

**Enhancing Resolvable Group Target Tracking: Integration of
Labeled Multi-Bernoulli Filter with Deep Learning
Approaches**

Journal:	<i>IEEE Transactions on Aerospace and Electronic Systems</i>
Manuscript ID	TAES-2024-0014.R1
Manuscript Type:	Regular Paper
Date Submitted by the Author:	26-Apr-2024
Complete List of Authors:	Yu, Yue; Harbin Institute of Technology Mei, Liu; Harbin Institute of Technology Li, Bo; Haerbin Engineering University
Keywords:	Tracking filters, Graph theory, Neural network applications

SCHOLARONE™
Manuscripts

Index Terms— Resolvable group target tracking, graph neural networks, labeled multi-Bernoulli filter, neural ordinary differential equation.

Enhancing Resolvable Group Target Tracking: Integration of Labeled Multi-Bernoulli Filter with Deep Learning Approaches

Yue Yu

Harbin Institute of Technology, Harbin 150001, China

Mei Liu

Harbin Institute of Technology, Harbin 150001, China

Bo Li

Harbin Engineering University, Harbin 150001, China

Abstract—This paper introduces an innovative approach for resolvable group targets tracking (RGTT) through the integration of classical Bayesian filtering and sophisticated deep learning techniques. Numerous extant RGTT methods oversimplify target motion models and interactions within groups, limiting their applicability and compromising tracking performance. Deep learning methodologies are recurrently leveraged to overcome these challenges owing to their adeptness in delineating intricate mappings. We develop a neural network tailored for RGTT, incorporating neural ordinary differential equation (Neural ODE) to articulate the motion model of targets. Additionally, it predominantly integrates recurrent neural network (RNN) and graph neural network (GNN) modules. Similar to multi-target tracking tasks, the RNN module is enlisted here to model nonlinear dynamic systems. The distinction lies in the domain of resolvable group targets, where an additional reliance on graph theory becomes imperative for estimating group structures. Consequently, this paper introduces a GNN module into the deep learning network to learn the interaction relationships among targets within the same group. Furthermore, the seamless fusion of this deep network with the labeled multi-Bernoulli (LMB) filter augments the pragmatic viability of the deep learning algorithm. Simulation results elucidate the efficacy of the proposed algorithm.

Manuscript received XXXXX 00, 0000; revised XXXXX 00, 0000; accepted XXXXX 00, 0000.

Yue Yu is with the School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail:yuyue@stu.hit.edu.cn). Mei Liu is the professor and a Ph.D. supervisor of signal and information processing with the School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail:liumei@hit.edu.cn). Bo Li is with the Key laboratory of Intelligent Technology and Application of Marine Equipment, the Key laboratory of Ship Intelligent System and Technologies, the Key laboratory of Environment Intelligent Perception, the College of Intelligent System Science and Engineering, Harbin Engineering University, Harbin 150001, China (e-mail:libo520209@126.com).

0018-9251 © 2020 IEEE

real-world uncertainties. [15] presents a group structure model with VL using SDE, while [16] employs a similar strategy, focusing on the LF model in conjunction with the labeled multi-Bernoulli (LMB) filter. Nevertheless, the identification of leaders exhibiting both rationality and robustness presents a formidable challenge, markedly constraining the scope of applicability for such models. Moreover, the VL model neglects the intricacies and interaction among actual constituents within the group, conceivably culminating in an imprecise emulation of group behavior. Another strategy characterizes the interaction among targets in a group as repulsive and restorative forces. [1] derives an approximate repulsive force vector under the LMB filter and employs evolving networks [17] to handle dynamic changes in group topology. Building upon [1], [2] provides an exact state transition function (STF) for the standard MB filter. In the realm of target tracking, the absence of prior motion model parameters is a prevalent challenge. The social force model compounds this issue by introducing parameters pertaining to repulsion and state control, thereby intensifying the complexity of accurate modeling. Furthermore, addressing the dispersion and diffusion matrix of Brownian motion in SDE remains an inescapable challenge for these approaches.

In summary, the aforementioned methods, employing physical models, simplify the inherent dynamics of the target to a certain degree, and the model incorporates antecedent parameters formulated empirically. When compared to hand-crafted models, deep learning methods prove more adaptive and better suited for addressing complex real-world challenges. Existing learning-based approaches for vehicle trajectory prediction, which account for the interaction between adjacent vehicles, hold the potential for application in RGTT. For instance, the interactions between neighboring agents are encoded in pooling tensors through convolutional neural networks (CNN) [18]. Nevertheless, the lack of interpretability in deep networks introduces the possibility of producing physically infeasible outputs. Recent developments have seamlessly amalgamated physical models with deep learning methods [19] [20] [21] [22], offering a synthesis of the best attributes from both domains. Drawing inspiration from these methodologies and leveraging the potent capabilities of graph neural networks (GNN) to capture intricate interactions among graph nodes, the MTP-GO¹ is introduced [23]. In this approach, GNN is embedded within a recurrent neural network (RNN) to compute the input and process noise of the state transition equation. The state transition equation is learned through neural ordinary differential equation (neural ODE) [24] without any prior constraints.

To obtain the target trajectory and ensure tracking consistency, this paper explores the use of the LMB filter. The principal contributions of the paper are delineated as follows:

1) GNN learns the interactions among group members, while neural ODE handle the motion model. Different from the approach in [23], we present the incorporation of evolving graph network (EGN) [17] in this context. The continuous estimation of the group structure throughout the tracking process enhances the algorithm's adaptability to dynamic topology changes, obviating the necessity for a fully connected graph to mitigate redundancy.

2) Deep learning is integrated into the LMB filter, optimizing the conventional prediction steps for RGTT. The consideration of target birth, target survival, environmental clutter, and other factors is unified within the LMB filtering framework. Subsequently, we elucidate the implementation process of the Gaussian mixture (GM) for the entire approach.

The subsequent sections of the paper are arranged as follows. Section II introduces foundational knowledge, including the LMB filter and graph theory. Section III delineates the proposed amalgamation of the LMB filter with a deep learning algorithm, succeeded by simulation results in Section IV. Ultimately, Section V draws insightful conclusions.

II. Preliminary

A. The Standard LMB Filter

The Generalized Labeled Multi-Bernoulli (GLMB) filter [25], based on labeled RFS theory, is a groundbreaking closed-form solution for Bayesian filtering in multi-target tracking. Unlike conventional filters, it not only estimates target states but also provides their trajectories simultaneously. In this paper, we utilize the LMB filter as a specific instance of the GLMB filter, essentially serving as a first-moment approximation. The key distinction between the LMB and GLMB filters lies in their treatment of data association uncertainty. The LMB filter assumes that all targets belong to a single component (i.e., a single set of track labels), whereas the GLMB filter accommodates multiple hypotheses, each representing potential sets of target track labels. Experiments detailed in [25] demonstrate that while its performance is slightly lower than the GLMB filter, it effectively reduces computational load and conserves storage space. Furthermore, a noteworthy advantage of the LMB filter in this paper lies in its representation of trajectories in discrete form, thereby facilitating seamless integration with a neural network to predict the existence probability and spatial distribution of each independent LMB component. Following this, we offer a brief introduction to the LMB filter.

The notion of labeled RFS is initially introduced in [25], followed by the proposal of the standard LMB filter in [8]. A succinct overview of the LMB filter is presented herein, and detailed implementation specifics can be referenced in [8]. Throughout the paper, we adhere to the same notations as stipulated in [8], as delineated in Table I. For the sake of simplicity, the time subscript for

¹Multi-agent Trajectory Prediction by Graph-enhanced neural Ordinary differential equations

TABLE I
The Notations for LMB Filter

Notation	Explanation
x, z	Single-target state and observation.
X, Z	Multi-target state and observation.
$\mathbf{x}, \mathbf{X}/\pi$	Labeled state/distribution.
$\mathbb{X}/\mathbb{Z}/\mathbb{L}$	Space of state/observation/label.
$\mathcal{F}(\cdot)$	Collection of all finite subsets.
$h^X = \prod_{x \in X} h(x)$	Multi-target exponential notation, h is a real-valued function, X is a set, $h^\phi = 1$.
$\delta_Y(X) = \begin{cases} 1 & \text{if } X = Y \\ 0 & \text{otherwise} \end{cases}$	Kronecker Delta function.
$\langle f, g \rangle = \int f(x)g(x)dx$	Inner product.
$\mathcal{L}(X)$	$\mathcal{L} : \mathbb{X} \times \mathbb{L} \rightarrow \mathbb{L}$ be the projection
$\mathcal{L}((x, \ell)) = \ell$	\mathcal{L} is the inclusion function.
$1_Y(X) = \begin{cases} 1 & \text{if } X \subseteq Y \\ 0 & \text{otherwise} \end{cases}$	Distinct label indicator.
$\Delta(\mathbf{X}) = \delta_{ \mathbf{X} }(\mathcal{L}(\mathbf{X}))$	Set of track labels.
I	Predicted state e.g. $x_+ = x_{k+1 k}$.
$+$	Label space of the newborn targets and $\mathbb{L} \cap \mathbb{B} = \emptyset$.
\mathbb{B}	Predicted label space.
$\mathbb{L}_+ = \mathbb{L} \cup \mathbb{B}$	Survival probability.
$P_S^{(\ell)}(\cdot)$	Single-target Markov transition density.
$f^{(\ell)}(x \cdot)$	Space of mappings $\theta : I_+ \rightarrow \{0, 1, \dots, Z \}$, $\theta_\ell = 0$ means track ℓ is undetected.
Θ_{I_+}	Detection probability.
$P_D^{(\ell)}(\cdot)$	Single-target measurement likelihood.
$g(z x, \ell)$	Poisson clutter intensity.

the prediction item is denoted as $+$, and the time subscript for the update item is omitted.

Prediction: Suppose the current LMB density is entirely characterized by the parameter set $\pi = \{(r^{(\ell)}, p^{(\ell)})\}_{\ell \in \mathbb{L}}$, wherein $r^{(\ell)}$ and $p^{(\ell)}$ represent the existence probability and spatial distribution of a target with label ℓ , respectively. The birth process adheres to an LMB RFS and is defined by the parameter set $\pi_B = \{(r_B^{(\ell)}, p_B^{(\ell)})\}_{\ell \in \mathbb{B}}$ or equivalently by

$$\pi_B(\mathbf{X}) = \Delta(\mathbf{X})w_B(\mathcal{L}(\mathbf{X}))[p_B]^{\mathbf{X}}, \quad (1)$$

where,

$$w_B(L) = \prod_{i \in \mathbb{B}} \left(1 - r_B^{(i)}\right) \prod_{\ell \in L} \frac{1_{\mathbb{B}}(\ell)r_B^{(\ell)}}{1 - r_B^{(\ell)}}, \quad (2)$$

$$p_B(x, \ell) = p_B^{(\ell)}(x), \quad (3)$$

then, the predicted LMB density is denoted as $\pi_+ = \{(r_+^{(\ell)}, p_+^{(\ell)})\}_{\ell \in \mathbb{L}_+}$ with $\mathbb{L}_+ = \mathbb{L} \cup \mathbb{B}$,

$$\pi_+ = \{(r_{+,S}^{(\ell)}, p_{+,S}^{(\ell)})\}_{\ell \in \mathbb{L}} \cup \{(r_B^{(\ell)}, p_B^{(\ell)})\}_{\ell \in \mathbb{B}}, \quad (4)$$

where,

$$r_{+,S}^{(\ell)} = \eta_S^{(\ell)}r^{(\ell)}, \quad (5)$$

$$p_{+,S}^{(\ell)}(x) = \left\langle P_S^{(\ell)}f^{(\ell)}(x|\cdot), p^{(\ell)} \right\rangle / \eta_S^{(\ell)}, \quad (6)$$

$$\eta_S^{(\ell)} = \left\langle P_S^{(\ell)}, p^{(\ell)} \right\rangle. \quad (7)$$

Update: Given the measurement set designated as Z , the posterior of the LMB can be expressed as

$$\pi(\mathbf{X}|Z) = \left\{ \left(r^{(\ell)}, p^{(\ell)} \right) \right\}_{\ell \in \mathbb{L}_+}, \quad (8)$$

where,

$$r^{(\ell)} = \sum_{(I_+, \theta) \in \mathcal{F}(L_+) \times \Theta_{I_+}} w^{(I_+, \theta)}(Z)1_{I_+}(\ell), \quad (9)$$

$$p^{(\ell)}(x) = \frac{1}{r^{(\ell)}} \sum_{(I_+, \theta) \in \mathcal{F}(\mathbb{L}_+) \times \Theta_{I_+}} w^{(I_+, \theta)}(Z)1_{I_+}(\ell)p^{(\theta)}(x, \ell), \quad (10)$$

and,

$$w^{(I_+, \theta)}(Z) \propto w_+(I_+) \left[\eta_Z^{(\theta)} \right]^{I_+}, \quad (11)$$

$$w_+(I_+) = \prod_{i \in \mathbb{L}_+} \left(1 - r_+^{(i)}\right) \prod_{\ell \in I_+} \frac{1_{\mathbb{L}_+}(\ell)r_+^{(\ell)}}{1 - r_+^{(\ell)}}, \quad (12)$$

$$p^{(\theta)}(x, \ell|Z) = p_+^{(\ell)}(x)\psi_Z(x, \ell; \theta)/\eta_Z^{(\theta)}(\ell), \quad (13)$$

$$\psi_Z(x, \ell; \theta) = \begin{cases} 1 - P_D^{(\ell)}(x) & \text{if } \theta(\ell) = 0 \\ \frac{P_D^{(\ell)}(x)g(z_{\theta(\ell)}|x, \ell)}{\kappa(z_{\theta(\ell)})} & \text{if } \theta(\ell) > 0 \end{cases}, \quad (14)$$

$$\eta_Z^{(\theta)}(\ell) = \left\langle p_+^{(\ell)}, \psi_Z(\cdot, \ell; \theta) \right\rangle. \quad (15)$$

B. Graph Theory

The fundamental challenge in RGTT lies in comprehending the collaboration among group targets. In graph theory [26], the relationships between graph vertices are represented by edges, establishing a natural and intrinsic connection with RGTT. In this paper, we conceptualize the group of targets as an undirected graph, and we briefly revisit the pertinent background of graph theory in this subsection.

The graph structure can be denoted as $G = (V, E)$, where N targets constitute the set of vertices $V = \{v_1, v_2, \dots, v_N\}$. The state, label, and covariance information for target i are all encapsulated in node v_i . Thus, for any two nodes v_i and v_j , their Mahalanobis distance d_{ij} can be calculated. If d_{ij} falls beneath the designated threshold, it is posited that an edge connection subsists between v_i and v_j . The set E embodies the entirety of edges linking vertices in V . Furthermore, G can be expressed as $G = (g_1, \dots, g_m)$, where g_m is a subgraph of G , i.e. a connected partition in G . Fig. 1 illustrates a graphical representation of groups, demonstrating that the dependencies of v_6 and v_7 are conveyed through v_5 but are not directly connected. In other words, the graph structure in this paper is non-fully connected.

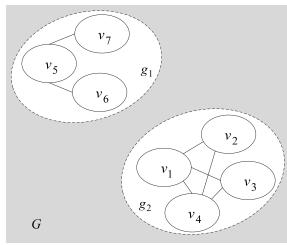


Fig. 1. Graphical representation of groups.

III. The Proposed GNN-LMB Filter

As is commonly understood, the conventional single-step nonlinear transformation with additive noise assumes the ensuing form:

$$\dot{x} = f(x, u) + w, \quad (16)$$

where $f(\cdot)$ represents a STF, u denotes an external input, and the process noise $w \sim \mathcal{N}(0, Q)$.

In RGTT, targets in close proximity with similar motion patterns interact, resulting in a more intricate STF, particularly evident in the prediction stage of the LMB filter. The paper's framework is illustrated in Fig. 2. incorporates a neural network module. This module draws insights from historical states $\pi_{1:k-1}$ and group structures $G_{1:k-1}$ to extract interactions among RGT, thereby learning and adapting the STF. This, in turn, predicts the LMB density $\{(r_{+,S}^{(\ell)}, p_{+,S}^{(\ell)})\}_{\ell \in \mathbb{L}}$ of surviving targets. Notably, the update of the group structure follows the LMB update. Subsections A and B introduce the group structure estimation module and the neural network module, respectively. The combination of these modules and the GM implementation are elaborated in subsection C.

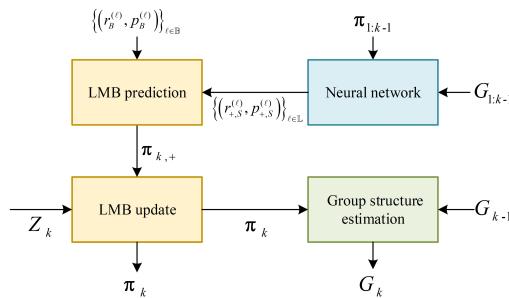


Fig. 2. The complete block diagram of the proposed filter.

A. Group Structure Estimation

In this paper, the distribution of targets is represented in GM form. To facilitate the estimation of group structure, the targets extracted at each time step are transformed into only one Gaussian component (GC) [27]. The mean and variance of GC are denoted as x_i and P_i , respectively. Employing graph theory to model group targets, the graph nodes v_i encompass the state x_i , variance P_i and label ℓ_i of RGT. For a detailed

algorithmic representation, refer to Appendix A for the complete pseudo-code of group structure estimation based on EGN [1]. Algorithm 1 is utilized for group structure initialization, and Algorithm 3 is applied for subsequent dynamic group management. To alleviate computational load, the center of the first-layer subgroup serves as the vertex to establish the second-layer group structure. The construction process is elaborated in Algorithm 2, as illustrated in Fig. 3. It is noteworthy that v_5 in Fig. 3, a singular node in the first-layer group structure, maintains its individual state in the second-layer group structure. Diverging from [1], the second-layer group also employs the group initialization algorithm instead of calculating the Mahalanobis distance between vertex pairs. This approach helps prevent ambiguity in the merging of multiple second-layer groups.

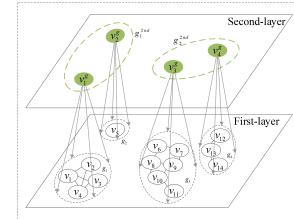


Fig. 3. Double-layered group structure.

B. Neural Network Module

The predicted LMB density of survival targets is denoted as $\{(r_{+,S}^{(\ell)}, p_{+,S}^{(\ell)})\}_{\ell \in \mathbb{L}}$. Since $p_{+,S}^{(\ell)}$ predominantly depends on the STF $f^{(\ell)}(x|\cdot)$, the primary challenge in this subsection is acquiring $r_{+,S}^{(\ell)}$ and $f^{(\ell)}(x|\cdot)$. The core of prediction lies in extrapolating from existing historical information. Gated Recurrent Units (GRU) [28] have demonstrated robust proficiencies in executing such endeavors, yet their focus is solely on the individual node under consideration. Consequently, the linear mapping component of GRU is replaced with GNN [29], which aggregates information from neighboring nodes while simultaneously considering the node's own features. This transformation creates a joint encoder that caters to the requirements of RGTT. Subsequently, a decoder is created with a structure largely similar to the encoder. The output of the decoder encapsulates the probability of existence, denoted as r , within the LMB density. It also incorporates the input term u and the covariance matrix Q for process noise in the state transition equation. Finally, the STF of the target is characterized by the neural ODE. The comprehensive model of this network is illustrated in Fig. 4, and the key modules are sequentially introduced below.

The components necessary for the GNN are defined as follows. Firstly, the graph structure, denoted as G , is obtained following the description in subsection A. Secondly, the node feature, in this context, is represented by the state of each RGT, encompassing position and velocity, as illustrated by q in Fig. 4. Lastly, the edge feature

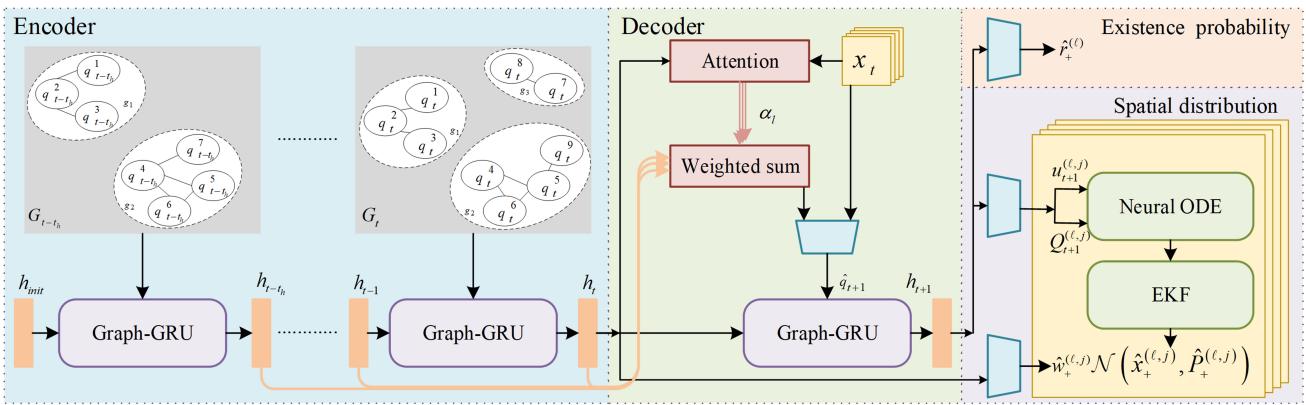


Fig. 4. The comprehensive flowchart of the neural network module. The symbol with subscript (+) in the figure denotes a general form. When the module is employed to solve the LMB density of the survival target, the subscript will be specified as (+, S).

is represented by the Mahalanobis distance between two mutually connected nodes.

(1) Encoding

For node v , Graph-GRU [23] is introduced to collectively extract features from the historical sequences of v and its neighboring nodes. The essence of Graph-GRU can be formulated as follows:

$$[\kappa_{r,i}^\nu \parallel \kappa_{z,i}^\nu \parallel \kappa_{h,i}^\nu] = GNN_q \left(q_i^\nu, \{q_\tau^\tau\}_{\tau \neq \nu} \right), \quad (17)$$

$$[\xi_{r,i}^\nu \parallel \xi_{z,i}^\nu \parallel \xi_{h,i}^\nu] = GNN_h \left(h_{i-1}^\nu, \{h_{\tau-1}^\tau\}_{\tau \neq \nu} \right), \quad (18)$$

where \parallel denotes the concatenation operation. q_i^ν represents the current input variable, and h_{i-1}^ν is the hidden variable from the previous moment. In the conventional GRU, a total of six intermediate vectors in (17) and (18) are obtained through the linear mapping of q_i and h_{i-1} , respectively. In this context, GNN is employed to replace the linear mapping, enabling the aggregation of features from neighboring nodes τ .

Following this, the hidden variable at the current moment functions as the output for this moment, calculated using the following formulas:

$$r_i^\nu = \sigma (\kappa_{r,i}^\nu + \xi_{r,i}^\nu + b_r), \quad (19)$$

$$z_i^\nu = \sigma (\kappa_{z,i}^\nu + \xi_{z,i}^\nu + b_z), \quad (20)$$

$$\tilde{h}_i^\nu = \phi (\kappa_{h,i}^\nu + r_i^\nu \odot \xi_{h,i}^\nu + b_h), \quad (21)$$

$$h_i^\nu = (1 - z_i^\nu) \odot \tilde{h}_i^\nu + z_i^\nu \odot h_{i-1}^\nu, \quad (22)$$

in these equations, the vectors b_r , b_z , and b_h correspond to additional bias terms. The symbol \odot signifies the Hadamard product, σ denotes the sigmoid function, and ϕ represents the hyperbolic tangent.

Presently, an array of GNNs has been developed extensively [30], finding applications across diverse domains and yielding commendable outcomes. Examples encompass GraphConv, graph convolutional network (GCN),

graph attention network (GAT), and more. It has been evidenced that augmenting the central node v with additional characterization enhances the efficacy of GAT+ [23], surpassing the base GAT in training. Consequently, this paper opts for GAT+ as the GNN module in (17) and (18), with the node representation q updating as follows:

$$q^\nu = b + W_1 q^\nu + \sum_{\tau \in \tilde{N}(\nu)} \tilde{\alpha}_{\nu,\tau} W_2 q^\tau, \quad (23)$$

where $\tilde{N}(\nu)$ is the inclusive neighborhood of a node v , $\tilde{\alpha}_{\nu,\tau}$ denotes the attention weights calculated as [31]. W_1 , W_2 , and b are learnable parameters.

In [23], when historical information from $t - t_h$ to t is given, node v and its neighbors at time t , $N_t(v)$, are chosen as the subject of investigation. The characteristics $\{\{q_i^\nu\}_{i=t-t_h}^t\}_{\nu \in \tilde{N}_t(\nu)}$ of the node set $\tilde{N}_t(\nu) = N_t(\nu) \cup \{v_t\}$ are subsequently employed as input for the network. However, in scenarios where the target is undetected, deceased, or has separated from the group structure depicted in Fig. 5, prediction accuracy may be compromised due to the lack of neighboring nodes. For instance, in scrutinizing target 1 as the subject of investigation, merely target 3 is identifiable as its neighboring target at time t . Nevertheless, in actuality, both target 2 and target 4 are concomitant members of the same group, exerting influence on the state of target 1 during the time interval spanning $t - t_h$ to t . This paper addresses this issue by introducing group structure estimation. By determining the subgroup to which node v belongs at each moment from $t - t_h$ to t , all identified neighboring nodes $\{\tilde{N}_i(\nu)\}_{i=t-h}^t$ can be considered. Moreover, the outcomes of group structure estimation at each instance can be directly utilized as input to the GAT+, as opposed to depending on the fully connected graph as illustrated in [23]. This strategy markedly alleviates the computational burden.

(2) Decoding

To obtain the LMB density $\{(r_{+,S}^{(t)}, p_{+,S}^{(t)})\}_{t \in \mathbb{L}}$ for the surviving target, Task 1 of the decoder produces the

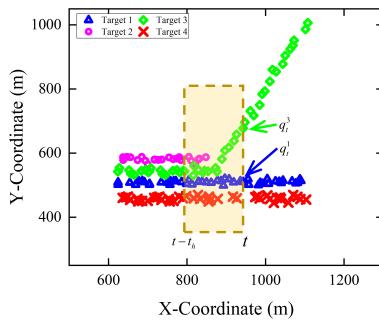


Fig. 5. An example showcasing the incomplete consideration of neighboring targets in the original method in [23] emphasizes the importance of the group structure estimation algorithm introduced in this paper.

survival probability of the target, denoted as $r_{+,S}^{(\ell)}$. For the computation of $p_{+,S}^{(\ell)}$, Task 2 of the decoder generates the 2-dimensional input term u and the noise covariance matrix Q in the target's motion model. The decoder's structure mirrors that of the encoder, and the hidden variable h can be directly inherited from the encoder. It's important to note that the input variable q requires additional computation using the attention mechanism [23].

In this paper, the application of the sliding window strategy is employed to predict the target's state at time $t+1$ leveraging historical information from the preceding t_h temporal intervals. The utilization of the teacher-forcing strategy is embraced in this context. During the training phase, the actual group structure at time $t+1$ is applied for decoding. However, in the filtering phase, the updated group structure at time t serves as the input to the decoder.

(3) Kinematics model

At present, prevailing motion models in RGTT tend to oversimplify the inherent dynamics of the target, relying on specific assumptions. Nonetheless, the motion model of a target can exhibit a myriad of patterns, encompassing unpredictable maneuvers. In this context, the application of the neural ODE [24] is embraced to learn the target's motion model unburdened by constraints, thereby augmenting the adaptability and versatility of the model within the authentic environment. An ODE can be formally expressed as $\frac{dx(t)}{dt} = f(x(t), \theta)$. The pivotal objective of the neural ODE lies in determining the optimal θ satisfying the above formula.

In this study, a second-order model is employed to ensure the smoothness of the estimation [32], with the formulation presented as follows:

$$\begin{aligned}\dot{p}_x &= v_x \\ \dot{p}_y &= v_y \\ \dot{v}_x &= f_1(v_x, v_y, u_1) \\ \dot{v}_y &= f_2(v_x, v_y, u_2).\end{aligned}\quad (24)$$

Hence, the STF can be effectively represented by a neural ODE. Through its integration with the output u and Q from the decoder, the comprehensive kinematics model described by (16) is achieved for prediction.

(4) Loss function

Illustrated in Fig. 4, the network output endeavors to align with a GM model that characterizes the spatial distribution of the target. This model produces multimodal probabilistic predictions, harmonizing seamlessly with the GM-based LMB filter. Assuming each output consists of M GCs, the result for the target ℓ is articulated as:

$$y_v = \left\{ \hat{w}_+^{(\ell,j)}, \mathcal{N} \left(\hat{x}_+^{(\ell,j)}, \hat{P}_+^{(\ell,j)} \right) \right\}_{j=1}^M, \quad (25)$$

where, $\hat{w}_+^{(\ell,j)}$ signifies the predicted weight of the j -th GC, and $\mathcal{N} \left(\hat{x}_+^{(\ell,j)}, \hat{P}_+^{(\ell,j)} \right)$ represents the Gaussian distribution denoted by the predicted mean and covariance. Consequently, the state prediction loss for target ℓ can be formulated using the negative log-likelihood (NLL) loss.

$$\mathcal{L}_{\text{NLL}}^\ell = -\log \left(\sum_j^M \hat{w}_+^{(\ell,j)} \mathcal{N} \left(x^\ell | \hat{x}_+^{(\ell,j)}, \hat{P}_+^{(\ell,j)} \right) \right), \quad (26)$$

where x^ℓ is the true state of target ℓ .

The neural network also produces the predicted existence probability, denoted as $\hat{r}_+^{(\ell)}$, for the survival target. Predicting $\hat{r}_+^{(\ell)}$ is approached as a binary classification task, and the binary cross-entropy (BCE) loss is employed to compute the existence probability prediction loss for target ℓ :

$$\mathcal{L}_{\text{BCE}}^\ell = -\delta(\ell) \log(\hat{r}_+^{(\ell)}) - (1 - \delta(\ell)) \log(1 - \hat{r}_+^{(\ell)}), \quad (27)$$

where $\delta(\ell)$ is an indicator of the existence of target ℓ , taking the value '1' if the target ℓ exists and '0' otherwise.

The overall regression loss and classification loss are presented as the sum of each target's NLL loss and BCE loss, respectively. The ultimate loss is derived by the weighted combination of these two losses:

$$\mathcal{L} = \alpha \sum_{\ell \in \mathbb{L}} \mathcal{L}_{\text{NLL}}^\ell + (1 - \alpha) \sum_{\ell \in \mathbb{L}} \mathcal{L}_{\text{BCE}}^\ell, \quad (28)$$

where \mathbb{L} is the label space of survival targets, and $0 < \alpha < 1$ is the loss weight.

C. GM Implementation of the GNN-LMB Filter

In this subsection, the complete GM implementation of the proposed approach is presented. This approach combines the temporal GNN and the LMB filter, referred to as the GNN-LMB filter. Considering that the extended kalman filter (EKF) exhibits the capability to handle nonlinear problems compared to the kalman filter (KF), and its calculations and implementation are relatively straightforward compared to the unscented kalman filter (UKF), this paper chooses to implement it based on EKF [33]. The posterior probability densities $p^{(\ell)}$ of all labeled Bernoulli tracks $\ell \in \mathbb{L}$ are delineated as follows:

$$p^{(\ell)}(x) = \sum_{j=1}^{J^{(\ell)}} w^{(\ell,j)} \mathcal{N} \left(x; \hat{x}^{(\ell,j)}, \hat{P}^{(\ell,j)} \right), \quad (29)$$

where $J^{(\ell)}$ represents the number of GC for the target labeled ℓ , while $w_{+}^{(\ell,j)}$ denotes the weight of each GC. Additionally, $\hat{x}_{+}^{(\ell,j)}$ is the estimated mean value of each GC, and $\hat{P}_{+}^{(\ell,j)}$ represents the corresponding estimation error covariance. The subsequent expressions with $\mathcal{N}(\cdot)$ have a similar meaning.

Prediction: For the survival target, $(r_{+,S}^{(\ell)}, p_{+,S}^{(\ell)})$ are directly generated by the neural network, circumventing the need for computation as outlined in equations (5)-(7). The formulation of $p_{+,S}^{(\ell)}$ is as follows:

$$p_{+,S}^{(\ell)}(x) = \sum_{j=1}^{J_M} \hat{w}_{+,S}^{(\ell,j)} \mathcal{N}\left(x; \hat{x}_{+,S}^{(\ell,j)}, \hat{P}_{+,S}^{(\ell,j)}\right), \quad (30)$$

where J_M represents the number of GC. Analogous to the formulation for survival targets, denote the newborn LMB RFS as $\{(r_B^{(\ell)}, p_B^{(\ell)})\}_{\ell \in \mathbb{B}}$, and presume that $p_B^{(\ell)}$ comprises J_B GCs:

$$p_B^{(\ell)} = \sum_{j=1}^{J_B} w_B^{(\ell,j)} \mathcal{N}\left(x; \hat{x}_B^{(\ell,j)}, \hat{P}_B^{(\ell,j)}\right). \quad (31)$$

As indicated in (4), the predicted LMB density π_+ is derived by amalgamating the information pertaining to survival and newborn targets.

Update: The aggregate of weights attributed to each GC corresponds to the probability of existence, denoted as $r^{(\ell)}$. Thus, the emphasis lies in resolving the updated spatial distribution $p^{(\theta)}(x, \ell | Z)$. Assuming a consistent detection probability P_D , irrespective of the position x , the predicted spatial distribution of track ℓ is uniformly articulated as follows, , regardless of whether it's a surviving target or a newborn one:

$$p_+^{(\ell)}(x) = \sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \mathcal{N}\left(x; \hat{x}_+^{(\ell,j)}, \hat{P}_+^{(\ell,j)}\right). \quad (32)$$

where $J_+^{(\ell)}$ represents the number of GC. Establish the nonlinear measurement equation as follows:

$$z = h(x) + v. \quad (33)$$

Define H as the linearized matrix of the measurement function $h(\cdot)$, and denote the measurement noise v as $v \sim \mathcal{N}(0, R)$.

When $\theta(\ell) \neq 0$, associate the measurement $z_{\theta(\ell)}$ with track ℓ , and the GM implementation is as follows:

$$p^{(\theta)}(x, \ell | Z) = \sum_{j=1}^{J_+^{(\ell)}} w^{(\ell,j,\theta)}(Z) \mathcal{N}\left(x; \hat{x}^{(\ell,j,\theta)}, \hat{P}^{(\ell,j)}\right), \quad (34)$$

$$w^{(\ell,j,\theta)}(Z) = \frac{\frac{1}{\kappa(z_{\theta(\ell)})} P_D w_+^{(\ell,j)} \mathcal{N}\left(z_{\theta(\ell)}; z_+^{(\ell,j)}, S^{(\ell,j)}\right)}{\eta_Z^{(\theta)}(\ell)}, \quad (35)$$

$$\eta_Z^{(\theta)}(\ell) = \frac{P_D}{\kappa(z_{\theta(\ell)})} \sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \mathcal{N}\left(z_{\theta(\ell)}; z_+^{(\ell,j)}, S^{(\ell,j)}\right), \quad (36)$$

$$S^{(\ell,j)} = H \hat{P}_+^{(\ell,j)} H^T + R, \quad (37)$$

$$z_+^{(\ell,j)} = H \hat{x}_+^{(\ell,j)}, \quad (38)$$

$$\hat{x}^{(\ell,j,\theta)}(Z) = \hat{x}_+^{(\ell,j)} + K^{(\ell,j)} \left(z_{\theta(\ell)} - z_+^{(\ell,j)} \right), \quad (39)$$

$$K^{(\ell,j)} = \hat{P}_+^{(\ell,j)} H^T \left[S^{(\ell,j)} \right]^{-1}, \quad (40)$$

$$\hat{P}^{(\ell,j)} = \hat{P}_+^{(\ell,j)} - K^{(\ell,j)} S^{(\ell,j)} \left[K^{(\ell,j)} \right]^T. \quad (41)$$

When $\theta(\ell) = 0$, no measurements are associated with the trajectory ℓ , and the predicted spatial distribution is utilized instead of the posterior distribution. In this case,

$$\eta_Z^{(\theta)}(\ell) = q_D = 1 - P_D. \quad (42)$$

IV. Numerical Simulation

A. Simulation Scenario and Comparison Approaches

In this subsection, we conduct a simulation experiment to assess the tracking performance of the proposed GNN-LMB filter. The evaluation criterion utilized is the optimal sub-pattern assignment (OSPA) distance [34] and the OSPA⁽²⁾ distance [35] (with window $L = 5$), employing an order parameter of $p = 1$ and a cut-off parameter of $c = 100$. The experimental scenario is set in two dimensions within the region $[-1500, 1500] \text{ m} \times [0, 3000] \text{ m}$. The sensor is positioned at the coordinates (0m,0m) and surveils the entire area with a data sampling frequency of 1 Hz. The monitoring time is 100s.

For all LMB filters expounded in this paper, a standard set of parameters is shared. The birth process is characterized by an LMB RFS with the parameter ensemble $\pi_B = \{(r_B^{(\ell)}, p_B^{(\ell)})\}_{\ell \in \mathbb{L}}$, where $r_B^{(\ell)} = 0.1$ and $p_B^{(\ell)} = \mathcal{N}(x; x_B^{(\ell)}, P_B)$, with $P_B = \text{diag}([50^2, 30^2, 50^2, 30^2]^\top)$, and $x_B^{(\ell)} = [p_x^{(\ell)}, v_x^{(\ell)}, p_y^{(\ell)}, v_y^{(\ell)}]$ configured in accordance with the initial state of the target in distinct scenarios. Furthermore, the nonlinear measurement model is employed and expressed as:

$$z = \begin{bmatrix} \text{atan2}(p_x - p_x^s, p_y - p_y^s) \\ \sqrt{(p_x - p_x^s)^2 + (p_y - p_y^s)^2} \end{bmatrix} + \epsilon, \quad (43)$$

where atan2 designates the 4-quadrant inverse tangent function, (p_x^s, p_y^s) represents the known sensor position, and the covariance of the measurement noise is

$$R = \text{diag}([\sigma_\theta^2, \sigma_r^2]^\top), \quad (44)$$

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

TABLE II
Simulation Parameters and Their Symbols and Values

Simulation Parameter	Symbol	Value
Detection probability	P_D	0.90
Clutter rate	λ_c	20
Sensor resolution in range	a_r	15m
Sensor resolution in azimuth	a_φ	2°
Sensor signal-to-noise ratio	SNR	10dB
False alarm probability	P_{FA}	10^{-3}
Maximum number of GCs	N_{GC}	15
Merging threshold	ξ_m	4
Pruning threshold	ξ_p	10^{-4}
Extraction threshold	ξ_e	0.5

TABLE III

The Parameters Governing the Kinematics of the Target within the Dataset

Parameters	Interval range
The acceleration of CA model	$-0.5 \sim 0.5 \text{ m/s}^2$
The turn rate of CT model	$-4 \sim 4^\circ/\text{s}$
The matrix coefficient σ^2 of (45)	$1 \sim 2 \text{ m/s}^2$
The birth time of target	$1 \sim 50\text{s}$
The death time of target	$60 \sim 100\text{s}$

with $\sigma_\theta = (\pi/180) \text{ rad}$ and $\sigma_r = 5 \text{ m}$. The remaining parameters involved in the simulation are presented in Table II.

For the GNN module, the maximum duration of the observation window, corresponding to the model input, is stipulated as 15s. The aim is to predict the state of the surviving target at a subsequent moment, indicating an output duration of 1s. Three distinct motion models govern the system: Constant Velocity (CV), Constant Acceleration (CA), and Constant Turn Rate (CT). The stochastic interference in the state transition process is modeled as Gaussian white noise, denoted as $w \sim \mathcal{N}(0, Q)$, with the covariance matrix given by:

$$Q = \sigma^2 \begin{bmatrix} \frac{T^4}{2} & \frac{T^3}{2} & 0 & 0 \\ \frac{T^3}{2} & T^2 & 0 & 0 \\ 0 & 0 & \frac{T^4}{2} & \frac{T^3}{2} \\ 0 & 0 & \frac{T^3}{2} & T^2 \end{bmatrix}, \quad (45)$$

where $T = 1\text{s}$ denotes the time interval. The parameters governing the motion models and the birth/death times of each target follow a uniform distribution, as detailed in Table III. The initial position of a new target manifests randomly within the detection range, and the initial velocity ranges between -40m/s and 40m/s . The count of transitions between motion modes throughout the target's entire existence adheres to a Poisson distribution with parameter $\lambda = 3$. The occurrences of mode transitions transpire randomly during the survival period. Following a transition, one of CV, CA, or CT is randomly selected. Following the aforementioned procedure, diverse RGT trajectories are generated, encompassing stochastic perturbations, various target motion models, and diverse evolution laws for group structures. The sequences of graph-

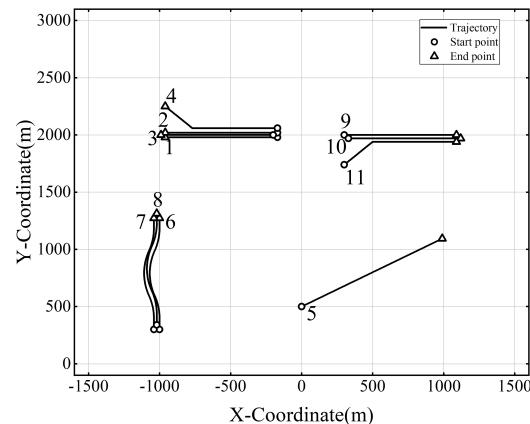


Fig. 6. The target trajectories used in Scenario 1.

structured data are derived by segmenting the generated RGT trajectory and are utilized as the dataset. The total size of the dataset is 160,000, with 80% allocated to the training set and the remaining 20% designated for the validation set. The neural network was built and trained using PyTorch [36], utilizing the Adam optimizer with a learning rate of 0.0001. The training process was carried out in batches of 64 samples, spanning a total of 200 epochs. The detailed architecture of the network depicted in Fig. 4 is outlined in Table IV. Hyperparameters were meticulously fine-tuned through grid search, conducted independently for each experiment, as detailed in [23].

The GNN-LMB filter proposed in this paper is contrasted with the standard LMB filter and the prevalent SDE-LMB filter, both of which lack a learnable component. The SDE-LMB filter [1] is introduced to capture the collective motion of RGT using SDE. Additionally, to emphasize the robust capability of GNN in capturing node interactions, it is compared with an intelligent approach utilizing CNN, referred to as the CNN-LMB filter. This approach, inspired by the concept outlined in [18], replaces the central module of GNN-LMB with convolutional social pooling layers, refining the concept of social pooling layers [39]. To ensure the rigor of the experimental design, the EGN is also incorporated into both the SDE-LMB filter and CNN-LMB filter. The comparative analysis is presented in subsection B. Furthermore, within the realm of RGTT, maintaining the consistency of target labels is of paramount importance. Elaborate experiments and discussions concerning this aspect are outlined in subsection C. To further scrutinize the contributions of individual system components to the overall performance, an ablation study is expounded upon in subsection D. Furthermore, all simulation results are averaged over 100 independent Monte Carlo (MC) runs.

B. Tracking Performance of GNN-LMB Filter

(1) Scenario 1

To facilitate a clear comparison of the performance of the proposed GNN-LMB filter, we initially consider

TABLE IV
Components and Dimensions of the Neural Network Module

Module	Component	Detail description
Encoder	GRU	With a hidden layer dimension of 64.
	GAT+	Comprising a fully connected (FC) layer and a single-headed GAT layer implemented by GATv2Conv module in PyTorch Geometric [37], both with an output dimension of 192.
Decoder	Attention	Comprising two FC layers, with output dimensions 64 and $t_h + 1$.
	FC layer used to calculate \hat{q}_{t+1}	With an output dimension of 64.
Existence probability	FC layer used to calculate $\hat{r}_+^{(\ell)}$	With an output dimension of 1.
	FC layer used to calculate $u_{t+1}^{(\ell,j)}$	Solving u_1 and u_2 in (24), with an output dimension of $2 * N_{GC}$.
Spatial distribution	FC layer used to calculate $Q_{t+1}^{(\ell,j)}$	Solving σ in (45), with an output dimension of N_{GC} .
	FC layer used to calculate $\hat{w}_+^{(\ell,j)}$	With an output dimension of N_{GC} .
	Neural ODE	Comprising two groups of FC layers, each with two layers, approximating f_1 and f_2 in (24), with output dimensions of 16 and 1.
	EKF	Computing the Jacobian matrix of f_1 and f_2 output by the Neural ODE module, using functorch [38].

TABLE V
The Initial State and the Birth/Death Time (Scenario 1)

Target ID	Initial State	Birth time (s)	Death time (s)
1	[-170, -10, 1980, 0]	1	80
2	[-170, -10, 2020, 0]	1	80
3	[-200, -10, 2000, 0]	1	80
4	[-170, -10, 2060, 0]	1	80
5	[0, 10, 500, 6]	1	100
6	[-1000, 0, 300, 10]	1	100
7	[-1040, 0, 300, 10]	1	100
8	[-1020, 0, 340, 10]	1	100
9	[300, 10, 2000, 0]	21	100
10	[330, 10, 1970, 0]	21	100
11	[300, 10, 1740, 10]	21	100

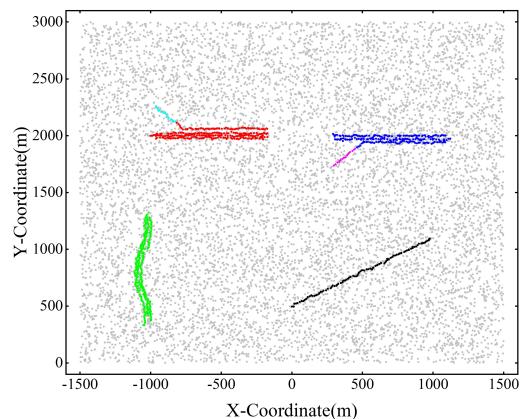


Fig. 7. The tracking outcome achieved by the GNN-LMB filter in Scenario 1. The gray dots symbolize clutter, whereas the colored dots illustrate the inferred trajectory of the target.

a straightforward scenario involving independent moving targets, group targets splitting and merging, where the motion models of each target are relatively simple. The kinematic parameters of the targets are detailed in Table V, and the corresponding target trajectory is illustrated in Fig. 6. In Fig. 7, the tracking results of the GNN-

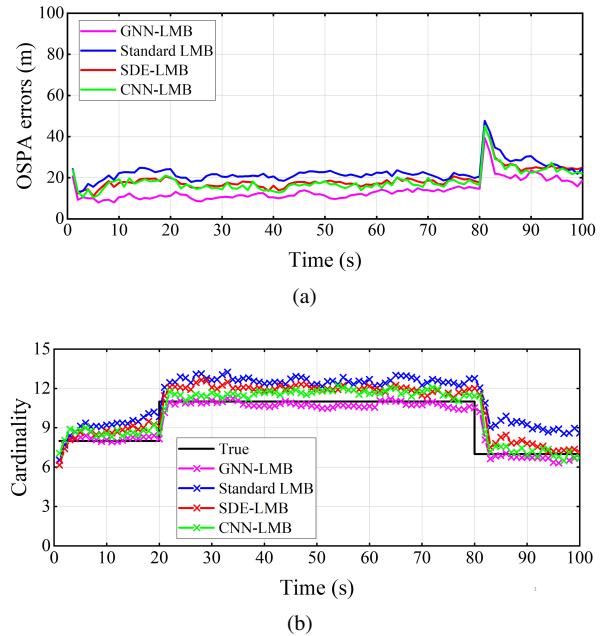


Fig. 8. The tracking performance of each filter in Scenario 1. (a) The OSPA error; (b) The target cardinality estimation.

LMB filter in a single MC run are presented. Different group structures are distinguished by color, providing a visual demonstration of the GNN-LMB filter's tracking capability for RGT.

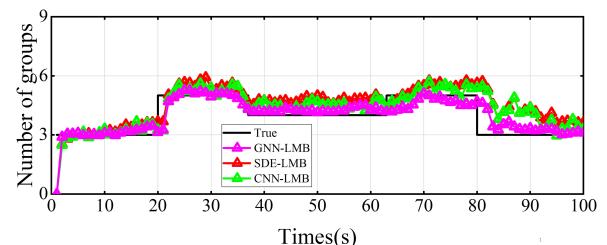


Fig. 9. The group number estimation results in Scenario 1.

TABLE VI
The Initial State and the Birth/Death Time (Scenario 2)

Target ID	Initial State	Birth time (s)	Death time (s)
1	[530, -20, 1880, 0]	1	80
2	[530, -20, 1920, 0]	1	80
3	[500, -20, 1900, 0]	1	80
4	[530, -20, 1960, 0]	1	80
5	[-220, 0, 820, 30]	1	100
6	[-200, 0, 800, 30]	1	100
7	[-240, 0, 800, 30]	1	100
8	[-220, 0, 840, 30]	1	100
9	[20, 30, 500, -10]	21	100
10	[50, 30, 470, 0]	21	100
11	[20, 30, 530, 10]	21	100

Fig. 8(a) and Fig. 8(b) showcase the tracking performance and target cardinality estimation of the GNN-LMB filter in comparison to other filters described in subsection A. As observed in Fig. 8(a) and Fig. 8(b), in this uncomplicated scenario, the tracking performance of the SDE-LMB filter, CNN-LMB filter, and GNN-LMB filter, which incorporate group structure information, surpasses that of the standard LMB filter. Notably, the proposed GNN-LMB filter, capable of learning the interaction between RGT more flexibly, exhibits the most outstanding performance.

Fig. 9 illustrates the group number estimation results of the three filters considering the group structure. While there is a slight delay in the estimation results when the group undergoes splits or merges, overall, the change in the number of groups can be effectively estimated, with the GNN-LMB filter demonstrating the most effective results.

(2) Scenario 2

To comprehensively assess the efficacy of the proposed algorithm, we devised more intricate simulation scenarios. In this particular setting, the target motion model remains elusive, and significant maneuvers are executed during motion. The trajectory of the target is delineated in Fig. 10, and the specific kinematic parameters are detailed in Table VI.

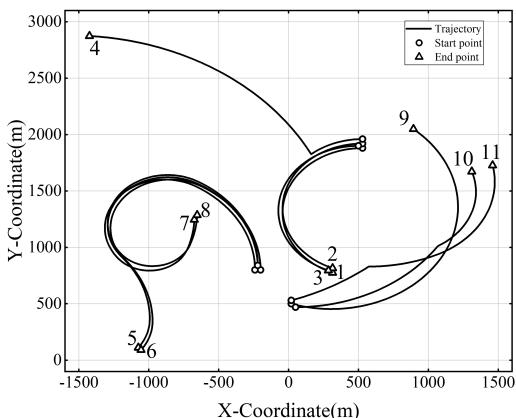
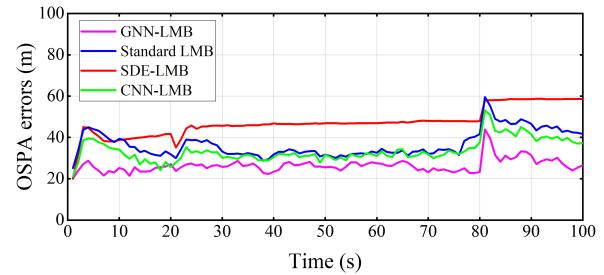
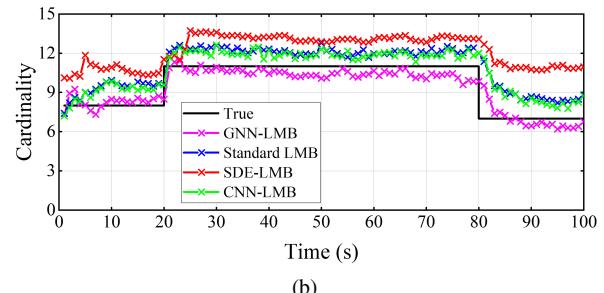


Fig. 10. The target trajectories used in Scenario 2.



(a)



(b)

Fig. 11. The tracking performance of each filter in Scenario 2. (a) The OSPA error; (b) The target cardinality estimation.

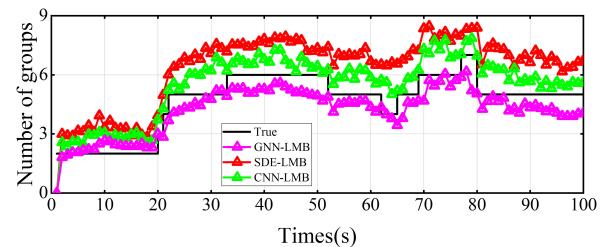
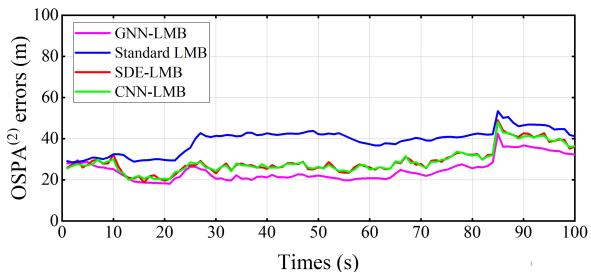


Fig. 12. The group number estimation results in Scenario 2.

Comparisons of the OSPA error and target number estimations for the GNN-LMB filter and alternative methods are presented in Fig. 11(a) and Fig. 11(b), respectively. It is discernible that the tracking precision of each approach diminishes in the presence of a highly maneuvering target without prior motion model information in the scene. Notably, the SDE-LMB filter, modeling interactions among targets in the same group as both attraction and repulsion, fails to accurately portray the intricate influences between targets. Surprisingly, its tracking performance deteriorates even more than the traditional standard LMB filter, which does not account for the group structure. In contrast, both the CNN-LMB and GNN-LMB filters, equipped with learnable components, exhibit a superior ability to maintain tracking performance. When comparing the two, the GNN-LMB filter outperforms due to its effective modeling of complex relationships between nodes and greater flexibility in processing graph data with irregular structures.

The estimated results for the number of groups are depicted in Fig. 12. It becomes evident that group merging and splitting transpire more frequently in this challenging scenario, underscoring the inherent difficulty in tracking RGT. Among the three approaches, the GNN-LMB filter

Fig. 13. The OSPA⁽²⁾ errors of each filter in Scenario 1.

demonstrates the most effective performance, while the SDE-LMB filter exhibits the least efficacy, consistent with the findings in Fig. 11.

C. Label Consistency Performance of GNN-LMB Filter

In this subsection, we present **Scenario 1** from subsection B as an illustrative case to evaluate the effectiveness of the proposed filter in maintaining label consistency. We employed the OSPA⁽²⁾ distance [35] as the evaluation criterion, and the results are depicted in Fig. 13. Notably, the influence of target births or deaths on OSPA⁽²⁾ distance persists due to the presence of the sliding window, resulting in a smoother curve evolution in Fig. 13 compared to Fig. 8. The GNN-LMB filter demonstrates superior performance by accurately capturing interactions among RGT, thereby facilitating more precise prediction of states for each target. This mitigates the challenge of associating data between dense targets and corresponding measurements, thereby reducing the frequent switching of RGT labels. Conversely, the standard LMB filter inherently tends to produce fragmented trajectories with inconsistent labels, particularly evident when handling targets in close proximity, leading to the poorest performance among the four filters.

Next, the clutter rate λ_c is incrementally increased every 20 intervals from 20 to 80, and the OSPA⁽²⁾ errors averaged over time for each filter are compared, as depicted in Table VII. The results indicate that while the GNN-LMB filter consistently outperforms the other filters, its relative performance improvement compared to the standard LMB filter diminishes with increasing clutter. For instance, there is a 36.76% improvement when $\lambda_c = 20$, contrasted with only a 16.32% improvement when $\lambda_c = 80$. This implies that as the clutter rate increases, the proposed GNN-LMB filter encounters challenges in effectively addressing label switching in RGTT.

It is noteworthy that recent studies have applied hypergraph matching techniques to RGTT data association, effectively reducing label switching [40] [41]. Indeed, GNN has been demonstrated to efficiently address the challenge of node matching in hypergraph data [42]. Therefore, besides employing the GNN-GRU module to learn interactions among RGT as proposed in this paper, exploring its potential in mitigating label switching

TABLE VII

The Average OSPA⁽²⁾ Errors (m) for Different Clutter Rate

	λ_c	20	40	60	80
Average OSPA ⁽²⁾	Standard LMB	39.23	40.86	44.18	48.96
	SDE-LMB	29.08	31.25	36.54	42.72
	CNN-LMB	28.96	31.13	36.65	42.54
	GNN-LMB	24.81	26.66	32.08	40.97

TABLE VIII

The Initial State and the Birth/Death Time (Scenario 3)

Target ID	Initial State	Birth time (s)	Death time (s)
1	[300, 10, 1100, 0]	1	80
2	[270, 10, 1130, 0]	1	80
3	[300, 10, 1050, 0]	1	80
4	[700, 10, 1150, 0]	41	80
5	[-470, 10, 2000, 0]	21	100
6	[-500, 10, 1970, 0]	21	100
7	[-500, 10, 2600, -30]	21	100
8	[-800, 0, 300, 20]	21	100
9	[-760, 0, 300, 20]	21	100
10	[-780, 0, 340, 20]	21	100

induced by intricate RGTT data association presents a promising avenue for future research.

D. Ablation Study

The ablation study endeavors to elucidate the most pivotal elements contributing to the system's efficacy, thus guiding its refinement and optimization. [23] conducted ablation experiments on each constituent of the neural network employed in this paper, elucidating that the elimination of EKF and neural ODE components significantly compromises performance. Moreover, in the comparative assessment of the encoder and decoder, it was noted that integrating GNN in the latter yielded more discernible enhancements in performance than in the former. Additionally, [23] juxtaposed the impact of various GNN models on overall performance, while [32] scrutinized the influence of diverse motion models and numerical solvers.

In this paper, the model in [23] undergoes refinement and adaptation to accommodate the RGTT problem, introducing two innovative components: a component for predicting the existence probability of LMB density and an EGN component. Furthermore, the exploration of implementing the LMB filter with GM model is paramount for practical applications. However, in [23], there's no explicit assessment of how a GM model influences system performance. Therefore, the ablation study of the three aforementioned components is conducted in this subsection.

To simulate a more intricate group structure evolution, a scenario involving target birth, merging, and separation within the same group (referred to as **Scenario 3** thereafter) is established, as illustrated in Fig. 14. The kinematic parameters are detailed in Table VIII. The

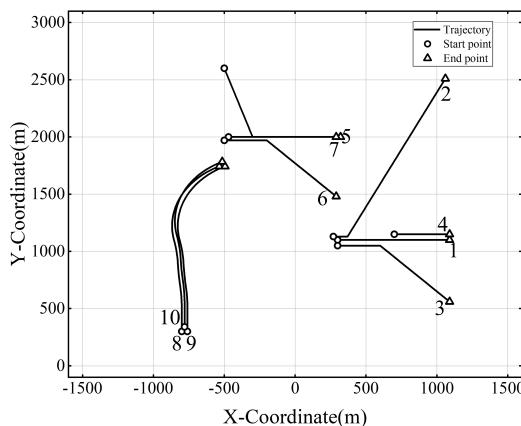


Fig. 14. The target trajectories used in Scenario 3.

TABLE IX
The Result of Ablation Study

Environment Index	Existence probability	EGN	GM model	Average OSPA	Average OSPA ⁽²⁾
<i>Scenario I</i>	I ₁₁	✓	✓	13.80	24.81
	I ₁₂	✗	✓	14.61	25.72
	I ₁₃	✓	✗	13.96	25.02
	I ₁₄	✗	✗	14.71	25.93
	I ₁₅	✓	✓	20.42	33.62
<i>Scenario 3</i>	I ₃₁	✓	✓	24.44	34.27
	I ₃₂	✗	✓	25.23	34.89
	I ₃₃	✓	✗	26.38	37.26
	I ₃₄	✗	✗	28.25	39.73
	I ₃₅	✓	✓	30.06	42.51

ablation study considers **Scenario 1** (from subsection B) and **Scenario 3** as illustrative examples, utilizing OSPA and OSPA⁽²⁾ as criteria for performance evaluation. The results are depicted in Table IX, where checkmarks and crosses indicate whether the component is included in the network.

In the experiment where the existence probability prediction component is excluded, setting $P_s^{(\ell)} = 0.99$, and then solving for $r_{+,S}^{(\ell)}$ with (5) and (7). It is observed that the absence of this component does indeed affect the overall performance to a certain extent (see I₁₂ or I₃₂). This is due to the fact that the value of 0.99 is empirically determined and cannot be dynamically adjusted according to the movement of the target. Consequently, this fixed and relatively high value may lead to an overestimation of the number of targets during tracking and a delay in response when targets disappear, thereby adversely impacting tracking performance.

The exclusion of the EGN in **Scenario 3** has a more significant impact on performance compared to **Scenario 1**, owing to the more intricate changes in group structure in **Scenario 3** (see I₁₁ vs I₁₃ and I₃₁ vs I₃₃). The primary distinction between a GNN-LMB filter without the EGN and one incorporating it resides in the input to the encoder. In the former, the initial step entails identifying the central target v and its neighboring nodes, whose Mahalanobis

distance from v is less than the threshold at time t . Subsequently, the sequence of fully connected graph structures for these targets is traced back from time $t - t_h$ to t , forming the input for the neural network encoder. In contrast, the latter employs the EGN to dynamically estimate the group structure at each moment, directly utilizing the group structure from time $t - t_h$ to t as its input. With the EGN, all targets in the scene are simultaneously predicted, eliminating the need for the concept of a center target. The OSPA error and target number estimation of I₃₁ and I₃₃ are elaborated upon in Fig. 15. The results underscore that the incorporation of the EGN enhances the accuracy of group structure estimation, preventing the oversight of valuable neighbor nodes. This, consequently, positively influences the overall enhancement in tracking performance and bolsters the adaptability of the model to complex environments.

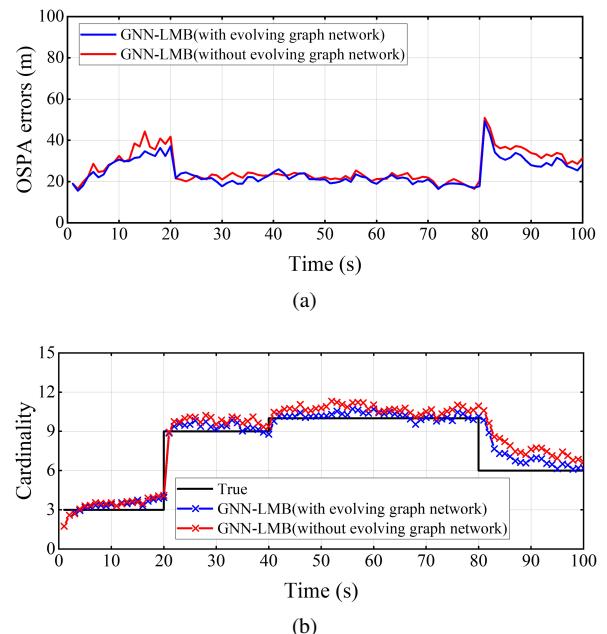
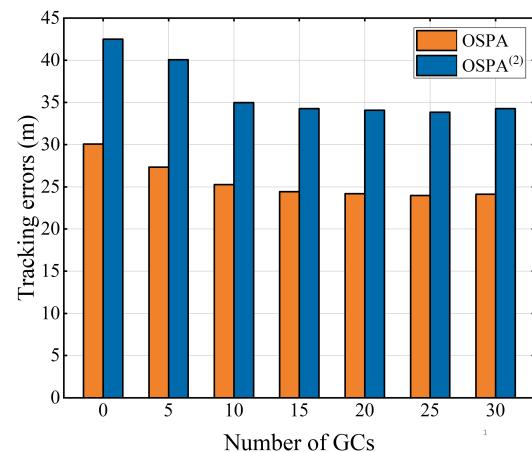


Fig. 15. The tracking performance of two GNN-LMB filters. (a) The OSPA error; (b) The target cardinality estimation.

Fig. 16. Tracking errors under different N_{GC} in Scenario 3.

In I₁₅ and I₃₅, the GM model is omitted, i.e., $N_{GC} = 0$. The absence of GM model results in a significant loss of tracking performance (see I₁₅ or I₃₅). This underscores the importance of GM model, which effectively captures data uncertainty, enhances the model's capacity to fit complex data distributions, and improves overall model expression and generalization capabilities. Moreover, the comparative analysis of tracking performance across various N_{GC} values is depicted in Fig. 16 (using **Scenario 3** as an example). The findings suggest that augmenting N_{GC} within a delimited scope enhances tracking performance. However, increasing model intricacy amplifies the risk of overfitting, and the proliferation of parameters complicates model training, potentially leading to sluggish or even declining performance enhancements.

V. Conclusion

In this paper, we introduce an LMB filter augmented with a deep neural network tailored for RGTT. This innovative approach transcends the constraints imposed by imperfect assumptions about target motion models and interactions among group members inherent in traditional RGTT methodologies. Our methodology incorporates a temporal GNN to assimilate interaction information among group members, and a neural ODE to articulate the motion model of the targets. Additionally, we integrate an evolving network to dynamically deduce the real-time structure of the target group. Through simulation experiments, we substantiate the efficacy of the proposed filter in achieving superior tracking performance and showcase the augmented capabilities of the evolving network. In forthcoming endeavors, our objective is to deepen the integration of the neural network component with the LMB filter, surpassing its current embedding solely in the prediction step. This is pivotal as the interaction between group targets also influences the association between measurements and trajectories.

APPENDIX

A. The complete pseudo-code of group structure estimation.

Algorithm 1 Group target initiation

```

1: INPUT: Vertices set  $V = \{v_1, \dots, v_n\} = \{(x_1, P_1, \ell_1) \dots, (x_n, P_n, \ell_n)\}$ , preset threshold  $\varepsilon$ .
2: OUTPUT: Group structure  $G = \{g_1, g_2, \dots, g_m\}$ .
3: function GROUP-INITIATION( $V, \varepsilon$ )
4:   · Set  $E = \emptyset$ ;
5:   for  $i = 1 : n$  do
6:     for  $j = i + 1 : n$  do
7:       ·  $d_{ij} = \sqrt{(x_i - x_j)(P_i + P_j)^{-1}(x_i - x_j)^T}$ ;
8:       if  $d_{ij} < \varepsilon$  then
9:         ·  $E = E \cup \{(i, j)\}$ ;
```

```

10:  · Identify the connected blocks of the graph based on  $E$ , forming the subgraph  $G = \{g_1, g_2, \dots, g_m\}$ ;
11:  return:  $G = \{g_1, g_2, \dots, g_m\}$ 
```

Algorithm 2 Construct the second-layer group vertices

```

1: INPUT: Vertices set  $V = \{v_1, \dots, v_n\} = \{(x_1, P_1, \ell_1) \dots, (x_n, P_n, \ell_n)\}$ , group structure  $G = \{g_1, g_2, \dots, g_m\}$ .
2: OUTPUT: Vertices set of the second-layer group structure  $V^g = \{v_1^g, \dots, v_m^g\}$ .
3: function CONSTRUCT-SECOND-VERTICES( $V, G$ )
4:   for  $i = 1 : m$  do
5:     · Set  $n_i^g$  denotes the number of vertices in subgroup  $g_i$ ;
6:     ·  $x_i^g = \frac{1}{n_i^g} \sum_{v_j \in g_i} x_j$ ;
7:     ·  $P_i^g = \frac{1}{n_i^g} \sum_{v_j \in g_i} P_j$ ;
8:     ·  $v_i^g = (x_i^g, P_i^g)$ ;
9:   return:  $V^g = \{v_1^g, \dots, v_m^g\}$ 
```

Algorithm 3 Dynamic graph structure estimation

```

1: INPUT: Group structure at moment  $k - 1$  denote as  $G_{k-1} = \{g_{k-1,1}, g_{k-1,2}, \dots, g_{k-1,m}\}$ , LMB update density  $\pi_k$ .
2: OUTPUT: Group structure at moment  $k$  denote as  $G_k = \{g_{k,1}, g_{k,2}, \dots, g_{k,z}\}$ .
3: function GROUP-ESTIMATION( $G_{k-1}, \pi_k$ )
4:   1) Group target update :
5:     · Update state of identical targets within each subgroup in accordance with  $\pi_k$ , obtaining  $G'_k = \{g'_{k,1}, \dots, g'_{k,m}\}$ ;
6:     · For those targets not subjected to updates, generate a virtual vertex based on their anticipated states;
7:     · Set  $V'_{k,i}$  denote the vertices set of subgroup  $g'_{k,i}$  and  $V'_k$  is the union set of all  $V'_{k,i}$ ;
8:   2) Group target split :
9:     · Set  $G_k^{sp} = \emptyset$ ;
10:    for  $i = 1 : m$  do
11:      ·  $g_{k,i}^{new} = \text{GROUP-INITIATION}(V'_{k,i}, \varepsilon)$ ;
12:      · Retain the original group ID without alteration and append the split-generated segment to the set  $G_k^{sp}$ ;
13:    ·  $G'_k = G'_k \cup G_k^{sp} = \{g'_{k,1}, \dots, g'_{k,p}\}$ ;
14:   3) Group target merge :
15:     · Find the second-layer group vertices:
16:       
$$\begin{aligned} V_k^g &= \{v_{k,1}^g, \dots, v_{k,p}^g\} \\ &= \text{CONSTRUCT-SECOND-VERTICES}(V'_k, G'_k); \end{aligned}$$

17:     · Initialize the second-layer group structure:
18:       
$$\begin{aligned} G_k^{2nd} &= \text{GROUP-INITIATION}(V_k^g, \varepsilon_m) \\ &= \{g_{k,1}^{2nd}, \dots, g_{k,q}^{2nd}\}; \end{aligned}$$

· Set  $G_k^{mer} = \emptyset$ ;
for  $i = 1 : q$  do
```

```

19:   if  $g_{k,i}^{2nd}$  comprises solely one second-vertex
20:     then
21:       · Merge the first-layer subgroup that cor-
22:         responds to  $g_{k,i}^{2nd}$  directly into  $G_k^{mer}$ ;
23:     else
24:       · Merge all first-layer vertices of  $g_{k,i}^{2nd}$  into
25:         a set  $V_{k,i}^{2nd}$ ;
26:       ·  $G_{k,i}^{mer} = \text{GROUP-INITIATION}(V_{k,i}^{2nd}, \varepsilon)$ ;
27:       ·  $G_k^{mer} = G_k^{mer} \cup G_{k,i}^{mer}$ ;
28:     · Organize  $G'_k$  into the form
29:        $G'_k = G_k^{mer} = \{g'_{k,1}, \dots, g'_{k,s}\}$ ;
30:   4) Group node addition and deletion :
31:     · Find the second-layer group vertices:
32:        $V_k^g = \{v_{k,1}^g, \dots, v_{k,s}^g\}$ 
33:       = CONSTRUCT-SECOND-VERTICES( $V_k^g, G'_k$ );
34:     · Set  $V_{new} = \{v_1, \dots, v_l\}$  denote the set of
35:       vertices that has not been utilized at the current
36:       moment;
37:     for  $i = 1 : l$  do
38:       · Iterate through all elements in set  $V_k^g$ , iden-
39:         tify  $\hat{v}_{k,j}^g$  with the minimum Mahalanobis dis-
40:         tance to  $v_i$ ;
41:       · Include  $v_i$  in subgroup  $g'_{k,j}$ ;
42:     · Eliminate virtual vertices that have not been
43:       updated for consecutive time intervals;
44:   return:  $G_k = \{g_{k,1}, g_{k,2}, \dots, g_{k,z}\}$ 

```

- [9] IEEE Transactions on Signal Processing, vol. 62, no. 12, pp. 3246–3260, 2014.
- [10] W. Liu, S. Zhu, C. Wen, and Y. Yu
Structure modeling and estimation of multiple resolvable group targets via graph theory and multi-bernoulli filter
Automatica, vol. 89, pp. 274–289, 2018.
- [11] W. Liu, Y. Chi, and G. Zhang
Multiple resolvable group estimation based on the glmb filter with graph structure
In *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE, 2018, pp. 960–964.
- [12] W. Liu and Y. Chi
Resolvable group state estimation with maneuver based on labeled rfs and graph theory
Sensors, vol. 19, no. 6, p. 1307, 2019.
- [13] X. Ru, Y. Chi, and W. Liu
A detection and tracking algorithm for resolvable group with structural and formation changes using the gibbs-glmb filter
Sensors, vol. 20, no. 12, p. 3384, 2020.
- [14] Z. Zhang, J. Sun, H. Zhou, and C. Xu
Group target tracking based on ms-member filters
Remote Sensing, vol. 13, no. 10, p. 1920, 2021.
- [15] Q. Li, S. J. Godsill, J. Liang, and B. I. Ahmad
Inferring dynamic group leadership using sequential bayesian methods
In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8911–8915.
- [16] S. K. Pang, J. Li, 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.
- [17] G. Li, G. Li, and Y. He
Labeled multi-bernoulli filter based multiple resolvable group targets tracking with leader-follower model
IEEE Transactions on Aerospace and Electronic Systems, 2023.
- [18] A. Gning, L. Mihaylova, S. Maskell, S. K. Pang, and S. Godsill
Group object structure and state estimation with evolving networks and monte carlo methods
IEEE Transactions on Signal Processing, vol. 59, no. 4, pp. 1383–1396, 2010.
- [19] N. Deo and M. M. Trivedi
Convolutional social pooling for vehicle trajectory prediction
In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE Computer Society, 2018, pp. 1549–15498.
- [20] H. Cui *et al.*
Deep kinematic models for kinematically feasible vehicle trajectory predictions
In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10563–10569.
- [21] A. Kullberg, I. Skog, and G. Hendeby
Online joint state inference and learning of partially unknown state-space models
IEEE Transactions on Signal Processing, vol. 69, pp. 4149–4161, 2021.
- [22] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone
Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data
arXiv preprint arXiv:2001.03093, 2020.
- [23] J. Li, H. Ma, Z. Zhang, J. Li, and M. Tomizuka
Spatio-temporal graph dual-attention network for multi-agent prediction and tracking
IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 8, pp. 10556–10569, 2021.
- [24] T. Westny, J. Oskarsson, B. Olofsson, and E. Frisk
Mtp-go: Graph-based probabilistic multi-agent trajectory prediction with neural odes

REFERENCES

- [1] L. Li, Q. Wu, B. Yang, S. Wei, and J. Wang
Labeled multi-bernoulli filter based group target tracking using sde and graph theory
In *2021 IEEE 24th International Conference on Information Fusion (FUSION)*. IEEE, 2021, pp. 1–8.
- [2] G. Li, G. Li, and Y. He
Resolvable group target tracking via multi-bernoulli filter and its application to sensor control scenario
IEEE Transactions on Signal Processing, vol. 70, pp. 6286–6299, 2022.
- [3] Y. Xia, K. Granström, L. Svensson, Á. F. García-Fernández, and J. L. Williams
Extended target poisson multi-bernoulli mixture trackers based on sets of trajectories
In *2019 22th International Conference on Information Fusion (FUSION)*. IEEE, 2019, pp. 1–8.
- [4] R. Mahler
Statistical multisource-multitarget information fusion. Artech, 2007.
- [5] Y. Bar and T. Shalom
Fortmann, “tracking and data association”
1988.
- [6] S. S. Blackman and R. Popoli
Design and analysis of modern tracking systems
(No Title), 1999.
- [7] X. Hao, Y. Liang, W. Zhang, and L. Xu
Structure identification and tracking of multiple resolvable group targets with circular formation
In *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, vol. 9. IEEE, 2020, pp. 910–915.
- [8] S. Reuter, B.-T. Vo, B.-N. Vo, and K. Dietmayer
The labeled multi-bernoulli filter

- [24] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud
Neural ordinary differential equations
Advances in neural information processing systems, vol. 31, 2018.

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

[26] D. B. West *et al.*
Introduction to graph theory. Prentice hall Upper Saddle River, 2001, vol. 2.

[27] A. R. Runnalls
Kullback-leibler approach to gaussian mixture reduction
IEEE Transactions on Aerospace and Electronic Systems, vol. 43, no. 3, pp. 989–999, 2007.

[28] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio
Empirical evaluation of gated recurrent neural networks on sequence modeling
arXiv preprint arXiv:1412.3555, 2014.

[29] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini
The graph neural network model
IEEE transactions on neural networks, vol. 20, no. 1, pp. 61–80, 2008.

[30] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip
A comprehensive survey on graph neural networks
IEEE transactions on neural networks and learning systems, vol. 32, no. 1, pp. 4–24, 2020.

[31] S. Brody, U. Alon, and E. Yahav
How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021.

[32] T. Westny, J. Oskarsson, B. Olofsson, and E. Frisk
Evaluation of differentially constrained motion models for graph-based trajectory prediction
arXiv preprint arXiv:2304.05116, 2023.

[33] S. Reuter
Multi-object tracking using random finite sets
Ph.D. dissertation, Universität Ulm, 2014.

[34] 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.

[35] M. Beard, B. T. Vo, and B.-N. Vo
Ospa (2): Using the ospa metric to evaluate multi-target tracking performance
In *2017 International Conference on Control, Automation and Information Sciences (ICCAIS)*. IEEE, 2017, pp. 86–91.

[36] A. Paszke *et al.*
Pytorch: An imperative style, high-performance deep learning library
Advances in neural information processing systems, vol. 32, 2019.

[37] M. Fey and J. E. Lenssen
Fast graph representation learning with pytorch geometric
arXiv preprint arXiv:1903.02428, 2019.

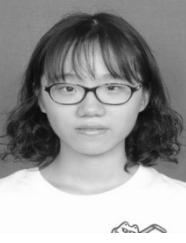
[38] R. H. He and R. Zou
functorch: Jax-like composable function transforms for pytorch 2021.

[39] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese
Social lstm: Human trajectory prediction in crowded spaces
In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2016, pp. 961–971.

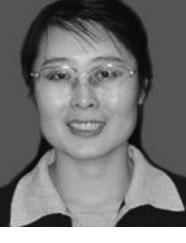
[40] G. Li, G. Li, and Y. He
Distributed multiple resolvable group targets tracking based on hypergraph matching

[41] H. Yu, W. An, R. Zhu, and R. Guo
A hypergraph matching labeled multi-bernoulli filter for group targets tracking
IEICE TRANSACTIONS on Information and Systems, vol. 102, no. 10, pp. 2077–2081, 2019.

[42] E. Faerman, O. Voggenreiter, F. Borutta, T. Emrich, M. Berrendorf, and M. Schubert
Graph alignment networks with node matching scores
Proceedings of Advances in Neural Information Processing Systems (NIPS), vol. 2, 2019.



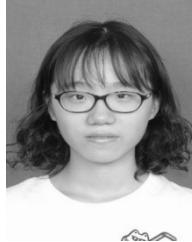
Yue Yu was born in 2000. She received the B.S. degree in the school of electronics and information engineering from Harbin Institute of Technology, Harbin, China, in 2021. Since then, she has continued to study for her doctorate at the same school. Her current research interests include nonlinear filtering, estimation, and information fusion. She is also interested in communication topology in networks.



Mei Liu was born in 1963. She received the Ph.D. degree in electronic and communication engineering from Harbin Institute of Technology, Harbin, China, in 2006. She is currently a Professor and a Ph.D. supervisor of signal and information processing with the school of electronics and information engineering, Harbin Institute of Technology. In recent years, she has been engaged in scientific research and teaching in multi-sensor data fusion, array signal processing, and satellite communication technology.



Bo Li ’s research interests include machine learning, deep learning, natural language processing, sparse signal processing, compressed sensing, etc. He is currently working in the Key laboratory of Intelligent Technology and Application of Marine Equipment, the Key laboratory of Ship Intelligent System and Technologies, the Key laboratory of Environment Intelligent Perception, the College of Intelligent System Science and Engineering, Harbin Engineering University.



Yue Yu was born in 2000. She received the B.S. degree in the school of electronics and information engineering from Harbin Institute of Technology, Harbin, China, in 2021. Since then, she has continued to study for her doctorate at the same school. Her current research interests include nonlinear filtering, estimation, and information fusion. She is also interested in communication topology in networks.



A black and white portrait of a woman with short dark hair and glasses, wearing a light-colored collared shirt. She is smiling at the camera.

Mei Liu was born in 1963. She received the Ph.D. degree in electronic and communication engineering from Harbin Institute of Technology, Harbin, China, in 2006. She is currently a Professor and a Ph.D. supervisor of signal and information processing with the school of electronics and information engineering, Harbin Institute of Technology. In recent years, she has been engaged in scientific research and teaching in multi-sensor data fusion, array signal processing, and satellite communication technology.



Bo Li 's research interests include machine learning, deep learning, natural language processing, sparse signal processing, compressed sensing, etc. He is currently working in the Key laboratory of Intelligent Technology and Application of Marine Equipment, the Key laboratory of Ship Intelligent System and Technologies, the Key laboratory of Environment Intelligent Perception, the College of Intelligent System Science and Engineering, Harbin Engi-

Enhancing Resolvable Group Target Tracking: Integration of Labeled Multi-Bernoulli Filter with Deep Learning Approaches

Yue Yu, Mei Liu, Bo Li

Abstract

This paper introduces an innovative approach for resolvable group targets tracking (RGTT) through the integration of classical Bayesian filtering and sophisticated deep learning techniques. Numerous extant RGTT methods oversimplify target motion models and interactions within groups, limiting their applicability and compromising tracking performance. Deep learning methodologies are recurrently leveraged to overcome these challenges owing to their adeptness in delineating intricate mappings. We develop a neural network tailored for RGTT, incorporating neural ordinary differential equation (Neural ODE) to articulate the motion model of targets. Additionally, it predominantly integrates recurrent neural network (RNN) and graph neural network (GNN) modules. Similar to multi-target tracking tasks, the RNN module is enlisted here to model nonlinear dynamic systems. The distinction lies in the domain of resolvable group targets, where an additional reliance on graph theory becomes imperative for estimating group structures. Consequently, this paper introduces a GNN module into the deep learning network to learn the interaction relationships among targets within the same group. Furthermore, the seamless fusion of this deep network with the labeled multi-Bernoulli (LMB) filter augments the pragmatic viability of the deep learning algorithm. Simulation results elucidate the efficacy of the proposed algorithm.

Index Terms

Yue Yu is with the School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail: yuyue@stu.hit.edu.cn).

Mei Liu is the professor and a Ph.D. supervisor of signal and information processing with the School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail: liumei@hit.edu.cn).

Bo Li is with the Key laboratory of Intelligent Technology and Application of Marine Equipment, the Key laboratory of Ship Intelligent System and Technologies, the Key laboratory of Environment Intelligent Perception, the College of Intelligent System Science and Engineering, Harbin Engineering University, Harbin 150001, China (e-mail: libo520209@126.com).

1
2 21 Resolvable group target tracking, graph neural networks, labeled multi-Bernoulli filter, neural ordi-
3 22 nary differential equation.
4
5
6
7 23 I. INTRODUCTION
8
9 24 Group targets [1] can be conceptualized as an assemblage of targets, exhibiting a specific
10 25 spatial proximity and a congruent motion pattern within a defined temporal frame. These group
11 26 targets are broadly categorized into two classifications: resolvable group targets [2], wherein each
12 27 target occupies a distinct resolution unit, and unresolvable group targets [3], wherein multiple
13 28 targets coexist within a singular resolution unit. In the realm of tracking, resolvable group targets
14 29 tracking (RGTT) holds the capability to differentiate individual targets within the group and
15 30 ascertain their interdependencies. This distinction holds particular significance in analyzing the
16 31 target's friend or foe attributes, predicting the target's intention, and estimating the threat posed
17 32 by the target within the exigent battlefield environment. Consequently, the primary focus of this
18 33 paper centers on RGTT.
19
20
21
22
23
24
25
26
27 Similar to traditional multi-target tracking, the fundamental challenge in RGTT lies in con-
28 34 currently addressing uncertainties in both target numbers and states. The random finite set
29 35 (RFS) theory [4], encompassing target birth, death, survival, and sensor detection, provides a
30 36 comprehensive theoretical framework for target tracking, obviating the need for intricate data
31 37 association. Consequently, when comparing joint probabilistic data association (JPDA) [5] and
32 38 multiple hypothesis tracking (MHT) [6] methods, it becomes evident that RFS is a more suitable
33 39 choice for addressing the problem of RGTT. Nevertheless, diverging from the conventional target
34 40 tracking paradigm where targets are perceived as independent entities, in RGTT, the significance
35 41 of the group structure and interactions among targets within the same group is emphasized,
36 42 thereby substantially enhancing the integrity, accuracy, and stability. [4] [7]. Graph theory [8]
37 43 furnishes a framework for illustrating resolvable group target (RGT), wherein individual targets
38 44 serve as nodes and their interactions are represented as edges in the graph.
39
40
41
42
43
44
45
46
47
48 Currently, numerous RGTT algorithms based on RFS and graph theory have been devel-
49 49 oped. [9] depicts group targets as a directed graph, wherein interactions among targets are
50 50 expressed through the impact of parent node states on their respective child nodes, and tracking
51 51 is executed within the framework of the multi-Bernoulli (MB) filter. This strategy is expanded
52 52 upon in [10] [11] [12], yet it is constrained to scenes with relatively fixed group structures
53 53 and imposes strict limitations on the motion models of the targets. To accommodate the diverse
54 54 55
56
57
58
59
60

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
52 and complex movements of targets in real-world scenarios, the virtual leader (VL) model [13] and leader-follower (LF) model [14] are widely adopted. A more general approach involves utilizing stochastic differential equations (SDE) to model group structures, capturing real-world uncertainties. [15] presents a group structure model with VL using SDE, while [16] employs a similar strategy, focusing on the LF model in conjunction with the labeled multi-Bernoulli (LMB) filter. Nevertheless, the identification of leaders exhibiting both rationality and robustness presents a formidable challenge, markedly constraining the scope of applicability for such models. Moreover, the VL model neglects the intricacies and interaction among actual constituents within the group, conceivably culminating in an imprecise emulation of group behavior. Another strategy characterizes the interaction among targets in a group as repulsive and restorative forces. [1] derives an approximate repulsive force vector under the LMB filter and employs evolving networks [17] to handle dynamic changes in group topology. Building upon [1], [2] provides an exact state transition function (STF) for the standard MB filter. In the realm of target tracking, the absence of prior motion model parameters is a prevalent challenge. The social force model compounds this issue by introducing parameters pertaining to repulsion and state control, thereby intensifying the complexity of accurate modeling. Furthermore, addressing the dispersion and diffusion matrix of Brownian motion in SDE remains an inescapable challenge for these approaches.

34 In summary, the aforementioned methods, employing physical models, simplify the inherent
35 dynamics of the target to a certain degree, and the model incorporates antecedent parameters
36 formulated empirically. When compared to hand-crafted models, deep learning methods prove
37 more adaptive and better suited for addressing complex real-world challenges. Existing learning-
38 based approaches for vehicle trajectory prediction, which account for the interaction between
39 adjacent vehicles, hold the potential for application in RGTT. For instance, the interactions be-
40 tween neighboring agents are encoded in pooling tensors through convolutional neural networks
41 (CNN) [18]. Nevertheless, the lack of interpretability in deep networks introduces the possibility
42 of producing physically infeasible outputs. Recent developments have seamlessly amalgamated
43 physical models with deep learning methods [19] [20] [21] [22], offering a synthesis of the
44 best attributes from both domains. Drawing inspiration from these methodologies and leveraging
45 the potent capabilities of graph neural networks (GNN) to capture intricate interactions among
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

graph nodes, the MTP-GO¹ is introduced [23]. In this approach, GNN is embedded within a recurrent neural network (RNN) to compute the input and process noise of the state transition equation. The state transition equation is learned through neural ordinary differential equation (neural ODE) [24] without any prior constraints.

To obtain the target trajectory and ensure tracking consistency, this paper explores the use of the LMB filter. The principal contributions of the paper are delineated as follows:

1) GNN learns the interactions among group members, while neural ODE handle the motion model. Different from the approach in [23], we present the incorporation of evolving graph network (EGN) [17] in this context. The continuous estimation of the group structure throughout the tracking process enhances the algorithm's adaptability to dynamic topology changes, obviating the necessity for a fully connected graph to mitigate redundancy.

2) Deep learning is integrated into the LMB filter, optimizing the conventional prediction steps for RGTT. The consideration of target birth, target survival, environmental clutter, and other factors is unified within the LMB filtering framework. Subsequently, we elucidate the implementation process of the Gaussian mixture (GM) for the entire approach.

The subsequent sections of the paper are arranged as follows. Section II introduces foundational knowledge, including the LMB filter and graph theory. Section III delineates the proposed amalgamation of the LMB filter with a deep learning algorithm, succeeded by simulation results in Section IV. Ultimately, Section V draws insightful conclusions.

II. PRELIMINARY

A. The Standard LMB Filter

The Generalized Labeled Multi-Bernoulli (GLMB) filter [25], based on labeled RFS theory, is a groundbreaking closed-form solution for Bayesian filtering in multi-target tracking. Unlike conventional filters, it not only estimates target states but also provides their trajectories simultaneously. In this paper, we utilize the LMB filter as a specific instance of the GLMB filter, essentially serving as a first-moment approximation. The key distinction between the LMB and GLMB filters lies in their treatment of data association uncertainty. The LMB filter assumes that all targets belong to a single component (i.e., a single set of track labels), whereas the GLMB filter accommodates multiple hypotheses, each representing potential sets of target track labels.

¹Multi-agent Trajectory Prediction by Graph-enhanced neural Ordinary differential equations

1
2 111 Experiments detailed in [25] demonstrate that while its performance is slightly lower than the
3 112 GLMB filter, it effectively reduces computational load and conserves storage space. Furthermore,
4 113 a noteworthy advantage of the LMB filter in this paper lies in its representation of trajectories
5 114 in discrete form, thereby facilitating seamless integration with a neural network to predict the
6 115 existence probability and spatial distribution of each independent LMB component. Following
7 116 this, we offer a brief introduction to the LMB filter.

8
9 117 The notion of labeled RFS is initially introduced in [25], followed by the proposal of the
10 118 standard LMB filter in [8]. A succinct overview of the LMB filter is presented herein, and
11 119 detailed implementation specifics can be referenced in [8]. Throughout the paper, we adhere to
12 120 the same notations as stipulated in [8], as delineated in Table I. For the sake of simplicity, the
13 121 time subscript for the prediction item is denoted as +, and the time subscript for the update item
14 122 is omitted.

15
16 123 **Prediction:** Suppose the current LMB density is entirely characterized by the parameter set
17 124 $\pi = \{(r^{(\ell)}, p^{(\ell)})\}_{\ell \in \mathbb{L}}$, wherein $r^{(\ell)}$ and $p^{(\ell)}$ represent the existence probability and spatial
18 125 distribution of a target with label ℓ , respectively. The birth process adheres to an LMB RFS
19 126 and is defined by the parameter set $\pi_B = \{(r_B^{(\ell)}, p_B^{(\ell)})\}_{\ell \in \mathbb{B}}$ or equivalently by

$$\pi_B(\mathbf{X}) = \Delta(\mathbf{X}) w_B(\mathcal{L}(\mathbf{X})) [p_B]^{\mathbf{X}}, \quad (1)$$

20
21 127 where,

$$w_B(L) = \prod_{i \in \mathbb{B}} \left(1 - r_B^{(i)}\right) \prod_{\ell \in L} \frac{1_{\mathbb{B}}(\ell) r_B^{(\ell)}}{1 - r_B^{(\ell)}}, \quad (2)$$

$$p_B(x, \ell) = p_B^{(\ell)}(x), \quad (3)$$

22
23 128 then, the predicted LMB density is denoted as $\pi_+ = \{(r_+^{(\ell)}, p_+^{(\ell)})\}_{\ell \in \mathbb{L}_+}$ with $\mathbb{L}_+ = \mathbb{L} \cup \mathbb{B}$,

$$\pi_+ = \{(r_{+,S}^{(\ell)}, p_{+,S}^{(\ell)})\}_{\ell \in \mathbb{L}} \cup \{(r_B^{(\ell)}, p_B^{(\ell)})\}_{\ell \in \mathbb{B}}, \quad (4)$$

24
25 130 where,

$$r_{+,S}^{(\ell)} = \eta_S^{(\ell)} r^{(\ell)}, \quad (5)$$

$$p_{+,S}^{(\ell)}(x) = \langle P_S^{(\ell)} f^{(\ell)}(x|\cdot), p^{(\ell)} \rangle / \eta_S^{(\ell)}, \quad (6)$$

TABLE I

THE NOTATIONS FOR LMB FILTER

Notation	Explanation
x, z	Single-target state and observation.
X, Z	Multi-target state and observation.
$\mathbf{x}, \mathbf{X}/\pi$	Labeled state/distribution.
$\mathbb{X}/\mathbb{Z}/\mathbb{L}$	Space of state/observation/label.
$\mathcal{F}(\cdot)$	Collection of all finite subsets.
$h^X = \prod_{x \in X} h(x)$	Multi-target exponential notation, h is a real-valued function, X is a set, $h^\emptyset = 1$.
$\delta_Y(X) = \begin{cases} 1 & \text{if } X = Y \\ 0 & \text{otherwise} \end{cases}$	Kronecker Delta function.
$\langle f, g \rangle = \int f(x)g(x)dx$	Inner product.
$\mathcal{L}(X)$	$\mathcal{L} : \mathbb{X} \times \mathbb{L} \rightarrow \mathbb{L}$ be the projection $\mathcal{L}((x, \ell)) = \ell$.
$1_Y(X) = \begin{cases} 1 & \text{if } X \subseteq Y \\ 0 & \text{otherwise} \end{cases}$	Inclusion function.
$\Delta(\mathbf{X}) = \delta_{ \mathbf{X} }(\mathcal{L}(\mathbf{X}))$	Distinct label indicator.
I	Set of track labels.
$+$	Predicted state e.g. $x_+ = x_{k+1 k}$.
\mathbb{B}	Label space of the newborn targets and $\mathbb{L} \cap \mathbb{B} = \emptyset$.
$\mathbb{L}_+ = \mathbb{L} \cup \mathbb{B}$	Predicted label space.
$P_S^{(\ell)}(\cdot)$	Survival probability.
$f^{(\ell)}(x \cdot)$	Single-target Markov transition density.
Θ_{I_+}	Space of mappings $\theta : I_+ \rightarrow \{0, 1, \dots, Z \}$, $\theta_\ell = 0$ means track ℓ is undetected.
$P_D^{(\ell)}(\cdot)$	Detection probability.
$g(z x, \ell)$	Single-target measurement likelihood.
$\kappa(\cdot)$	Poisson clutter intensity.

132

$$\eta_S^{(\ell)} = \left\langle P_S^{(\ell)}, p^{(\ell)} \right\rangle. \quad (7)$$

133 **Update:** Given the measurement set designated as Z , the posterior of the LMB can be

134 expressed as

52
$$\pi(\mathbf{X}|Z) = \left\{ (r^{(\ell)}, p^{(\ell)}) \right\}_{\ell \in \mathbb{L}_+}, \quad (8)$$

53

1
2 135 where,

$$r^{(\ell)} = \sum_{(I_+, \theta) \in \mathcal{F}(L_+) \times \Theta_{I_+}} w^{(I_+, \theta)}(Z) 1_{I_+}(\ell), \quad (9)$$

$$p^{(\ell)}(x) = \frac{1}{r^{(\ell)}} \sum_{(I_+, \theta) \in \mathcal{F}(L_+) \times \Theta_{I_+}} w^{(I_+, \theta)}(Z) 1_{I_+}(\ell) p^{(\theta)}(x, \ell), \quad (10)$$

13
14 137 and,

$$w^{(I_+, \theta)}(Z) \propto w_+(I_+) \left[\eta_Z^{(\theta)} \right]^{I_+}, \quad (11)$$

$$w_+(I_+) = \prod_{i \in \mathbb{L}_+} \left(1 - r_+^{(i)} \right) \prod_{\ell \in I_+} \frac{1_{L_+}(\ell) r_+^{(\ell)}}{1 - r_+^{(\ell)}}, \quad (12)$$

$$p^{(\theta)}(x, \ell | Z) = p_+^{(\ell)}(x) \psi_Z(x, \ell; \theta) / \eta_Z^{(\theta)}(\ell), \quad (13)$$

$$\psi_Z(x, \ell; \theta) = \begin{cases} 1 - P_D^{(\ell)}(x) & \text{if } \theta(\ell) = 0 \\ \frac{P_D^{(\ell)}(x) g(z_{\theta(\ell)} | x, \ell)}{\kappa(z_{\theta(\ell)})} & \text{if } \theta(\ell) > 0 \end{cases}, \quad (14)$$

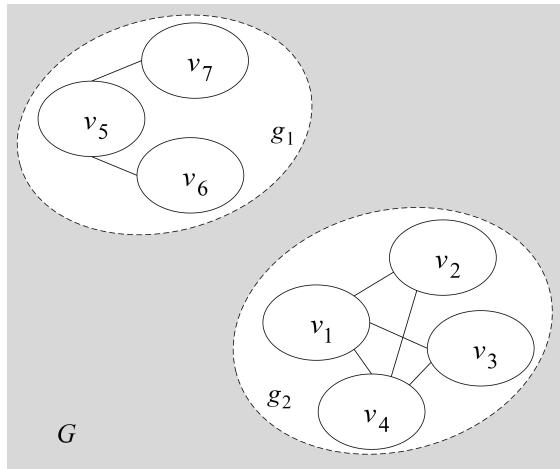
$$\eta_Z^{(\theta)}(\ell) = \left\langle p_+^{(\ell)}, \psi_Z(\cdot, \ell; \theta) \right\rangle. \quad (15)$$

38 142 B. Graph Theory

40 143 The fundamental challenge in RGTT lies in comprehending the collaboration among group
41 targets. In graph theory [26], the relationships between graph vertices are represented by edges,
42 establishing a natural and intrinsic connection with RGTT. In this paper, we conceptualize the
43 group of targets as an undirected graph, and we briefly revisit the pertinent background of graph
44 theory in this subsection.

45 148 The graph structure can be denoted as $G = (V, E)$, where N targets constitute the set of
46 vertices $V = \{v_1, v_2, \dots, v_N\}$. The state, label, and covariance information for target i are all
47 encapsulated in node v_i . Thus, for any two nodes v_i and v_j , their Mahalanobis distance d_{ij} can
48 be calculated. If d_{ij} falls beneath the designated threshold, it is posited that an edge connection
49 subsists between v_i and v_j . The set E embodies the entirety of edges linking vertices in V .

1
2
3 Furthermore, G can be expressed as $G = (g_1, \dots, g_m)$, where g_m is a subgraph of G , i.e. a
4 connected partition in G . Fig. 1 illustrates a graphical representation of groups, demonstrating
5 that the dependencies of v_6 and v_7 are conveyed through v_5 but are not directly connected. In
6 other words, the graph structure in this paper is non-fully connected.
7
8



27 Fig. 1. Graphical representation of groups.
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

III. THE PROPOSED GNN-LMB FILTER

156
157 As is commonly understood, the conventional single-step nonlinear transformation with addi-
158 tive noise assumes the ensuing form:
159

$$\dot{x} = f(x, u) + w, \quad (16)$$

160 where $f(\cdot)$ represents a STF, u denotes an external input, and the process noise $w \sim \mathcal{N}(0, Q)$.
161

162 In RGTT, targets in close proximity with similar motion patterns interact, resulting in a more
163 intricate STF, particularly evident in the prediction stage of the LMB filter. The paper's framework
164 is illustrated in Fig. 2. incorporates a neural network module. This module draws insights from
165 historical states $\pi_{1:k-1}$ and group structures $G_{1:k-1}$ to extract interactions among RGT, thereby
166 learning and adapting the STF. This, in turn, predicts the LMB density $\{(r_{+,S}^{(\ell)}, p_{+,S}^{(\ell)})\}_{\ell \in \mathbb{L}}$ of
167 surviving targets. Notably, the update of the group structure follows the LMB update. Subsections
168 III-A and III-B introduce the group structure estimation module and the neural network module,
169 respectively. The combination of these modules and the GM implementation are elaborated in
subsection III-C.

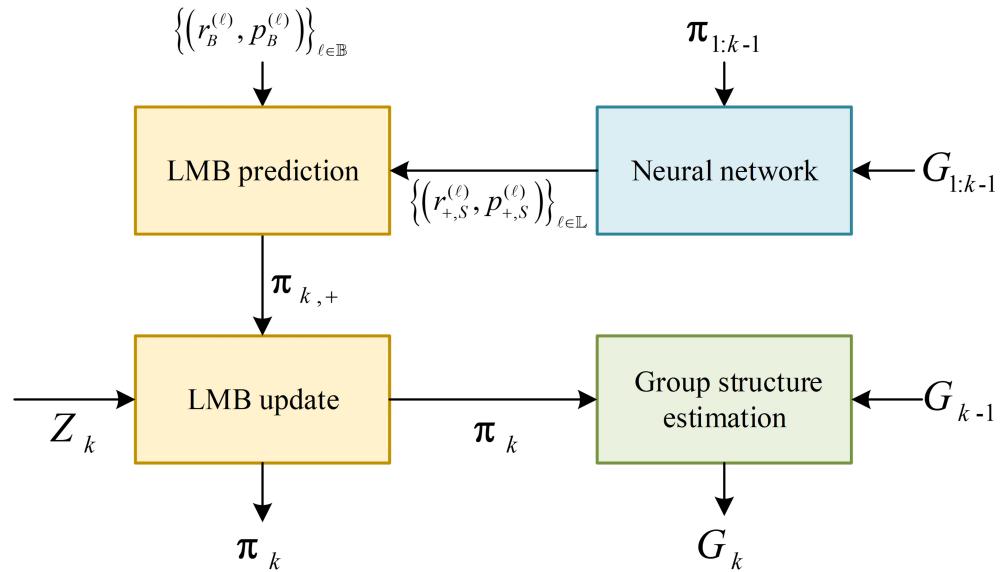


Fig. 2. The complete block diagram of the proposed filter.

A. Group Structure Estimation

In this paper, the distribution of targets is represented in GM form. To facilitate the estimation of group structure, the targets extracted at each time step are transformed into only one Gaussian component (GC) [27]. The mean and variance of GC are denoted as x_i and P_i , respectively. Employing graph theory to model group targets, the graph nodes v_i encompass the state x_i , variance P_i and label ℓ_i of RGT. For a detailed algorithmic representation, refer to Appendix A for the complete pseudo-code of group structure estimation based on EGN [1]. Algorithm 1 is utilized for group structure initialization, and Algorithm 3 is applied for subsequent dynamic group management. To alleviate computational load, the center of the first-layer subgroup serves as the vertex to establish the second-layer group structure. The construction process is elaborated in Algorithm 2, as illustrated in Fig. 3. It is noteworthy that v_5 in Fig. 3, a singular node in the first-layer group structure, maintains its individual state in the second-layer group structure. Diverging from [1], the second-layer group also employs the group initialization algorithm instead of calculating the Mahalanobis distance between vertex pairs. This approach helps prevent ambiguity in the merging of multiple second-layer groups.

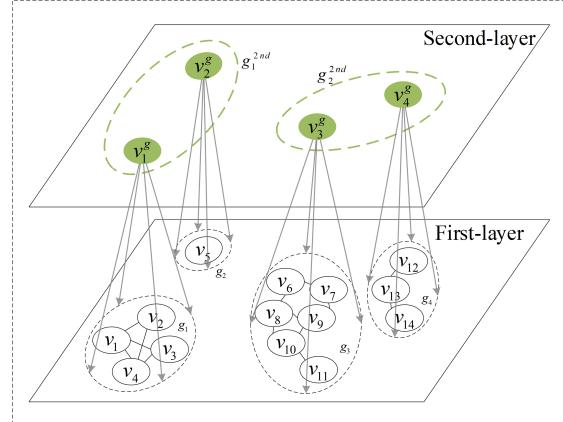


Fig. 3. Double-layered group structure.

185 B. Neural Network Module

186 The predicted LMB density of survival targets is denoted as $\left\{ \left(r_{+,S}^{(\ell)}, p_{+,S}^{(\ell)} \right) \right\}_{\ell \in \mathbb{L}}$. Since $p_{+,S}^{(\ell)}$
 187 predominantly depends on the STF $f^{(\ell)}(x|\cdot)$, the primary challenge in this subsection is acquiring
 188 $r_{+,S}^{(\ell)}$ and $f^{(\ell)}(x|\cdot)$. The core of prediction lies in extrapolating from existing historical informa-
 189 tion. Gated Recurrent Units (GRU) [28] have demonstrated robust proficiencies in executing such
 190 endeavors, yet their focus is solely on the individual node under consideration. Consequently, the
 191 linear mapping component of GRU is replaced with GNN [29], which aggregates information
 192 from neighboring nodes while simultaneously considering the node's own features. This transfor-
 193 mation creates a joint encoder that caters to the requirements of RGTT. Subsequently, a decoder
 194 is created with a structure largely similar to the encoder. The output of the decoder encapsulates
 195 the probability of existence, denoted as r , within the LMB density. It also incorporates the input
 196 term u and the covariance matrix Q for process noise in the state transition equation. Finally, the
 197 STF of the target is characterized by the neural ODE. The comprehensive model of this network
 198 is illustrated in Fig. 4, and the key modules are sequentially introduced below.

199 The components necessary for the GNN are defined as follows. Firstly, the graph structure,
 200 denoted as G , is obtained following the description in subsection III-A. Secondly, the node
 201 feature, in this context, is represented by the state of each RGT, encompassing position and
 202 velocity, as illustrated by q in Fig. 4. Lastly, the edge feature is represented by the Mahalanobis
 203 distance between two mutually connected nodes.

204 (1) Encoding

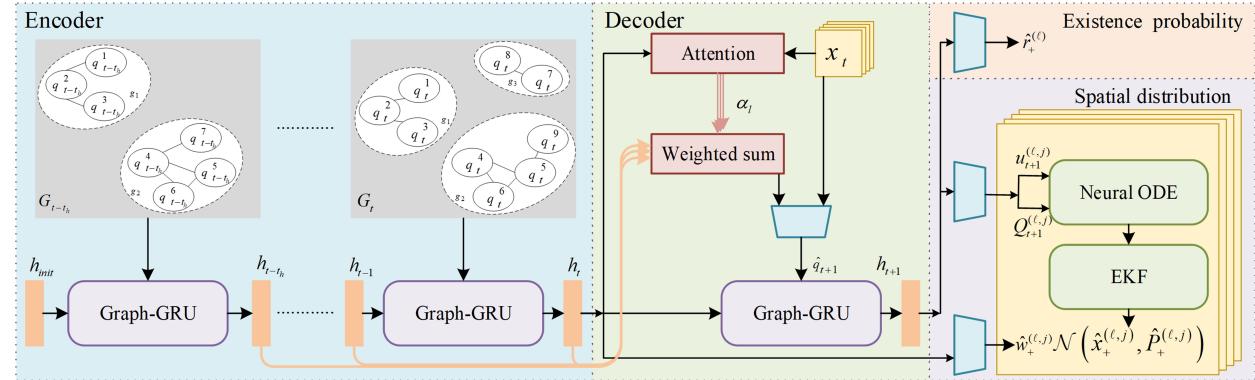


Fig. 4. The comprehensive flowchart of the neural network module. The symbol with subscript (+) in the figure denotes a general form. When the module is employed to solve the LMB density of the survival target, the subscript will be specified as (+, S).

For node v , Graph-GRU [23] is introduced to collectively extract features from the historical sequences of v and its neighboring nodes. The essence of Graph-GRU can be formulated as follows:

$$[\kappa_{r,i}^{\nu} \parallel \kappa_{z,i}^{\nu} \parallel \kappa_{h,i}^{\nu}] = GNN_q \left(q_i^{\nu}, \{q_i^{\tau}\}_{\tau \neq \nu} \right), \quad (17)$$

$$[\xi_{r,i}^{\nu} \parallel \xi_{z,i}^{\nu} \parallel \xi_{h,i}^{\nu}] = GNN_h \left(h_{i-1}^{\nu}, \{h_{i-1}^{\tau}\}_{\tau \neq \nu} \right), \quad (18)$$

where \parallel denotes the concatenation operation. q_i^{ν} represents the current input variable, and h_{i-1}^{ν} is the hidden variable from the previous moment. In the conventional GRU, a total of six intermediate vectors in (17) and (18) are obtained through the linear mapping of q_i and h_{i-1} , respectively. In this context, GNN is employed to replace the linear mapping, enabling the aggregation of features from neighboring nodes τ .

Following this, the hidden variable at the current moment functions as the output for this moment, calculated using the following formulas:

$$r_i^{\nu} = \sigma (\kappa_{r,i}^{\nu} + \xi_{r,i}^{\nu} + b_r), \quad (19)$$

$$z_i^{\nu} = \sigma (\kappa_{z,i}^{\nu} + \xi_{z,i}^{\nu} + b_z), \quad (20)$$

217

$$\tilde{h}_i^\nu = \phi(\kappa_{h,i}^\nu + r_i^\nu \odot \xi_{h,i}^\nu + b_h), \quad (21)$$

218

$$h_i^\nu = (1 - z_i^\nu) \odot \tilde{h}_i^\nu + z_i^\nu \odot h_{i-1}^\nu, \quad (22)$$

in these equations, the vectors b_r , b_z , and b_h correspond to additional bias terms. The symbol \odot signifies the Hadamard product, σ denotes the sigmoid function, and ϕ represents the hyperbolic tangent.

Presently, an array of GNNs has been developed extensively [30], finding applications across diverse domains and yielding commendable outcomes. Examples encompass GraphConv, graph convolutional network (GCN), graph attention network (GAT), and more. It has been evidenced that augmenting the central node v with additional characterization enhances the efficacy of GAT+ [23], surpassing the base GAT in training. Consequently, this paper opts for GAT+ as the GNN module in (17) and (18), with the node representation q updating as follows:

$$q'^\nu = b + W_1 q^\nu + \sum_{\tau \in \tilde{N}(\nu)} \tilde{\alpha}_{\nu,\tau} W_2 q^\tau, \quad (23)$$

where $\tilde{N}(\nu)$ is the inclusive neighborhood of a node v , $\tilde{\alpha}_{\nu,\tau}$ denotes the attention weights calculated as [31]. W_1 , W_2 , and b are learnable parameters.

In [23], when historical information from $t - t_h$ to t is given, node v and its neighbors at time t , $N_t(v)$, are chosen as the subject of investigation. The characteristics $\{\{q_i^\nu\}_{i=t-t_h}^t\}_{\nu \in \tilde{N}_t(v)}$ of the node set $\tilde{N}_t(v) = N_t(v) \cup \{v_t\}$ are subsequently employed as input for the network. However, in scenarios where the target is undetected, deceased, or has separated from the group structure depicted in Fig. 5, prediction accuracy may be compromised due to the lack of neighboring nodes. For instance, in scrutinizing target 1 as the subject of investigation, merely target 3 is identifiable as its neighboring target at time t . Nevertheless, in actuality, both target 2 and target 4 are concomitant members of the same group, exerting influence on the state of target 1 during the time interval spanning $t - t_h$ to t . This paper addresses this issue by introducing group structure estimation. By determining the subgroup to which node v belongs at each moment from $t - t_h$ to t , all identified neighboring nodes $\{\tilde{N}_i(\nu)\}_{i=t_h}^t$ can be considered. Moreover, the outcomes of group structure estimation at each instance can be directly utilized as input to the GAT+, as opposed to depending on the fully connected graph as illustrated in [23]. This strategy markedly alleviates the computational burden.

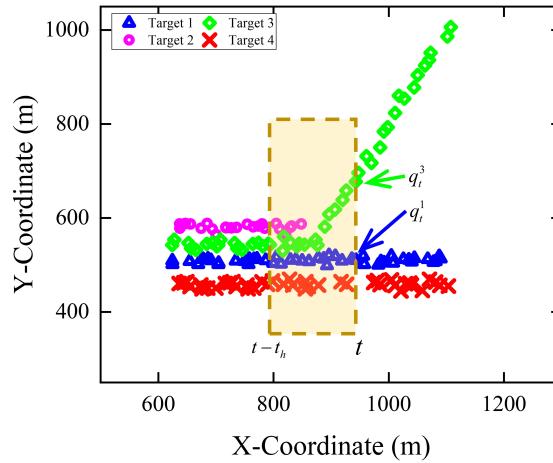


Fig. 5. An example showcasing the incomplete consideration of neighboring targets in the original method in [23] emphasizes the importance of the group structure estimation algorithm introduced in this paper.

(2) Decoding

To obtain the LMB density $\left\{ \left(r_{+,S}^{(\ell)}, p_{+,S}^{(\ell)} \right) \right\}_{\ell \in \mathbb{L}}$ for the surviving target, Task 1 of the decoder produces the survival probability of the target, denoted as $r_{+,S}^{(\ell)}$. For the computation of $p_{+,S}^{(\ell)}$, Task 2 of the decoder generates the 2-dimensional input term u and the noise covariance matrix Q in the target's motion model. The decoder's structure mirrors that of the encoder, and the hidden variable h can be directly inherited from the encoder. It's important to note that the input variable q requires additional computation using the attention mechanism [23].

In this paper, the application of the sliding window strategy is employed to predict the target's state at time $t + 1$ leveraging historical information from the preceding t_h temporal intervals. The utilization of the teacher-forcing strategy is embraced in this context. During the training phase, the actual group structure at time $t + 1$ is applied for decoding. However, in the filtering phase, the updated group structure at time t serves as the input to the decoder.

(3) Kinematics model

At present, prevailing motion models in RGTT tend to oversimplify the inherent dynamics of the target, relying on specific assumptions. Nonetheless, the motion model of a target can exhibit a myriad of patterns, encompassing unpredictable maneuvers. In this context, the application of the neural ODE [24] is embraced to learn the target's motion model unburdened by constraints, thereby augmenting the adaptability and versatility of the model within the authentic environment. An ODE can be formally expressed as $\frac{dx(t)}{dt} = f(x(t), \theta)$. The pivotal objective of the neural

1
2 ²⁶³ ODE lies in determining the optimal θ satisfying the above formula.
3
4 ²⁶⁴ In this study, a second-order model is employed to ensure the smoothness of the estimation
5
6 ²⁶⁵ [32], with the formulation presented as follows:
7

$$\begin{aligned} \dot{p}_x &= v_x \\ \dot{p}_y &= v_y \\ \dot{v}_x &= f_1(v_x, v_y, u_1) \\ \dot{v}_y &= f_2(v_x, v_y, u_2). \end{aligned} \tag{24}$$

16
17 ²⁶⁶ Hence, the STF can be effectively represented by a neural ODE. Through its integration with
18
19 ²⁶⁷ the output u and Q from the decoder, the comprehensive kinematics model described by (16) is
20
21 ²⁶⁸ achieved for prediction.

22 (4) Loss function

23
24 Illustrated in Fig. 4, the network output endeavors to align with a GM model that characterizes
25
26 the spatial distribution of the target. This model produces multi-modal probabilistic predictions,
27
28 harmonizing seamlessly with the GM-based LMB filter. Assuming each output consists of M
29
30 GCs, the result for the target ℓ is articulated as:

$$y_v = \left\{ \hat{w}_+^{(\ell,j)}, \mathcal{N}\left(\hat{x}_+^{(\ell,j)}, \hat{P}_+^{(\ell,j)}\right) \right\}_{j=1}^M, \tag{25}$$

31
32 ²⁷⁴ where, $\hat{w}_+^{(\ell,j)}$ signifies the predicted weight of the j -th GC, and $\mathcal{N}\left(\hat{x}_+^{(\ell,j)}, \hat{P}_+^{(\ell,j)}\right)$ represents the
33
34 Gaussian distribution denoted by the predicted mean and covariance. Consequently, the state
35
36 prediction loss for target ℓ can be formulated using the negative log-likelihood (NLL) loss.
37
38

$$\mathcal{L}_{\text{NLL}}^\ell = -\log \left(\sum_j^M \hat{w}_+^{(\ell,j)} \mathcal{N}\left(x^\ell \mid \hat{x}_+^{(\ell,j)}, \hat{P}_+^{(\ell,j)}\right) \right), \tag{26}$$

39
40 ²⁷⁷ where x^ℓ is the true state of target ℓ .
41

42
43 ²⁷⁸ The neural network also produces the predicted existence probability, denoted as $\hat{r}_+^{(\ell)}$, for the
44
45 survival target. Predicting $\hat{r}_+^{(\ell)}$ is approached as a binary classification task, and the binary cross-
46
47 entropy (BCE) loss is employed to compute the existence probability prediction loss for target
48
49 ℓ :
50

$$\mathcal{L}_{\text{BCE}}^\ell = -\delta(\ell) \log(\hat{r}_+^{(\ell)}) - (1 - \delta(\ell)) \log(1 - \hat{r}_+^{(\ell)}), \tag{27}$$

51
52 ²⁸² where $\delta(\ell)$ is an indicator of the existence of target ℓ , taking the value '1' if the target ℓ exists
53
54 ²⁸³ and '0' otherwise.
55
56
57
58
59
60

The overall regression loss and classification loss are presented as the sum of each target's NLL loss and BCE loss, respectively. The ultimate loss is derived by the weighted combination of these two losses:

$$\mathcal{L} = \alpha \sum_{\ell \in \mathbb{L}} \mathcal{L}_{\text{NLL}}^{\ell} + (1 - \alpha) \sum_{\ell \in \mathbb{L}} \mathcal{L}_{\text{BCE}}^{\ell}, \quad (28)$$

where \mathbb{L} is the label space of survival targets, and $0 < \alpha < 1$ is the loss weight.

C. GM Implementation of the GNN-LMB Filter

In this subsection, the complete GM implementation of the proposed approach is presented. This approach combines the temporal GNN and the LMB filter, referred to as the GNN-LMB filter. Considering that the extended kalman filter (EKF) exhibits the capability to handle nonlinear problems compared to the kalman filter (KF), and its calculations and implementation are relatively straightforward compared to the unscented kalman filter (UKF), this paper chooses to implement it based on EKF [33]. The posterior probability densities $p^{(\ell)}$ of all labeled Bernoulli tracks $\ell \in \mathbb{L}$ are delineated as follows:

$$p^{(\ell)}(x) = \sum_{j=1}^{J^{(\ell)}} w^{(\ell,j)} \mathcal{N} \left(x; \hat{x}^{(\ell,j)}, \hat{P}^{(\ell,j)} \right), \quad (29)$$

where $J^{(\ell)}$ represents the number of GC for the target labeled ℓ , while $w^{(\ell,j)}$ denotes the weight of each GC. Additionally, $\hat{x}^{(\ell,j)}$ is the estimated mean value of each GC, and $\hat{P}^{(\ell,j)}$ represents the corresponding estimation error covariance. The subsequent expressions with $\mathcal{N}(\cdot)$ have a similar meaning.

Prediction: For the survival target, $(r_{+,S}^{(\ell)}, p_{+,S}^{(\ell)})$ are directly generated by the neural network, circumventing the need for computation as outlined in equations (5)-(7). The formulation of $p_{+,S}^{(\ell)}$ is as follows:

$$p_{+,S}^{(\ell)}(x) = \sum_{j=1}^{J_M} \hat{w}_{+,S}^{(\ell,j)} \mathcal{N} \left(x; \hat{x}_{+,S}^{(\ell,j)}, \hat{P}_{+,S}^{(\ell,j)} \right), \quad (30)$$

where J_M represents the number of GC. Analogous to the formulation for survival targets, denote the newborn LMB RFS as $\left\{ (r_B^{(\ell)}, p_B^{(\ell)}) \right\}_{\ell \in \mathbb{B}}$, and presume that $p_B^{(\ell)}$ comprises J_B GCs:

$$p_B^{(\ell)} = \sum_{j=1}^{J_B} w_B^{(\ell,j)} \mathcal{N} \left(x; \hat{x}_B^{(\ell,j)}, \hat{P}_B^{(\ell,j)} \right). \quad (31)$$

As indicated in (4), the predicted LMB density π_+ is derived by amalgamating the information pertaining to survival and newborn targets.

Update: The aggregate of weights attributed to each GC corresponds to the probability of existence, denoted as $r^{(\ell)}$. Thus, the emphasis lies in resolving the updated spatial distribution $p^{(\theta)}(x, \ell | Z)$. Assuming a consistent detection probability P_D , irrespective of the position x , the predicted spatial distribution of track ℓ is uniformly articulated as follows, , regardless of whether it's a surviving target or a newborn one:

$$p_+^{(\ell)}(x) = \sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \mathcal{N}\left(x; \hat{x}_+^{(\ell,j)}, \hat{P}_+^{(\ell,j)}\right). \quad (32)$$

where $J_+^{(\ell)}$ represents the number of GC. Establish the nonlinear measurement equation as follows:

$$z = h(x) + v. \quad (33)$$

Define H as the linearized matrix of the measurement function $h(\cdot)$, and denote the measurement noise v as $v \sim \mathcal{N}(0, R)$.

When $\theta(\ell) \neq 0$, associate the measurement $z_{\theta(\ell)}$ with track ℓ , and the GM implementation is as follows:

$$p^{(\theta)}(x, \ell | Z) = \sum_{j=1}^{J_+^{(\ell)}} w^{(\ell,j,\theta)}(Z) \mathcal{N}\left(x; \hat{x}^{(\ell,j,\theta)}, \hat{P}^{(\ell,j)}\right), \quad (34)$$

$$w^{(\ell,j,\theta)}(Z) = \frac{\frac{1}{\kappa(z_{\theta(\ell)})} P_D w_+^{(\ell,j)} \mathcal{N}\left(z_{\theta(\ell)}; z_+^{(\ell,j)}, S^{(\ell,j)}\right)}{\eta_Z^{(\theta)}(\ell)}, \quad (35)$$

$$\eta_Z^{(\theta)}(\ell) = \frac{P_D}{\kappa(z_{\theta(\ell)})} \sum_{j=1}^{J_+^{(\ell)}} w_+^{(\ell,j)} \mathcal{N}\left(z_{\theta(\ell)}; z_+^{(\ell,j)}, S^{(\ell,j)}\right), \quad (36)$$

$$S^{(\ell,j)} = H \hat{P}_+^{(\ell,j)} H^T + R, \quad (37)$$

$$z_+^{(\ell,j)} = H \hat{x}_+^{(\ell,j)}, \quad (38)$$

$$\hat{x}^{(\ell,j,\theta)}(Z) = \hat{x}_+^{(\ell,j)} + K^{(\ell,j)} \left(z_{\theta(\ell)} - z_+^{(\ell,j)} \right), \quad (39)$$

322

$$K^{(\ell,j)} = \hat{P}_+^{(\ell,j)} H^T [S^{(\ell,j)}]^{-1}, \quad (40)$$

323

$$\hat{P}^{(\ell,j)} = \hat{P}_+^{(\ell,j)} - K^{(\ell,j)} S^{(\ell,j)} [K^{(\ell,j)}]^T. \quad (41)$$

When $\theta(\ell) = 0$, no measurements are associated with the trajectory ℓ , and the predicted spatial distribution is utilized instead of the posterior distribution. In this case,

$$\eta_Z^{(\theta)}(\ell) = q_D = 1 - P_D. \quad (42)$$

IV. NUMERICAL SIMULATION

A. Simulation Scenario and Comparison Approaches

In this subsection, we conduct a simulation experiment to assess the tracking performance of the proposed GNN-LMB filter. The evaluation criterion utilized is the optimal sub-pattern assignment (OSPA) distance [34] and the OSPA⁽²⁾ distance [35] (with window $L = 5$), employing an order parameter of $p = 1$ and a cut-off parameter of $c = 100$. The experimental scenario is set in two dimensions within the region $[-1500, 1500] \text{ m} \times [0, 3000] \text{ m}$. The sensor is positioned at the coordinates (0m,0m) and surveils the entire area with a data sampling frequency of 1 Hz. The monitoring time is 100s.

For all LMB filters expounded in this paper, a standard set of parameters is shared. The birth process is characterized by an LMB RFS with the parameter ensemble $\pi_B = \left\{ \left(r_B^{(\ell)}, p_B^{(\ell)} \right) \right\}_{\ell \in \mathbb{L}}$, where $r_B^{(\ell)} = 0.1$ and $p_B^{(\ell)} = \mathcal{N}(x; x_B^{(\ell)}, P_B)$, with $P_B = \text{diag}([50^2, 30^2, 50^2, 30^2]^\top)$, and $x_B^{(\ell)} = [p_x^{(\ell)}, v_x^{(\ell)}, p_y^{(\ell)}, v_y^{(\ell)}]$ configured in accordance with the initial state of the target in distinct scenarios. Furthermore, the nonlinear measurement model is employed and expressed as:

$$z = \begin{bmatrix} \text{atan2}(p_x - p_x^s, p_y - p_y^s) \\ \sqrt{(p_x - p_x^s)^2 + (p_y - p_y^s)^2} \end{bmatrix} + \epsilon, \quad (43)$$

where atan2 designates the 4-quadrant inverse tangent function, (p_x^s, p_y^s) represents the known sensor position, and the covariance of the measurement noise is

$$R = \text{diag}([\sigma_\theta^2, \sigma_r^2]^\top), \quad (44)$$

with $\sigma_\theta = (\pi/180) \text{ rad}$ and $\sigma_r = 5 \text{ m}$. The remaining parameters involved in the simulation are presented in Table II.

TABLE II

SIMULATION PARAMETERS AND THEIR SYMBOLS AND VALUES

Simulation Parameter	Symbol	Value
Detection probability	P_D	0.90
Clutter rate	λ_c	20
Sensor resolution in range	a_r	15m
Sensor resolution in azimuth	a_φ	2°
Sensor signal-to-noise ratio	SNR	10dB
False alarm probability	P_{FA}	10^{-3}
Maximum number of GCs	N_{GC}	15
Merging threshold	ξ_m	4
Pruning threshold	ξ_p	10^{-4}
Extraction threshold	ξ_e	0.5

TABLE III

THE PARAMETERS GOVERNING THE KINEMATICS OF THE TARGET WITHIN THE DATASET

Parameters	Interval range
The acceleration of CA model	$-0.5 \sim 0.5 \text{ m/s}^2$
The turn rate of CT model	$-4 \sim 4^\circ/\text{s}$
The matrix coefficient σ^2 of (45)	$1 \sim 2 \text{ m/s}^2$
The birth time of target	$1 \sim 50\text{s}$
The death time of target	$60 \sim 100\text{s}$

For the GNN module, the maximum duration of the observation window, corresponding to the model input, is stipulated as 15s. The aim is to predict the state of the surviving target at a subsequent moment, indicating an output duration of 1s. Three distinct motion models govern the system: Constant Velocity (CV), Constant Acceleration (CA), and Constant Turn Rate (CT). The stochastic interference in the state transition process is modeled as Gaussian white noise, denoted as $w \sim \mathcal{N}(0, Q)$, with the covariance matrix given by:

$$Q = \sigma^2 \begin{bmatrix} \frac{T^4}{4} & \frac{T^3}{2} & 0 & 0 \\ \frac{T^3}{2} & T^2 & 0 & 0 \\ 0 & 0 & \frac{T^4}{4} & \frac{T^3}{2} \\ 0 & 0 & \frac{T^3}{2} & T^2 \end{bmatrix}, \quad (45)$$

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60 where $T = 1s$ denotes the time interval. The parameters governing the motion models and the birth/death times of each target follow a uniform distribution, as detailed in Table III. The initial position of a new target manifests randomly within the detection range, and the initial velocity ranges between -40m/s and 40m/s . The count of transitions between motion modes throughout the target's entire existence adheres to a Poisson distribution with parameter $\lambda = 3$. The occurrences of mode transitions transpire randomly during the survival period. Following a transition, one of CV, CA, or CT is randomly selected. Following the aforementioned procedure, diverse RGT trajectories are generated, encompassing stochastic perturbations, various target motion models, and diverse evolution laws for group structures. The sequences of graph-structured data are derived by segmenting the generated RGT trajectory and are utilized as the dataset. The total size of the dataset is 160,000, with 80% allocated to the training set and the remaining 20% designated for the validation set. The neural network was built and trained using PyTorch [36], utilizing the Adam optimizer with a learning rate of 0.0001. The training process was carried out in batches of 64 samples, spanning a total of 200 epochs. The detailed architecture of the network depicted in Fig. 4 is outlined in Table IV. Hyperparameters were meticulously fine-tuned through grid search, conducted independently for each experiment, as detailed in [23].

32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60 The GNN-LMB filter proposed in this paper is contrasted with the standard LMB filter and the prevalent SDE-LMB filter, both of which lack a learnable component. The SDE-LMB filter [1] is introduced to capture the collective motion of RGT using SDE. Additionally, to emphasize the robust capability of GNN in capturing node interactions, it is compared with an intelligent approach utilizing CNN, referred to as the CNN-LMB filter. This approach, inspired by the concept outlined in [18], replaces the central module of GNN-LMB with convolutional social pooling layers, refining the concept of social pooling layers [39]. To ensure the rigor of the experimental design, the EGN is also incorporated into both the SDE-LMB filter and CNN-LMB filter. The comparative analysis is presented in subsection IV-B. Furthermore, within the realm of RGTT, maintaining the consistency of target labels is of paramount importance. Elaborate experiments and discussions concerning this aspect are outlined in subsection IV-C. To further scrutinize the contributions of individual system components to the overall performance, an ablation study is expounded upon in subsection IV-D. Furthermore, all simulation results are averaged over 100 independent Monte Carlo (MC) runs.

TABLE IV
COMPONENTS AND DIMENSIONS OF THE NEURAL NETWORK MODULE

Module	Component	Detail description
Encoder	GRU	With a hidden layer dimension of 64.
	GAT+	Comprising a fully connected (FC) layer and a single-headed GAT layer implemented by GATv2Conv module in PyTorch Geometric [37], both with an output dimension of 192.
Decoder	Attention	Comprising two FC layers, with output dimensions 64 and $t_h + 1$.
	FC layer used to calculate \hat{q}_{t+1}	With an output dimension of 64.
Existence probability	FC layer used to calculate $\hat{r}_+^{(\ell)}$	With an output dimension of 1.
Spatial distribution	FC layer used to calculate $u_{t+1}^{(\ell,j)}$	Solving u_1 and u_2 in (24), with an output dimension of $2 * N_{GC}$.
	FC layer used to calculate $Q_{t+1}^{(\ell,j)}$	Solving σ in (45), with an output dimension of N_{GC} .
	FC layer used to calculate $\hat{w}_+^{(\ell,j)}$	With an output dimension of N_{GC} .
Neural ODE	Neural ODE	Comprising two groups of FC layers, each with two layers, approximating f_1 and f_2 in (24), with output dimensions of 16 and 1.
	EKF	Computing the Jacobian matrix of f_1 and f_2 output by the Neural ODE module, using functorch [38].

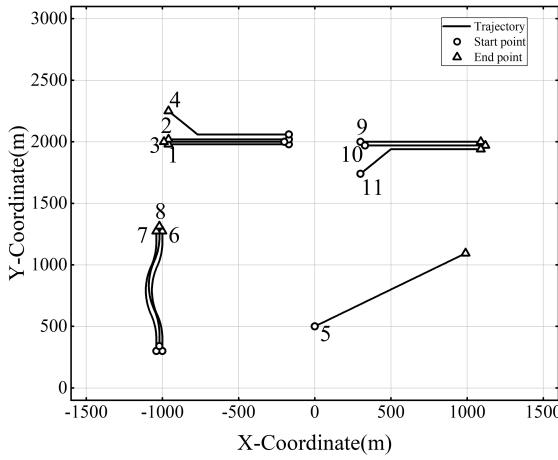


Fig. 6. The target trajectories used in Scenario 1.

381 B. Tracking Performance of GNN-LMB Filter

382 (1) Scenario 1

383 To facilitate a clear comparison of the performance of the proposed GNN-LMB filter, we
 384 initially consider a straightforward scenario involving independent moving targets, group targets

TABLE V

THE INITIAL STATE AND THE BIRTH/DEATH TIME (SCENARIO 1)

Target ID	Initial State	Birth time (s)	Death time (s)
1	[-170, -10, 1980, 0]	1	80
2	[-170, -10, 2020, 0]	1	80
3	[-200, -10, 2000, 0]	1	80
4	[-170, -10, 2060, 0]	1	80
5	[0, 10, 500, 6]	1	100
6	[-1000, 0, 300, 10]	1	100
7	[-1040, 0, 300, 10]	1	100
8	[-1020, 0, 340, 10]	1	100
9	[300, 10, 2000, 0]	21	100
10	[330, 10, 1970, 0]	21	100
11	[300, 10, 1740, 10]	21	100

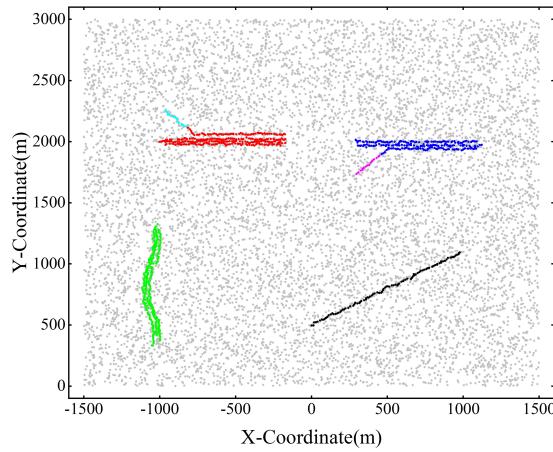
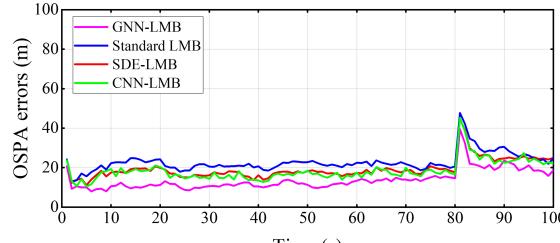


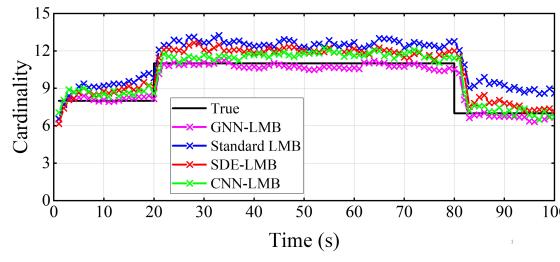
Fig. 7. The tracking outcome achieved by the GNN-LMB filter in Scenario 1. The gray dots symbolize clutter, whereas the colored dots illustrate the inferred trajectory of the target.

splitting and merging, where the motion models of each target are relatively simple. The kinematic parameters of the targets are detailed in Table V, and the corresponding target trajectory is illustrated in Fig. 6. In Fig. 7, the tracking results of the GNN-LMB filter in a single MC run are presented. Different group structures are distinguished by color, providing a visual demonstration of the GNN-LMB filter's tracking capability for RGT.

Fig. 8(a) and Fig. 8(b) showcase the tracking performance and target cardinality estimation



(a)



(b)

Fig. 8. The tracking performance of each filter in Scenario 1. (a) The OSPA error; (b) The target cardinality estimation.

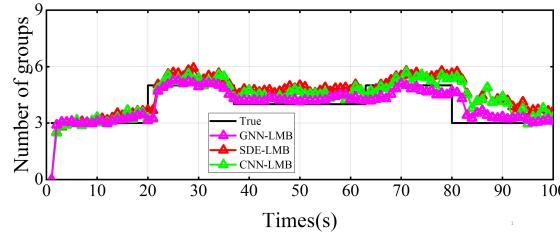


Fig. 9. The group number estimation results in Scenario 1.

of the GNN-LMB filter in comparison to other filters described in subsection IV-A. As observed in Fig. 8(a) and Fig. 8(b), in this uncomplicated scenario, the tracking performance of the SDE-LMB filter, CNN-LMB filter, and GNN-LMB filter, which incorporate group structure information, surpasses that of the standard LMB filter. Notably, the proposed GNN-LMB filter, capable of learning the interaction between RGT more flexibly, exhibits the most outstanding performance.

Fig. 9 illustrates the group number estimation results of the three filters considering the group structure. While there is a slight delay in the estimation results when the group undergoes splits or merges, overall, the change in the number of groups can be effectively estimated, with the GNN-LMB filter demonstrating the most effective results.

TABLE VI

THE INITIAL STATE AND THE BIRTH/DEATH TIME (SCENARIO 2)

Target ID	Initial State	Birth time (s)	Death time (s)
1	[530, -20, 1880, 0]	1	80
2	[530, -20, 1920, 0]	1	80
3	[500, -20, 1900, 0]	1	80
4	[530, -20, 1960, 0]	1	80
5	[-220, 0, 820, 30]	1	100
6	[-200, 0, 800, 30]	1	100
7	[-240, 0, 800, 30]	1	100
8	[-220, 0, 840, 30]	1	100
9	[20, 30, 500, -10]	21	100
10	[50, 30, 470, 0]	21	100
11	[20, 30, 530, 10]	21	100

(2) Scenario 2

To comprehensively assess the efficacy of the proposed algorithm, we devised more intricate simulation scenarios. In this particular setting, the target motion model remains elusive, and significant maneuvers are executed during motion. The trajectory of the target is delineated in Fig. 10, and the specific kinematic parameters are detailed in Table VI.

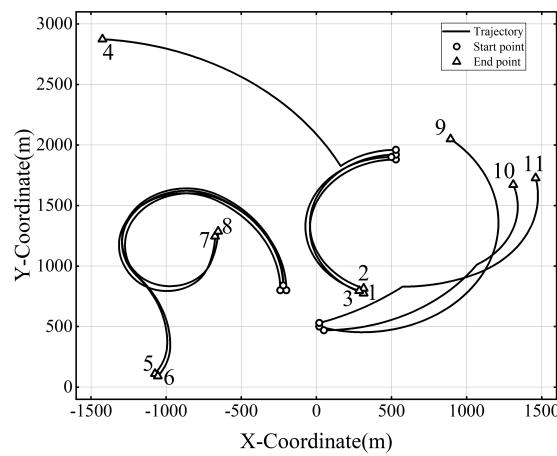
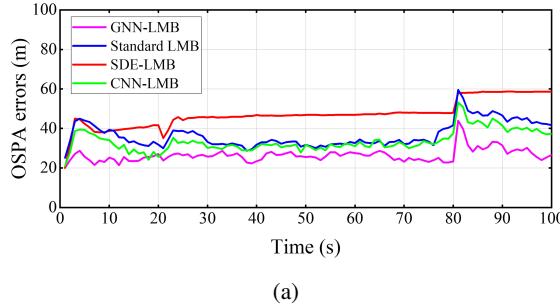
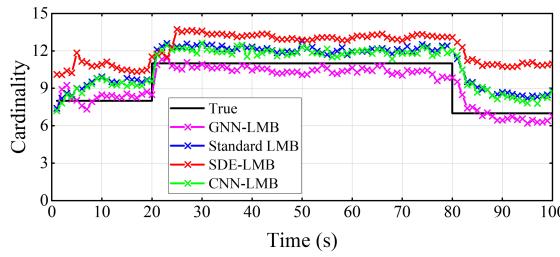


Fig. 10. The target trajectories used in Scenario 2.

Comparisons of the OSPA error and target number estimations for the GNN-LMB filter and alternative methods are presented in Fig. 11(a) and Fig. 11(b), respectively. It is discernible that



(a)



(b)

Fig. 11. The tracking performance of each filter in Scenario 2. (a) The OSPA error; (b) The target cardinality estimation.

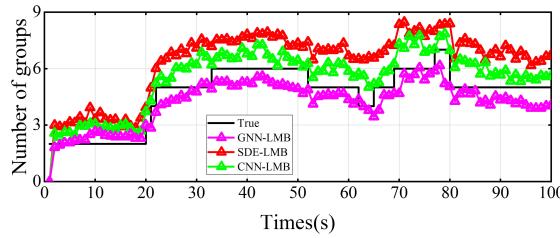


Fig. 12. The group number estimation results in Scenario 2.

the tracking precision of each approach diminishes in the presence of a highly maneuvering target without prior motion model information in the scene. Notably, the SDE-LMB filter, modeling interactions among targets in the same group as both attraction and repulsion, fails to accurately portray the intricate influences between targets. Surprisingly, its tracking performance deteriorates even more than the traditional standard LMB filter, which does not account for the group structure. In contrast, both the CNN-LMB and GNN-LMB filters, equipped with learnable components, exhibit a superior ability to maintain tracking performance. When comparing the two, the GNN-LMB filter outperforms due to its effective modeling of complex relationships between nodes and greater flexibility in processing graph data with irregular structures.

The estimated results for the number of groups are depicted in Fig. 12. It becomes evident

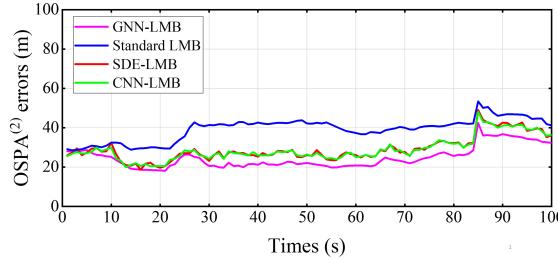


Fig. 13. The OSPA⁽²⁾ errors of each filter in Scenario 1.

418 that group merging and splitting transpire more frequently in this challenging scenario, under-
 419 scoring the inherent difficulty in tracking RGT. Among the three approaches, the GNN-LMB
 420 filter demonstrates the most effective performance, while the SDE-LMB filter exhibits the least
 421 efficacy, consistent with the findings in Fig. 11.

422 C. Label Consistency Performance of GNN-LMB Filter

423 In this subsection, we present **Scenario 1** from subsection IV-B as an illustrative case to
 424 evaluate the effectiveness of the proposed filter in maintaining label consistency. We employed the
 425 OSPA⁽²⁾ distance [35] as the evaluation criterion, and the results are depicted in Fig. 13. Notably,
 426 the influence of target births or deaths on OSPA⁽²⁾ distance persists due to the presence of the
 427 sliding window, resulting in a smoother curve evolution in Fig. 13 compared to Fig. 8. The GNN-
 428 LMB filter demonstrates superior performance by accurately capturing interactions among RGT,
 429 thereby facilitating more precise prediction of states for each target. This mitigates the challenge
 430 of associating data between dense targets and corresponding measurements, thereby reducing
 431 the frequent switching of RGT labels. Conversely, the standard LMB filter inherently tends
 432 to produce fragmented trajectories with inconsistent labels, particularly evident when handling
 433 targets in close proximity, leading to the poorest performance among the four filters.

434 Next, the clutter rate λ_c is incrementally increased every 20 intervals from 20 to 80, and
 435 the OSPA⁽²⁾ errors averaged over time for each filter are compared, as depicted in Table VII.
 436 The results indicate that while the GNN-LMB filter consistently outperforms the other filters,
 437 its relative performance improvement compared to the standard LMB filter diminishes with
 438 increasing clutter. For instance, there is a 36.76% improvement when $\lambda_c = 20$, contrasted with
 439 only a 16.32% improvement when $\lambda_c = 80$. This implies that as the clutter rate increases, the

TABLE VII
THE AVERAGE OSPA⁽²⁾ ERRORS (m) FOR DIFFERENT CLUTTER RATE

	λ_c	20	40	60	80
Average OSPA ⁽²⁾	Standard LMB	39.23	40.86	44.18	48.96
	SDE-LMB	29.08	31.25	36.54	42.72
	CNN-LMB	28.96	31.13	36.65	42.54
	GNN-LMB	24.81	26.66	32.08	40.97

440 proposed GNN-LMB filter encounters challenges in effectively addressing label switching in
 441 RGTT.

442 It is noteworthy that recent studies have applied hypergraph matching techniques to RGTT data
 443 association, effectively reducing label switching [40] [41]. Indeed, GNN has been demonstrated
 444 to efficiently address the challenge of node matching in hypergraph data [42]. Therefore, besides
 445 employing the GNN-GRU module to learn interactions among RGT as proposed in this paper,
 446 exploring its potential in mitigating label switching induced by intricate RGTT data association
 447 presents a promising avenue for future research.

448 *D. Ablation Study*

449 The ablation study endeavors to elucidate the most pivotal elements contributing to the system's
 450 efficacy, thus guiding its refinement and optimization. [23] conducted ablation experiments on
 451 each constituent of the neural network employed in this paper, elucidating that the elimination
 452 of EKF and neural ODE components significantly compromises performance. Moreover, in the
 453 comparative assessment of the encoder and decoder, it was noted that integrating GNN in the
 454 latter yielded more discernible enhancements in performance than in the former. Additionally,
 455 [23] juxtaposed the impact of various GNN models on overall performance, while [32] scrutinized
 456 the influence of diverse motion models and numerical solvers.

457 In this paper, the model in [23] undergoes refinement and adaptation to accommodate the
 458 RGTT problem, introducing two innovative components: a component for predicting the existence
 459 probability of LMB density and an EGN component. Furthermore, the exploration of implement-
 460 ing the LMB filter with GM model is paramount for practical applications. However, in [23],
 461 there's no explicit assessment of how a GM model influences system performance. Therefore,
 462 the ablation study of the three aforementioned components is conducted in this subsection.

TABLE VIII

THE INITIAL STATE AND THE BIRTH/DEATH TIME (SCENARIO 3)

Target ID	Initial State	Birth time (s)	Death time (s)
1	[300, 10, 1100, 0]	1	80
2	[270, 10, 1130, 0]	1	80
3	[300, 10, 1050, 0]	1	80
4	[700, 10, 1150, 0]	41	80
5	[-470, 10, 2000, 0]	21	100
6	[-500, 10, 1970, 0]	21	100
7	[-500, 10, 2600, -30]	21	100
8	[-800, 0, 300, 20]	21	100
9	[-760, 0, 300, 20]	21	100
10	[-780, 0, 340, 20]	21	100

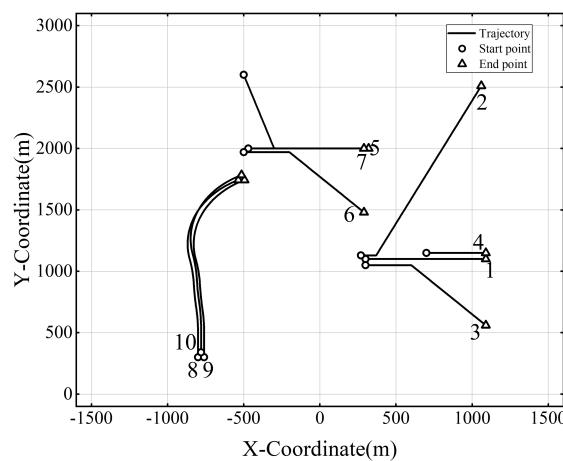


Fig. 14. The target trajectories used in Scenario 3.

To simulate a more intricate group structure evolution, a scenario involving target birth, merging, and separation within the same group (referred to as *Scenario 3* thereafter) is established, as illustrated in Fig. 14. The kinematic parameters are detailed in Table VIII. The ablation study considers *Scenario 1* (from subsection IV-B) and *Scenario 3* as illustrative examples, utilizing OSPA and OSPA⁽²⁾ as criteria for performance evaluation. The results are depicted in Table IX, where checkmarks and crosses indicate whether the component is included in the network.

In the experiment where the existence probability prediction component is excluded, setting $P_s^{(\ell)} = 0.99$, and then solving for $r_{+,S}^{(\ell)}$ with (5) and (7). It is observed that the absence of this

TABLE IX

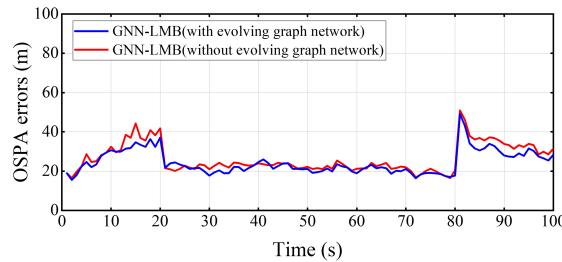
THE RESULT OF ABLATION STUDY

Environment	Index	Existence probability	EGN	GM model	Average OSPA	Average OSPA ⁽²⁾
Scenario 1	I ₁₁	✓	✓	✓	13.80	24.81
	I ₁₂	✗	✓	✓	14.61	25.72
	I ₁₃	✓	✗	✓	13.96	25.02
	I ₁₄	✗	✗	✓	14.71	25.93
	I ₁₅	✓	✓	✗	20.42	33.62
Scenario 3	I ₃₁	✓	✓	✓	24.44	34.27
	I ₃₂	✗	✓	✓	25.23	34.89
	I ₃₃	✓	✗	✓	26.38	37.26
	I ₃₄	✗	✗	✓	28.25	39.73
	I ₃₅	✓	✓	✗	30.06	42.51

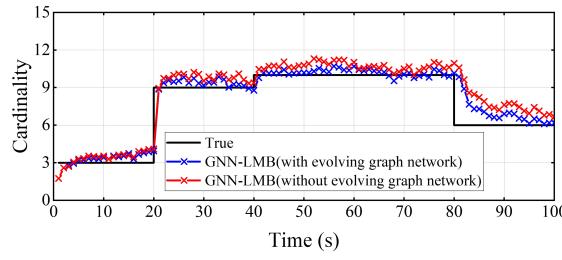
component does indeed affect the overall performance to a certain extent (see I₁₂ or I₃₂). This is due to the fact that the value of 0.99 is empirically determined and cannot be dynamically adjusted according to the movement of the target. Consequently, this fixed and relatively high value may lead to an overestimation of the number of targets during tracking and a delay in response when targets disappear, thereby adversely impacting tracking performance.

The exclusion of the EGN in **Scenario 3** has a more significant impact on performance compared to **Scenario 1**, owing to the more intricate changes in group structure in **Scenario 3** (see I₁₁ vs I₁₃ and I₃₁ vs I₃₃). The primary distinction between a GNN-LMB filter without the EGN and one incorporating it resides in the input to the encoder. In the former, the initial step entails identifying the central target v and its neighboring nodes, whose Mahalanobis distance from v is less than the threshold at time t . Subsequently, the sequence of fully connected graph structures for these targets is traced back from time $t - t_h$ to t , forming the input for the neural network encoder. In contrast, the latter employs the EGN to dynamically estimate the group structure at each moment, directly utilizing the group structure from time $t - t_h$ to t as its input. With the EGN, all targets in the scene are simultaneously predicted, eliminating the need for the concept of a center target. The OSPA error and target number estimation of I₃₁ and I₃₃ are elaborated upon in Fig. 15. The results underscore that the incorporation of the EGN enhances the accuracy of group structure estimation, preventing the oversight of valuable neighbor nodes. This, consequently, positively influences the overall enhancement in tracking performance and

490 bolsters the adaptability of the model to complex environments.



(a)



(b)

Fig. 15. The tracking performance of two GNN-LMB filters. (a) The OSPA error; (b) The target cardinality estimation.

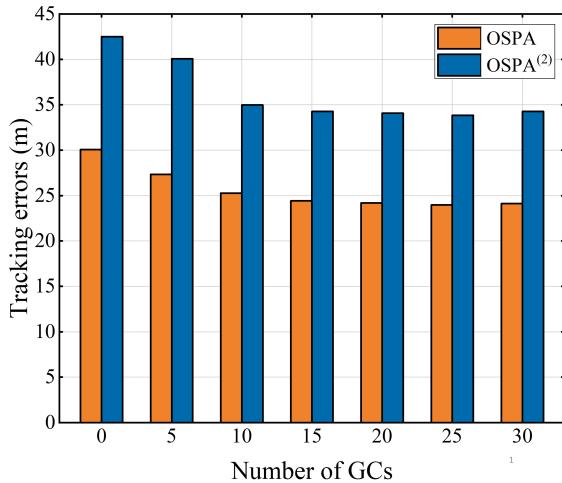


Fig. 16. Tracking errors under different N_{GC} in Scenario 3.

491 In I₁₅ and I₃₅, the GM model is omitted, i.e., $N_{GC} = 0$. The absence of GM model results in a
 492 significant loss of tracking performance (see I₁₅ or I₃₅). This underscores the importance of GM
 493 model, which effectively captures data uncertainty, enhances the model's capacity to fit complex
 494 data distributions, and improves overall model expression and generalization capabilities. More-
 495 over, the comparative analysis of tracking performance across various N_{GC} values is depicted in

496 Fig. 16 (using **Scenario 3** as an example). The findings suggest that augmenting N_{GC} within a
497 delimited scope enhances tracking performance. However, increasing model intricacy amplifies
498 the risk of overfitting, and the proliferation of parameters complicates model training, potentially
499 leading to sluggish or even declining performance enhancements.

V. CONCLUSION

501 In this paper, we introduce an LMB filter augmented with a deep neural network tailored for
502 RGTT. This innovative approach transcends the constraints imposed by imperfect assumptions
503 about target motion models and interactions among group members inherent in traditional RGTT
504 methodologies. Our methodology incorporates a temporal GNN to assimilate interaction infor-
505 mation among group members, and a neural ODE to articulate the motion model of the targets.
506 Additionally, we integrate an evolving network to dynamically deduce the real-time structure of
507 the target group. Through simulation experiments, we substantiate the efficacy of the proposed
508 filter in achieving superior tracking performance and showcase the augmented capabilities of
509 the evolving network. In forthcoming endeavors, our objective is to deepen the integration of
510 the neural network component with the LMB filter, surpassing its current embedding solely in
511 the prediction step. This is pivotal as the interaction between group targets also influences the
512 association between measurements and trajectories.

APPENDIX A

THE COMPLETE PSEUDO-CODE OF GROUP STRUCTURE ESTIMATION.

Algorithm 1 Group target initiation

- ```

516 1: INPUT: Vertices set $V = \{v_1, \dots, v_n\} = \{(x_1, P_1, \ell_1), \dots, (x_n, P_n, \ell_n)\}$, preset threshold ε .
517 2: OUTPUT: Group structure $G = \{g_1, g_2, \dots, g_m\}$.
518 3: function GROUP-INITIATION(V, ε)
519 4: · Set $E = \emptyset$;
520 5: for $i = 1 : n$ do
521 6: for $j = i + 1 : n$ do
522 7: · $d_{ij} = \sqrt{(x_i - x_j)(P_i + P_j)^{-1}(x_i - x_j)^T}$;
523 8: if $d_{ij} < \varepsilon$ then
524 9: · $E = E \cup \{(i, j)\}$;
525 10: · Identify the connected blocks of the graph based on E , forming the subgraph $G = \{g_1, g_2, \dots, g_m\}$;

```

1  
2  
3 530 11: **return:**  $G = \{g_1, g_2, \dots, g_m\}$

---

4  
5 531 **Algorithm 2** Construct the second-layer group vertices

---

6  
7 532 1: **INPUT:** Vertices set  $V = \{v_1, \dots, v_n\} = \{(x_1, P_1, \ell_1), \dots, (x_n, P_n, \ell_n)\}$ , group structure  
8  
9  $G = \{g_1, g_2, \dots, g_m\}$ .  
10  
11 533 2: **OUTPUT:** Vertices set of the second-layer group structure  $V^g = \{v_1^g, \dots, v_m^g\}$ .  
12  
13 534 3: **function** CONSTRUCT-SECOND-VERTICES( $V, G$ )  
14  
15 535 4: **for**  $i = 1 : m$  **do**  
16  
17 536 5:     · Set  $n_i^g$  denotes the number of vertices in subgroup  $g_i$ ;  
18  
19 537 6:     ·  $x_i^g = \frac{1}{n_i^g} \sum_{v_j \in g_i} x_j$ ;  
20  
21 538 7:     ·  $P_i^g = \frac{1}{n_i^g} \sum_{v_j \in g_i} P_j$ ;  
22  
23 539 8:     ·  $v_i^g = (x_i^g, P_i^g)$ ;  
24  
25 540 9: **return:**  $V^g = \{v_1^g, \dots, v_m^g\}$

---

26  
27 543 **Algorithm 3** Dynamic graph structure estimation

---

28  
29 544 1: **INPUT:** Group structure at moment  $k - 1$  denote as  $G_{k-1} = \{g_{k-1,1}, g_{k-1,2}, \dots, g_{k-1,m}\}$ ,  
30 LMB update density  $\pi_k$ .  
31  
32 545 2: **OUTPUT:** Group structure at moment  $k$  denote as  $G_k = \{g_{k,1}, g_{k,2}, \dots, g_{k,z}\}$ .  
33  
34 546 3: **function** GROUP-ESTIMATION( $G_{k-1}, \pi_k$ )  
35  
36 547 4: 1) *Group target update :*  
37  
38 548 5:     · Update state of identical targets within each subgroup in accordance with  $\pi_k$ , obtaining  
39      $G'_k = \{g'_{k,1}, \dots, g'_{k,m}\}$ ;  
40  
41 549 6:     · For those targets not subjected to updates, generate a virtual vertex based on their  
42     anticipated states;  
43  
44 550 7:     · Set  $V'_{k,i}$  denote the vertices set of subgroup  $g'_{k,i}$  and  $V'_k$  is the union set of all  $V'_{k,i}$ ;  
45  
46 551 8: 2) *Group target split :*  
47  
48 552 9:     · Set  $G_k^{sp} = \emptyset$ ;  
49  
50 553 10:     **for**  $i = 1 : m$  **do**  
51  
52 554 11:         ·  $g_{k,i}^{new} = \text{GROUP-INITIATION}(V'_{k,i}, \varepsilon)$ ;  
53  
54 555 12:         · Retain the original group ID without alteration and append the split-generated  
55         segment to the set  $G_k^{sp}$ ;  
56  
57 556 13:         ·  $G'_k = G'_k \cup G_k^{sp} = \{g'_{k,1}, \dots, g'_{k,p}\}$ ;  
58  
59 558 14: 3) *Group target merge :*

```

1
2 559 15: · Find the second-layer group vertices:
3
4
5 $V_k^g = \{v_{k,1}^g, \dots, v_{k,p}^g\}$
6
7 = CONSTRUCT-SECOND-VERTICES(V'_k, G'_k);
8
9 560 16: · Initialize the second-layer group structure:
10
11 $G_k^{2nd} = \text{GROUP-INITIATION}(V_k^g, \varepsilon_m)$
12
13 = $\{g_{k,1}^{2nd}, \dots, g_{k,q}^{2nd}\}$;
14
15 561 17: · Set $G_k^{mer} = \emptyset$;
16
17 562 18: for $i = 1 : q$ do
18
19 563 19: if $g_{k,i}^{2nd}$ comprises solely one second-vertex then
20
21 · Merge the first-layer subgroup that corresponds to $g_{k,i}^{2nd}$ directly into G_k^{mer} ;
22
23 565 21: else
24
25 · Merge all first-layer vertices of $g_{k,i}^{2nd}$ into a set $V_{k,i}^{2nd}$;
26
27 · $G_{k,i}^{mer} = \text{GROUP-INITIATION}(V_{k,i}^{2nd}, \varepsilon)$;
28
29 568 24: · $G_k^{mer} = G_k^{mer} \cup G_{k,i}^{mer}$;
30
31 571 25: · Organize G'_k into the form $G'_k = G_k^{mer} = \{g'_{k,1}, \dots, g'_{k,s}\}$;
32
33 572 26: 4) Group node addition and deletion :
34
35 573 27: · Find the second-layer group vertices:
36
37 $V_k^g = \{v_{k,1}^g, \dots, v_{k,s}^g\}$
38
39 = CONSTRUCT-SECOND-VERTICES(V'_k, G'_k);
40
41 574 28: · Set $V_{new} = \{v_1, \dots, v_l\}$ denote the set of vertices that has not been utilized at the
42
43 current moment;
44
45 575 29: for $i = 1 : l$ do
46
47 · Iterate through all elements in set V_k^g , identify $\hat{v}_{k,j}^g$ with the minimum Mahalanobis
48
49 distance to v_i ;
50
51 · Include v_i in subgroup $g'_{k,j}$;
52
53 577 31: · Eliminate virtual vertices that have not been updated for consecutive time intervals;
54
55 580 33: return: $G_k = \{g_{k,1}, g_{k,2}, \dots, g_{k,z}\}$

```

---

## REFERENCES

- 583 [1] L. Li, Q. Wu, B. Yang, S. Wei, and J. Wang, "Labeled multi-bernoulli filter based group target tracking using sde and  
584 graph theory," in *2021 IEEE 24th International Conference on Information Fusion (FUSION)*. IEEE, 2021, pp. 1–8.

- 1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60
- [2] G. Li, G. Li, and Y. He, "Resolvable group target tracking via multi-bernoulli filter and its application to sensor control scenario," *IEEE Transactions on Signal Processing*, vol. 70, pp. 6286–6299, 2022.
- [3] Y. Xia, K. Granström, L. Svensson, Á. F. García-Fernández, and J. L. Williams, "Extended target poisson multi-bernoulli mixture trackers based on sets of trajectories," in *2019 22th International Conference on Information Fusion (FUSION)*. IEEE, 2019, pp. 1–8.
- [4] R. Mahler, *Statistical multisource-multitarget information fusion*. Artech, 2007.
- [5] Y. Bar and T. Shalom, "Fortmann," tracking and data association", 1988.
- [6] S. S. Blackman and R. Popoli, "Design and analysis of modern tracking systems," (*No Title*), 1999.
- [7] X. Hao, Y. Liang, W. Zhang, and L. Xu, "Structure identification and tracking of multiple resolvable group targets with circular formation," in *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, vol. 9. IEEE, 2020, pp. 910–915.
- [8] S. Reuter, B.-T. Vo, B.-N. Vo, and K. Dietmayer, "The labeled multi-bernoulli filter," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3246–3260, 2014.
- [9] W. Liu, S. Zhu, C. Wen, and Y. Yu, "Structure modeling and estimation of multiple resolvable group targets via graph theory and multi-bernoulli filter," *Automatica*, vol. 89, pp. 274–289, 2018.
- [10] W. Liu, Y. Chi, and G. Zhang, "Multiple resolvable group estimation based on the glmb filter with graph structure," in *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE, 2018, pp. 960–964.
- [11] W. Liu and Y. Chi, "Resolvable group state estimation with maneuver based on labeled rfs and graph theory," *Sensors*, vol. 19, no. 6, p. 1307, 2019.
- [12] X. Ru, Y. Chi, and W. Liu, "A detection and tracking algorithm for resolvable group with structural and formation changes using the gibbs-glmb filter," *Sensors*, vol. 20, no. 12, p. 3384, 2020.
- [13] Z. Zhang, J. Sun, H. Zhou, and C. Xu, "Group target tracking based on ms-member filters," *Remote Sensing*, vol. 13, no. 10, p. 1920, 2021.
- [14] Q. Li, S. J. Godsill, J. Liang, and B. I. Ahmad, "Inferring dynamic group leadership using sequential bayesian methods," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8911–8915.
- [15] S. K. Pang, J. Li, 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.
- [16] G. Li, G. Li, and Y. He, "Labeled multi-bernoulli filter based multiple resolvable group targets tracking with leader-follower model," *IEEE Transactions on Aerospace and Electronic Systems*, 2023.
- [17] A. Gning, L. Mihaylova, S. Maskell, S. K. Pang, and S. Godsill, "Group object structure and state estimation with evolving networks and monte carlo methods," *IEEE Transactions on Signal Processing*, vol. 59, no. 4, pp. 1383–1396, 2010.
- [18] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE Computer Society, 2018, pp. 1549–15 498.
- [19] H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, J. Schneider, D. Bradley, and N. Djuric, "Deep kinematic models for kinematically feasible vehicle trajectory predictions," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10 563–10 569.
- [20] A. Kullberg, I. Skog, and G. Hendeby, "Online joint state inference and learning of partially unknown state-space models," *IEEE Transactions on Signal Processing*, vol. 69, pp. 4149–4161, 2021.
- [21] T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," *arXiv preprint arXiv:2001.03093*, 2020.

- 1  
2  
3 [627] [22] J. Li, H. Ma, Z. Zhang, J. Li, and M. Tomizuka, "Spatio-temporal graph dual-attention network for multi-agent prediction  
4 and tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 10 556–10 569, 2021.  
5  
6 [629] [23] T. Westny, J. Oskarsson, B. Olofsson, and E. Frisk, "Mtp-go: Graph-based probabilistic multi-agent trajectory prediction  
7 with neural odes," *IEEE Transactions on Intelligent Vehicles*, 2023.  
8  
9 [631] [24] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," *Advances in neural  
10 information processing systems*, vol. 31, 2018.  
11  
12 [633] [25] B.-T. Vo and B.-N. Vo, "Labeled random finite sets and multi-object conjugate priors," *IEEE Transactions on Signal  
13 Processing*, vol. 61, no. 13, pp. 3460–3475, 2013.  
14  
15 [635] [26] D. B. West *et al.*, *Introduction to graph theory*. Prentice hall Upper Saddle River, 2001, vol. 2.  
16  
17 [636] [27] A. R. Runnalls, "Kullback-leibler approach to gaussian mixture reduction," *IEEE Transactions on Aerospace and Electronic  
18 Systems*, vol. 43, no. 3, pp. 989–999, 2007.  
19  
20 [638] [28] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence  
21 modeling," *arXiv preprint arXiv:1412.3555*, 2014.  
22  
23 [640] [29] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE  
24 transactions on neural networks*, vol. 20, no. 1, pp. 61–80, 2008.  
25  
26 [642] [30] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE  
27 transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.  
28  
29 [644] [31] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?" *arXiv preprint arXiv:2105.14491*, 2021.  
30  
31 [645] [32] T. Westny, J. Oskarsson, B. Olofsson, and E. Frisk, "Evaluation of differentially constrained motion models for graph-based  
32 trajectory prediction," *arXiv preprint arXiv:2304.05116*, 2023.  
33  
34 [647] [33] S. Reuter, "Multi-object tracking using random finite sets," Ph.D. dissertation, Universität Ulm, 2014.  
35  
36 [648] [34] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *IEEE  
37 transactions on signal processing*, vol. 56, no. 8, pp. 3447–3457, 2008.  
38  
39 [650] [35] M. Beard, B. T. Vo, and B.-N. Vo, "Ospa (2): Using the ospa metric to evaluate multi-target tracking performance," in  
40 *2017 International Conference on Control, Automation and Information Sciences (ICCAIS)*. IEEE, 2017, pp. 86–91.  
41  
42 [652] [36] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*,  
43 "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*,  
44 vol. 32, 2019.  
45  
46 [655] [37] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric," *arXiv preprint arXiv:1903.02428*,  
47 2019.  
48  
49 [657] [38] R. H. He and R. Zou, "functorch: Jax-like composable function transforms for pytorch," 2021.  
50  
51 [658] [39] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction  
52 in crowded spaces," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer  
53 Society, 2016, pp. 961–971.  
54  
55 [661] [40] G. Li, G. Li, and Y. He, "Distributed multiple resolvable group targets tracking based on hypergraph matching," *IEEE  
56 Sensors Journal*, 2023.  
57  
58 [663] [41] H. Yu, W. An, R. Zhu, and R. Guo, "A hypergraph matching labeled multi-bernoulli filter for group targets tracking,"  
59 *IEICE TRANSACTIONS on Information and Systems*, vol. 102, no. 10, pp. 2077–2081, 2019.  
60  
61 [665] [42] E. Faerman, O. Voggenreiter, F. Borutta, T. Emrich, M. Berrendorf, and M. Schubert, "Graph alignment networks with  
62 node matching scores," *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, vol. 2, 2019.



667 **Yue Yu** was born in 2000. She received the B.S. degree in the school of electronics and information  
668 engineering from Harbin Institute of Technology, Harbin, China, in 2021. Since then, she has continued  
669 to study for her doctorate at the same school. Her current research interests include nonlinear filtering,  
670 estimation, and information fusion. She is also interested in communication topology in networks.  
671



672 **Mei Liu** was born in 1963. She received the Ph.D. degree in electronic and communication engineering  
673 from Harbin Institute of Technology, Harbin, China, in 2006. She is currently a Professor and a Ph.D.  
674 supervisor of signal and information processing with the school of electronics and information engineering,  
675 Harbin Institute of Technology. In recent years, she has been engaged in scientific research and teaching  
676 in multi-sensor data fusion, array signal processing, and satellite communication technology.  
677



678 **Bo Li**'s research interests include machine learning, deep learning, natural language processing, sparse  
679 signal processing, compressed sensing, etc. He is currently working in the Key laboratory of Intelligent  
680 Technology and Application of Marine Equipment, the Key laboratory of Ship Intelligent System and  
681 Technologies, the Key laboratory of Environment Intelligent Perception, the College of Intelligent System  
682 Science and Engineering, Harbin Engineering University.  
683

## Authors' responses of the manuscript

**Manuscript number:** TAES-2024-0014

**Title:** Enhancing Resolvable Group Target Tracking: Integration of Labeled Multi-Bernoulli Filter with Deep Learning Approaches.

We express our profound appreciation to the editors and reviewers for their invaluable insights, which we have utilized to refine the quality of our manuscript. We have diligently revised the manuscript in accordance with the comments and suggestions provided by the editorial board and reviewers. Below, we present comprehensive responses addressing each comment. Significant modifications incorporated in the revised version of the manuscript are delineated beneath the respective comment, [along with the corresponding page and line numbers of the one-column format, underscored for clarity](#). Specifically, we have rectified the title of Table II as suggested by the editor (in the revised version it becomes Table III in Page 18), and we extend our gratitude for the editor's meticulousness in verifying this. We apologize for the oversights in the original manuscript, and we firmly believe that the revisions resulting from the critique of editors and reviewers have significantly enhanced the manuscript. We trust that it now aligns with the standards for publication in "IEEE Transactions on Aerospace and Electronic Systems."

## Response to the reviewers

### Reviewer 1

**Reviewer Comment 1.1** — The title of the paper uses the term “resolvable group target tracking” (RGTT), but elsewhere in the abstract and the introduction, and “s” is added to “target” in the RGTT definition. Please keep consistent and preferably in line with the convention in the literature which I believe would be without the “s.”

**Reply:** We sincerely appreciate your thorough review of our paper and the insightful comments you provided. Your attention to detail is greatly valued. Upon reevaluation of the terminology used in our paper, we acknowledge that the inconsistency in the use of the term “RGTs” was an oversight on our part. We apologize for any confusion this may have caused.

After careful consideration and consultation of existing literature [1] [2], we agree that maintaining consistency with the convention of using “RGT” without the appended “s” is more appropriate. Consequently, a detailed check was conducted throughout the entirety of the full text to eliminate any unnecessary occurrences of “s” and ensure consistency in the terminology.

**Reviewer Comment 1.2** — I believe the paper would benefit from mentioning the Generalized Labeled Multi-Bernoulli (GLMB) filter and the assumptions that the LMB makes from this more recent filter.

**Reply:** We sincerely appreciate your valuable input to include discussion regarding the Generalized Labeled Multi-Bernoulli (GLMB) filter in our paper. In response to your comment, we have briefly introduced GLMB filter in the manuscript and elaborated the relationship between LMB and GLMB filter. Specifically, we have emphasized that the LMB filter can be viewed as a special case of the GLMB filter, and we have outlined the assumptions made by the LMB filter in comparison to the more general framework provided by the GLMB filter. This addition enhances the clarity and coherence of our paper, providing readers with a better understanding of the context and the advantages of utilizing the LMB filter.

The augmentations to Section II-A are delineated as follows:

The Generalized Labeled Multi-Bernoulli (GLMB) filter [3], based on labeled RFS theory, is a groundbreaking closed-form solution for Bayesian filtering in multi-target tracking. Unlike conventional filters, it not only estimates target states but also provides their trajectories simultaneously. In this paper, we utilize the LMB filter as a specific instance of the GLMB filter, essentially serving as a first-moment approximation. The key distinction between the LMB and GLMB filters lies in their treatment of data association uncertainty. The LMB filter assumes that all targets belong to a single component (i.e., a single set of track labels), whereas the GLMB filter accommodates multiple hypotheses, each representing potential sets of target track labels. Experiments detailed in [3] demonstrate that while its performance is slightly lower than the GLMB filter, it effectively reduces computational load and conserves storage space. Furthermore, a noteworthy advantage of the LMB filter in this paper lies in its representation of trajectories in discrete form, thereby facilitating seamless integration with a neural network to predict the existence probability and spatial distribution of each independent LMB component. Following this, we offer a brief introduction to the LMB filter. [\(Pages 4-5, Lines 103-116\)](#)

**Reviewer Comment 1.3** — Page 4, Column 2, Line 29: “Following that, create a...” uses the active voice which clashes with the passive voice in the rest of the subsection.

**Reply:** Thank you for bringing this matter to our attention, and we value your thorough review in enhancing the quality of our work. Unfortunately, we apologize for the inconsistency between the active and passive voices. We have rectified this oversight by ensuring consistency in voice usage throughout the subsection. The revised expression is as follows: “Subsequently, a decoder is created with a structure ...”. [\(Page 10, Lines 193-194\)](#) Additionally, we have conducted a comprehensive examination of the entire manuscript to mitigate similar discrepancies.

**Reviewer Comment 1.4** — The caption of Fig 5. is replicated from Fig. 4.

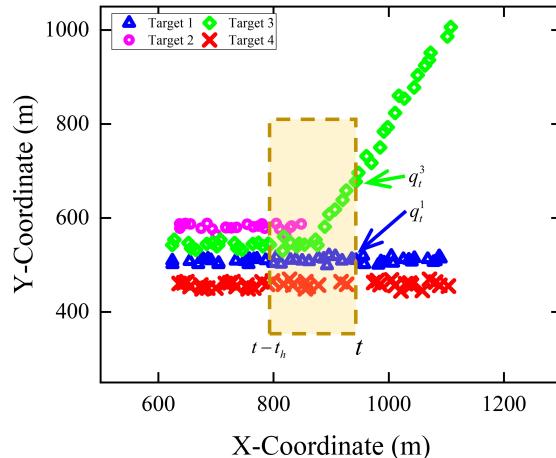


Fig. 5. An example showcasing the incomplete consideration of neighboring targets in the original method in [4] emphasizes the importance of the group structure estimation algorithm introduced in this paper.

**Reply:** We appreciate your meticulous scrutiny of our manuscript. Regrettably, we must acknowledge an oversight that resulted in the duplication of the caption from Fig. 4 for Fig. 5. We apologize for this error. We have promptly corrected this mistake, and the correct title for Fig. 5 is provided here. [\(Page 13\)](#) Additionally, we have conducted a comprehensive review of all figure and table captions throughout the manuscript to avoid similar occurrences.

**Reviewer Comment 1.5** — Equation 43: the line in the center is misleading as to the two elements of the vector measurement. A solid line typically denotes a fraction.

**Reply:** We are grateful for your comprehensive examination of our manuscript. With regard to Equation 43, we acknowledge your concern regarding the potential confusion arising from the utilization of a solid line in the center, typically indicative of a fraction. We concur that this might mislead readers regarding the structure of the vector measurement. Consequently, the revised version has eliminated the line in the center of the formula: (Page 17)

$$z = \left[ \frac{\text{atan}2(p_x - p_x^s, p_y - p_y^s)}{\sqrt{(p_x - p_x^s)^2 + (p_y - p_y^s)^2}} \right] + \epsilon, \quad (43)$$

## Reviewer 2

The manuscript proposes improving RGTT by integrating deep learning techniques with the Labeled Multi-Bernoulli (LMB) filter. The key contribution lies in the novel use of Graph Neural Networks (GNN) and Recurrent Neural Networks (RNN) to better capture the dynamics and interactions within target groups, aiming to enhance tracking accuracy and resolution.

**Reviewer Comment 2.1** — However, the methodology presented lacks detailed explanations necessary for reproducibility and replication. Critical elements such as algorithmic specifics, parameter settings, and implementation details are not sufficiently elaborated, making it challenging for other researchers to replicate the study or assess the method's robustness.

**Reply:** We appreciate your feedback on the lack of detailed explanations in our methodology, especially regarding algorithmic specifics, parameter settings, and implementation details. We apologize for any confusion caused by insufficient elaboration on neural network parameters in our manuscript. In response, we have implemented substantial enhancements, notably by integrating a detailed table format to present neural network architecture parameters. This facilitates easier reference and enhances readability. The revised content is presented as follows:

The neural network was built and trained using PyTorch [5], utilizing the Adam optimizer with a learning rate of 0.0001. The training process was carried out in batches of 64 samples, spanning a total of 200 epochs. The detailed architecture of the network depicted in Fig. 4 (Page 11) is outlined in Table IV. (Page 19, Lines 361-364)

**Reviewer Comment 2.2** — Regarding the numerical simulation, the paper does not provide clear information on how the simulation environment is designed to validate the proposed method effectively. Specific details like sensor resolution, which greatly affects the system's ability to distinguish closely spaced targets, are not mentioned. This omission raises questions about the simulation's ability to accurately reflect real-world conditions and test the system's performance comprehensively.

TABLE IV  
COMPONENTS AND DIMENSIONS OF THE NEURAL NETWORK MODULE

| Module                | Component                                         | Detail description                                                                                                                                                    |
|-----------------------|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Encoder               | GRU                                               | With a hidden layer dimension of 64.                                                                                                                                  |
|                       | GAT+                                              | Comprising a fully connected (FC) layer and a single-headed GAT layer implemented by GATv2Conv module in PyTorch Geometric [6], both with an output dimension of 192. |
| Decoder               | Attention                                         | Comprising two FC layers, with output dimensions 64 and $t_h + 1$ .                                                                                                   |
|                       | FC layer used to calculate $\hat{q}_{t+1}$        | With an output dimension of 64.                                                                                                                                       |
| Existence probability | FC layer used to calculate $\hat{r}_+^{(\ell)}$   | With an output dimension of 1.                                                                                                                                        |
| Spatial distribution  | FC layer used to calculate $u_{t+1}^{(\ell,j)}$   | Solving $u_1$ and $u_2$ in (24), with an output dimension of $2 * N_{GC}$ .                                                                                           |
|                       | FC layer used to calculate $Q_{t+1}^{(\ell,j)}$   | Solving $\sigma$ in (45), with an output dimension of $N_{GC}$ .                                                                                                      |
|                       | FC layer used to calculate $\hat{w}_+^{(\ell,j)}$ | With an output dimension of $N_{GC}$ .                                                                                                                                |
|                       | Neural ODE                                        | Comprising two groups of FC layers, each with two layers, approximating $f_1$ and $f_2$ in (24), with output dimensions of 16 and 1.                                  |
| EKF                   |                                                   | Computing the Jacobian matrix of $f_1$ and $f_2$ output by the Neural ODE module, using functorch [7].                                                                |

**Reply:** We sincerely appreciate your insightful remarks regarding the clarity and comprehensiveness of information concerning the numerical simulation environment employed to validate the proposed method. In response to your invaluable feedback, we have undertaken significant revisions to the manuscript, incorporating a comprehensive description of the simulation environment and sensor parameters.

The sensor is positioned at the coordinates (0m,0m) and surveils the entire area with a data sampling frequency of 1 Hz. (Page 17, Lines 332-333). The correlation between sensor resolution and measurement error in both range and azimuth dimensions, as established in this paper, is modeled by the following formula [8]:

$$\sigma_r = \frac{r_\sigma}{\sqrt{2}\sqrt{SNR}}, \sigma_a = \frac{\theta_B}{\sqrt{2}\sqrt{SNR}}$$

where  $r_\sigma$  and  $\theta_B$  represent the resolution of the two dimensions for range and azimuth, respectively. Additionally,  $\sigma_r$  and  $\sigma_a$  denote the measurement error in these two dimensions. Furthermore, environmental conditions and external disturbances in practical applications, such as atmospheric turbulence, pulse jitter, and receiving channel delay, have the potential to adversely impact measurement error. Consequently, the measurement error parameter is adjusted slightly higher than the result calculated directly by the formula. Additionally, the interrelation between detection probability, false alarm probability, and signal-to-noise ratio is elucidated in [9].

These details are presented in a clear and structured manner, utilizing a table format to facilitate ease of reference and readability. We firmly believe that these additions augment the transparency and credibility of our study. The revised content is as follows:

The remaining parameters involved in the simulation are presented in Table II. (Page 17, Lines 342-343)

**Reviewer Comment 2.3** — Moreover, in scenarios involving closely moving targets, it is unclear whether the system can handle label switching—a critical aspect in RGTT where maintaining consistent target identities is essential. The absence of a discussion on label switching and its impact on tracking performance is a notable gap.

1  
2  
3           **TABLE II**  
4           SIMULATION PARAMETERS AND THEIR SYMBOLS AND VALUES  
5

| 6     Simulation Parameter   | 7     Symbol | 8     Value |
|------------------------------|--------------|-------------|
| Detection probability        | $P_D$        | 0.90        |
| Clutter rate                 | $\lambda_c$  | 20          |
| Sensor resolution in range   | $a_r$        | 15m         |
| Sensor resolution in azimuth | $a_\varphi$  | 2°          |
| Sensor signal-to-noise ratio | SNR          | 10dB        |
| False alarm probability      | $P_{FA}$     | $10^{-3}$   |
| Maximum number of GCs        | $N_{GC}$     | 15          |
| Merging threshold            | $\xi_m$      | 4           |
| Pruning threshold            | $\xi_p$      | $10^{-4}$   |
| Extraction threshold         | $\xi_e$      | 0.5         |

19           **Reply:** We genuinely appreciate your insightful commentary concerning the management of label switching  
 20 in scenarios involving closely moving targets within the realm of RGTT. In this field, a plethora of studies  
 21 have been conducted. For instance, [10] presents a multi-scan version of the GLMB model, which integrates  
 22 backward smoothing with forward filtering, consequently diminishing label switching and enhancing estimation  
 23 performance. Additionally, [11] introduces a method with reduced computational complexity, and subsequently,  
 24 [12] adapts the approach from the GLMB filter to the LMB filter.

25           We regret the oversight of not discussing label switching, a crucial aspect in RGTT for maintaining consistent  
 26 target identities. In response to your feedback, we have allocated Section IV-C of the revised manuscript to  
 27 addressing label switching and its influence on tracking performance. We are confident that the incorporation of  
 28 this discussion enriches the comprehensiveness and significance of our work. The revised content is outlined as  
 29 follows:

31           In this subsection, we present **Scenario 1** from subsection II-B as an illustrative case to evaluate the effectiveness  
 32 of the proposed filter in maintaining label consistency. We employed the OSPA<sup>(2)</sup> distance [13] as the evaluation criterion, and the results are depicted in Fig. 13. Notably, the influence of target births or deaths on  
 33 OSPA<sup>(2)</sup> distance persists due to the presence of the sliding window, resulting in a smoother curve evolution in  
 34 Fig. 13 compared to Fig. 8. (Page 22) The GNN-LMB filter demonstrates superior performance by accurately  
 35 capturing interactions among RGT, thereby facilitating more precise prediction of states for each target. This  
 36 mitigates the challenge of associating data between dense targets and corresponding measurements, thereby  
 37 reducing the frequent switching of RGT labels. Conversely, the standard LMB filter inherently tends to produce  
 38 fragmented trajectories with inconsistent labels, particularly evident when handling targets in close proximity,  
 39 leading to the poorest performance among the four filters.

41           Next, the clutter rate  $\lambda_c$  is incrementally increased every 20 intervals from 20 to 80, and the OSPA<sup>(2)</sup> errors  
 42 averaged over time for each filter are compared, as depicted in Table VII. The results indicate that while the  
 43 GNN-LMB filter consistently outperforms the other filters, its relative performance improvement compared  
 44 to the standard LMB filter diminishes with increasing clutter. For instance, there is a 36.76% improvement  
 45 when  $\lambda_c = 20$ , contrasted with only a 16.32% improvement when  $\lambda_c = 80$ . This implies that as the clutter  
 46 rate increases, the proposed GNN-LMB filter encounters challenges in effectively addressing label switching in  
 47 RGTT.

48           It is noteworthy that recent studies have applied hypergraph matching techniques to RGTT data association,  
 49 effectively reducing label switching [14] [15]. Indeed, GNN has been demonstrated to efficiently address the  
 50 challenge of node matching in hypergraph data [16]. Therefore, besides employing the GNN-GRU module to  
 51 learn interactions among RGT as proposed in this paper, exploring its potential in mitigating label switching  
 52 induced by intricate RGTT data association presents a promising avenue for future research. (Pages 25-26,  
 53 Lines 422-447)

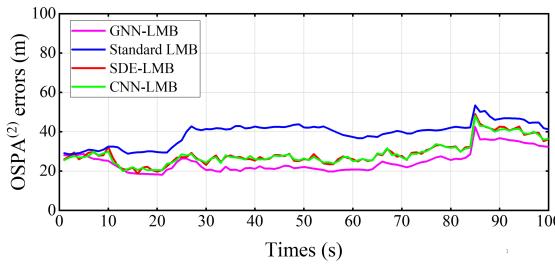


Fig. 13. The OSPAn(2) errors of each filter in Scenario 1.

TABLE VII  
THE AVERAGE OSPAn(2) ERRORS (m) FOR DIFFERENT CLUTTER RATE

|                                | $\lambda_c$  | 20    | 40    | 60    | 80    |
|--------------------------------|--------------|-------|-------|-------|-------|
| Average<br>OSPA <sup>(2)</sup> | Standard LMB | 39.23 | 40.86 | 44.18 | 48.96 |
|                                | SDE-LMB      | 29.08 | 31.25 | 36.54 | 42.72 |
|                                | CNN-LMB      | 28.96 | 31.13 | 36.65 | 42.54 |
|                                | GNN-LMB      | 24.81 | 26.66 | 32.08 | 40.97 |

**Reviewer Comment 2.4** — To substantiate the proposed method's advantages, a comprehensive ablation study is necessary. Such a study would help in understanding the contribution of each component of the system, particularly the deep learning integration, to the overall performance. It would also highlight the system's behaviour under various conditions and parameter settings, offering insights into its adaptability and efficiency in different tracking scenarios.

**Reply:** We are immensely grateful for your insightful suggestion regarding the imperative need for a comprehensive ablation study to substantiate the advantages of the proposed method. In accordance with your recommendation, we have expanded upon the contents of the original Section IV-C to create the revised Section IV-D within the manuscript, concentrating on conducting a thorough ablation study. We firmly believe that these additions substantially augment the rigor and comprehensiveness of our work, offering valuable insights into the adaptability and efficacy of the proposed method. The complete modified content is as follows:

The ablation study endeavors to elucidate the most pivotal elements contributing to the system's efficacy, thus guiding its refinement and optimization. [4] conducted ablation experiments on each constituent of the

TABLE VIII  
THE INITIAL STATE AND THE BIRTH/DEATH TIME (SCENARIO 3)

| Target ID | Initial State         | Birth time (s) | Death time (s) |
|-----------|-----------------------|----------------|----------------|
| 1         | [300, 10, 1100, 0]    | 1              | 80             |
| 2         | [270, 10, 1130, 0]    | 1              | 80             |
| 3         | [300, 10, 1050, 0]    | 1              | 80             |
| 4         | [700, 10, 1150, 0]    | 41             | 80             |
| 5         | [-470, 10, 2000, 0]   | 21             | 100            |
| 6         | [-500, 10, 1970, 0]   | 21             | 100            |
| 7         | [-500, 10, 2600, -30] | 21             | 100            |
| 8         | [-800, 0, 300, 20]    | 21             | 100            |
| 9         | [-760, 0, 300, 20]    | 21             | 100            |
| 10        | [-780, 0, 340, 20]    | 21             | 100            |

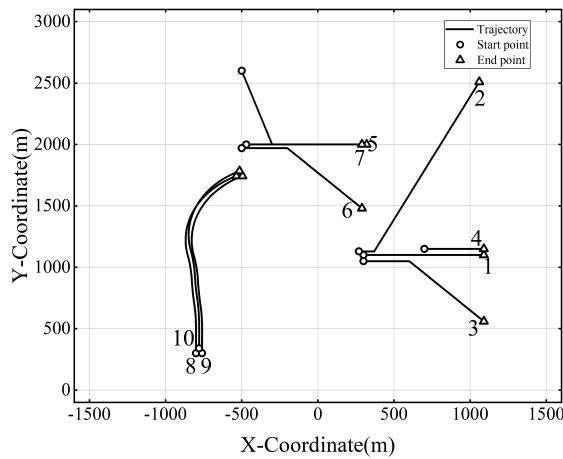


Fig. 14. The target trajectories used in Scenario 3.

TABLE IX  
THE RESULT OF ABLATION STUDY

| Environment Index | Existence probability | EGN | GM model | Average OSPA | Average OSPA <sup>(2)</sup> |
|-------------------|-----------------------|-----|----------|--------------|-----------------------------|
| <b>Scenario 1</b> | I <sub>11</sub>       | ✓   | ✓        | ✓            | 13.80                       |
|                   | I <sub>12</sub>       | ✗   | ✓        | ✓            | 14.61                       |
|                   | I <sub>13</sub>       | ✓   | ✗        | ✓            | 13.96                       |
|                   | I <sub>14</sub>       | ✗   | ✗        | ✓            | 14.71                       |
|                   | I <sub>15</sub>       | ✓   | ✓        | ✗            | 20.42                       |
| <b>Scenario 3</b> | I <sub>31</sub>       | ✓   | ✓        | ✓            | 24.44                       |
|                   | I <sub>32</sub>       | ✗   | ✓        | ✓            | 25.23                       |
|                   | I <sub>33</sub>       | ✓   | ✗        | ✓            | 26.38                       |
|                   | I <sub>34</sub>       | ✗   | ✗        | ✓            | 28.25                       |
|                   | I <sub>35</sub>       | ✓   | ✓        | ✗            | 30.06                       |

neural network employed in this paper, elucidating that the elimination of EKF and neural ODE components significantly compromises performance. Moreover, in the comparative assessment of the encoder and decoder, it was noted that integrating GNN in the latter yielded more discernible enhancements in performance than in the former. Additionally, [4] juxtaposed the impact of various GNN models on overall performance, while [17] scrutinized the influence of diverse motion models and numerical solvers.

In this paper, the model in [4] undergoes refinement and adaptation to accommodate the RGTT problem, introducing two innovative components: a component for predicting the existence probability of LMB density and an EGN component. Furthermore, the exploration of implementing the LMB filter with GM model is paramount for practical applications. However, in [4], there's no explicit assessment of how a GM model influences system performance. Therefore, the ablation study of the three aforementioned components is conducted in this subsection.

To simulate a more intricate group structure evolution, a scenario involving target birth, merging, and separation within the same group (referred to as **Scenario 3** thereafter) is established, as illustrated in Fig. 14. The kinematic parameters are detailed in Table VIII. The ablation study considers **Scenario 1** (from subsection IV-B) and **Scenario 3** as illustrative examples, utilizing OSPA and OSPA<sup>(2)</sup> as criteria for performance evaluation. The results are depicted in Table IX, where checkmarks and crosses indicate whether the component is included

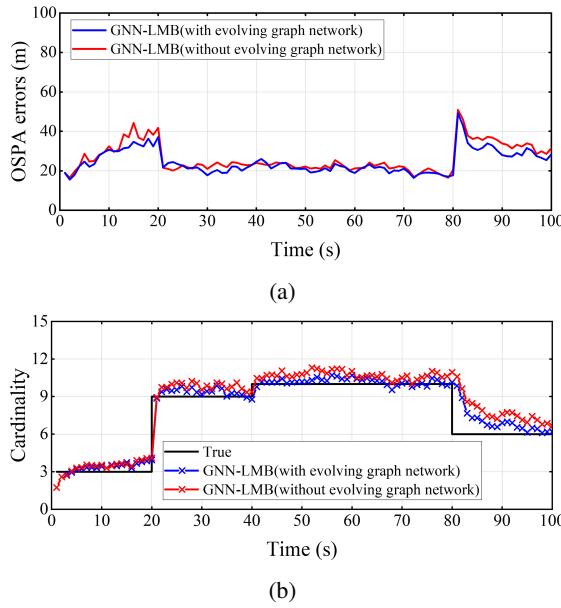


Fig. 15. The tracking performance of two GNN-LMB filters. (a) The OSPA error; (b) The target cardinality estimation.

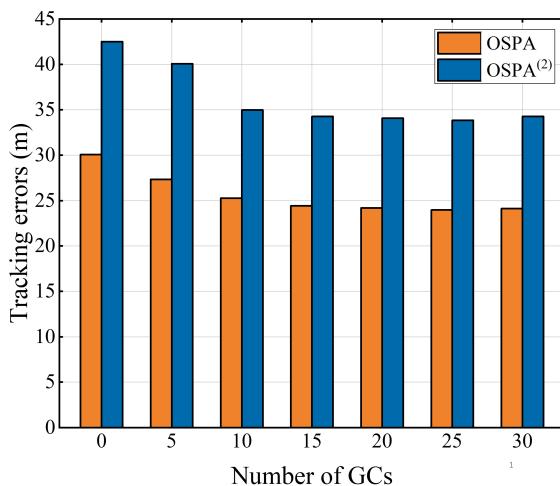
in the network.

In the experiment where the existence probability prediction component is excluded, setting  $P_s^{(\ell)} = 0.99$ , and then solving for  $r_{+,S}^{(\ell)}$  with (5) and (7). (Pages 5-6) It is observed that the absence of this component does indeed affect the overall performance to a certain extent (see I<sub>12</sub> or I<sub>32</sub>). This is due to the fact that the value of 0.99 is empirically determined and cannot be dynamically adjusted according to the movement of the target. Consequently, this fixed and relatively high value may lead to an overestimation of the number of targets during tracking and a delay in response when targets disappear, thereby adversely impacting tracking performance.

The exclusion of the EGN in *Scenario 3* has a more significant impact on performance compared to *Scenario 1*, owing to the more intricate changes in group structure in *Scenario 3* (see I<sub>11</sub> vs I<sub>13</sub> and I<sub>31</sub> vs I<sub>33</sub>). The primary distinction between a GNN-LMB filter without the EGN and one incorporating it resides in the input to the encoder. In the former, the initial step entails identifying the central target  $v$  and its neighboring nodes, whose Mahalanobis distance from  $v$  is less than the threshold at time  $t$ . Subsequently, the sequence of fully connected graph structures for these targets is traced back from time  $t - t_h$  to  $t$ , forming the input for the neural network encoder. In contrast, the latter employs the EGN to dynamically estimate the group structure at each moment, directly utilizing the group structure from time  $t - t_h$  to  $t$  as its input. With the EGN, all targets in the scene are simultaneously predicted, eliminating the need for the concept of a center target. The OSPA error and target number estimation of I<sub>31</sub> and I<sub>33</sub> are elaborated upon in Fig. 15. The results underscore that the incorporation of the EGN enhances the accuracy of group structure estimation, preventing the oversight of valuable neighbor nodes. This, consequently, positively influences the overall enhancement in tracking performance and bolsters the adaptability of the model to complex environments.

In I<sub>15</sub> and I<sub>35</sub>, the GM model is omitted, i.e.,  $N_{GC} = 0$ . The absence of GM model results in a significant loss of tracking performance (see I<sub>15</sub> or I<sub>35</sub>). This underscores the importance of GM model, which effectively captures data uncertainty, enhances the model's capacity to fit complex data distributions, and improves overall model expression and generalization capabilities. Moreover, the comparative analysis of tracking performance across various  $N_{GC}$  values is depicted in Fig. 16 (using *Scenario 3* as an example). The findings suggest that augmenting  $N_{GC}$  within a delimited scope enhances tracking performance. However, increasing model intricacy amplifies the risk of overfitting, and the proliferation of parameters complicates model training, potentially

1  
2  
3 leading to sluggish or even declining performance enhancements. (Pages 26-30, Lines 448-499)  
4  
5



22 Fig. 16. Tracking errors under different  $N_{GC}$  in Scenario 3.  
23  
24  
25  
26  
27  
28  
29

## References

- [1] G. Li, G. Li, and Y. He, "Resolvable group target tracking via multi-bernoulli filter and its application to sensor control scenario," *IEEE Transactions on Signal Processing*, vol. 70, pp. 6286–6299, 2022.
- [2] G. Li, G. Li, and Y. He, "Labeled multi-bernoulli filter based multiple resolvable group targets tracking with leader-follower model," *IEEE Transactions on Aerospace and Electronic Systems*, 2023.
- [3] B.-T. Vo and B.-N. Vo, "Labeled random finite sets and multi-object conjugate priors," *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 3460–3475, 2013.
- [4] T. Westny, J. Oskarsson, B. Olofsson, and E. Frisk, "Mtp-go: Graph-based probabilistic multi-agent trajectory prediction with neural odes," *IEEE Transactions on Intelligent Vehicles*, 2023.
- [5] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.
- [6] M. Fey and J. E. Lenssen, "Fast graph representation learning with pytorch geometric," *arXiv preprint arXiv:1903.02428*, 2019.
- [7] R. H. He and R. Zou, "functorch: Jax-like composable function transforms for pytorch," 2021.
- [8] I. S. Merrill *et al.*, "Introduction to radar systems," *Mc Grow-Hill*, vol. 7, no. 10, 2001.
- [9] M. I. Skolnik *et al.*, *Introduction to radar systems*. McGraw-hill New York, 1980, vol. 3.
- [10] B.-N. Vo and B.-T. Vo, "A multi-scan labeled random finite set model for multi-object state estimation," *IEEE Transactions on signal processing*, vol. 67, no. 19, pp. 4948–4963, 2019.
- [11] T. T. D. Nguyen and D. Y. Kim, "Glmb tracker with partial smoothing," *Sensors*, vol. 19, no. 20, p. 4419, 2019.

- [12] H. Van Nguyen, T. T. D. Nguyen, C. Shim, and M. Anuar, "The smooth trajectory estimator for lmb filters," in *2023 12th International Conference on Control, Automation and Information Sciences (ICCAIS)*. IEEE, 2023, pp. 115–120.
- [13] M. Beard, B. T. Vo, and B.-N. Vo, "Ospa (2): Using the ospa metric to evaluate multi-target tracking performance," in *2017 International Conference on Control, Automation and Information Sciences (ICCAIS)*. IEEE, 2017, pp. 86–91.
- [14] G. Li, G. Li, and Y. He, "Distributed multiple resolvable group targets tracking based on hypergraph matching," *IEEE Sensors Journal*, 2023.
- [15] H. Yu, W. An, R. Zhu, and R. Guo, "A hypergraph matching labeled multi-bernoulli filter for group targets tracking," *IEICE TRANSACTIONS on Information and Systems*, vol. 102, no. 10, pp. 2077–2081, 2019.
- [16] E. Faerman, O. Voggenreiter, F. Borutta, T. Emrich, M. Berrendorf, and M. Schubert, "Graph alignment networks with node matching scores," *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, vol. 2, 2019.
- [17] T. Westny, J. Oskarsson, B. Olofsson, and E. Frisk, "Evaluation of differentially constrained motion models for graph-based trajectory prediction," *arXiv preprint arXiv:2304.05116*, 2023.