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

Ryoma Yataka\textsuperscript{1,2}, Pu Wang\textsuperscript{1}, Petros Boufounos\textsuperscript{1}, Ryuhei Takahashi\textsuperscript{2}
\textsuperscript{1}Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA
\textsuperscript{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 consistency 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

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.

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.

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 multimodal, 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.

Multiple 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 \( I_t \in \mathbb{R}^{C \times H \times W} \) into high-level features and accentuate the position of objects.

\[
\mathbf{Z}_t := \mathcal{F}_\theta (I_t) \in \mathbb{R}^{C \times \frac{H}{s} \times \frac{W}{s}},
\]

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

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 \( \hat{c}_t \)
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 \( L_{\text{Det}} \) and pseudo-direction loss \( L_{\text{Dir}} \) are used to train the pipeline end-to-end for object detection and tracking.

Exploiting Temporality: For radar perception, it is necessary to 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., \( Z_t = \mathcal{F}_\theta (I_1) \). To exploiting the feature-level temporal relation, TempoRadar [27] introduces a temporal relation layer (TRL) that selects top-\( K \) features \( H_t \in \mathbb{R}^{C \times K} \) from \( Z_t = \mathcal{F}_\theta (I_{t-1,t-1}) \) and \( H_{t-1} \in \mathbb{R}^{C \times K} \) from \( Z_{t-1} = \mathcal{F}_\theta (I_{t-2,t}) \), 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 \( S_K \):

\[
H_t = S_K (Z_t), \quad t = \{t-1,t\}. \tag{3}
\]

By concatenating the 2\( K \) selected features as \( H_{t,t-1} = [H_t,H_{t-1}]^\top \), TRL further computes masked multi-head cross-attention (MCA) as

\[
\mathcal{A}(\mathbf{V}, \mathbf{X}) := \operatorname{softmax} \left( \frac{\mathbf{M} + q(\mathbf{X}) k(\mathbf{X})^\top}{\sqrt{d}} \right) v(\mathbf{V}) \tag{4}
\]

where \( \mathbf{V} = H_{t,t-1}, \mathbf{X} = H_{t,t-1}^{\text{pos}} \) is the concatenated feature \( H_{t,t-1} \) supplemented by the positional encoding, \( \{q(\cdot), k(\cdot), v(\cdot)\} \) 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 refill back to \( Z_t \) and \( Z_{t-1} \) at corresponding spatial coordinates and fed to the decoder for object detection and tracking. Refer to Appendix 8 for the top-\( K \) feature selector \( S_K \) 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.,

\[
Z_t := \mathcal{F}_\theta (I_{t,t-1,\ldots,t-U+1}), \quad Z_{t-1} := \mathcal{F}_\theta (I_{t-1,t-2,\ldots,t-U+1}), \cdots, \quad Z_{t-U+1} := \mathcal{F}_\theta (I_{t-U+1,t,t-1,\ldots,t-U+2}). \tag{5}
\]
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)

$$H_t = S_K(Z_t), \quad t = \{t, t - 1, \cdots, t - U + 1\}. \quad (6)$$

By concatenating features from the temporal window of $U$ frames, we have $H_t^{t-1, t-U+1} = [H_t^{t-1}, \cdots, H_t^{t-U+1}]^T$, where the superindex denotes the layer index in the ETR model and $H_t^0$ takes $H_t$ of (6) as input for the first layer. We apply the masked MCA of (4) $H_t$ times to $H_t^{t-1, t-U+1}$ with a masking matrix $M$ of dim $UK \times UK$ for cross-frame feature attention within each window. Collecting from all windows, the TW A block obtains the features $H_t^i, \cdots, H_t^{i-T+1}$ 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 $\Omega$ 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 $H_t^i[\omega] \in \mathbb{R}^{C \times M}, \omega = 1, \cdots, \Omega$. 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

$$F_t^i(\omega) := \{H_t^i[\omega], H_t^{i-U}[\omega], \cdots, H_t^{i-T+U}[\omega]\}^T, \quad (7)$$

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) $H_t$ times over the aggregated feature $F_t^i(\omega)$ in each new window with an affordable cross-window attention complexity of $TM/\Omega \times TM/\Omega$.

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 ($\omega$) indices. In the case of overlapping partitioning, i.e., $S < M$, a patch merging operation $M$ is necessary to merge the features $H_t^{i-t} = M\{H_t^{i-1}[i], \cdots, H_t^{i+1}[\omega]\}$ at the overlapping positions. Patch merging operations (mean, sum and max) will be examined in Section 4.3. The TRW block outputs $H_t^{t-1}, \cdots, H_t^{i-T+1}$ for all $T$ frames, sharing the same dimension as the input $H_t^i, \cdots, H_t^{i-T+1}$.

![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.](image)

**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 $H_t^{i+1}, \cdots, H_t^{i+1}$ back to $\{Z_t, \cdots, Z_{t-T+1}\}$ 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.

TempoRadar: $(TK)^2 L$ \quad (8)
ETR: $(TWA + TRWA) L = K^2TLU + MT^2KL/U \quad (9)$

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 $\{Z_t\}$ from the ETR output, and applies the decoding heads on each $Z_t$ for bounding box estimation. To further exploit motion consistency, we introduce two MC
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 \( \tau \)-step pseudo-direction \( \hat{d}_{T|T-\tau} \) from the past frame \( T-\tau \) to frame \( T \), we design a specific decoder head \( g_\theta^{\text{DEst}}(\cdot) \): direction estimation (DEst) with learnable parameters \( \theta \) in Fig. 4,

\[
\hat{d}_{T|T-\tau} = g_\theta^{\text{DEst}}(Z_T, Z_{T-\tau}) \in \mathbb{R}^2,
\]

where \( Z_T \) and \( Z_{T-\tau} \) are temporally enhanced features at frame \( T \) and \( T-\tau \). \( \hat{d}_{T|T-\tau} \) is a two-dimensional coordinate, and \( \tau = 1, 2, \ldots, T-1 \). 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 \( \tau \) frames.

The estimated vectors represent the positional differences of objects across \( \tau \) 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.

\footnote{With slightly abused notation, we use \( T \) to denote not only the number of frames, but also current frame index in this section.}

To this end, we further introduce the concept of pseudo-tracklet, constructed from the above pseudo-direction estimation. A pseudo-tracklet consists of a pair of vectors: \( \{\hat{z}_t\}_{t=1}^{T} \), \( \{\hat{v}_t\}_{t=2}^{T} \). \( \hat{z}_t \) is an estimated observation with pseudo-direction \( \hat{d}_{T|T-\tau} \) and \( z_T \) (Top of Fig. 5), and \( \hat{v}_t \) 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:

\[
C_{\text{MCTrack}} = \lambda C_{\text{single}} + (1 - \lambda) C_{\text{tracklet}},
\]

\[
C_{\text{tracklet}} = \frac{1}{T-1} \sum_{\tau=1}^{T-1} \text{GIoU} \left( B_{z\tau\tau-\tau}, B_{\hat{x}\tau\tau-\tau} \right),
\]

\[
C_{\text{single}} = \text{GIoU} \left( B_{z\tau}, B_{\hat{x}\tau\tau-\tau} \right),
\]

where \( \lambda \) is the weighting coefficient, \( B \) represents the BBox with subscripts, and \( \text{GIoU} [46] \) denotes the similarity determined based on the distance between two BBoxes. In other words, \( C_{\text{tracklet}} \) and \( C_{\text{single}} \) represent the similarity between the similarity between the pseudo-tracklet and the trajectory of the KF, and the current observation \( z_T \) and the rotated state \( \hat{x}_{\tau\tau-\tau} \) of the KF, respectively.

\footnote{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\textsuperscript{tracklet} directly correlates the observations $z_i$ of the KF trajectory with the estimated observations $\hat{z}_i$ 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\textsuperscript{tracklet} and C\textsuperscript{angle} is reported in Section 4.3. Refer to 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\textsuperscript{angle}, the predicted state $\hat{x}_{T−1}$ with KF from the previous state $\hat{x}_{T−1}$ is rotated with a rotation matrix $R$ of angle $\phi$\textsubscript{ave}. It can be calculated as $p_{z_i}\hat{x}_{T−1} = R(p_{\hat{x}_{T−1}} − p_{\hat{x}_{T−1}}) + p_{\hat{x}_{T−1}}$, where the angle $\text{avg}$ can be calculated as $\hat{\phi}_{\text{ave}} = \frac{1}{T−2} \sum_{\tau=0}^{T−3} \hat{\phi}_{T−\tau}$ such that $\hat{\phi}_{T−\tau} = \cos^{-1}\left(\frac{\hat{v}_{T−\tau} \cdot \hat{v}_{T−\tau} - 1}{\|\hat{v}_{T−\tau}\|\|\hat{v}_{T−\tau}\|}\right)$. By using this rotated state $\hat{x}_{T−T−1}$, we can avoid a high correlation between the predicted state assuming linear motion and the incorrect observation $z_{t}^{\text{true}}$.

Our approach exploits the proposition that the temporally enhanced features across multiple frames from ETR allows for more robust estimation of the pseudo-direction $\hat{d}_{T−T−\tau}$ 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 ($L_t^{\text{BBox}}$), and a loss of the pseudo-direction in which an object has moved between frames and the actual movement direction ($L_t^{\text{DEst}}$), as shown in Fig. 2.

$$L_\theta := \sum_{t=1}^{T} \left( L_t^{\text{DEst}} + L_t^{\text{BBox}} \right).$$

(14)

For each training step, our training procedure calculates $L_\theta$ and does the backward for both $t = 1$ to $t = T$ and $t = T$ to $t = 1$ simultaneously. Therefore, optimization $\min_\theta L_\theta$ can be viewed as a bidirectional backward-forward training through $T$ frames. For more clear training 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 $L_t^{h}$, width & Length loss $L_t^{l}$, orientation loss $L_t^{l}$, and offset loss $L_t^{l}$, compose the training objective by a linear combination:

$$L_t^{\text{BBox}} = \frac{1}{N_{gt}} \sum_{k=1}^{N_{gt}} \left( L_{t,k}^{\text{H}} + L_{t,k}^{\text{L}} + L_{t,k}^{\text{O}} \right) - \frac{1}{N} \sum_{i=1}^{N} L_{t,i}^{h},$$

(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: $L_t^{\text{DEst}}$ represents a pseudo-direction estimation loss:

$$L_t^{\text{DEst}} = \frac{1}{N_{gt}} \sum_{k=1}^{N_{gt}} L_{t,k}^{\text{DEst}},$$

(16)

$$L_{t,k}^{\text{DEst}} = \frac{1}{T−1} \sum_{\tau=1}^{T−1} \left\{ S_L\left(\|\widehat{d}_{t,\tau} - \hat{d}_{t,\tau}^{\text{gt}}\|\right) \right\} \tau \neq t,$n

$$\tau = t,$n

(17)

where $\widehat{d}_{t,\tau} = G_0^{\text{DEst}}(Z_t, Z_\tau)\left[p_{t,k}^{\text{gt}}\right]$ denotes a two-dimensional direction from a position of time $\tau$ to a position of time $t$ as mentioned in Section 3.3, $p_{t,k}^{\text{gt}}$ denotes the coordinate $(x_{t,k}, y_{t,k})$ of the center of $k$-th ground truth object and $S_L(\cdot)$ is a smooth $L_1$ loss [15]. $\hat{d}_{t,\tau} = p_{t,k}^{\text{gt}} - p_{t,k}^{\text{gt}}$ 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
We report the detection results in Table 1. The number following the model name indicates the # of layers in the ResNet, and the number in parentheses indicates the # of frames T.

We show the visualization results in Fig. 6. 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 $256 \times 256$ upon input images and exclude the targets outside this scope. We additionally report the detection results with the full size ($1152 \times 1152$) images in Appendix 15.

### Metrics
We adopt the mean average precision (mAP) with intersection over union (IoU) at 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 $+3\text{ mAP}@0.3$ and about $+2.5\text{ mAP}@0.5$ from single frame of RetinaNet, CenterPoint, BBAVectors to two frames of the TempoRadar, and further more of about $+5\text{ mAP}@0.3$ and about $+4\text{ mAP}@0.5$ 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.

#### 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 $L_t^{Dest}$ 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.
Table 3. SIRA ablation experiments on Radiate. If not specified, we used SIRA-34 (4) trained on train good weather and followed the experimental settings for other parameters. The best performance is marked in gray.

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

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 $\lambda$.

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 MC-Track. 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


[27] Peizhao Li, Pu Wang, Karl Berntorp, and Hongfu Liu. Exploiting temporal relations on radar perception for autonomous driving. In *the 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


[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 au-


