ICJK
Chinese Journal of Information Fusion
http://dx.doi.org/10.62762/CJIF.2025.986522

RESEARCH ARTICLE

Dynamic Target Association Algorithm for Unknown Models and Strong Interference

Xiangqi Gu[1,*], **Ziran Ding**[1], **Shutao Xia**[1] **and Wei Xiong**[1,*]

[1] Research Institute of Information Fusion, Naval Aviation University, Yantai 264001, China

Abstract

To address the performance degradation of traditional data association algorithms caused by unknown target motion models, environmental interference, and strong maneuvering behaviors in complex dynamic scenarios, this paper proposes an innovative fusion algorithm that integrates reinforcement learning and deep learning. By constructing a policy network that combines Long Short-Term Memory (LSTM) memory units and reinforcement learning dynamic decision-making, a dynamic prediction model for "measurement-target" association probability is established. Additionally, a hybrid predictor incorporating Bayesian networks and multi-order curve fitting is designed to formulate the reward function. To tackle practical interference, a dynamic ring gate screening mechanism and a trajectory consistency-based error correction module are developed, effectively suppressing clutter interference and enabling autonomous correction of association errors. Experimental results demonstrate that the proposed method significantly improves association accuracy in high-noise environments compared to traditional algorithms, enhancing robustness in complex unknown scenarios.

Keywords: data association, feature fusion, motion model, LSTM, Bayesian networks.

1 Introduction

Data association serves as a critical information fusion technology in multi-sensor systems, enabling the accurate identification of true plots and tracks by establishing temporal-spatial connections between heterogeneous sensor measurements. Conventional methods focus on predicting track values, applying specific criteria to filter relevant plots, and utilizing these plots for further filtering [1–5]. Nonetheless, these algorithms are highly sensitive to sensor inaccuracies, environmental clutter, and target motion dynamics. To address these challenges, numerous advanced algorithms have been developed [6–9]. For instance, the Truncated Joint Probability Data Association Filter (JPDAF) introduced in [3] effectively filters clutter by leveraging target motion characteristics. Fan et al. [4] tackles association issues in environments with uncertain measurement errors

Academic Editor:
Gongjian Zhou

Submitted: 01 March 2025
Accepted: 18 March 2025
Published: 12 April 2025

Vol. 2, **No.** 2, 2025.
10.62762/CJIF.2025.986522

***Corresponding authors:**
✉ Xiangqi Gu
guxiangqi1314@163.com
✉ Wei Xiong
xiongwei@csif.org.cn

Citation
Gu, X., Ding, Z., Xia, S., & Xiong, W. (2025). Dynamic Target Association Algorithm for Unknown Models and Strong Interference. *Chinese Journal of Information Fusion*, 2(2), 100–111.

© 2025 by the Authors. Published by Institute of Central Computation and Knowledge. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

and motion models by integrating fuzzy recursive least squares filtering with JPDAF, ensuring robust tracking. Qin et al. [7] employs the DBSCAN algorithm and Sequential Random Sample Consensus (RANSAC) to preprocess measurement data, significantly reducing computational demands. With the advent of artificial intelligence, researchers have incorporated AI techniques into data association, yielding promising results [10–12]. While these methods show efficacy in specific scenarios, they remain rooted in traditional frameworks, necessitating the use of filters for data processing. This approach assumes a known system model, a condition often unattainable in real-world environments.

Reinforcement Learning (RL) [20–22], a groundbreaking AI technique, has evolved over decades, producing notable methodologies such as Q-learning [13–15], dynamic programming [16–18], Policy Gradients [19], and Deep Q-Networks (DQN) [23, 24]. At its core, RL involves a machine learning autonomously in an unfamiliar environment under predefined rules, with actions aligned to the real world and feedback provided through rewards or penalties, culminating at a designated endpoint. In essence, the target data association process seeks to identify true plots generated by the target track, organizing them chronologically to form a complete track. This process parallels the quest for an optimal path and can be likened to a game of Snake, where the machine adapts to the target environment governed by motion-based rules. Thus, theoretically, RL is well-suited to address target data association challenges.

In real-world scenarios, where the target system model is unknown and association results are influenced by environmental clutter, sensor errors, and target maneuvers, this paper introduces a novel data association network architecture based on RL. First, a policy network is designed to predict the association probabilities between measurements and their potential source targets, utilizing RL's dynamic exploration capabilities and the long-term memory of LSTM networks [25]. Next, the Bayesian network analyzes the order of multi-order least squares curve fitting for the current track, predicting the next plot position using least squares curve fitting. The input Bayesian recursive function then calculates the reward value for each plot. Finally, tailored mechanisms are proposed, such as the dynamic ring gate screening mechanism for partial clutter suppression, trajectory consistency-based error correction module.

The paper is structured as follows: Section 2 designs the policy network architecture of the algorithm and explains the definition methods for the state space and action space; Section 3 defines and analyzes the reward function; Section 4 introduces the design concepts of the dynamic ring gate screening mechanism and the error correction module; Section 5 trains the network using extensive simulation data, tests its performance, and provides analysis; Section 6 summarizes the algorithm's strengths and weaknesses and explores future research directions.

2 Policy Network Design and Modeling

This section proposes a novel information fusion framework that combines the temporal modeling capability of LSTM with the decision-making ability of reinforcement learning. The fusion architecture enables effective handling of uncertain system models and noisy measurement data. During the correlation of data, sensors capture measurement information in a predefined temporal order. This information includes both authentic target markers and extraneous noise caused by sensor disruptions, environmental conditions, and similar factors. Occasionally, the positions of certain noise elements and genuine target markers in the measurement data may closely align at a specific time, complicating the accurate identification of target-originated markers. Consequently, this paper proposes a policy network capable of dynamically predicting the association probability of points, as illustrated in Figure 1. In scenarios devoid of prior knowledge, the architecture initially employs a strategic network to determine the likelihood of association between the target and measurements, proceeding to link the marker with the greatest probability.

When objects move, they exhibit inertia. This inherent property ensures that the movement patterns of a target remain consistent over short periods. Leveraging this consistency, it's feasible to pinpoint the points that originate from a target by analyzing its movement trajectory, thereby facilitating the precise alignment of measurements with the target. The trajectory of a target is not depicted by isolated points at individual moments but is rather approximated by a series of points over successive intervals. This paper therefore designs a "sliding-window" state, that is, the state of the points associated with the previous N moments. N is the "window" and the size of N is given by the sampling interval of sensor and clutter density. Usually, the larger the sampling interval is,

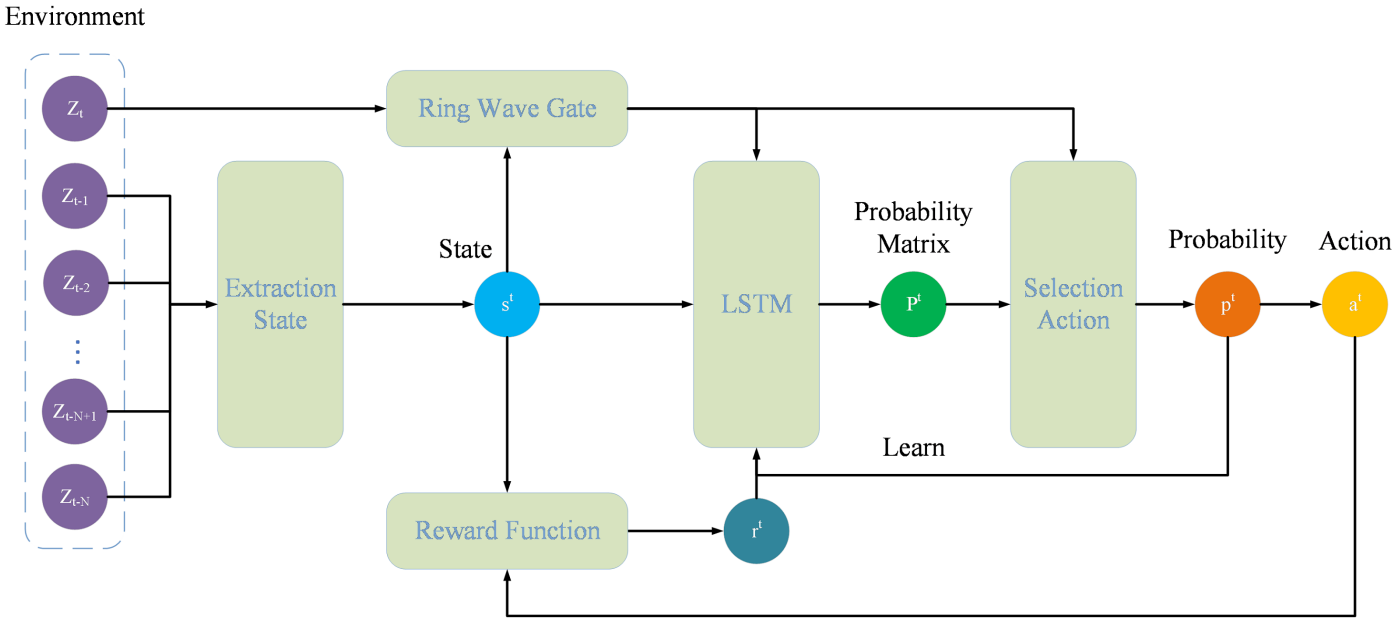


Figure 1. Policy network.

the smaller N is, and vice versa. And the stronger the clutter density, the greater the amount of data to be processed, and the higher the computational load on the hardware, resulting in a smaller N, and vice versa.

As shown in Figure 1, Z_{t-N} is the set of measurements when $t-N$, and s^t is the state consisting of points in $\{Z_{t-1}, Z_{t-2}, \ldots, Z_{t-N}\}$ originating from the target when t. Suppose $z_i^{t-N} = \begin{bmatrix} x_i^{t-N} \\ y_i^{t-N} \end{bmatrix}$ is the point in Z_{t-N}. x_i^{t-N} and y_i^{t-N} are the positions of x and y axes in the Cartesian coordinate, respectively. If s^t consists of $\{z_i^{t-1}, z_i^{t-2}, \ldots, z_i^{t-N}\}$, the expression of s^t is as follows:

$$s^t = \begin{bmatrix} X^t \\ Y^t \end{bmatrix} = \begin{bmatrix} x_i^{t-1} & x_i^{t-2} & \ldots & x_i^{t-N} \\ y_i^{t-1} & y_i^{t-2} & \ldots & y_i^{t-N} \end{bmatrix} \tag{1}$$

The construction of states dictates that every state comprises points linked to the preceding N intervals. However, at the initiation of the association process, the agent's initial state must be drawn from the measurement data of the prior N intervals. Consequently, this study introduces an Initial State Extraction mechanism tailored for the agent. Given the random dispersion of clutter, assembling clutter across N successive intervals into a unified track poses a significant challenge. Thus, the mechanism stipulates that a fundamental requirement for identifying an agent is the presence of target-derived points across N consecutive intervals. The operational sequence of this mechanism is illustrated as follows:

a) Conduct a comprehensive scan of the measurement data spanning N intervals to uncover all conceivable agents. It is essential that the points corresponding to each agent at successive intervals adhere to the predefined velocity constraints:

$$v_{min} \leq \frac{\left| z_i^{t-N} - z_i^{t-N+1} \right|}{T_{sample}} \leq v_{max}, \tag{2}$$

where T_{sample} is the sensor's sampling interval, z denotes the measurement data, and v_{max}/v_{min} represent the upper/lower velocity limits of the target.

b) According to the cosine theorem formula, the motion trend factor of the trajectory over three consecutive moments can be calculated, namely:

$$f_i^{t-N+1} = \frac{\overrightarrow{z_i^{t-N} z_i^{t-N+1}} \bullet \overrightarrow{z_i^{t-N+1} z_i^{t-N+2}}}{\left\| \overrightarrow{z_i^{t-N} z_i^{t-N+1}} \right\| \left\| \overrightarrow{z_i^{t-N+1} z_i^{t-N+2}} \right\|}, \tag{3}$$
$$f_i^{t-N+1} \in [-1, 1]$$

The motion trends of all agents are derived from the equation above, where $f = \left\{ f_i^{t-2}, f_i^{t-3}, \ldots, f_i^{t-N+1} \right\}$ defines the trend of one agent over N consecutive moments.

c) Variance quantifies the variability within a dataset. For samples of identical size, higher sample variance indicates greater volatility and lower stability. Applied to target trajectories, reduced data variability implies a steadier motion pattern

and higher reliability. Therefore, the stability metric for agent motion trends is defined as:

$$Variance = var(f), \tag{4}$$

where f represents the historical motion trend series. The optimal agent state is then selected as the set of target points with the minimal variance $Variance$.

As previously mentioned, the action space is composed of the points associated with the agent. The designed learning framework in this study is segmented into a training phase and a testing phase. Throughout the training phase, the agent endeavors to discover the most effective strategy that closely aligns with the target's movement patterns through iterative experimentation. This phase incorporates a Selection Action module known as Random Choice, where the agent picks a random probability value from P^t for a point to facilitate training. During the testing phase, the agent utilizes the insights gained to link points accurately. This phase features a Selection Action module referred to as Argmax Choice, where the agent identifies and selects the point from P^t with the greatest probability value for the purpose of association.

3 Reward Function Definition

This paper introduces a novel reward function designed to evaluate the efficacy of actions chosen in a given state by calculating their true merit. The essence of this function lies in its ability to determine the reward of an action by integrating the anticipated position of the target, as forecasted by the least squares method, into a Bayesian recursive framework at the present moment. The order of the least squares method, crucial for the accuracy of this prediction, is dynamically ascertained by the insights provided by a Bayesian network.

The target's trajectory is characterized by a spectrum of motion patterns, including but not limited to Constant Velocity (CV), Coordinate Turn (CT), and Constant Acceleration (CA). In the tapestry of real-world scenarios, a target's journey may weave through various motion models, rendering the use of a static least squares order inadequate for capturing the nuanced dynamics of its path. To address this, a "sliding window" approach is employed, which, anchored in the current state, forecasts the optimal least squares order capable of modeling the target's motion over a span of N moments. Drawing inspiration from the Bayesian network model outlined in reference [26], an M-category Bayesian network is architected, where M symbolizes the stratification of least squares orders into distinct classes. A modest M may fail to encapsulate the full gamut of potential motion trajectories, whereas an excessive M could erode the model's resilience, leading to significant discrepancies in the fitted results. The network ingests the current state as its input and yields the probabilities for each order class as its output, with the least squares order being the one crowned with the highest probability.

Leveraging the predicted order g, the least squares method [27] fits the state data to forecast the target's position. The fitted state $\widetilde{s}^t$ and predicted position $\widetilde{z}^t$ are defined as: $\widetilde{s}^t = \begin{bmatrix} \widetilde{X}^t \\ \widetilde{Y}^t \end{bmatrix}$, $\widetilde{z}^t = \begin{bmatrix} \widetilde{x}^t \\ \widetilde{y}^t \end{bmatrix}$, where the components are calculated through:

$$\widetilde{X}^t = M_t W_x \tag{5}$$

$$\widetilde{Y}^t = M_t W_y \tag{6}$$

$$\widetilde{x}^t = M_{t+1} W_x \tag{7}$$

$$\widetilde{y}^t = M_{t+1} W_y \tag{8}$$

The coefficient matrices W_x and W_y are derived from the least squares solution:

$$W_x = (M_t^T M_t)^{-1} M_t^T X^t \tag{9}$$

$$W_y = (M_t^T M_t)^{-1} M_t^T Y^t \tag{10}$$

where M_t is the design matrix constructed from time terms up to order g:

$$M_t = [T_t^0, T_t^1, \ldots, T_t^g] \tag{11}$$

$$T_t^g = [t - 1^g, t - 2^g, \ldots, t - N^g] \tag{12}$$

Finally, the reward function r^t is defined via the Bayesian recursive framework [7]:

$$r^t = \frac{P_Dr^{t-1} q^t(z_i^t)}{K_t(Z^t) + P_Dr^{t-1} q^t(z_i^t)} \tag{13}$$

$$q^t(z_i^t) = N(z_i^t; \widetilde{z}^t, S^t) \tag{14}$$

$$S^t = \mathrm{cov}(z_i^t, \widetilde{z}^t) + R \tag{15}$$

where z_i^t is the point i selected from the measurement data Z^t when t. $K_t(Z^t)$ is the clutter intensity when t, namely $K_t(Z^t) = \frac{\mathrm{num}_{z^t}}{TS}$. In the measurement data Z^t, the number of points is num_{z^t}. The area of sensor detection region is TS. R is the covariance matrix of measurement, which is also the detection error of the sensor.

4 Special Mechanism

4.1 Dynamic Ring Gate Screening Mechanism

The points originating from the target are inherently produced by it and must adhere to its motion dynamics. By leveraging the target's maximum speed v_max and minimum speed v_min, the annular region of possible points can be delineated, enabling the identification of all potential traces generated by the target. However, these points are subject to the sensor's detection inaccuracy σ_v. Consequently, the speed must dynamically adjust in response to variations in detection error, ensuring alignment with the evolving uncertainty introduced by the sensor's limitations.

$$v_max = v_max + \frac{\sigma_v^2}{T_sample} \tag{16}$$

$$v_min = v_min - \frac{\sigma_v^2}{T_sample} \tag{17}$$

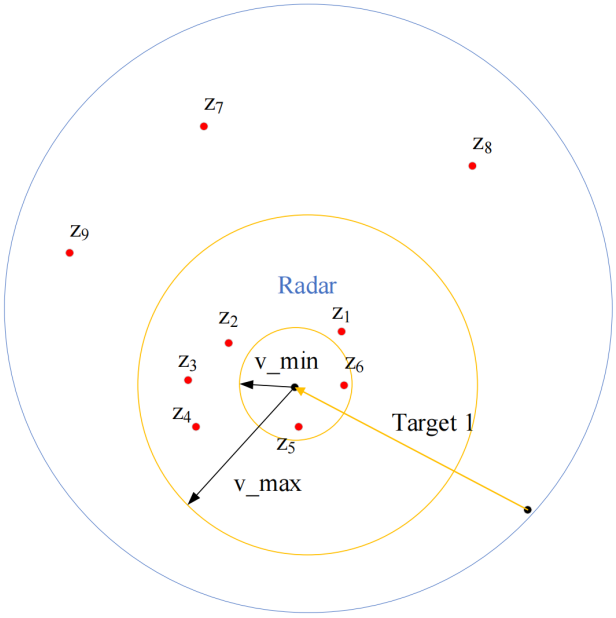


Figure 2. Dynamic ring gate screening.

Depicted in Figure 2, a target is traversing the monitored zone, with the sensor capturing nine distinct traces at this instance. Upon examining the ring-shaped gating boundary in the illustration, it is discernible that four of these traces are probable candidates for having been generated by the target, namely z_1, z_2, z_3 and z_4.

Based on the basic information of the target and the sensor, the Ring Wave Gate Screening mechanism eliminates some of the clutter from the measurement data, which not only increases the possibility of selecting the points originating from the target, but also saves the computational resources of policy network.

Utilizing essential details about the target and the sensor, the dynamic ring gate screening mechanism filters out extraneous noise from the measurement dataset. This process not only boosts the probability of detecting traces that stem from the target but also optimizes the computational efficiency of the policy network.

4.2 Error Association Module

In practical settings, targets exhibit a variety of motion patterns, and the network must learn numerous strategies, making comprehensive training unfeasible. When evaluating a new strategy, the unpredictability of the real environment complicates the selection of traces originating from the target. To address this, a module was developed to dynamically refine motion trends during testing, enhancing the limited transferability of RL, as depicted in Figure 3. This module operates by verifying at the test's conclusion; if measurement data persists into the next moment, it signifies an erroneous association, prompting the identification and rectification of the wrongly selected action.

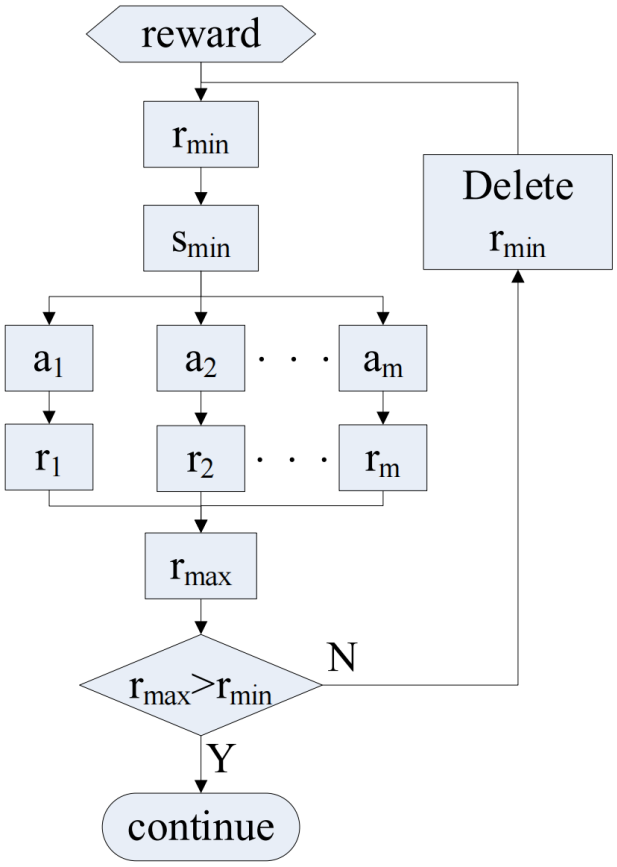


Figure 3. Error correction module.

Illustrated in Figure 3, this module unfolds in four distinct stages.

a) Identify the minimum reward value r_min and its corresponding state s_min from the set of reward values $reward$.

b) Calculate the reward r_i for each valid action a_i $(i = 1, 2, \ldots, m)$ according to equation (13).

c) Judgment: If the maximum reward value $r_max > r_min$, select this action to proceed with testing; otherwise, remove r_min from $reward$ and continue the computation.

d) Examine each reward value within $reward$ in sequence; if no adjustments are necessary, the test concludes.

Throughout the target data association evaluation, this module promptly identifies instances of association errors while bypassing the phase of environmental re-adaptation, thereby boosting the algorithm's applicability and efficiency.

5 Simulation Experiment and Result Analysis

5.1 Simulation Environment Settings

The sampling interval T_{sample} of radar sensor is 1, and the "window" N is 5. The detection probability P_D of target is 1. The initial positions are randomly distributed at (-500m, 500m), and the initial velocity is randomly selected between 30m/s and 100m/s. The maximum speed of target is $v_max = 150$m/s, and the minimum speed of target is $v_min = 10$m/s. The number of clutter follows Poisson distribution, and the mean of Poisson distribution is λ. The acceleration in CA model is randomly chosen between -10m/s^2 and 10m/s^2. The turning angular acceleration in CT model is randomly chosen between 0 and 0.8.

The following equation is the state transition equation, which is consistent with the motion process of the target:

$$X_k = FX_{k-1} + \Gamma v_{k-1} \tag{18}$$

where F is the state transition matrix, Γ is the process noise distribution matrix. v_{k-1} is the additive white noise, and its covariance matrix is $Q_{k-1} = \text{diag}([5^2, 5^2])$.

In the CA model, CV model and CT model, the F and Γ representations are shown below:

$$F_{CA} = \begin{bmatrix} 1 & T & \frac{T^2}{2} & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 \\ 0 & 0 & 1 & T & \frac{T^2}{2} & 0 \\ 0 & 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \Gamma_{CA} = \begin{bmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{19}$$

$$F_{CV} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \Gamma_{CV} = \begin{bmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \end{bmatrix} \tag{20}$$

$$F_{CT} = \begin{bmatrix} 1 & \frac{\sin \omega T}{\omega} & 0 & \frac{\cos \omega T - 1}{\omega} \\ 0 & \cos \omega T & 0 & -\sin \omega T \\ 0 & \frac{1 - \cos \omega T}{\omega} & 1 & \frac{\sin \omega T}{\omega} \\ 0 & \sin \omega T & 0 & \cos \omega T \end{bmatrix}, \quad \Gamma_{CT} = \begin{bmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \end{bmatrix} \tag{21}$$

The following equation is the state transition equation, which is consistent with the motion process of the target:

$$Z_k = HX_k + W_k \tag{22}$$

where H is the measurement matrix. W_k is the white Gaussian noise, and its covariance matrix is $R_k = \text{diag}([\sigma_v^2, \sigma_v^2])$. The measurement error σ_v is determined by sensor performance.

In the CA model, CV model and CT model, the H representations are shown below:.

$$H_{CA} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{23}$$

$$H_{CV} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{24}$$

$$H_{CT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{25}$$

5.2 Result Analysis

The target data association study comprises two distinct phases: the training phase and the evaluation phase. During the training phase, the target's motion duration is 30 seconds. The training dataset is organized into 5 subsets, each corresponding to a specific measurement error $\sigma_v = 0, 10, 20, 30, 40$, with 10,000 data entries per subset. The clutter parameter λ is randomly generated within the interval $(0, 100)$, and each trajectory's motion model is a random blend of CV, CA, and CT models.

In the evaluation phase, the target's motion spans 50 seconds. The evaluation dataset is segmented into 55 groups based on measurement error $\sigma_v = 0, 10, 20, 30, 40$ and clutter parameter $\lambda = 0, 10, 20, \ldots, 90, 100$, with each group containing 100 data points, reflecting 100 Monte Carlo simulation trials. The true trajectory is modeled using a combination of CV, CA, CT1, CT2, and CT3 models. Figure 4 depicts the actual trajectory, with the red line indicating the target's true path, "Start" marking the origin, and "End" denoting the destination. Two cases, $\sigma_v = 20, \lambda = 30$ and $\sigma_v = 40, \lambda = 100$, are used as examples to show the distribution of points, as shown in Figure 5.

The state transition matrix and process noise distribution matrix remain consistent for the CV and CA models, namely, $F_{CV}, \Gamma_{CV}, F_{CA}, \Gamma_{CA}$. For the CT models, F_{CT} and Γ_{CT} are predominantly stable, with only the parameter ω in F_{CT} undergoing changes, adhering to a predefined relationship that maps angular rates to distinct maneuver patterns:

$$\begin{cases} \omega_1 = 0.3 & \text{for CT1 model} \\ \omega_2 = -0.3 & \text{for CT2 model} \\ \omega_3 = 0.5 & \text{for CT3 model} \end{cases} \tag{26}$$

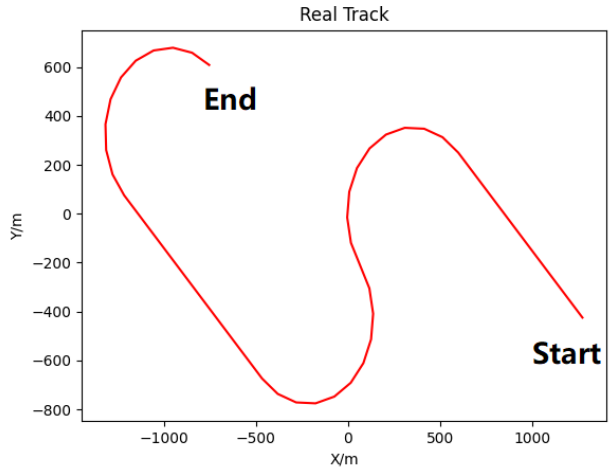
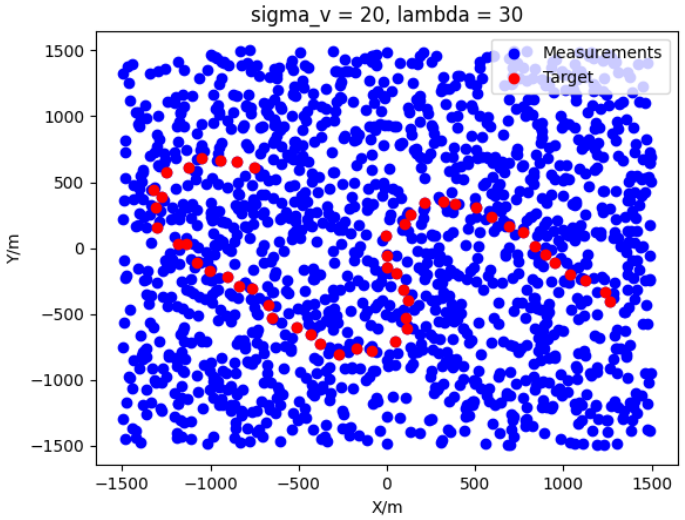
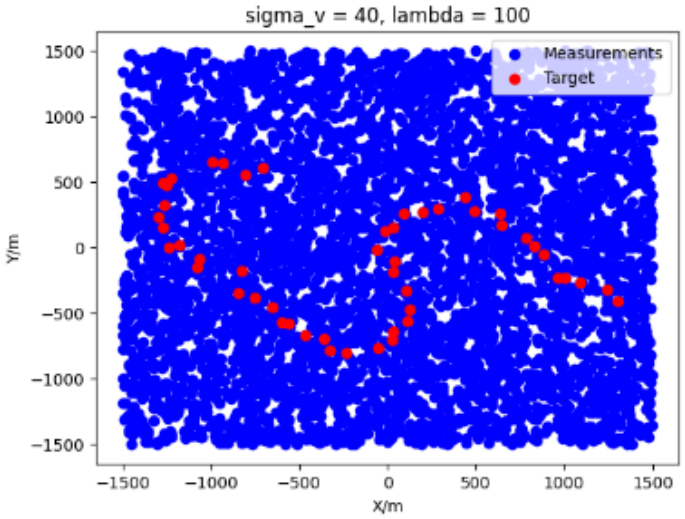


Figure 4. Real track.

The algorithm is designated as RLDA for short. Through adjustments made in the training phase, the learning rate of the network was set to 0.001, the discount factor to 0.1, and the reward decay to 0.9. Figures 6, 7 and 8 respectively illustrate the association results of the RLDA algorithm under the conditions of $\sigma_v = 0, 20, 40$ and $\lambda = 0, 10, 20, \ldots, 90, 100$. In the RLDA association diagrams, the red markers highlight the true measurements from the target, and the green lines depict the linked trajectories.



(a) $\sigma_v = 20,\ \lambda = 30$



(b) $\sigma_v = 40,\ \lambda = 100$

Figure 5. Measurement.

Figures 6, 7 and 8 reveal that as σ_v and λ increase, numerous incorrect associations appear in the results, causing the RLDA algorithm's accuracy to decline. These errors can be grouped into three main types. The first type involves sporadic instances of "mis-selection" during the association process, which do not interfere with subsequent steps. For example, this behavior is observed in the results of $\sigma_v = 0, \lambda = 100$, $\sigma_v = 20, \lambda = 90$, and others. All these figures demonstrate that the agent's future decisions remain unaffected by the mis-selection at the current moment. Moreover, this does not activate the adaptive adjustment mechanism, making it challenging for the agent to detect and rectify the error promptly. However, on a broader scale, this type of error has a negligible impact and does not disrupt the overall association process.

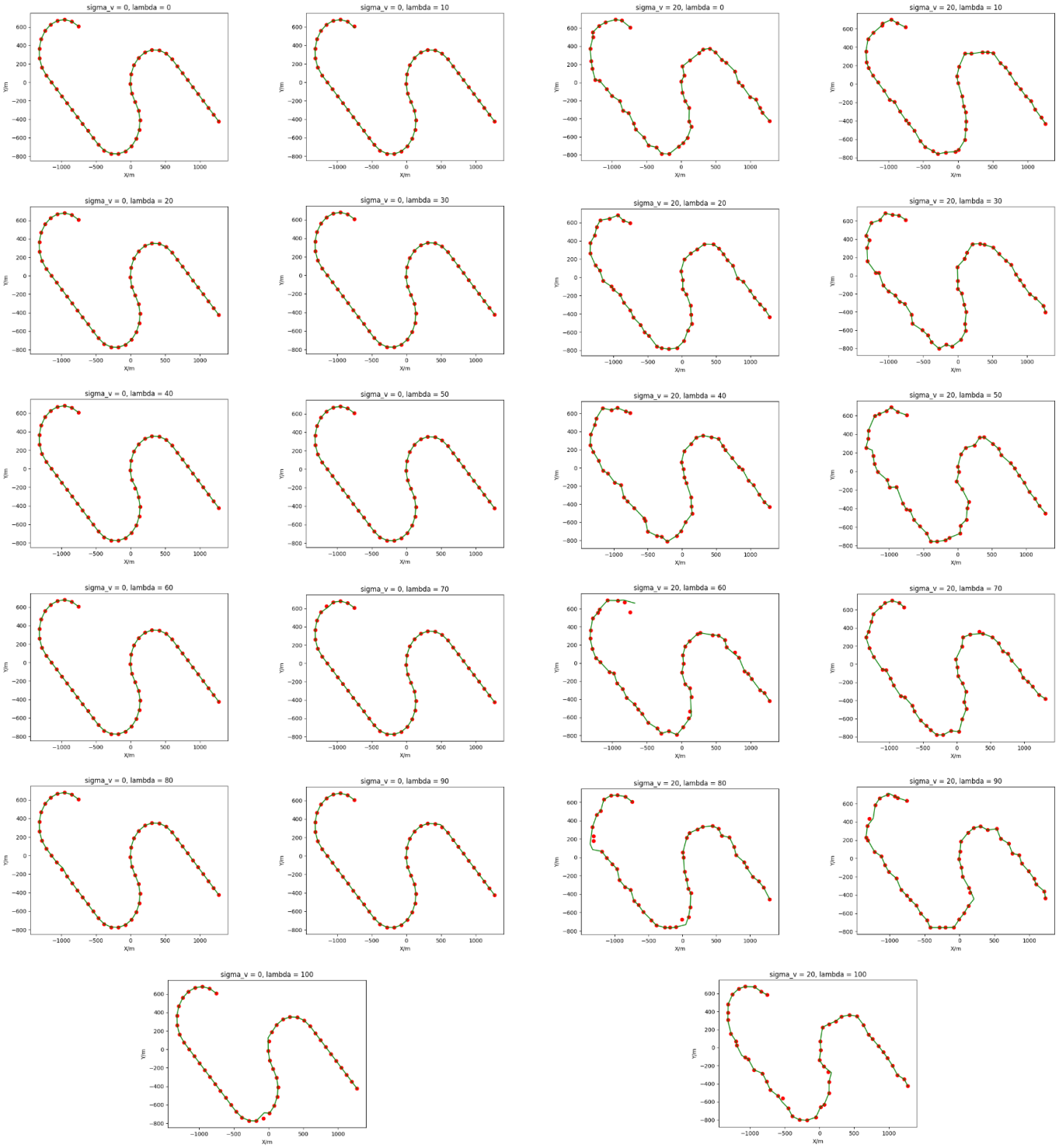


Figure 6. Association result of RLDA when σ_v=0.

Figure 7. Association result of RLDA when σ_v=20.

The second type refers to instances where a sequence of incorrect associations arises during a specific phase of the association process. This occurrence is strongly linked to σ_v, especially becoming more frequent during $\sigma_v = 40$. To explore the underlying causes, the RLDA association results for $\sigma_v = 40, \lambda = 100$ are examined in detail. As illustrated in Figure 9, three segments of erroneous associations are identified, labeled as "1", "2", and "3". Label "1" clearly aligns with the first category, and the associated points better reflect the target's motion trajectory compared to the actual target points, suggesting the agent's decision was more suitable. Labels "2" and "3" both fall into the second category, displaying significant mis-associations due to dense clutter and substantial sensor detection inaccuracies. Since the association process remains continuous and

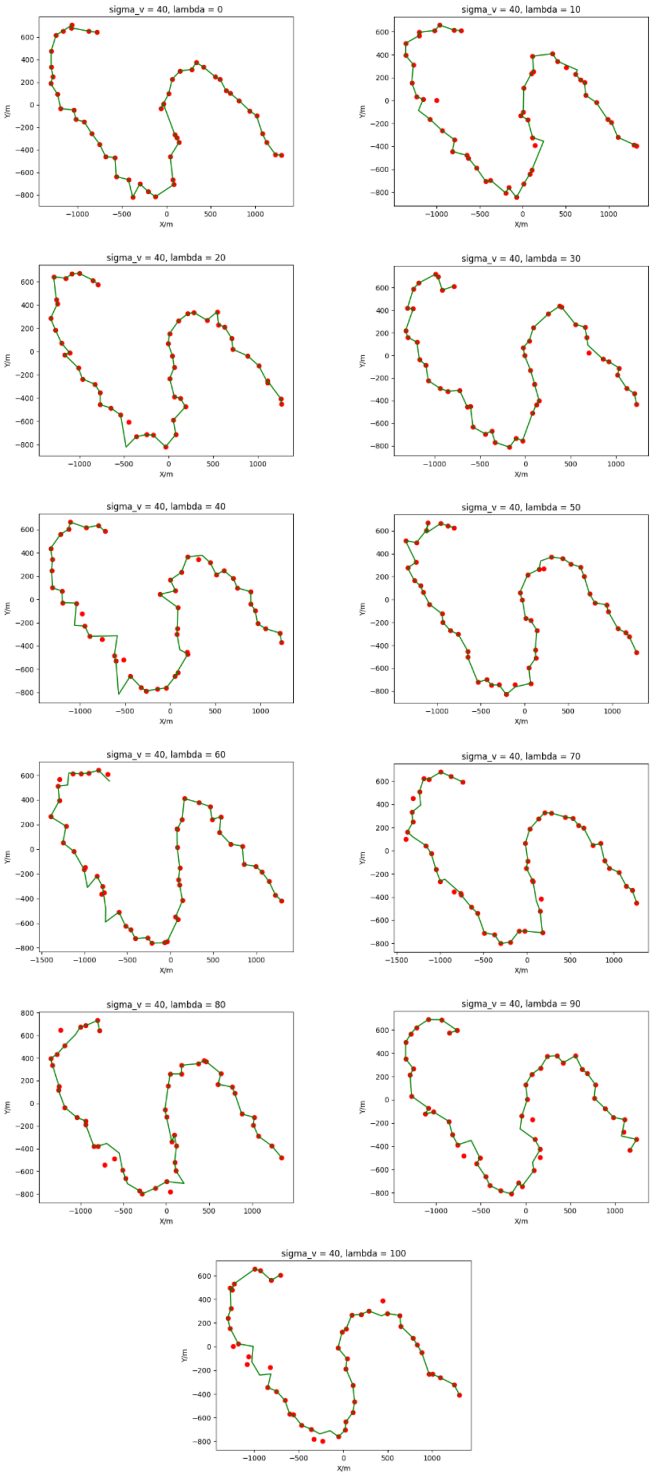


Figure 8. Association result of RLDA when σ_v=40, λ=100.

does not activate the adaptive adjustment mechanism, the agent cannot promptly identify and correct these errors. Nonetheless, as depicted in Figure 9, this type of error does not disrupt subsequent associations and has a negligible effect on the overall process.

The third type refers to association errors that might emerge as the process nears its conclusion. This behavior is observed in the outcomes of $\sigma_v = 20, \lambda = 60$, $\sigma_v = 40, \lambda = 60$, and similar cases. The cause lies in the target's intense maneuvers, which, under the influence of σ_v

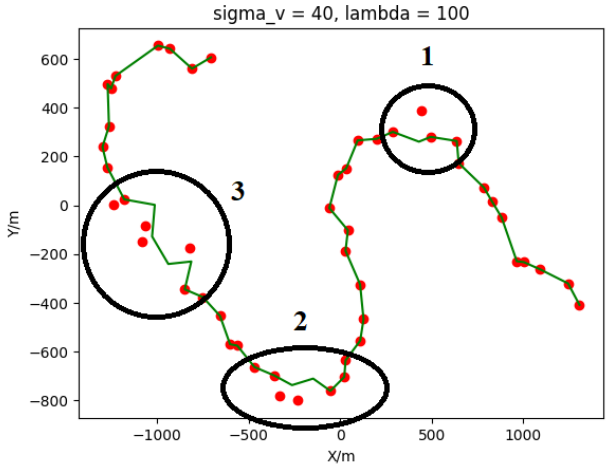


Figure 9. Association result of RLDA when σ_v=40, λ=100.

To summarize, as σ_v and λ grow larger, the RLDA algorithm's effectiveness diminishes, and incorrect associations may arise. Nonetheless, the results show that these issues do not compromise the entire process, with only a minor decline in the algorithm's overall performance, while the precision of the association outcomes remains strong. Additionally, this highlights that the RLDA algorithm is capable of effectively handling the association complexities of highly dynamic targets without depending on impractical assumptions, fulfilling the demands of real-world applications.

This research evaluates the RLDA algorithm against the Nearest Neighbor Data Association Filter (NNDAF) [28], the Probability Data Association Filter (PDAF), and the Point-track Association Method with Unknown System Model (USMA) [29]. Given that the simulation involves highly maneuvering targets, the IMM model is incorporated into NNDAF and PDAF, yielding the IMM-NNDAF and IMM-PDAF algorithms, referred to as NNDA and PDA, respectively. To mitigate the inherent limitations of these algorithms, the target's initial position and motion model are predefined. The simulation environment for the USMA algorithm mirrors that of the RLDA algorithm.

At the end of the test, the association accuracy P can be found by comparing the test results with the real points originating from the target, namely $P = \frac{N_{correct}}{N_{all}}$ (The number of real points originating from the target in the association result of agent is $N_{correct}$, and the number

of real points originating from the target is N_{all}, here $N_{all} = 50$)). For each group of data, the average association accuracy $\overline{P}$ can be calculated, namely $\overline{P} = \frac{1}{N_{data}} \sum_{i=1}^{N_{data}} P_i$ (The number of data in the test set is N_{data}, here $N_{data} = 100$). The experimental results of test are shown in Tables 1, 2 and 3.

Table 1. Performance comparison of $\bar{P}$ when $\sigma_v = 0$.

	$\bar{P}$	$\sigma_v = 0$			
		NNDA	PDA	USMA	RLDA
	0	1	0.731	1	1
	10	0.945	0.684	0.988	1
	20	0.831	0.639	0.960	1
	30	0.650	0.608	0.943	1
	40	0.424	0.555	0.928	0.996
λ	50	0.213	0.521	0.901	0.992
	60	0.204	0.477	0.887	0.982
	70	0.189	0.449	0.871	0.970
	80	0.114	0.407	0.852	0.964
	90	0.088	0.395	0.828	0.962
	100	0.058	0.358	0.801	0.951

Table 2. Performance comparison of $\bar{P}$ when $\sigma_v = 20$.

	$\bar{P}$	$\sigma_v = 20$			
		NNDA	PDA	USMA	RLDA
	0	1	0.5	1	1
	10	0.942	0.490	0.953	1
	20	0.821	0.464	0.932	1
	30	0.552	0.446	0.907	0.996
	40	0.381	0.444	0.871	0.988
λ	50	0.219	0.413	0.830	0.970
	60	0.198	0.432	0.806	0.962
	70	0.157	0.372	0.771	0.956
	80	0.081	0.401	0.738	0.951
	90	0.054	0.375	0.699	0.938
	100	0.088	0.371	0.665	0.919

The experimental results show that the $\overline{P}$ of RLDA algorithm is inversely proportional to σ_v and λ in terms of the overall trend. However, a closer look at the changes in the data reveals that as σ_v becomes larger, the effect of λ on $\overline{P}$ also becomes larger. When $\sigma_v = 0, 20, 40$, $\overline{P}$ decreases by 4.9%, 8.1% and 13.8% during the increase of λ from 0 to 100, respectively. Even so, $\overline{P}$ remains above 86%, reflecting the good association performance.

According to the association mechanism of NNDA algorithm, this algorithm is centered on determining the predicted value of track and directly selects the closest point to the center for association. As the results shown in Tables 1, 2 and 3, this algorithm is strongly influenced by clutter. $\overline{P}$ decreases by more than 90% as λ increases from 0 to 100. Compared to λ, this algorithm is minimally affected by σ_v.

Table 3. Performance comparison of $\bar{P}$ when $\sigma_v = 40$.

	$\bar{P}$	$\sigma_v = 40$			
		NNDA	PDA	USMA	RLDA
	0	1	0.22	1	1
	10	0.930	0.183	0.891	0.990
	20	0.733	0.207	0.864	0.987
	30	0.440	0.218	0.828	0.969
	40	0.314	0.170	0.790	0.955
λ	50	0.179	0.194	0.757	0.947
	60	0.158	0.191	0.712	0.935
	70	0.119	0.188	0.676	0.923
	80	0.093	0.167	0.644	0.909
	90	0.056	0.159	0.608	0.897
	100	0.044	0.145	0.537	0.862

As the results shown in Tables 1, 2 and 3, the performance of PDA algorithm gradually decreases as σ_v and λ become larger, and $\overline{P}$ decreases by about 60%.

The USMA algorithm is similar to RLDA algorithm in that both escape the limitations of traditional association algorithms and directly associate the points originating from the target in the real environment. However, the USMA algorithm does not consider the effect of σ_v. As shown in the Tables 1, 2 and 3, when $\sigma_v = 0, 20, 40$, $\overline{P}$ decreases by 19.9%, 33.5% and 46.3% during the increase of λ from 0 to 100. $\overline{P}$ can only stay above 50% and the association performance of this algorithm is ordinary.

In summary, the RLDA algorithm has the best association performance, followed by the USMA algorithm, and the PDA algorithm and the NNDA algorithm have poor performance. There are too many preconditions to implement the PDA algorithm and NNDA algorithm, but most of these assumptions cannot be predicted in advance (e.g., target motion model, number of targets, etc.). The USMA algorithm is not considered comprehensively enough, and the algorithm performance is unstable when facing the complex real environment.

6 Conclusion

In scenarios where the system model is uncertain, and challenges like intense clutter interference, sensor inaccuracies, and highly maneuvering targets arise, this study introduces a data association algorithm

leveraging the LSTM-RL network. This approach moves beyond conventional data association methods by integrating RL technology, creating a unique framework for data association. The architecture combines RL's dynamic exploration capabilities with the long-term memory functions of LSTM networks, enabling it to compute the association probability between the agent and selected measurement points. Target positions are predicted using Bayesian networks and multi-order least squares curve fitting, while reward values are derived from a Bayesian recursive function, providing direct feedback on network performance. Furthermore, practical mechanisms are designed by utilizing the characteristics of target and measurement data to ensure precise association between target points and tracks. The algorithm's effectiveness is demonstrated through comparative experiments with three advanced algorithms across diverse environments.

This work transcends traditional data association limitations, proposing an innovative solution to achieve accurate point-to-track alignment in real-world settings. It underscores the algorithm's practical engineering value and holds significant theoretical research importance. In future research endeavors, we will further deepen and expand upon the findings of this paper, with a particular focus on the exploration and resolution of multi-target track association challenges. Through ongoing technological innovation and methodological refinement, we are committed to achieving more groundbreaking progress in this field, aiming to deliver more precise and efficient solutions for relevant application scenarios.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Huang, X. P., & Wang, Y. (2015). Kalman filter principle and application.*Publishing House of Electronics Industry: Beijing, China*.

[2] Lexa, M., Coraluppi, S., Carthel, C., & Willett, P. (2020, March). Distributed MHT and ML-PMHT approaches to multi-sensor passive sonar tracking. In *2020 IEEE Aerospace Conference* (pp. 1-12). IEEE. [CrossRef]

[3] Li, Q., Song, L., & Zhang, Y. (2021). Multiple extended target tracking by truncated JPDA in a clutter environment. *IET Signal Processing, 15*(3), 207-219. [CrossRef]

[4] Fan, E., Xie, W., Pei, J., Hu, K., Li, X., & Podpečan, V. (2018). Improved joint probabilistic data association (JPDA) filter using motion feature for multiple maneuvering targets in uncertain tracking situations. *Information, 9*(12), 322. [CrossRef]

[5] He, S., Shin, H. S., & Tsourdos, A. (2020). Distributed multiple model joint probabilistic data association with Gibbs sampling-aided implementation. *Information Fusion, 64*, 20-31. [CrossRef]

[6] Ma, M., Wang, D., Sun, H., & Zhang, T. (2021). Radiation intensity Gaussian mixture PHD filter for close target tracking. *Signal Processing, 188*, 108196. [CrossRef]

[7] Qin, Z., Liang, Y., Li, K., & Zhou, J. (2021). Measurement-driven sequential random sample consensus GM-PHD filter for ballistic target tracking. *Mechanical Systems and Signal Processing, 155*, 107407. [CrossRef]

[8] Li, T., Prieto, J., Fan, H., & Corchado, J. M. (2018). A robust multi-sensor PHD filter based on multi-sensor measurement clustering. *IEEE Communications Letters, 22*(10), 2064-2067. [CrossRef]

[9] Streit, R., Angle, R. B., Efe, M., Streit, R., Angle, R. B., & Efe, M. (2021). Multi-Bernoulli Mixture and Multiple Hypothesis Tracking Filters. *Analytic Combinatorics for Multiple Object Tracking*, 113-143. [CrossRef]

[10] Leung, H. (1996). Neural network data association with application to multiple-target tracking. Optical Engineering, 35(3), 693-700. *Journal of Intelligent Systems, 31*(1), 1-14. [CrossRef]

[11] Tian, Y., Dehghan, A., & Shah, M. (2018). On detection, data association and segmentation for multi-target tracking. *IEEE transactions on pattern analysis and machine intelligence, 41*(9), 2146-2160. [CrossRef]

[12] Wadley, J. E., & Engebretson, K. R. (2022). Applying Artificial Intelligence to the Data Association Problem. In *AIAA SCITECH 2022 Forum* (p. 0248). [CrossRef]

[13] Zhang, X., Li, P., Zhu, Y., Li, C., Yao, C., Wang, L., ... & Li, S. (2021). Coherent beam combination based on Q-learning algorithm. *Optics Communications, 490*, 126930. [CrossRef]

[14] Li, H., Zhang, X., Bai, J., & Sun, H. (2021, April). Quadric Lyapunov Algorithm for Stochastic Networks Optimization with Q-learning Perspective. In *Journal of Physics: Conference Series* (Vol. 1885, No. 4, p. 042070). IOP Publishing. [CrossRef]

[15] Zhang, Y., Ma, R., Zhao, D., Huangfu, Y., & Liu, W.

(2021). A novel energy management strategy based on dual reward function Q-learning for fuel cell hybrid electric vehicle. *IEEE Transactions on Industrial Electronics, 69*(2), 1537-1547. [CrossRef]

[16] Li, M., Wang, Z., Li, K., Liao, X., Hone, K., & Liu, X. (2021). Task allocation on layered multiagent systems: When evolutionary many-objective optimization meets deep Q-learning. *IEEE Transactions on Evolutionary Computation, 25*(5), 842-855. [CrossRef]

[17] Zhao, B., Ren, G., Dong, X., & Zhang, H. (2021). Distributed Q-learning based joint relay selection and access control scheme for IoT-oriented satellite terrestrial relay networks. *IEEE Communications Letters, 25*(6), 1901-1905. [CrossRef]

[18] Zhang, Q., Lin, M., Yang, L. T., Chen, Z., & Li, P. (2017). Energy-efficient scheduling for real-time systems based on deep Q-learning model. *IEEE transactions on sustainable computing, 4*(1), 132-141. [CrossRef]

[19] Huang, R., Yu, T., Ding, Z., & Zhang, S. (2020). Policy gradient. *Deep Reinforcement Learning: Fundamentals, Research and Applications*, 161-212. [CrossRef]

[20] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *nature, 518*(7540), 529-533. [CrossRef]

[21] Liu, Y., Hu, Y., Gao, Y., Chen, Y., & Fan, C. (2019, August). Value Function Transfer for Deep Multi-Agent Reinforcement Learning Based on N-Step Returns. In *IJCAI* (pp. 457-463).

[22] Tirinzoni, A., Sessa, A., Pirotta, M., & Restelli, M. (2018, July). Importance weighted transfer of samples in reinforcement learning. In *International Conference on Machine Learning* (pp. 4936-4945). PMLR.

[23] Gamrian, S., & Goldberg, Y. (2019, May). Transfer learning for related reinforcement learning tasks via image-to-image translation. In *International conference on machine learning* (pp. 2063-2072). PMLR.

[24] Khan, A., Jiang, F., Liu, S., & Omara, I. (2019). Playing a FPS doom video game with deep visual reinforcement learning. *Automatic Control and Computer Sciences, 53*(3), 214-222. [CrossRef]

[25] Jithesh, V., Sagayaraj, M. J., & Srinivasa, K. G. (2017, February). LSTM recurrent neural networks for high resolution range profile based radar target classification. In *2017 3rd International conference on computational intelligence & communication technology (CICT)* (pp. 1-6). IEEE. [CrossRef]

[26] Li, B., & Yang, Y. (2018). Complexity of concept classes induced by discrete Markov networks and Bayesian networks. *Pattern Recognition, 82*, 31-37. [CrossRef]

[27] Wang, C., Wang, H. P., Xiong, W., & He, Y. (2016). Data association algorithm based on least square fitting. *Acta Aeronautica et Astronautica Sinica, 37*(5), 1603-1613.

[28] Faisal, S., & Tutz, G. (2021). Multiple imputation using nearest neighbor methods. *Information Sciences, 570*, 500-516. [CrossRef]

[29] Wei, X., Gu, X., & Yaqi, C. (2022). Point-track association method with unknown system model. *IET Radar, Sonar & Navigation, 16*(11), 1779-1795. [CrossRef]



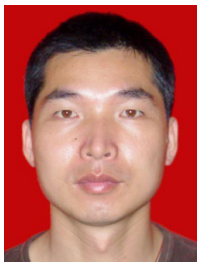
Xiangqi Gu received the Eng.D degree in information and communication engineering from Naval Aviation University, China, in 2023. (Email: guxiangqi1314@163.com)



Ziran Ding received the Eng.M degree in information and communication engineering from Naval Aviation University, China, in 2016. (Email: yylx@vip.qq.com)



Shutao Xia received the Eng.M degree in communication and information system from Naval Aviation Engineering University, China, in 2008. (Email: 1123103211@qq.com)



Wei Xiong received the Eng.D degree in communication and information system from Naval Aviation Engineering University, China, in 2005. (Email: xiongwei@csif.org.cn)