Journal of *Marine Science and Engineering*

MDPI

Article

TrajE2E-MOT: Trajectory-Aware End-to-End Multi-Object Tracking in Maritime Radar

Zhan Kong [1,*], Wei Xiong [1] and Yaqi Cui [1,2,3]

[1] Institute of Information Fusion, Naval Aviation University, Yantai 264001, China; xiongwei@csif.org.cn (W.X.); cui_yaqi@126.com (Y.C.)
[2] AVIC Shenyang Aircraft Design and Research Institute, Shenyang 110801, China
[3] School of Automation, Northwestern Polytechnical University, Xi'an 710072, China
* Correspondence: kz2020001@163.com

Abstract

For autonomous maritime perception and situational awareness, the end-to-end multi-object tracking paradigm has achieved complete learning, from image sequences to tracking results, reducing the reliance on manually designed association rules and holding great potential. However, in maritime radar video multi-object tracking, due to the limited visual features of targets and significant feature variations under long-term tracking, problems such as identity switching are prone to occur, making it difficult to directly apply existing end-to-end approaches. To solve these problems, this paper proposes a trajectory-aware end-to-end multi-object tracking method. The real-time trajectory of the targets contains temporal context information. This work uses it as prior knowledge to enhance visual feature encoding and compensate for the shortcomings of single-frame visual features. Specifically, the trajectory feature is encoded by the trajectory encoder module while, simultaneously, the visual features are encoded through the backbone and the visual feature encoder module. Then, in the frame-trajectory cross-modal attention module, the trajectory feature encoding is used to reconstruct the visual feature encoding with cross-attention, dynamically enhancing the features related to the target identity. Experiments on actual collected maritime radar video data show that the proposed method is effective, achieving improvements in several key indicators.

Keywords: multi-object tracking; trajectory aware; end to end; radar video data

Check for updates

Academic Editors: Jiqiang Li, Guoqing Zhang, Antonio M. Pascoal and Weidong Zhang

Received: 1 June 2026
Revised: 29 June 2026
Accepted: 30 June 2026
Published: 2 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Multi-object tracking is a key perception technology for autonomous ship navigation and maritime situational awareness. Typical research paradigms include detection-based tracking and joint detection and tracking. Detection-based tracking decouples the detection and tracking processes. This paradigm offers high flexibility, but it requires independent training and optimization for both modules, and the detection error directly affects the tracking results. Under the paradigm of joint detection and tracking, it shares the same backbone network for detection and tracking, integrating the detection and tracking tasks into a unified model for joint learning and optimization. Some algorithms under this paradigm—that proposed by FairMOT [1] being a typical example—despite simultaneously outputting detection boxes and appearance features in the same network, still rely on manually designed association rules in the association stage and lack adaptive capabilities for complex scenarios.

Another route for joint detection and tracking is end-to-end multi-object tracking [2,3], which aims to build a unified model to directly achieve tracking based on the image sequence. It not only realizes joint learning of detection and tracking but also transforms the traditionally rule-based data association process into a learnable method, realizing a fully end-to-end tracking method. This method has shown great potential in the field of computer vision and provides a new idea for building more intelligent tracking.

In natural video, the deployment of the end-to-end paradigm faces challenges. It is difficult to meet the real-time processing requirements of high-frame-rate videos (around 20 fps) using inference delay (typically less than 10 FPS [2]). However, in radar applications, this limitation is inherently mitigated. Take navigation radar as an example; each complete 360-degree scan generates one radar image frame. The sequence of images produced during continuous scanning can be viewed as ultra-low-frame-rate video data (with the radar's scan cycle typically on the order of seconds, whereas typical natural video-frame updates occur at millisecond-level intervals). This low-frame-rate characteristic provides a rare deployment opportunity for end-to-end tracking models with greater computational requirements.

Nevertheless, the end-to-end models designed for natural videos are difficult to directly apply to radar videos. As Figure 1a [4] shows, a person in natural video has rich texture, color, and shape features, which remain stable in different frames. These features provide the model with more discriminative information. In contrast, as shown in Figure 1b, the visual features of targets in radar video are limited and lack internal texture features. Moreover, due to the imaging mechanism of radar, target maneuvering and occlusion, the appearance features of targets change significantly over a long time. These characteristics result in the low robustness of association based solely on single-frame visual features, which are prone to cause errors such as identity switching.

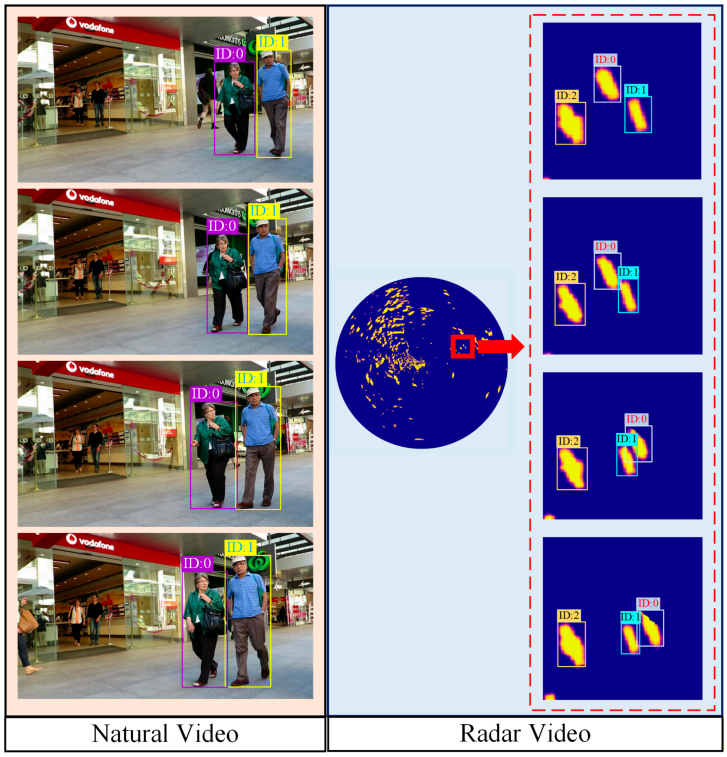


Figure 1. Comparison of natural video and radar video. (**a**) Images of multiple frames in the natural video. (**b**) Images of multiple frames in the radar video.

In view of this, we recognize that in radar video scenarios, it is necessary to find enhanced information beyond single-frame visual information. The real-time trajectory of the target contains temporal context information, which can be used as prior knowledge to enhance visual feature encoding, compensating for the shortcomings of single-frame visual features and avoiding the introduction of additional sensors. Inspired by this, we propose TrajE2E-MOT, a trajectory-aware end-to-end multi-object tracking model for radar video sequences. The proposed method is built upon the query-based end-to-end tracking framework of TrackFormer [2], including its Transformer-based encoder–decoder structure and set-prediction tracking paradigm. However, TrajE2E-MOT is not a direct application of TrackFormer to maritime radar videos. The key difference is that we introduce an explicit trajectory-aware encoding mechanism to exploit the trajectory information of surviving targets to enhance the visual features. Specifically, we construct a trajectory-aware encoder (TAE) to enhance single-frame visual features using the trajectory information of surviving targets. The TAE contains a trajectory feature-encoding module, a visual feature-encoding module, and a frame-trajectory cross-modal attention module. The trajectory feature-encoding module extracts both object-level and point-level trajectory representations, while the visual feature-encoding module encodes the Cartesian radar-image features obtained from the backbone network. Then, the frame-trajectory cross-modal attention module fuses the visual feature encoding with the trajectory feature encoding to obtain a reconstructed encoding matrix. Finally, the enhanced encoding matrix is decoded by the Transformer decoder to produce the tracking results. In addition, we present a training method suitable for the proposed network in which historical target positions from annotation files are used to establish prior tracks that guide the learning of trajectory-aware feature enhancement. Compared with TrackFormer, the main novelty of TrajE2E-MOT lies in the trajectory-aware encoder, the object-level and point-level trajectory feature representations, and the frame-trajectory cross-modal attention mechanism for radar visual feature enhancement. The contributions of this paper are summarized as follows:

(1) We present an approach that uses the real-time trajectories of targets to enhance the single-frame visual feature encoding, thereby mitigating the limitations of end-to-end tracking in radar videos. This method addresses the challenges posed by limited visual features and their significant variations during long-term tracking.
(2) A Trajectory-Aware Encoder (TAE) is proposed. Within the TAE, the real-time trajectories of surviving targets are encoded, then utilized to reconstruct single-frame visual feature encodings via a frame-trajectory cross-modal attention module.
(3) A training method is designed for the trajectory-aware end-to-end tracking model, incorporating specific training strategies to ensure the effective collaborative learning of trajectories and visual features.

The remainder of this paper is organized as follows. Section 2 reviews related studies on visual multi-target tracking and multi-target tracking in radar video. Section 3 presents the proposed TrajE2E-MOT framework. Section 4 describes the radar video sequence dataset, experimental settings, comparison results, ablation studies, computational complexity and runtime analysis and limitation analysis. Finally, Section 5 concludes the paper and discusses future research directions.

2. Related Work

Relevant prior work includes studies of visual multi-target tracking and multi-target tracking in radar video.

2.1. Visual Multi-Target Tracking

The research on visual multi-target tracking mainly involves two paradigms: detection-based tracking [5–8] and joint detection and tracking [1,9–11].

For detection-based tracking, the detection and tracking modules are decoupled and independently optimized. SORT [5] is a classic algorithm under this paradigm that obtains the detection boxes of all targets in the current frame through an independent detector. In the association stage, the Hungarian algorithm is used to match the predictions of the Kalman filter with the detection information. As an improved version of SORT, Deepsort [6] incorporates extracted appearance features for matching during the association stage, and it performs more robustly. However, in the association stage of DeepSort, a serial approach is adopted, without considering the complementarity between two features. Inspired by this, TOPICTrack [12] proposes a parallel association method, simultaneously using motion features and appearance features as matching metrics, and adaptively selects the optimal features to reduce false alarms. Bytetrack [7] focuses primarily on the utilization of detection boxes. Through a simple and efficient two-stage association strategy, it fully utilizes the high and low score detection boxes and can stably track, even when the appearance features are unreliable, further improving the tracking ability in scenarios with target occlusion and blurry scenes. To cope with different scenarios, Qin, Z et al. proposed a generalized tracking model called Generaltrack [8], which is based on the analysis of the attributes of various tracking scenarios and has strong generalization ability.

The detection-based tracking paradigm has its own flexibility, but the accumulation of detection errors will directly affect the tracking results. Another popular paradigm is the joint detection and tracking method. In this paradigm, detection and tracking share a network to generate detection boxes and Re-Identification (REID) feature embeddings. FairMOT adopts an anchor-free design, improving the unfairness problem in early algorithms such as JDE [13], effectively improving the quality of REID features, but its association stage also relies on Kalman filtering for motion prediction and uses the Hungarian algorithm for matching, which encounters difficulty in handling long-term tracking in complex scenarios. CenterTrack [9] directly treats the target as a point and predicts the offset of the target to the next frame through the network to achieve association. End-to-end tracking is also a method under the joint detection and tracking paradigm. Trackformer [2] and MOTR [3] have shown great potential, representing current research hotspots. This route attempts to achieve conversion from video input to tracking results using a single model, discarding the manually designed association rules. Trackformer is designed based on the Transformer architecture, introducing the concept of a tracking query. By maintaining a tracking query for each target, passing it between adjacent frames to query the position of the corresponding target, and combining a set of learnable object queries to query newly emerging targets, it achieves end-to-end tracking without manually designed association rules. MOTR eliminates the object query and unifies it as a track query. However, these end-to-end methods have slow reasoning speeds and currently encounter difficulty in meeting the tracking deployment requirements for high-frame-rate videos (such as 20 fps or higher). Nevertheless, this limitation is alleviated in radar video scenarios. Taking navigation radar as an example, the data update cycle is usually on the order of seconds, providing a rare deployment opportunity for more computationally demanding end-to-end tracking models.

2.2. Multi-Target Tracking in Radar Video

In research on multi-target tracking in radar video, methods based on kernel correlation filters and their variants [14–16] mainly rely on the appearance characteristics of the tracking targets, but they are relatively sensitive to scale changes. To further enhance

the ability to discriminate against targets, some studies have attempted to jointly utilize motion features and appearance features. For instance, the authors of [17] simultaneously employed the Mahalanobis distance of the target and the cosine distance of the appearance features during the data association stage. The authors of [18] also jointly used motion features and appearance features for tracking, but the proposed appearance model is sensitive to the changes in the target's appearance. The study reported in [19] constructed four feature quantities—size, amplitude, shape, and location—which provided a more comprehensive feature representation, but the weight coefficients between different features were adjusted empirically and had low universality. Although [20] extended the state vector to six dimensions and employed CIOU [21] as the association metric, the method's performance remains susceptible to errors originating from the detection stage. The authors of [22,23] introduced AIS information for assistance when estimating a ship's heading and speed, but this improvement was ineffective when the target's AIS signal could not be obtained. ET-JPDA [24] is an extension of the traditional probabilistic data association algorithm, capable of achieving multi-target tracking in radar videos. However, it also relies on the motion model of the target. GP-PMBM [25] uses Gaussian processes as the target model, in combination with the Poisson Multi-Bernoulli Mixture (PMBM) tracking framework. It has advantages in target contour estimation and heading estimation, but it suffers from the problem of local optimal solutions and is highly sensitive to prior selection. Methods proposed in existing research make it difficult to break away from the reliance on manually designed correlation rules, which are prone to parameter mismatch in different tracking scenarios. Consequently, it is of significant interest to develop end-to-end multi-object tracking methods suitable for radar video scenarios to enhance tracking robustness and intelligence.

3. Proposed Method

We present TrajE2E-MOT, a trajectory-aware end-to-end multi-object tracking model for radar video sequences. This model is based on the Trackformer [2] architecture and incorporates the real-time trajectory information of surviving targets to enhance the visual feature encoding. This section introduces the processing procedure for radar video data and how to use trajectory information to enhance the visual feature encoding. Additionally, we explain how to train this end-to-end model and apply it for tracking.

3.1. Framework Overview

As shown in Figure 2, the overall model framework consists of two modules: the trajectory-aware encoder (TAE) and the decoder. The trajectory-aware encoder includes three sub-modules: the trajectory encoder module, the visual feature encoder module, and the frame-trajectory cross-modal attention module. The details of the decoder ae reported in [2]. In the trajectory encoder, the real-time trajectory information of surviving targets is introduced, and in this module, trajectory features are extracted for encoding to obtain a trajectory feature matrix. The visual encoder module achieves visual feature encoding on the radar video frame and outputs a visual feature matrix. In the frame-trajectory cross-modal attention module, the visual feature encoding is reconstructed using the trajectory feature encoding, thereby obtaining the reconstructed encoding matrix after trajectory enhancement. In the decoder module, object querying and track querying are completed for the reconstructed encoding matrix, and the tracking results of the current frame are obtained.

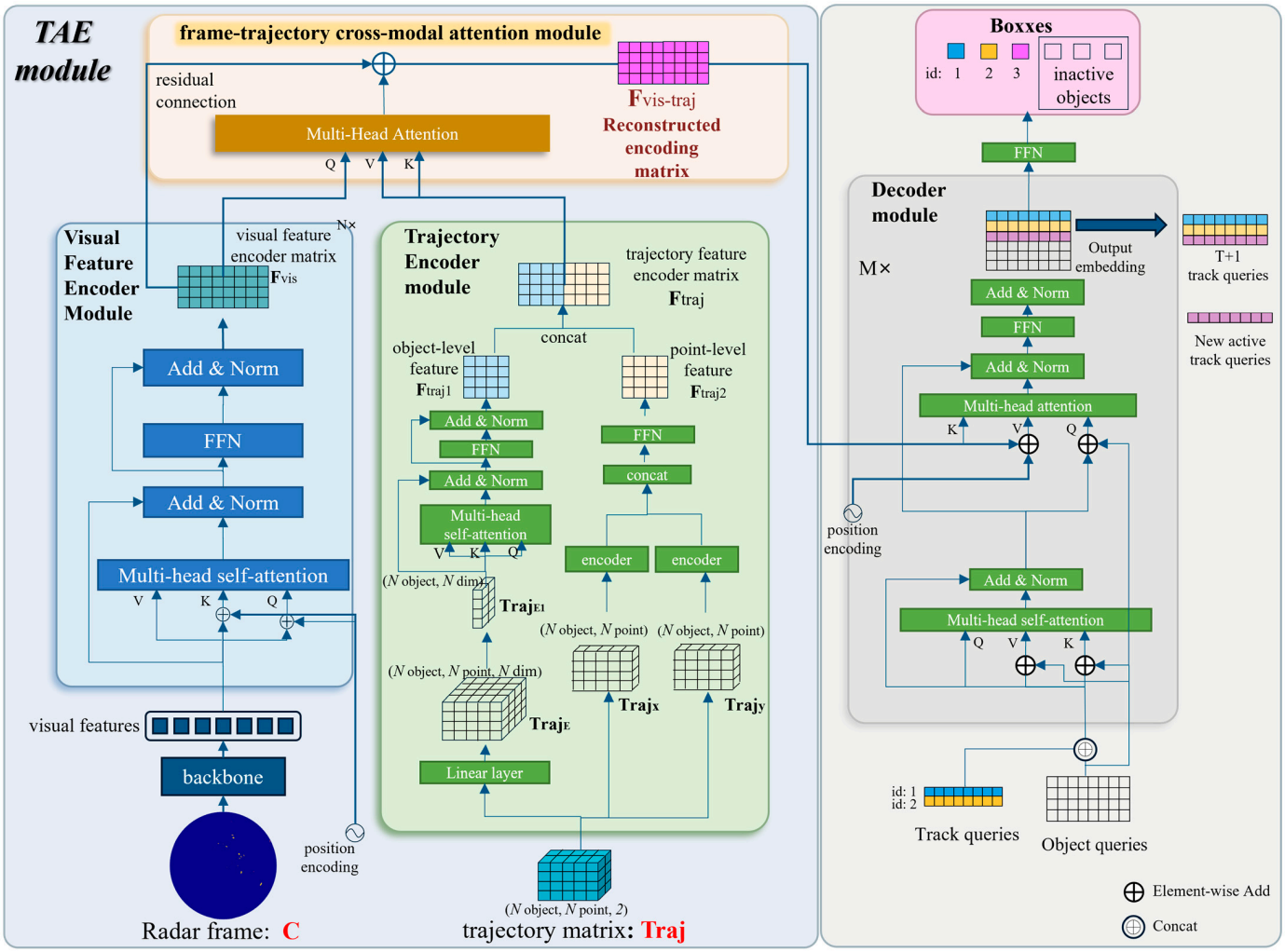


Figure 2. Diagram of the proposed method.

3.2. Radar Video Data

The radar generates a data matrix ($\mathbf{P} \in \mathbb{R}^{r \times \theta}$) after each scan cycle, which represents the data in the polar coordinate system, where r and θ denote the range and angle dimensions, respectively. $\mathbf{P}$ is transformed into Cartesian coordinates for further processing:

$$\mathbf{C} = \Lambda(\mathbf{P}), \mathbf{C} \in \mathbb{R}^{h \times w} \tag{1}$$

where h and w denote the height and width of $\mathbf{C}$ and Λ represents the transformation process. Specifically, Λ consists of the following steps. First, each radar cell is mapped from the polar coordinate system to the Cartesian coordinate system according to its range (r) and azimuth angle (θ):

$$x = r \sin\theta, y = r \cos\theta \tag{2}$$

After coordinate mapping, a Cartesian grid is generated. In the implementation, the resolution of the Cartesian output grid is 1500 × 1500 pixels. Bilinear interpolation is then used to estimate the point data at each Cartesian grid position based on the surrounding polar-coordinate samples.

3.3. Trajectory-Aware Encoder (TAE)

3.3.1. Trajectory Encoder Module

We reconstruct and enhance the visual feature encoding of radar video frames through the trajectory information of the target. Specifically, during the tracking process, the trajectories of surviving targets are cached and updated in real time at each step, maintaining a fixed number of steps. The trajectory matrix of the current frame is represented as $\mathbf{Traj} \in \mathbb{R}^{n_{\text{object}} \times n_{\text{point}} \times 2}$. n_{object} refers to the number of targets surviving more than n_{point} steps. Each track point has two dimensions. Two branches are designed to separately extract its object-level features and point-level features.

In branch 1, after passing through a fully connected layer, the dimension of the track points in **Traj** is increased to enrich the kinematic feature representation of the target, resulting in $\mathbf{Traj}_{\text{E}} \in \mathbb{R}^{n_{\text{object}} \times n_{\text{point}} \times n_{\text{dim}}}$, where n_{dim} represents the expanded dimension of track points. The average of the second dimension of $\mathbf{Traj}_{\text{E}}$ is calculated, resulting in $\mathbf{Traj}_{\text{E1}} \in \mathbb{R}^{n_{\text{object}} \times n_{\text{dim}}}$:

$$\mathbf{Traj}_{\text{E1}}(i,k) = \frac{1}{n_{\text{point}}} \sum_{j=1}^{n_{\text{point}}} \mathbf{Traj}_{\text{E}}(i,j,k) \tag{3}$$

where $\mathbf{Traj}_{\text{E}}(i,j,k)$ represents the value at index (i,j,k) in $\mathbf{Traj}_{\text{E}}$ and $\mathbf{Traj}_{\text{E1}}(i,k)$ represents the value at index (i,k) in $\mathbf{Traj}_{\text{E1}}$ ($i \in [1, n_{\text{object}}]$, $j \in [1, n_{\text{point}}]$, $k \in [1, n_{\text{dim}}]$). $\mathbf{Traj}_{\text{E1}}$ is input into multi-head attention and FFN [2] to generate an object-level feature matrix ($\mathbf{F}_{\text{traj1}} \in \mathbb{R}^{n_{\text{object}} \times n_{\text{dim}}}$).

In branch 2, the x and y dimensional data in the **Traj** are first extracted separately to obtain $\mathbf{Traj}_x \in \mathbb{R}^{n_{\text{object}} \times n_{\text{point}}}$ and $\mathbf{Traj}_y \in \mathbb{R}^{n_{\text{object}} \times n_{\text{point}}}$. Then, these two matrices are encoded and concatenated and, finally, input into the FFN to obtain a point-level feature matrix ($\mathbf{F}_{\text{traj2}} \in \mathbb{R}^{n_{\text{object}} \times n_{\text{dim}}}$). The final trajectory feature matrix is obtained by concatenating $\mathbf{F}_{\text{traj1}}$ and $\mathbf{F}_{\text{traj2}}$:

$$\mathbf{F}_{\text{traj}} = \text{Concat}\left(\mathbf{F}_{\text{traj1}}, \mathbf{F}_{\text{traj2}}\right), \mathbf{F}_{\text{traj}} \in \mathbb{R}^{(n_{\text{object}} + n_{\text{object}}) \times n_{\text{dim}}} \tag{4}$$

where Concat represents the matrix concatenation operation.

3.3.2. Visual Feature Encoder Module

In this module, the frame data (**P**) are first transformed into **C** by (1). **C** can be regarded as single-channel image data. Then, **C** is input into the backbone to obtain the visual features. These features are then added to the fixed positional encoding [26] and processed through the encoder layer, which includes a multi-head attention mechanism and feedforward neural network. The specific structural details of this encoder layer are reported in [27]. This results in a visual feature encoding matrix ($\mathbf{F}_{\text{vis}} \in \mathbb{R}^{m \times n_{\text{dim}}}$).

3.3.3. Frame-Trajectory Cross-Modal Attention Module

In this module, the trajectory feature ($\mathbf{F}_{\text{traj}}$) provides real-time movement information. The cross-attention mechanism is used to establish an attention relationship between the visual feature ($\mathbf{F}_{\text{vis}}$) and trajectory feature ($\mathbf{F}_{\text{traj}}$), enabling the model to focus on relevant regions, enhance target perception, and thereby optimize the current frame encoding. The reconstructed encoding matrix is obtained through a residual connection:

$$\mathbf{F}_{\text{vis-traj}} = \psi(\mathbf{F}_{\text{vis}}) + \mathbf{F}_{\text{vis}} \tag{5}$$

where $\psi(\cdot)$ represents the cross-attention calculation and specifically includes the following steps:

$$\mathbf{Q} = \mathbf{F}_{\text{vis}}\mathbf{W}^{Q}, \mathbf{K} = \mathbf{F}_{\text{traj}}\mathbf{W}^{K}, \mathbf{V} = \mathbf{F}_{\text{traj}}\mathbf{W}^{V} \tag{6}$$

$$\mathbf{Z}_i = \text{softmax}(\frac{\mathbf{Q}_i\mathbf{K}_i}{\sqrt{d_k}})\mathbf{V}_i, i = 1, \cdots, N \tag{7}$$

$$\mathbf{Z} = \text{Concat}(\mathbf{Z}_i, \cdots, \mathbf{Z}_N)\mathbf{W}_o \tag{8}$$

In Equation (6), the query matrix $\mathbf{Q}$ is generated from the current-frame visual feature matrix ($\mathbf{F}_{\text{vis}}$), while the key matrix ($\mathbf{K}$) and value matrix ($\mathbf{V}$) are generated from the trajectory feature matrix ($\mathbf{F}_{\text{traj}}$). This design allows the visual features of the current radar frame to query the trajectory features of surviving targets. In this way, trajectory information is used as motion guidance to enhance the current-frame visual representation. $\mathbf{W}^{Q}, \mathbf{W}^{K}, \mathbf{W}^{V}, \mathbf{W}_o \in \mathbb{R}^{n_{\text{dim}} \times n_{\text{dim}}}$ are linear projection layers. In Equation (7), $\mathbf{Z}_i$ refers to the output of the i-th attention head. The attention weights are calculated according to the similarity between the projected visual features and trajectory features. These weights are then used to aggregate the trajectory value features so that the motion information can be introduced into the current-frame visual encoding. $\sqrt{d_k}$ is a scaling factor to prevent the dot-product result from being too large and causing the Softmax gradient to disappear. In Equation (8), the outputs of all attention heads are concatenated and projected by $\mathbf{W}_o$. The final cross-attention output ($\mathbf{Z} \in \mathbb{R}^{m \times n_{\text{dim}}}$) has the same dimension as the visual feature matrix ($\mathbf{F}_{\text{vis}}$). Therefore, it can be combined with $\mathbf{F}_{\text{vis}}$ through the residual connection in Equation (5), producing the reconstructed encoding matrix of the visual feature matrix ($\mathbf{F}_{\text{vis-traj}}$).

3.4. TrajE2E-MOT Training

Algorithm 1 shows the key steps of our TrajE2E-MOT training. During the training process, the loss of the current frame (t) measures the differences between the outputs of all queries (including object queries ($\mathbf{Q}^{t}_{\text{object}}$) and track queries ($\mathbf{Q}^{t}_{\text{track}}$)) and their corresponding ground-truth targets in terms of categories and bounding boxes. The track query is maintained for the survival target and is directly associated between frames. Therefore, training is based on the two adjacent frames (t and $t-1$), that is, the output embedding generated by object queries in frame ($t-1$) are used as track queries ($\mathbf{Q}^{t}_{\text{track}}$) for frame t. As step 1 in Algorithm 1 shows, during the encoding process of the $\mathbf{C}^{t-1}$ at frame $t-1$, $\mathbf{Traj}^{t-2} \in \mathbb{R}^{n^{t-2}_{\text{object}} \times n_{\text{point}} \times 2}$ is required, which represents the trajectories of surviving targets in frame $t-2$. Specifically, the positions of all targets in frame $t-2$ are obtained from $t-3$ to $t-2-n_{\text{point}}$, and the target tracks with a length of less than n_{point} are filtered out, obtaining $\mathbf{Traj}^{t-2}$. Then, $\mathbf{C}^{t-1}$ and $\mathbf{Traj}^{t-2}$ are input into the trajectory encoder module and the visual feature encoder module of our trajectory-aware encoder (TAE), respectively. The output embeddings obtained from the object queries are used as the track queries for frame t. Similarly, for the encoding process of $\mathbf{C}^{t}$ at frame t, $\mathbf{Traj}^{t-1}$ is required. In step 7 of Algorithm 1, λ_{cls} and λ_{reg} are weights of class loss ($loss_{\text{cls}}$) and bounding-box loss ($loss_{\text{reg}}$) respectively. $loss_{\text{cls}}$ and $loss_{\text{reg}}$ are defined in [12].

Algorithm 1 TrajE2E-MOT training

Require:
radar video data at frame $t-1$ and t: $\mathbf{C}^{t-1}$, $\mathbf{C}^{t}$;
trajectories of surviving targets at frame $t-2$ and $t-1$: $\mathbf{Traj}^{t-2}$, $\mathbf{Traj}^{t-1}$;
output: the weight parameters of the trained model: θ^{*}

$\mathbf{F}^{t-1}_{\text{vis-traj}} \leftarrow \text{TAE}\left(\mathbf{C}^{t-1},\ \mathbf{Traj}^{t-2}\right)$
$\mathbf{Q}^{t}_{\text{track}} \leftarrow \text{Decoder}\left(\mathbf{F}^{t-1}_{\text{vis-traj}}, \mathbf{Q}^{t}_{\text{object}}\right)$
$\mathbf{F}^{t}_{\text{vis-traj}} \leftarrow \text{TAE}\left(\mathbf{C}^{t},\ \mathbf{Traj}^{t-1}\right)$
$\mathbf{embedding} \leftarrow \text{Decoder}\left(\mathbf{F}^{t}_{\text{vis-traj}}, \mathbf{Q}^{t}_{\text{track}}, \mathbf{Q}^{t}_{\text{object}}\right)$
$\hat{\mathbf{y}} \leftarrow \text{FFN_cls}(\mathbf{embedding})$
$\widehat{\mathbf{bbox}} \leftarrow \text{FFN_reg}(\mathbf{embedding})$
$loss \leftarrow \lambda_{\text{cls}} loss_{\text{cls}}(\hat{\mathbf{y}}, \mathbf{y}) + \lambda_{\text{reg}} loss_{\text{reg}}\left(\widehat{\mathbf{bbox}}, \mathbf{bbox}\right)$
backpropagate($loss$)
optimizer.zero_grad()
$\theta \leftarrow$ optimizer.step()
Repeat 1–10 until convergence
$\theta^{*} \leftarrow \theta$

4. Experiments and Discussion

4.1. Dataset and Evaluation Metrics

4.1.1. Radar Video Sequence Dataset

To the best of our knowledge, there is currently no publicly available radar video dataset suitable for network model training. To experimentally validate the method proposed in this paper, we conducted real-world data collection. Specifically, radar video data were collected in the sea area near Yantai Port in China, containing both moving targets and stationary targets. Each frame of data has an update cycle of about 3 s, and the detection range is 6 nautical miles.

The data were collected in a real maritime surveillance scenario. The environmental conditions include sea-clutter interference, port-background clutter, and the coexistence of moving and stationary targets. The dataset also contains challenging tracking situations, such as target maneuvering, target overlap, and occlusion.

The raw radar data are polar coordinate matrices transformed into Cartesian radar images for network input. The Cartesian radar images were manually annotated with bounding boxes and identity labels. Each target was assigned a unique identity, and the same identity was maintained across consecutive frames when the target remained visible. We conducted a manual recheck of the annotation results to ensure the consistency of target locations and identities. To avoid information leakage between adjacent radar frames, the dataset was split according to temporal sequences rather than by randomly selecting individual frames. The training set contains 801 frames, and the test set contains 525 frames.

To provide a more detailed description of the experimental conditions, target characteristics, and benchmark protocol, the main information of the radar video sequence dataset is summarized in Table 1. The moving targets are typical maritime ships such as cargo ships, passenger ships, fishing ships, sightseeing ships, and so on. In this study, these targets are treated as generic maritime radar targets. The annotation and evaluation focus on target localization and identity continuity across frames.

Table 1. Details of the radar video sequence dataset.

Item	Description
Collection area	Sea area near Yantai Port, China
Radar specification	Pulse compression radar
Scan period	About 3 s
Detection range	6 nautical miles
Raw radar format	Polar coordinate matrix
Training frames	801
Testing frames	525
Number of annotated target trajectories	71
Average trajectory length of moving targets (frames)	359
Maximum trajectory length (frames)	801
Percentage of target overlaps/occlusions (%)	7.4
Percentage of frames with overlaps/occlusions (%)	63.2
Maneuvering-track ratio (%)	39.4
Stationary-track ratio (%)	60.6
Annotation type	Bounding boxes + identity labels

The experiments were conducted using the PyTorch deep learning framework (version 1.4.0 with CUDA 10.0) on a computing station equipped with an NVIDIA RTX 2080Ti GPU (NVIDIA Corporation, Santa Clara, CA, USA), an Intel(R) Core (TM) i7-9700 CPU (Intel Corporation, Santa Clara, CA, USA), and 16 GB of RAM, running a 64-bit Ubuntu 20.04.2 operating system.

4.1.2. Evaluation Metrics

Multiple Object Tracking Accuracy (MOTA) [28] is widely used to evaluate the overall performance of multi-object tracking, formally defined as follows:

$$\mathrm{MOTA} = 1 - \frac{\mathrm{FN} + \mathrm{FP} + \mathrm{IDs}}{\mathrm{GT}} \tag{9}$$

where FN indicates the number of false negatives, FP indicates the number of false positives, IDs indicates the number of ID switches and GT indicates the number of ground-truth objects. Another important multi-object tracking evaluation metric is Multiple Object Tracking Precision (MOTP), which is used to quantify localization accuracy. The calculation formula is expressed as follows:

$$\mathrm{MOTP} = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t} \tag{10}$$

where c_t denotes the total number of matched object pairs in frame t and $d_{t,i}$ denotes the distance between the i-th matched predicted box and the ground-truth box. The Mostly Tracked (MT) and Mostly Lost (ML) metrics are also used to evaluate trajectory continuity by measuring the proportion of ground-truth tracks maintained for $\geq$80% (MT) and <20% (ML) of their duration, respectively.

4.2. Comparative Experiments

4.2.1. Overall Comparison

In this section, the proposed method was evaluated against several state-of-the-art multi-object tracking algorithms designed for natural videos (Centertrack [9], FairMOT [1], Bytetrack [7], TrackTrack [29], Generaltrack [8], MOTR [3], TransTrack [30], and TrackFormer [2]) and conventional radar video multi-object tracking methods (ET-JPDA [24] and GP-PMBM [25]). To ensure a fair comparison, all compared deep learning-based methods,

including CenterTrack, FairMOT, ByteTrack, TrackTrack, GeneralTrack, MOTR, TransTrack, and TrackFormer, were retrained on the radar video training set constructed in this work. ET-JPDA and GP-PMBM are model-driven methods and required no training. All compared methods were evaluated on the same radar video test set using the same evaluation metrics. As Table 2 shows, in addition to the MOTP and FP metrics, the proposed algorithm outperforms other algorithms on all other metrics. Specifically, the MOTA metric of the proposed model reached 90.09, which is 3.50 higher than that of TransTrack, indicating that the proposed model achieves excellent comprehensive performance. The MOTP metric of the proposed model was 0.5842, which is very close to the 0.5717 of Bytetrack (only 0.0125 higher and 0.1 lower than others), demonstrating excellent localization accuracy. The greatest advantage is mainly reflected in IDs, ML, MT, and FN. These metrics reflect the model's ability to continuously track the target. Specifically, the IDs of the proposed method was 12, which is lower than that of all compared methods, and the ML metric of the proposed model was 0, which is the same as that of GP-PMBM and superior to that of other models. The MT metric of the proposed model was 28, which is better than that of other models. The FN of the proposed model was only 79, and the best performance among other models was 383. The proposed model enhances continuous tracking by incorporating real-time track information of batch targets, allowing it to focus on the target's likely positions. This improves both tracking completeness (i.e., the target is followed most of the time) and ID continuity (i.e., fewer identity switches). These improvements are reflected in better ML, MT, and FN metrics. GeneralTrack achieved a relatively low ML value, but its IDs and FP were high, leading to a much lower MOTA. However, it is slightly weak in target ID consistency, and its false-alarm rate was high, with FP reaching 2474, which is the highest among all experimental models, resulting in a lower comprehensive MOTA metric, indicating that it still has shortcomings in the radar video target-tracking scenario. Bytetrack also performed very well; its simple structure enables it to achieve good generalization in other scenarios, and its MOTP and FP metrics even exceeded those of the proposed model, but other indicators were not as good as those of the proposed model. The performance of Bytetrack in radar video target tracking can be further improved. The Transformer-based methods, including MOTR, TransTrack, and TrackFormer, also showed relatively competitive performance. The shortcomings of the proposed model should also be noted. The proposed model suffers from a high false-alarm rate of FP = 915, which is significantly higher than FairMOT's (FP = 210), suggesting that its false-alarm control needs improvement. The main reason for this result is the influence of background noise in radar images, especially sea clutter, which can lead to false alarms. Such clutter responses often exhibit strong echoes in radar images and may appear similar to real targets, making them difficult to distinguish from true targets. The proposed TAE is mainly designed to reconstruct and enhance visual feature encoding by incorporating trajectory information, thereby improving target continuity in long-term tracking. Therefore, the improvements are more significant in IDs, ML, and MT, while no explicit clutter-suppression module is introduced in the current framework. Further improvements in the visual encoding network, optimized track initialization, and adaptive confidence-threshold adjustment are potential strategies for reducing false positives. These observations provide useful guidance for future research. In addition, conventional radar video multi-object tracking methods exhibit significant limitations in false-alarm suppression and missed-detection control, leading to relatively low MOTA scores. A significant reason for these shortcomings is the lack of utilization of the visual features of radar video targets, which, in turn, motivates our exploration of vision-based solutions.

Table 2. Comparison with state-of-the-art methods.

Method	MOTA ↑	MOTP ↓	IDs ↓	ML ↓	MT ↑	FN ↓	FP ↓
Centertrack	83.67	6.8194	157	3	21	862	639
FairMOT	78.65	0.6761	21	6	18	1937	**210**
Bytetrack	84.08	**0.5717**	23	8	19	1355	238
Generaltrack	66.32	0.7898	159	2	22	787	2474
TrackTrack	79.37	0.8231	21	3	20	988	1086
MOTR	82.38	0.6483	23	2	25	401	1365
TransTrack	86.59	0.7016	27	2	27	383	952
TrackFormer	85.87	0.6992	18	2	27	412	1005
ET-JPDA	42.67	6.6213	26	4	24	2114	3681
GP-PMBM	33.79	2.0252	68	**0**	25	547	6107
TrajE2E-MOT (Our)	**90.09**	0.5842	**12**	**0**	**28**	**79 (0.8%)**	915 (9%)

↑ indicates that higher is better and ↓ indicates that lower is better. The best values are shown in bold.

4.2.2. Comparative Analysis Under Varying Tracking Lengths

We conducted experiments on the performance of each model when tracking different numbers of frames to verify that the model can achieve effective tracking in both short-term and long-term tracking scenarios. Specifically, we set up several sets of comparative experiments with tracking-frame numbers of 50, 100, 200, 300, 400, and 500. The experimental results are shown in Tables 3 and 4, with MOTA@50 and MOTP@50 representing the MOTA and MOTP values of the corresponding model when tracking the 50th frame respectively.

Table 3. Comparative analysis of MOTA metrics under varying tracking lengths.

Method	MOTA@50	MOTA@100	MOTA@200	MOTA@300	MOTA@400	MOTA@500
Centertrack	89.25	89.01	88.45	86.96	85.2	84.36
FairMOT	85.85	82.87	83.51	82.07	79.85	79.01
Bytetrack	84.47	84.53	85.73	84.28	84.01	84.32
Generaltrack	62.20	63.67	70.66	70.30	67.99	66.76
TrackTrack	82.81	82.37	83.90	82.27	80.37	79.76
MOTR	84.74	86.70	85.68	86.30	85.89	82.71
TransTrack	85.66	88.78	89.44	89.69	88.62	86.95
TrackFormer	84.38	88.09	88.26	88.90	88.00	86.45
ET-JPDA	40.63	41.18	43.59	44.12	42.05	40.09
GP-PMBM	47.33	41.74	42.96	42.39	38.60	38.14
TrajE2E-MOT (Our)	**94.67**	**95.29**	**94.99**	**94.53**	**92.94**	**91.01**

The best values are shown in bold.

Table 4. Comparative analysis of MOTP metrics under varying tracking lengths.

Method	MOTP@50	MOTP@100	MOTP@200	MOTP@300	MOTP@400	MOTP@500
Centertrack	6.9544	6.7426	6.7947	6.7775	6.8196	6.8267
FairMOT	0.6504	0.7212	0.7363	0.7007	0.6735	0.6736
Bytetrack	**0.5444**	**0.6166**	**0.6088**	**0.5498**	**0.5321**	**0.5563**
Generaltrack	0.7694	0.8946	0.8836	0.8220	0.7775	0.7805
TrackTrack	0.8475	0.9746	0.9435	0.8591	0.8144	0.8137
MOTR	0.6425	0.7371	0.7152	0.6725	0.6479	0.6573
TransTrack	0.6555	0.7584	0.7475	0.7201	0.6965	0.7022
TrackFormer	0.6683	0.7740	0.7658	0.7375	0.7140	0.7199
ET-JPDA	5.5703	6.6763	8.5801	7.7741	7.2770	7.0515
GP-PMBM	2.9216	3.1161	2.6741	2.5059	2.4275	2.5525
TrajE2E-MOT (Our)	**0.5540**	**0.6080**	**0.6149**	**0.5972**	**0.5808**	**0.5826**

The best values are shown in bold.

In Table 3, it can be seen that the MOTA values of the proposed model in each experimental group are all higher than those of other models. Although the MOTA values slightly decrease as the tracking-frame number increases, they always remain above 90, while the MOTA values of other models are all below 90. Centertrack has a relatively high MOTA index in short-term tracking (close to 90), but as the tracking-frame number increases, the MOTA value drops from 89.25 to below 85 in long-term tracking. FairMOT and TrackTrack have MOTA values that decrease to below 80 in long-term tracking scenarios as the tracking-frame number increases. Bytetrack's MOTA is relatively stable, with little variation in short-term and long-term tracking, although always remaining below 85, which is much lower than the proposed model. Generaltrack has poor false-alarm performance, resulting in consistently poor MOTA indicators. The Transformer-based MOT methods, including MOTR, TransTrack, and TrackFormer, show relatively strong performance compared with several conventional methods. In particular, TransTrack and TrackFormer achieve relatively high MOTA values. However, their MOTA values are still lower than those of the proposed TrajE2E-MOT under all tracking lengths.

As shown in Table 4, the proposed model also achieves competitive performance on the MOTP metric. The MOTP values are all below 0.62—very close to that of Bytetrack. Only Bytetrack and the proposed model are both below 0.7 in short-term and long-term tracking, and both can maintain excellent results below 0.62. However, the MOTA value of Bytetrack is significantly lower than that of the proposed model. In general, the proposed model is better able to handle radar video scenarios with multiple targets than Bytetrack. Centertrack has difficulty accurately regressing the position of small targets, with a high MOTA value but poor position accuracy. In radar video target tracking, there is still room for optimization for the small target problem. FairMOT, GeneralTrack, and TrackTrack show moderate performance, with average MOTP values that are 17.47%, 39.3%, and 48.49% higher than those of the proposed model. For the Transformer-based methods, MOTR, TransTrack, and TrackFormer also show reasonable localization performance, but their MOTP values are generally higher than those of the proposed method.

4.3. Ablation Study

4.3.1. Ablation Study on TAE Components

In this section, we conducted an ablation experiment to verify the effectiveness of the proposed TAE module. The experiments were divided into four groups: the baseline without trajectory encoding and frame-trajectory cross-modal attention, the baseline with only the object-level trajectory branch, the baseline with only the point-level trajectory branch, and the complete TAE module with both trajectory branches. Since the cross-modal attention module is used to fuse trajectory features with visual features, it is enabled when either trajectory branch is introduced. The experimental results are shown in Table 5.

Table 5. Ablation study results of TAE components.

Baseline	Object-Level Branch	Point-Level Branch	Cross-Modal Attention	MOTA ↑	MOTP ↓	IDs ↓	ML ↓	MT ↑	FN ↓	FP ↓
√				85.87	0.6992	18	2	27	412	1005
√	√		√	88.88	0.6322	15	1	**28**	107	1007
√		√	√	88.83	0.6297	14	1	**28**	111	1009
√	√	√	√	**90.09**	**0.5842**	**12**	**0**	**28**	**79**	**915**

√ indicates that the corresponding module is used. ↑ indicates that higher is better and ↓ indicates that lower is better. The best values are shown in bold.

Compared with the baseline, both trajectory branches improve the tracking performance. When only the object-level trajectory branch is used, MOTA increases from 85.87 to 88.88, IDs decrease from 18 to 15, and FN decreases from 412 to 107. When only the point-level trajectory branch is used, MOTA reaches 88.83, IDs decrease to 14, and FN decreases to 111. These results show that both object-level and point-level trajectory information are effective in improving tracking continuity.

The complete TAE module achieves the best overall performance. Compared with the baseline, the MOTA value increased from 85.87 to 90.09; the MOTP value decreased from 0.6992 to 0.5842; IDs decreased from 18 to 12; ML, FN, and FP decreased from 2, 412, and 1005 to 0, 79, and 915 respectively; and MT increased by 1. By introducing trajectory information in the TAE module and using the trajectory information to enhance the radar video, the encoding quality of the radar video was improved, enabling the model to track the target more accurately. In long-term tracking scenarios, the tracking ability of the model can still be enhanced.

4.3.2. Sensitivity Analysis of Trajectory Length

To further analyze the influence of trajectory length on the proposed TAE module, we conducted a sensitivity analysis by setting the number of trajectory points to 1, 3, 5, 7, and 10. The experimental results are shown in Table 6.

Table 6. Performance evaluation of different trajectory lengths.

Trajectory Length	MOTA ↑	MOTP ↓	IDs ↓	ML ↓	MT ↑	FN ↓	FP ↓
1	86.90	0.6359	16	1	27	369	945
3	89.51	0.6094	15	1	27	188	862
5	90.09	**0.5842**	**12**	**0**	**28**	79	915
7	**91.20**	0.6002	13	1	27	83	**797**
10	90.02	0.5948	**12**	1	**28**	**72**	929

↑ indicates that higher is better and ↓ indicates that lower is better. The best values are shown in bold.

As shown in Table 6, when the trajectory length is 1, the model can only use very limited motion information, resulting in a relatively low MOTA and a high number of IDs. As the trajectory length increases from 1 to 5, the tracking performance improves. Specifically, MOTA increases from 86.90 to 90.09, IDs decrease from 16 to 12, ML decreases from 1 to 0, and FN decreases from 369 to 79. This indicates that using a moderate number of trajectory points can effectively provide useful motion information for improved tracking continuity and reduced missed detections.

When the trajectory length is further increased to 7 or 10, the model still maintains strong performance, and the MOTA values remain above 90. The best MOTA is obtained when the trajectory length is 7. However, longer trajectories do not consistently improve all metrics. For example, compared with a trajectory length of 5, the settings of 7 and 10 slightly increase ML or reduce MT, and the MOTP values are also higher. This might be due to the excessively long track length, resulting in information redundancy, and it has little effect on further improvements in the model's performance. Considering the overall performance, the trajectory length was set to 5 in the main experiments. This setting achieves the best MOTP, the lowest IDs, the lowest ML, and the highest MT while maintaining a high MOTA and a low FN value.

4.4. Visual Verification of Results

In this section, we present a visual representation of the tracking results. Figure 3 shows the regions where targets are present. Figure 3a–f present the tracking results at frames 30, 50, 150, 250, 350, and 450, respectively. The tracking boxes are represented by

blue rectangles, and the ID value above each rectangle indicates the tracking identification number. The red lines represent the real-time trajectory of the target. It can be observed that both stationary and moving targets coexist in the tracking scene, with varying target sizes: large targets appear as large patches in the radar image, while small targets appear as small patches. The tracking results demonstrate that most moving targets are completely tracked, from their appearance to disappearance within the detection range, maintaining continuous trajectories. For stationary targets, the algorithm also achieves consistent and stable tracking.

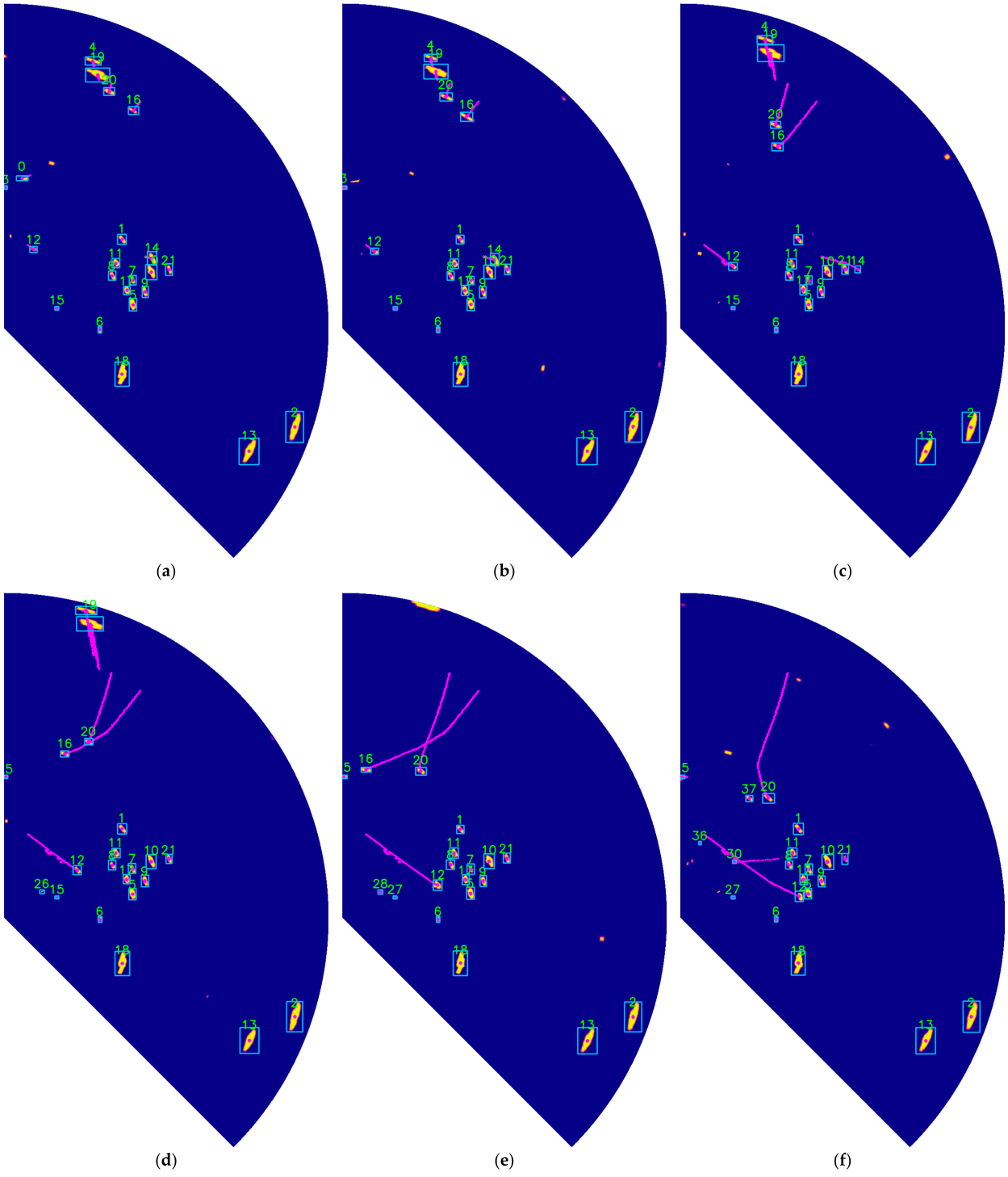
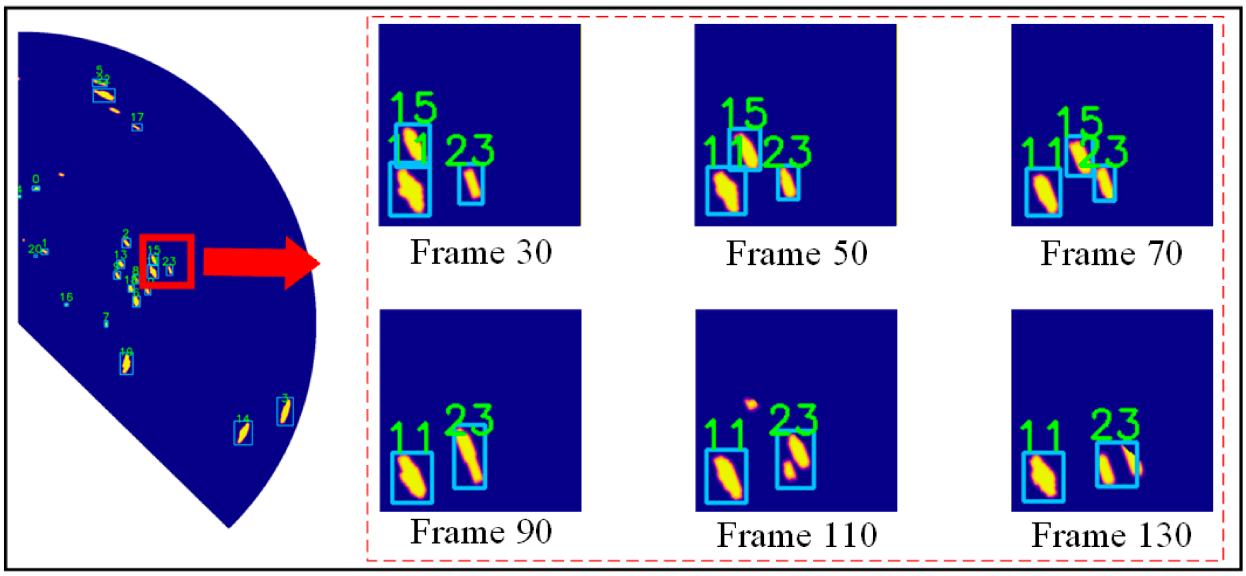
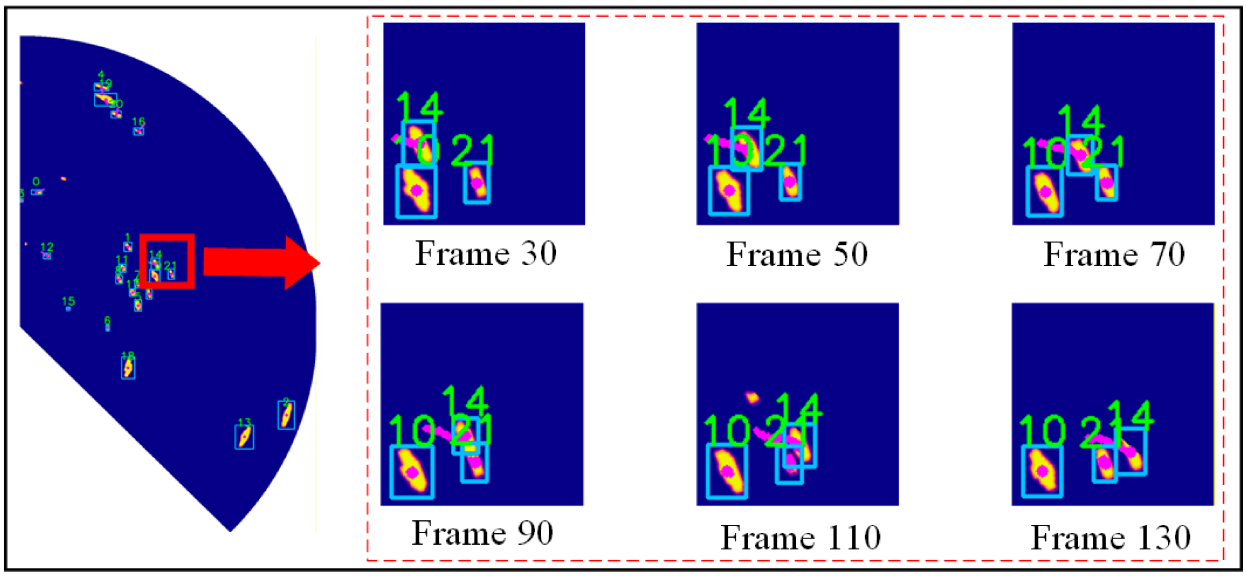


Figure 3. Visualization of tracking results. (**a**–**f**) Tracking results at frames 30, 50, 150, 250, 350, and 450 respectively.

To provide a closer look at the local tracking details, Figure 4 shows a cropped view of the challenging tracking scenario. Figure 4a shows the tracking without trajectory enhancement, and Figure 4b shows the tracking after trajectory enhancement. The same target should have the same ID value in different frames to indicate successful tracking. The target (ID 15 in Figure 4a and ID 14 in Figure 4b at frame 30) initially approached another target, then moved away. During target approach, the radar's limited resolution caused signal overlap in the radar video, making tracking difficult. As shown in Figure 4a, at frame 90, when target ID 15 overlapped with target ID 23, the model failed to continue tracking ID 15. As ID 15 moved away from ID 23, the model incorrectly merged the two targets and lost ID 15. In Figure 4b, although tracking target ID 14 was challenging during approach, overlap, and separation from target ID 21 based on visual features alone, the model with the TAE correctly tracked both targets and maintained their IDs through trajectory enhancement.



(**a**)



(**b**)

Figure 4. Visualization of a cropped and magnified local region from the tracking results. (**a**) shows Tracking situation without using the TAE module and without trajectory enhancement. (**b**) Tracking situation after adding the TAE module and performing trajectory enhancement.

4.5. Computational Complexity and Runtime Analysis

To evaluate the additional computational cost introduced by the proposed TAE module, we compare the computational complexity and runtime of the original TrackFormer and the proposed TrajE2E-MOT. The comparison includes the number of parameters, FLOPs, memory consumption, inference time, and FPS.

As shown in Table 7, compared with the original TrackFormer, the proposed TrajE2E-MOT increases the number of parameters from 43.409 M to 43.813 M and the FLOPs from 432.561 G to 441.394 G. The memory consumption increases from 2360 MB to 2420 MB. In terms of inference speed, the inference time increases from 168.77 ms/frame to 181.59 ms/frame, and the FPS decreases from 5.93 to 5.32. These results indicate that the trajectory encoder and frame-trajectory cross-modal attention module introduce only moderate computational overhead. Although the proposed method is slightly slower than the original TrackFormer, it still achieves an inference speed of 5.32 FPS, corresponding to 181.59 ms per frame. Considering that maritime radar video usually has a relatively low update rate compared with natural video, this inference speed can basically meet the online processing requirements of radar video tracking. Therefore, TrajE2E-MOT achieves improved tracking performance with an acceptable increase in computational complexity.

Table 7. Computational complexity and runtime results.

Method	Parameters (M)	FLOPs (G)	Memory Consumption (MB)	Inference Time (ms/Frame)	FPS
TrackFormer	43.409	432.561	2360	168.77	5.93
TrajE2E-MOT (Our)	43.813	441.394	2420	181.59	5.32

4.6. Limitation Analysis

Although the proposed TrajE2E-MOT can effectively improve identity continuity in long-term tracking, some limitations still exist in cluttered maritime radar scenarios. Figures 5 and 6 show a representative example involving target ID 12 under clutter interference. As shown in Figure 5, this target is continuously tracked throughout the selected frames, indicating that the proposed method can maintain its identity. However, in several frames, clutter responses appear near the target. These responses may be caused by sea clutter or wave-induced background interference. Since the additional radar echo is spatially close to the real target, the tracker tends to regard the target echo and the adjacent clutter response as a whole. As a result, the tracking bounding box becomes larger than the actual target region in these frames, and the target center is shifted from its true position. This effect can also be observed in Figure 6. The red circle highlights the localization drift of target ID 12. Although the target identity is maintained, the estimated position is temporarily affected by nearby clutter responses, resulting in a local deviation of the estimated trajectory.

This example indicates that the proposed TAE is effective for maintaining identity continuity, but it sometimes cannot fully suppress clutter-induced localization errors when clutter responses are tightly adjacent to the target echo. This limitation is related to the weak visual separability of maritime radar targets. The proposed TAE reconstructs and enhances visual feature encoding by using trajectory information, which is beneficial for reducing identity switches and maintaining long-term target continuity. However, it does not explicitly distinguish true target echoes from adjacent clutter responses and is not very effective in overcoming false alarms. Further improvements in the visual encoding network, optimized track initialization, and adaptive confidence-threshold adjustment are potential

strategies for reducing false positives and overcoming localization drift in maritime radar multi-object tracking.

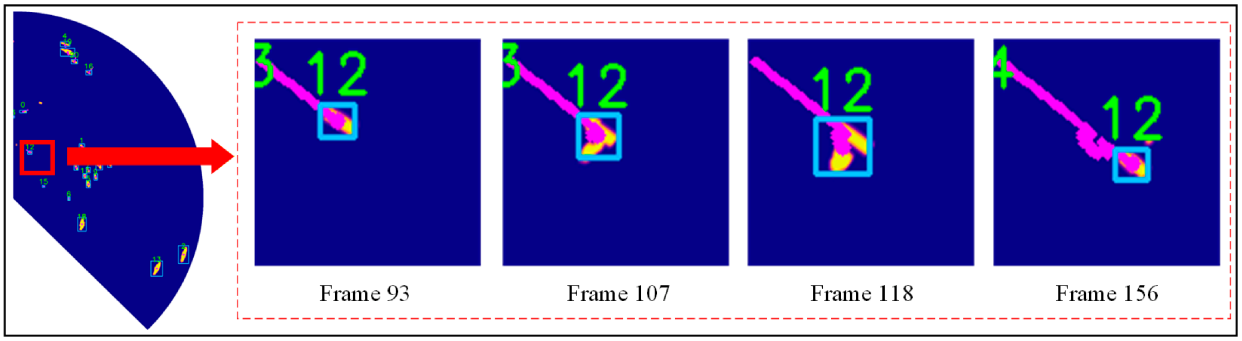


Figure 5. Multi-frame tracking results of target ID 12 under clutter interference.

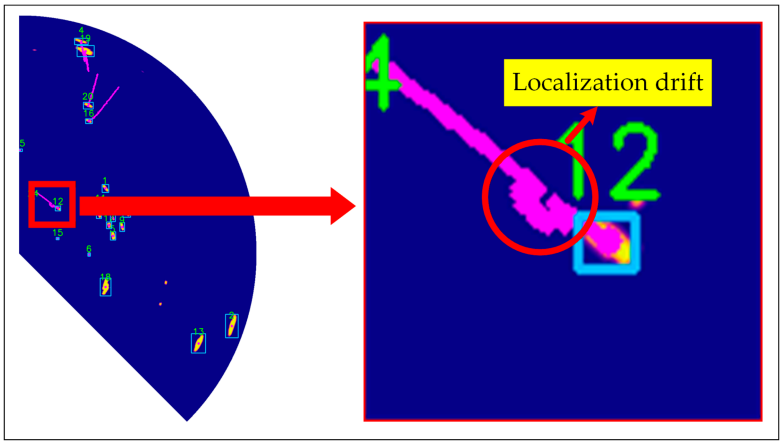


Figure 6. Localization drift of target ID 12 caused by clutter interference.

5. Conclusions

This paper presents a trajectory-aware end-to-end multi-object tracking method for radar video sequences. This method incorporates a novel TAE structure that utilizes the real-time trajectories of surviving targets to reconstruct the visual encoding. This method addresses visual feature limitations and significantly reduces ID switches during long-term radar tracking. It outperforms previous state-of-the-art methods on our collected radar dataset. Furthermore, the proposed method fills a research gap in end-to-end multi-target tracking for radar applications.

In the future, we will focus on solving the problems existing in the engineering implementation of this algorithm, promoting the intelligent upgrade of radar systems. We will also continue to collect radar video data and explore more cost-effective annotation methods to promote the progress of data-driven radar video multi-target tracking technology.

Author Contributions: Conceptualization, Z.K. and W.X.; methodology, Z.K. and W.X.; software, Y.C.; validation, Z.K. and Y.C.; formal analysis, Y.C.; data curation, Y.C.; writing—original draft preparation, Z.K.; writing—review and editing, W.X.; visualization, Y.C.; supervision, W.X. and Y.C.; project administration, W.X.; funding acquisition, W.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant numbers 62171453 and U2433216).

Data Availability Statement: The radar video dataset, source code, and trained model weights used in this study are not publicly available due to the latest institutional management regulations and

the potentially sensitive nature of maritime radar surveillance data. The dataset, source code, and trained model weights may be made available by the corresponding author upon reasonable request, subject to institutional review and approval.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

TrajE2E-MOT	Trajectory-aware end-to-end multi-object tracking model
TAE	Trajectory-aware encoder
GP	Gaussian processes
PMBM	Poisson Multi-Bernoulli Mixture
MOTA	Multiple Object Tracking Accuracy
IDs	ID switches
MOTP	Multiple-Object Tracking Precision
FN	False negative
FP	False positive
GT	Ground truth
MT	Mostly tracked
ML	Mostly lost
REID	Re-identification

References

1. Zhang, Y.; Wang, C.; Wang, X.; Zeng, W.; Liu, W. FairMOT: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vis.* **2021**, *129*, 3069–3087.
2. Meinhardt, T.; Kirillov, A.; Leal-Taixe, L.; Feichtenhofer, C. Trackformer: Multi-object tracking with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE/CVF: New York, NY, USA, 2022; pp. 8844–8854.
3. Zeng, F.; Dong, B.; Zhang, Y.; Wang, T.; Zhang, X.; Wei, Y. MOTR: End-to-End Multiple-Object Tracking with Transformer. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2022.
4. Leal-Taixé, L.; Milan, A.; Reid, I.; Roth, S.; Schindler, K. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. *arXiv* **2015**, arXiv:1504.01942.
5. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In *Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP)*; IEEE: New York, NY, USA, 2016; pp. 3464–3468.
6. Wojke, N.; Bewley, A.; Paulus, D. Simple online and realtime tracking with a deep association metric. In *Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP)*; IEEE: New York, NY, USA, 2017; pp. 3645–3649.
7. Zhang, Y.; Sun, P.; Jiang, Y.; Yu, D.; Weng, F.; Yuan, Z.; Luo, P.; Liu, W.; Wang, X. Bytetrack: Multi-object tracking by associating every detection box. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 1–21.
8. Qin, Z.; Wang, L.; Zhou, S.; Fu, P.; Hua, G.; Tang, W. Towards generalizable multi-object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE/CVF: New York, NY, USA, 2024; pp. 18995–19004.
9. Zhou, X.; Koltun, V.; Krähenbühl, P. Tracking objects as points. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 474–490.
10. Bergmann, P.; Meinhardt, T.; Leal-Taixe, L. Tracking without bells and whistles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*; IEEE/CVF: New York, NY, USA, 2019; pp. 941–951.
11. Shuai, B.; Berneshawi, A.; Li, X.; Modolo, D.; Tighe, J. Siammot: Siamese multi-object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*; IEEE/CVF: New York, NY, USA, 2021; pp. 12372–12382.
12. Cao, X.; Zheng, Y.; Yao, Y.; Qin, H.; Cao, X.; Guo, S. TOPIC: A parallel association paradigm for multi-object tracking under complex motions and diverse scenes. *IEEE Trans. Image Process.* **2025**, *34*, 743–758. [CrossRef]
13. Wang, Z.; Zheng, L.; Liu, Y.; Li, Y.; Wang, S. Towards real-time multi-object tracking. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 107–122.
14. Zhou, Y.; Su, H.; Tian, S.; Liu, X.; Suo, J. Multiple-kernelized-correlation-filter-based track-before-detect algorithm for tracking weak and extended target in marine radar systems. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 3411–3426.
15. Zhou, Y.; Wang, T.; Hu, R.; Su, H.; Liu, Y.; Liu, X.; Suo, J.; Snoussi, H. Multiple kernelized correlation filters (MKCF) for extended object tracking using X-band marine radar data. *IEEE Trans. Signal Process.* **2019**, *67*, 3676–3688. [CrossRef]

16. Han, J.; Hou, Y.; Tao, C.; Duan, Z. A two-stage detection method based on improved DP-TBD for marine weak extended targets. *Digit. Signal Process.* **2025**, *160*, 105054.
17. Kim, H.; Kim, D.; Lee, S.-M. Marine object segmentation and tracking by learning marine radar images for autonomous surface vehicles. *IEEE Sens. J.* **2023**, *23*, 10062–10070. [CrossRef]
18. Zhang, J.; Sun, J.; Zhang, Y.; Zhu, X. Multi-target Tracker Using Combined Motion and Appearance Model for Radar Video Data. In *Proceedings of the 2022 IEEE 17th Conference on Industrial Electronics and Applications (ICIEA)*; IEEE: New York, NY, USA, 2022; pp. 1–6.
19. Sun, X.; Xia, L.; Tian, C.; Zheng, H.; Cao, W.; Zong, C. Association Tracking of Vessels Using X-Band Radar Data in Complex Waterborne Environment. In *Proceedings of the 2023 7th International Conference on Transportation Information and Safety (ICTIS)*; IEEE: New York, NY, USA, 2023; pp. 469–474.
20. Zhang, J.; Zuo, Z.; Bei, S.; Wu, P.; Huang, H. Ukf-ec: Combining the unscented Kalman filter and the maximum weight algorithms for moving target tracking in marine radar images. In *Proceedings of the International Conference on Autonomous Unmanned Systems*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 1830–1844.
21. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI: Washington, DC, USA, 2020; pp. 12993–13000.
22. Xu, X.; Wu, B.; Xie, L.; Teixeira, Â.P.; Yan, X. A novel ship speed and heading estimation approach using radar sequential images. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 11107–11120. [CrossRef]
23. Xu, X.; Wu, B.; Teixeira, Â.P.; Yan, X.; Soares, C.G. Integration of radar sequential images and AIS for ship speed and heading estimation under uncertainty. *IEEE Trans. Intell. Transp. Syst.* **2023**, *25*, 5688–5702. [CrossRef]
24. Yang, S.; Thormann, K.; Baum, M. Linear-time joint probabilistic data association for multiple extended object tracking. In *Proceedings of the 2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*; IEEE: New York, NY, USA, 2018; pp. 6–10.
25. Baerveldt, M.; López, M.E.; Brekke, E.F. Extended target PMBM tracker with a Gaussian Process target model on LiDAR data. In *Proceedings of the 2023 26th International Conference on Information Fusion (FUSION)*; IEEE: New York, NY, USA, 2023; pp. 1–8.
26. Parmar, N.; Vaswani, A.; Uszkoreit, J.; Kaiser, L.; Shazeer, N.; Ku, A.; Tran, D. Image transformer. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge, MA, USA, 2018; pp. 4055–4064.
27. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762v7. [CrossRef]
28. Bernardin, K.; Stiefelhagen, R. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP J. Image Video Process.* **2008**, *2008*, 246309. [CrossRef]
29. Shim, K.; Ko, K.; Yang, Y.; Kim, C. Focusing on Tracks for Online Multi-Object Tracking. In *Proceedings of the Computer Vision and Pattern Recognition Conference*; Springer: Berlin/Heidelberg, Germany, 2025; pp. 11687–11696.
30. Sun, P.; Cao, J.; Jiang, Y.; Zhang, R.; Xie, E.; Yuan, Z.; Wang, C.; Luo, P. Transtrack: Multiple object tracking with transformer. *arXiv* **2020**, arXiv:2012.15460. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.