Spatial-Temporal Attention Network for Track-Track Association with Biased Data

Haowei Jia
School of
Computer Science & Engineering,
Nanjing University of Science & Technology
Nanjing, China

Gan Wang
School of
Integrated Circuits & Electronics,
Beijing Institute of Technology
Beijing, China

Huajun Liu[†]
School of
Computer Science & Engineering,
Nanjing University of Science & Technology
Nanjing, China

***Abstract*—Track-track association (TTA) in complex environment for multi-sensor fusion is a challenging topic due to the uncertainty of measurements, biased data, mismatch caused by different resolution etc. In this work, we proposed an end-to-end deep learning model, named the *spatial-temporal attention network (STAN)* for TTA tasks in complex scenarios. Three modules in the backbone of STAN for intra-track and inter-track feature representation are based on self-attention mechanism, e.g., the motion mode encoder (MME) module to encode the motion pattern of single moving targets, the spatial structure extraction (SSE) module for capturing the inter-track spatial interaction relation of an individual sensor, and the spatial-temporal fusion (STF) module for intra-track modeling on temporal dimensions, respectively. A relation reasoning head (RRH) is built for track-track relation reasoning based on the encoded track features. Experimental results on different tasks show that our proposed method achieved superior performance for track-track association compared with previous methods.**

***Index Terms*—Track-track association, complex environment, spatial-temporal attention network, self-attention mechanism**

I. Introduction

In complex environment, it is necessary to integrate multiple sensors to achieve more accurate detection and recognition due to the possible false alarm, missed detection and limited measurement accuracy of a single sensor. Nevertheless, distributed sensor systems can make up the shortcomings of a single sensor by fusing measurements from different sensors to obtain a more consistent situation map. Track-track association (TTA) is a necessary and essential step for distributed sensor fusion, which is a challenging topic due to the uncertainty of measurements, biased data, mismatch caused by different resolution etc.

Earlier work on TTA can be categorized into two types, e.g., traditional methods and learning-based methods. In the case of simple scenes, where sensors are accurate enough, without bias and without missed detection, researchers depicted this topic as the assignment problem, and some classical algorithms [1]–[3] are proposed for this task in the last few decades. And some methods for complex scenes [4]–[8] have been proposed to obtain a more consistent result and to eliminate ghost tracks as accurate as possible. Meanwhile, a series of classical methods [9]–[12] have been proposed for the joint bias estimation and track association. And some methods reference topological patterns modeling [13]–[15] or fuzzy clustering [16]–[18] are proposed to deal with the systematic bias problem. By replacing the absolute coordinates with a reference topology to compute the correlations, the reference topological pattern method [13] increases the robustness of the model to systematic bias and has good performance. By constructing appropriate membership functions, the fuzzy clustering method [17] realizes the computation of association probabilities between targets and the construction of probability matrix, and achieved reasonable results in the TTA task.

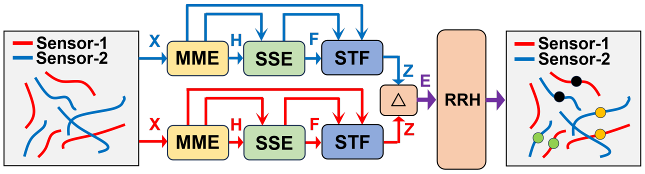

Fig. 1. The overall architecture of the proposed STAN.

Above most methods are built on the association cost matrix between individual tracks, and global nearest neighbour algorithms aim to search the optimal association relation by analytical methods and algorithms, which are sensitive to the co-existence of measurement noise and systematic bias.

Recently, machine learning and deep learning methods show great potential on TTA tasks [19], [20]. For instance, in [21], the TTA task is modeled as a classification problem, which can achieve satisfactory results in several scenarios. And in [22], [23], the TTA is regarded as a maximum likelihood estimation problem and researchers proposed a global nearest pattern method based on K-best searching algorithms. And in [24], an AE-3D-CNN deep learning model is proposed to correlate the tracks of the automatic identification system and radar by learning both track features and scene features. However, those machine learning methods are based on distance metric or estimation covariance, which are sensitive to systematic bias and would degrade greatly as the scenarios become more complex.

In this work, the TTA task is modeled as an end-to-end deep learning problem to predict the association probability matrix, and the spatial-temporal attention network (STAN) (seen in

[†] Corresponding author (email: liuhj@njust.edu.cn).

the Fig. 1) are proposed for TTA tasks. Specifically, the main contributions of this work can be summarized as follows:

- Modeling the complex TTA issues with an end-to-end deep representation learning framework to predict the association probability matrix from individual encoded track features.
- Proposing a spatial-temporal self-attention network to deal with TTA tasks, including several modules on track encoding, spatial structure extraction, spatial-temporal fusion, and association probability reasoning.
- Experimental results show the proposed STAN achieves better performance on several tasks, and it is more flexible for complex scenarios with different track length and missed detections.

II. Problem Formation

We assume that two track sets collected by two individual sensors are $Set_1 = \{A_0, A_1, ..., A_M\}$ and $Set_2 = \{B_0, B_1, ..., B_N\}$, respectively, where M or N represent the number of tracks. Each track of sensors is composed of τ track points and each track point consists of four-dimensional information in the Cartesian coordinates, e.g., the position along X-axis, position along Y-axis, speed, and heading of the target. Thus each track $A_i \in Set_1$ or $B_j \in Set_2$ can be denoted as $[r_1, r_2, ..., r_\tau] \in \mathbb{R}^{\tau \times 4}$ respectively. And all the tracks in Set_1 and Set_2 could be denoted as $A \in \mathbb{R}^{M \times \tau \times 4}$ and $B \in \mathbb{R}^{N \times \tau \times 4}$, respectively, and S is the predicted association probability matrix. Then the TTA issue can be modelled as a deep representation learning framework as follows:

$$\begin{cases} A_{rep} = \phi(A), \\ B_{rep} = \phi(B), \\ S = R(A_{rep}, B_{rep}), \end{cases} \tag{1}$$

where $\phi(\cdot)$ is a sub-network in the backbone aiming to transform the raw tracks into high-dimensional encoded track features, and $R(\cdot)$ represents the relational reasoning sub-network on the two tracks aiming to reason the association results between tracks, and $S \in \mathbb{R}^{M \times N}$ is an association probability matrix to be predicted, and its element $S_{ij} \in (0, 1)$ denotes the association probability between the ith track in Set_1 and the jth track in Set_2.

Finally, the association matrix can be computed by a post-processing step as follows:

$$C = \psi(S), \tag{2}$$

where $C \in \mathbb{R}^{M \times N}$, every element of which is 0 or 1. And the function ψ is a post-processing function, which is defined as follows in detail:

(1) set a threshold β to set all elements of the association probability matrix less than β to 0;

(2) select the largest element of the matrix, set the value of the element to 1, and set all other elements in the row and column of the element to 0;

(3) repeat step (2) until all the elements are 0 or 1.

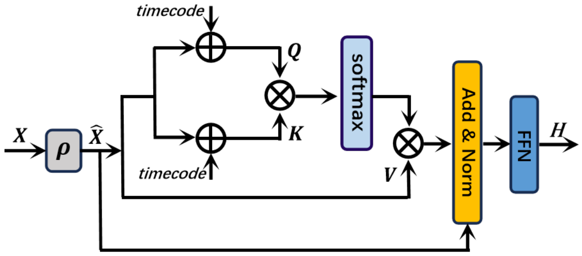

Fig. 2. The architecture of MME.

III. Our Method

A. Overall architecture

As shown in Fig. 1, a deep representation learning framework based on the popular backbone-head pipeline is proposed for TTA tasks in complex environment, where the backbone is designed for spatial-temporal feature representation over track clusters, and the head is for track association relation reasoning. Specifically, the backbone is composed of three attention based modules, e.g., a motion mode encoder (MME), a spatial structure extraction (SSE), and a spatial-temporal fusion (STF). The MME module aims to encode the low-level motion states of all targets through attentional manipulation, thus extracting the temporal relation from the original sequence. The SSE module is designed to capture interactions between different targets at different time frames. The STF module aims to capture the temporal dependence in the sequence of interactive representations being output by SSE, i.e., in a sense, fusing temporal and spatial features acquired from the above modules.

Special attention should be paid to the fact that input tracks from different sensors do not interact during the feature extraction phase. The three modules, in a sequential manner, incrementally complete the feature fusion of all tracks collected by a single sensor.

Finally, the information interactions among tracks from different sensors are realized in the RRH module, with the aim of inferring the association probability between tracks from different sensors.

B. Motion mode encoder (MME) module

The motion of targets does not always follow a fixed pattern and is usually maneuverable, which means the tracks are diverse. The MME is designed to encode the movement tracks of single targets. The architecture of MME is shown in Fig. 2. $X \in \mathbb{R}^{M \times \tau \times 4}$ denotes the original tracks. First, the input motion state at each time point within the track is transformed into an embedding representation by a fully connected layer.

$$\hat{X} = \rho\left(X\right), \tag{3}$$

where ρ represents a linear layer with a LeakyRelu activation function, $\hat{X} \in \mathbb{R}^{M \times \tau \times d_M}$ is the set of coded feature vectors

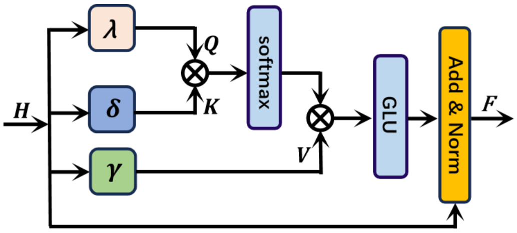

Fig. 3. The architecture of SSE.

in τ time steps for all tracks, where d_M is the dimension of the feature. Then we use $\hat{X}$ to obtain three different vectors

$$\begin{cases} Q &= \hat{X} \oplus timecode, \\ K &= \hat{X} \oplus timecode, \\ V &= \hat{X}, \end{cases} \tag{4}$$

where tensors $Q, K, V \in \mathbb{R}^{M \times \tau \times d_M}$ are referred to as queries, keys and values, respectively. Here, $\oplus$ denotes the operation of broadcasting $timecode \in \mathbb{R}^{\tau \times d_M}$ to the same dimension as $\hat{X}$ and summing it up. It's worth mentioning that the $timecode$ is the temporal encoding we construct, which is more appropriate for our task. Specifically, we initialize a lookup table containing learnable embeddings $L \in \mathbb{R}^{\mu \times d_M} (\mu > \tau)$. For τ points in a track, we extract the time steps $(1, ..., \tau)$ as the index to obtain the first to the τth vector in L to compose the $timecode$:

$$timecode = \mathrm{L}\left(1, ..., \tau\right). \tag{5}$$

Then, the scaled dot-product attention operation following the original Transformer [25] is formulated as follows:

$$Y = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_M}}\right)V, \tag{6}$$

where $Y \in \mathbb{R}^{M \times \tau \times d_M}$ and the operator T transposes the matrix K in the last two dimensions. $H \in \mathbb{R}^{M \times \tau \times d_M}$ is obtained by further processing of the encodings as follows:

$$H = \mathrm{FFN}\left(\mathrm{LayerNorm}\left(\hat{X} + Y\right)\right), \tag{7}$$

where LayerNorm [26] is the normalization function and FFN stands for the fully connected feedforward neural network which contains two linear layers and a LeakyRelu activation function layer between them. This structure, similar to the encoder layer in Transformer [25], has been shown to be suitable for track feature encoding.

C. Spatial structure extraction (SSE) module

In general, the spatial topology formed by the same target and the surrounding targets is similar with measurements in different sensors. Therefore, capturing the spatial relationship between the target and other targets within the field of view is an essential part of the process of extracting target features, and this spatial structure feature will be helpful to distinguish different targets. In the above MME module, the targets are considered to be independent of each other and their relationship with other targets is ignored. Therefore it is vital to capture information about the interaction of each target with other targets, which contributes to the SSE module.

The architecture of SSE is shown in Fig. 3. Specifically, the first two dimensions of the matrix H are transposed to obtain $\hat{H} = [H_1, H_2, ..., H_\tau] \in \mathbb{R}^{\tau \times M \times d_M}$, which can be viewed as being divided into multiple time frame representations. Then, the attention layer on $\hat{H}$ in the spatial dimension is performed to capture the inter-track spatial interaction features for each frame within a single sensor.

$$\begin{cases} Q_s &= \lambda(\hat{H}, W_s^q), \\ K_s &= \delta(\hat{H}, W_s^k), \\ V_s &= \gamma(\hat{H}, W_s^v), \end{cases} \tag{8}$$

$$H_s = \mathrm{softmax}\left(\frac{Q_s K_s^T}{\sqrt{d_M}}\right)V_s, \tag{9}$$

$$F = \mathrm{LayerNorm}(\mathrm{GLU}(H_s) + H), \tag{10}$$

where λ, δ and γ are three learnable linear transformations, and W_s^q, W_s^k, $W_s^v \in \mathbb{R}^{d_M \times d_M}$ are parameters of these independent linear layers which operate on the last dimension of the tensor. Subsequently, those track features are regularized by the Gated Linear Units (GLUs) [27] and LayerNorm [26]. The residual structure facilitates effective information flow between different layers, and the GLU further enhances the semantic information.

D. Spatial-temporal fusion (STF) module

The SSE module is used for spatial structure representations within individual timestamps. However, for different temporal frames, especially neighboring frames, their spatial structures are temporally correlated. Therefore, in the STF module, the connection of the spatial structures of sequential frames is captured by the attention mechanism again.

In other words, the STF module aims to fuse temporal and spatial interaction features of input tracks by enabling cross-frame interaction of spatial structural features output by the previous module on different frames. The processing flow of STF is similar to that of the SSE module. Specifically, the first two dimensions of the matrix F are transposed to obtain $\hat{F} = [F_1, F_2, ..., F_M] \in \mathbb{R}^{M \times \tau \times d_M}$ and the attention layer operates on the temporal dimension as:

$$\begin{cases} Q_f &= \lambda(\hat{F}, W_f^q) \oplus timecode, \\ K_f &= \delta(\hat{F}, W_f^k) \oplus timecode, \\ V_f &= \gamma(\hat{F}, W_f^v), \end{cases} \tag{11}$$

$$F_f = \mathrm{softmax}\left(\frac{Q_f K_f^T}{\sqrt{d_M}}\right)V_f, \tag{12}$$

where $W_f^q, W_f^k, W_f^v \in \mathbb{R}^{d_M \times d_M}$ are the parameters of three independent linear layers which operate on the last dimension of the tensor, and temporal embedding is added again to give the model the ability to re-arrange temporal tokens. Furthermore, the GLU is used to enhance the semantic information,

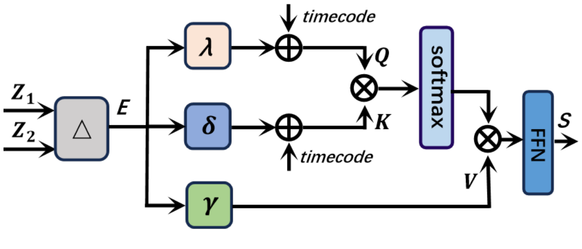

Fig. 4. The architecture of RRH.

and the residual structure facilitates to fuse the information across modules above mentioned as follows

$$Z = \mathrm{LayerNorm}(\mathrm{GLU}(F_f) + F + H). \tag{13}$$

E. Relation reasoning head (RRH) module

Based on the temporal and spatial interaction features obtained from the STF module, the association probability reasoning between tracks is conducted in the RRH module. The architecture of RRH is depicted in Fig. 4. Specifically, all the encoded features of the two individual sensors output by parallel STF modules are denoted as $Z_1 \in \mathbb{R}^{M\times\tau\times d_M}$ and $Z_2 \in \mathbb{R}^{N\times\tau\times d_M}$, respectively. Then each of the M encodings of Z_1 is repeated to N copies and spliced with each of the encodings of Z_2, resulting in $E \in \mathbb{R}^{(M\times N)\times 2\tau\times d_M}$.

$$E = Z_1 \triangle Z_2, \tag{14}$$

where $\triangle$ represents the repeat and splice operation as described above. Then, for each vector in E, each element in the vector can acquire full information of two track features in an intra-track and inter-track way through the attention operation.

$$\begin{cases} Q_r &= \lambda(E, W_r^q) \oplus timecode, \\ K_r &= \delta(E, W_r^k) \oplus timecode, \\ V_r &= \gamma(E, W_r^v), \end{cases} \tag{15}$$

$$\hat{E} = \mathrm{softmax}\left(\frac{Q_r K_r^T}{\sqrt{d_M}}\right) V_r, \tag{16}$$

where the tensors $Q_r, K_r, V_r \in \mathbb{R}^{(M\times N)\times 2\tau\times d_M}$, the $W_r^q, W_r^k, W_r^v \in \mathbb{R}^{d_M\times d_M}$ are the parameters of three independent linear layers which operate on the last dimension of the tensor. Then, the last two dimensions of $\hat{E} \in \mathbb{R}^{(M\times N)\times 2\tau\times d_M}$ are merged, and the association probability matrix $S \in \mathbb{R}^{M\times N}$ is predicted by a FFN module as:

$$S = \mathrm{FFN}(\hat{E}), \tag{17}$$

where the FFN contains a linear layer with a parameter matrix $W_r \in \mathbb{R}^{(2\tau\times d_M)\times 1}$ aiming to perform further reasoning. Subsequently, a Sigmoid activation function aims to constrain the probability to be between 0 and 1.

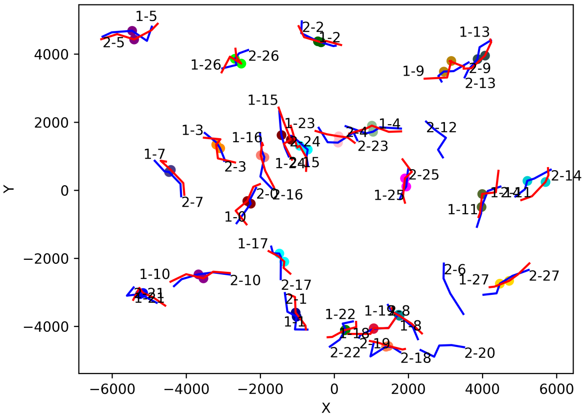

Fig. 5. Visualized results (Best viewed in color). The tracks in sensor 1 or 2 are colored red or blue and the track number is labeled at the beginning or end of the track respectively. Tracks judged to be associated are marked at the midpoint by two circles with the same color.

F. Loss function

In essence, the TTA task in this work is modeled as a classification problem in our end-to-end deep learning framework. The STAN is then trained using the cross-entropy loss function, which is calculated based on the ground-truth association matrix G and the predicted association probability matrix S by STAN. The loss function is define as follows:

$$L = -\frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{N} G_{ij} log(S_{ij}). \tag{18}$$

IV. Experimental Results and Analysis

In this section, the dataset is first introduced and the different task settings are described. Subsequently, the evaluation metrics for TTA tasks and implementation details of our method are presented. Finally, the experimental results and analysis are given and the rationality of the overall structure of the model as well as the validity of some of the details are verified by a series of ablation experiments.

A. The TTA dataset and task configuration

Training and evaluation for TTA tasks on synthetic data is an accustomed method in previous work, and up to now, there aren't widely accepted real data benchmarks to evaluate deep learning methods for general TTA tasks. Referring to the simulation method of [24], we assume that the number of targets within the observation range is uniformly distributed over the specified range. Targets are accelerated with Gaussian white noise with zero mean value, and the standard deviation of the acceleration in both X- and Y-directions is $2m/s^2$. The initial position, velocity, and heading of targets follow a uniform distribution, ranging from -5 km to 5 km, 50 m/s to 100 m/s, and 0° to 360°, respectively. There is Gaussian white noise with zero mean value on the position measurement in the X- and Y-directions of the both sensors and the measurement period is uniformly set to $4s$. A systematic bias is also added to the position measurement in the X- and Y- directions of

TABLE I
THE PARAMETER SETTINGS OF DIFFERENT TASKS.

Tasks	1	2	3	4	5	6	7	8
N	16∼32	32∼64	16∼32	32∼64	32∼64	32∼64	32∼64	32∼64
L	5	5	5	5	15	25	5	5
T_p	0.9	0.7	0.9	0.7	0.7	0.7	0.7	0.7
P_p	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.7
$S_e(m)$	(0,100)	(0,100)	(200,400)	(200,400)	(200,400)	(200,400)	(200,400)	(200,400)
$SD_1(m/s^2)$	[100, 100]	[100, 100]	[200, 200]	[200, 200]	[200, 200]	[200, 200]	[200, 200]	[200, 200]
$SD_2(m/s^2)$	[120, 120]	[120, 120]	[250, 250]	[250, 250]	[250, 250]	[250, 250]	[250, 250]	[250, 250]

Note: N, L, T_p, P_p and S_e represent, respectively, the range of the number of targets, the length of the track, the target detection probability of the both sensors, the detection probability of the points in a track of the both sensors, the range of the systematic error in the second sensor (same for X-direction position and Y-direction position). SD_1 denotes, in the first sensor, the standard deviation of the measurement noise added in the x and y directions, respectively. And SD_2 is for the second sensor.

TABLE II
THE PERFORMANCE OF DIFFERENT METHODS ON TASK 1-4.

Tasks	1			2			3			4		
Metrics(%)	Rec	N_Acc	Pre	Rec	N_Acc	Pre	Rec	N_Acc	Pre	Rec	N_Acc	Pre
FC [17]	97.72	97.76	97.49	95.30	93.01	87.06	89.06	90.46	90.33	82.10	80.87	70.90
REP [13]	96.96	93.95	90.69	84.40	75.61	62.48	91.70	89.09	85.84	73.80	67.80	54.66
AE-3D-CNN [24]	96.74	97.32	97.49	95.32	95.38	92.16	90.10	91.64	91.81	86.67	89.65	85.45
STAN (Ours)	**99.06**	**99.08**	**98.97**	**98.39**	**98.04**	**96.39**	**95.00**	**95.46**	**95.23**	**89.91**	**92.08**	**88.73**

TABLE III
EXPERIMENTAL ANALYSIS ON DIFFERENT TRACK LENGTHS.

Tasks	4	5	6
$Rec(\%)$	89.91	93.67	98.45
$N\_Acc(\%)$	92.08	94.98	98.72
$Pre(\%)$	88.73	92.78	98.08

TABLE IV
EXPERIMENTAL ANALYSIS ON DIFFERENT MISSED DETECTIONS.

Tasks	4	7	8
$Rec(\%)$	89.91	87.46	82.38
$N\_Acc(\%)$	92.08	90.23	86.42
$Pre(\%)$	88.73	86.19	80.98

the second sensor. For task 1-4, we use task 1 as a baseline task (similar to the task set up in [24] which proves to be a reasonable setting). Considering the reality of some bad situations, we increase the difficulty of the task. In task 2, we increase the target density and decrease the target detection probability. In task 3, we increase the standard deviation of the measurement noise and the systematic error of sensor 2. In task 4, we increase the target density, systematic bias, random error, meanwhile decrease the target detection probability. Furthermore, task 4 is used as a baseline to set up two comparison experiments. And task 5-6 is set to verify the model's ability on handling different track lengths, task 7-8 is set to verify the model's ability to handle the case of missing points. More specific parameter configurations are listed in Tab. I. For above each task, 60,000 Monte Carlo experiments corresponding to 60,000 scenarios were randomly generated online for training and testing with the ratio of 5:1.

B. Performance metrics

The algorithms are evaluated by computing the metrics for each scenario and calculating the mean. For each scene, the task is treated as a binary classification problem. Assuming that the scene has M tracks for sensor 1 and N tracks for sensor 2, the number of samples is $M \times N$, which represents the number of track pairs obtained by combining each track of sensor 1 with each track of sensor 2. And the associated pairs are treated as positive samples and vice versa for negative samples. Then Recall, Accuracy, Precision could be used as evaluation metrics as below.

$$\begin{cases} Rec & = & \frac{TP}{TP+FN} \\ Acc & = & \frac{TP+TN}{TP+TN+FP+FN} \\ Pre & = & \frac{TP}{TP+FP} \end{cases} \tag{19}$$

Consider that even in the worst case, the prediction matrix differs from the ground truth by only $2\times min(M,N)$ elements, which results in the Acc metric approximating 1 in any case. Therefore, in our work, the normalized accuracy (N_Acc) is used for evaluation:

$$Acc_{min} = \frac{M \times N - 2 \times min(M,N)}{M \times N} \tag{20}$$

$$N\_Acc = \frac{Acc - Acc_{min}}{1 - Acc_{min}}. \tag{21}$$

C. Implementation details

In the MME module, the linear layer in the ρ function is configured with 64 neurons, matching the size of d_M. Additionally, the two linear layers in the FFN are set to 192 and 64 neurons, respectively. In instances of missing data points, the sequence is padded to the track's maximum length, and the corresponding positions are masked during the attention operation. Subsequently, the values at these positions are also adjusted to 0.

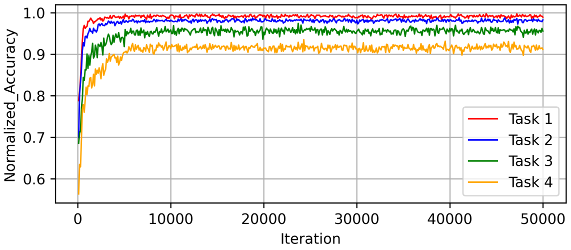

Fig. 6. The accuracy analysis for task 1-4.

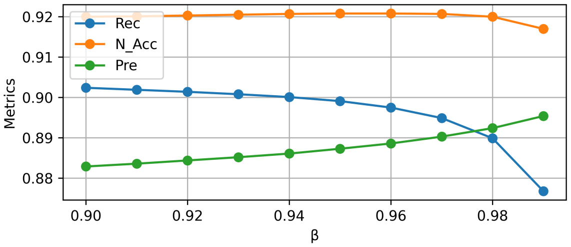

Fig. 7. The analysis on the parameter β.

D. Results analysis

Visualized results for task 1 are shown in Fig. 5, where exact associated tracks are marked with color circles. And the N_Acc curves during training the STAN model for task 1-4 are shown in Fig. 6, it can be seen that the model will converge within about 8,000 iterations. On task 1-4, we conduct comparison analysis with three methods, among them, two traditional methods are fuzzy clustering (FC) [17] and reference topology (REP) [13], and the third one is the AE-3D-CNN [24] which have shown strong capability in temporal and spatial simultaneous representation before this.

And we use the same training set and test set to evaluate the AE-3D-CNN model and STAN for task 1-4. The performance of four methods on task 1-4 are listed in Tab. II. It can be concluded that STAN outperformed the FC method by 1.32% on metric N_Acc in task 1, and 1.34% on Rec, and 1.48% on Pre, respectively. In more challenging environments, STAN outperforms the state-of-the-art method (e.g., AE-3D-CNN) by $2.66\%, 3.82\%$, and 2.43% on N_Acc in task 2-4, respectively. Moreover, it can be seen that STAN is adaptable to different track lengths and cases of missing points well from the results in Tab. III and Tab. IV respectively, where the task 4 is regarded as the baseline, and in task 5-6, we can get more track points from 10 to 20 than in task 4, and in task 7-8, the detection rate decreases from 1.0 to 0.9 and 0.7 respectively. Besides, an ablation study of the parameter β in Sec.2 is conducted on task 4, and it can be seen from Fig. 7 that as β increases, Rec will decrease gradually, and Pre will increase a bit, finally N_Acc will touch the peak when β is 0.96, which is a suitable parameter for these TTA tasks.

E. Ablation studies

In this section, a series of ablation studies on task 1 are conducted to explore the effects of model components and the details inside the model. This includes examining the necessity of specific characterization modules, the order in which these modules are arranged, and the impact of certain details (such as specific coding and residual structure) on performance. The results are presented in Tab. V.

TABLE V
ABLATION RESULTS

Items	1	2	3	4	5
$N\_Acc(\%)$	98.43	98.29	98.39	98.00	80.06

In the first set of experiments (Item 1, Item 2 and Item 3), when the temporal embeddings in MME, STF and RRH are removed individually, the association accuracy will decrease by 0.65%, 0.79% and 0.69% respectively.

In the second set of experiments (Item 4 and Item 5), if we remove the residual connections in SSE and STF individually, the association accuracy will decrease by 1.08% and 19.02%, respectively, which shows it is necessary to use skip-connection across modules.

Additionally, we conduct experiments, where the MME module, the SSE module, and the STF module are removed individually, and reordering above modules as well. In all cases, the association accuracy will degrade to less than 70%.

The ablation experiments demonstrate the necessity of the model components and validate the importance of certain details (e.g., residual connections) in our model. These results confirm the reasonableness and necessity of the overall architecture of the model.

V. CONCLUSION

In this work, we proposed a spatial and temporal attention network based on an end-to-end deep learning framework for track feature representation and reasoning about track-track relationships. Multiple evaluation experiments show that our method performed better than previous methods in complex environments, benefiting from the advantages of attention modules on capturing long range information. In addition, we demonstrate the flexibility of STAN in handling more complex scenarios, including tracks of different lengths and missed detections. Finally, the validity and rationality of the structure and some details of STAN are further demonstrated through a series of ablation experiments.

REFERENCES

[1] Allen J. Kanyuck and Robert A. Singer, "Correlation of multiple-site track data," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-6, no. 2, pp. 180–187, 1970.

[2] Michitaka Kosaka, Shoji Miyamoto, and Hirokazu Ihara, "A track correlation algorithm for multi-sensor integration," *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 2, pp. 166–171, 1987.

[3] Christopher L. Bowman, "Maximum likelihood track correlation for multisensor integration," in *18th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, 1979, vol. 2, pp. 374–376.

[4] Wei Tian, Gaoming Huang, Huafu Peng, Xuebao Wang, and Xiaohong Lin, "Sensor bias estimation based on ridge least trimmed squares," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 2, pp. 1645–1651, 2019.

[5] Wei Tian, Yue Wang, Xiuming Shan, and Jian Yang, "Analytic performance prediction of track-to-track association with biased data in multi-sensor multi-target tracking scenarios," *Sensors*, vol. 13, no. 9, pp. 12244–12265, 2013.

[6] Dimitri J Papageorgiou and John-David Sergi, "Simultaneous track-to-track association and bias removal using multistart local search," in *2008 IEEE Aerospace Conference*, 2008, pp. 1–14.

[7] Jun Wang, Yajun Zeng, Shaoming Wei, Zixiang Wei, Qinchen Wu, and Yvon Savaria, "Multi-sensor track-to-track association and spatial registration algorithm under incomplete measurements," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3337–3350, 2021.

[8] Aybars Tokta and Ali Koksal Hocaoglu, "Sensor bias estimation for track-to-track association," *IEEE Signal Processing Letters*, vol. 26, no. 10, pp. 1426–1430, 2019.

[9] Zhenhua Li, Siyue Chen, Henry Leung, and Eloi Bosse, "Joint data association, registration, and fusion using em-kf," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 2, pp. 496–507, 2010.

[10] Hongyan Zhu, Chen Wang, Wen Jiang, Chongzhao Han, and Yan Lin, "Integrated data association and bias estimation in the presence of missed detections," in *17th International Conference on Information Fusion*, 2014, pp. 1–8.

[11] Hongyan Zhu and Chen Wang, "Joint track-to-track association and sensor registration at the track level," *Digital Signal Processing*, vol. 41, pp. 48–59, 2015.

[12] Hao Zhu, Henry Leung, and Ka-Veng Yuen, "A joint data association, registration, and fusion approach for distributed tracking," *Information Sciences*, vol. 324, pp. 186–196, 2015.

[13] Wei Tian, "Reference pattern-based track-to-track association with biased data," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 1, pp. 501–512, 2016.

[14] Wei Tian, Gaoming Huang, Xiongjie Du, Yue Wang, and Huafu Peng, "Track to track association based on iterative reference pattern," in *2017 International Conference on Computer Technology, Electronics and Communication*, 2017, pp. 289–294.

[15] Hanbao Wu, Lun Li, and Kefei Zhang, "Track association method based on target mutual-support of topology," in *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference*, 2020, vol. 9, pp. 2078–2082.

[16] Ashraf M Aziz, "A new fuzzy clustering approach for data association and track fusion in multisensor-multitarget environment," in *2011 Aerospace Conference*, 2011, pp. 1–10.

[17] Shuaixin Hong, Dongliang Peng, and Yifang Shi, "Track-to-track association using fuzzy membership function and clustering for distributed information fusion," in *2018 37th Chinese Control Conference*, 2018, pp. 4028–4032.

[18] Mousa Nazari, Saeid Pashazadeh, and Leyli Mohammad-Khanli, "An adaptive density-based fuzzy clustering track association for distributed tracking system," *IEEE Access*, vol. 7, pp. 135972–135981, 2019.

[19] Xu Yasheng, Ding Chibiao, Ren Wenjuan, and XU Guangluan, "Multi-feature combination track-to-track association based on histogram statistics feature," *Journal of Radars*, vol. 8, no. 1, pp. 25–35, 2019.

[20] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[21] Yaqi Cui, Yu Liu, Tiantian Tang, and Hongfeng Zhu, "A new adaptive track correlation method for multiple scenarios," *IET Radar, Sonar & Navigation*, vol. 15, no. 9, pp. 1112–1124, 2021.

[22] Mark Levedahl and John D. Glass, "Optimal non-assignment costs for the gnp problem," in *2020 IEEE Aerospace Conference*, 2020.

[23] Mark Levedahl and John D. Glass, "Analysis of costs for the gnp problem," *Journal of Advances in Information Fusion*, vol. 16, pp. 17–30, 2021.

[24] Biao Jin, Yufeng Tang, Zhenkai Zhang, Zhuxian Lian, and Biao Wang, "Radar and ais track association integrated track and scene features through deep learning," *IEEE Sensors Journal*, vol. 23, no. 7, pp. 8001–8009, 2023.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[26] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[27] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*, 2017, pp. 933–941.