Trajectory Generation and Dynamic Continuous Activity Recognition for Radar Swarm Targets

Zhiyuan Zou, Qing Miao, Jianwei Wei, Yiru Lin, Xinwei Wei and Wei Yi
School of Information and Communication Engineering
University of Electronic Science and Technology of China
Chengdu, P.R.China
Email: zhiyuan_zou417@163.com, miaoqing313@163.com, weijianwei408@outlook.com,
linyiru2021@163.com, Syinwei.Wei@gmail.com, kussoyi@gmail.com

***Abstract*—The swarm targets have shown great potential for both military and civilian applications, driving a high demand for reliable trajectory generation and accurate activity recognition. In this paper, we propose a trajectory generation method and establish an end-to-end deep learning model for dynamic continuous activity recognition of swarm targets. First, we devise an activity transition model of the drone swarm based on a continuous-time Markov chain (CTMC). Subsequently, the minimum snap trajectory generation algorithm is employed to generate the trajectories. After that, to recognize the dynamic continuous activity of targets, we develop an end-to-end neural network model to extract spatial and temporal features for swarm targets detected by radar across multiple frames. Finally, we demonstrate the effectiveness and robustness of our proposed method through simulation results.**

***Index Terms*—Swarm targets; trajectory generation; activity recognition; deep learning**

I. Introduction

Swarms have gained widespread application in various scenarios, such as military operations or entertainment shows [1]. Swarm-type targets are formed when similar-sized individuals are clustered together and operate in the same action mode, such as fishes, birds, persons, and drones [2], [3]. In contrast to the single target, swarm targets exhibit morphological diversity, numerosity, and self-organization [4]. Their ability to coordinate activity patterns among individuals poses significant management challenges [5], [6]. Consequently, the generation of realistic trajectories and accurate recognition of swarm activities are significant for military defense and public safety.

The reliable generation of flight trajectory for drone swarms, a typical swarm target, is a challenging task. In [7], a minimum snap trajectory generation method is developed for drones operating under indoor environmental constraints. However, this approach fails to address collisions among drones. To solve this, mixed-integer constraints are introduced into the piecewise polynomial functions in [8], aiming to prevent collisions. Furthermore, to ensure efficient sparse computation of higher-order polynomials for drone trajectories, a novel method is proposed in [9], which jointly optimizes polynomial path segments within an unconstrained quadratic program (QP). Nevertheless, these algorithms neglect the problem of trajectory generation for the large swarm of drones and fail to account for the transition of drone swarm activities.

As a primary sensing device, radar has distinct advantages such as all-weather, round-the-clock, and long-range capabilities, surpassing the image and infrared sensors [10]. Therefore, robust activity recognition of the swarm using radar represents a desirable solution. Internal target motion generates micro-Doppler features in radar echoes, and by leveraging a deep learning network, these features can be effectively utilized for target activity recognition [11]–[15]. In addressing the challenge of dynamic continuous activity recognition, developing a network capable of extracting multi-frame range-Doppler image information stands out as a compelling approach [16], [17]. This approach aims to comprehensively capture the range, velocity, and temporal features in radar echoes. However, none of the above methods take into account the activity recognition of swarm targets.

In this paper, we propose a trajectory generation method and a dynamic continuous activity recognition method for the large drone swarm. The principal contributions of this paper are as follow:

- Departing from conventional methods, we propose an innovative trajectory generation approach that combines a continuous-time Markov chain (CTMC) model for activity transition with a minimum snap trajectory generation algorithm. This novel method allows for efficient activity transition, thereby contributing a unique solution for the trajectory generation of drone swarms.
- An end-to-end deep learning framework integrating ResNet, Long Short-Term Memory (LSTM) network, and the connectionist temporal classification (CTC) algorithm is proposed for dynamic continuous activity recognition. Our network effectively captures the spatial and temporal features for radar echoes across multiple frames.

Simulation results are given to validate the robustness and effectiveness of the proposed approach.

This work was supported in part by the National Natural Science Foundation of China under Grant 62231008 and 62301127, in part by the China Postdoctoral Science Foundation under Grant BX20220057, 2023M730509 and GZB20230112, the "Tianfu Qingcheng" Plan of Sichuan Province under Grant1332 and 1395.

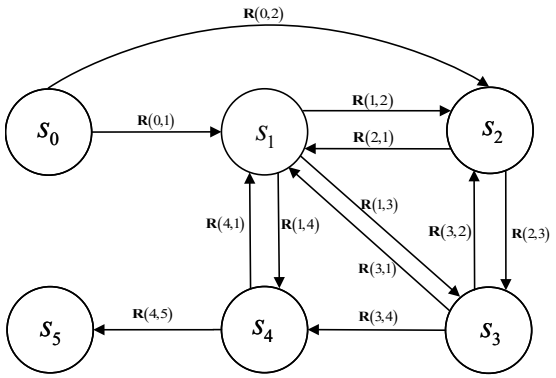

Fig. 1. The CTMC state transition model.

II. Trajectory Generation of the Swarm

In this section, we propose a novel trajectory generation approach that includes various activities for a large drone swarm. In contrast to the previous work [7]–[9], our method accounts for the drone state transition model between different activities and analyzes the minimum snap trajectory generation algorithm for drone swarms. This approach can provide sufficient data samples for the training of the activity recognition network.

A. Activity Transition Model

The CTMC is used to describe the evolution of probabilities over continuous time [18]. According to the CTMC, the activity state of a drone swarm at a given moment is known, and the state after a time period depends only on the present and is independent of the past.

Definition 1: A CTMC $\mathcal{M}$ is a tuple $(S, s_{init}, \mathbf{R}, L)$ with S as a finite set of states, $s_{init} \in S$ as the initial state, $\mathbf{R}$ as the transfer rate matrix, and L as the labeling function.

If $\mathbf{R}(i,j) > 0$, denotes that the state s_i can transfer to the state s_j, the probability that the transfer occurs within T time units is $1 - e^{-\mathbf{R}(i,j)\cdot T}$. Given that multiple transitions from state s_i are possible, the possibility of transitioning from a non-absorbing state s_i to a specific state s_j within a duration of T time units is expressed as follows:

$$\mathbf{P}\left(i,j,T\right) = \frac{\mathbf{R}\left(i,j\right)}{\mathbf{E}(i)} \cdot \left(1 - e^{-\mathbf{E}(i)\cdot T}\right), \tag{1}$$

where $\mathbf{E}(i) = \sum_{s_j \in S} \mathbf{R}\left(i,j\right)$ denotes the sum of the rates of transfer out of state s_i.

Considering an unlabeled CTMC, there are six activities of a swarm in our CTMC-based activity transition model, which are:

- s_0: Converging on a straight line;
- s_1: Hovering;
- s_2: Flying on a straight line;
- s_3: Flying back and forth;
- s_4: Encircling;
- s_5: Attacking on the target point,

where the $s_{init} = s_0$ is the starting state and the s_5 is the absorbing state. The details of the state transition model are shown in Fig. 1.

Through the above theoretical analysis, we can establish a CTMC-based activity transition model for a drone swarm. According to the transitions of different activities, the waypoints are set on the trajectory space to realize the transition between drone target states.

In the initial configuration, a swarm of N drones is generated at a set of random waypoints $\{(x_{g1}, y_{g1}), (x_{g2}, y_{g2}), ..., (x_{gN}, y_{gN})\}$. To achieve the initial state s_0, the drones can transition from their generation points to $\{(x_{c1}, y_{c1}), (x_{c2}, y_{c2}), ..., (x_{cN}, y_{cN})\}$. Subsequently, the drones have the option to traverse between two sets of parallel waypoints, $\{(x_{b1}, y_{b1}), (x_{b2}, y_{b2}), ..., (x_{bN}, y_{bN})\}$ and $\{(x_{f1}, y_{f1}), (x_{f2}, y_{f2}), ..., (x_{fN}, y_{fN})\}$, to reach either state s_2 or s_3. Notably, the state s_1 involving hovering is achieved as the drones rotate around the above parallel points. Furthermore, the drones could achieve a state of encircling s_4 by reaching the encirclement points $\{(x_{e1}, y_{e2}), (x_{e2}, y_{e2}), ..., (x_{eN}, y_{eN})\}$. The activity sequence concludes when the drone swarm reaches the attack point (x_a, y_a).

B. Minimum Snap Trajectory Generation

Trivial trajectories, which satisfy the given constraints, are generated by interpolating straight lines between waypoints. Nevertheless, this method of trajectory generation is inefficient due to the infinite curvature exhibited by drones at each waypoint. However, it is convenient to write the trajectory $\mathbf{r}(t) = [x_u(t), y_u(t)]^\top$ of each drone as a polynomial function of order m over k time intervals [8], which is as follows:

$$\mathbf{r}(t) = \begin{cases} \sum_{i=0}^{m} \mathbf{r}_{i1} t^i & t_0 \le t < t_1 \\ \sum_{i=0}^{m} \mathbf{r}_{i2} t^i & t_1 \le t < t_2 \\ \quad \vdots & \\ \sum_{i=0}^{m} \mathbf{r}_{ik} t^i & t_{k-1} \le t \le t_k. \end{cases} \tag{2}$$

Polynomial trajectories can be efficiently derived as the optimal solution to a QP problem that aims to minimize the cost function of path derivatives. The above problem (2) can be formulated as a QP by writing the Cartesian coordinate system position of a drone in constants as a $2mk \times 1$ decision variable vector $\mathbf{p}$. The cost function, which imposes a penalty on the squared derivatives of $\mathbf{p}$, can be expressed as:

$$\begin{aligned} \mathbf{J} &= \int_0^T a_0 \mathbf{p}^2 + a_1 \mathbf{p}^{'2} + a_2 \mathbf{p}^{''2} + \ldots + a_n \mathbf{p}^{(n)2} dt \\ &= \mathbf{p}^T \mathbf{Q} \mathbf{p}, \end{aligned} \tag{3}$$

where the construction of the hessian matrix $\mathbf{Q}$ relies on differentiating the square of the polynomial concerning its coefficients.

We employ the mapping matrix $\mathbf{A}$ to establish constraints on the i-th segment of the trajectory. This matrix establishes a relationship between the polynomial coefficients and the

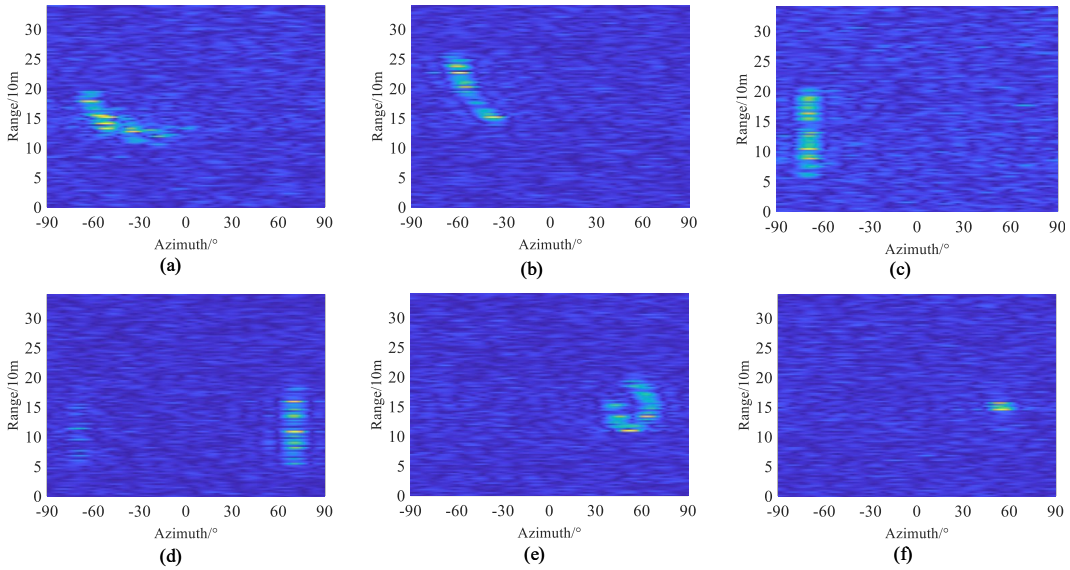

Fig. 2. The radar RA spectrum containing a cluster of a hundred drones for the different states. (a) Converging. (b) Hovering. (c) Flying on a straight line. (d) Flying back and forth. (e) Encircling. (f) Attacking.

derivatives of the trajectory endpoints. The constraints can be reformulated as:

$$\mathbf{A}_i\mathbf{p}_i = \mathbf{d}_i, \quad \mathbf{A}_i = \begin{bmatrix} \mathbf{A}_0 \\ \mathbf{A}_k \end{bmatrix}_i, \quad \mathbf{d}_i = \begin{bmatrix} d_0 \\ d_k \end{bmatrix}_i, \tag{4}$$

where $\mathbf{d}_i$ denotes a derivative value vector, and $\mathbf{A}_i$ is an element of the block diagonal matrix $\mathbf{A}$.

Then we can substitute the constraints into the initial cost function:

$$\mathbf{J} = \mathbf{d}^T\mathbf{A}^{-T}\mathbf{Q}\mathbf{A}^{-1}\mathbf{d}. \tag{5}$$

The decision variable for the revised quadratic cost function is now the segmented trajectory endpoint derivatives. We categorize the decision variables into two groups: fixed derivatives $\mathbf{d}_F$ and free derivatives $\mathbf{d}_P$. The reordering is accomplished using a substitution matrix $\mathbf{C}$ consisting of zeros and ones. Then we can have:

$$\begin{aligned} \mathbf{J} &= \begin{bmatrix} \mathbf{d}_F \\ \mathbf{d}_P \end{bmatrix}^T \underbrace{\mathbf{C}\mathbf{A}^{-T}\mathbf{Q}\mathbf{A}^{-1}\mathbf{C}^T}_{R} \begin{bmatrix} \mathbf{d}_F \\ \mathbf{d}_P \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{d}_F \\ \mathbf{d}_P \end{bmatrix}^T \begin{bmatrix} \mathbf{R}_{FF} & \mathbf{R}_{FP} \\ \mathbf{R}_{PF} & \mathbf{R}_{PP} \end{bmatrix} \begin{bmatrix} \mathbf{d}_F \\ \mathbf{d}_P \end{bmatrix}, \end{aligned} \tag{6}$$

where $\mathbf{R}$ is the new augmented cost matrix and divides it according to fixed and free derivatives. Differentiating J and equaling 0 yields the optimal vector of free derivatives in terms of the new augmented cost matrix and the fixed derivatives:

$$\mathbf{d}_P^* = -\mathbf{R}_{PP}^{-1}\mathbf{R}_{FP}^T\mathbf{d}_F \tag{7}$$

The polynomials can be recovered from the constraint equations mapping from the derivative space back to the coefficient space.

A swarm of drones must modify their trajectories in order to navigate independently in a shared space and avoid collision problems. We enforce a safe distance between the drones, which can be mathematically represented by the following set of constraints for drones 1 and 2:

$$\begin{aligned} \forall t_k: \quad & x_{u1}(t_k) - x_{u2}(t_k) \le d_x \\ \text{and } & y_{u1}(t_k) - y_{u2}(t_k) \le d_y, \end{aligned} \tag{8}$$

where d_x and d_y denote the safe distance in the x-axis direction and y-axis direction respectively. We incorporate constraints (8) into the multi-drone version of (2). Finally, we obtained the drone swarm trajectories by summing the polynomials in (2).

III. Radar Signal Models for Drone Swarms

The detection of the drone swarm that is flying on a defined state s with radar allows for the generation of their corresponding signals, which facilitates the subsequent analysis of the activity of the swarm.

Suppose that the transmit pulse of the pulsed radar is a chirp signal, and the transmit pulse $S(t)$ is denoted as

$$S(t) = \text{rect}\left(\frac{t}{T_r}\right)\exp\left(j2\pi f_0 t + j\pi K_r t^2\right), \tag{9}$$

where t is the range fast time, T_r denotes the pulsewidth, $\text{rect}(\cdot)$ represents a rectangular window, and K_r serves the chirp rate.

Considering there are N drones in a swarm, the position at time t of the i-th drone relative to the radar in the Cartesian coordinate system generated in Section II is $[x_{ui}(t), y_{ui}(t)]$. The angle θ_i relative to the radar is $\arctan[x_{ui}(t)/y_{ui}(t)]$, and the range r_i relative to the radar is $\left(\sqrt{x_{ui}(t)^2 + y_{ui}(t)^2}\right)$. The time delay t_i of the i-th drone can be expressed as $2r_i/c$.

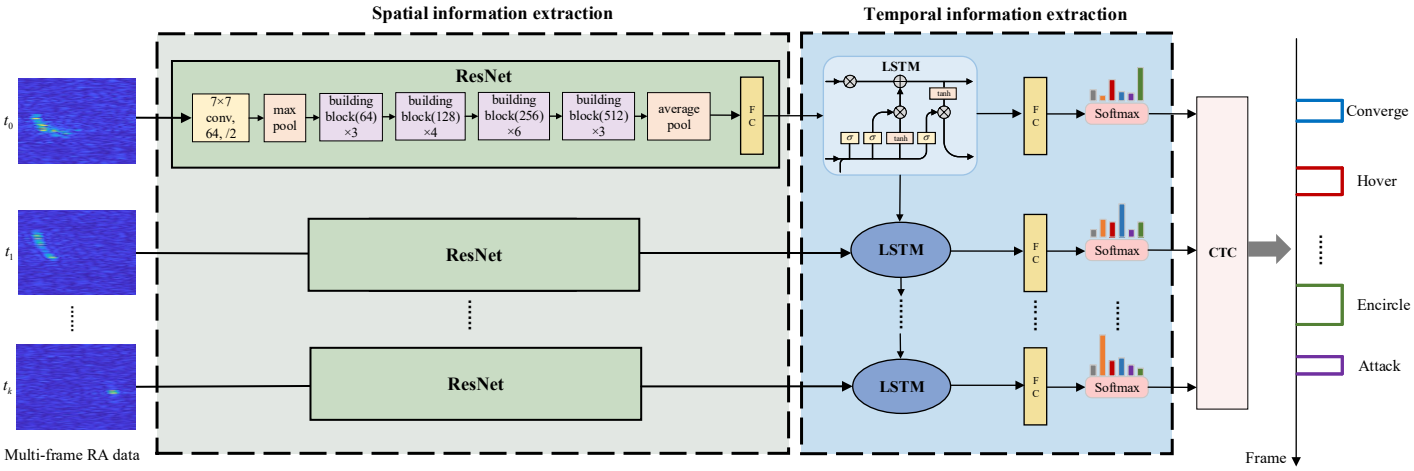

Fig. 3. The neural network architecture. The input to the end-to-end deep neural network is multi-frame RA spectrum data and actual labels of the activity sequence. The ResNet extracts the spatial information, the LSTM extracts the temporal information, and finally, the CTC loss accurately maps the unsegmented activity sequence and the output labeled sequence.

We assume that a radar echo comprises a vectorial superposition of echoes emanating from multiple targets with noise [19], which can be expressed as

$$S(t,\tau)=\sum_{i=1}^{N}u_i\cdot H\left(\tau-\tau_i\right)\operatorname{rect}\left(\frac{t-t_i}{T_r}\right)\cdot\exp\left(j2\pi f_0\left(t-t_i\right)+j\pi K_r\left(t-t_i\right)^2\right)+s_n(t,\tau), \tag{10}$$

where u_i represents the scattering coefficient of the i-th target, τ denotes the azimuthal slow time, τ_i is the scanning time of the i-th target, $H(\tau)$ serves the modulation of antenna pattern, and $s_n(t,\tau)$ signifies the noise.

After eliminating the carrier frequency and pulse compression operations, we transform the echo signal to the range-angle (RA) domain, considering $t_i=(\theta-\theta_i)/\omega$, where ω is the scanning speed, (10) can be expressed as follows:

$$S(r,\theta)=\sum_{i=1}^{N}u_i\cdot T_r\cdot H\left(\theta-\theta_i\right)\cdot\operatorname{sinc}\left[\frac{2B}{c}\left(r-r\left(\theta_i\right)\right)\right]\cdot\exp\left(-j4\pi\frac{r\left(\theta_i\right)}{\lambda_0}\right)+s_n(r,\theta). \tag{11}$$

Perform a two-dimensional Fourier transform on the echo signal to obtain the radar RA spectrum $\mathbf{I}$, shown in Fig. 2. Radar echoes from swarm targets show differences in different activity states.

IV. Deep Neural Network Model

To recognize the dynamic continuous activities of the drone swarm, we propose an end-to-end deep learning framework by combining ResNet, LSTM, and a CTC loss. The network architecture is clearly shown in Fig.3.

We use a set of multi-frame radar RA spectrum $\{\mathbf{I}_i\}_{1\le i\le M}$ as input to capture richer contextual information. The outputs of the network are predictive activity label sequences.

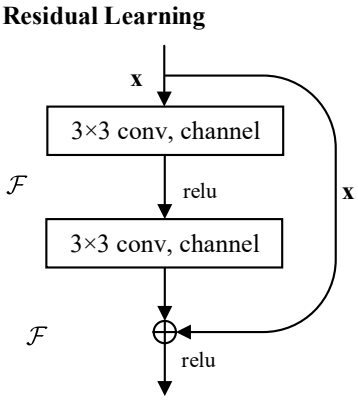

Fig. 4. Residual module with shortcut connection.

A. Spatial Information Extraction

We first use ResNet to extract spatial features for multi-frame radar RA spectrum $\{\mathbf{I}_i\}_{1\le i\le M}$. Since there are fewer resolution cells in the radar spectrum, we expect to use deeper convolutional neural networks (CNNs) to extract target features. However, a previous study found that as the network's depth grows, the training accuracy decreases after reaching saturation and exposes a degradation problem [20]. To overcome this, a residual neural network named ResNet is proposed to optimize the training of CNN [21].

ResNet addresses the degradation problem by adding a deep learning framework, including identity mapping, as represented by Fig. 4. The residual learning is implemented by defining a building block with weights W_i as

$$\mathcal{H}(\mathbf{x})=\mathcal{F}\left(\mathbf{x},\{W_i\}\right)+\mathbf{x}, \tag{12}$$

where the underlying mapping that meets the expectation is denoted as $\mathcal{H}(\mathbf{x})$, but it is more difficult to let the stacked layers directly fit $\mathcal{H}(\mathbf{x})=\mathbf{x}$. The task of learning the original mapping can be construed as learning the residual

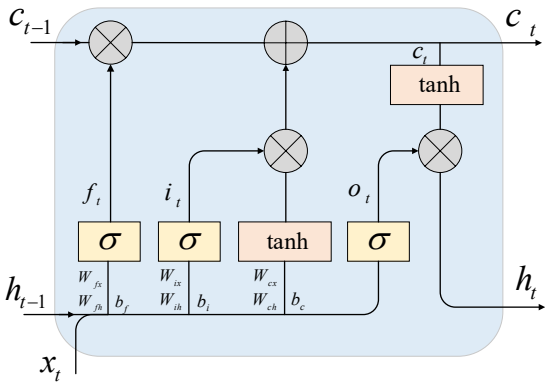

Fig. 5. The structure of a LSTM cell.

function, defined as $\mathcal{F}(\mathbf{x}) = \mathcal{H}(\mathbf{x}) - \mathbf{x}$. Furthermore, the solver addresses the degradation issue of the neural network as the depth increases by simply setting the weights of multiple nonlinear layers to zero.

In this paper, we implement ResNet34, a 34-layer residual network, to extract spatial features from radar RA spectrum data, aiming to solve the degradation issue during deep CNN training.

B. Temporal Information Extraction

LSTM is a network model for extracting temporal features [22]. LSTM controls the rate of information accumulation by introducing a gating mechanism, which includes selectively adding new information or forgetting previously accumulated information. The detailed cell structure of LSTM is illustrated in Fig. 5.

The LSTM implements three gate computations, i.e., the forgetting gate $\mathbf{f}$, the input gate $\mathbf{i}$, and the output gate $\mathbf{o}$, which are used to protect and control the cell state $\mathbf{c}$. The forget gate $\mathbf{f}$ determines the extent to which the cell state $\mathbf{c}_{t-1}$ from the preceding moment is carried forward to the current moment's state $\mathbf{c}_t$, i.e., it decides what information is discarded from the cell state $\mathbf{c}$, which can be expressed as:

$$\mathbf{f}_t = \sigma\left(W_{fx}\mathbf{x}_t + W_{fh}\mathbf{h}_{t-1} + b_f\right), \tag{13}$$

$$\mathbf{c}_t = \mathbf{f}_t * \mathbf{c}_{t-1} + \mathbf{i}_t * \tanh\left(W_{cx}\mathbf{x}_t + W_{ch}\mathbf{h}_{t-1} + b_c\right), \tag{14}$$

where σ represents the sigmoid function, W stands for the weight, $\mathbf{x}$ signifies the input, and b serves the bias. Furthermore, the input gate $\mathbf{i}$, the output gate $\mathbf{o}$, and the output vector $\mathbf{h}$ of the LSTM can be formulated as follows:

$$\mathbf{i}_t = \sigma\left(W_{ix}\mathbf{x}_t + W_{ih}\mathbf{h}_{t-1} + b_i\right), \tag{15}$$

$$\mathbf{o}_t = \sigma\left(W_{ox}\mathbf{x}_t + W_{oh}\mathbf{h}_{t-1} + b_o\right), \tag{16}$$

$$\mathbf{h}_t = \mathbf{o}_t * \tanh\left(\mathbf{c}_t\right), \tag{17}$$

where the input gate $\mathbf{i}$ is responsible for deciding how much new information can be used, the output gate $\mathbf{o}$ determines how many outputs are available for the unit state at the current moment in time, and $\mathbf{h}$ denotes the output vector of the LSTM.

C. CTC Loss

The features extracted from the LSTM layer are subsequently fed into a softmax layer, enabling the generation of predictive labels for each discrete time point. Our network structure employs the CTC algorithm as a loss function, enabling a precise alignment between predicted activity sequences and the corresponding actual labels [23].

Let $\mathcal{L} = \{s_0, s_1, s_2, s_3, s_4, s_5\}$ be the label dictionary of activities designed in section II. First, the CTC algorithm expands the initial dictionary with a blank class as $\mathcal{L} = \mathcal{L} \cup \{\text{blank}\}$. The network computes the probability of observing a particular activity state s (or no activity) at time t in an input sequence $\mathbf{O}$ can be expressed as $p_t\left(s \mid \mathbf{O}_\text{s}\right) = e_t^s$. Subsequently, we define π as the mapping path that connects the input sequence $\mathbf{O}_\text{s}$ with the corresponding actual label $\mathbf{L}$. The probability of a path π can be obtained by integrating the probabilities of all predicted labels on the path at each moment: $p\left(\pi \mid \mathbf{O}_\text{s}\right) = \prod_t e_t^{\pi_t}$.

The sequence of labels can be generated by defining the compression operation $\mathcal{B}$, which removes the blank label and condenses the repeated labels [24]. For example, one possible compression process is $\mathcal{B}(-s_0 s_0 - s_2 - s_5) = \{s_0 s_2 s_5\}$, in which "$-$" is the blank label. By further combining all possible paths of the actual labeled sequence $\mathbf{L}$ can be produced, the conditional probability of the unsegmented sequence of activities can be expressed as:

$$p\left(\mathbf{L} \mid \mathbf{O}_\text{s}\right) = \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{L})} p\left(\pi \mid \mathbf{O}_\text{s}\right), \tag{18}$$

where $\mathcal{B}^{-1}(\mathbf{L})$ denotes all the paths where we can get the label $\mathbf{L}$. Finally, the CTC loss function can be derived as $\ln\left(p\left(\mathbf{L} \mid \mathbf{O}_\text{s}\right)\right)$.

V. Simulation Result

In this section, the proposed algorithm is evaluated and analyzed via simulated data. We will use the pre-segmentation ResNet-LSTM method [25] to compare with the proposed approach.

A. Result of Trajectory Generation

We assume that there are six activities of a swarm containing 100 drones formulated to fly in a two-dimensional plane field. According to one of the paths derived from the CTMC, we position waypoints across the field, as shown in Fig. 6(a). Utilizing the minimum snap trajectory generation algorithm, which employs a polynomial function of order $m = 6$, we generate the two-dimensional planar trajectory for this large drone swarm. The two-dimensional planar trajectory of this large drone swarm is shown in Fig. 6(b).

This comprehensive simulation enables us to assess the performance of our proposed recognition approach in realistic drone swarm scenarios.

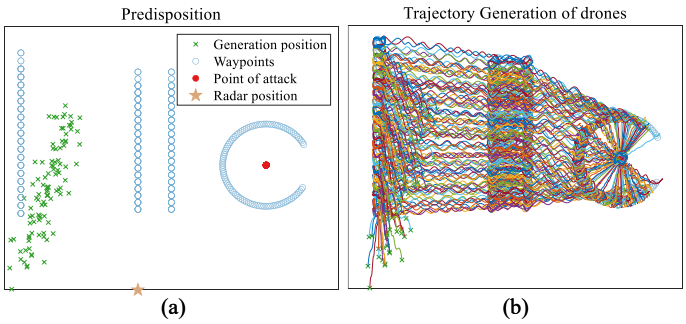

Fig. 6. The plane field of the trajectory generation result. (a) Predisposition plane. (b) Trajectory generation of drones.

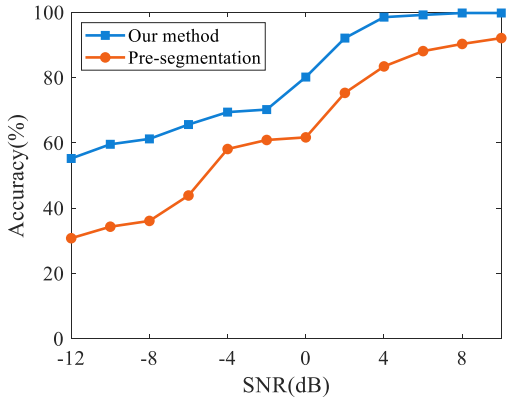

Fig. 7. The recognition accuracy at each SNR.

B. Model Training

Actual label sequence $\mathbf{L}$ with a set of M multi-frame RA spectrum $\{\mathbf{I}_i\}_{1\leq i\leq M}$ are used as inputs to the network. The ratio of training and test sets is divided into 4:1. The model was trained for 100 epochs with Adam optimizer and a learning rating of 10^{-4}.

To verify the robustness of the proposed method to noise, we process the test set to obtain the accuracy at different signal-to-noise ratios (SNRs) according to (20).

$$Acc = \frac{TN + TP}{TN + TP + FP + FN}, \tag{19}$$

where TN is true negative, TP denotes true positive, FP represents false positive, and FN is false negative.

C. Result and Analysis of Recognition

As can be seen in Fig. 7, the recognition accuracy increases with SNR increasing. Notably, our proposed approach consistently outperforms the comparison method across various SNRs.

To visualize the network recognition results, we labeled the recognition probability of different activities at each frame for one test result. The hand-labeled ground truth is shown in Fig. 8(a), providing a benchmark for comparison. Meanwhile, the CTC algorithm effectively compresses repeated activity labels within the time slots into concise peaks, simplifying the recognition process, as can be seen in Fig. 8(b).

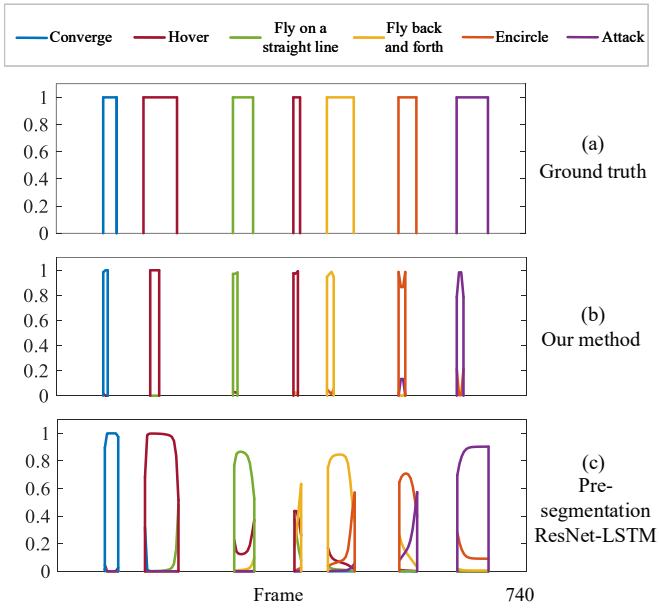

Fig. 8. The recognition probability of an activity sequence at each frame. (a) The ground truth. (b) Our method. (c) Pre-segementation ResNet-LSTM.

The pre-segmentation ResNet-LSTM method labels the truth value for each frame of radar data and calculates the loss using the cross-entropy function. Through further comparison of the result shown in Fig. 8(c), the comparison method has poor recognition during activity transitions. This is primarily attributed to the challenges associated with accurately distinguishing the boundaries between different activities. In contrast, our approach demonstrates superior performance in capturing these transitions, highlighting its effectiveness in recognizing complex drone swarm activities.

VI. Conclusion

In this paper, we propose a trajectory generation method for the large drone swarm based on a CTMC activity transition model. The trajectories generated by this method are effective and help to ensure the authenticity of subsequent activity recognition efforts. After that, an end-to-end deep learning network can extract spatial and temporal feature information from the multi-frame radar RA spectrum for robust activity recognition. The simulation results demonstrate the robustness and effectiveness of our approach.

As further research work, we plan to expand the activities of the drone swarm and solve problems in higher dimensions to obtain accuracy gains.

References

[1] Y. Wei, M. B. Blake, and G. R. Madey, "An operation-time simulation framework for uav swarm configuration and mission planning," *Procedia Computer Science*, vol. 18, pp. 1949–1958, 2013.

[2] R. Bouffanais, *Design and control of swarm dynamics*. Singapore: Springer, 2016, vol. 1.

[3] H. Hildenbrandt, C. Carere, and C. K. Hemelrijk, "Self-organized aerial displays of thousands of starlings: a model," *Behavioral Ecology*, vol. 21, no. 6, pp. 1349–1359, 2010.

[4] Y. Wei, M. B. Blake, and G. R. Madey, "An operation-time simulation framework for uav swarm configuration and mission planning," *Procedia Computer Science*, vol. 18, pp. 1949–1958, 2013.

[5] H. Wu, H. Li, R. Xiao, and J. Liu, "Modeling and simulation of dynamic ant colony's labor division for task allocation of uav swarm," *Physica A: Statistical Mechanics and its Applications*, vol. 491, pp. 127–141, 2018.

[6] G. D. Goh, S. Agarwala, G. Goh, V. Dikshit, S. L. Sing, and W. Y. Yeong, "Additive manufacturing in unmanned aerial vehicles (uavs): Challenges and potential," *Aerospace Science and Technology*, vol. 63, pp. 140–151, 2017.

[7] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 2520–2525.

[8] D. Mellinger, A. Kushleyev, and V. Kumar, "Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 477–483.

[9] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics Research: The 16th International Symposium ISRR*. Springer, 2016, pp. 649–666.

[10] J. Ochodnickỳ, Z. Matousek, M. Babjak, and J. Kurty, "Drone detection by ku-band battlefield radar," in *2017 International Conference on Military Technologies (ICMT)*. IEEE, 2017, pp. 613–616.

[11] Y. Kim and H. Ling, "Human activity classification based on micro-doppler signatures using a support vector machine," *IEEE transactions on geoscience and remote sensing*, vol. 47, no. 5, pp. 1328–1337, 2009.

[12] J. Park, R. J. Javier, T. Moon, and Y. Kim, "Micro-doppler based classification of human aquatic activities via transfer learning of convolutional neural networks," *Sensors*, vol. 16, no. 12, p. 1990, 2016.

[13] X. Huang, J. Ding, D. Liang, and L. Wen, "Multi-person recognition using separated micro-doppler signatures," *IEEE Sensors Journal*, vol. 20, no. 12, pp. 6605–6611, 2020.

[14] B. Jokanović and M. Amin, "Fall detection using deep learning in range-doppler radars," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 1, pp. 180–189, 2017.

[15] M. S. Seyfioglu, B. Erol, S. Z. Gurbuz, and M. G. Amin, "Dnn transfer learning from diversified micro-doppler for motion classification," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 5, pp. 2164–2180, 2018.

[16] S. Wang, J. Song, J. Lien, I. Poupyrev, and O. Hilliges, "Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 2016, pp. 851–860.

[17] M. Li, T. Chen, and H. Du, "Human behavior recognition using range-velocity-time points," *IEEE Access*, vol. 8, pp. 37 914–37 925, 2020.

[18] W. J. Anderson, *Continuous-time Markov chains: An applications-oriented approach*. Springer Science & Business Media, 2012.

[19] Z. Zou, Y. Tian, Y. Lin, J. Yang, and W. Yi, "A two-stage deep network for the quantity estimation and detection of multiple swarm-type targets using radar rd images," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2023, pp. 3557–3562.

[20] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5353–5360.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[24] Z. Zhang, H. Lai, D. Huang, X. Fang, M. Zhou, and Y. Zhang, "Reta: 4d radar-based end-to-end joint tracking and activity estimation for low-observable pedestrian safety in cluttered traffic scenarios," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[25] R. Cui, H. Liu, and C. Zhang, "A deep neural framework for continuous sign language recognition by iterative training," *IEEE Transactions on Multimedia*, vol. 21, no. 7, pp. 1880–1891, 2019.