Hindawi
Journal of Sensors
Volume 2022, Article ID 8556477, 20 pages
https://doi.org/10.1155/2022/8556477

Research Article

Multiradar Joint Tracking of Cluster Targets Based on Graph-LSTMs

Xirui Xue, Shucai Huang, Daozhi Wei, and Jiahao Xie

Air and Missile Defense Collage, Airforce Engineering University, Xi'an 710051, China

Correspondence should be addressed to Xirui Xue; rayngu@126.com

Received 11 August 2022; Revised 19 September 2022; Accepted 31 October 2022; Published 14 November 2022

Academic Editor: Giovanni Diraco

The cluster target brings a serious challenge to the traditional multisensor multitarget tracking algorithm because of its large number of members and the cooperative interaction between members. Using multiradar joint tracking cluster target is an alternative method to solve the problem of cluster target tracking, but it inevitably brings the problem of radar-target assignment and tracking information fusion. Aiming at the problem of radar-target assignment and tracking information fusion, a joint tracking method based on graph-long short-term memory neural nets (Graph-LSTMs) is proposed. Firstly, we use multivariable stochastic differential equations (SDE) to model the cooperative interaction of cluster members and transform the derived state space model of cluster members into the same form as the constant velocity (CV) motion model, and the target state equation of cluster which can be used for Bayesian filtering iteration is established. Secondly, based on the detection relationship between radars and cluster members, we introduce the detection confirmation matrix and propose a radar-target assignment method to achieve multiple measurements of single member and detection coverage of all cluster members. Then, each radar uses δ-GLMB filter to estimate the motion state of the assigned targets. Finally, on the basis of spatial discretization, the labels of multiple estimates of cluster member states are obtained. We use the designed Graph-LSTMs to learn the cooperative relationship between target states to fuse the labels and obtain better tracking effect. The experimental results show that the proposed method effectively simulates the cluster motion and realizes the joint estimation of cluster target motion state by multiradar. Our method makes up for the defect that a single radar cannot stably track adjacent multiple targets and achieves better estimation fusion effect than the expectation-maximization (EM) algorithm and mean method.

1. Introduction

The cluster target represented by UAV cluster is the research hotspot of target tracking. It has different tracking characteristics from multitarget and extended target. Compared with multitarget, there is cooperative interaction among cluster members, and the movement of members is consistent. However, due to the large number of members and the small distance between members, it is easy to lead to serious crossing of detection gates and difficult to carry out measurement association. Compared with the extended target, the cluster target is "non-rigid," and the size and shape of the target are easy to change.

At present, according to whether sensors can resolve cluster members, cluster target tracking can be divided into two research routes: cluster member tracking and overall cluster tracking. When the cluster is far away from the sensor and the members cannot be completely resolved, the common method is to model the cluster extension shape as an ellipse that is represented by a random matrix obeying the inverse Wishart distribution [1]. In this method, there is no need to consider the collaborative interaction between members but to track the cluster as a whole.

When the members become resolvable to the sensor, the problem to be solved in cluster target tracking is the accurate state estimation of each cluster member. When solving this problem, the comprehensive application of various measurement methods [2] can ensure the resolution of the target to the greatest extent. For an example, multiradar networking, partition detection, and data fusion in the detection process are common methods in the tracking of UAV cluster.

However, when using these methods, we need to solve not only the problem of accurate tracking of individual member but also the data fusion problem inevitably brought by multisensor joint tracking [3].

This paper would focus on the tracking of resolvable cluster targets. All clusters in this paper refer to resolvable clusters. The first step to achieve accurate cluster target tracking is to obtain the prior information of cluster motion. Therefore, it is necessary to accurately establish the cluster target motion model. In terms of the general law of cluster movement, Reynolds summarizes that the cooperative rule is three different actions of separation, alignment, and cohesion between cluster members based on the distance from each other [4]. Viscek et al. study the speed consistency of cluster members and put forward the Viscek model [5]. In the model, they believe that the velocity consistency is achieved because each member adjusts the velocity direction to the average value of the velocity direction of other members in its neighborhood. Couzin et al. study the phenomenon of effective guidance and group decision-making and discuss the splitting phenomenon of clusters [6]. However, clusters described only by the above rules are prone to fracture due to the dispersion of member positions. Under the above rules, Olfati-Saber [7] adds the virtual leader and proposes the Olfati-Saber model to solve the problem of cluster fracture. The typical target we aim at in this paper is UAV cluster. The main description methods of UAV cluster motion include pilot following method [8], virtual structure method [9], behavior-based method [10], graph-based method [11], and consistency theory [12]. At present, most of the methods to describe UAV cluster motion are from the perspective of cluster control, which is difficult to adapt to the Bayesian iterative estimation process. In order to solve this problem, we try to deduce the motion model of a single-cluster member described in the state space.

Radar-target assignment is another problem to be considered in the joint tracking of cluster members using multiradar [13, 14]. Under the limited radar resources, it is a contradiction to improve the detection density and ensure the full coverage of radar system to all cluster members. In this paper, the concept of detection confirmation matrix is introduced, and radar-target assignment problem is briefly discussed based on the concept.

Aiming at the multitarget tracking (MTT) problem in the clutter environment, the multitarget tracking algorithm represented by the probabilistic data association [15] uses tracking gate to distinguish measurement and clutter and preliminarily solves the correlation problem between measurement and target state. Reference [16] proposes a multitarget tracking algorithm based on maximum entropy fuzzy C-means clustering joint probabilistic data association, which avoids confirmation matrix splitting and reduces the computational load of the JPDA algorithm. However, due to the large number of members in the cluster, the data association is very difficult, and only relying on the confirmation matrix for measuring data association is difficult to meet the requirements of tracking accuracy. At this time, due to the avoidance of data association, tracking methods based on random finite sets (RFS) gradually rise and are widely used [17]. These methods include probability hypothesis density (PHD) filter [18, 19], cardinalized probability hypothesis density (CPHD) filter [20, 21], multitarget multi-Bernoulli (MeMBer) filter [22, 23], and generalized label multi-Bernoulli (GLMB) filter [24, 25]. Reference [26] uses the MCMC particle filter algorithm to track the cooperative cluster with few members and solves the problems of cluster structure inference and joint estimation of cluster and member state. Reference [27] uses δ-GLMB filter to track cluster targets with splitting, merging, and reorganization behavior. Usually, RFS-based tracking methods set the detection probability as a constant a priori, but this does not conform to the actual sensor detection process. Reference [28] uses active sonar equation to model the detection probability and proposes the Pd-GM-PHD and Pd-GM-CPHD algorithms to achieve more accurate tracking effect. The existing filter based on RFS avoids the problem of data association and realizes the joint estimation of the number of targets and states. Compared with the probabilistic data association algorithm, the number of targets that can be tracked is increased. However, these methods cannot meet the requirements of the number of tracking cluster members only by using a single sensor, and they do not consider the problem of tracking information fusion caused by using multiple sensors in the actual situation.

The traditional method of multiradar tracking information fusion is to approximate the posterior distribution of the target by Gaussian mixture. A fusion method of Gaussian mixture distribution using the Chebyshev information is proposed in [29]. A consensus CPHD filter based on Gaussian mixture distribution is designed in [30]. And a likelihood function distribution approximation method based on polynomial weighting is designed in [31]. However, these methods do not further consider the useful information that the relationship between targets may provide. In recent years, machine learning algorithms based on neural networks provide a new way for mining information in target tracking [32, 33].

In this paper, in order to achieve better tracking effect, we try to use Graph-LSTMs [34–36] to realize the fusion of multiradar estimation information and mine the collaborative information among cluster members at the same time. The key contributions of this research are as follows.

(1) We establish a stochastic differential equation model to describe the cooperative interaction relationship within the cluster and unify it under the framework of CV model, which would be used as the target state equation of the cluster in the iterative filtering process

(2) Considering the problem of multiradar detection of cluster targets, we propose a radar-target assignment method. Benefiting from the introduction of detection confirmation matrix, this method can not only ensure that multiple radar systems detect all cluster members but also realize multiple detection of the same member, so as to obtain multiple target state estimates

(3) Based on the cluster motion model proposed in the paper, the cluster motion trajectory dataset is

simulated. We use δ-GLMB filter to realize the state estimation of radar to assigned targets

(4) The bipartite graph of radar detection relation is established and expanded into a tree structure. The space around the predicted value of the target state is discretized to obtain the label of the target state estimated value. The Graph-LSTM model is used to fuse the label of the estimated value, and a more accurate fusion value is obtained. Experimental results show that our method has more accurate fusion labels and smaller OSPA distance than the mean method and EM algorithm

The content of this paper is arranged as follows: Section 2 introduces the construction method of the cooperative interaction model based on multivariable stochastic differential equation. Section 3 introduces the multiradar tracking method of cluster targets and puts forward a radar-target assignment method. On this basis, the state estimation of the assigned targets is realized by using the δ-GLMB filter. Section 4 introduces an estimation fusion method based on Graph-LSTMs. In Section 5, the cluster motion simulation is carried out, and the estimation fusion experiment is carried out using the simulation data set to verify the effectiveness of the proposed method.

2. Cooperative Interaction Motion Model

In this section, on the basis of cluster cooperation rules, the self-organizing cooperative interaction motion model of cluster is established, and its discrete form in state space is deduced. In the model, the motion of cluster members is directly related to other members in the neighborhood, and cluster members act according to the motion state of other members. Finally, the velocity of cluster members is gradually consistent, and there is no collision between members. Multivariable stochastic differential equation (SDE) would be an effective tool for us to establish cluster member cooperative interaction motion model [37].

2.1. Continuous-Time Collaborative Interaction Model. Considering that the motion state of cluster members is jointly determined by their own state and the states of other members in the neighborhood, the continuous-time collaborative interaction motion model of cluster member i at time t is constructed as follows:

$$d\dot{s}_{t,i} = \left\{-\alpha[\dot{s}_{t,i} - f(\dot{s}_t)] + \beta r_i(s_{t,i})\right\}dt + dW^s_{t,i} + dB^s_t, \quad (1)$$

where $s_{t,i}$ and $\dot{s}_{t,i}$, respectively, represent the position and velocity of the target i at time t, α represents the velocity control parameter, β represents the potential force control parameter, $W^s_{t,i}$ represents the motion noise of the members, B^s_t represents the cluster motion noise, and $f(\dot{s}_t)$ represents the average velocity of the cluster where target i is located. Let the number of cluster members be N, and the expression of $f(\dot{s}_t)$ is as follows:

$$f(\dot{s}_t) = \frac{1}{N}\sum_{j=1}^{N} \dot{s}_{t,j}. \quad (2)$$

The introduction of $f(\dot{s}_t)$ makes the average velocity information of the cluster pass through the whole cluster, makes the topology of the cluster structure have weak connectivity, and realizes the gradual consistency of the velocity of members.

$r_i(s_{t,i})$ represents the total potential force of member i by other members in the neighborhood, which can avoid collision between cluster members. Its expression is as follows:

$$r_i(s_t) = \sum_{\forall j, j\neq i} r\left(s_{t,i}, s_{t,j}\right), \quad (3)$$

where $r(s_{t,i}, s_{t,j})$ represents the potential force of member j to member i in the cluster, which is generated by the potential function $U_j(\cdot)$. $U_j(\cdot)$ needs to meet the following two properties. First, $U_j(\cdot)$ should be a continuous, differentiable, and nonnegative function; second, $U_j(\cdot)$ needs to obtain a unique minimum value at a certain required distance. In this paper, a definition of $U_j(\cdot)$ is given as follows:

$$U_j(d) = \begin{cases} R_{11}\ln\dfrac{d_b + R_{12}}{d + R_{12}} + \dfrac{R_{11}}{d_b + R_{12}}(d - d_b), & d \le d_b, \\ R_{21}\ln\dfrac{d_m - d_b + R_{22}}{-d + d_m + R_{22}} + \dfrac{R_{21}}{d_m - d_b + R_{22}}(d_b - d), & d_b < d < d_m. \end{cases} \quad (4)$$

Here, $d = \|s_{t,i} - s_{t,j}\|$ and d_r are the balance distance between two members. When the distance is less than d_r, the potential force is repulsive force. Otherwise, it is attractive force. d_m is the maximum distance that can generate potential force. The cluster potential field is shown in Figure 1. R_{11}, R_{12}, R_{21}, and R_{22} are the parameters controlling the intensity of potential field. Under the above definition of potential function, as the negative gradient of potential function, the potential force $r(s_{t,i}, s_{t,j})$ is calculated as follows:

$$r\left(s_{t,i}, s_{t,j}\right) = -\operatorname{grad}\left(U_j(d)\right) = \begin{cases} \dfrac{R_{11}}{d + R_{12}} - \dfrac{R_{11}}{d_b + R_{12}}, & d \le d_b, \\ \dfrac{-R_{21}}{d_m - d + R_{22}} + \dfrac{R_{21}}{d_m - d_b + R_{22}}, & d_b < d < d_m. \end{cases} \quad (5)$$

2.2. Continuous-Time Joint Motion Model. Because the cooperative interaction model of cluster member contains the motion information of other members in cluster, it cannot be solved separately. The models of all members in the cluster need to be combined to obtain the analytical solution together.

In 2-dimensional space, member position $s_{t,i} = [x_{t,i}, y_{t,i}]$. Use X_t to represent the joint state of all members in the

cluster; then, we can get $X_t = [x_{t,1}, \dot{x}_{t,1}, y_{t,1}, \dot{y}_{t,1}, \cdots, x_{t,N}, \dot{x}_{t,N}, y_{t,N}, \dot{y}_{t,N}]^T$. From (1), the linear differential equation of the joint state in continuous time can be obtained as follows:

$$dX_t = AX_t dt + Rdt + CdW_t + DdB_t. \tag{6}$$

Matrix $A \in R^{4N\times 4N}$ is defined as follows:

$$A = \begin{bmatrix} A_1 & A_2 & \cdots & A_2 \\ A_2 & A_1 & \cdots & A_2 \\ \vdots & & \ddots & \vdots \\ A_2 & A_2 & \cdots & A_1 \end{bmatrix},$$

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \dfrac{(1-N)\alpha}{N} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & \dfrac{(1-N)\alpha}{N} \end{bmatrix}, \tag{7}$$

$$A_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \dfrac{\alpha}{N} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dfrac{\alpha}{N} \end{bmatrix}.$$

Matrix $R \in R^{4N\times 1}$ is defined as follows:

$$R = \left[0, \beta r_{i,x}(s_t), 0, \beta r_{i,y}(s_t), \cdots, 0, \beta r_{N,x}(s_t), 0, \beta r_{N,y}(s_t)\right]^T. \tag{8}$$

Matrix $C \in R^{4N\times 2N}$ is defined as follows:

$$C = \begin{bmatrix} C_1 & 0 & \cdots & 0 \\ 0 & C_1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & C_1 \end{bmatrix},$$

$$C_1 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}. \tag{9}$$

Matrix $D \in R^{4N\times 2}$ is defined as follows:

$$D = \left[C_1^T, C_1^T, \cdots, C_1^T\right]^T. \tag{10}$$

Motion noise $W_t = [W_{t,1}^x, W_{t,1}^y, \cdots, W_{t,N}^x, W_{t,N}^y]^T$, and $B_t = [B_t^x, B_t^y]^T$. According to the general practice, motion noise is usually modeled as white noise with Gaussian distribution,

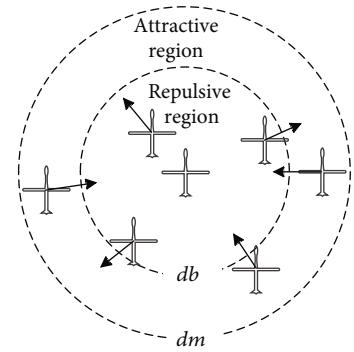

Figure 1: Schematic diagram of potential field.

and their covariance matrices are $Q_W = \text{diag}([\sigma_x^2, \sigma_y^2, \cdots, \sigma_x^2, \sigma_y^2])$ and $Q_B = \text{diag}([\sigma_g^2, \sigma_g^2])$, respectively. Because they have the same form, in order to simplify the model, the two noises can be combined into cluster joint motion noise M_t and its covariance matrix $Q_M = \text{diag}([\sigma_x^2, \sigma_y^2, \cdots, \sigma_x^2, \sigma_y^2, \sigma_g^2, \sigma_g^2])$. Accordingly, after combining the two coefficient matrices, there is $H = [C, D]$. Equation (6) has the following form:

$$dX_t = AX_t dt + Rdt + HdM_t. \tag{11}$$

The following analytical solution is obtained by solving (11).

$$X_{t+\tau} = e^{A\tau}X_t + \int_{h=t}^{h=t+\tau} e^{A(t+\tau-h)}dh \times R + \int_{h=t}^{h=t+\tau} e^{A(t+\tau-h)}HdM_t, \tag{12}$$

where τ is the sampling time. Define the member joint state transition matrix $F_c(\tau) = e^{A\tau}$. Define the potential force joint gain matrix $E_c(\tau) = \int_{h=t}^{h=t+\tau} e^{A(t+\tau-h)}dh$, and the noise joint covariance matrix is calculated as follows:

$$Q_c(\tau) = \int_{h=t}^{h=t+\tau} e^{A(t+\tau-h)}HQ_M H^T\left(e^{A(t+\tau-h)}\right)^T dh. \tag{13}$$

It can be seen from (12) that $E_c(\tau)$ and $Q_c(\tau)$ are only related to the sampling time τ. Therefore, the time discretization of (12) can be realized by making $\tau = 1$, and the following equation is obtained:

$$X_{k+1} = F_c X_k + E_c R + \Gamma_c w_k. \tag{14}$$

Equation (14) is the discrete-time joint state model of cluster, where $F_c = F_c(1)$ and $E_c = E_c(1)$; w_k is the Gaussian white noise with covariance matrix $Q_c = Q_c(1)$; and Γ_c is the state noise factor matrix, which is used to convert the noise dimension and add the noise to the joint state X_{k+1}.

2.3. Discrete-Time State Space Model. After obtaining discrete-time joint state model of cluster, it is necessary to

separate the state space model of a single member from the joint state model and further simplify it to a general form suitable for iterative tracking algorithm. By analyzing the coefficient matrix and covariance matrix, we can see that they have the following symmetric forms, which would be the basis for us to realize the above separation.

$$F_c = \begin{bmatrix} F_c^s & F_c^c & \cdots & F_c^c \\ F_c^c & F_c^s & \cdots & F_c^c \\ \vdots & & \ddots & \vdots \\ F_c^c & F_c^c & \cdots & F_c^s \end{bmatrix},$$
$$E_c = \begin{bmatrix} E_c^s & E_c^c & \cdots & E_c^c \\ E_c^c & E_c^s & \cdots & E_c^c \\ \vdots & & \ddots & \vdots \\ E_c^c & E_c^c & \cdots & E_c^s \end{bmatrix}, \quad (15)$$
$$Q_c = \begin{bmatrix} Q_c^s & Q_c^c & \cdots & Q_c^c \\ Q_c^c & Q_c^s & \cdots & Q_c^c \\ \vdots & & \ddots & \vdots \\ Q_c^c & Q_c^c & \cdots & Q_c^s \end{bmatrix}.$$

Among them, the diagonal matrices with superscript s are related to the motion state of the members themselves, and other matrices with superscript c are related to the states of other members in the cluster.

Further solving matrix F_c shows that for any member i, its own state transition matrix F_c^s and the cooperation matrix F_c^c are as follows:

$$F_c^s = \begin{bmatrix} 1 & f_1 & 0 & 0 \\ 0 & f_2 & 0 & 0 \\ 0 & 0 & 1 & f_1 \\ 0 & 0 & 0 & f_2 \end{bmatrix},$$
$$F_c^c = \begin{bmatrix} 0 & \frac{1-f_1}{N-1} & 0 & 0 \\ 0 & \frac{1-f_2}{N-1} & 0 & 0 \\ 0 & 0 & 0 & \frac{1-f_1}{N-1} \\ 0 & 0 & 0 & \frac{1-f_2}{N-1} \end{bmatrix}. \quad (16)$$

f_1 and f_2 are constants, which are related to the member number N and parameter α. Assuming that the influence of potential force on the motion state of member i at a certain time is only related to the potential force on member i at the current time, there are

$$E_c^s = \begin{bmatrix} 1 & \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{2} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (17)$$

And $E_{ct}^c = O_{4\times4}$. $O_{4\times4}$ represents the fourth-order zero matrix.

In the cluster movement dominated by collaborative behavior, the motion randomness of other members has little impact on the member i and would offset each other due to synergy. The motion randomness of members is more expressed as endogenous motion uncertainty. Therefore, it can be considered that the noise covariance matrix of member i can be directly simplified to Q_{ct}^s.

Based on the above assumptions, the discrete-time state space model of cluster member i can be given as follows:

$$x_{k+1,i} = F_c^s x_{k,i} + F_c^c \sum_{\forall j, j\neq i} x_{k,j} + c_{k,i} + \Gamma_c^s w_{k,i}, \quad (18)$$

where

$$c_{k,i} = \sum_{\forall j, j\neq i} c_{k,(i,j)} = \left[\frac{\beta r_{i,x}}{2}, \beta r_{i,x}, \frac{\beta r_{i,y}}{2}, \beta r_{i,y}\right]^T,$$
$$c_{k,(i,j)} = \left[\frac{\beta r_{i,x}(x_{k,i}, x_{k,j})}{2}, \beta r_{i,x}(x_{k,i}, x_{k,j}), \frac{\beta r_{i,y}(x_{k,i}, x_{k,j})}{2}, \beta r_{i,y}(x_{k,i}, x_{k,j})\right]^T. \quad (19)$$

Γ_c^s is the state noise factor matrix, and $w_{k,i}$ is the Gaussian noise with covariance matrix Q_{ct}^s.

Although the above equation separates the motion states of member i from other members in the cluster, the number of members still affects the values of f_1 and f_2.

In order to completely separate the influence of collaborative interaction, simple transformation is carried out under the framework of constant velocity (CV) motion model, and the discrete-time state space model of cluster members is proposed as follows:

$$x_{k+1,i} = F_k x_{k,i} + w_{k,i}^0. \quad (20)$$

Here,

$$F_k = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (21)$$

$w_{k,i}^0$ is a new collaborative noise, which includes all the

effects of cooperative interaction and motion noise on a cluster member motion state. Its calculation method is shown in (22).

$$w_{k,i}^0 = \Delta c_{k,i} + \Gamma_k w_{k,i}. \tag{22}$$

Here, $\Gamma_k = \Gamma_c^s$ and $\Delta c_{k,i}$ are calculated as follows:

$$\Delta c_{k,i} = \sum_{\forall j, j \neq i} \omega_k \left[\frac{c_{k,(i,j)}}{\omega_k} - F_k \left(\dot{x}_{k,i} - \dot{x}_{k,j} \right) \right], \tag{23}$$

$$\omega_k = \frac{1}{N-1} \left(I_{4\times 4} - \text{diag}\left([f_1, f_2, f_1, f_2] \right) \right). \tag{24}$$

In (23), $\dot{x}_{k,i}$ is the velocity vector and $\dot{x}_{k,i} = [0, \dot{x}_{k,i}, 0, \dot{y}_{k,i}]^T$.

It is assumed that the interaction is a random variable obeying to Gaussian distribution, that is, $c_{k,(i,j)} \sim \mathcal{N}(\bar{c}_{k,(i,j)}, S_{k,(i,j)})$. Then, the collaborative noise $w_{k,i}^0$ can also be modeled as Gaussian distribution, that is, $w_{k,i}^0 \sim \mathcal{N}(\mu_{k,i}^0, Q_{k,i}^0)$.

Here,

$$\mu_{k,i}^0 = \sum_{\forall j, j \neq i} \omega_k \left[\frac{\bar{c}_{k,(i,j)}}{\omega_k} - F_k \left(\hat{\dot{x}}_{k,i} - \hat{\dot{x}}_{k,j} \right) \right],$$

$$Q_{k,i}^0 = \sum_{\forall j, j \neq i} S_{k,(i,j)} + \omega_k F_k \left[\sum_{\forall j, j \neq i} \left(P_{k,j} - P_{k,i} \right) \right] F_k^T \omega_k^T + \Gamma_k Q_c^s \Gamma_k^T, \tag{25}$$

where $P_{k,i}$ is the state covariance matrix of $\dot{x}_{k,i}$. The whole process of model construction is shown in Figure 2. The model construction has experienced the transformation from member to cluster and then to member and realizes the discretization of continuous-time model.

However, even if model (20) is obtained, it is difficult to directly apply the traditional tracking algorithm. There are two reasons. On the one hand, the noise $w_{k,i}^0$ is not Gaussian white noise, but the noise with time-varying mean and covariance. On the other hand, due to the large number of members in the cluster and the limited number of radar channels, it is often difficult for a single radar to track the whole cluster, and it is prone to the problem of measurement-track association. In order to solve the above problems, multiradar joint tracking becomes an alternative method.

3. Multiradar Joint Tracking Method

In order to solve the problem of obtaining measurement information in the process of cluster target detection and reduce the impact of collaborative noise on cluster tracking, the following solutions are proposed in this paper. Firstly, as shown in Figure 3, multiple radars are used to detect a cluster at the same time, and each radar obtains some measurement information of cluster, respectively. The sum of all measurement information needs to ensure full coverage of all cluster members. Secondly, the states of the members are estimated by using the multitarget filter. Finally, the state estimation of each member is obtained by information fusion. In order to ensure the effect of information fusion, each member needs to estimate as much as possible.

3.1. Acquisition of Detection Information

3.1.1. Detection Relationship Construction. In order to reflect the detection relationship between multiradar and cluster members, we introduce the concept of detection confirmation matrix Ω, which is defined as

$$\Omega = [\omega_{st}], \quad s = 1, 2, \cdots, M, t = 1, 2, \cdots, T, \tag{26}$$

where ω_{st} is a binary variable. $\omega_{st} = 1$ indicates that cluster member t is detected by radar s. And $\omega_{st} = 0$ indicates that cluster member t is not detected by radar s. The average detection density A_d is defined as the average detection times of all members in the cluster. Ω needs to meet the following three assumptions:

(1) Each member is detected by at least one radar, i.e.,

$$\sum_{s=1}^{M} \omega_{st} \geq 1, \quad t = 1, 2, \cdots, N \tag{27}$$

(2) The number of members detected by each radar is less than the number of radar channels N_c, i.e.,

$$\sum_{t=1}^{N} \omega_{st} \leq N_c, \quad s = 1, 2, \cdots, M \tag{28}$$

(3) The average detection density A_d should be greater than the given value $\bar{A}_d$, i.e.,

$$A_d = \frac{\sum_{t=1}^{N} \sum_{s=1}^{M} \omega_{st}}{N \geq \bar{A}_d} \tag{29}$$

3.1.2. Radar Measurement Model. When the radar detection relationship is determined, the measurement information model of the target follows the general measurement model. For member i, there are the following measurement models:

$$z_{k+1,i} = H_{k+1} x_{k+1,i} + v_{k+1,i}, \tag{30}$$

where

$$H_{k+1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{31}$$

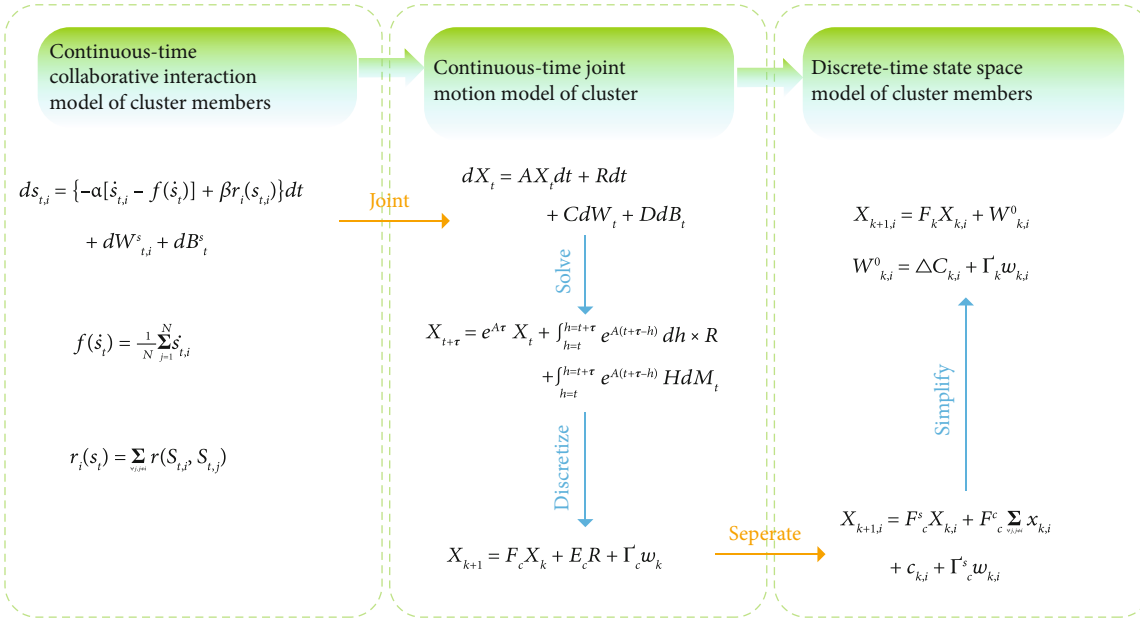

Figure 2: Model construction process.

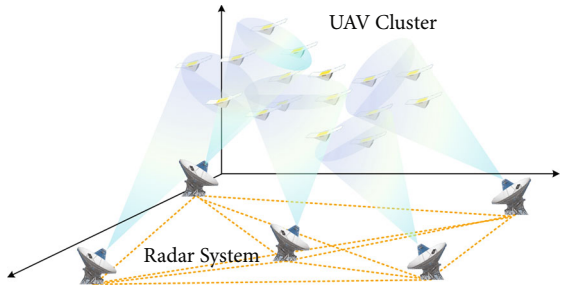

Figure 3: Multiradar detection.

The measurement noise $v_{k+1,i}$ is a Gaussian white noise with covariance matrix R_{k+1}.

Equations (20) and (30) together constitute the system equation of the cluster target tracking algorithm. Considering the problem of radar-target assignment and tracking information fusion, the filter needs to have the ability to propagate multiple hypotheses in the label set of different trajectories and output the target label. We choose the δ-generalized label multi-Bernoulli (δ-GLMB) filter [38, 39] for target tracking.

3.2. Application of δ-GLMB Filter. In the process of cluster target tracking, an accurate state estimation algorithm should include the estimation of cooperative interaction. However, we usually cannot obtain the priori information of cooperative interaction. Therefore, we use the method of multiradar estimation fusion to solve this problem, and each radar runs a δ-GLMB filter.

The δ-GLMB filter recursively transmits the δ-GLMB filtering density forward over time through Bayesian prediction and update equation. The difference from GLMB is that when predicting, δ-GLMB is summed on the label space $(I_+, \vartheta) \in \mathcal{F}(\mathbb{L}_+) \times \Xi$ at the next time. Here, $\mathbb{L}_+ = \mathbb{L} \cup \mathbb{B}$. From the perspective of multitarget tracking, this is more intuitive because it shows how the prediction introduces a new target label. δ-GLMB is represented by parameter set $\{(w^{(I,\vartheta)}, p^{(\vartheta)}): (I_+, \vartheta) \in \mathcal{F}(\mathbb{L}_+) \times \Xi\}$. The parameter set of δ-GLMB can be regarded as an enumeration of all assumptions and their related weights and track density $\{(I^{(h)}, \vartheta^{(h)}, w^{(h)}, p^{(h)})\}_{h=1}^{H}$.

3.2.1. Prediction. At time k, the distribution of the random finite set of δ-GLMB is as follows:

$$\pi(X) = \Delta(X) \sum_{(I,\vartheta)\in F(\mathbb{L})\times\Xi} w^{(I,\vartheta)} \delta_I(\mathcal{L}(X)) \left[p^{(\vartheta)}\right]^X. \tag{32}$$

Then, the multitarget prediction density at time $k+1$ is

$$\pi_+(X_+) = \Delta(X_+) \sum_{(I_+,\vartheta)\in F(\mathbb{L}_+)\times\Xi} w_+^{(I_+,\vartheta)} \delta_{I_+}(\mathcal{L}(X_+)) \left[p_+^{(\vartheta)}\right]^{X_+}, \tag{33}$$

where

$$\begin{aligned}
w_+^{(I_+,\vartheta)} &= w_\gamma(I_+ \cap \mathbb{B}) w_S^{(\vartheta)}(I_+ \cap \mathbb{L}),\\
w_s^{(\vartheta)}(L) &= \left[\eta_S^{(\vartheta)}\right]^L \sum_{I\supseteq L} \left[1-\eta_S^{(\vartheta)}\right]^{I-L} w^{(I,\vartheta)},\\
\eta_S^{(\vartheta)}(l) &= \left\langle p_S(\cdot,l), p^{(\vartheta)}(\cdot,l)\right\rangle,\\
p_+^{(\vartheta)}(x,l) &= 1_{\mathbb{L}}(l) p_{+,S}^{(\vartheta)}(x,l) + 1_{\mathbb{B}}(l) p_\gamma(x,l),\\
p_{+,S}^{(\vartheta)}(x,l) &= \frac{\left\langle p_S(\cdot,l)\phi(x|\cdot,l), p^{(\vartheta)}(\cdot,l)\right\rangle}{\eta_S^{(\vartheta)}(l)}.
\end{aligned} \tag{34}$$

Here, $w_\gamma(I_+ \cap \mathbb{B})$ is the weight of newborn label $(I_+ \cap \mathbb{B})$, and $w_S^{(\vartheta)}(I_+ \cap \mathbb{L})$ is the weight of survival label. $\phi(x|\cdot, l)$ is the single-target transfer density of trajectory l, and $\eta_S^{(\vartheta)}(l)$ is the survival probability of trajectory l. $p_\gamma(x, l)$ is the probability density of newborn state, and $p_{+,S}^{(\vartheta)}(x, l)$ is the probability density of survival state.

3.2.2. Update. After obtaining the multitarget prediction density at time $k+1$, it can be proved that the updated multitarget filter density is also δ-GLMB.

The update process is shown as follows:

$$\pi(X|Z) = \Delta(X) \sum_{(I,\vartheta)\in F(\mathbb{L})\times\Xi} \sum_{\theta\in\Theta(I)} w^{(I,\vartheta,\theta)}(Z)\delta_I(\mathrm{L}(X)) \left[p^{(\vartheta,\theta)}(\cdot\,|Z)\right]^X, \tag{35}$$

where

$$\begin{aligned} w^{(I,\vartheta,\theta)}(Z) &= \frac{\delta_{\theta^{-1}(\{0:|Z|\})}(I)\left[\eta_Z^{(\vartheta,\theta)}\right]^I w^{(I,\vartheta)}}{\sum_{(I,\vartheta)\in F(\mathbb{L})\times\Xi}\sum_{\theta\in\Theta(I)}\delta_{\theta^{-1}(\{0:|Z|\})}(I)\left[\eta_Z^{(\vartheta,\theta)}\right]^I w^{(I,\vartheta)}} \propto \left[\eta_Z^{(\vartheta,\theta)}\right]^I w^{(I,\vartheta)}, \\ \eta_Z^{(\vartheta,\theta)}(l) &= \left\langle p^{(\vartheta)}(\cdot\,,l), \varphi_Z(\cdot\,,l;\theta)\right\rangle, \\ p^{(\vartheta,\theta)}(x,l|Z) &= \frac{p^{(\vartheta)}(x,l)\varphi_Z(x,l;\theta)}{\eta_Z^{(\vartheta,\theta)}(l)}. \end{aligned} \tag{36}$$

Here, $\Theta(I)$ represents a subset of the current association mapping with domain I. $\varphi_Z(x, l; \theta)$ is defined as follows:

$$\begin{aligned} \varphi_Z(x,l;\theta) &= \frac{\delta_0(\theta(l))(1-p_D(x,l)) + (1-\delta_0(\theta(l)))p_D(x,l)g\left(z_{\theta(l)}|x,l\right)}{\kappa\left(z_{\theta(l)}\right)} \\ &= \begin{cases} \dfrac{p_D(x,l)g\left(z_{\theta(l)}|x,l\right)}{\kappa\left(z_{\theta(l)}\right)}, & \theta(l) > 0, \\ 1-p_D(x,l), & \theta(l) = 0. \end{cases} \end{aligned} \tag{37}$$

Here, $p_D(x, l)$ is the detection probability on (x, l). $g(z_{\theta(l)}|x, l)$ is the single-target likelihood function of $z_{\theta(l)}$ to (x, l). $\kappa(\cdot)$ is the Poisson clutter density.

3.2.3. Realization of δ-GLMB Filter. For cluster member motion model, $p_S(x, l) = p_S$, and $\phi(x_+|x, l) = N(x_+; F_k x, Q_c^s)$. Since the newborn density matrix is in the form of Gaussian mixture, if the single-target density is in the form of Gaussian mixture, there is

$$p^{(\vartheta)}(\cdot\,,l) = \sum_{i=1}^{J^{(\vartheta)}(l)} \omega_i^{(\vartheta)}(l)\mathcal{N}\left(x; m_i^{(\vartheta)}(l), P_i^{(\theta)}(l)\right). \tag{38}$$

Then,

$$\begin{aligned} \eta_S^{(\vartheta)}(l) &= p_S, \\ p_+^{(\vartheta)}(x,l) &= 1_{\mathbb{L}}(l)\sum_{i=1}^{J^{(\vartheta)}(l)} \omega_i^{(\vartheta)}(l)\mathcal{N}\left(x; m_{S,i}^{(\vartheta)}(l), P_{S,i}^{(\vartheta)}(l)\right) + 1_{\mathbb{B}}(l)p_\gamma^{(l)}, \end{aligned} \tag{39}$$

where

$$\begin{aligned} m_{S.i}^{(\vartheta)}(l) &= F_k m_i^{(\vartheta)}(l), \\ P_{S,i}^{(\vartheta)}(l) &= F_k P_i^{(\vartheta)}(l)F_k^T + Q_c^s. \end{aligned} \tag{40}$$

When the motion model parameters are related to the label, we just need to substitute $p_S = p_S(l)$, $F_k = F_k(l)$, and $Q_c^s = Q_c^s(l)$ into the above equation.

In the update process, $p_D(x, l) = p_D$. For the measurement model shown in (30), the measurement likelihood is $g(z|x, l) = \mathcal{N}(z; H_{k+1}x, R_{k+1})$. Gaussian mixture expression provides the most general configuration for linear Gaussian model. It is assumed that each single-target density $p^{(\vartheta)}(\cdot\,,l)$ is in the form of Gaussian mixture.

$$p^{(\vartheta)}(\cdot\,,l) = \sum_{n=1}^{J^{(\vartheta)}(l)} \omega_n^{(\vartheta)}(l)\mathcal{N}\left(x; m_n^{(\vartheta)}(l), P_n^{(\vartheta)}(l)\right). \tag{41}$$

Then,

$$c_{i,j} = -\ln\left[\frac{p_D\sum_{n=1}^{J^{(\vartheta)}(l_i)}\omega_n^{(\vartheta)}(l_i)q_n^{(\vartheta)}\left(z_j; l_i\right)}{(1-p_D)\kappa\left(z_j\right)}\right]. \tag{42}$$

In addition, for the updated association process (ϑ, θ), there are

$$\begin{aligned} \eta_Z^{(\vartheta,\theta)}(l) &= \sum_{n=1}^{J^{(\vartheta)}(l)} \omega_{Z,n}^{(\vartheta,\theta)}(l), \\ p^{(\vartheta,\theta)}(x,l|Z) &= \sum_{n=1}^{J^{(\vartheta)}(l)} \frac{\omega_{Z,n}^{(\vartheta,\theta)}(l)}{\eta_Z^{(\vartheta,\theta)}(l)}\mathcal{N}\left(x; m_{Z,n}^{(\vartheta,\theta)}(l), P_n^{(\vartheta,\theta)}(l)\right). \end{aligned} \tag{43}$$

Here,

$$w_{Z,n}^{(\vartheta,\theta)}(l) = \omega_n^{(\vartheta)}(l) \times \begin{cases} \dfrac{p_D q_n^{(\vartheta)}\left(z_{\theta(l)}; l\right)}{\kappa\left(z_{\theta(l)}\right)}, & \theta(l) > 0, \\ (1 - p_D), & \theta(l) = 0, \end{cases}$$

$$q_n^{(\vartheta)}(z; l) = \mathcal{N}\left(z; H_{k+1} m_n^{(\vartheta)}(l), H_{k+1} P_n^{(\vartheta)}(l) H_{k+1}^T + R_{k+1}\right),$$

$$m_{Z,n}^{(\vartheta,\theta)}(l) = \begin{cases} m_n^{(\vartheta)}(l) + G_n^{(\vartheta,\theta)}(l)\left(z_{\theta(l)} - H_{k+1} m_n^{(\vartheta)}(l)\right), & \theta(l) > 0, \\ m_n^{(\vartheta)}(l), & \theta(l) = 0, \end{cases}$$

$$P_n^{(\vartheta,\theta)}(l) = \left(I - G_n^{(\vartheta,\theta)}(l) H_{k+1}\right) P_n^{(\vartheta)}(l),$$

$$G_n^{(\vartheta,\theta)}(l) = \begin{cases} P_n^{(\vartheta)}(l) H_{k+1}^T \left(H_{k+1} P_n^{(\vartheta)}(l) H_{k+1}^T + R_{k+1}\right)^{-1}, & \theta(l) > 0, \\ 0, & \theta(l) = 0. \end{cases} \tag{44}$$

When the parameters of the measurement model are dependent on the label l, it is only necessary to substitute $p_D = p_D(l)$, $H_{k+1} = H_{k+1}(l)$, and $R_{k+1} = R_{k+1}(l)$ into the above equation.

4. Multiradar Estimation Fusion Based on Graph-LSTMs

Through the above process, we can obtain multiple estimations of the same member state. An obvious fact is that in the tracking process, we only consider the explicit relationship between radar and target in detection and tracking but hardly consider the implicit relationship of motion consistency among cluster members which may play a positive role in improving the tracking accuracy. In this section, the purpose of applying the Graph-LSTM algorithm is to capture this implicit relationship.

4.1. Construction Tracking Relationship Graph

4.1.1. Bipartite Graph Model. In order to reflect the detection and tracking relationship between radar and cluster members, we establish the graph of radar and cluster. Radar system and cluster can be divided into two subsets (S, T), and we connect the radar s_i that has detection relationship and obtains the target estimation with the corresponding target t_j by edge $E \subseteq S \times T$, and Figure 4 is as follows. According to the definition of bipartite graph, this graph is a bipartite graph, expressed as $G = \{(S, T), E\}$.

4.1.2. Build Node Unfolding Tree. In the bipartite graph of detection and tracking relationship between radar and target, the size and topology of the graph neighborhood are variable. However, the common machine learning algorithms can only deal with feature vectors with fixed length. In order to apply the machine learning algorithm to mining information, an effective solution is to unfold the bipartite graph into a tree at a given depth and the learning algorithm

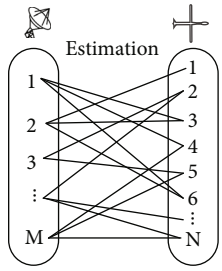

Figure 4: Bipartite graph of radar-target.

is used to learn the sequence from the child node to the root node.

For a bipartite graph, given the depth $D = 2$, the tree T_{t_j} is constructed with the target t_j as the root node, and the unfolding results are shown in Figure 5. In the unfolding tree, the root node depth is 0. The radars with detection and tracking relationship with target t_j are taken as child nodes, and the depth is 1. Other detection and tracking targets of these radars are the child nodes of the radar, with a depth of 2. In the process of building T_{t_j}, when the depth $d > 1$, the same target may be repeatedly detected by other radars. At this time, for the same node k, we would add superscript to represent its copy with k' and k''. From the depth $d \geq 2$, there is an upward transmission of other target motion information in the tree, which makes the acquisition of fusion apply the implicit information between targets.

4.2. State Space Labeling. In this paper, we would apply graph-long short-term memory neural nets (Graph-LSTMs) to sequence learning. The core of Graph-LSTMs is multilevel LSTMs, which is called multilevel sequence learners (MLSL). However, MLSL can only output the label of the target sate, and the continuous state of the target is difficult to adapt to this output form. In order to apply MLSL, it is necessary to label the state space and convert the true state and multiple estimations of the target into label space.

The labeling of state space needs to be carried out in position space and velocity space, respectively. In position space, given the number of labels N_L and taking the predicted position as the center, we can generate an orthogonal grid of $(N_L - 1) \times (N_L - 1)$, as shown in Figure 6. The space around the predicted position is discretized into $(N_L - 1)^2$ small spaces. We call the small space as label space. We use L_p to represent its label, and $L_p = [L_x, L_y]^T$. Similarly, we can get label L_v in velocity space and $L_v = [L_{v_x}, L_{v_y}]^T$. Then, we can obtain the label of the target estimation $L = [L_x, L_{v_x}, L_y, L_{v_y}]^T$.

At time k, the estimated state of target t_j is $x_{k|k,j}$. At time $k + 1$, the label $L_{k+1,j}^i$ of radar s_i estimation $x_{k+1|k+1,j}^i$ for target t_j is calculated as follows:

$$L_{k+1,j}^i = \left\lfloor \left(x_{k+1|k+1,j}^i - x_{k+1|k,j}\right) \odot d_{\Delta L}^{-1} \right\rfloor + L_{k+1|k,j}, \tag{45}$$

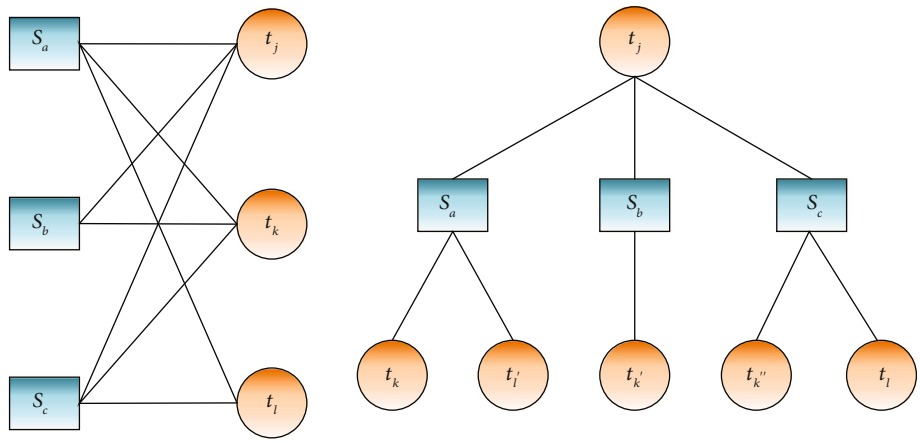

Figure 5: An example of a graph and its unfolding tree at node t_j for depth 2.

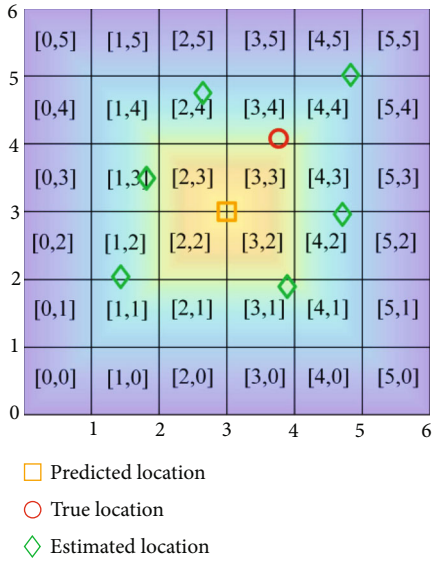

Figure 6: Position space labeling.

where $x_{k+1|k,j}$ is the state prediction at time $k+1$; $L^i_{k+1,j}$ represents the label of the estimation $x^i_{k+1|k+1,j}$ of radar s_i; $L_{k+1|k,j}$ represents the label of $x_{k+1|k,j}$, which is usually used as the central label of label space; $\odot$ represents the product of the corresponding element; and $d_{\Delta L}$ represents the side length of label space, which can reflect the fineness of state space labeling. In the paper, $d_{\Delta L} = [d_{\Delta L_x}, d_{\Delta L_{v_x}}, d_{\Delta L_y}, d_{\Delta L_{v_y}}]^T$. The elements of $d_{\Delta L}$ are preset by us. $\lfloor \cdot \rfloor$ means rounding down. Therefore, the smallest label is 0. After the label space is given, if the calculated target estimation label is outside the label space, the estimation would be discarded.

4.3. Sequence Learning Using MLSL

4.3.1. Learning Strategy. Next, we would traverse the tree T_{t_j} from bottom to top; use the MLSL to calculate the state label y_{t_j} of root node. In the process of network training, y_{t_j} is compared with the expected state label $\bar{y}_{t_j}$, the loss is calculated, and backpropagation is carried out to train the network parameters.

4.3.2. MLSL Model. LSTM is the basis for efficient sequence learning in MLSL, which can alleviate the problem of gradient disappearance with the increase of time step in recurrent neural network. The basis of sequence learning ability of LSTM is the introduction of three gates in its basic unit. The structure and operation flow of LSTM unit are shown in Figure 7.

In the picture, i_d is the input gate, which controls the information of the input vector x_d and updates the current information flow C_d. f_d is the forget gate, which determines whether the value information is retained in C_d. o_d is the output gate, which includes the input information of level $d+1$ and determines the value of hidden state h_d of level d. σ and tanh are the activation functions. The operation update mechanism of each gate is as follows:

$$\begin{aligned} f_d &= \sigma\left(W_f \times [h_{d+1}, x_d] + b_f\right), \\ i_d &= \sigma(W_i \times [h_{d+1}, x_d] + b_i), \\ \widehat{C}_d &= \tanh\left(W_C \times [h_{d+1}, x_d] + b_C\right), \\ C_d &= f_d \times C_{d+1} + i_d \times \widehat{C}_d, \\ o_d &= \sigma(W_o \times [h_{d+1}, x_d] + b_o), \\ h_d &= o_d \times \tanh\left(C_d\right), \end{aligned} \tag{46}$$

where W_f, W_i, W_C, and W_o are the weight parameters of each gate obtained by network training and b_f, b_i, b_C, and b_o are the bias obtained by network training.

LSTMs composed of basic cells in series can accept any length of eigenvector as input and generate an output vector. If an LSTM can accept a sequence of vectors containing N vectors and generate an output vector of size K, we call its shape (N, K). After obtaining the unfolding tree, D kinds of LSTM cell are applied to the tree with depth D, and the number of each kind of LSTM cell is the same as the number of nodes in its depth. At its depth, each LSTM cell processes the relevant node information to obtain the output vector.

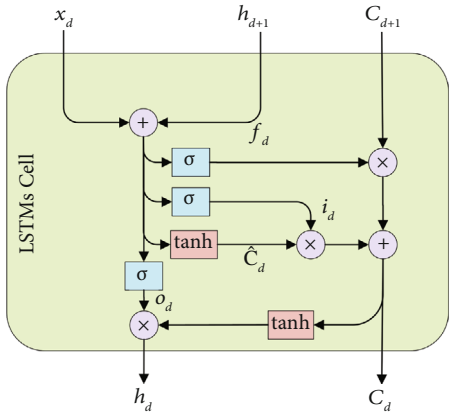

Figure 7: LSTMs cell.

For LSTMs with depth $d(0 < d < D)$, it is expressed as L_d and the shape is $(M + K_{d+1}, K_d)$. M indicates that there are M edges connected to the child nodes. K_{d+1} indicates the output from the lower level LSTMs, and $M + K_{d+1}$ indicates the series connection of the two eigenvectors. Taking Figure 5 as an example, the output $f(t_j)$ of the target t_j is calculated as follows:

$$\begin{aligned} f(t_j) &= f(s_a) \cap g(s_a, t_j) + f(s_b) \cap g(s_b, t_j) + f(s_c) \cap g(s_c, t_j), \\ f(s_a) &= g(s_a, t_k) + g(s_a, t_l), \\ f(s_b) &= g(s_b, t_k), \\ f(s_c) &= g(s_c, t_k) + g(s_c, t_l). \end{aligned} \tag{47}$$

Here, $f(\cdot)$ and $g(\cdot)$ are the input and output vectors of LSTMs, respectively. $g(s_i, t_j)$ is the state estimation label of target t_j obtained by radar s_i. The symbol $\cap$ represents the concatenation of feature vectors, and the symbol + represents the collection of information.

4.3.3. Parameter Update Process. L_d with the same depth have the same parameters, which are expressed as parameter vector $W^{(d)}$. When updating, nodes in the same level independently forward and backward propagation to minimize the loss function. In the training process, repeatedly select the same root node t_j^*, unfold graph $G_{t_j^*}$ into tree $T_{t_j^*}$, and carry out forward and backward propagation to update parameters.

Forward propagation is carried out from the lowest level child node to the root node. For node u with arbitrary depth d, if it has $d + 1$ level LSTMs, it obtains the output $f(u)$ from $d + 1$ level. The information that node u propagates to the parent node v is $g(v, u) \cap f(u)$.

In backward propagation, after obtaining the output eigenvector $y_{t_j} = f(t_j)$, the loss $\mathscr{L}(y_{t_j})$ is calculated through the loss function.

$$\mathscr{L}\left(y_{t_j}\right) = y_{t_j} - \bar{y}_{t_j}. \tag{48}$$

Here, $\bar{y}_{t_j}$ is the expected output eigenvector. In a training, after the loss is obtained, the derivative $\partial\mathscr{L}/\partial y_{t_j}$ of the loss to the output vector can be obtained. Then, the derivative of the loss to any input vector is obtained. When the input vector is like $g(v, u) \cap f(u)$, since $g(v, u)$ does not need to propagate, the backward propagation of loss only needs to calculate $\partial\mathscr{L}/\partial f(u)$. The derivative of the loss propagates from the root node to the lowest level child node to stop when the depth is D and the input vector is only $g(v, u)$.

Since there are many L_d at the same depth and have the same parameters, the L_d at the same depth need to calculate and update the parameters in parallel and then fuse the parameters for each L_d. Suppose that there are mL_d in one level, which are labeled as $L_d^{(1)}, L_d^{(2)}, \ldots, L_d^{(m)}$. If the updated parameter $\Delta_i W^{(d)}$ can be obtained for $L_d^{(i)}$, the updated parameter $\Delta W^{(d)}$ is calculated as follows:

$$\Delta W^{(d)} = \frac{\Delta_1 W^{(d)} + \Delta_2 W^{(d)} + \cdots + \Delta_m W^{(d)}}{m}. \tag{49}$$

4.3.4. Overall Network Structure and Output. Taking depth $D = 2$ as an example, the estimation fusion network is shown in Figure 8. Firstly, a bipartite graph is constructed according to the detection and tracking relationship between radar and target, and we unfold the graph into a tree with the tracked target as the root node. Secondly, the target state is labeled, and the labels of the estimation are obtained from the velocity and position space, respectively. Then, we input the estimation label into the trained MLSL. Finally, we input the network output vector into the Softmax full connection layer to obtain the probability of each label in the label space. The state label fusion $\hat{L}_{k+1,j}$ of the target is the label with the highest probability in the label space.

After obtaining the target label fusion $\hat{L}_{k+1,j}$, we need to convert the label fusion of the target into the state fusion. The transformation method is as follows:

$$x_{k+1|k+1,j} = \left(\hat{L}_{k+1,j} - L_{k+1|k,j}\right) \odot d_{\Delta L} + x_{k+1|k,j}. \tag{50}$$

5. Experiments and Results

5.1. Simulation Scene Setting. In order to evaluate the tracking performance of the algorithm proposed in this paper, we set up a simulation experiment scene in 2-dimensional space. In this simulation scenario, a total of 3000 simulation tracks are simulated. 30 targets form a cluster and move at the same time. Then, there are 100 clusters in total. Total time of cluster movement $T = 100$ s. 80% of the simulation data set is used for training and 20% for testing.

Set the cluster motion model parameters as shown in Table 1. 30 cluster members are randomly dropped in area $[0,100] \times [0,100]$ (m^2), and the initial velocity in each direction is a random number in the range of [0,140] m/s. After the cluster is dropped, the members adjust the actual velocity through self-organizing cooperative interaction. Due to

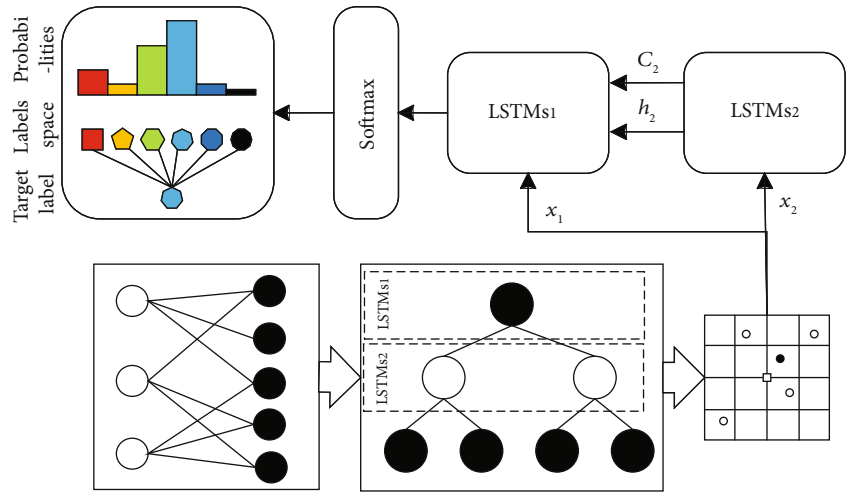

Figure 8: Overall network structure.

Table 1: Cluster motion model parameters.

Parameter	Value	Parameter	Value
d_b	20	d_m	50
α	0.1	β	0.2
σ_x	1	σ_y	1
σ_g	1	R_{11}	40
R_{12}	3	R_{21}	40
R_{22}	1		

the cooperative interaction, the actual velocity of the member is not $(\dot{x}, \dot{y})$ in the state vector. If the sampling interval is τ, the calculation method of actual velocity v is shown in (51). The movement process of a cluster is shown in Figure 9, and the actual velocity of each member is shown in Figure 10.

$$v = \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \frac{1}{\tau}\left(\begin{bmatrix} x_{t+\tau} \\ y_{t+\tau} \end{bmatrix} - \begin{bmatrix} x_t \\ y_t \end{bmatrix}\right). \tag{51}$$

As can be seen from Figures 9 and 10, after the cluster falls, the members begin to diffuse under the action of repulsive force due to the small spacing between each member. Affected by the cooperative interaction, the actual velocity of each member begins to converge. v_x converges to about 60 m/s, and v_y converges to about 75 m/s in about 30 s. After the velocity of cluster converges, the cluster forms a stable structure and moves at almost the same velocity. At this time, the dispersion range of the cluster is about $1600 \times 1600\ m^2$.

Figure 9 also shows that after the cluster forms a cooperative motion with almost the same velocity, the members in the cluster still have dynamic adjustment due to the action of motion noise and potential force. This adjustment action is very common, which would have a negative impact on the stable tracking of δ-GLMB filter.

5.2. δ-GLMB Filter Tracking Results. When the number of cluster members $N = 30$, given the minimum value of the average detection density $\bar{A}_d = 2.5$, select the number of radars $m = 10$ and set the maximum number of radar tracking targets $N_c = 8$. In this case, the average detection density $A_d \approx 2.7 > \bar{A}_d$. According to the above parameters, a detection confirmation matrix Ω is determined. The graphical representation of the matrix Ω is shown in Figure 11, in which 30 columns of the grid represent 30 targets and 10 rows represent 10 radars. Those marked with color in the grid are regarded as having detection relationship.

Set the initial state covariance of cluster targets $P_0 = \text{diag}\left(\left[1 \times 10^4, 1 \times 10^3, 1 \times 10^4, 1 \times 10^3\right]^T\right)$, the measurement noise covariance $R_k = \text{diag}\left(\left[4 \times 10^2, 4 \times 10^2\right]^T\right)$, and the potential force noise $S_k = \text{diag}\left(\left[5, 2.5, 2.5, 5\right]^T \times 10^{-4}\right)$. The clutter obeys the Poisson distribution of the expected value $\lambda_c = 8$ in time.

The δ-GLMB algorithm parameters are set as follows: target detection probability $p_D = 0.98$ and target survival probability $p_s = 0.99$. The newborn target model is determined according to the measurement set at time k, i.e., $\pi_\gamma = \{r_\gamma^{(i)}, p_\gamma^{(i)}\}_{i=1}^{|Z_k|}$, where the newborn probability $r_\gamma^{(i)} = 0.02$ and state distribution of newborn targets $p_\gamma^{(i)}(x) = \mathcal{N}(x; m_g^{(i)}, P_g)$. Here, $m_\gamma^{(i)} = [z_{k,i}, 0]^T$ and $P_\gamma = \text{diag}\left(\left[1 \times 10^4, 5 \times 10^3, 1 \times 10^4, 5 \times 10^3\right]^T\right)$.

In the process of tracking 100 clusters, δ-GLMB filter runs 1000 times in total. The main filtering conditions are shown in Figure 12, and the proportion of each condition is shown in Table 2.

As shown in Figure 12, under the set parameters, the δ-GLMB algorithm can track all 8 assigned cluster members with a proportion of 91.6%. Figure 12(a) shows that the filter can accurately track all targets in a short time and achieve stable tracking in the whole process of target movement. This condition is the most ideal, accounting for 57.6% of all major conditions. Figure 12(b) shows that the filter can

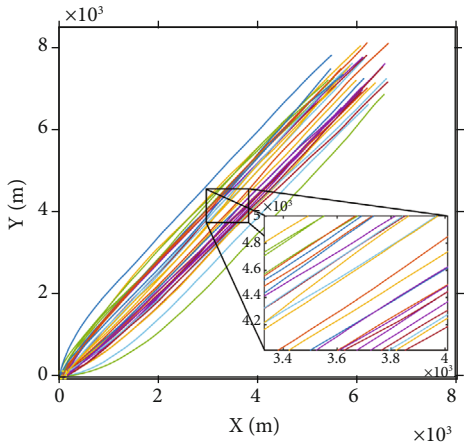

Figure 9: A cluster movement process.

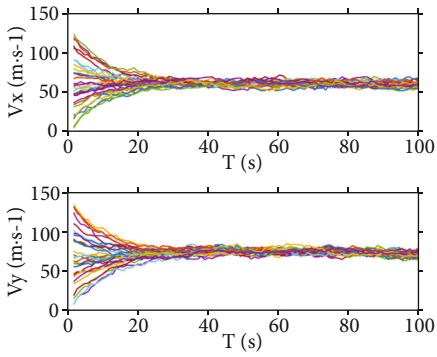

Figure 10: Actual velocity of each member.

achieve stable tracking of 8 targets after 20 s. This is due to the slow acquisition of the target by the filter because of the wide distribution of the target, which is unfavorable to the accurate tracking of the target in the early stage of motion. Figure 12(c) reflects that in the later stage of tracking, the target filtering results of some filters have the problems of inaccurate track association and filtering tracks crossing, which is related to the close spacing of the assigned targets. The tracking condition shown in Figure 12(d) is not ideal. The filter has been unable to track the 2 assigned targets, accounting for 7.1%. However, we need not worry too much about this, because other filters may obtain the motion estimation of these targets and make up for these defects in the fusion stage. Other tracking conditions include discontinuous tracks, multiple track initiation, and other problems, accounting for a small proportion, which is caused by the defects of the random finite set tracking algorithm itself.

Accumulate the number of targets tracked by 10 radars to obtain the total number of targets that can be tracked by the radar system, and calculate the average value of the total number of 100 experiments. The results are shown in Figure 13.

It can be seen from Figure 13 that the radar system can only track about 20 targets at the initial time when $A_d \approx$ 2.7. Then, the total number of targets tracked increases sharply to about 28. After 15 s, almost all targets can be tracked stably.

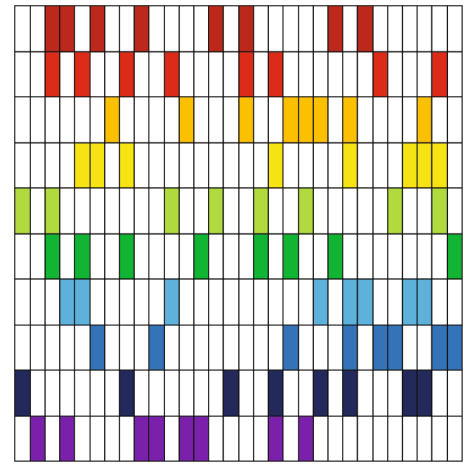

Figure 11: Graphical representation of matrix Ω.

5.3. Estimation Fusion Results Using Graph-LSTMs

5.3.1. Network Training Results. In order to verify the effectiveness of Graph-LSTMs for multiradar estimation fusion, set the label space length $d_{\Delta L_x} = d_{\Delta L_y} = 2$ m and $d_{\Delta L_{v_x}} = d_{\Delta L_{v_y}} = 0.5$ m/s. Set the number of labels $N_L = 11$ on each dimension, that is, $\{0, 1, 2, \cdots, 10\}$. The training set is used to train the network parameters, and the test set is used to test the fusion effect. The results are shown in Figures 14 and 15.

Figure 14 shows the label prediction accuracy of 1-MLSL with different training times. 1-MLSL indicates MLSL with depth $D = 1$. Figure 14(b) shows the mean and standard deviation of the probability of correct label output from Softmax layer. It can be seen from Figure 14(a) that the more training times, the better the network fusion effect. When the training times are less than 10000, the prediction accuracy of target labels increases rapidly with the increase of training times. When the number of training times is more than 10000, the prediction accuracy increases slowly, and the network parameter training is completed. Figure 14(b) shows that after the accuracy of label prediction increases to a certain level, the prediction probability of correct label can only increase to about 0.38. However, with the increase of training times, the standard deviation decreases, and the network prediction is more stable.

The curves shown in Figure 15 are the fitting result of the prediction accuracy of the target label under different depths and different training times. Figure 14 shows that when the training times are less, the network prediction effect with depth $D = 1$ is better. When the number of training times increases, the prediction effect of the network with depth $D = 1$ improves significantly and exceeds the effect of $D = 1$. The effect of depth $D = 1$ is always slightly worse than that of $D = 1, 2$. This shows that when the number of training is

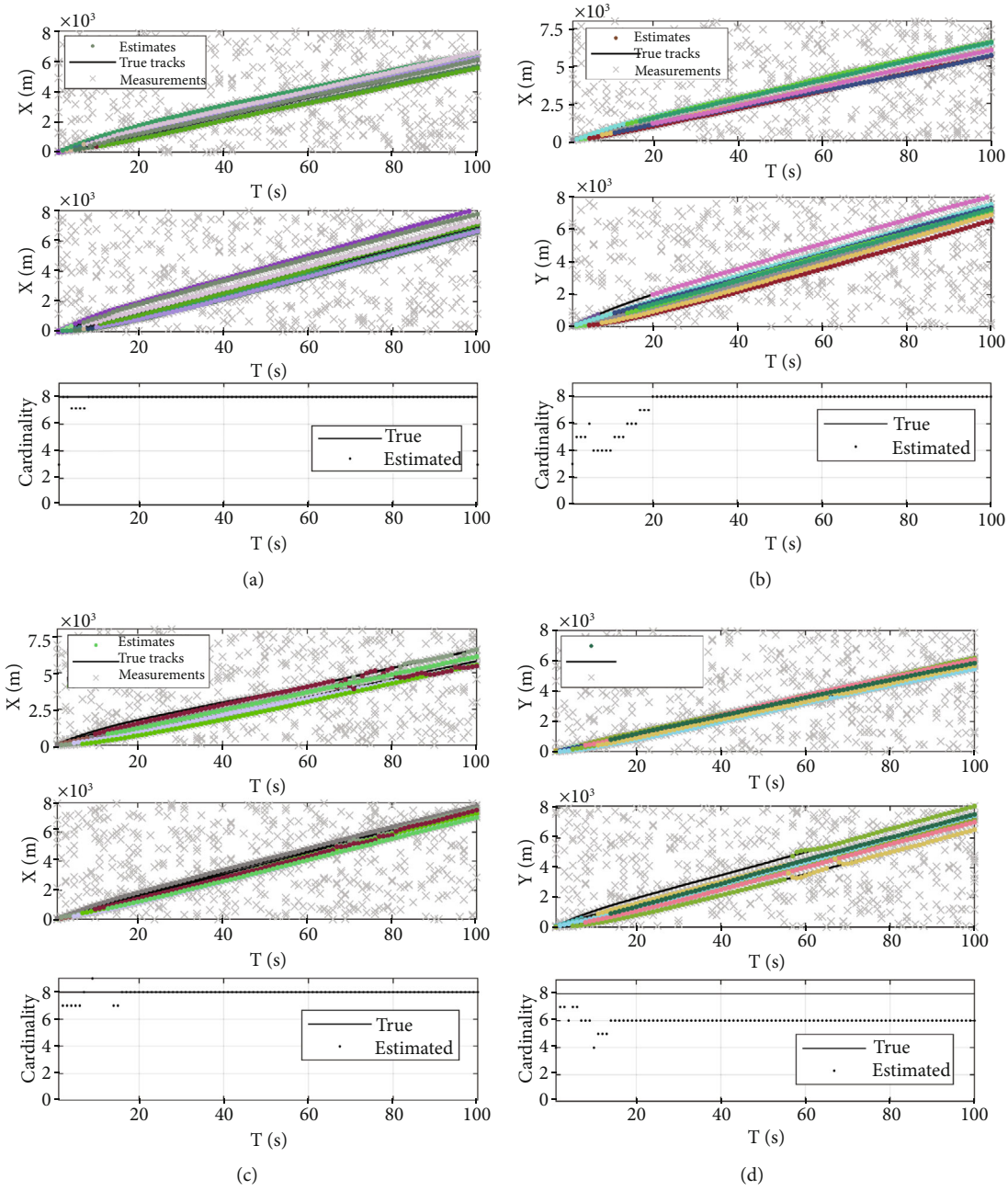

Figure 12: δ-GLMB filtering condition.

Table 2: Proportion of filtering condition.

Condition	Proportion	Condition	Proportion
Figure 12(a)	57.6%	Figure 12(b)	28.3%
Figure 12(c)	5.7%	Figure 12(d)	7.1%
Others	1.3%		

less, the simple network that only considers the detection and tracking relationship between radar and target can obtain better fusion effect, and with the increase of training times, the complex network that can mine the cooperative relationship between targets can play a better prediction effect. When the network depth $D = 3$, the relationship between radars is considered, which may have a better effect on mining deeper information, but it would take a lot of training time.

5.3.2. Estimation Fusion Effect. In this experiment, we compare the prediction probability of the target true label by using the Graph-LSTM algorithm proposed in this paper, the commonly used EM algorithm [40, 41] and the mean method, and give the corresponding OSPA distance in target tracking. The Graph-LSTM algorithm sets the depth $D = 2$. The OSPA distance is used to evaluate the tracking

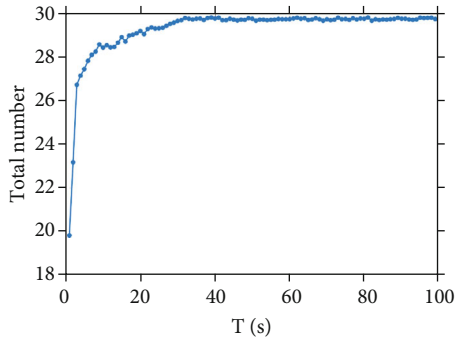

Figure 13: Total number of targets tracked.

performance of the algorithm in the experiment. The OSPA distance is defined as follows [42].

At time k, the true state set of cluster members is $X_k = \{x_{k,1}, x_{k,2}, \cdots, x_{k,m}\}$, and the target estimation state set given by the estimation algorithm is $X_{k|k} = \{x_{k|k,1}, x_{k|k,2}, \cdots, x_{k|k,n}\}$, where m and n represent the actual number of targets and the estimated number of targets in the cluster, respectively. $x_{k,i} \in X_k$ and $x_{k|k,i} \in X_{k|k}$ represent the actual value and estimated value of the target i state vector, respectively. The OSPA distance between the true state set and the estimated state set is calculated as follows:

$$\mathrm{OSPA}_{p,c}\left(X_k, X_{k|k}\right) = \begin{cases} \left[\frac{1}{n}\left(\min\limits_{\pi \in \Pi_n} \sum\limits_{i=1}^{m} \left(d_c\left(x_{k,i}, x_{k|k,\pi(i)}\right)\right)^p + (n-m)\cdot c^p\right)\right]^{1/p}, & m \le n, \\ \mathrm{OSPA}_{p,c}\left(X_{k|k}, X_k\right), & m > n, \end{cases} \tag{52}$$

where $\Pi_n = p_n^m$ represents all permutations of m elements in the state set.

$$d_c\left(x_{k,i}, x_{k|k,\pi(i)}\right) = \min\left(c, \left\|x_{k,i} - x_{k|k,\pi(i)}\right\|\right). \tag{53}$$

$\min\limits_{\pi \in \Pi_n}$ represents the group with the smallest distance difference between the true point trace and the estimated point trace of all targets. c and p are distance sensitivity parameters and correlation sensitivity parameters, respectively.

Set $c = 100$ and $p = 1$ in the experiment; the network fusion effect and OSPA distance are shown in Figure 16.

Figure 16 shows the prediction accuracy of target state label and the average OSPA distance of 20 test sets in tracking. It can be seen from Figure 16 that at the initial time, the prediction accuracy of target state label is low and the corresponding OSPA distance is large because the radar system is difficult to track all targets. As the filtering process proceeds, all targets are tracked and the OSPA distance decreases. At about 80 s, the OSPA distance increases due to correlation errors in some filters.

In the whole tracking process, the Graph-LSTM algorithm used in this paper has better prediction accuracy of correct labels than the comparison methods, which is related to the fact that the Graph-LSTM algorithm with depth $D = 2$ can consider the cooperative interaction between cluster targets. Accordingly, the method of using Graph-LSTM algorithm to calculate the fusion has less tracking error, and it can better alleviate the problem of inaccurate target tracking caused by correlation error in the later stage of tracking.

5.4. More Simulation Samples. In order to reflect the improving effect of our method on the accuracy of obtaining multiradar fusion estimates, we set more cluster motion trajectory data and obtain more fusion results of target estimators. There are four starting points of cluster motion in the sensor's field of view, namely, $\pm 8 \times 10^3, \pm 8 \times 10^3$, and the parameters of the cluster motion model are consistent with those above. The starting point and direction of cluster motion are random. We obtain 20 cluster motion trajectories. Cluster center trajectories are shown in Figure 17.

Six detection allocation schemes are randomly generated for each cluster, and a set of detection schemes is shown in Figure 18. We can obtain 120 sets of multiradar estimates. The proposed method, mean method, and EM algorithm are, respectively, used to fuse multiple estimates of the target. The average OSPA distance of the fusions is shown in Figure 19.

According to Figure 19, the average OSPA distance medians of the proposed method, EM algorithm, and mean method are 22.4, 27.8, and 34.3, respectively. Our method is superior to the mean method and EM algorithm in terms of fusion accuracy of estimates, which reflects the effectiveness and superiority of our method for multiple estimate fusion. However, the effect of the EM algorithm is more stable. This is because there are large differences between some test data and training data in the whole estimation process, and the Graph-LSTM algorithm parameters cannot be accurately obtained by learning, which makes our method not effective. In order to further improve the robustness of the proposed algorithm, the generalization ability of the algorithm can be improved by increasing the number of training samples or pretraining.

6. Discussion

6.1. Selection of $d_{\triangle L}$. In the estimation fusion method using Graph-LSTMs proposed in this paper, $d_{\Delta L}$ is the key parameter to determine the fusion effect. Compared with $d_{\Delta L_v}$, $d_{\Delta L_p}$ play a more major role. In order to compare the effects of different $d_{\Delta L_x}$ and $d_{\Delta L_y}$ on the estimation fusion, set $d_{\Delta L_x} = d_{\Delta L_y} = d_{\Delta L_p} = 1, 2, 5, 10$ m for experiments under the above experimental conditions. The experimental results are shown in Figure 20.

It can be seen from Figure 20 that different $d_{\Delta L_p}$ have different label prediction accuracy and the OSPA distance. Generally, the larger the $d_{\Delta L_p}$, the higher the label prediction accuracy. However, when $d_{\Delta L_p}$ is too large, the OSPA distance cannot be reduced accordingly. The main reason for the above phenomenon is that the larger the $d_{\Delta L_p}$, the larger the actual space occupied by each label. Radar estimate and target true state are more likely to have the same label. Therefore, the higher the label prediction accuracy can be

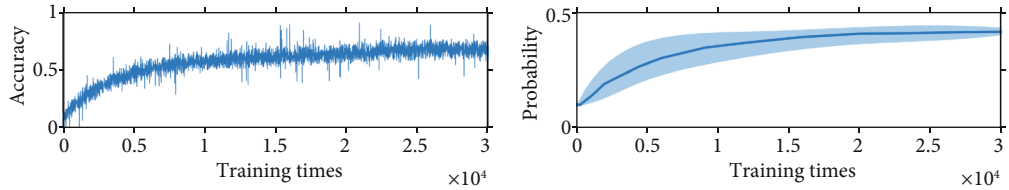

Figure 14: Training effect of 1-MLSL.

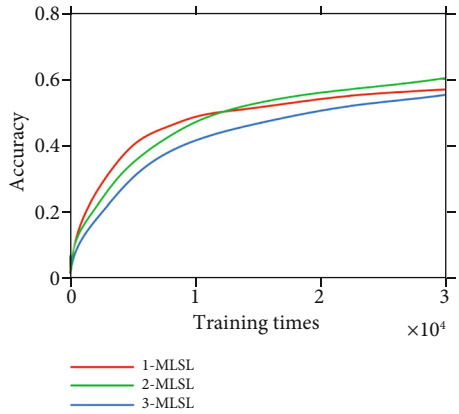

Figure 15: Training effect of different depth networks.

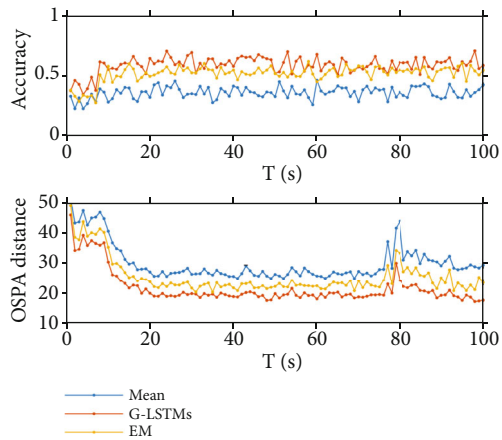

Figure 16: Network fusion effect and OSPA distance.

achieved. However, the essence of state space labeling is to use the state at the label to represent all the states in the state space represented by the label. According to (23), the fusion state is related to the actual state space occupied by each label. If it is larger, the OSPA distance between the fusion state and the true state of the target may be larger, and the performance of the algorithm would be reduced. For this experiment, when $d_{\Delta L_p} = 1$, the OSPA distance is large due to the low accuracy of label prediction, while when $d_{\Delta L_p} = 10$, the OSPA distance is large due to the excessive setting of $d_{\Delta L_p}$. $d_{\Delta L_p} = 2$ is a better choice that can balance the prediction accuracy and OSPA distance under the experimental conditions of this paper. The determination of $d_{\Delta L_p}$ is mainly related to the distance between the estimations of multiradar and the true state of target. In general, we can obtain some prior information to determine $d_{\Delta L_p}$ from the posterior probability of δ-GLMB filter.

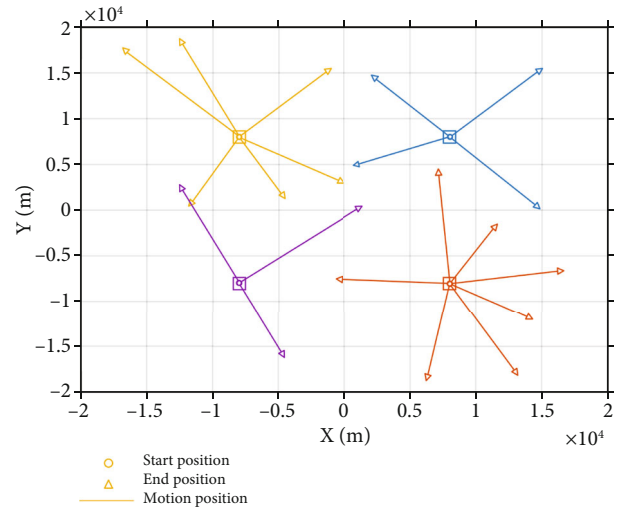

Figure 17: Cluster center trajectories.

6.2. Selection of A_d. When the number of radars is constant in multiradar system, the number of members in cluster and the number of targets tracked by single radar are both important factors affecting the fusion accuracy of cluster member estimation, and the average detection density A_d reflects the relative quantity relationship between them. In order to measure the influence of A_d, the number of targets estimated by a single radar is gradually increased under the experimental conditions of 10 radars and 30 cluster members. The OSPA distance of the fusion at different average detection densities is shown in Figure 21.

It can be seen from Figure 21 that at the initial stage, with the increase of average detection density, the fusion error of cluster target state estimation decreases gradually. OSPA = 17.9 is the minimum fusion error when $N_c = 11$. This is because the fusion becomes more accurate as the number of target estimates increases when the tracking load is lower than the maximum tracking capability of a single radar. However, with the further increase of N_c, the tracking load exceeds the maximum tracking capability of radar. The

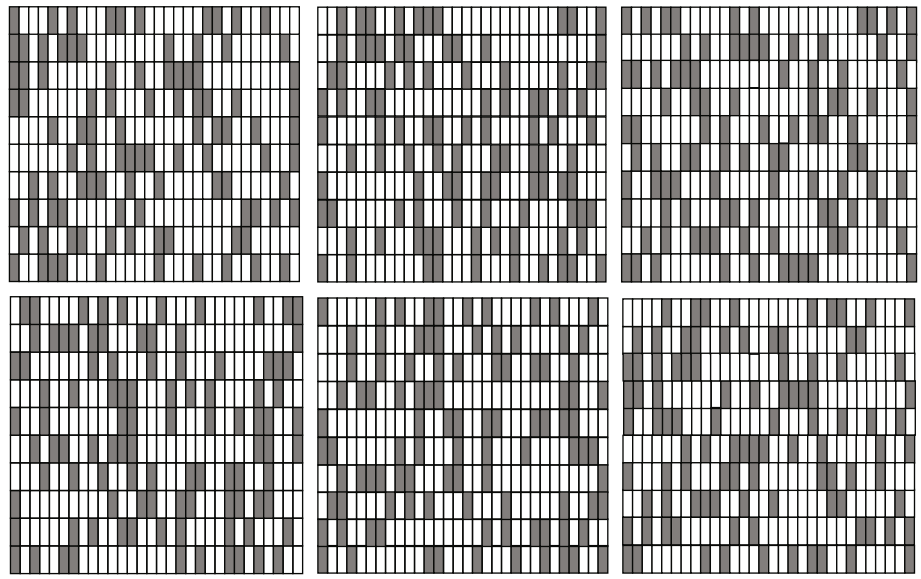

Figure 18: A set of cluster detection schemes.

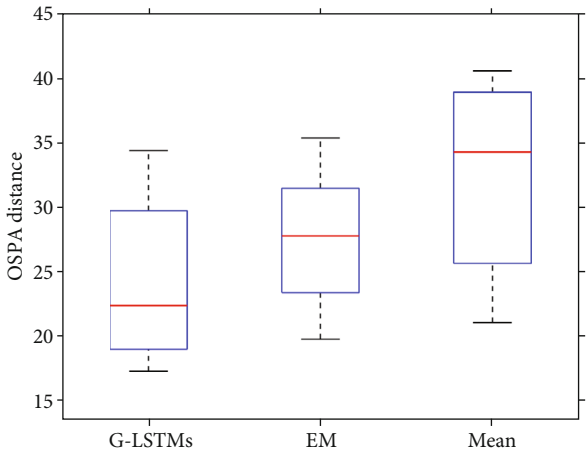

Figure 19: The average OSPA distance of the fusions.

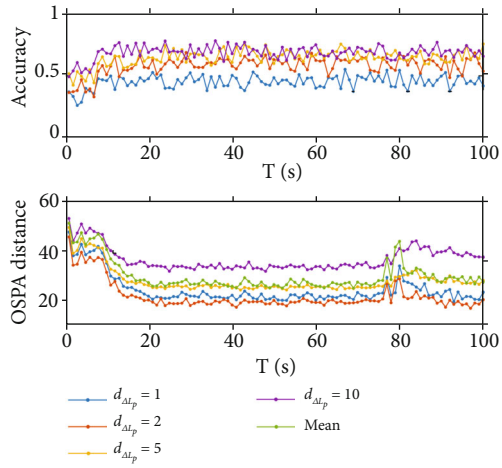

Figure 20: Tracking effect at different $d_{\triangle Lp}$.

inaccurate association increases first, and the estimation error starts to increase slowly. Subsequently, the proportion of missed detection gradually increases, and the actual correct tracking density does not increase with the detection density but decreases, and the fusion error of estimates increases rapidly.

6.3. Limitations and Improvements of Our Method. Our proposed radar-target assignment method preliminarily solves the problem of full coverage of cluster members, but the whole assignment process is given randomly without considering the optimal allocation scheme. In the actual tracking process, different members of the cluster have different values, and the members who are leaders should be tracked accurately first. At the same time, different radars also have different detection capabilities. The factors that affect the detection ability of radar include radar resolution and the position of the target in the beam. Higher-resolution radars should be used to detect higher-value targets. Therefore, we regard the problem of radar-target allocation as a nonlinear integer programming problem [43], and the mathematical model is as follows:

$$\text{Max} f = \sum_{t \in T} V_t \left(1 - \prod_{s \in S} (1 - p_{st})^{\omega_{st}}\right) \tag{54}$$

$$\text{s.t.} \sum_{t=1}^{N} \omega_{st} \le N_c, \quad \forall s \in S \tag{55}$$

$$\sum_{s=1}^{M} \omega_{st} \ge 1, \quad \forall t \in T \tag{56}$$

$$\sum_{t=1}^{N} \sum_{s=1}^{M} \omega_{st} \ge N \cdot A_d, \quad \forall s \in S, t \in T \tag{57}$$

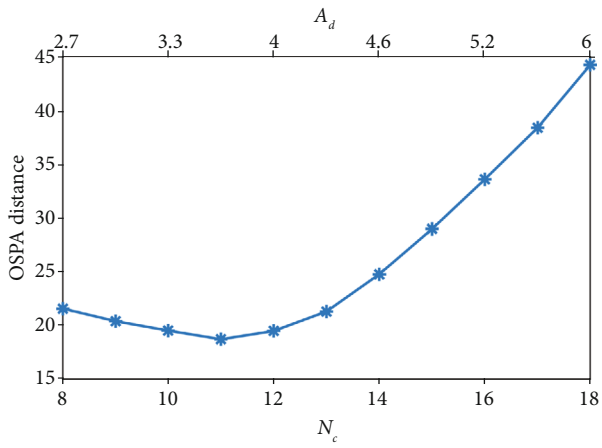

Figure 21: The OSPA distance of the fusion at different average detection densities.

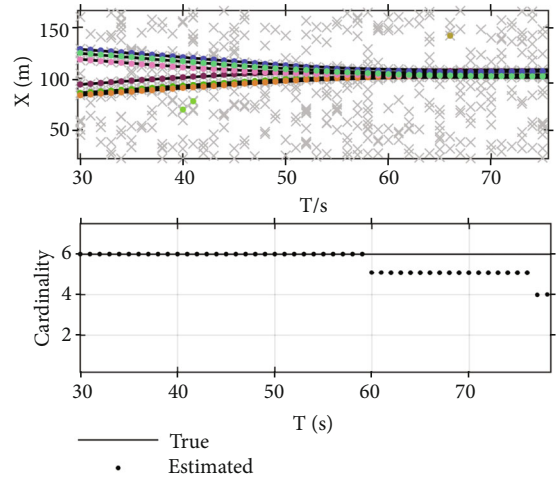

Figure 22: Tracking results during cluster merging.

$$\omega_{st} \in z_+, \quad \forall s \in S, t \in T, \tag{58}$$

where T is the cluster member set, S is the radar set, V_t represents the value of target t, and p_{st} represents the detection and tracking capability of radar s to target t. The definitions of other parameters are consistent with those in the paper. Equation (54) is the objective function to maximize the target value of the detection target cluster. Equations (55)–(58) are constraint conditions, which represent the full coverage of radar to members, the limit of the number of radar channels, and the reasonable detection density, respectively.

To solve equation (54), we can use traditional optimization algorithms, such as branch and bound method and cutplane method. We also can use intelligent optimization methods, such as genetic algorithm, particle swarm optimization algorithm, and ant colony algorithm. The optimization results of multiradar detection allocation can effectively support the generation of better tracking results.

Our algorithm has achieved good results in tracking a stable cluster target with a large number of members, but more advanced methods need to be proposed to deal with the possible splitting or merging of clusters. For large-scale clusters, splitting and merging are important moving characteristics that reflect the intelligence of clusters, which brings greater difficulty to the application of tracking algorithms. The state estimation results of the merging cluster using δ-GLMB filtering are shown in Figure 22.

Figure 22 shows the tracking results of δ-GLMB filter when two clusters are merging. It can be seen that the filter can still achieve effective tracking when the cluster is merging with a small maneuver. However, with cluster merging, the distance between the two cluster members decreases, and the filter loses track of one member at 60 s. When using multiradar to track clusters, the exchange of information between radars can make up for the problem of losing track, but it would still lead to the decline of effective tracking density and tracking accuracy.

The reason for the above phenomenon is that when the cluster merges, the original cooperative relationship is destroyed, and the newly established cooperative relationship leads to the failure of the target state equation describing the original motion. Similarly, this phenomenon occurs when a cluster is split. In order to solve the above problems, we should establish the cluster equation of motion in a multimodel way [44]. During tracking, the algorithm automatically detects whether cluster splitting or merging occurs and switches the motion model accordingly.

7. Conclusion

In this paper, we propose a method of using multiradar to track cluster targets. According to the characteristics of cluster target motion, we simulate the cooperative interaction behavior with multivariable stochastic differential equations and obtain the state space motion model of cluster members that can be used for iterative tracking algorithm. In the construction of the detection relationship between multiradar and cluster members, this paper proposes a radar-target assignment method. After a single radar obtains the assigned target, the target state is estimated by δ-GLMB filter. After obtaining the estimate, the label of the estimate can be obtained by using the state space labeling method proposed in this paper. On this basis, multiple estimation labels of the same target are input into the Graph-LSTMs applied to obtain the fusion label, and then, the joint estimate of target states by multiradar is solved. In the simulation experiment, we simulate the cluster motion and analyze the estimation of the filtering algorithm. Then, we compare the fusion effect of the Graph-LSTM algorithm applied in this paper with the EM algorithm and the mean method and discuss the influence of parameters $d_{\Delta L}$ and A_d on the fusion effect. The results show that multiradar can effectively track cluster targets and obtain more accurate estimation fusion in the method proposed in this paper. However, the radar-target assignment method proposed in this paper is not the optimal detection situation, and the method does not consider the impact of different value members on detection. In the discussion, we explore the improvement of the assignment method. In addition, the tracking algorithm adopted in this paper does not take into account the merging or splitting

of cluster targets, which would be the focus of our next research.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 61703424.

References

[1] J. W. Koch, "Bayesian approach to extended object and cluster tracking using random matrices," *IEEE Transactions on Aerospace Electronic Systems*, vol. 44, no. 3, pp. 1042–1059, 2008.

[2] J. Peng, W. Xu, B. Liang, and A. G. Wu, "Pose measurement and motion estimation of space non-cooperative targets based on laser radar and stereo-vision fusion," *IEEE Sensors Journal*, vol. 19, no. 8, pp. 3008–3019, 2008.

[3] Y. Zhu, S. Liang, M. Gong, and J. Yan, "Decomposed POMDP optimization based sensor management for multi-target tracking in passive multi-sensor systems," *IEEE Sensors Journal*, vol. 22, no. 4, pp. 3565–3578, 2022.

[4] C. Reynolds, "Flocks, herds and schools: a distributed behavioral model," in *Proceedings of the 14th annual conference on Computer graphics and interactive techniques - SIGGRAPH '87*, pp. 25–34, Anaheim, California, United States of America, 1987.

[5] T. Vicsek, A. Czirok, E. Ben-Jacob, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Physical Review Letters*, vol. 75, no. 6, pp. 1226–1229, 1995.

[6] I. Couzin, J. Krause, N. Franks, and S. A. Levin, "Effective leadership and decision-making in animal groups on the move," *Nature*, vol. 433, no. 7025, pp. 513–516, 2005.

[7] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[8] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Controlling formations of multiple mobile robots," in *Proc of IEEE International Conference on Robotics and Automation*, Leuven, Belgium, 1998.

[9] M. A. Lewis and K. H. Tan, "High precision formation control of mobile robots using virtual structures," *Autonomous Robots*, vol. 4, no. 4, pp. 387–5403, 1997.

[10] H. Qiu, H. Duan, and Y. Fan, "Multiple unmanned aerial vehicle autonomous formation based on the behavior mechanism in pigeon flocks," *Control Theory & Applications*, vol. 32, no. 10, pp. 1298–1304, 2015.

[11] G. Laman, "On graphs and rigidity of plane skeletal structures," *Journal of Engineering Mathematics*, vol. 4, no. 4, pp. 331–340, 1970.

[12] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 44, no. 9, pp. 1520–1533, 2004.

[13] B. Xin, Y. Wang, and J. Chen, "An efficient marginal-return-based constructive heuristic to solve the sensor-weapon-target assignment problem," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 12, pp. 2536–2547, 2019.

[14] S. Y. Wang, G. Wang, and J. R. Zhang, "Multi-sensor target assignment method based on variable weight artificial bee colony algorithm," *Basic & Clinical Pharmacology & Toxicology*, vol. 123, p. 92, 2018.

[15] L. Svensson, D. Svensson, M. Guerriero, and P. Willett, "Set JPDA filter for multitarget tracking," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4677–4691, 2011.

[16] X. Chen, Y. Li, J. Yu, and Y. Li, "Developing the fuzzy c-means clustering algorithm based on maximum entropy for multitarget tracking in a cluttered environment," *Journal of Applied Remote Sensing*, vol. 12, no. 1, p. 1, 2018.

[17] M. Beard, B. T. Vo, and B. N. Vo, "Bayesian multi-target tracking with merged measurements using labelled random finite sets," *IEEE Transactions on Signal Processing*, vol. 63, no. 6, pp. 1433–1447, 2015.

[18] K. Granstrom and U. Orguner, "A PHD filter for tracking multiple extended targets using random matrices," *IEEE Transaction on Signal Processing*, vol. 60, no. 11, pp. 5657–5671, 2012.

[19] K. Granström, P. Willett, and Y. Bar-Shalom, "PHD filter with approximate multi-object density measurement update," in *2015 18th International Conference on Information Fusion (Fusion)*, pp. 1802–1809, Washington, DC, 2015.

[20] R. P. S. Mahler, B. T. Vo, and B. N. Vo, "CPHD filtering with unknown clutter rate and detection profile," *IEEE Transaction on Signal Processing*, vol. 59, no. 8, pp. 3497–3513, 2011.

[21] D. S. Bryant, E. D. Delande, S. Gehly, J. Houssineau, D. E. Clark, and B. A. Jones, "The CPHD filter with target spawning," *IEEE Transactions on Signal Processing*, vol. 65, no. 5, pp. 13124–13138, 2016.

[22] B. T. Vo, B. N. Vo, and A. Cantoni, "The cardinality balanced multi-target multi-Bernoulli filter and its implementations," *IEEE Transaction on Signal Processing*, vol. 57, no. 2, pp. 409–423, 2009.

[23] J. Yin, J. Zhang, and J. Zhao, "The Gaussian particle multi-target multi-Bernoulli filter," in *2010 2nd International Conference on Advanced Computer Control*, Shenyang, China, 2010.

[24] B. T. Vo and B. N. Vo, "Labeled random finite sets and multi-object conjugate priors," *IEEE Transaction on Signal Processing*, vol. 61, no. 13, pp. 3460–3475, 2013.

[25] B. N. Vo, B. T. Vo, and D. Phung, "Labeled random finite sets and the Bayes multi-target tracking filter," *IEEE Transaction on Signal Processing*, vol. 62, no. 24, pp. 6554–6567, 2014.

[26] S. K. Pang, L. Jack, and S. J. Godsill, "Detection and tracking of coordinated groups," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 1, pp. 472–502, 2011.

[27] X. Xue, S. Huang, J. Xie, J. Ma, and N. Li, "Resolvable cluster target tracking based on the DBSCAN clustering algorithm and labeled RFS," *IEEE Access*, vol. 9, pp. 43364–43377, 2021.

[28] X. Chen, Y. Li, and J. Yu, "PHD and CPHD algorithms based on a novel detection probability applied in an active sonar tracking system," *Applied Sciences*, vol. 8, p. 36, 2018.

[29] S. J. Julier, "An empirical study into the use of Chernoff information for robust, distributed fusion of Gaussian mixture models," in *2006 9th International Conference on Information Fusion*, Florence, Italy, 2006.

[30] G. Battistelli, L. Chisci, C. Fantacci, A. Farina, and A. Graziano, "Consensus CPHD filter for distributed multitarget tracking," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 508–520, 2013.

[31] O. Sluciak, O. Hlinka, M. Rupp, F. Hlawatsch, and P. M. Djurić, "Sequential likelihood consensus and its application to distributed particle filtering with reduced communications and latency," in *Circuits, systems and computers, 1977. 2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR*, pp. 1766–1770, Pacific Grove, CA, USA, 2011.

[32] J. Wang, X. Su, L. Zhao, and J. Zhang, "Deep reinforcement learning for data association in cell tracking," *Frontiers in Bioengineering and Biotechnology*, vol. 8, p. 298, 2020.

[33] H. Liu, H. Zhang, and C. Mertz, "DeepDA: LSTM-based deep data association network for multi-targets tracking in clutter," in *2019 22th International Conference on Information Fusion (FUSION)*, pp. 1–8, Ottawa, ON, Canada, 2019.

[34] R. Agrawal, L. D. Alfaro, and V. Polychronopoulos, "Learning from graph neighborhoods using LSTMs," 2016, http://arxiv.org/abs/1611.06882.

[35] X. Shu, L. Zhang, Y. Sun, and J. Tang, "Host–parasite: graph LSTM-in-LSTM for group activity recognition," *IEEE Transactions On Neural Networks And Learning Systems*, vol. 32, no. 2, pp. 663–674, 2021.

[36] R. Verma, R. Rajesh, and M. S. Easwaran, "Modular multitarget tracking using long short-term memory networks," *Journal of Aerospace Information Systems*, vol. 18, no. 10, pp. 751–754, 2021.

[37] B. Oksendal, *Stochastic Differential Equations: An Introduction with Applications*, Springer Science & Business Media, 2013.

[38] C. Fantacci and P. Francesco, "Scalable multisensor multitarget tracking using the marginalized δ-GLMB density," *IEEE Signal Processing Letters*, vol. 23, no. 6, pp. 863–867, 2016.

[39] F. Lian, L. Hou, J. Liu, and C. Han, "Constrained multi-sensor control using a multi-target MSE bound and a δ-GLMB filter," *Sensors*, vol. 18, no. 7, p. 2308, 2018.

[40] P. A. Dempster, M. L. Nan, and B. R. Donald, "Maximum likelihood from incomplete data via theEMAlgorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[41] W. Huang and Y. Chen, "The multiset EM algorithm," *Statistics & Probability Letters*, vol. 126, pp. 41–48, 2017.

[42] D. Schuhmacher, B. T. Vo, and B. N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE Transactions on Signal Processing*, vol. 56, no. 8, pp. 3447–3457, 2008.

[43] A. Kline, D. Ahner, and R. Hill, "The weapon-target assignment problem," *Computers & Operations Research*, vol. 105, pp. 226–236, 2019.

[44] W. Yun, G. P. Hu, and H. Zhou, "GSR-TDMA: a geometric spatial reuse-time division multiple access MAC protocol for multihop underwater acoustic sensor networks," *Journal of Sensors*, vol. 2016, Article ID 6024610, 14 pages, 2016.