scientific reports

Check for updates

OPEN

Reynolds rules in swarm fly behavior based on KAN transformer tracking method

Qi Yang[1], Jiajun Ji[1], Ruomiao Jing[1], Haifeng Su[1]✉, Shuohong Wang[2] & Aike Guo[1]

The analysis of complex flight patterns and collective behaviors in swarming insects has emerged as a significant focus across biological and computational fields. Tracking these insects, like fruit fly, presents persistent challenges due to their rapid motion patterns and frequent occlusions in densely populated environments. To address these challenges, we propose a tracking method using particle filter framework combined with a Kolmogorov–Arnold Network (KAN)-Transformer model to extract the global features and fine-grained features of the trajectory. Additionally, manually annotated ground truth datasets are established to enable thorough assessment of tracking methods. Experimental results demonstrate the effectiveness and robustness of our proposed tracking method. Analysis of tracked trajectories revealed the Reynolds rules of flocking behavior.

Understanding collective animal behavior is crucial for uncovering the mechanisms of group coordination, decision-making and information transfer. Such studies not only deepen our knowledge of biological systems but also inspire applications in multi-agent systems like swarm robotics. Previous studies have explored collective behavior across various species, revealing how groups form[1–3], how information spreads within groups[4–7] and the generation of behavioral patterns[8,9]. For instance, in bird flocks, interactions between individuals are shown to depend on topological distance (a fixed number of nearest neighbors) rather than metric distance (absolute spatial proximity), enabling groups to maintain cohesion and coordination across varying densities[10]. Hierarchical structures have also been observed in pigeon flocks, where a small number of leader individuals exert a disproportionate influence on the group's movement decisions, while others adjust their motion based on local neighbors' behavior[11]. Similarly, mosquito mating swarms exhibit short-term synchronized flight patterns driven by local velocity alignment, highlighting the role of pairwise interactions in generating coordinated group behavior[12]. In fish schools, visual sensory networks have been identified as the primary mechanism for information transfer, outperforming traditional metric- and topology-based interaction models in predicting how behavioral responses propagate during leadership events[13].

To analyze these collective behaviors quantitatively, accurate tracking of individual animals within groups is essential[14]. While some studies utilize GPS devices for larger animals[11], computer vision-based multi-object tracking at different frames has emerged as a powerful tool for studying smaller targets in laboratory settings. Wojke introduced DeepSORT[15], which integrates deep appearance features with Kalman filtering and cascade matching using Mahalanobis and cosine distances, establishing a robust real-time tracking framework. Zhang proposed FairMOT[16], a single-network architecture that jointly handles detection and re-identification feature extraction, demonstrating that a well-designed single network can outperform separate networks for detection and Re-ID. Zhang developed ByteTrack[17], which innovatively utilizes both high-score and low-score detections in different association processes, achieving state-of-the-art performance on MOT benchmarks. Recent advancements in transformer-based approaches have opened new possibilities for multi-object tracking. Yuan proposed ETDMOT[18], a novel end-to-end transformer-based framework specifically designed for drone-based tracking scenarios. Their approach integrates object detection and tracking into a unified pipeline, leveraging self-attention mechanisms to capture complex inter-object relationships. Various methods have been developed for tracking groups of organisms in biological studies. The GReTA[19] algorithm offers robust 3D multi-object tracking through global optimization and recursive divide-and-conquer, yet it faces challenges in computational complexity, sensitivity to initial links and handling non-confined data. Dell reviewed automated image-based tracking techniques[20], highlighting methods like background subtraction and fingerprinting for tracking small organisms in complex environments. TRex[21] used deep-learning-based visual identification to track large groups of individuals in behavioral studies. Some of these appearance-based tracking methods, like Dell's[20] and TRex[21], heavily rely on visual features such as shape and appearance of the targets. However, the small size and similar

[1]School of Life Sciences, Shanghai University, Shanghai 200444, China. [2]Department of Molecular and Cellular Biology and Center for Brain Science, Harvard University, Cambridge, MA 02138, USA. ✉email: hfsu@shu.edu.cn

appearance of *Drosophila* often appear as dark spots in images with minimal texture or color variations. Hence, it is challenging to extract discriminative visual features to identify the same *Drosophila* across consecutive frames.

Various approaches have been developed to track small and featureless target like *Drosophila*. Manoukis[22] used a constant-velocity Markov process with random perturbations for 3D tracking in high-density environments. Angarita-Jaimes[23] adopted the idea of maximizing temporal smoothness for determining inter-frame associations. Nevertheless, these conventional tracking methods exhibit significant limitations when confronted with dense environments with rapidly moving targets. CNN-based detection methods demonstrated enhanced capabilities but remained susceptible to challenges such as partial occlusions and dense clustering of *Drosophila*[24]. Wu[25] introduced a novel approach combining particle filtering with 2D tracking in multiple views, utilizing linear assignment for track matching. Despite its innovations, this method suffered from trajectory fragmentation when targets failed to associate with detections in subsequent frames. Although Wu's later work[26] attempted to address this limitation by relaxing the one-to-one matching constraint, the system's efficacy remained heavily dependent on detection accuracy. An alternative strategy involves establishing cross-view associations through feature matching to reconstruct 3D observations, followed by cross-frame association. Ardekani[27] successfully implemented this approach for multiple *Drosophila* tracking. However, the method faces significant challenges in accurately distinguishing numerous small objects with similar appearances in 2D space and reconstructing their 3D positions. Wang[28] enhanced the tracking accuracy by integrating long short-term memory (LSTM)[29] networks with particle filtering to model *Drosophila* kinematics. Yin[30] integrated improved object detection with a Transformer[31] model, arguing that the Transformer architecture is naturally suited for Lévy flight trajectory prediction. Nevertheless, these approaches exhibited limitations in maintaining continuous trajectories during occlusions and struggled with long-term tracking performance.

Recent research[32–35] has shown promising results using Kolmogorov-Arnold Networks (KANs)[36] for time series forecasting, demonstrating advantages in theoretical foundations, interpretability and prediction accuracy while maintaining a clear mathematical relationship between network architecture and function approximation capabilities. Xu[32] and Vaca-Rubio[33] explored variants of KANs integrated with traditional forecasting models, as well as new symbolic regression techniques within KANs, for dynamic univariate and multivariate forecasting. These innovations highlight KANs' ability to offer enhanced analytical capabilities and improved interpretability in time series tasks. Han[34] developed the Reversible Mixture of KAN experts (RMoK), utilizing a mixture-of-experts approach to adaptively assign time series variables to specialized KANs, thereby improving model performance through detailed feature weight analysis. Genet[35] introduced the Temporal Kolmogorov-Arnold Transformer (TKAT), which combines KANs with transformer architectures to handle complex multivariate data and long-range dependencies. However, this approach does not fully leverage the strengths of both models, as it uses KAN to encode the time series before applying multi-head attention.

Considering that the Transformer excels at capturing long-range dependencies in time series data, while KAN's superior function approximation capabilities enhance the model's ability to process local temporal patterns, we propose KAN-Transformer, a novel time sequence forecasting model. Integrated with a probabilistic tracking framework with particle filtering, our model addresses both global temporal relationships and fine-grained details in tracking multiple objects under challenging crowded scenarios. Using a manually labeled ground truth dataset, our method outperforms some existing approaches. We further discover collision avoidance and companion flying phenomena in swarm flying fruit flies.

Method

The proposed tracking method implements detection and tracking stages in a sequential manner until the entire video sequence has been processed, as illustrated in Fig. 1. During the detection stage, the image blob of each target is detected and extracted from the background-subtracted frames. The tracking stage incorporates a particle filter framework, where a KAN-Transformer network is employed to learn and implement the kinematic pattern of individual targets as the dynamic model. The target's position state is estimated through weighted particles. This method enables accurate prediction of target position even in scenarios with abrupt motion.

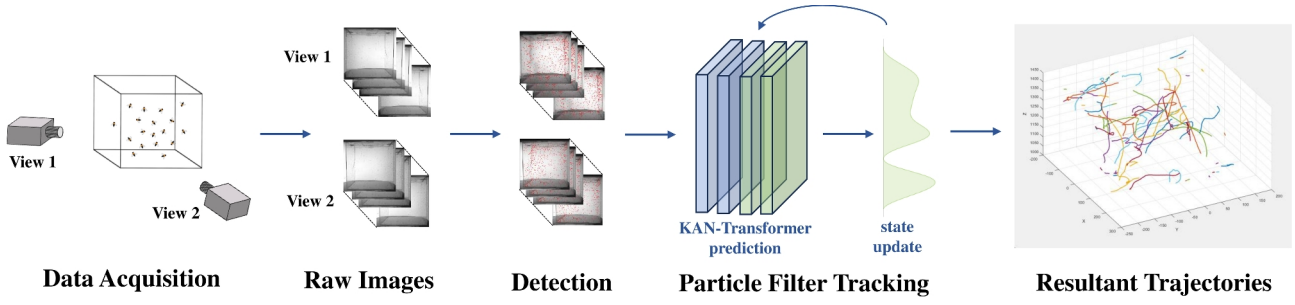


Fig. 1. The workflow of the proposed 3D tracking method, consisting of detection and tracking stages. The detection stage identifies and extracts target image blobs from background-subtracted frames, while the tracking stage utilizes a particle filter framework with a KAN-Transformer network as the dynamic model.

Fly stocks

The wild type *Drosophila melanogaster* strain used was the Canton-Special (CS) strain. Fly stocks were raised on standard food at 25 °C and 50% relative humidity under a 12:12 h light:dark cycle. The experimental fruit flies are approximately 5–10 days old.

Experimental setup and camera configuration

The experimental setup consisted of two high-speed monochrome CMOS cameras (Manufacturer: IO Industries Inc. Country: Canada, Device name: Flare 4M 180-CL), each equipped with a Computar M1614-MP2 lens with a focal length of 16 mm. The cameras were positioned orthogonally at a distance of 90 cm from a $40 \times 40 \times 40$ cm cubic transparent acrylic container. Approximately 400–500 flies were released for the experiment. The cameras' geometric relationship was established through a standard chessboard pattern calibration process, ensuring precise spatial alignment necessary for 3D trajectory reconstruction. For calibration and triangulation of 3D positions, we utilized the Camera Calibration Toolbox for Matlab developed by Caltech[37], which provided robust camera alignment for accurate 3D reconstruction.

Operating at 100 frames per second with a 2048×2040 pixel resolution, the cameras were synchronized via hardware triggers to capture simultaneous frames of rapid *Drosophila* movements. To ensure uniform and naturalistic illumination, an infrared lightbox was employed. The lightbox consisted of two perpendicular planes, each equipped with a series of infrared LED lights diffused through white plastic panels to create soft and even lighting. The background was brighter than the objects, facilitating clear differentiation of the flies. This configuration enabled high-fidelity tracking of the fast-moving insects while maintaining temporal consistency between camera views. The experiment captured a total of 5790 frames.

The body model

Following the previous work[28], we model the body of *Drosophila* by approximating an ellipsoid. The state vector of each target consists of the coordinate of the ellipsoid center (x, y, z):

$$S_t = \{x, y, z\} \tag{1}$$

Detection

Although the images captured from each view are cluttered with swarms of targets, the background remains relatively stable over short time intervals. Target detection can be efficiently and accurately performed through background removal[38,39]. The background-removed image of each view v at time t is given by Eq. (2):

$$I_\nu'^t = \left| I_\nu^t - \text{mean}\left(I_\nu^{t-w/2}, \ldots, I_\nu^{t+w/2}\right) \right| \tag{2}$$

where w is selected based on the average velocity of the targets and the frame rate. The region of each target is then segmented based on the the thresholded binary image of I' and further processed using ellipse fitting. In the target overlap issue, when two targets overlap in view 1, they are detected as a single ellipse. We can use the epipolar line to project the center of this ellipse into view 2, thereby achieving the separation of overlapping targets. Finally, we use stereo matching to reconstruct them into ellipsoids.

Trajectory prediction

We introduce our proposed model, KAN-Transformer, which replaces the linear layers in the traditional Transformer architecture with Kolmogorov-Arnold Network (KAN) to predict future trajectories based on historical trajectories. The overall architecture of our model is shown in Fig. 2.

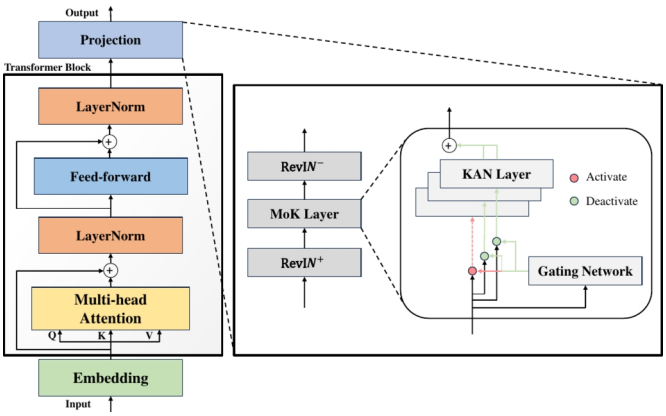


Fig. 2. The overall architecture of the KAN-Transformer. The left panel shows the Transformer model, where the linear layer is replaced by the Reversible Mixture of KAN Experts (RMoK) model, shown in the right panel.

Problem definition
Given a time series $\{X_{t-h}, \ldots, X_{t-1}\}$, where X_t represents object state at time *t*, our objective is to predict their future values $\{X_t, \ldots, X_{t+T}\}$ over the next *T* time steps based solely on *h* historical time steps.

Transformer
The architectural framework of the Transformer model is shown in the left panel of Fig. 2. First, the embedding layer embeds the time series $\{X_{t-h}, \ldots, X_{t-1}\}$ into a sequence of vectors $\mathbf{E^0} = \{E_{t-h}, \ldots, E_{t-1}\}$. Then E_0 would be processed by each of the transformer layers successively. We denote the output of the *j*-th layer as $\mathbf{E^j}, (0 < j \leq N)$.

The key component of the transformer layer is self-attention, which allows the model to weigh the importance of each time step of the input sequence. In the *j*-th transformer layer, the input E_{j-1} is projected into three vectors: Query (Q), Key (K) and Value (V), through three learnable matrices, respectively. Then we compute the output of self-attention as:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

where d_k is the scaling factor, typically the dimension of the Key vector. The softmax function written as Eq. (4), ensures that the attention weights sum up to 1, allowing the model to distribute its focus among the input sequence.

$$softmax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}} \tag{4}$$

In the traditional Transformer architecture, the output embeddings $\mathbf{E^N}$ from the final transformer layer are concatenated and projected into the predicted trajectory $\{X_t, \ldots, X_{t+T}\}$ through a fully connected layer. In this work, we propose to replace the conventional linear projection layer with Reversible Mixture of KAN Experts (RMoK) to enhance the model's capability.

RMoK
In contrast to Multilayer Perceptrons (MLPs), which are founded on the universal approximation theorem, Kolmogorov-Arnold Networks (KANs) are based on the Kolmogorov-Arnold representation theorem, also referred to as the Kolmogorov-Arnold superposition theorem. This theorem represents a seminal contribution to the theory of dynamical systems and ergodic theory, independently formulated by Andrey Kolmogorov and Vladimir Arnold during the mid-twentieth century.

The theorem establishes that any multivariate continuous function *f*, dependent on the vector $\mathrm{x} = [x_1, x_2, \ldots, x_n]$ within a bounded domain, can be decomposed into a finite composition of simpler continuous univariate functions. More precisely, a real-valued, smooth and continuous multivariate function $f(\mathrm{x}) : [0, 1]^n \to \mathbb{R}$ can be expressed as a finite superposition of single-variable functions[40]:

$$f(\mathrm{x}) = \sum_{i=1}^{2n+1} \Phi_i \left(\sum_{j=1}^{n} \phi_{i,j}(x_j) \right) \tag{5}$$

where $\Phi_i : \mathbb{R} \to \mathbb{R}$ and $\phi_{i,j} : [0, 1] \to \mathbb{R}$ denote the so-called outer and inner functions, respectively.

As noted by Liu[36], Eq. (5) exhibits a two-layer nonlinear structure comprising $2n + 1$ terms in the intermediate layer. The key challenge lies in identifying the appropriate univariate functions-both the inner functions $\phi_{i,j}$ and outer functions Φ_i-to achieve accurate function approximation. The one-dimensional inner functions $\phi_{i,j}$ can be effectively approximated using B-splines, which are piecewise polynomial functions defined by a set of control points or knots.

B-splines are particularly valuable for their ability to provide smooth and continuous interpolation or approximation of data points. These functions are characterized by two primary parameters: the order *k* (with $k = 3$ being a commonly adopted value), which determines the degree of the polynomial functions used for interpolation between control points, and the number of intervals *G*, which specifies the number of segments between adjacent control points. In the context of spline interpolation, these segments connect successive data points to generate a smooth curve consisting of $G + 1$ grid points.

A KAN layer is defined by a matrix Φ[36] composed by univariate functions $\{\phi_{i,j}(\cdot)\}$ with $i = 1, ..., N_{in}$ and $j = 1, ..., N_{out}$, where N_{in} and N_{out} denote the number of inputs and the number of outputs, respectively, and $\phi_{i,j}$ are the trainable spline functions described above. Therefore, the computation process of a KAN layer can be expressed as:

$$x_j = \sum_{i=1}^{N_{in}} \phi_{i,j}(x_i) \tag{6}$$

where x_i represents the *i*-th input and x_j denotes the *j*-th output of the KAN layer.

Given the substantial distributional heterogeneity observed in real-world time series data and the specialized nature of spline functions in modeling specific data distributions, we adopt the Reversible Mixture of KAN Experts Model (RMoK) as proposed by Han[34]. The architectural framework of the RMoK model is illustrated in the right panel of Fig. 2.

The Mixture of KAN (MoK) layer incorporates a gating network that allocates KAN layers to variables based on temporal characteristics, with each expert specializing in a specific subset of the data. Given that KAN and its variants differ solely in their spline function implementations, we adopt the notation $\mathscr{K}(\cdot)$ to uniformly represent these methodologies throughout this paper. The proposed MoK layer, comprising *N* experts, can be succinctly formulated as:

$$\mathrm{x}_{l+1} = \sum_{i=1}^{N} \mathscr{G}(\mathrm{x}_l)_i \mathscr{K}_i(\mathrm{x}_l) \tag{7}$$

where $\mathscr{G}(\cdot)$ denotes the gating network. This mixture of experts architecture effectively addresses the heterogeneous nature of time series data, enabling each expert to capture distinct temporal features, thereby enhancing the model's performance in time series forecasting tasks. The model employs the sparse gating network architecture introduced by Shazeer[41], which selectively activates only the top-*k* best-matching experts. This network augments the input time series with Gaussian noise through $\mathrm{w}_{\mathrm{noise}}$ during training and employs a $\mathrm{KeepTopK}$ operation to preserve the *k* experts with the highest activation values. The noise helps diversify the training data and promotes balanced utilization of experts in the Mixture of KAN framework[41]. The sparse gating network can be formally expressed as:

$$\mathscr{G}_{sparse}(\mathrm{x}) = softmax(KeepTopK(H(\mathrm{x}), k)) \tag{8}$$

$$H(\mathrm{x}) = \mathrm{x}\mathrm{w}_g + Norm(softplus(\mathrm{x}\mathrm{w}_{noise})) \tag{9}$$

where $Norm(\cdot)$ represents standardization.

Inspired by several successful single-layer methods[42,43], RMoK integrates Reversible Instance Normalization (RevIN)[44] with a single MoK layer. RevIN is a normalization-and-denormalization method designed to address distribution shifts in time-series data, where statistical properties like mean and variance change over time. First, RevIN^+ (the normalization operation of RevIN) applies a learnable affine transformation to normalize the input time series for each variable. Then, the MoK layer processes these normalized temporal features to generate predictions. Finally, RevIN^- (the denormalization component of RevIN) transforms the predictions back to the original distribution space using the identical affine transformation parameters from the first step. This approach significantly improves forecasting accuracy.

Tracking via particle filter framework

As the dynamic system of flying swarms exhibits highly nonlinear characteristics and the posterior density is typically non-Gaussian, the particle filtering framework is employed to approximate the target's state posterior using a set of N weighted particles, denoted as $\{(\mathrm{S}_t^i, w_t^i)\}_{i=1,\ldots,N}$, where S_t^i denotes the states of *i*-th particle and w_t^i denotes it's weight. Each particle, sampled from the previous state through importance sampling and propagated by the dynamic model, is weighted according to the observation likelihood at time t given the particle state, denoted as $w_t^i \propto p(\mathrm{Z}_t|\tilde{\mathrm{S}}_t^i)$, which constitutes the observation model. The expected target state $\hat{\mathrm{S}}_t$ is then computed as:

$$\hat{\mathrm{S}}_t = E(\mathrm{S}_t|\mathrm{Z}_{1:t}) = \sum_{i=1}^{N} w_t^i \mathrm{S}_t^i \tag{10}$$

The dynamic model in a tracking method predicts the state of the target at each time step. In our proposed method, a KAN-Transformer mentioned in Method is applied to learn the kinematic pattern of the flying object and the learned kinematic pattern is used as the dynamic model. The output of KAN-Transformer X_t is the hypothetical translation. Thus the predicted state of each target is $\tilde{\mathrm{S}}_t = \mathrm{S}_{t-1} + \mathrm{X}_t$.

Evaluation metric

The tracking performance was evaluated using the widely adopted CLEAR MOT metrics[45]: Multiple Object Tracking Precision (MOTP) and Multiple Object Tracking Accuracy (MOTA). MOTP measures the algorithm's ability to precisely estimate target states. It is calculated as:

$$\mathrm{MOTP} = \frac{\sum_{i,t} d_t^i}{\sum_t c_t} \tag{11}$$

where d_t^i represents the Euclidean distance between the center position of each correctly tracked target and its corresponding ground truth, and c_t denotes the number of correctly tracked targets in each frame. Note that for MOTP, lower values indicate better performance, as it reflects higher precision in target localization. While MOTP focuses on localization precision, MOTA assesses the tracker's performance in terms of missed targets, false positives and identity switches:

$$\mathrm{MOTA} = 1 - \overline{m} - \overline{fp} - \overline{mme} \tag{12}$$

For MOTA, higher values indicate better performance, as it reflects fewer tracking errors (misses, false positives and identity switches). The components of MOTA are defined as follows:

Ratio of misses, which indicates the proportion of ground truth objects that were not tracked:

$$\overline{m} = \frac{\sum_i m_i}{\sum_i g_i} \tag{13}$$

where m_i is the number of missed targets and g_i is the number of ground truth objects.

Ratio of false positives, which represents the proportion of false detections relative to the ground truth:

$$\overline{fp} = \frac{\sum_i fp_i}{\sum_i g_i} \tag{14}$$

where fp_i represents the number of false positive detections.

Ratio of mismatches, which measures the frequency of identity switches in tracking:

$$\overline{mme} = \frac{\sum_i mme_i}{\sum_i g_i} \tag{15}$$

where mme_i denotes the number of identity switches.

Specifically, for a tracked trajectory $S = \{S_t, \ldots, S_{t+T}\}$, we compute the Euclidean distance between the first position S_t and the closest ground truth target G_t^i at frame *t*. If the center of S_t lies within the bounding box of G_t^i, the trajectory *S* is considered to correspond to the ground truth trajectory $\{G_t^i, \ldots, G_{t+T}^i\}$. Otherwise, *S* is classified as a false positive. For frames $j = t+1, \ldots, t+T$, if the position S_j lies outside the bounding box of G_j^i, it is considered a mismatch. Additionally, any ground truth trajectory points that are not matched to a tracked point are classified as misses.

Given that trajectory completeness is crucial for subsequent behavioral interaction analysis, we quantitatively evaluated the distribution of tracked trajectory lengths.

Training details

The KAN-Transformer model was trained to predict the translation of a target based on its historical trajectory. The model was optimized using AdamW with an initial learning rate of $1e-4$, beta values of (0.95, 0.9), and weight decay of $1e-5$. A StepLR scheduler was employed with a step size of 1 epoch and a decay rate of 0.5. We constructed the training dataset using 400 video clips of flying fruit fly, where each clip contains only one individual fly. Consecutive point pairs were extracted from each video clip to capture the flight trajectory of the fly. The dataset is partitioned into training, validation, and test sets with a ratio of 8:1:1. The model was trained for 10 epochs, and the best-performing model weights were selected based on the highest performance on the validation set. All experiments were conducted on a workstation equipped with an NVIDIA RTX3090 GPU.

Dynamic time warping

The Dynamic Time Warping (DTW) algorithm constructs a cost matrix to store the cumulative distances between elements of two sequences. By leveraging a dynamic programming approach, DTW minimizes the cumulative distance and efficiently calculates the optimal alignment path by reusing previously computed

results. In our analysis of fruit fly behavior, we employ DTW to calculate the distance between the angular accelerations of two tracked trajectories.

Statistical analysis

All data from this experiment were analyzed using MATLAB. The normality of the data distribution was assessed using the Jarque–Bera test. Non-normally distributed data were subjected to the non-parametric Mann-Whitney test to evaluate differences between groups. Data are presented as the mean ± standard deviation (std). Statistical significance was set at $^{****}p < 0.0001$.

Results

Dataset collection and ground truth annotation

To facilitate the training of the proposed KAN-Transformer model, an extensive dataset of ground truth sequences was established. The training dataset comprises 400 video clips, each containing a single flying target within the arena, thereby eliminating potential ambiguities in cross-view and cross-frame associations. In total, the dataset includes 39,647 frames of trajectory data.

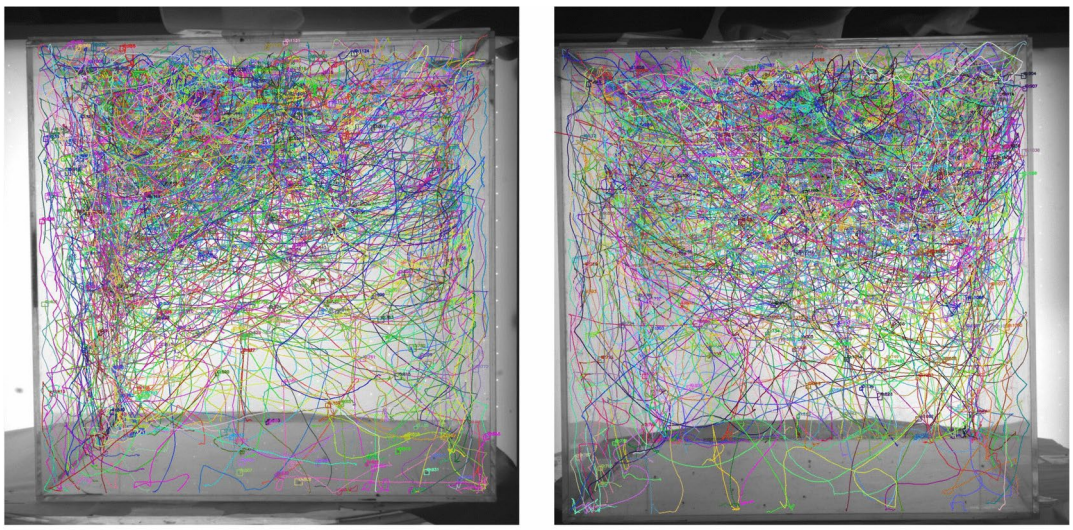
To address the lack of tracking accuracy evaluation on real-world fly swarm recordings in previous studies, ground truth data was obtained through manual annotation of real-world video sequences. The annotated dataset consists of approximately 400–500 insects observed over 1400 frames captured from two distinct perspectives. The annotation process was conducted by a team of approximately 40 trained annotators. The visualized trajectories are shown in Fig. 3. To supplement the validation of our evaluation results, we annotated two additional video sequences (50 frames each) that were acquired using the same experimental setup (These datasets were not visualized here). Unlike previous studies[28,30,46,47] that only recorded and tracked fruit flies, this work presents the first large-scale manual annotation of collective *Drosophila* flight trajectories.

Superior performance over state-of-the-art methods

We compare our tracking method with two state-of-the-art methods, an LSTM-based method[28] and a Transformer-based framework[30], in MOTP, MOTA and trajectory length under identical experimental conditions, with the trajectory prediction model being the only variable component. Additionally, we also compare with two conventional methods in terms of MOTP and MOTA , referred to as Markov[22] and SeqFileProcessing2D[23]. All methods are tested on our annotated ground truth dataset, which was not used for training.

The tracked trajectories were projected onto each camera view's 2D image plane, and MOTP was calculated against manual annotations using Eq. (11). The comparison of MOTP performance is presented in Fig. 4a, where our proposed method demonstrates an improvement of over 9.4% improvement compared to the existing approaches, except for Markov. Interestingly, our experimental results reveal that the LSTM-based method outperforms the Transformer-based approach, which appears to contradict the findings reported in Yin[30]. We attribute this discrepancy to the fact that Yin[30] implemented multiple modifications simultaneously, including both the Transformer architecture and alterations to the object detection method, without conducting ablation studies to isolate the effectiveness of the Transformer component specifically.

The comparison of MOTA metrics presented in Fig. 4b, showing that our method slightly outperforms the state-of-the-art methods and significantly exceeds the performance of traditional approaches. It's worth noting that the constant-velocity Markov process with random perturbations is not well-suited for environments with high crowd density and fast movement, leading to a significantly higher trajectory mismatch rate. While



(a) **(b)**

Fig. 3. Visualization of annotated trajectories of *Drosophila*. The dataset comprises 700 frames with 400–500 flies. (**a**) Trajectories from view 1. (**b**) Trajectories from view 2.

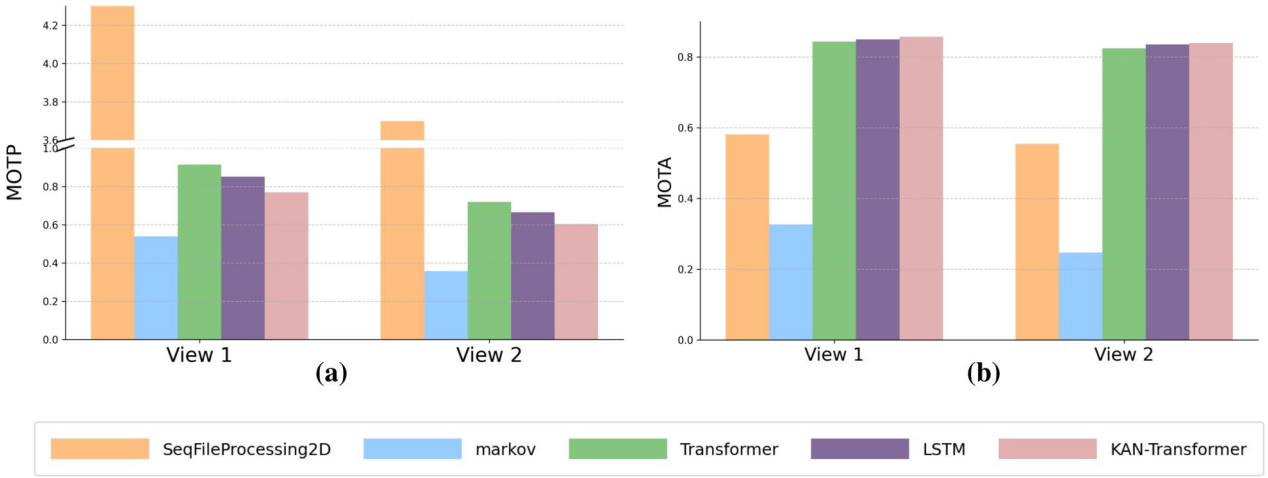


Fig. 4. Tracking Performance comparison between KAN-Transformer and other methods in two orthogonal views of the dataset. The dataset comprises 700 frames with 400–500 flies. (**a**) MOTP: lower values indicate better performance; (**b**) MOTA: higher values indicate better performance.

the Markov method offers more accurate tracking positions than the particle filter-based algorithm, it fails to correctly track the intended trajectories. SeqFileProcessing2D employs the principle of maximizing temporal smoothness, but it is also unsuitable for fast-moving scenarios, leading to a higher miss rate and discontinuous trajectories. We also evaluated the performance of each method on two additional manually annotated datasets, which were captured under experimental conditions consistent with the previous one. The only difference is that these two datasets are annotated for 50 frames. The evaluation results are presented in Supplementary Figures 1, similarly demonstrating our model's robust tracking performance.

Due to the significantly high miss rate and mismatch rate of SeqFileProcessing2D and the Markov method, we have opted not to include visualizations comparing their trajectory. The histogram in Fig. 5a illustrates the distribution of trajectory lengths between different methods. The Trajectory Length Ratio refers to the ratio of the tracking trajectory length to the total recording sequence length. The average trajectory length ratio of LSTM, Transformer and KAN-Transformer are 1.31%, 1.18% and 1.45%, respectively. Some randomly selected cases are illustrated in Fig. 5b. The proposed method achieved superior performance in maintaining extended trajectories, benefiting from the improved accuracy of its trajectory estimation model.

We observed that KAN-Transformer consistently generates longer trajectories, while LSTM and conventional Transformer models tend to lose track of targets at turning points, resulting in trajectory interruptions. These fragmented short trajectories impede subsequent analysis of fruit fly behavioral interactions. We hypothesize that the lower prediction accuracy of LSTM and Transformer models for trajectory forecasting leads to reduced probability of successful frame-to-frame matching, as incorrect predictions of positions in subsequent frames compromise the tracking continuity. To validate this hypothesis, we conducted experiments comparing the trajectory prediction performance of KAN-Transformer, LSTM and Transformer models on the single-target dataset described above. Additionally, we compared the performance of three KAN-based time series forecasting models: KAN[32], RMoK[34] and TKAT[35]. The two conventional methods[22,23] are excluded from the comparison since they are not time-series forecasting algorithms. All these models were evaluated solely as temporal prediction models without incorporating particle filtering. The performance of these models was evaluated using two metrics: Mean Squared Error (MSE) and Mean Absolute Error (MAE), computed on the test set, which was kept entirely separate from the training and validation data. For both MSE and MAE, lower values indicate better performance, as they reflect smaller errors between the predicted and ground truth values.

$$\begin{aligned} \mathrm{MAE}(x, \hat{x}) &= \frac{1}{N}\sum_{i=0}^{N} |x_i - \hat{x}_i| \\ \mathrm{MSE}(x, \hat{x}) &= \frac{1}{N}\sum_{i=0}^{N} (x_i - \hat{x}_i)^2 \end{aligned} \tag{16}$$

The experimental results are presented in Table 1. The KAN-Transformer model demonstrates superior performance in time series prediction compared to baseline models, confirming our initial hypothesis. Additionally, we can observe that the RMoK model outperforms the basic KAN, and RMoK's performance is close to that of the Transformer. Our method combines the strengths of both RMoK and the Transformer, yielding superior results. The performance of TKAT shows only a slight improvement over KAN and is much lower than that of our KAN-Transformer. We believe this is primarily due to TKAT using KAN to extract features from the raw time series, which cannot effectively capture temporal dependencies. In contrast, our model first uses

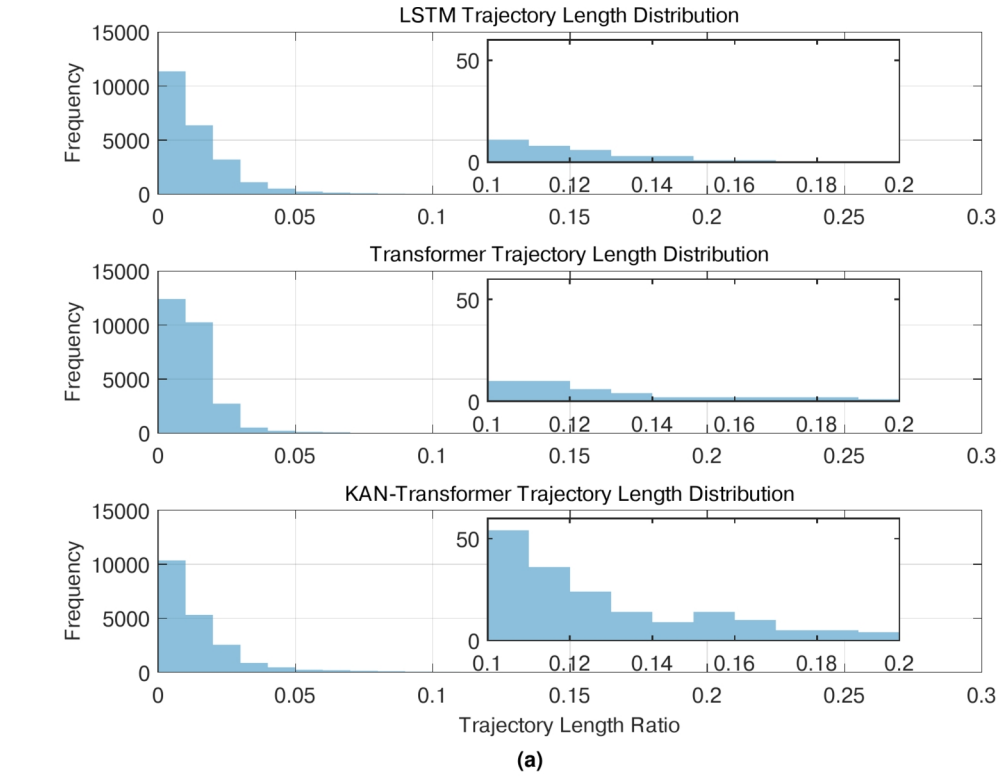


Fig. 5. The KAN-Transformer consistently generates longer and more continuous trajectories. (**a**) Length Distribution of Tracked Trajectories for KAN-Transformer and State-of-the-art Methods on a dataset comprising 700 frames with 400–500 flies. (**b**) Some randomly selected trajectories cases of KAN-Transformer and State-of-the-art Methods. The average length of these visualized trajectories is 11, 10 and 26 frames for LSTM, Transformer and our method, respectively.

Metrics	KAN[32]	RMoK[34]	TKAT[35]	LSTM[29]	Transformer[31]	KAN-transformer (ours)
MAE (↓)	1.3598	1.2536	1.2721	1.4248	1.2317	1.0333
MSE (↓)	3.7833	2.8441	3.3508	4.0881	2.6599	1.9358

Table 1. Evaluation of KAN-Transformer model with other time series forecasting models.

the Transformer to capture temporal dependencies and then decodes with KAN, fully leveraging the temporal dependency extraction capability of the Transformer and the superior function-fitting ability of KAN.

To evaluate the effectiveness of KAN and Transformer modules in our proposed method, we conducted an ablation study comparing KAN-only, Transformer-only and the complete KAN-Transformer architecture.

Table 2 presents the tracking performance in terms of MOTP, MOTA and the ratio of the average tracking length to the total recording sequence length for these variants.

The results of the ablation study demonstrate the clear advantages of integrating KAN and Transformer modules in our architecture. The complete KAN-Transformer method achieves notable improvements in both tracking accuracy and trajectory maintenance compared to its single-module variants. Specifically, the proposed method shows reduced MOTP error and maintains a higher tracking length ratio compared to the Transformer-only approach. These improvements suggest that the KAN module effectively complements the Transformer architecture, where KAN enhances the local feature extraction while the Transformer captures long-range dependencies, resulting in more robust tracking performance.

Reynolds rules in swarm fly behavior

Previous studies on collective behavior in *Drosophila* have primarily focused on crawl experiments[48–50], while their aerial collective properties remain largely unexplored. Flying animals are known to exhibit collective behaviors through local interactions[10,11]. However, the collective aerial properties of swarm *Drosophila* lack systematic investigation. During flight, fruit flies perform a series of straight flight transitions, interspersed with extremely rapid saccades. While turning, they decelerate, which is manifested as the decomposition of velocity into angular velocity and tangential linear velocity changes. Therefore, we plotted the angular acceleration patterns of all tracked fruit flies in Fig. 6a.

We use Dynamic Time Warping (DTW) analysis[51] to calculate the distance between the angular accelerations of two tracked trajectories. DTW is an algorithm that utilizes dynamic programming to measure the similarity between two time series or sequences, allowing for non-linear alignments. The DTW-based distance matrix of trajectory pairs is visualized in Fig. 6b, indicating significant temporal correlations in angular acceleration among *Drosophila*.

Flock-forming animals[10–13] follow the three principles of Reynolds rules[52] , steering to avoid collisions with nearby neighbors (Separation), steering towards the average heading of nearby neighbors (Alignment) and steering towards the average position of neighbors (Cohesion). Individuals must maintain sufficient proximity to preserve group integrity while avoiding collisions with neighbors, resulting in coordinated movement. To further investigate whether fruit fly groups possess certain social properties, we employed the Silhouette Score method to determine the optimal number of clusters followed by K-means clustering, using the DTW distance as the clustering metric. The higher Silhouette Score suggests stronger cluster cohesion within groups and better separation between groups. Based on this quantitative evidence, we applied K-means clustering with $k = 2$ to group the trajectory pairs (Fig. 6c). For visualization purposes, the high-dimensional data was projected onto a two-dimensional space using Multidimensional Scaling (Fig. 6d). As shown in Fig. 7, our clustering results revealed two behavioral patterns during rapid flight: collision avoidance and accompanying flight, which align with Reynolds rules[52] of flocking behavior. When in close proximity, fruit flies take evasive maneuvers to avoid collisions. The acceleration changes of two fruit flies reflect a consistency in sequence, with one fruit fly initiating a turn or change in speed, followed by the other, possibly with a very similar timescale (Fig. 7a). Although the accompanying flight phenomenon shows similar changes in acceleration as the collision-avoidance cases, we find that the two fruit flies maintain a constant distance within a certain time window (Fig. 7b and Supplementary Figure 2). Actually, these two behavioral patterns could occur simultaneously, which is shown in Fig. 7c, when the distance between two fruit flies becomes too close during accompanying flight, collision avoidance behavior emerges. By calculating the Euclidean distance between fruit flies (Fig. 6d), we observe that the distance values in Cluster 1, which represents accompanying flight cases, are more concentrated, as shown in Fig. 6e. This reflects the cohesion property of fruit fly groups. In contrast, Cluster 2, which represents collision-avoidance cases, shows more dispersed distances between fruit flies. This suggests that fruit flies move from far to near and then from near to far again during the collision-avoidance process (Fig. 7a). In Cluster 1, we further demonstrate the alternate accompanying flight phenomenon of multiple fruit flies (Fig. 7d, Supplementary Figure 3). The duration of multiple fruit flies' continuous accompanying flight varies from 20 to 50 milliseconds. Considering the flight speed of fruit flies, this should maintain a distance of 6–15 cm, which may reveal the cohesion property of Reynolds rules. Even in a high-density flight arena (400–500 flies in a 40 cm cube), the accompanying flight behavior between fruit flies could potentially be a collision-avoidance strategy in swarm flight. Although our *Drosophila* groups comprised mixed-gender populations, and male flies might pursue females during flight, unlike mosquito[53], there is no evidence suggesting that flies can distinguish gender-specific characteristics during flight. Previous studies have documented the aggregation behavior in crawling fruit flies[48,54–56], challenging the stereotype that fruit flies are merely non-social creatures. Our observations of collision avoidance and accompanying flight in aerial interactions provide further evidence of social dynamics within *Drosophila* populations.

Methods	MOTP (↓)		MOTA (↑)		Tracking length ratio (↑) (%)
	View 1	View 2	View 1	View 2	
KAN-only	0.7884	0.6159	0.8543	0.8376	1.4134
Transformer-only	0.9141	0.7195	0.8435	0.8242	1.1783
KAN-transformer	0.7687	0.6029	0.8573	0.8394	1.4548

Table 2. Evaluation of our tracking method variants without KAN or Transformer.

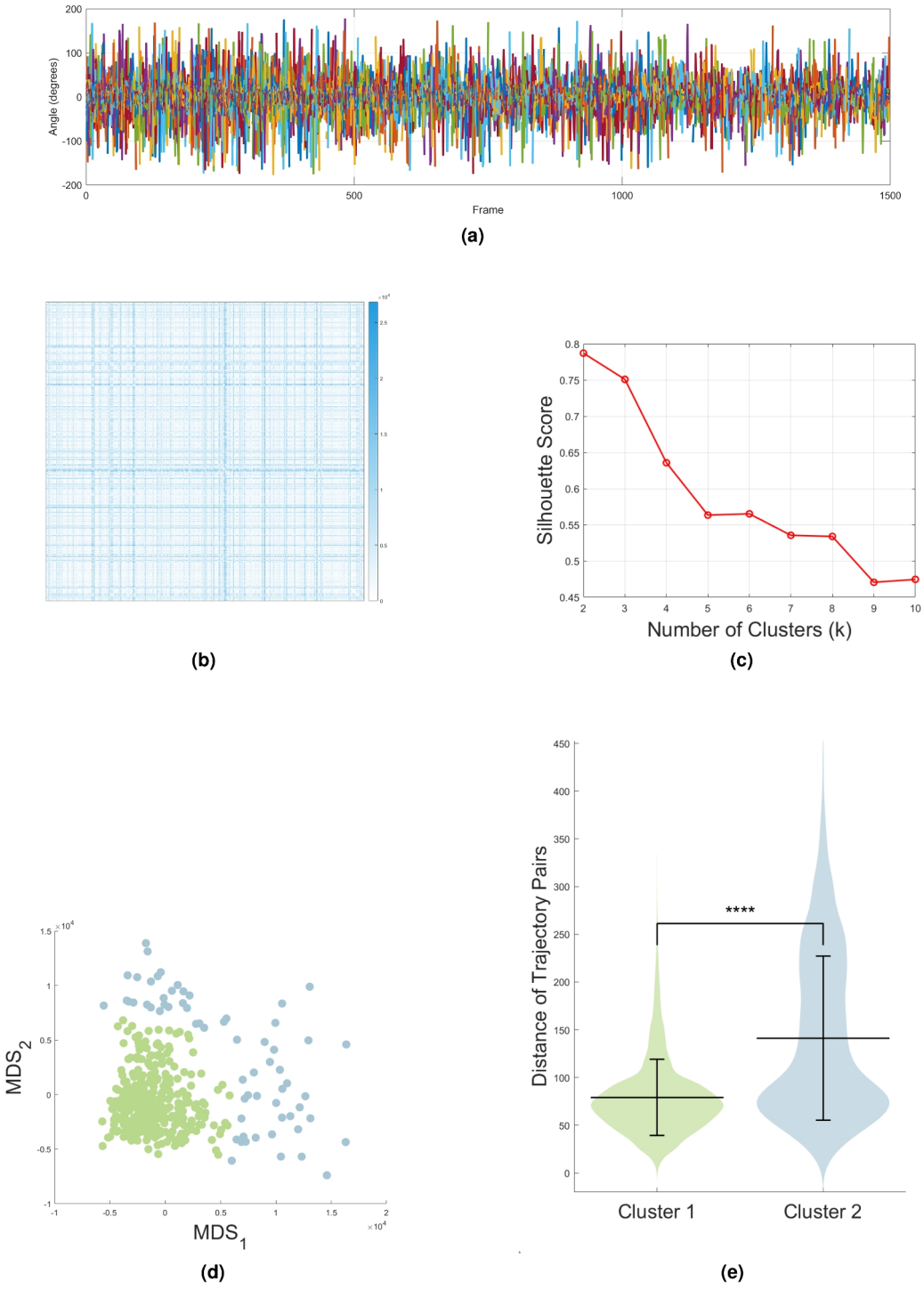


Fig. 6. Temporal correlations analysis in angular acceleration among *Drosophila* using DTW and K-means. (**a**) Angular acceleration of tracked trajectories. (**b**) Visualization of DTW-based distance matrix showing pairwise distances between trajectories. (**c**) Silhouette Score analysis demonstrating optimal clustering. (**d**) A two-dimensional space of final clustering based on the DTW distance matrix. (**e**) Euclidean distance between fruit flies in two clusters which are derived from (**d**). Cluster 1 represents accompanying flight cases and Cluster 2 represents collision-avoidance cases (Mann-Whitney test, mean ± std).

Discussion

In this paper, we present manually annotated datasets of real-world *Drosophila* trajectories to comprehensively evaluate the tracking performance of existing tracking methods. We design KAN-Transformer, a multi-object tracking method using the particle filter framework. Individual kinematic patterns are learned via a KAN-Transformer network that captures both long-range dependencies and fine-grained details in trajectory.

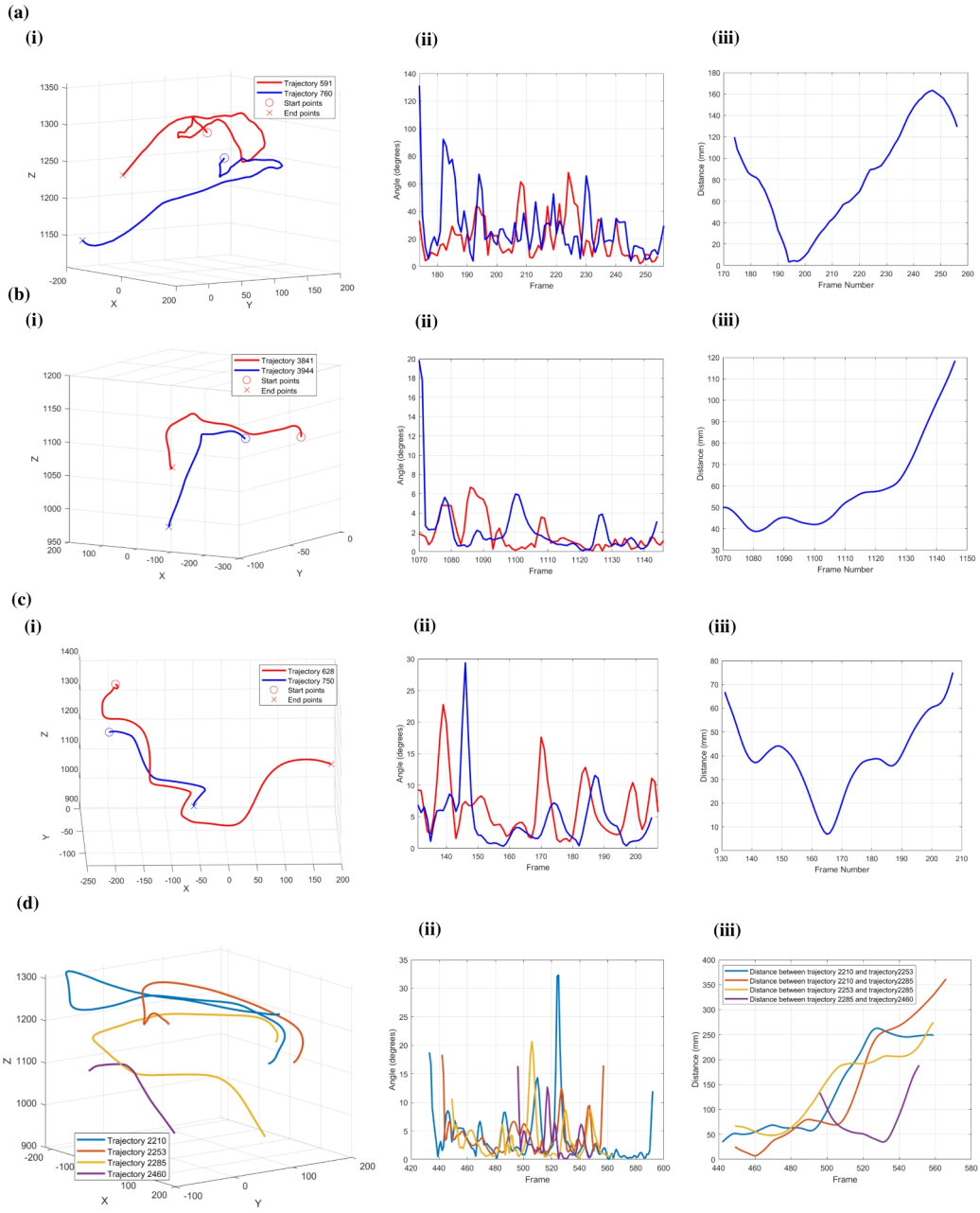


Fig. 7. Two behavioral patterns were identified in the flight trajectories, characterized by accompanying flight and collision avoidance maneuvers. (**a**) Example of collision avoidance behavior between two *Drosophila*. (**b**) Example of accompanying flight behavior between two *Drosophila*. (**c**) Complex interaction demonstrating the emergence of collision avoidance during accompanying flight when inter-fly distance decreases below a critical threshold. (**d**) Example of multiple fruit flies alternate in accompanying flight. (i) spatial trajectory visualization, (ii) angular acceleration and (iii) distance between trajectories..

The proposed method demonstrates state-of-the-art performance in tracking *Drosophila* within crowded environments. Analysis of tracked trajectories demonstrated two distinct behavioral patterns in *Drosophila*–accompanying flight and collision avoidance, which revealed the Reynolds rules of flocking behavior.

Although the method presented in this study performs well within the current experimental setup, which involves a small, enclosed space with approximately 400–500 dynamic agents, its application may face challenges in more complex dynamic backgrounds. Factors such as lighting conditions and the number of moving objects can also affect the performance of the algorithm. Therefore, in the detection phase, while the sliding window approach for background removal has proven effective, the integration of deep learning methods could further enhance system performance.

Recent advances in deep learning architectures[57,58], particularly in image processing and pattern recognition, might offer superior detection capabilities and robustness against environmental variations. In terms of

inference time cost on our test dataset, SeqFileProcessing2D[23] and Markov[22] were 0.34 and 3.92 seconds per frame, respectively. However, the time of LSTM[28], Transformer[30] and our method were 366.79, 374.53 and 368.83 seconds per frame, respectively. The reason is that our method here, as well as two other methods utilize particle filtering algorithms which relies on a large number of particles. This limitation could be mitigated through the implementation of GPU-accelerated parallel computing, which warrants further exploration in future development.

The inherent explainability features of KAN networks represent a promising avenue for future research, for example, action recognition[59–61]. A more thorough investigation into the network's interpretable components could provide valuable insights into Lévy flight[62] and potentially lead to more transparent and trustworthy results. Furthermore, while we have validated Reynolds rules in flying *Drosophila* groups, there remains significant potential for deeper behavioral analysis. Specifically, future investigations could explore whether *Drosophila* exhibit sophisticated local interactions and navigation strategies similar to those observed in other social animals[63–66]. Previous studies by Dickinson et al.[67,68] have revealed detailed body torque changes in individual flies facing looming and optic flow stimuli. Extending this approach to multi-fly scenarios through high-speed imaging could provide valuable insights into the fine-grained dynamics of collective flight behavior. These findings could inform both the development of collective intelligence systems[69] and the optimization of unmanned aerial vehicle performance[70].

Data availability

All data generated and annotated during this study are available upon reasonable request; please contact the corresponding author, hfsu@shu.edu.cn, for access.

Code availability

Code is available upon reasonable request.

Received: 4 December 2024; Accepted: 21 February 2025
Published online: 27 February 2025

References

1. Reynolds, A. M. & Ouellette, N. T. Swarm formation as backward diffusion. *Phys. Biol.* **20**, 026002 (2023).
2. Cavagna, A. et al. Natural swarms in 3.99 dimensions. *Nat. Phys.* **19**, 1043–1049 (2023).
3. Kelley, D. H. & Ouellette, N. T. Emergent dynamics of laboratory insect swarms. *Sci. Rep.* **3**, 1073 (2013).
4. Attanasi, A. et al. Collective behaviour without collective order in wild swarms of midges. *PLoS Comput. Biol.* **10**, e1003697 (2014).
5. Attanasi, A. et al. Information transfer and behavioural inertia in starling flocks. *Nat. Phys.* **10**, 691–696 (2014).
6. Katz, Y., Tunstrøm, K., Ioannou, C. C., Huepe, C. & Couzin, I. D. Inferring the structure and dynamics of interactions in schooling fish. *Proc. Natl. Acad. Sci.* **108**, 18720–18725 (2011).
7. Couzin, I. D., Krause, J., Franks, N. R. & Levin, S. A. Effective leadership and decision-making in animal groups on the move. *Nature* **433**, 513–516 (2005).
8. Cavagna, A. et al. Flocking and turning: A new model for self-organized collective motion. *J. Stat. Phys.* **158**, 601–627 (2015).
9. Buhl, J. et al. From disorder to order in marching locusts. *Science* **312**, 1402–1406 (2006).
10. Ballerini, M. et al. Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proc. Natl. Acad. Sci.* **105**, 1232–1237. https://doi.org/10.1073/pnas.0711437105 (2008).
11. Nagy, M., Ákos, Z., Biro, D. & Vicsek, T. Hierarchical group dynamics in pigeon flocks. *Nature* **464**, 890–893 (2010).
12. Shishika, D., Manoukis, N. C., Butail, S. & Paley, D. A. Male motion coordination in anopheline mating swarms. *Sci. Rep.* **4**, 6318 (2014).
13. Strandburg-Peshkin, A. et al. Visual sensory networks and effective information transfer in animal groups. *Curr. Biol.* **23**, R709–R711 (2013).
14. Rahwan, I. et al. Machine behaviour. *Nature* **568**, 477–486 (2019).
15. Wojke, N., Bewley, A. & Paulus, D. Simple online and realtime tracking with a deep association metric. In *2017 IEEE International Conference on Image Processing (ICIP)*, 3645–3649, https://doi.org/10.1109/ICIP.2017.8296962 (2017).
16. Zhang, Y., Wang, C., Wang, X., Zeng, W. & Liu, W. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vision* **129**, 3069–3087. https://doi.org/10.1007/s11263-021-01513-4 (2021).
17. Zhang, Y. *et al.* Bytetrack: Multi-object tracking by associating every detection box. In *Computer Vision - ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXII*, 1-21, https://doi.org/10.1007/978-3-031-20047-2_1 (Springer-Verlag, Berlin, Heidelberg, 2022).
18. Yuan, Y., Wu, Y., Zhao, L., Liu, Y. & Pang, Y. End-to-end multiple object tracking in high-resolution optical sensors of drones with transformer models. *Sci. Rep.* **14**, 25543 (2024).
19. Attanasi, A. et al. Greta-a novel global and recursive tracking algorithm in three dimensions. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**, 2451–2463. https://doi.org/10.1109/TPAMI.2015.2414427 (2015).
20. Dell, A. I. et al. Automated image-based tracking and its application in ecology. *Trends Ecol. Evol.* **29**, 417–428 (2014).
21. Walter, T. & Couzin, I. D. Trex, a fast multi-animal tracking system with markerless identification, and 2d estimation of posture and visual fields. *eLife* **10**, e64000. https://doi.org/10.7554/eLife.64000 (2021).
22. Manoukis, N. C., Butail, S., Diallo, M., Ribeiro, J. M. & Paley, D. A. Stereoscopic video analysis of anopheles gambiae behavior in the field: Challenges and opportunities. *Acta Tropica* **132**, S80–S85. https://doi.org/10.1016/j.actatropica.2013.06.021 (2014).
23. Angarita-Jaimes, N. et al. A novel video-tracking system to quantify the behaviour of nocturnal mosquitoes attacking human hosts in the field. *J. R. Soc. Interface* **13**, 20150974 (2016).
24. Jiang, Z., Chazot, P. L., Celebi, M. E., Crookes, D. & Jiang, R. Social behavioral phenotyping of drosophila with a 2d–3d hybrid CNN framework. *IEEE Access* **7**, 67972–67982 (2019).
25. Wu, H. S., Zhao, Q., Zou, D. & Chen, Y. Q. Automated 3d trajectory measuring of large numbers of moving particles. *Opt. Express* **19**, 7646–7663. https://doi.org/10.1364/OE.19.007646 (2011).
26. Wu, Z., Hristov, N. I., Hedrick, T. L., Kunz, T. H. & Betke, M. Tracking a large number of objects from multiple views. In *2009 IEEE 12th International Conference on Computer Vision*, 1546–1553, https://doi.org/10.1109/ICCV.2009.5459274 (2009).
27. Ardekani, R. et al. Three-dimensional tracking and behaviour monitoring of multiple fruit flies. *J. R. Soc. Interface* **10**, 20120547 (2013).

28. Wang, S. H. *et al.* Tracking the 3d position and orientation of flying swarms with learned kinematic pattern using lstm network. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, 1225–1230, https://doi.org/10.1109/ICME.2017.8019406 (2017).
29. Schmidhuber, J. et al. Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997).
30. Yin, C., Liu, X., Zhang, X., Wang, S. & Su, H. Long 3d-pot: A long-term 3d drosophila-tracking method for position and orientation with self-attention weighted particle filters. *Appl. Sci.* https://doi.org/10.3390/app14146047 (2024).
31. Vaswani, A. et al. Attention is all you need. In Guyon, I. *et al.* (eds.) *Advances in Neural Information Processing Systems*, vol. 30 (Curran Associates, Inc., 2017).
32. Xu, K., Chen, L. & Wang, S. Kolmogorov-Arnold networks for time series: Bridging predictive power and interpretability. arXiv preprint arXiv:2406.02496 (2024).
33. Vaca-Rubio, C. J., Blanco, L., Pereira, R. & Caus, M. Kolmogorov-Arnold networks (kans) for time series analysis. arXiv preprint arXiv:2405.08790 (2024).
34. Han, X., Zhang, X., Wu, Y., Zhang, Z. & Wu, Z. Kan4tsf: Are kan and kan-based models effective for time series forecasting? arXiv preprint arXiv:2408.11306 (2024).
35. Genet, R. & Inzirillo, H. A temporal kolmogorov-arnold transformer for time series forecasting. arXiv preprint arXiv:2406.02486 (2024).
36. Liu, Z. *et al.* Kan: Kolmogorov-arnold networks. arXiv preprint arXiv:2404.19756 (2024).
37. Bouguet, J.-Y. Camera calibration toolbox for matlab (1.0), https://doi.org/10.22002/D1.20164 (2022).
38. Zivkovic, Z. Improved adaptive gaussian mixture model for background subtraction. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 2, 28–31 Vol.2, https://doi.org/10.1109/ICPR.2004.1333992 (2004).
39. Hou, X. H. & Liu, H. H. Research of background modeling algorithm method based on multi-frame average method in moving target detection. *Appl. Mech. Mater.* **380**, 1390–1393 (2013).
40. Kolmogorov, A. N. *On the Representation of Continuous Functions of Several Variables by Superpositions of Continuous Functions of a Smaller Number of Variables* (American Mathematical Society, 1961).
41. Shazeer, N. et al. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations* (2017).
42. Li, Z., Qi, S., Li, Y. & Xu, Z. Revisiting long-term time series forecasting: An investigation on linear mapping. arXiv:abs/2305.10721 (2023).
43. Zeng, A., Chen, M., Zhang, L. & Xu, Q. Are transformers effective for time series forecasting? In Williams, B., Chen, Y. & Neville, J. (eds.) *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, 11121–11128, https://doi.org/10.1609/AAAI.V37I9.26317 (AAAI Press, 2023).
44. Kim, T. et al. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations* (2022).
45. Bernardin, K. & Stiefelhagen, R. Evaluating multiple object tracking performance: The clear mot metrics. *J. Image Video Process.* https://doi.org/10.1155/2008/246309 (2008).
46. Liu, Y., Wang, S. & Chen, Y. Q. Automatic 3d tracking system for large swarm of moving objects. *Pattern Recogn.* **52**, 384–396. https://doi.org/10.1016/j.patcog.2015.11.014 (2016).
47. Cheng, X. E., Wang, S. H. & Chen, Y. Q. Estimating orientation in tracking individuals of flying swarms. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1496–1500, https://doi.org/10.1109/ICASSP.2016.7471926 (IEEE Press, 2016).
48. Ramdya, P. et al. Mechanosensory interactions drive collective behaviour in drosophila. *Nature* **519**, 233–236 (2015).
49. Jiang, L. et al. Emergence of social cluster by collective pairwise encounters in *Drosophila*. *eLife* **9**, e51921. https://doi.org/10.7554/eLife.51921 (2020).
50. Zabala, F. et al. A simple strategy for detecting moving objects during locomotion revealed by animal-robot interactions. *Curr. Biol.* **22**, 1344–1350. https://doi.org/10.1016/j.cub.2012.05.024 (2012).
51. Lu, F. Y., Liu, X., Su, H. F. & Wang, S. H. Comparative analysis of tracking and behavioral patterns between wild-type and genetically modified fruit flies using computer vision and statistical methods. *Behav. Proc.* **222**, 105109. https://doi.org/10.1016/j.beproc.2024.105109 (2024).
52. Reynolds, C. W. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, 25-34, https://doi.org/10.1145/37401.37406 (Association for Computing Machinery, New York, NY, USA, 1987).
53. Cavagna, A. et al. Characterization of lab-based swarms of anopheles gambiae mosquitoes using 3d-video tracking. *Sci. Rep.* **13**, 8745 (2023).
54. Guo, A. et al. Vision, memory, and cognition in drosophila. *Learn. Theory Behav.* 483–503 (2017).
55. Burg, E. D., Langan, S. T. & Nash, H. A. Drosophila social clustering is disrupted by anesthetics and in narrow abdomen ion channel mutants. *Genes Brain Behav.* **12**, 338–347 (2013).
56. Simon, A. F. et al. A simple assay to study social behavior in drosophila: Measurement of social space within a group 1. *Genes Brain Behav.* **11**, 243–252 (2012).
57. He, K., Gkioxari, G., Dollár, P. & Girshick, R. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, 2961–2969 (2017).
58. Varghese, R. & M., S. Yolov8: A novel object detection algorithm with enhanced performance and robustness. In *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, 1–6, https://doi.org/10.1109/ADICS58448.2024.10533619 (2024).
59. Ma, N., Wu, Z., Feng, Y., Wang, C. & Gao, Y. Multi-view time-series hypergraph neural network for action recognition. *IEEE Trans. Image Process.* **33**, 3301–3313. https://doi.org/10.1109/TIP.2024.3391913 (2024).
60. Zhou, Y. *et al.* Blockgcn: Redefine topology awareness for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2049–2058 (2024).
61. Liu, J., Chen, C. & Liu, M. Multi-modality co-learning for efficient skeleton-based action recognition. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, 4909-4918, https://doi.org/10.1145/3664647.3681015 (Association for Computing Machinery, New York, NY, USA, 2024).
62. Viswanathan, G. M. et al. Lévy flight search patterns of wandering albatrosses. *Nature* **381**, 413–415 (1996).
63. Bird, C. M. & Burgess, N. The hippocampus and memory: Insights from spatial processing. *Nat. Rev. Neurosci.* **9**, 182–194 (2008).
64. Robinson, N. T. et al. Targeted activation of hippocampal place cells drives memory-guided spatial behavior. *Cell* **183**, 1586-1599.e10. https://doi.org/10.1016/j.cell.2020.09.061 (2020).
65. Ergorul, C. & Eichenbaum, H. The hippocampus and memory for "what," "where," and "when". *Learn. Memory* **11**, 397–405 (2004).
66. Patel, R. N., Kempenaers, J. & Heinze, S. Vector navigation in walking bumblebees. *Curr. Biol.* **32**, 2871–2883 (2022).
67. Muijres, F. T., Elzinga, M. J., Melis, J. M. & Dickinson, M. H. Flies evade looming targets by executing rapid visually directed banked turns. *Science* **344**, 172–177 (2014).
68. Lindsay, T., Sustar, A. & Dickinson, M. The function and organization of the motor system controlling flight maneuvers in flies. *Curr. Biol.* **27**, 345–358. https://doi.org/10.1016/j.cub.2016.12.018 (2017).

69. Naeem, M. R. et al. Harnessing AI and analytics to enhance cybersecurity and privacy for collective intelligence systems. *PeerJ Comput. Sci.* **10**, e2264 (2024).
70. Poudel, S., Arafat, M. Y. & Moh, S. Bio-inspired optimization-based path planning algorithms in unmanned aerial vehicles: A survey. *Sensors* **23**, 3051 (2023).

Acknowledgements
We thank Prof. Chi Wang and his graduate student Han Lu for their invaluable assistance in completing a comparative experiment. We thank Wenbo Shen, Ruiyi Xue, Lingxin Luo, and all 40 students in the course "Artificial Intelligence in Biological and Pharmaceutical Engineering" at Shanghai University for the 2024-2025 fall semester for their invaluable efforts in data annotation.

Author contributions
Qi Yang and Haifeng Su wrote the main manuscript text, Qi Yang prepared Figs. 1–7, Tables 1 and 2 and supplementary figures. All authors reviewed the manuscript.

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-025-91674-w.

Correspondence and requests for materials should be addressed to H.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.