Our approach outperforms previous state-of-the-art by a big margin in both detection and tracking.

References

[1] Jie Bai, Lianqing Zheng, Sen Li, Bin Tan, Sihan Chen, and Libo Huang. Radar Transformer: An object classification network based on 4d mmw imaging radar. *Sensors*, 21(11), 2021. 1, 2

[2] Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. UNILMv2: Pseudo-masked language models for unified language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*. JMLR.org, 2020. 12

[3] Marcus Baum and Uwe D. Hanebeck. Extended object tracking with random hypersurface models. *IEEE Transactions on Aerospace and Electronic Systems*, 50(1):149–159, 2014. 2

[4] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016. 1, 12

[5] Igal Bilik, Oren Longman, Shahar Villeval, and Joseph Tabrikian. The rise of radar for autonomous vehicles: Signal processing solutions and future research directions. *IEEE Signal Processing Magazine*, 36(5):20–31, 2019. 26

[6] Peter Broßeit, Bharanidhar Duraisamy, and Jürgen Dickmann. The volcanormal density for radar-based extended target tracking. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2017. 2

[7] Jinkun Cao, Hao Wu, and Kris Kitani. Track targets by dense spatio-temporal position encoding. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022. 12

[8] Jinkun Cao, Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris Kitani. Observation-Centric SORT: Rethinking SORT for robust multi-object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9686–9696, 2023. 1, 5, 7, 12

[9] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 764–773, 2017. 5

[10] Fangqiang Ding, Andras Palffy, Dariu M. Gavrila, and Chris Xiaoxuan Lu. Hidden Gems: 4D radar scene flow learning using cross-modal supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9340–9349, 2023. 2

[11] Oliver E. Drummond. Track and tracklet fusion filtering. In *Signal and Data Processing of Small Targets 2002*, pages 176 – 195. International Society for Optics and Photonics, SPIE, 2002. 5

[12] Yunhao Du, Zhicheng Zhao, Yang Song, Yanyun Zhao, Fei Su, Tao Gong, and Hongying Meng. StrongSORT: Make DeepSORT great again. *IEEE Transactions on Multimedia*, pages 1–14, 2023. 1, 12

[13] Felix Fent, Philipp Bauerschmidt, and Markus Lienkamp. RadarGNN: Transformation invariant graph neural network for radar-based perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 182–191, 2023. 2

[14] Xiangyu Gao, Guanbin Xing, Sumit Roy, and Hui Liu. RAMP-CNN: A novel neural network for enhanced automotive radar object recognition. *IEEE Sensors Journal*, 21(4): 5119–5132, 2021. 1, 2

[15] Ross Girshick. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015. 6

[16] Karl Granström and Marcus Baum. Extended Object Tracking: Introduction, overview and applications. *CoRR*, abs/1604.00970, 2016. 2

[17] Karl Granstrom, Maryam Fatemi, and Lennart Svensson. Poisson multi-bernoulli mixture conjugate prior for multiple extended target filtering. *IEEE Transactions on Aerospace and Electronic Systems*, 56(1):208–225, 2020. 2

[18] Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, and P-J Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on signal processing*, 50(2): 425–437, 2002. 2

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 2

[20] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, 2018. 12

[21] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3463–3472, 2019. 12

[22] Texas Instruments. Short range radar reference design using awr1642 (rev. b), 2018. 25

[23] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, pages 182–193. International Society for Optics and Photonics, 1997. 2

[24] Rudolf Emil Kalman et al. Contributions to the theory of optimal control. *Bol. soc. mat. mexicana*, 5(2):102–119, 1960. 1

[25] Johann Wolfgang Koch. Bayesian approach to extended object and cluster tracking using random matrices. *IEEE Transactions on Aerospace and Electronic Systems*, 44(3):1042–1059, 2008. 2

[26] Jian Li and Petre Stoica. *MIMO Radar Signal Processing*. John Wiley & Sons, 2008. 26

[27] Peizhao Li, Pu Wang, Karl Berntorp, and Hongfu Liu. Exploiting temporal relations on radar perception for autonomous driving. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17050–17059, 2022. 1, 2, 3, 4, 6, 7, 12, 13, 17, 19, 26

[28] Yu-Jhe Li, Shawn Hunt, Jinhyung Park, Matthew O'Toole, and Kris Kitani. Azimuth super-resolution for FMCW radar in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17504–17513, 2023. 1, 2

[29] 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

[30] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017. 6

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021. 3

[32] Jonathon Luiten, Aljosa Osep, Patrick Dendorfer, Philip H. S. Torr, Andreas Geiger, Laura Leal-Taixé, and Bastian Leibe. HOTA: A higher order metric for evaluating multi-object tracking. *Int. J. Comput. Vision*, 129(2):548–578, 2021. 7, 16

[33] Gerard Maggiolino, Adnan Ahmad, Jinkun Cao, and Kris Kitani. Deep OC-SORT: Multi-pedestrian tracking by adaptive re-identification. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 3025–3029, 2023. 12

[34] Yunze Man, Liang-Yan Gui, and Yu-Xiong Wang. Bev-guided multi-modality fusion for driving perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21960–21969, 2023. 2

[35] Anton Milan, Laura Leal-Taixe, Ian Reid, Stefan Roth, and Konrad Schindler. MOT16: A benchmark for multi-object tracking, 2016. 7, 16, 17, 18

[36] Mohammadreza Mostajabi, Ching Ming Wang, Darsh Ranjan, and Gilbert Hsyu. High resolution radar dataset for semi-supervised learning of dynamic objects. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 450–457, 2020. 12

[37] Umut Orguner. A variational measurement update for extended target tracking with random matrices. *IEEE Transactions on Signal Processing*, 60(7):3827–3834, 2012. 2

[38] Arthur Ouaknine, Alasdair Newson, Patrick Pérez, Florence Tupin, and Julien Rebut. Multi-view radar semantic segmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15651–15660, 2021. 1, 2

[39] Andras Palffy, Ewoud Pool, Srimannarayana Baratam, Julian F. P. Kooij, and Dariu M. Gavrila. Multi-class road user detection with 3+1d radar in the view-of-delft dataset. *IEEE Robotics and Automation Letters*, 7(2):4961–4968, 2022. 1, 2

[40] Ashish Pandharipande, Chih-Hong Cheng, Justin Dauwels, Sevgi Z. Gurbuz, Javier Ibanez-Guzman, Guofa Li, Andrea Piazzoni, Pu Wang, and Avik Santra. Sensing and machine learning for automotive perception: A review. *IEEE Sensors Journal*, 23(11):11097–11115, 2023. 1, 2

[41] Jiangmiao Pang, Linlu Qiu, Xia Li, Haofeng Chen, Qi Li, Trevor Darrell, and Fisher Yu. Quasi-dense similarity learning for multiple object tracking. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 164–173, 2021. 12

[42] Kun Qian, Shilin Zhu, Xinyu Zhang, and Li Erran Li. Robust multimodal vehicle detection in foggy weather using complementary lidar and radar signals. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 444–453, 2021. 2

[43] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), 2020. 12

[44] 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. 26

[45] Julien Rebut, Arthur Ouaknine, Waqas Malik, and Patrick Pérez. Raw high-definition radar for multi-task learning. In *Proceedings of the IEEE/CVF CVPR*, pages 17021–17030, 2022. 12

[46] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized Intersection Over Union: A metric and a loss for bounding box regression. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 658–666, 2019. 5

[47] Marcel Sheeny, Emanuele De Pellegrin, Saptarshi Mukherjee, Alireza Ahrabian, Sen Wang, and Andrew Wallace. RADIATE: A radar dataset for automotive perception in bad weather. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–7, 2021. 1, 2, 6, 7, 12, 13, 17, 25, 26

[48] Gerald L Smith, Stanley F Schmidt, and Leonard A McGee. *Application of statistical filter theory to the optimal estimation of position and velocity on board a circumlunar vehicle*. National Aeronautics and Space Administration, 1962. 2

[49] Niklas Wahlstrom and Emre Ozkan. Extended target tracking using gaussian processes. *IEEE Transactions on Signal Processing*, 63(16):4165–4178, 2015. 2

[50] Pu Wang, Petros T. Boufounos, Hassan Mansour, and Philip V. Orlik. Slow-time MIMO-FMCW automotive radar detection with imperfect waveform separation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 8634–8638. IEEE, 2020. 25, 26

[51] Yingjie Wang, Jiajun Deng, Yao Li, Jinshui Hu, Cong Liu, Yu Zhang, Jianmin Ji, Wanli Ouyang, and Yanyong Zhang. Bi-LRFusion: Bi-directional lidar-radar fusion for 3d dynamic object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13394–13403, 2023. 2

[52] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017. 12

[53] 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

[54] Tetsutaro Yamada, Masato Gocho, Kei Akama, Ryoma Yataka, and Hiroshi Kameda. Multiple hypothesis tracking with merged bounding box measurements considering occlusion. *IEICE Transactions on Information and Systems*, E105.D(8):1456–1463, 2022. 12

[55] Bin Yang, Runsheng Guo, Ming Liang, Sergio Casas, and Raquel Urtasun. RadarNet: Exploiting radar for robust perception of dynamic objects. In *Computer Vision – ECCV 2020*, pages 496–512, 2020. 2

[56] Ryoma Yataka, Pu Wang, Petros Boufounos, and Ryuhei Takahashi. Radar perception with scalable connective temporal relations for autonomous driving. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13266–13270, 2024. 16

[57] 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 *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2149–2158, 2021. 6

[58] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11779–11788, 2021. 2

[59] Shuqing Zeng and James N. Nickolaou. *Automotive Radar*. CRC Press, 2014. 1

[60] Ao Zhang, Farzan Erlik Nowruzi, and Robert Laganiere. RADDet: Range-azimuth-doppler based radar object detection for dynamic road users. In *2021 18th Conference on Robots and Vision (CRV)*, pages 95–102, 2021. 1, 2

[61] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. FairMOT: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vision*, 129(11):3069–3087, 2021. 12

[62] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. ByteTrack: Multi-object tracking by associating every detection box. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, page 1–21. Springer-Verlag, 2022. 1, 12

[63] Lianqing Zheng, Zhixiong Ma, Xichan Zhu, Bin Tan, Sen Li, Kai Long, Weiqi Sun, Sihan Chen, Lu Zhang, Mengyue Wan, Libo Huang, and Jie Bai. Tj4dradset: A 4d radar dataset for autonomous driving. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 493–498, 2022. 12

[64] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *CoRR*, abs/1904.07850, 2019. 3, 6

[65] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV*, page 474–490. Springer-Verlag, 2020. 2, 7

SIRA: Scalable Inter-frame Relation and Association for Radar Perception

Supplementary Material

6. Related Work for Visual Tracking

In recent years, KF-based approaches have gained popularity in the context of visual tracking, and various extensions have been proposed [7, 8, 12, 33, 41, 52, 54, 61, 62], exemplified by SORT [4]. SORT can achieve high tracking performance, but it relies on the assumption that objects have consistent linear motion in a short time, which requires continuous observations. Therefore, it can face challenges when objects exhibit occlusion or nonlinear motion, requiring a high frame rate. To overcome occlusion problems, ByteTrack [62] uses the similarity between tracklets and low-scoring detection boxes to recover the true objects and filter out background detections. OC-SORT [8] introduces object motion computed from pre- and post-occlusion time pairs to address occlusion and non-linear motion. Our proposed framework extends recent KF-based methods and learning-based approaches by assuming high-density radar detection points. It explicitly considers strong object-level consistency by using multiple frames to capture the nonlinear motion of objects.

7. Related Work for Radar Datasets

If we categorize open radar datasets into the ones with sparse detection points, dense points and low-level heatmap, RADIATE[47] is the largest dense-point dataset with bounding box and tracking ID labels for both detection and tracking as shown in Table 4. We will use these datasets for further evaluation in the future.

8. TempoRadar [27]

Our ETR generalizes TempoRadar [27] into a long time horizon and shares several key building blocks such as the top-K feature selector $\mathcal{S}_K$ and the design of the (temporal) masking matrix $\mathbf{M}$ in the masked multi-head attention (MCA).

Top-K Feature Selector $\mathcal{S}_K$: To exploit the feature-level temporal relation, TempoRadar introduces a temporal relation layer (TRL). Given the extracted features $\mathbf{Z}_t := \mathcal{F}_\theta\left(I_{t,t-1}\right)$ and $\mathbf{Z}_{t-1} := \mathcal{F}_\theta\left(I_{t-1,t}\right)$ from the encoder, where $I_{t-1,t}$ concatenates two consecutive radar frames I_{t-1} and I_t along the channel dimension in the order of $(t-1,t)$, the feature selector $\mathcal{S}_K$ of (3) is defined as:

$$\mathbf{H}_t = \mathcal{S}_K\left(\mathbf{Z}_t\right) := \mathbf{Z}_t\left[P_t^{\text{pre-hm}}\right],$$

$$\mathbf{H}_{t-1} = \mathcal{S}_K\left(\mathbf{Z}_{t-1}\right) := \mathbf{Z}_{t-1}\left[P_{t-1}^{\text{pre-hm}}\right],$$

Table 4. A list of open radar datasets in the format of dense points.

Dataset	# of data	Radar format	BBox	Tracking ID
RADIATE [47]	44K	dense points	2D	✓
Zendar [36]	4.8K	dense points	2D	✓
TJ4DRadSet [63]	7.7K	dense points	2D	✓
RADIal [45]	25K	heatmap+points	2D	

where $\mathbf{H}_{t/t-1} \in \mathbb{R}^{C\times K}$ and $P_t^{\text{pre-hm}}$ is defined as the set of (x,y) coordinates corresponding to the K selected features,

$$P_t^{\text{pre-hm}} := \left\{(x,y) \,\middle|\, \{\mathbf{C}_t\}_{xy} \geq \{\mathbf{C}_t\}_K\right\}, \tag{18}$$

where $\mathbf{C}_t = \mathcal{G}_\theta^{\text{pre-hm}}\left(\mathbf{Z}_t\right)$ maps the channel dimension of the feature map via a learnable feedforward neural network (FNN) module $\mathcal{G}_\theta^{\text{pre-hm}} : \mathbb{R}^{C\times\frac{H}{s}\times\frac{W}{s}} \rightarrow \mathbb{R}^{1\times\frac{H}{s}\times\frac{W}{s}}$ into a scalar feature map for feature ranking, $\{\mathbf{C}_t\}_K$ stands for the K-th largest value in $\mathbf{C}_t$ over the spatial space $\frac{H}{s}\times\frac{W}{s}$, and the subscript xy takes value at the coordinate (x,y).

Design of $\mathbf{M}$ in Masked MCA: Let us stack the top-K selected features from the two consecutive radar frames as $\mathbf{H}_{t,t-1} := \{\mathbf{H}_t, \mathbf{H}_{t-1}\}^\top \in \mathbb{R}^{2K\times C}$. The masked MCA takes $\mathbf{H}_{t,t-1}$ and applies cross-frame attention over the two sets of features, as shown in Fig. 7a.

Since the position is lost in $\mathbf{H}_{t,t-1}$, we generate the position information of the selected top-K features via a learnable positional encoding network $\mathcal{E}_\theta$ from the coordinate set $P_t^{\text{pre-hm}}$ of (18)

$$\mathbf{P}_t^{\text{enc}} = \mathcal{E}_\theta\left(P_t^{\text{pre-hm}}\right) \in \mathbb{R}^{K\times D_{\text{pos}}},$$

where D_{pos} is the dimension of positional encoding. We then supplement the positional encoding into feature vectors

$$\mathbf{H}_{t,t-1}^{\text{pos}} = \left\{\mathbf{H}_{t,t-1}, \mathbf{P}_{t,t-1}^{\text{enc}}\right\} \in \mathbb{R}^{2K\times(C+D_{\text{pos}})},$$

where $\mathbf{P}_{t,t-1}^{\text{enc}} = \left\{\mathbf{P}_t^{\text{enc}}, \mathbf{P}_{t-1}^{\text{enc}}\right\}^\top \in \mathbb{R}^{2K\times D_{\text{pos}}}$, and pass it to the masked MCA for temporal attention.

In computing the temporal relation, the masked MCA follows [2, 20, 21, 43] and uses a temporal inductive bias with a masking matrix $\mathbf{M}$

$$\mathcal{A}\left(\mathbf{V},\mathbf{X}\right) := \text{ softmax}\left(\frac{\mathbf{M} + q\left(\mathbf{X}\right)k\left(\mathbf{X}\right)^\top}{\sqrt{d}}\right)v\left(\mathbf{V}\right),$$

where $q\left(\cdot\right)$, $k\left(\cdot\right)$ and $v\left(\cdot\right)$ are linear transformation layers and are referred to as query, keys and values, respectively, and d is the dimension of the query and the keys. For the temporal attention over $\{t, t-1\}$, we have $\mathcal{A}\{\mathbf{H}_{t,t-1}, \mathbf{H}^{\text{pos}}_{t,t-1}\}$ with $\mathbf{V} = \mathbf{H}_{t,t-1}$ for the value and $\mathbf{X} = \mathbf{H}^{\text{pos}}_{t,t-1}$ for the key and query. The masking matrix $\mathbf{M}$ is given as

$$\mathbf{M} := \begin{bmatrix} \mathbb{I}_K, \mathbf{1}_K \\ \mathbf{1}_K, \mathbb{I}_K \end{bmatrix} + \sigma\left(\begin{bmatrix} \mathbf{1}_K, \mathbf{0}_K \\ \mathbf{0}_K, \mathbf{1}_K \end{bmatrix} - \mathbb{I}_{2K}\right), \quad (19)$$

where $\mathbb{I}_K$ is the identity matrix of size K, $\mathbf{1}_K$ and $\mathbf{0}_K$ are the all-one and all-zero matrix with size $K \times K$, respectively, and σ is a large negative constant, e.g., -10^{10}, to guarantee a near-zero value in the output through the softmax function. It can be shown that diagonal blocks in $\mathbf{M}$ disable attention between features within the same frame, while off-diagonal blocks allow for cross-frame attention. The masked MCA may repeat multiple times.

9. Details of BBox Loss

We pick the object's center coordinates from the heatmap, and learn its attributes from feature representations through regression. Regression functions, which are heatmap loss $\mathcal{L}^{\text{h}}_t$, width & Length loss $\mathcal{L}^{\text{b}}_t$, orientation loss $\mathcal{L}^{\text{r}}_t$, and offset loss $\mathcal{L}^{\text{o}}_t$, compose the training objective by a linear combination as (15):

$$\mathcal{L}^{\text{BBox}}_t = \frac{1}{N_{\text{gt}}} \sum_{k=1}^{N_{\text{gt}}} \left(\mathcal{L}^{\text{b}}_{t,k} + \mathcal{L}^{\text{r}}_{t,k} + \mathcal{L}^{\text{o}}_{t,k}\right) - \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}^{\text{h}}_{t,i},$$

where N denotes the total number of pixels in the heatmap and N_{gt} is the total number of ground truth bounding boxes. Each loss is as follows:

$$\begin{aligned} \mathcal{L}^{\text{h}}_{t,i} =& \mathbb{1}_{\{c_{t,i}=1\}} \left(1 - \widehat{c}_{t,i}\right)^{\alpha} \log\left(\widehat{c}_{t,i}\right) \\ &+ \mathbb{1}_{\{c_{t,i} \neq 1\}} \left(1 - c_{t,i}\right)^{\beta} \widehat{c}^{\,\alpha}_{t,i} \log\left(1 - \widehat{c}_{t,i}\right), \end{aligned} \quad (20)$$

where $c_{t,i}$ and $\widehat{c}_{t,i}$ denote the ground-truth and predicted value at i-th coordinate in $\mathcal{G}^{\text{hm}}_{\theta}\left(\mathbf{Z}^{\text{hm}}_t\right)$, and α and β are hyper-parameters and are chosen empirically with 2 and 4, respectively.

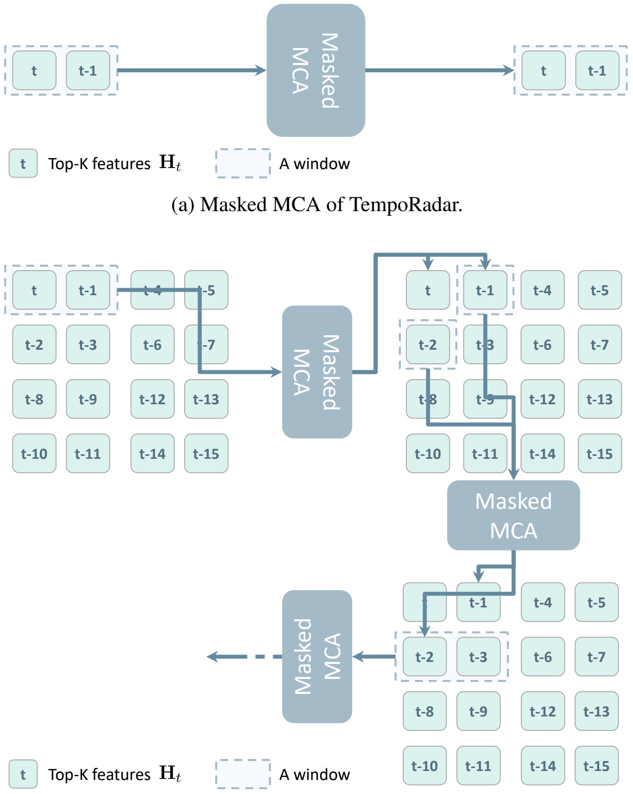
$$\mathcal{L}^{\text{b}}_{t,k} = S_{L_1}\left(\left\| \mathcal{G}^{b}_{\theta}\left(\mathbf{Z}_t\left[P^{\text{gt}}_{t,k}\right]\right) - \left(w_{t,k}, h_{t,k}\right)^{\top} \right\|\right), \quad (21)$$

$$\mathcal{L}^{\text{r}}_{t,k} = S_{L_1}\left(\left\| \mathcal{G}^{r}_{\theta}\left(\mathbf{Z}_t\left[P^{\text{gt}}_{t,k}\right]\right) - \left(\cos\vartheta_{t,k}, \sin\vartheta_{t,k}\right)^{\top} \right\|\right), \quad (22)$$

$$\mathcal{L}^{\text{o}}_{t,k} = S_{L_1}\left(\left\| \mathcal{G}^{o}_{\theta}\left(\mathbf{Z}_t\left[P^{\text{gt}}_{t,k}\right]\right) - \left(o_{x,t,k}, o_{y,t,k}\right)^{\top} \right\|\right), \quad (23)$$

where $P^{\text{gt}}_{t,k}$ denotes the coordinate $(x_{t,k}, y_{t,k})$ of the center of k-th ground truth object, $(w_{t,k}, h_{t,k})$ is the width & length, and $(o_{x,t,k}, o_{y,t,k})$ is the offset as follows:

$$\left(o_{x,t,k}, o_{y,t,k}\right) = \left(\frac{x_{t,k}}{s} - \left\lfloor \frac{x_{t,k}}{s} \right\rfloor, \frac{y_{t,k}}{s} - \left\lfloor \frac{y_{t,k}}{s} \right\rfloor\right). \quad (24)$$



(a) Masked MCA of TempoRadar.

(b) Masked MCA of SeTR.

Figure 7. Masked MCA. (a) TempoRadar [27] computes masked multi-head cross-attention (MCA) over the top-K selected features from a time horizon of only $T = 2$ consecutive radar frames. (b) SeTR computes masked MCA for two consecutive radar frames at a time, the same as the TempoRadar in (a), but slides the window of two frames after each MCA sequentially to cover a longer time horizon of $T > 2$ frames.

10. Sequential TempoRadar (SeTR)

One might postulate: "What are the implications of extending *TempoRadar* to cover more consecutive radar frames?" The answer might be two-fold. On the one hand, one should expect improved performance under the assumption that most radar features are present over more than just 2 frames, considering a typical radar frame rate of $>$ 4 fps (*Radiate* dataset has 4 fps [47]). On the other hand, directly applying temporal attention to a longer time horizon incurs a quadratic computation complexity (refer to (8)) over the number of features from each frame K and the number of frames T.

One straightforward way for a scalable TempoRadar is to stack temporal feature attention for two consecutive frames and sequentially connect them, which we refer to as *sequential TempoRadar* (SeTR). As illustrated in Fig. 7b, SeTR computes masked MCA for two consecutive radar frames at a time, the same as the TempoRadar in Fig. 7a, but slides the window of two frames after each MCA sequentially to cover a longer time horizon of $T > 2$ frames.

11. Training and Inference Pipelines for SIRA

Training Pipeline for SIRA: To train SIRA, we takes T consecutive radar frames, pass them into the training pipeline in the Top diagram of Fig. 8, and compute the loss function $\mathcal{L}^{\text{BBox}}$ of (15) at the decoder output for detection loss and the pseudo-direction loss $\mathcal{L}^{\text{DEst}}$ of (16) at the output of the DEst module (detailed in **Motion Consistency for Training** of Section 3.3). Through backpropagation, the learnable modules, hatched in light green, are updated using the derived loss value.

Inference Pipeline for SIRA: In the bottom of Fig. 8, we show the inference pipeline for SIRA. Noticeably, a tracker is attached to the DEst module to further enforce the motion consistency via the concept of pseudo-tracklet, detailed in **Motion Consistency for Inference** of Section 3.3. All learnable parameters during training are frozen in the inference. We further include the pseudo-code of the inference pipeline in Algorithm 1. A typical tracker consists of five steps: Prediction, Association, Update, Deletion, and Initialization. We can integrate our MCTrack with standard trackers (e.g., OC-SORT) by incorporating the key components (highlighted in **green** in Algorithm 1), e.g., the use of motion similarity of (11), to the Association step.

Extension with higher-order KF: As shown in Fig. 9, we expect SIRA can deal with nonlinear motion to an extent as the average predicted angle $\widehat{\phi}_{\text{ave}}$ can correct the KF predicted state $\widehat{\mathbf{x}}_{T|T-1}$ closer to the right observation $\mathbf{z}_T$. SIRA can be extended with higher-order KF (e.g., extended/unscented KF) to further improve the predicted state $\widehat{\mathbf{x}}_{T|T-1}$ with a proper nonlinear model and the average predicted angle $\widehat{\phi}_{\text{ave}}$, yielding improved trajectory predictions.

The choice of patch size: Since the patch is a subset of Top-K features in each frame, we have the patch size $M \in [1, K]$. Given the number of frames U in each window, the smaller the patch size M, the smaller the window size UM in the re-grouping stage of the TRWA block (see the top right of Fig. 3 for an example of $U = 4$ frames and $M = K/2$), and the lower the computational complexity of the window-based attention which is quadratic with respect to UM. On the other hand, a small M may limit the number of features to be correlated across windows and reduces the connectivity of temporal attention.

Algorithm 1: Pseudo-code of SIRA for Inference.

```
Input: A radar frame sequence V; encoder Enc; decoder Dec;
       object detector ETR; direction estimator DEst; detection
       score threshold $\gamma$; birth threshold $\beta$
Output: Tracks $\mathcal{T}$ of the video
Initialization: $\mathcal{T} \leftarrow \emptyset$
for frame $f_k$ in V do
    /* Fig.2, and Fig.8 */
    /* **predict bboxes with ETR** */
    $\mathcal{F}_k \leftarrow$ Enc($f_k$)
    $\mathcal{F}_k \leftarrow$ ETR($\mathcal{F}_k$)
    $\mathcal{D}_k \leftarrow$ Dec($\mathcal{F}_k$)
    /* **tracking with MCTrack** */
    $\mathcal{J}_k \leftarrow$ DEst($\mathcal{F}_k$)
    $\mathcal{D}_{high} \leftarrow \emptyset$
    $\mathcal{J}_{high} \leftarrow \emptyset$
    for $d, j$ in $\mathcal{D}_k, \mathcal{J}_k$ do
        if $d.score > \gamma$ then
            $\mathcal{D}_{high} \leftarrow \mathcal{D}_{high} \cup \{d\}$
            $\mathcal{J}_{high} \leftarrow \mathcal{J}_{high} \cup \{j\}$
        end
    end
    /* predict new locations of tracks */
    for $t$ in $\mathcal{T}$ do
        $t \leftarrow$ KalmanFilter.predict($t$)
    end
    /* Fig.5 */
    /* association */
    Associate $\mathcal{T}$ and $\mathcal{D}_{high}$&$\mathcal{J}_{high}$ with Similarity Eq.11
    $\mathcal{D}_{remain} \leftarrow$ remaining unmatched object from $\mathcal{D}_{high}$
    $\mathcal{T}_{remain} \leftarrow$ remaining matched tracks from $\mathcal{T}$
    /* update status of matched tracks */
    for $t$ in $\mathcal{T}_{remain}$ do
        $t \leftarrow$ KalmanFilter.update($t$)
    end
    /* delete unmatched tracks */
    $\mathcal{T} \leftarrow \mathcal{T} \setminus \mathcal{T}_{remain}$
    /* initialize new tracks */
    for $d$ in $\mathcal{D}_{remain}$ do
        if $d.score > \beta$ then
            $\mathcal{T} \leftarrow \mathcal{T} \cup \{d\}$
        end
    end
end
Return: $\mathcal{T}$
```

In **green** is the key of our method.

12. Details of Experimental Settings

Dataset To facilitate the research on robust and reliable vehicle perception, *Radiate* dataset was collected in 7 scenarios under various weather and lighting conditions: Sunny (Parked), Sunny/Overcast (Urban), Overcast (Motorway), Night (Motorway), Rain (Suburban), Fog (Suburban) and Snow (Suburban). It includes multiple sensor modalities from radar and optical images to 3D LiDAR point clouds and GPS. 8 object classes, i.e., car, van, truck, bus, motorbike, bicycle, pedestrian and group of pedestrian, were annotated on the radar frames. The data for-

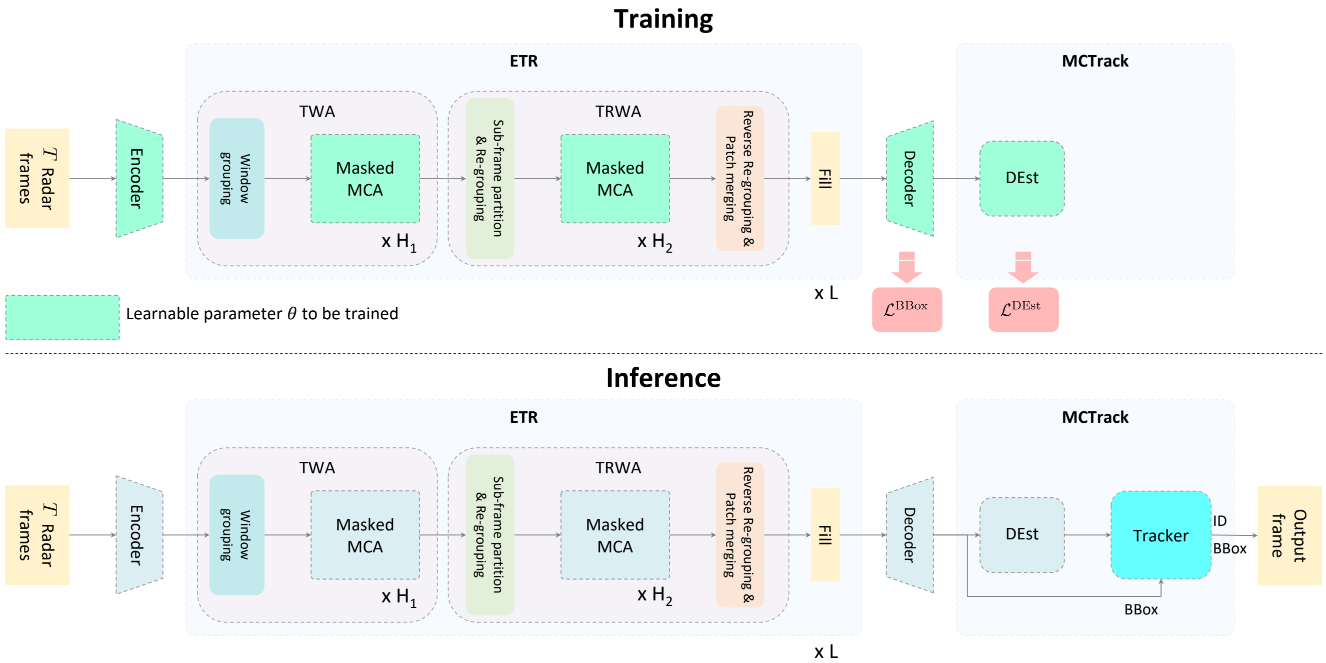


Figure 8. Training and Inference Pipelines for SIRA. (Top) SIRA takes T consecutive radar frames into the training pipeline, and computes $\mathcal{L}^{\text{BBox}}$ at the decoder and the pseudo-direction loss $\mathcal{L}^{\text{DEst}}$ at the DEst module in **Motion Consistency for Training** of Section 3.3. Learnable modules are hatched in light green. (Bottom) SIRA attaches a tracker at the DEst module output to further enforce **Motion Consistency for Inference** of Section 3.3. The tracker incorporates the motion similarity of (11) for association. Learnable parameters are frozen during inference.

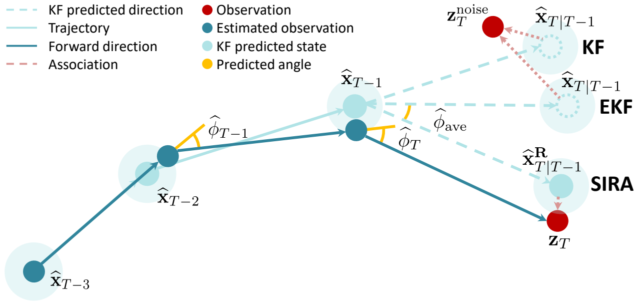

KF predicted direction
Trajectory
Forward direction
Association
Observation
Estimated observation
KF predicted state
Predicted angle
$\mathbf{z}_T^{\text{noise}}$
$\widehat{\mathbf{x}}_{T|T-1}$
KF
$\widehat{\mathbf{x}}_{T-1}$
$\widehat{\mathbf{x}}_{T|T-1}$
EKF
$\widehat{\phi}_{T-1}$
$\widehat{\phi}_{\text{ave}}$
$\widehat{\phi}_T$
$\widehat{\mathbf{x}}^{\text{R}}_{T|T-1}$
SIRA
$\mathbf{z}_T$
$\widehat{\mathbf{x}}_{T-2}$
$\widehat{\mathbf{x}}_{T-3}$

Figure 9. Trajectory prediction of SIRA with KF and EKF.

mat of radar frames generated from dense point clouds, where the pixel values indicate radar reflection magnitude. Radiate adopted the Navtech CTS350-X FMCW radar, a scanning radar that provides 360° high-resolution range-azimuth BEV images at 4 Hz. It was set to have 100-meter maximum operating range with a distance resolution of 0.175 m, an azimuth resolution of 1.8° and an elevation resolution of 1.8°. It does not provide Doppler information. Radar frames in Cartesian are provided as .png at 1152 × 1152 resolution. Nearest neighbour interpolation was used to convert the radar framess from the polar coordinate to the Cartesian one. Each pixel in the Cartesian coordinate represents a grid of $0.17361 \times 0.17361\text{m}^2$. In other words, the field of view is about $[-100\text{m}, 100\text{m}]$ in one axis and $[-100\text{m}, 100\text{m}]$ in the other axis in BEV. Radiate dataset has official 3 splits: "train in good weather" which consists of 31 sequences (22383 frames, only in good weather, sunny or overcast), "train good & bad weather" which consists of 12 sequences (9749 frames, both good & bad weather conditions), and "test" which consists of 18 sequences (11305 frames, all kinds of weather conditions). Fig. 10 shows sampled RGB and corresponding radar frames under adverse weather and low lighting conditions. We separately train models on the former two training sets and evaluate on the test set.

Hyper-parameters The hyper-parameters used in our experiments of Section 4 are shown in Table 5. The table is divided into three parts, Data, Architecture, and Training, each with parameter names, notations, and values.

13. Comparison of Complexity Analysis

Fig. 11 compares the computational complexity of TempoRadar in (8) and ETR in (9) as a function of the number of consecutive radar frames T, under two settings of the number of selected features $K = 8$ and $K = 16$ (grouped in

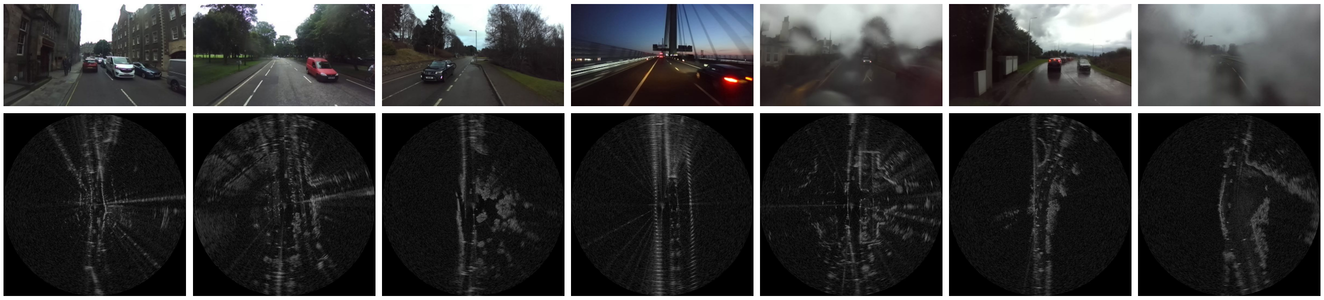


Figure 10. Visualization of RGB and corresponding radar frames. From left to right, the scenes are from City-3-7, City-7-0, Junction-1-10, Night-1-4, Fog-6-0, Rain-4-0 and Snow-1-0 in *Radiate*. Albeit of more coarse-grained and less semantic features, radar frames are much more resilient than RGB frames in adverse weather and low lighting conditions.

Table 5. Hyper-parameters used in our experiments.

	Name	Notation	Value
Data	dataset	-	*Radiate*
	train good weather	-	22383
	train good & bad weather	-	9749
	test	-	11305
	cropped image size	$H \times W$	256×256
	full image size	$H \times W$	1152×1152
Architecture	position dimention	D_{pos}	64
	downsampling ratio	s	4
	# of top-Ks	K	8
	# of sets of top-K	U	2
	# of ETR stages	L	1
	# of masked MCAs: TWA	H_1	2
	# of masked MCAs: TRWA	H_2	2
	operation	$\mathcal{M}$	max
	coefficient	λ	0.5
	detection score threshold	γ	0.08
	birth threshold	β	0.20
Training	batch size	-	16
	epoch	-	10
	optimizer	-	Adam
	learning rate	-	5e-4
	schedule for train good weather $\times 0.1$	-	5
	schedule for train good & bad weather $\times 0.1$	-	2
	weight decay for detection	-	1e-2
	weight decay for tracking	-	1e-5
	# of GPUs	-	1

two different colors). For each setting of K, we further include four ETR variants (denoted by different markers) with different combinations of hyper-parameters of the number of consecutive radar frames within one temporal window U and the number of features within one patch M. Under both settings, ETR provides more affordable temporal over longer time horizons T than TempoRadar.

While (9) represents the complexity of the general ETR module, [56] presents a special case of ETR with $U = 2$, $M = K/2$ and a stride $K/4$ (a special sub-frame partition with 50% overlapping). For this special case, the computational complexity is shown to be $2K^2\left(3T - 4\right)L$.

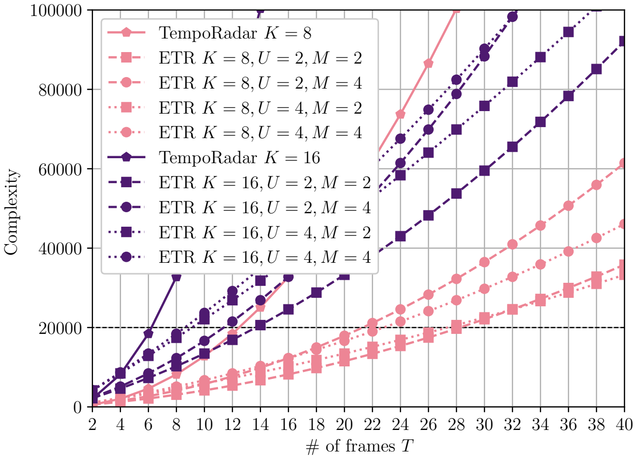


Figure 11. Comparison of computational complexities of TempoRadar (solid) and ETR (dashed) as a function of the number of frames T (along the x-axis) and the number of selected features K (grouped in different colors).

14. Definition of MOT Metrics

We adopt the series of MOT metrics [32, 35] for evaluation. We pick several key metrics in the experiments: MOTA (Multiple Object Tracking Accuracy), IDF1, ID switch (IDs), track fragmentations (Frag.), mostly tracked (MT), and partially tracked (PT). The MOTA score is calculated by

$$\text{MOTA} = 1 - \frac{\sum_t (\text{FN}_t + \text{FP}_t + \text{IDSW}_t)}{\sum_t \text{GT}_t},$$

where t is the frame index, GT is the number of ground-truth objects, and FN and FP refer to false negative and false positive detection, respectively. The value of MOTA is in the range $(-\infty, 100]$. It can be deemed as the combination of detection and tracking performance, and is widely used as the main metric for accessing multiple object tracking quality.

Table 6. Additional results of object detection on *Radiate* for mAP@0.7. The number following the model name indicates the # of layers in the ResNet, and the number in parentheses indicates the # of frames T.

mAP@0.7	**Train good weather**	**Train good & bad weather**
RetinaNet-18 (1)	8.46±0.61	6.97±1.24
CenterPoint-18 (1)	19.02±1.80	14.43±2.56
BBAVectors-18 (1)	19.72±1.10	15.07±1.76
TR-18 (2)	20.57±1.47	15.59±2.31
TR-18 (4)	19.59±0.78	19.62±1.33
SeTR-18 (4)	21.90±1.12	19.65±0.84
SIRA-18 (4)	**21.95**±1.72	**19.66**±1.87
RetinaNet-34 (1)	7.67±1.71	6.93±1.60
CenterPoint-34 (1)	18.93±1.46	13.43±1.92
BBAVectors-34 (1)	19.86±1.36	14.67±1.45
TR-34 (2)	21.08±1.66	14.35±2.15
TR-34 (4)	22.46±1.76	19.03±1.10
SeTR-34 (4)	21.68±1.24	19.63±1.29
SIRA-34 (4)	**22.81**±0.86	**19.85**±0.95

Table 7. Comparison on object detection with full size images. Comparison on object detection with full size images.

Train good weather	mAP@0.3	mAP@0.5
FasterRCNN-50 (1) [47]	-	45.31
FasterRCNN-101 (1) [47]	-	45.84
TR-18 (2) [27]	-	48.02
TR-34 (2) [27]	-	48.66
ETR-34 (4)	65.10	49.19
SIRA-34 (4)	**65.67**	**51.49**
ETR-34 (6)	67.19	49.37
SIRA-34 (6)	**67.72**	**52.14**
ETR-34 (8)	65.53	50.59
SIRA-34 (8)	**67.82**	**52.55**
ETR-34 (10)	64.24	50.12
SIRA-34 (10)	**66.03**	**50.77**

IDF1 evaluates the identity preservation ability and focuses on the association performance. Specifically, IDF1 calculates a bijective (one-to-one) mapping between the sets of ground truth trajectories and predicted trajectories (unlike MOTA at the detection level) and is a function of

- IDTPs (identity true positives): the matches in the overlapping sections of trajectories that are correctly associated with the same identity;
- IDFNs (identity false negatives): instances where the ground truth has an identity that the prediction fails to identify. This often occurs in non-overlapping sections of matched trajectories or when the tracker loses track of an object;
- IDFPs (identity false positives): instances where the prediction assigns an identity that does not exist in the ground truth. This often happens in the case of over-segmentation or incorrect identity assignments;

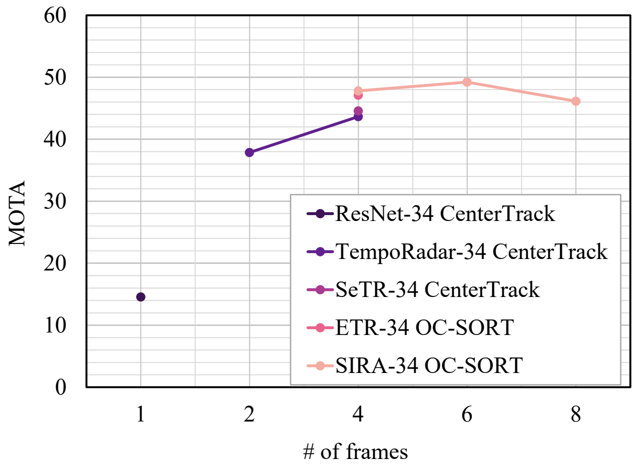


Figure 12. Tracking performance as a function of number of frames T. Compared with the single-frame baseline (ResNet-34 CenterTrack), SIRA with $T = 6$ consecutive frames results in a margin of +34.67 MOTA.

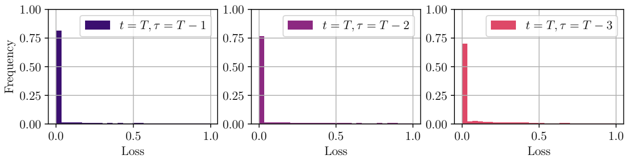


Figure 13. Pseudo-direction smooth L_1 loss in different time steps.

$$\mathrm{IDF1} = \frac{2\mathrm{IDTP}}{2\mathrm{IDTP} + \mathrm{IDFP} + \mathrm{IDFN}}. \tag{25}$$

The rest of these metrics all reflect the quality of predicted tracklets. For detailed definitions and calculations of MOT metrics, please refer to [35].

15. Additional Ablation Study

We present supplementary experimental results from our ablation studies. Each experimental setting aligns with the conditions detailed in Section 4.

Detection Results at mAP@0.7: For Table 1 in Section 4.2, additional results for mAP@0.7 are shown in Table 6. Compared to mAP@0.3 and mAP@0.5, all mAP@0.7 values are lower as expected when the IoU threshold increases. It is seen that SIRA provides consistently better detection performance than the baseline methods.

Detection Results With Full Size Radar Frames: We keep the original resolution with full size 1152 × 1152 to make a fair comparison to the results from [47]. Regarding variations in image size, a marginal decline in detection

Table 8. Experimental results of multiple object tracking on *Radiate*. The number following the model name indicates the # of layers in the Resnet backbone, and the number in parentheses indicates the # of frames T.

Train good weather	MOTA↑	MOTP↑	IDF1↑	IDs↓	FP↓	FN↓	Frag.↓	MT↑	ML↓	PT↑
ResNet-18 (1) CenterTrack	13.01	70.26	-	873	-	-	920	269	-	254
ResNet-34 (1) CenterTrack	14.55	70.05	-	802	-	-	831	282	-	279
TR-18 (2) CenterTrack	33.59	73.49	-	349	-	-	498	145	-	330
TR-34 (2) CenterTrack	37.85	71.85	39.90	457	970	6114	511	108	422	246
TR-18 (4) CenterTrack	42.77	70.38	44.91	519	1061	5206	520	244	196	336
TR-34 (4) CenterTrack	43.64	71.58	44.17	503	854	5892	538	197	253	326
SeTR-18 (4) CenterTrack	42.11	68.71	50.33	658	1481	4672	561	261	198	317
SeTR-34 (4) CenterTrack	44.57	71.65	48.72	875	1511	4606	602	348	129	299
ETR-34 (4) CenterTrack	46.06	70.23	50.81	1832	1141	4904	613	345	126	305
ETR-34 (4) OC-SORT	47.11	70.08	50.04	540	1411	4523	481	343	120	313
SIRA-34 (4) CenterTrack[*]	47.30	70.19	50.16	1249	1218	4756	566	354	122	300
SIRA-34 (4) OC-SORT	47.79	70.09	51.13	523	1408	4513	488	342	120	314

[*] For CenterTrack, C^{tracklet} is only used for association since this tracker is not based on SORT.

Table 9. Ablation study of various number of frames for multiple object tracking on train good weather. We used SIRA-34 OC-SORT.

# of frames T	MOTA↑	MOTP↑	IDF1↑	IDs↓	FP↓	FN↓	Frag.↓	MT↑	ML↓	PT↑
4	47.79	70.09	51.13	523	1408	**4513**	488	**342**	**120**	314
6	**49.22**	**71.70**	**51.87**	**399**	**1032**	4692	**306**	255	172	**349**
8	46.12	69.55	50.21	487	1076	4746	449	312	139	325

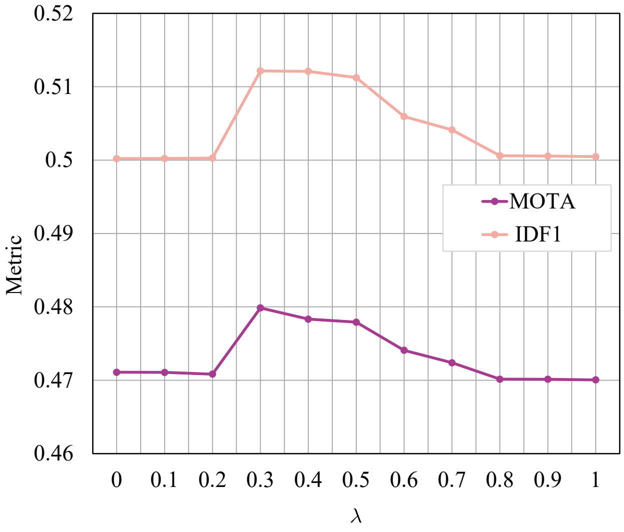


Figure 14. Performance variation due to different parameter λ.

performance is observed when dealing with larger scopes from Table 7. However, empirical evidence has shown that the utilization of SIRA consistently leads to superior performance compared to the TemporRadar (TR).

Tracking Results with Complete MOT Metrics: In Section 4.2, we showed tracking results in Table 2 with selected metrics. Here, we show the tracking results with complete metrics including MOTP, FP, FN and ML [35]. The tracking results are shown in Table 8 for comparison of the trackers. ID switches vary based on # of predicted BBox. With increased false negatives (FNs), both # of BBoxes and ID switches reduce. Table 8 suggests that TR generates more FNs and SIRA fewer FNs, thus higher ID switches. Nevertheless, we highlight the high IDF1 score in Table 2 and Table 8 of SIRA.

Number of Frames on Tracking: According to Table 9, considering longer time horizon contributes to the improvement in tracking performance in metrics such as MOTA and IDF1. These results clarify the significance of extending to longer time horizon while maintaining computational scalability. Fig. 12 illustrates the benefits of integrating more radar frames for the tracking performance over a range of methods. Compared with the single-frame baseline (ResNet-34 CenterTrack), SIRA with $T = 6$ consecutive frames results in a margin of $+34.67$ MOTA.

Effect of λ in (11): Fig. 14 illustrates the results obtained by varying λ in (11) in the main paper, which corresponds to C^{angle} when $\lambda = 1$ and C^{tracklet} when $\lambda = 0$. Fig. 14 appears to suggest that a combination of C^{angle} and C^{tracklet}, i.e., $\lambda \in [0.3, 0.7]$, consistently improves the tracking performance.

Performance of the Pseudo-Direction Estimation: We evaluated the pseudo-direction estimation performance in the terms of the smooth L_1 loss in (17) over the test dataset. Fig. 13 shows the loss histogram for three time steps $\tau = T-1/T-2/T-3$ and it confirms that the majority of estimation errors are close to 0, indicating a high accuracy.

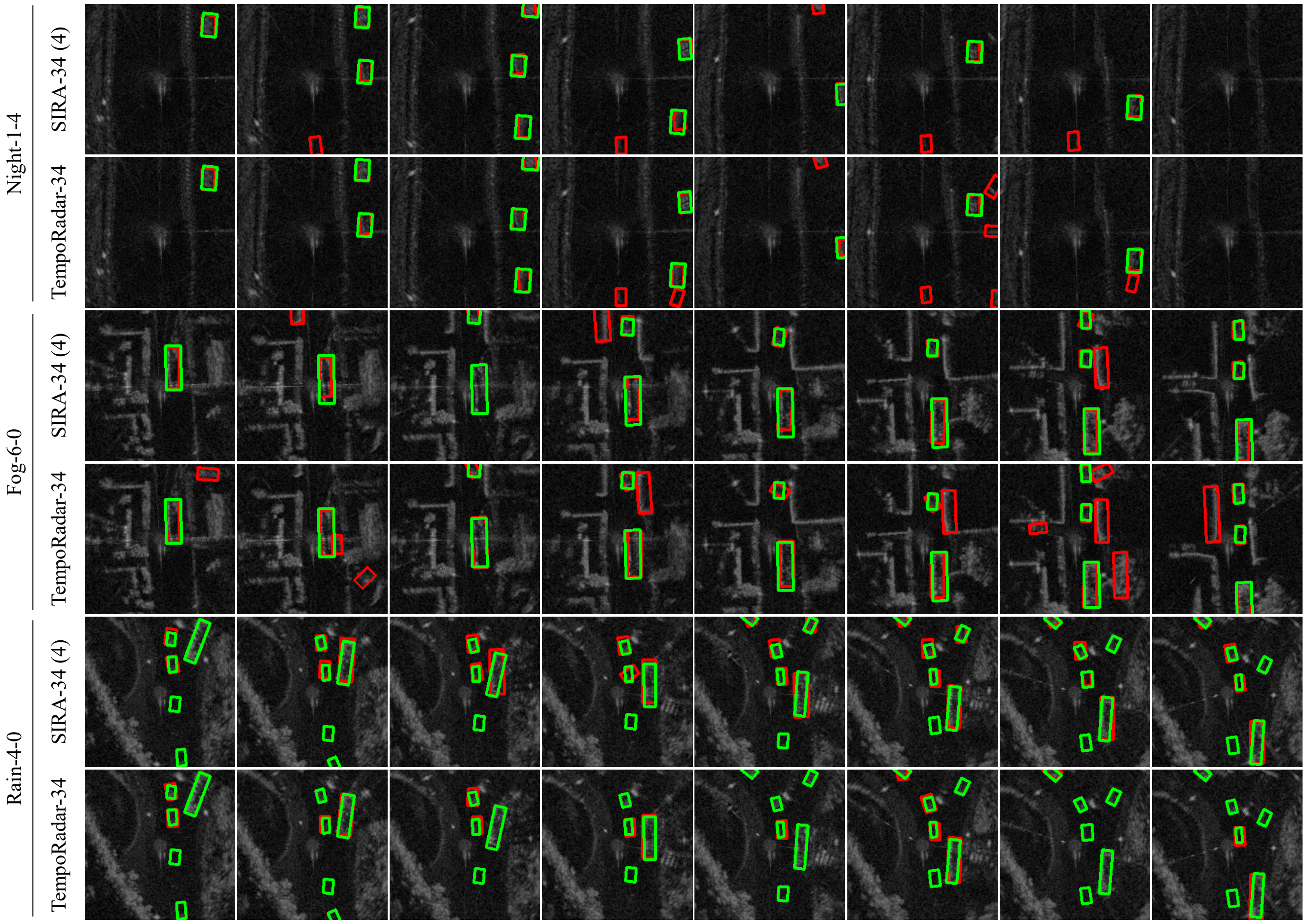


Figure 15. **Sampled detection results with cropped radar frames** on three scenarios: Night-1-4 (Top 2 Rows), Fog-6-0 (Middle 2 Rows) and Snow-4-0 (Bottom 2 Rows) on *Radiate*. For each scenario, we compare the SIRA and TempoRadar. Green boxes represent ground truth and red ones are predictions. The column represents consecutive radar frames. TempoRadar shows more false positives (FNs as unpaired red boxes) than SIRA, particularly in the first two scenarios.

16. Visualization Results

Detection with Cropped Radar Frames: Fig. 15 visualizes the detection results of Table 1 in Section 4.2 in adverse weather conditions: Night-1-4, Fog-6-0, and Rain-4-0, where green boxes represent the ground truth and red ones are the predictions. In this case, with $T = 4$ consecutive radar frames, SIRA allows for less FNs (unpaired green boxes) and less FPs (unpaired red boxes) in the BBox prediction than TempoRadar.

Detection with Full Size Radar Frames: Sampled detection results of Table 7 are visualized in Fig. 16. SIRA demonstrates robust performance even when applied to full size radar frames. However, compared to the cropped frames, there is a slight decline in performance with full size radar frames. Upon closer investigation of this phenomenon, it is observed that object shapes in regions distant from the radar appear blurred due to lower angular resolution, leading to a slight increase in both FPs and FNs. Furthermore, this blurring increases the difficulty of predicting angles, resulting in a lower IoU. FP predictions are also attributed to ghost objects present in the radar signal, as pointed out by Li et al. [27].

Tracking: Fig. 17 is in good weather, and Fig. 18 and Fig. 19 are in bad weather. From Fig. 17, TempoRadar faces by numerous FNs and frequent ID switches. In contrast, SIRA, leveraging longer temporal information for consideration of spatio-temporal consistency, exhibits fewer FNs and a reduced ID switches. As a result, SIRA consistently achieves stable tracking. Moreover, Fig. 18 illustrates that SIRA can detect and track objects even in adverse weather conditions. Particularly in the Rain-4-0 environment, where

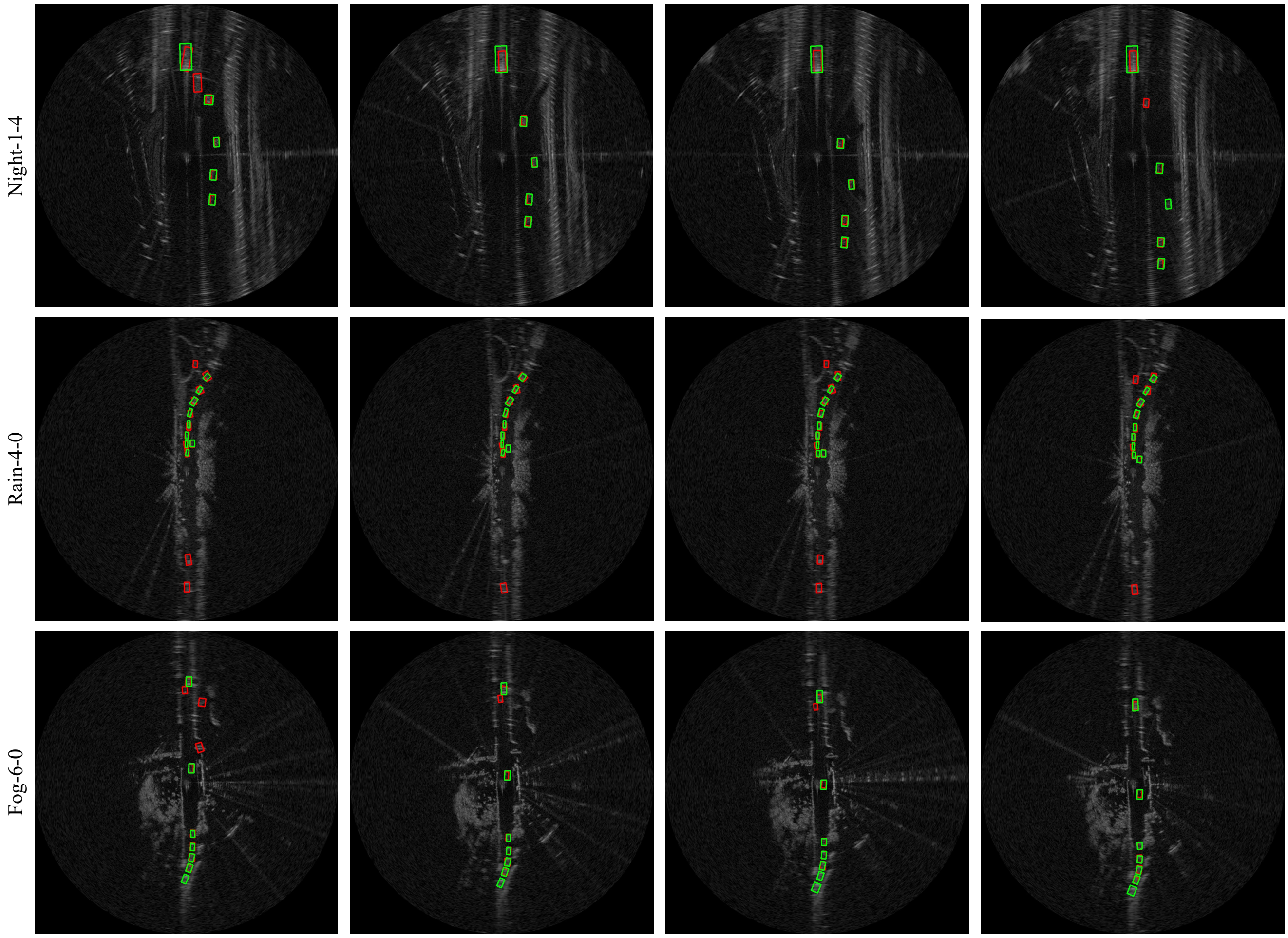


Figure 16. **Sampled detection results of SIRA-34 (6) with full size radar frames** on three scenarios: Night-1-4 (Top), Rain-4-0 (Middle) and Fog-6-0 (Bottom) on *Radiate*. Green boxes represent ground truth and red ones are predictions. The column represents consecutive radar frames.

vehicles exhibit nonlinear movement, the continuous tracking without interruptions underscores the effectiveness of MCTrack. However, in Fig. 19, SIRA does exhibit a slight presence of FPs, likely influenced by reflections from multipath or ghost objects, due to tracking across consecutive frames. Addressing such false information poses an intriguing challenge.

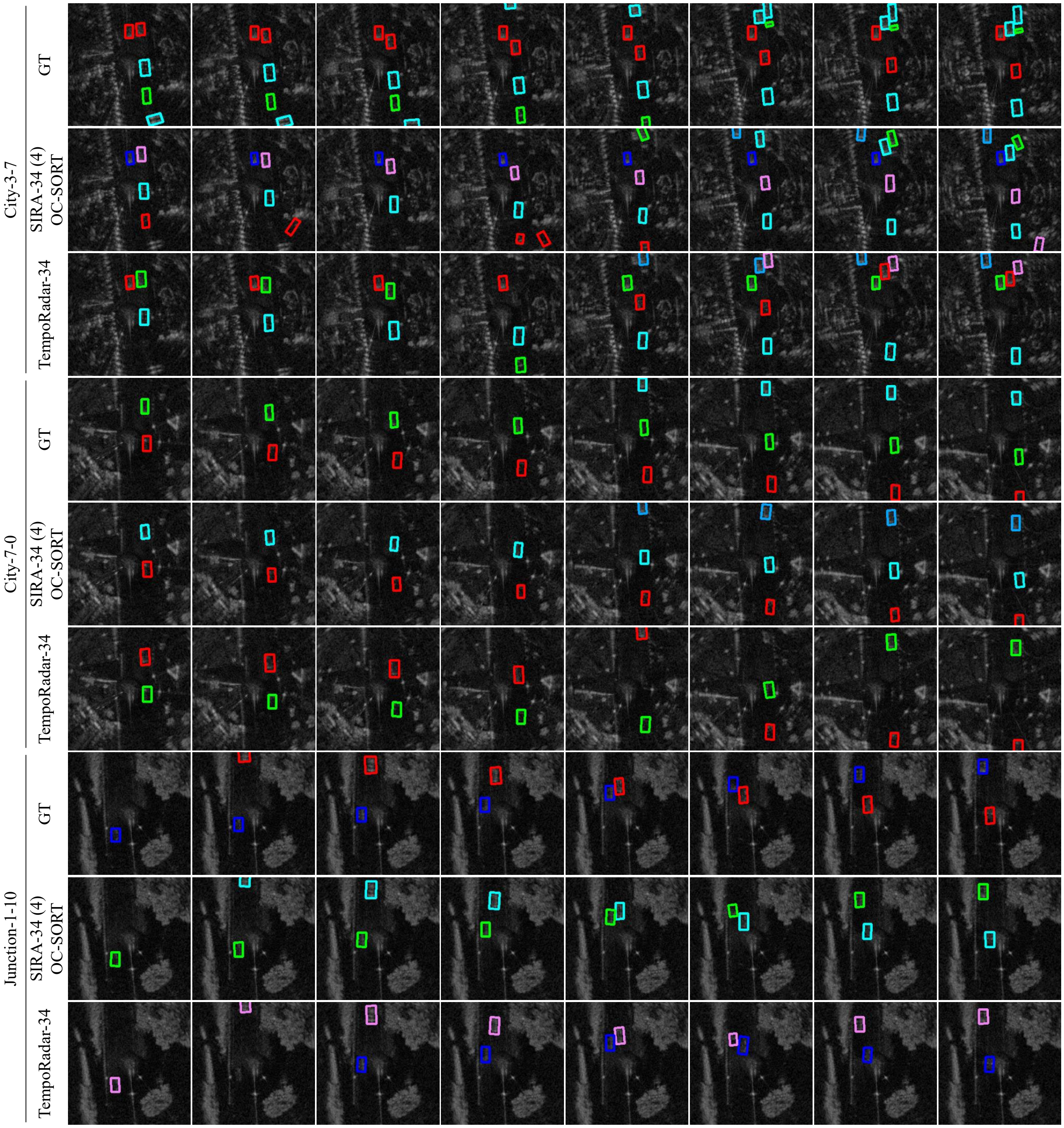


Figure 17. **Sampled tracking results** on three scenarios: City-3-7 (Top 3 Rows), City-7-0 (Middle 3 Rows) and Junction-1-10 (Bottom 3 Rows) on *Radiate*. For each scenario, we include ground truth (GT), SIRA (SIRA-34 (4)) and TempoRadar (TempoRadar-34). The color of bounding boxes represents the object ID. The column represents consecutive radar frames.

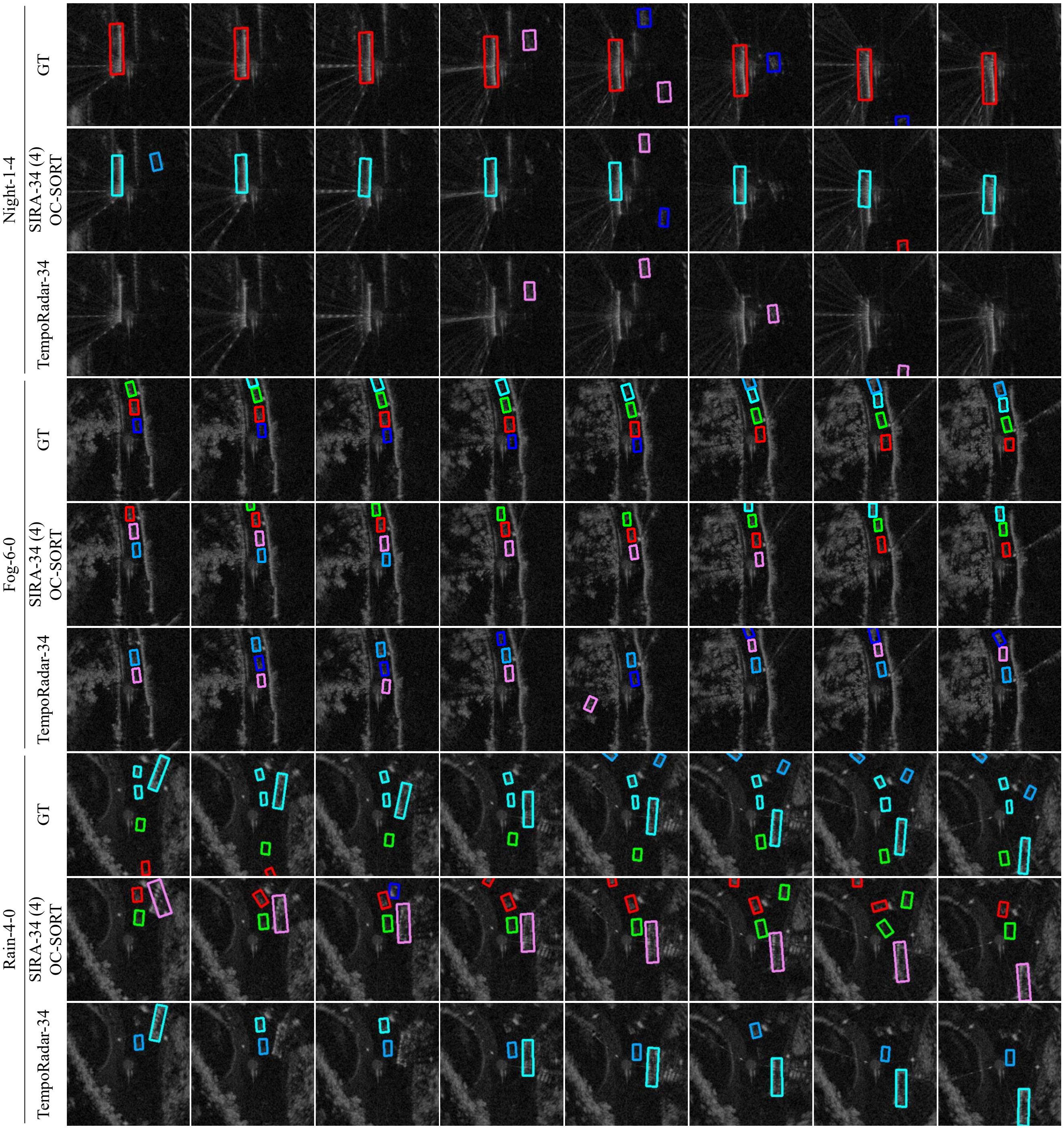


Figure 18. **Sampled tracking results** on three scenarios: Night-1-4 (Top 3 Rows), Fog-6-0 (Middle 3 Rows) and Rain-4-0 (Bottom 3 Rows) on *Radiate*. For each scenario, we include ground truth (GT), SIRA (SIRA-34 (4)) and TempoRadar (TempoRadar-34). The color of bounding boxes represents the object ID. The column represents consecutive radar frames.

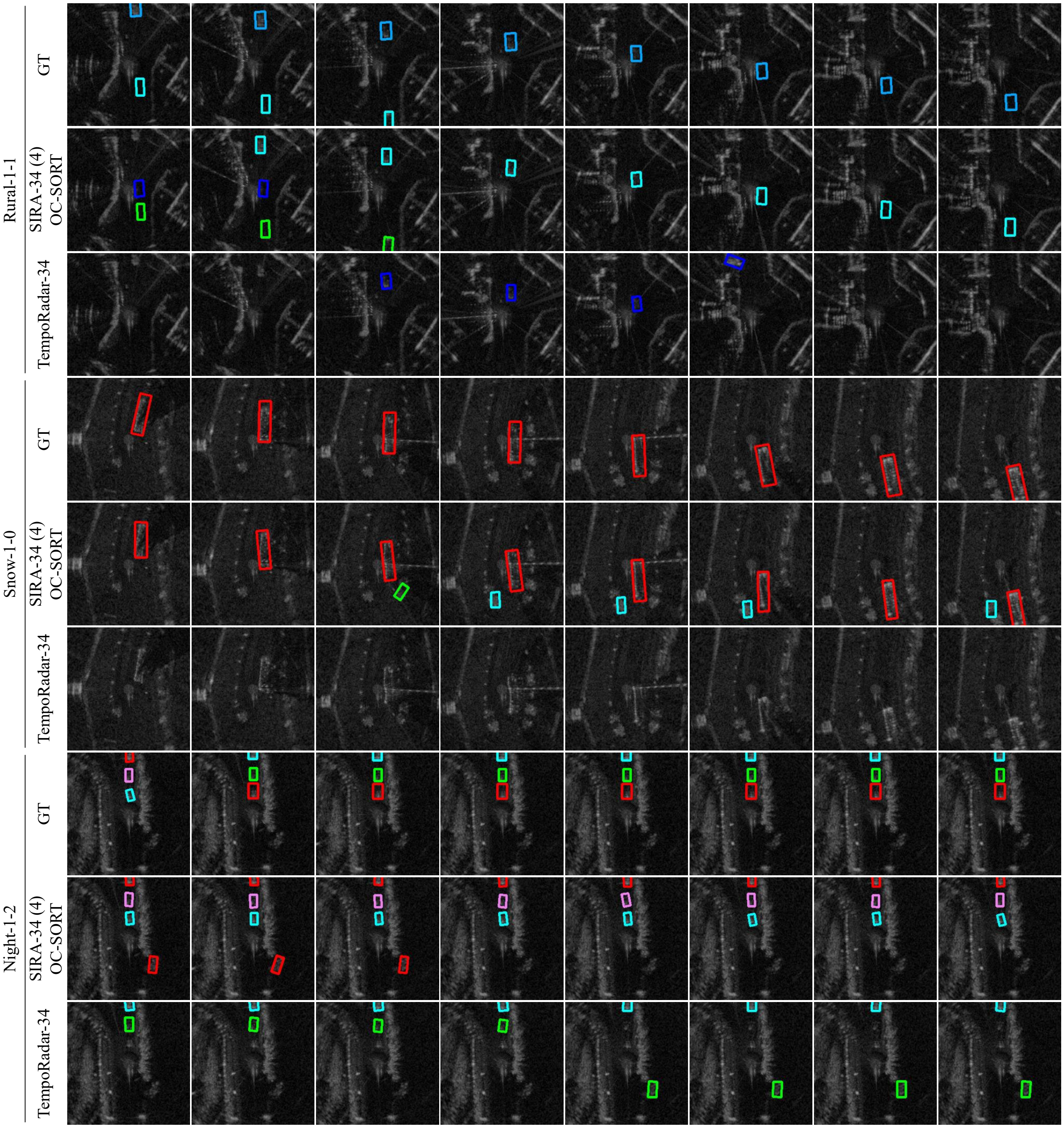


Figure 19. **Sampled tracking results** on three scenarios: Rural-1-1 (Top 3 Rows), Snow-1-0 (Middle 3 Rows) and Night-1-2 (Bottom 3 Rows) on *Radiate*. For each scenario, we include ground truth (GT), SIRA (SIRA-34 (4)) and TempoRadar (TempoRadar-34). The color of bounding boxes represents the object ID. The column represents consecutive radar frames.

17. KF-based Multiple Object Tracking for Radar Perception

Kalman Filter KF is a linear estimator for discretized dynamical systems in the time domain. KF operates by utilizing state estimations from the previous time step and current measurements to predict the target state at the next time step. The filter maintains two key variables: the posterior state estimate represented as $\mathbf{x}$, and the posterior estimate covariance matrix denoted as $\mathbf{P}$.

In the context of object tracking, the KF process is defined by several components, including the state transition model $\mathbf{F}$, the observation model $\mathbf{H}$, the process noise covariance $\mathbf{Q}$, and the measurement noise covariance $\mathbf{R}$. In each time step t, when presented with observations $\mathbf{z}_t$, the KF operates through a sequence of predict and update stages.

$$\text{predict}\begin{cases} \widehat{\mathbf{x}}_{t|t-1} = \mathbf{F}_t \widehat{\mathbf{x}}_{t-1|t-1} \\ \mathbf{P}_{t|t-1} = \mathbf{F}_t \mathbf{P}_{t-1|t-1} \mathbf{F}_t^\top + \mathbf{Q}_t, \end{cases} \tag{26}$$

$$\text{update}\begin{cases} \mathbf{K}_t = \mathbf{P}_{t|t-1} \mathbf{G}_t^\top \left(\mathbf{G}_t \mathbf{P}_{t|t-1} \mathbf{G}_t^\top + \mathbf{R}_t \right)^{-1} \\ \widehat{\mathbf{x}}_{t|t} = \widehat{\mathbf{x}}_{t|t-1} + \mathbf{K}_t \left(\mathbf{z}_t - \mathbf{G}_t \widehat{\mathbf{x}}_{t|t-1} \right) \\ \mathbf{P}_{t|t} = \left(\mathbf{I} - \mathbf{K}_t \mathbf{G}_t \right) \mathbf{P}_{t|t-1} \end{cases} . \tag{27}$$

The prediction stage involves calculating the state estimations for the subsequent time step t. In contrast, the update stage is focused on refining the posterior parameters within the KF when presented with measurfor thents of target states for time step t. In many scenarios, this measurement is derived from the observation model $\mathbf{H}$ and is commonly referred to as an observation.

KF parameters In MOT, KF-based typically consists of five steps: Prediction, Association, Update, Deletion, and Initialization. The prediction and update phases are handled by KF. In our setting for radar perception, the KF's state $\mathbf{x}_t$ and observation $\mathbf{z}_t$ is defined as follows:

$$\mathbf{x}_t := \left(x_t, y_t, s_t, r_t, \vartheta_t, \dot{x}_t, \dot{y}_t, \dot{s}_t, \dot{\vartheta}_t \right)^\top , \tag{28}$$

$$\mathbf{z}_t := \left(x_t, y_t, \widehat{w}_t, \widehat{h}_t, \widehat{\vartheta}_t, \widehat{c}_t \;\middle|\; \widehat{c}_t > \gamma \right)^\top , \tag{29}$$

where (x_t, y_t) is the two-dimensional coordinates of the object center in the image. $s = w \times h$ is the bounding box scale (area), r is the bounding box aspect ratio and ϑ is object orientation, where w and h are the width and height of the object. The aspect ratio $r = \frac{w}{\text{float}\,(h+1\text{e}-6)}$ is assumed to be constant. The other four variables, $\dot{x}$, $\dot{y}$, $\dot{s}$ and $\dot{\vartheta}$ are the corresponding time derivatives. The detection confidence is c. The observation model is

$$\mathbf{G}_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{30}$$

We note the process noise as in practice: $\mathbf{Q}_t = \text{diag}\left(\sigma_x^2, \sigma_y^2, \sigma_s^2, \sigma_r^2, \sigma_\vartheta^2, \sigma_{\dot{u}}^2, \sigma_{\dot{v}}^2, \sigma_{\dot{s}}^2, \sigma_{\dot{\vartheta}}^2 \right)$. In the practice of SORT, we have to suppress the noise from velocity terms because it is too sensitive. We achieve it by setting a proper value for the process noise:

$$\mathbf{Q}_t = \text{diag}\left(0.1, 5, 1^{-4}, 1^{-4}, 10, 0.01, 0.01, 1^{-4}, 0.1 \right) . \tag{31}$$

We note the linear transition model as:

$$\mathbf{F}_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} , \tag{32}$$

We set the measurement noise covariance as:

$$\mathbf{R}_t = 10\mathbf{I}_5 . \tag{33}$$

We need to choose an initial value for $\mathbf{P}_{t-1|t-1}$, call it $\mathbf{P}_{0|0}$. If we were absolutely certain that our initial state estimate $\mathbf{x}_0 = \mathbf{0}$ was correct, we would let $\mathbf{P}_{0|0} = \mathbf{0}$. However, given the uncertainty in our initial estimate $\mathbf{x}_0$, choosing $\mathbf{P}_{0|0} = \mathbf{0}$ would cause the filter to initially and always believe $\mathbf{x}_t = \mathbf{0}$. Assuming some uncertainty in the initial state, we set as follows:

$$\mathbf{P}_{0|0} = \text{diag}\left(10, 10, 10, 10, 10, 10, 10, 10000, 10000 \right) , \tag{34}$$

where $\dot{\vartheta}$ and $\dot{s}$ are set to a large value as the uncertainty is particularly high. On the other hand, we use the estimated pseudo-direction $\widehat{\mathbf{d}}_{T|T-1}$ as the initial value for $\dot{u}, \dot{v}$. Therefore, we set small uncertainties for these.

18. Fundamentals of FMCW for Automotive Radar

Radar technology offers a sensing solution that exhibits increased resilience to adverse weather conditions such as fog, rain, and snow. Typically, it generates low-resolution imagery, presenting significant challenges for tasks like object recognition and semantic segmentation. Contemporary automotive radar systems are primarily based on the Multiple Input Multiple Output (MIMO) technique, which employs multiple transmitters and receivers to determine the direction of arrival (DOA) [22]. Although this approach is cost-effective, existing configurations often suffer from limited azimuth resolution. For example, a commercial radar system with a 15° angular resolution produces a cross-range image with an approximate span of 10 meters at a distance of 20 meters. Consequently, radar imagery does not provide the level of detail necessary for effective object recognition and detailed scene mapping. On the contrary, the scanning radar employs a mobile antenna to measure azimuth at each point, leading to significantly improved azimuth resolution [47].

Transmitter From [50], automotive radar predominantly employs a frequency-modulated continuous waveform (FMCW) for object detection, generating point clouds across multiple physical domains. As shown in Fig. 20, this is achieved by transmitting a series of K coded FMCW pulses from one of its M Tx transmitting antennas, given by the expression of the radio frequency (RF) wave form on Tx antenna m:

$$s_m(t) = \sum_{k=0}^{K-1} c_m(k)\, s_p(t - nT_{\text{PRI}})\, e^{j2\pi f_c t}, \tag{35}$$

$$s_p(t) = \begin{cases} e^{j\pi\beta t^2} & 0 \le t \le T \\ 0 & \text{otherwise} \end{cases}, \tag{36}$$

where $s_p(t)$ is the baseband FMCW waveform (chirp pulse) with β denoting the chirp rate and T the pulse duration, and is repeated K times. k is the index for pulse, and $c_m(k)$ is the slow-time orthogonal code for the k-the pulse at the m-th Tx antenna, which satisfies the following:

$$\sum_{k=0}^{K-1} c_i(k)\, c_m(k) = \begin{cases} K & \text{if } i = m \\ 0 & \text{otherwise} \end{cases}. \tag{37}$$

T_{PRI} is pulse repetition interval and f_c is the carrier frequency, e.g., $f_c = 79$ GHz. The bandwidth of the FMCW waveform is $B = \beta T$. The baseband waveform is repeated at each antenna before being multiplied by orthogonal codes $c_m(k)$, for example, the Hadamard code.

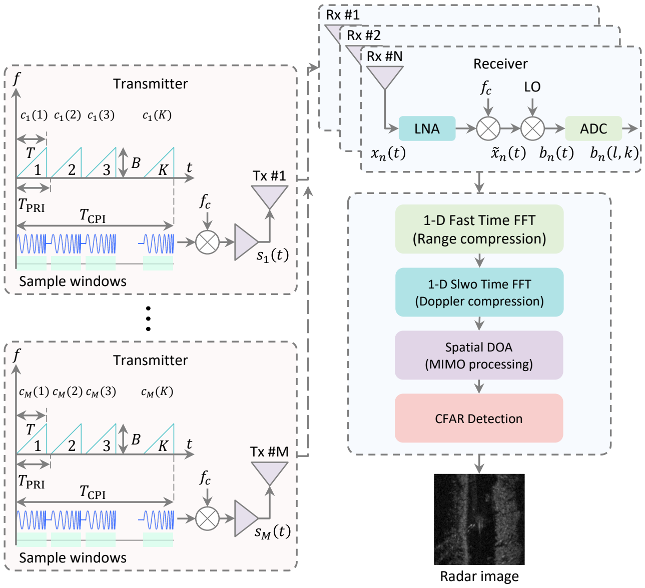


Figure 20. The slow-time FMCW automotive radar architecture from [50]. On the left, a sequence of FMCW pulses with orthogonal slow-time (pulse) codes are sent from M transmitting antennas while, on the right, each of N receivers uses the same source FMCW waveform to sample the beat signal followed by range-doppler processing and slow-time waveform separation for spatial detection.

Receiver An object at a range of R_0 with a radial velocity v 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 Rx receiver RF chains (including the low noise amplifier (LNA), local oscillator (LO), and analog-to-digital converter (ADC)) of Fig. 20. The round-trip propagation delay from m-th Tx antenna to its n-th Rx antenna is

$$\tau_{mn}(t) = 2\frac{R_0 + vt}{c} + m\frac{d_t \sin(\theta_t)}{c} + n\frac{d_r \sin(\theta_r)}{c}, \tag{38}$$

where d_t, d_r, θ_t and θ_r are the inter-element spacing and azimuthal angle for the transmitting and receiving antennas, respectively. We assume co-located radars and the far-field approximation, i.e. $\theta_r = \theta_t = \theta$. c is the speed of propagation. In the presence of an object at angle θ, the n-th Rx receiver receives the signal of a sum of M attenuated and delayed transmitting waveforms:

$$x_n(t) = \alpha \sum_{m=0}^{M-1} s_m(t - \tau_{mn})\, e^{j2\pi f_c (t - \tau_{mn})}. \tag{39}$$

Subsequently, the baseband signal after LNA and carrier frequency down conversion is as follows:

$$\tilde{x}_n(t) = x_n(t)\, e^{-j2\pi f_c t} \tag{40}$$

$$\approx \tilde{\alpha} \sum_{m=0}^{M-1} s_m(t - \tau_0)\, e^{-j2\pi f_c \frac{2vt}{c}}\, e^{-j2\pi (m d_t + n d_r)\frac{\sin(\theta)}{\lambda}}, \tag{41}$$

where $\tau_0 = \frac{2R_0}{c}$ is the time taken from the transmission to the reception, and $\lambda = \frac{c}{f_c}$ is wave length. We assume that

$s_m(t-\tau_{mn}) = s_m(t-\tau_0)$, and $\tilde{\alpha}$ absorbs constant phase factors. By using LO, the signals at all receivers are mixed with the source chirp to generate the analog beat signal:

$$b_n(t) = \tilde{x}_n(t) \sum_{k=0}^{K-1} s_p^*(t - kT_{\mathrm{PRI}}), \tag{42}$$

where $*$ denotes its conjugate. This analog beat signal is then sampled at $t = kT_{\mathrm{PRI}} + l\Delta T$ with ADC sampling, where ΔT and T_{PRI} are the fast-time and slow-time sampling intervals, respectively, and digital beat signal is represented as follows:

$$b_n(l,k) = \tilde{\alpha} \sum_{m=0}^{M-1} c_m(k) \underbrace{e^{-j2\pi f_r l}}_{\text{Range}} \underbrace{e^{-j2\pi f_d k}}_{\text{Doppler}} \underbrace{e^{-j2\pi\left(f_s^t m + f_s^r n\right)}}_{\text{Virtual Spatial Array}}, \tag{43}$$

where $f_r = \left(\beta\tau_0 + 2f_c\frac{v}{c}\right)\Delta T$ is normalized range (fast-time) frequency, $f_d = 2f_c T_{\mathrm{PRI}}\frac{v}{c}$ is the normalized Doppler (slow-time) frequency, and f_s^t and f_s^r are the normalized spatial frequency at the transmitting and receiving antennas. f_s^t is usually different from f_s^r due to different Tx/Rx spacings. In other words, the beat signal $b_n(l,k)$ at n-th receiver is the sum of the object responses originating from all transmitted waveforms, coded using $c_m(k)$. The beat signal at each of N Rx receiver forms a matrix:

$$\mathbf{B}_n = \begin{bmatrix} b_n(1,1) & b_n(2,1) & \dots & b_n(L,1) \\ b_n(1,2) & b_n(2,1) & \dots & b_n(L,2) \\ \vdots & \vdots & \ddots & \vdots \\ b_n(1,K) & b_n(2,2) & \dots & b_n(L,K) \end{bmatrix}. \tag{44}$$

The induced modulation from the target is captured by the baseband signal processing block (including fast Fourier transforms (FFT) over range, Doppler, and spatial domains). 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 [5, 26, 27, 50]. Considering the computing and cost constraints, automotive radar manufacturers may define the radar point clouds in a subset of the full four dimensions. For example, traditional automotive radar generates detection points in the range-Doppler domain, whereas some produce the points in the range-Doppler azimuth plane [44]. In the *Radiate* dataset [47] 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 a radar frame as $I_t \in \mathbb{R}^{1\times H\times W}$.