Transportation Research Part C 174 (2025) 105115

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

A learning-based interacting multiple model filter for trajectory prediction of small multirotor drones considering differential sequences

Rong Tang [a], Kam K.H. Ng [a,b], Lishuai Li [c], Zhao Yang [d,*]

[a] Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hung Hom, Hong Kong Special Administrative Region of China
[b] Research Centre for Unmanned Autonomous Systems, The Hong Kong Polytechnic University, Hung Hom, Hong Kong Special Administrative Region of China
[c] School of Data Science, City University of Hong Kong, Kowloon Tong, Hong Kong Special Administrative Region of China
[d] College of General Aviation and Flight, Nanjing University of Aeronautics and Astronautics, Jiangjun Road No. 29, Nanjing 211106, China

ARTICLE INFO

Keywords:
Drone
Uncrewed traffic management
Trajectory prediction
Machine learning
Interacting multiple models

ABSTRACT

Reliable trajectory prediction is a cornerstone component for supporting various higher-level applications in uncrewed traffic management (UTM). The high flexibility of multirotor drones presents significant challenges to achieving accurate trajectory prediction. In scenarios involving the tracking of non-cooperative targets, insufficient information hinders the effectiveness of machine learning-based methods for forecasting future trajectories. To address these limitations, we conceptualise the problem as a multivariate time series prediction task and propose an innovative integrated flight trajectory prediction approach, including two distinct modules. The initial module employs three modified Transformer models to forecast future sequences of position, velocity, and acceleration in parallel for capturing diverse flight patterns. The subsequent module features a learning-based interacting multiple model (IMM) filter designed to fuse the three predicted sequences, regarded as pseudo measurements, by adaptively learning time-invariant transition probabilities. We conducted two experiments using 17 multirotor drone trajectory datasets collected from industrial and academic applications. The results demonstrate: i) integrated position sequence and discrete velocity approach can significantly enhance trajectory prediction accuracy; ii) the modified Transformer architecture shows substantial potential compared to baselines; iii) the learning-based IMM method yields superior prediction results on 15 new trajectory datasets, effectively simulating scenarios of managing unidentified drones in real-world contexts.

1. Introduction

With the booming commercial unmanned aerial vehicles (UAVs) market, small-scale consumer quadrotors today have been widely employed in various fields to offer more conveniences. In industry, drones are versatile tools designed to complete a variety of missions, including delivering products, assisting agencies in security monitoring, and providing temporary communication services in

* Corresponding author.
E-mail address: yangzhao@nuaa.edu.cn (Z. Yang).

https://doi.org/10.1016/j.trc.2025.105115
Received 13 July 2024; Received in revised form 9 February 2025; Accepted 20 March 2025
Available online 3 April 2025

disasters, etc. (Jin et al., 2024; Maza et al., 2011; Ri et al., 2024). For individuals, quadrotors as recreational devices can offer novel perspectives for photography compared to ground views or allow users to experience fast flying. However, these technological products have also been posing significant threats to airspace management authorities. Drones could be used to carry out malicious attacks or illegal reconnaissance due to their compact size, stealth, and low assembly threshold. These agile and autonomous gadgets have become right-hand accessories for carrying deadly bombs in terrorist attacks and local conflicts. Additionally, accidents involving unintended invasions of military and protected facilities have dramatically increased with the popularisation of small consumer drones (Solodov et al., 2018). Particularly, illegal flying activities have caused several interruptions in civil aviation, even leading to the temporary closure of the affected airports (Lykou et al., 2020; Pyrgies, 2019). In December 2018, for example, a drone intruded on the runway at London Gatwick Airport, causing the airport to be closed for several days and resulting in the cancellation of about 800 flights, affecting the travel of nearly 100,000 passengers (Swinney and Woods, 2022). In most cases, such drones posing threats are unidentified and non-cooperative objects that may not strictly follow procedures for mission approval, airspace application, and flight plan declaration.

To govern the low-altitude airspace, prevent sensitive installations from invasions, and ensure safe aviation operations, existing air traffic management frameworks such as Detect and Avoid (DAA) systems that offer surveillance, alerting, and guidance to integrate Unmanned Aircraft System (UAS) operations into the National Airspace System (NAS), should be equipped with advanced monitoring technologies to keep UAVs clear of other aircraft (e.g., maintaining separation and avoiding collision), particularly when dealing with non-cooperative drones (Cone et al., 2019; Fern et al., 2015). EUROCONTROL also published the specification of trajectory prediction to support the Single European Sky ATM Research (SESAR) programme, particularly in ATM deployment. The profile thoroughly introduced the logical breakdown of trajectory prediction, from flight intent estimation to trajectory computation (Eurocontrol, 2017). Real-time trajectory prediction is expected to be crucial in supporting UTM by enabling potential conflict detection and providing early warnings of abnormal flying behaviour (Ruseno and Lin, 2024). Similarly, in air traffic management, a substantial amount of research has focused on effectively mining massive flight trajectory, supporting trajectory-based operations in the next-generation air transportation system (Hrastovec and Solina, 2016; Mondoloni and Rozen, 2020; Sun et al., 2019; Verdonk Gallego et al., 2019, 2018).

In the field of trajectory prediction for mobile entities like aircraft, drones, and pedestrians, extensive research has employed data-driven approaches, notably multivariate time series modelling (Corbetta et al., 2019; Rudenko et al., 2020; Ruseno and Lin, 2024; Yang et al., 2020; Zeng et al., 2022). These models are well-designed to learn the movement patterns of existing homogeneous objects and to predict the future paths of target subjects over a time horizon. Typically, these hand-crafted models are constructed with different considerations. For example, predicting the future trajectory of a commercial aircraft involves considering its flight plan, airway network, and weather conditions (Zeng et al., 2022), while understanding the complex and subtle interactions between pedestrians is crucial for accurately forecasting their movements (Alahi et al., 2016; Liu et al., 2022). For small-scale multirotor drones, the movement patterns are distinct from fixed-wing drones due to their sudden manoeuvring and hovering. Applying machine learning-based methods to predict the trajectory of small-scale drones encounters two main challenges. First, achieving more accurate future position estimates requires integrating multiple data sources, such as onboard data and environmental factors. Second, these methods typically divide collected flight trajectories into three subsets for training, validation, and model evaluation. However, real-time data distribution shifts can degrade model performance in practical applications.

In scenarios involving the management of unidentified drones, such as those approaching no-fly zones in low-altitude airspace, positioning data is typically first captured by monitoring equipment like radar. However, other sources of data from various onboard sensors are often inaccessible simultaneously due to the lack of a direct data link between the surveillance platform and the monitored unknown target. On the other hand, the GPS positions during flight correspond to the integration of the drone's ground speed. It is assumed that the predicted trajectories implicitly account for environmental factors such as local winds, which are generally stable over short periods. Therefore, relying solely on the basic trajectory tracked to infer the flying patterns of the non-cooperative target and forecast its future flying states is interesting. It motivates us to develop a practicable trajectory prediction method under such circumstances. Given that the trajectory predicted in this study is defined as a sequence of waypoints and does not incorporate detailed engine performance, controllability, or weight configurations, we specify that our model is suitable for estimating flight intent for multi-rotor aircraft, consistent with the scope outlined in the profile (Eurocontrol, 2017).

This study focuses on multirotor drone surveillance and management in low-altitude airspace. We aim to (a) propose a machine learning-based approach that incorporates additional velocity and acceleration estimation into the future trajectory prediction of small multirotor drones, aiming to capture diverse flight patterns; (b) develop a learning-based interacting multiple model (IMM) model to adaptively integrate multiple upstream predictions, thereby enhancing the robustness of trajectory prediction; (c) validate the superior performance of the proposed method using up to 17 multirotor trajectory datasets collected from drone application scenarios in both industry and academia; and (d) demonstrate that employing models pre-trained on distribution-shifted trajectory data could significantly degrade prediction accuracy.

The rest of the paper is structured as follows. Section 2 reviews the related literature performed in the field of moving object prediction. The following section introduces the open-sourced trajectory dataset collected from different applications of multirotor drones. It shows how to discretise position sequences into velocity and acceleration series, which are used to formulate additional trajectory predictions. The novel learning-based IMM (LIMM) trajectory prediction model is described in Section 4. In Section 5, the numerical experiments compare and analyse the predictive performances between the proposed approach and the benchmark models. A brief discussion about the effectiveness of explicitly differencing trajectory into discrete velocity and acceleration series for improving prediction accuracy is presented in Section 6. Finally, Section 7 summarises the conclusions and discusses potential future studies of real-time trajectory prediction, particularly for non-cooperative drone management.

2. Related works

Similar to other moving targets such as commercial aircraft, vehicles, ships, and pedestrians (Abbas et al., 2020; Li et al., 2023a; Malviya and Kala, 2022; Zhang et al., 2023), trajectory prediction methods developed for UAVs can be broadly categorised into two fields: kinematic modelling-based and machine learning-based approaches.

(a) Kinematic modelling-based approaches

Kinematic modelling-based methods aim to construct dynamic and kinematic models by analysing the motion patterns of the target in question. To tackle the problems in which several dynamic modes or states could appear in uncertain probabilities and thus affect air traffic safety, the techniques of mixing multiple models have been adopted in various applications, including tracking aircraft positions and estimating their intentions for conflict detection and resolution (Jilkov et al., 2019; Rong Li and Jilkov, 2005; Wang and Huang, 2014), and identifying active guidance modes for aircraft to predict their future states in vertical climb and descent (Dalmau et al., 2018; Khaledian et al., 2023, 2022, 2020). For rotor drones, these methods utilise multi-degree-of-freedom kinematic equations combined with current states to predict future trajectories (Corbetta et al., 2019; Huang et al., 2016; Jiang et al., 2020; Luo et al., 2013). However, these approaches are often limited by the challenge of accurately measuring complex kinematics-related parameters. Consequently, they are primarily used for short-term trajectory prediction due to the significant accumulation of error over time.

(b) Machine learning-based approaches

In contrast to kinematic modelling-based methods, machine learning-based trajectory prediction methods rely solely on historical trajectory data treated as MTS. The core idea is that the influences of various factors on flight trajectory are implicitly embedded in the trajectory patterns. It is believed that many of these influencing mechanisms can be captured by learning from historical data, thus avoiding the need to establish complex motion models that require an understanding of hidden interactions between flight trajectories and various factors. Given this, increasing studies have focused on predicting the future trajectory of moving objects using machine learning over the past decade (Rudenko et al., 2020; Zeng et al., 2022). Through mining the historical data, particularly spatiotemporal trajectories, the moving patterns could be learned by functional regression models (Corbetta et al., 2019; Tastambekov et al., 2014), Gaussian mixture models (Lin et al., 2018; Wiest et al., 2012), Markov models and their variants (Ayhan and Samet, 2016; Liu and Hwang, 2011; Qiao et al., 2015), and Autoregressive Integrated Moving Average (ARIMA) models (Abebe et al., 2022; Alzyout et al., 2019).

In recent years, various neural networks have gained popularity and have been tailored to handle trajectory prediction tasks, demonstrating superior performance compared to conventional methods. Many of these approaches treat each trajectory as a multivariate time series in three or four-dimensional space and employ neural networks to forecast future trajectories. Notably, variants of recurrent neural networks (RNNs) such as long short-term memory (LSTM) and gated recurrent unit (GRU) have been widely adopted in trajectory prediction due to their ability to capture both short and long-term dependencies from historical data (Guan et al., 2023; Pang et al., 2019; Ruseno and Lin, 2024; Sahadevan et al., 2022; Shi et al., 2018; Yang et al., 2020). Moreover, to model complexities in hybrid systems, increasingly sophisticated model architectures, particularly attention-based mechanisms, have been specifically designed for deep learning of comprehensive information interactions in trajectory prediction (Geng et al., 2023; Guo et al., 2024; Jia et al., 2022; Li et al., 2023b; Liu et al., 2023; Zhang et al., 2023).

(c) Challenges

While machine learning-based methods often achieve higher prediction accuracy compared to kinematic modelling, they require extensive and diverse historical data sources to adequately train their models. In contrast, kinematic modelling-based methods are less reliant on historical data but require detailed performance parameters of the target, which can be difficult to obtain in certain scenarios. For instance, in urban low-altitude safety management and airport no-fly zone control, there is a need to prevent intrusion from non-cooperative drones that appear without rich historical trajectory data or detailed performance information. This presents a significant challenge when applying machine learning-based trajectory prediction methods in practical applications.

3. Trajectory prediction based on multivariate time series modelling

In this section, we first collected and organised a batch of trajectory datasets comprising 17 types of multirotor drones to support our experiments. Next, we introduced the task of machine learning-based flight trajectory prediction for non-cooperative drones, framed from the perspective of multivariate time series, emphasising the importance of considering stationarity in differential sequences. Our approach involves transforming trajectory (position) data into discretised velocity and acceleration sequences to enhance prediction accuracy. To capture diverse flight patterns, we employed three distinct Transformer models to forecast future positions, velocities, and accelerations in parallel. These models serve as the upstream prediction components of the integrated method discussed in the following section.

3.1. Data Preparation

(a) Data sources

To benchmark our proposed quadcopter trajectory prediction model against baselines and explore the potential of machine learning-based methods in air traffic surveillance and management within low-altitude airspace, we collected extensive flight trajectory data of multirotor drones. Our team collected the MavicAir2 dataset, while the remaining datasets were sourced from the website (https://cfreds.nist.gov/), where contributors share historical flight records of various drone types (CFReDS Portal, 2024).

In addition, we extracted four trajectory sets from the real-world industrial and academic scenarios: payload delivery (Rodrigues et al., 2021), agricultural applications (Mokhtarzadeh and Colten, 2015), UAV mosaicking and change detection (Avola et al., 2020), and UAV urban street localisation (Majdik et al., 2017), denoted as Payload, Farm, UMCD, and ZUMAVD, respectively. The first two datasets were collected from industrial activities, while the latter two were used in academic research. These datasets provide crucial trajectory-related information (e.g., GPS records), essential for evaluating the efficacy of machine learning models in practical trajectory prediction applications.

For extracting GPS data from historical flight records of commercial quadcopters like those from DJI, the previous works have provided the details on extracting GPS data from DAT or txt files and exporting them in csv format suitable for time series data analysis and pre-processing (Kumar and Agrawal, 2021). Notably, our model is designed to predict the trajectory of non-cooperative drones, given that onboard sensors cannot be monitored continuously by management agencies. The historical locations, however, can be tracked based on existing surveillance equipment. Thus, the data used in our model comprises 4D trajectory only (latitude, longitude, altitude, time) sampled at 1 Hz, despite the availability of various onboard sensor data (Inertial Measurement Unit (IMU), Gyroscope, Accelerometer, etc.).

(b) Data processing

Initially, the latitude and longitude records extracted from trajectories are in geodetic coordinates, and different altitude references (mean sea level and above ground level) are adopted across datasets. To facilitate data normalisation, we employ a fundamental method that transforms geodetic coordinates into local Cartesian coordinates (Ligas and Banasik, 2011). Subsequently, latitude and longitude units are converted from degrees to meters, with the initial position serving as the origin of the local Cartesian coordinate system. To meet the input and output length requirements of trajectory prediction models, each quadcopter flight trajectory is segmented into fixed-length segments. Additionally, to augment the dataset for deep learning purposes, we utilise a sliding window mechanism to generate overlapping trajectory segments from each flight trajectory (Chu, 1995). Given the potential for missing or erroneous position points in flight records, we implement a trajectory segment screening method based on maximum ground speed limitations and rates of climb and descent to filter out unreliable trajectory segments (Yang et al., 2020). Table 1 presents the number of segments obtained after segmentation and screening for each dataset, with each trajectory segment of 60 s. The table also includes partial statistics of trajectories for each dataset.

To compare the distributions of the trajectory segments shown in Table 1, the box plots for the discretised velocity and acceleration of each dataset are presented in Fig. 1. It can be observed that the ranges of velocity and acceleration vary significantly across datasets, presenting challenges for trajectory prediction when the monitored drone manoeuvres in patterns divergent from those in the training samples prepared beforehand. With this in mind, we conduct experiments to evaluate the performance of trajectory prediction models on datasets with both identical and shifted distributions in Section 5.

Table 1
The list of the trajectory and segment for each dataset.

Dataset	Flight duration (s)	Max velocity (m/s) lat/long/alt	Segment size
MavicAir2	25088	15.8/11.5/5.7	23528
Mavic2Enterprise	3573	10.2/6.9/2.2	3033
Phantom4Pro	2396	22.5/13.2/9.0	2097
Phantom3	2020	17.5/12.5/4.7	1720
Inspire2	2124	26.7/27.3/8.4	1646
Phantom4ProV2	1765	14.5/9.2/6.5	1585
MavicPro	1701	12.9/9.4/4.3	1521
Mavic2Zoom	1827	20.9/23.0/6.9	1467
Spark	1364	6.3/6.2/3.4	1027
MavicAir	1064	20.9/14.1/4.1	884
Inspire	955	23.4/21.1/6.2	776
Mavic2Pro	750	14.8/14.5/5.1	454
Matrice600Pro	318	22.9/14.8/3.8	258
Payload	38821	11.7/11.4/4.4	26284
Farm	2459	15.8/23.9/2.3	2399
UMCD	913	8.1/1.2/2.9	370
ZUMAVD	2708	3.4/3.9/0.0*	2468

* The drone nearly maintained level flight.

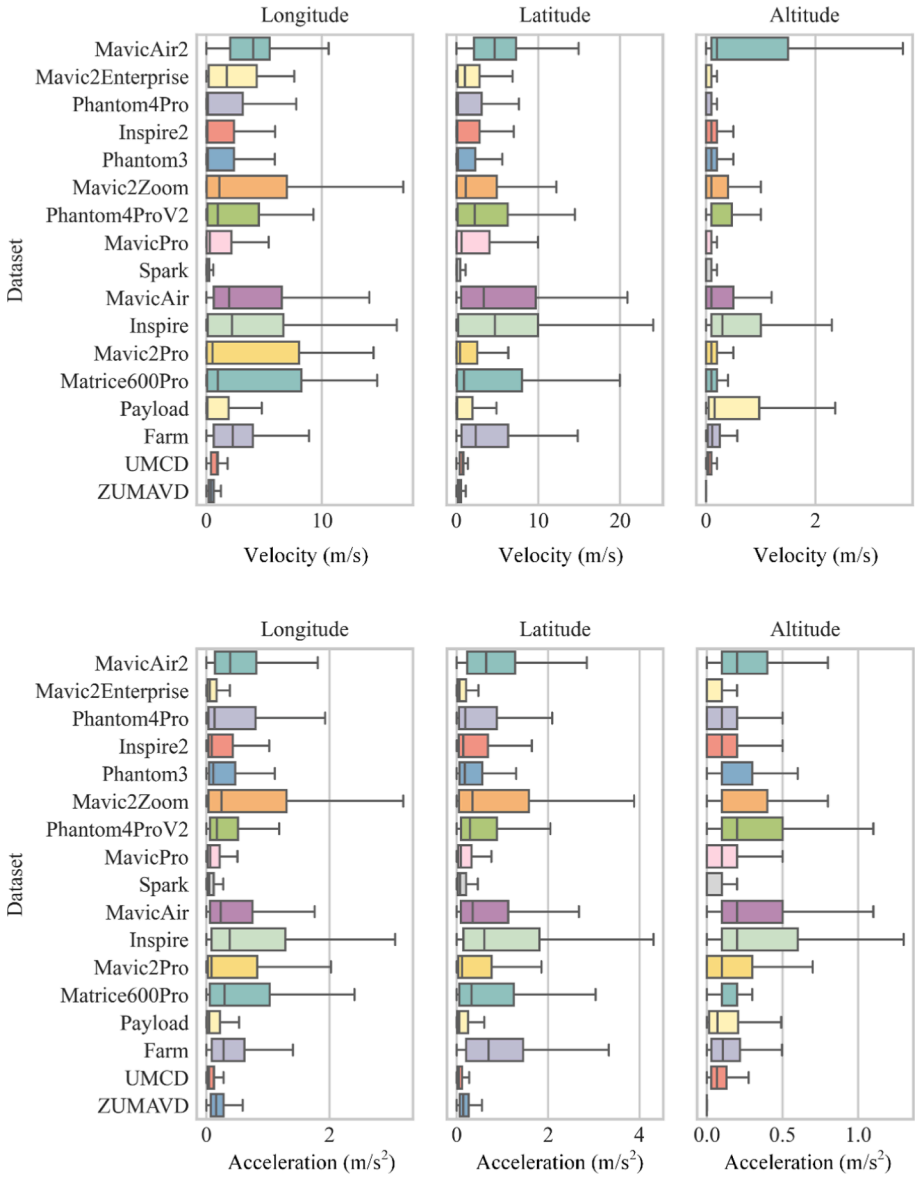

Fig. 1. The distributions of the velocity and acceleration for each dataset.

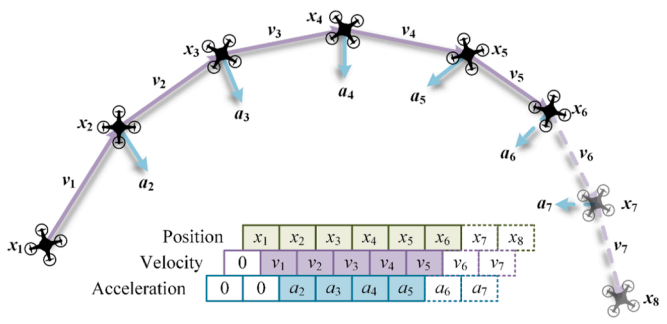

Fig. 2. A toy example of generating discrete velocity and acceleration sequences.

3.2. Differential trajectory prediction in parallel

To predict the future trajectory of a drone based on its past movements, we frame the task as a multivariate time series regression problem. The observed trajectory over the period $[1, T_{obs}]$ is represented as $\boldsymbol{X} = \{\boldsymbol{x}_t \in \mathbb{R}^3 | t = 1, 2, 3, \cdots, T_{obs}\}$, where $\boldsymbol{x}_t$ denotes the 3D location of the target at observation time t in coordinates. Similarly, the predicted trajectory at the current moment T_{obs} can be denoted as $\boldsymbol{Y} = \{\boldsymbol{x}_t \in \mathbb{R}^3 | t = T_{obs} + 1, \cdots, T_{obs} + T_{pred}\}$, with T_{pred} representing the length of the prediction horizon. Given the fixed time interval of 1 s between consecutive track points, the temporal dimension of the 4D trajectory is treated as the independent variable in this study for simplicity. Meanwhile, we focus on short-term trajectory prediction for real-time drone monitoring and management, with a look-ahead time of 20 s given the trajectory data from the preceding 40 s (i.e., $T_{obs} = 40$ and $T_{pred} = 20$).

In addition to inputting the historical position series to predict future locations directly, we propose generating discrete sequences of past velocities and accelerations. By forecasting future velocity and acceleration states, the corresponding positions can also be derived based on Newton's Second Law of Motion. Fig. 2 illustrates how to approximate the generation of discretised velocity and acceleration sequences from observed trajectory points.

Notably, $\boldsymbol{v}_t$ denotes the average velocity between positions $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$, while $\boldsymbol{a}_t$ represents the approximate acceleration at location $\boldsymbol{x}_t$. Combining the definition of trajectory $\boldsymbol{X}$ and the illustrative example in Fig. 2, we can utilise basic kinematic equations to estimate the future position at the time stamp $T_{obs} + 1$. Mathematically, these equations can be expressed as follows, assuming a time interval of one second:

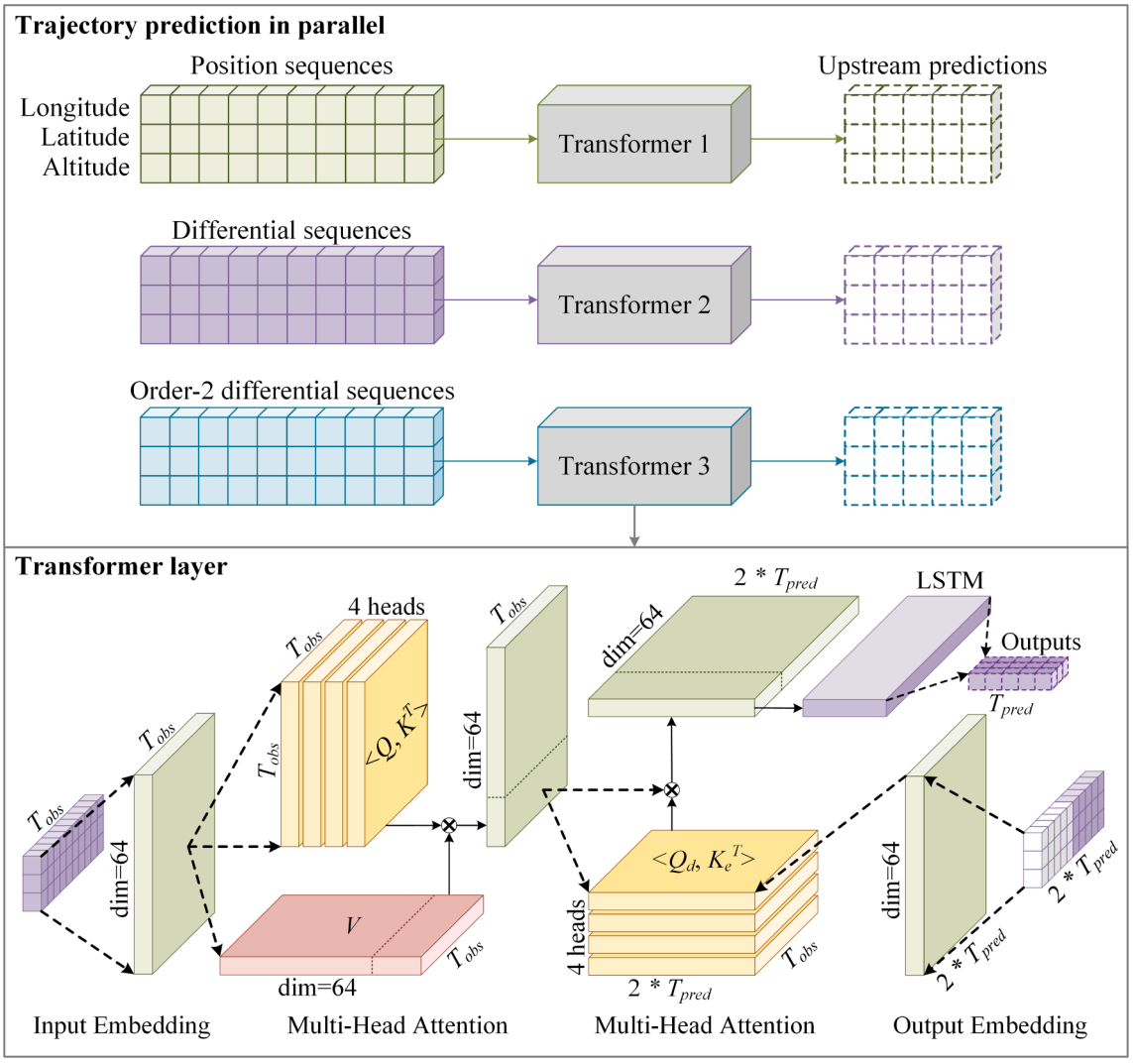

Fig. 3. The framework of differential trajectory prediction in parallel.

$$\begin{aligned} \boldsymbol{x}_{T_{obs}+1} &= f\left(\boldsymbol{X} = \left\{\boldsymbol{x}_t \in \mathbb{R}^3 | t = 1, 2, ..., T_{obs}\right\}\right) \\ \boldsymbol{x}_{T_{obs}+1} &= \boldsymbol{x}_{T_{obs}} + f_v\left(\boldsymbol{X}_v = \left\{\boldsymbol{v}_t \in \mathbb{R}^3 | t = 1, 2, ..., T_{obs} - 1\right\}\right) \\ \boldsymbol{x}_{T_{obs}+1} &= 2\boldsymbol{x}_{T_{obs}} - \boldsymbol{x}_{T_{obs}-1} + f_a\left(\boldsymbol{X}_a = \left\{\boldsymbol{a}_t \in \mathbb{R}^3 | t = 2, ..., T_{obs} - 1\right\}\right) \end{aligned} \tag{1}$$

where f f_v and f_a are mapping functions (e.g., kinematic motion equations and neural networks) that take as input historical position, velocity, and acceleration sequences, denoted as $\boldsymbol{X}$, $\boldsymbol{X}_v$, and $\boldsymbol{X}_a$, respectively. These equations provide a straightforward approach to predict future multi-step positions iteratively based on the estimated velocity and acceleration derived from historical trajectory data.

By leveraging the additional velocity and acceleration sequences derived from the original trajectory, we employ three separate prediction models in parallel to forecast future flight states, as illustrated in Fig. 3. In this study, we select Transformer architecture, renowned for its effectiveness in multivariate time series forecasting, including trajectory prediction for pedestrians and moving objects (Gu et al., 2022; Ren et al., 2023), weather forecasting (Bi et al., 2023) and other UCR time series regression tasks (Zerveas et al., 2021), as the core predictor. We modified the vanilla Transformer architecture as follows. Specifically, one of the three stacked layers in the modified architecture is depicted at the bottom of Fig. 3, where light green cubes represent the output features of each marked embedding and attention layer, and purple cubes indicate the inputs and outputs of the prediction model. We add an earlier slice before the output sequence of the decoder as designed in previous works (Zhou et al., 2021). This slice is identical to the last segment of the input and shares an equal length with the output. It should be noted that the structure shown here includes embedding and attention modules. Additional forward connection layers, such as layer normalisation and dense layers, are omitted. Further details can be found in the sources (Vaswani et al., 2017).

In the decoding module, we introduce an LSTM layer between the decoder and the output layer to recursively build the relationships between multi-step predictions. In addition, the multi-step trajectory is predicted once, regularised by the integral loss, rather than adopting iterative prediction as executed in aircraft trajectory prediction at cruise level (Zhang et al., 2023). It is regarded as an important training policy in such manoeuvring drone flight trajectory prediction after comparisons, which shows that the accumulative prediction error increases dramatically with the extension of the prediction horizon for iterative prediction. By contrast, sequence-based prediction gains from the additional correlation between consecutive trajectory points and tends to minimise the overall loss across multiple steps with less overfitting, though it might be slightly inferior to the iterative prediction in the initial step.

These predicted trajectories, termed upstream predictions, serve as inputs to the final integrated trajectory prediction model detailed in the subsequent section.

4. Integrated trajectory prediction

In this section, we first introduce the baseline IMM model. An integrated flight trajectory prediction method is designed to combine multiple upstream trajectory predictions from different models using IMM filtering. The time-variant transition probability matrix is dynamically learned from trajectory data by a neural network.

4.1. Overview of the basic IMM

The IMM-based algorithms, renowned for their effectiveness in state estimation for hybrid systems, are extensively used in tracking manoeuvring targets where multiple filters estimate states (García et al., 2015; Li and Bar-Shalom, 1993; Maeder et al., 2011; Mazor et al., 1998). These algorithms incorporate elemental filters such as Kalman Filter (KF), Particle Filter (PF), or probabilistic data association filters (PDAF) to leverage different modelling (Mazor et al., 1998). The basic IMM algorithm is designed to recursively integrate multiple predictions and weigh filtered estimates based on their likelihoods. Each cycle of the algorithm comprises four main modules: interaction, filtering, probability evaluation, and combination. Adapting the baseline IMM to predict the flight trajectory of multirotor drones to get the final estimation of the position at time $k+1$, one cycle of the algorithm can be formulated as follows:

(a) Interaction

When getting the combined state $\widehat{\boldsymbol{x}}_k^j$ and covariance $\widehat{\boldsymbol{P}}_k^j$ for each mode $m^j, j \in [1, M]$ at the last step k, instead of directly using them to predict the state and covariance for the next step, an interaction among a set of modes is performed to mix the multiple predictions based on the transition probability matrix $\boldsymbol{A}$ in which the element p^{ij} indicates the possibility from mode i at step k to mode j at step $k+1$. The transition probability is governed by the first-order homogeneous Markov chain, defined as:

$$p^{ij} = P\left\{m_{k+1}^j | m_k^i\right\}, \forall i, j \in M \tag{2}$$

Combining the updated weights $\{\mu_k^i \mid i \in [1, M]\}$ at the last step k, the mixed transition weight $\mu^{i|j}$, indicating the weighted possibility from mode i to j over the two sequential steps for $\forall i, j \in [1, M]$, can be expressed as:

$$\begin{aligned} \mu_k^{i|j} &= p^{ij}\mu_k^i \Big/ \overline{c}^j \\ \overline{c}^j &= \sum_i p^{ij}\mu_k^i \end{aligned} \tag{3}$$

Then, the interaction is designed as the weighted accumulation between these quantities, to produce the mixed state $\overline{\boldsymbol{x}}_k^j$ and covariance $\overline{\boldsymbol{P}}_k^j$ at the last step k, respectively.

$$\begin{aligned}\overline{\boldsymbol{x}}_k^j &= \sum_i \widehat{\boldsymbol{x}}_k^i \mu_k^{ij} \\ \overline{\boldsymbol{P}}_k^j &= \sum_i \left[\widehat{\boldsymbol{P}}_k^i + \left(\widehat{\boldsymbol{x}}_k^i - \overline{\boldsymbol{x}}_k^j\right)\left(\widehat{\boldsymbol{x}}_k^i - \overline{\boldsymbol{x}}_k^j\right)^T\right]\mu_k^{ij}\end{aligned} \tag{4}$$

which will serve as inputs for each estimator in the next step.

(b) Filtering

The mode-conditioned filtering in baseline IMM follows a standard cycle of the basic KF and is conducted for each mode in parallel. Further details can be found in relevant literature (Khodarahmi and Maihami, 2023). To summarise, given the mode-dependent process noise with mean $\overline{\boldsymbol{v}}_k^j$ and covariance $\overline{\boldsymbol{Q}}_k^j$, the predicted state and covariance for mode j at the next step $k+1$ are formulated as:

$$\begin{aligned}\widehat{\boldsymbol{x}}_{k+1|k}^j &= \boldsymbol{F}_k^j \overline{\boldsymbol{x}}_k^j + \Gamma_k^j \overline{\boldsymbol{v}}_k^j \\ \widehat{\boldsymbol{P}}_{k+1|k}^j &= \boldsymbol{F}_k^j \overline{\boldsymbol{P}}_k^j \left(\boldsymbol{F}_k^j\right)^T + \Gamma_k^j \boldsymbol{Q}_k^j \left(\Gamma_k^j\right)^T\end{aligned} \tag{5}$$

where $\boldsymbol{F}_k^j$ and Γ_k^j are the state transition matrix and process noise gain matrix, respectively. After that, the update step involves the calculation of the residual $\boldsymbol{r}_{k+1}^j$ and the corresponding covariance $\boldsymbol{S}_{k+1}^j$ between the measurement $\boldsymbol{z}_{k+1}^j$ and predicted state $\widehat{\boldsymbol{x}}_{k+1|k}^j$ at the step $k+1$ for mode j. They will be used to obtain the filter gain $\boldsymbol{K}_{k+1}^j$:

$$\begin{aligned}\boldsymbol{r}_{k+1}^j &= \boldsymbol{z}_{k+1}^j - \boldsymbol{H}_{k+1}^j \widehat{\boldsymbol{x}}_{k+1|k}^j \\ \boldsymbol{S}_{k+1}^j &= \boldsymbol{H}_{k+1}^j \widehat{\boldsymbol{P}}_{k+1|k}^j \left(\boldsymbol{H}_{k+1}^j\right)^T + \boldsymbol{R}_{k+1}^j \\ \boldsymbol{K}_{k+1}^j &= \widehat{\boldsymbol{P}}_{k+1|k}^j \left(\boldsymbol{H}_{k+1}^j\right)^T \left(\boldsymbol{S}_{k+1}^j\right)^{-1}\end{aligned} \tag{6}$$

where $\boldsymbol{H}_{k+1}^j$ is used to map the state to measurements space and is set as an identity matrix in this work. $\boldsymbol{R}_{k+1}^j$ represents the covariance of the measurement noise. Subsequently, the filtered state $\widehat{\boldsymbol{x}}_{k+1}^j$ and covariance $\widehat{\boldsymbol{P}}_{k+1}^j$ can be obtained:

$$\begin{aligned}\widehat{\boldsymbol{x}}_{k+1}^j &= \widehat{\boldsymbol{x}}_{k+1|k}^j + \boldsymbol{K}_{k+1}^j \boldsymbol{r}_{k+1}^j \\ \widehat{\boldsymbol{P}}_{k+1}^j &= \widehat{\boldsymbol{P}}_{k+1|k}^j - \boldsymbol{K}_{k+1}^j \boldsymbol{H}_{k+1}^j \widehat{\boldsymbol{P}}_{k+1|k}^j\end{aligned} \tag{7}$$

(c) Probability evaluation

The possibility of residual in the normal distribution with the mean 0 and covariance $\boldsymbol{S}_{k+1}^j$ is regarded as the likelihood for mode j, denoted $\boldsymbol{\Lambda}_{k+1}^j$:

$$\boldsymbol{\Lambda}_{k+1}^j = \mathcal{N}\left(\boldsymbol{r}_{k+1}^j; 0, \boldsymbol{S}_{k+1}^j\right) \tag{8}$$

It's then used to update the combination weight μ_{k+1}^i as follows:

$$\begin{aligned}c &= \sum_i \boldsymbol{\Lambda}_{k+1}^j \overline{\boldsymbol{c}}^i \\ \mu_{k+1}^j &= \boldsymbol{\Lambda}_{k+1}^j \overline{\boldsymbol{c}}^j \Big/ c\end{aligned} \tag{9}$$

where c and $\overline{c}$ are both normalising factors, the later one is calculated in the interaction section.

(d) Combination

Finally, a weighted sum of the filtered state estimations of all modes yields the final state $\widehat{\boldsymbol{x}}_{k+1}$ at step $k+1$. Likewise, the combined covariance $\widehat{\boldsymbol{P}}_{k+1}$ is updated as:

$$\begin{aligned}\widehat{\boldsymbol{x}}_{k+1} &= \sum_j \widehat{\boldsymbol{x}}_{k+1}^j \mu_{k+1}^j \\ \widehat{\boldsymbol{P}}_{k+1} &= \sum_j \left[\widehat{\boldsymbol{P}}_{k+1}^j + \left(\widehat{\boldsymbol{x}}_{k+1}^j - \widehat{\boldsymbol{x}}_{k+1}\right)\left(\widehat{\boldsymbol{x}}_{k+1}^j - \widehat{\boldsymbol{x}}_{k+1}\right)^T\right]\mu_{k+1}^j\end{aligned} \tag{10}$$

So far, a cycle of the basic IMM model has been completed. We can get multistep predictions by iteratively executing the four modules, with each cycle producing progressively updated estimates of the state and covariance, thereby allowing for robust and accurate trajectory prediction over multiple future time steps.

4.2. A learning-based IMM

Combining with the structure of the basic IMM introduced in the last Section, the learning-based IMM is implemented through the following designs.

(a) Pseudo measurements

As outlined in Section 3, we developed three distinct neural networks to forecast future trajectories over multiple steps. They are designed to input historical sequences of position, velocity, and acceleration, respectively, and minimize the corresponding loss between the predicted values and the ground truth. The primary goal is to capture flight patterns from varying perspectives. Despite their similar structures, these models are expected to learn distinct motion patterns due to their different learning objectives. In the experiments section, it becomes evident that models trained on positional, velocity, and acceleration trends exhibit notable performance variations in multistep trajectory prediction. Motivated by these findings, we aim to enhance prediction accuracy and robustness using information fusion through IMM models, treating the predicted position, velocity, and acceleration results as pseudo-measurements.

(b) Embedding neural networks into the filtering module

Artificial neural networks have been successfully embedded into KF as a kind of new-format predictor to estimate the future states of moving objects due to the capability of arbitrary nonlinear mapping (Coskun et al., 2017). Instead of specifying fixed linear or nonlinear state transition matrix $\boldsymbol{F}$, process noise covariance $\boldsymbol{Q}$, and measurement noise covariance $\boldsymbol{R}$, we leveraged three different neural networks to dynamically estimate these components, respectively. In each mode, the prediction procedure in the filtering

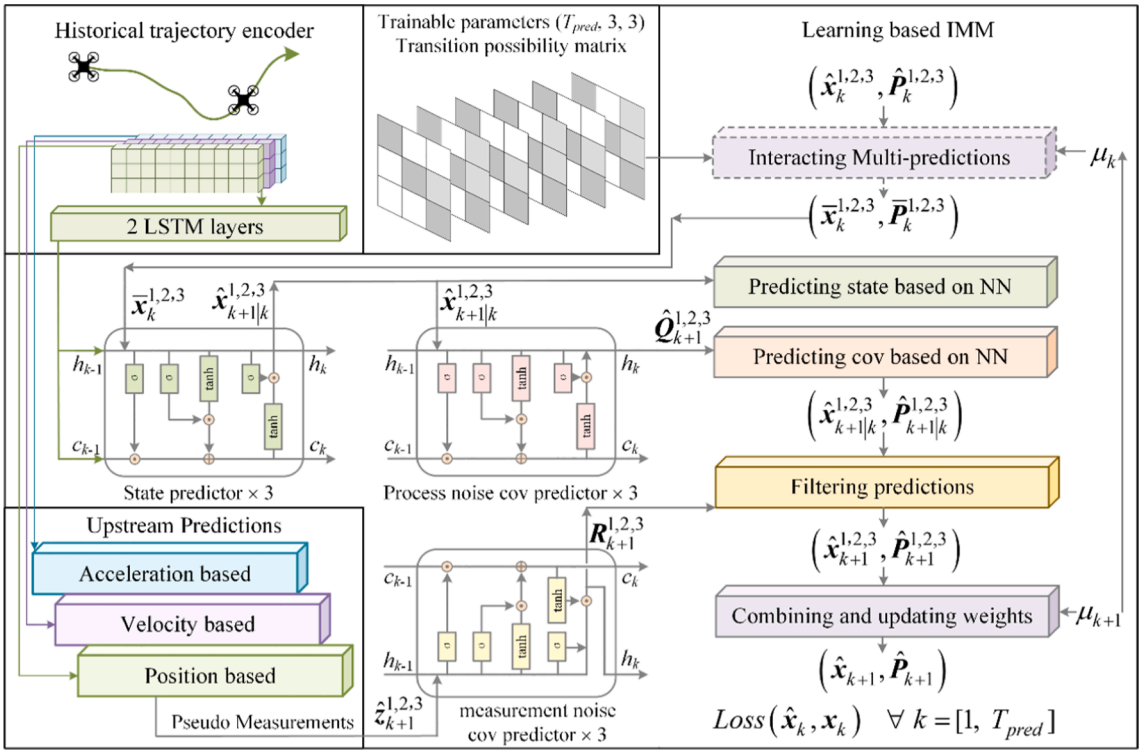

Fig. 4. The diagram of the learning-based IMM trajectory prediction model.

module (i.e., Eq. (5)) could be rewritten as follows:

$$\begin{aligned}\widehat{\boldsymbol{x}}^{j}_{k+1|k} &= f_x\left(\overline{\boldsymbol{x}}^{j}_{k}\right)\\ \widehat{\boldsymbol{P}}^{j}_{k+1|k} &= \overline{\boldsymbol{P}}^{j}_{k} + f_Q\left(\widehat{\boldsymbol{x}}^{j}_{k+1|k}\right)\end{aligned} \tag{11}$$

Meanwhile, the matrix $\boldsymbol{R}^{j}_{k+1}$ in Eq. (6) could be estimated as:

$$\boldsymbol{R}^{j}_{k+1} = f_R\left(\boldsymbol{z}^{j}_{k+1}\right) \tag{12}$$

where f_x, f_Q, and f_R are modelled using three different LSTM cells, leveraging their recursive prediction capabilities (Graves, 2013). Their initial hidden states and cell states are initialised with the final states of the last encoding LSTM layer. The state transition matrix $\boldsymbol{F}$ is omitted for simplification, under the assumption that neural network models can perform nearly arbitrary nonlinear mappings. Originally, it should be the Jacobian matrix of f_x with respect to the input $\overline{\boldsymbol{x}}^{j}_{k}$.

(c) Adaptive time-variant transition probability matrix

A crucial step in applying the IMM model for interactively predicting flight trajectory involves setting the transition probability matrix, which governs the switching between different prediction models. In tracking moving objects, previous studies often use predefined transition probability matrices based on empirical knowledge (Jo et al., 2012; Zhu et al., 2016; Zubača et al., 2022). Moreover, while time-variant transition probabilities have been utilised for predicting vehicle trajectories, they often adhere to fixed patterns and rely extensively on domain-specific expertise (Xie et al., 2018). Motivated by the potential of estimating parameters in hybrid systems through gradient descent (Brandenburger et al., 2023), we propose to learn the transition probability matrix adaptively by constructing an end-to-end neural network model. This allows the IMM model to learn the most likely transitions between model candidates based on historical trajectory data. As a result, the learned transition probability matrix is time-variant and capable of adjusting the model's behaviour to improve the accuracy of state estimation.

The overall feature flow of the proposed LIMM model is depicted in Fig. 4. Specifically, the transition probability matrix $\boldsymbol{A} \in \mathbb{R}^{T_{pred}\times 3\times 3}$ in the built LIMM model acting as a decoder, will be a group of learned parameters, being updated along with the embedded LSTM predictors through the backpropagation of the training loss. In this way, it is enabled to distribute different weights to multiple predictions across timestamps in the interaction step, guided by the minimum loss of the final trajectory prediction. To execute the IMM model as the pipeline, including 4 steps, described in Section 4.1, multiple measurements $\boldsymbol{z}$ which are essential in filter-based methods, are played with the upstream predictions described in Section 3, which is figured on the lower left in Fig. 4. Combined with the predictors alternated from the traditional kinematics to LSTM layers to generate the state $\widehat{x}$, covariance $\widehat{\boldsymbol{P}}$, process noise covariance $\boldsymbol{Q}$, and measurement noise covariance $\boldsymbol{R}$, as depicted in the middle of Fig. 4. So far, all essential components are prepared well and sent into the IMM filtering framework constructed in Section 4.1 to sequentially predict the future trajectory with the loss constraint as shown on the right in Fig. 4. The LIMM model consists of both end-to-end LSTM layers used to learn the suitable parameters based on historical data and the filtering process in the classical IMM model for explicitly mixing of multiple predictions.

It's noted that skipping the interaction process can generate an approximate result (i.e., using the final combined state $\widehat{x}$ to predict the next one, instead of the mixed state $\overline{x}$) as marked dash box in Fig. 4, to speed up the training process.

5. Experiments

To verify the effectiveness of the proposed integrated model on flight trajectory prediction of multirotor drones, this section conducts a series of comparisons by utilising the prepared trajectory datasets in Section 3.1. The setup of the training procedure in this work is presented first. Then, the results from leveraging differential sequences and learning-based IMM filtering to improve prediction accuracy, are presented in Sections 5.2 and 5.3, respectively.

5.1. Experimental setup

To assess prediction performance on both identical and shifted distributions, two experiments are conducted, as detailed in Sections 5.2 and 5.3. The first experiment evaluates the performance of the proposed model on data with an identical distribution, while the second experiment serves a dual purpose: to demonstrate the superiority of the LIMM model and to highlight the underestimated challenges of using pre-trained models to predict trajectories under shifted distributions.

In the first experiment, the MavicAir2 and Payload datasets are treated as the historical data pool. The trajectories in each dataset are carefully split into training, validation, and testing sets in a 7:1:2 ratio, maintaining the timeline order. Specifically, earlier trajectories are used for training, while more recent trajectories are assigned to the validation and testing sets. A sliding window with a predefined length (e.g., 60 s) is then applied to generate trajectory segments, treated as "samples" for trajectory prediction. Unlike the first experiment, where the test set is split from the same historical data as the training and validation sets and is thus considered identically distributed, the second experiment evaluates prediction performance on shifted distributions. For this, 15 additional trajectory datasets are used as fresh test data to simulate newly encountered samples in real-world applications. These test datasets have

no temporal connections with the two training datasets. The rationale behind selecting MavicAir2 and Payload as the combined training set while using the other datasets as test sets primarily revolves around segment sizes, as shown in Table 1. We desire that the datasets used for training should be substantial and comprehensive, while the test sets should be diverse to reflect potential future scenarios. Consequently, we train the prediction models on two large datasets and evaluate their performance on several distinct datasets. To train the trajectory prediction models properly, more detailed procedures are described below.

(a) Trajectory normalisation

Considering the divergence in flight speeds and the variability in manoeuvres across different trajectory datasets, global z-score normalisation may not adequately address such multimodal datasets. When the test set exhibits a significantly different distribution, it can lead to inaccuracies as the predicted normalised results may not align with the global parameters established during training (Passalis et al., 2020). Indeed, the challenges posed by spatiotemporal trajectories extend beyond what traditional normalisation methods like z-score can effectively handle. Unlike classically stationary time series, where the mean and standard deviation remain relatively constant over time, spatiotemporal trajectories exhibit varying means and deviations as the object moves through different spatial and temporal contexts. This variability makes it difficult for z-score normalisation, which assumes a static statistical distribution, to appropriately scale the data across all time points (Ogasawara et al., 2010). That's one of the motivations for differencing position series into velocity and acceleration sequences.

Min-Max and Maximum Absolute (MaxAbs) normalisation methods in time series prediction suffer from fixed global upper and lower bounds determined by the training dataset, limiting their ability to accurately map test data that extends beyond these bounds. These methods are also sensitive to outliers, as extreme points in the training data can skew the minimum and maximum values. Additionally, normalisation based on the global training dataset may lead to either overestimation or underestimation of prediction errors. For instance, consider Table 1 data: if the training set, such as the Payload dataset, has a narrower range of flight speeds, Min-Max and MaxAbs scaling may not be suitable for testing the Inspire2 dataset, which features a broader range. Conversely, even a minor adjustment in normalised space can result in disproportionately large transformations in the physical world.

To address these issues, we propose a re-normalisation method that aims to overcome the limitations mentioned by adjusting the scaling scope within each trajectory independently rather than using global dataset bounds. Specifically, it incorporates self-scope adjustment into the initial MaxAbs normalisation for every trajectory individually, integrating extended lower and upper bounds for each feature. For each sample $i = 1, 2, \ldots, N$, the feature $x_{i,k,d}$ for $\forall\ d = 1, 2, \ldots, D$ at time k in initial normalisation can be expressed as:

$$x'_{i,k,d} = \frac{x_{i,k,d}}{x_d^{\max}} \tag{13}$$

where $x_d^{\max}$ denotes the extended maximum absolute value of feature d in all samples, defined as follows:

$$x_d^{\max} = \max_{i\in[1,N],k\in[1,T]}\left(\left\{\left|x_{i,k,d}\right|\right\}\right) + 1.5 \times IQR_d \tag{14}$$

where IQR_d is the interquartile range for feature d in all samples. The primary concept behind extending the maximum absolute value of each feature is to leave a margin for the potential cumulative increase of the position values of flying drones.

While testing, we proposed to re-normalise the input series if they include a wider range of features, causing the normalised values to exceed the scope of [1-, 1]. Specifically, we calculate the extended scalers for each feature based on the input series of each normalised sample i itself and use these scalers to rescale the normalised sample within the range of [-1, 1]. Thus, the renormalised value of the feature d of sample i at time k is obtained:

$$\begin{aligned} x''_{i,k,d} &= \frac{x'_{i,k,d}}{s_{i,d}^{\max}} \\ s_{i,d}^{\max} &= \max_{k\in[1,T]}\left(\left\{\left|x'_{i,k,d}\right|\right\}\right) + 1.5 \times IQR'_{i,d} \end{aligned} \tag{15}$$

where $IQR'_{i,d}$ is the interquartile range for feature d in normalised sample i. When inversely transforming the predicted value into original space, we can multiply the two scales $s_{i,d}^{\max}$ and $x_d^{\max}$ for each sample i. In fact, the renormalisation based on each sample itself is the case of instance normalisation for time series if regarding each feature sequence as the whole pixels in one channel (Ulyanov et al., 2017).

(b) Baselines

To test the effectiveness of the proposed IMM trajectory prediction model, several categories of neural network architectures are simultaneously compared as baselines. They have been widely utilised in the field of trajectory prediction of moving objects. To be specific, the abbreviated labels and core structures of the selected benchmarks are described as follows:

LSTM and the **flattened** output **(LSTM_F)**: it consists of two modules, using three LSTM layers to encode the input features and a fully connected layer to generate the flattened 3D trajectory outputs. For the task of forecasting T-step 3D flight trajectory in the future,

for example, the fully connected layer will output a 1D series with the length of T * 3.

GRU and the **flattened** output **(GRU_F)**: the LSTM layers in LSTM_F are substituted with GRU layers, another improved RNN (Chung et al., 2014).

Bidirectional LSTM and the **flattened** output **(BiLSTM_F)**: two bidirectional LSTM (BiLSTM) layers are used in comparison with LSTM_F to capture reverse hidden features (Bahdanau et al., 2016).

1D Convolutional LSTM and the **flattened** output **(ConvLSTM1D_F)**: a convolution-based model including three 1D convolutional LSTM layers and a final fully connected layer, are built to predict flattened multistep trajectory as same as LSTM_F (SHI et al., 2015).

LSTM and the **sequential** output with **attention** mechanism **(LSTM_SA)**: an additional LSTM layer with an attention mechanism is employed in the decoder to recursively predict multistep trajectory, called sequence to sequence (Cho et al., 2014; Sutskever et al., 2014).

GRU and the **sequential** output with **attention** mechanism **(GRU_SA)**: it is similar to LSTM_SA, while the LSTM components are replaced with GRU counterparts.

BiLSTM and the **sequential** output with **attention** mechanism **(BiLSTM_SA)**: correspondingly, two BiLSTM layers are used to substitute the LSTM layers in LSTM_SA.

Transformer: three encoding and three decoding transformer layers, followed by a fully connected layer to get long sequential outputs in parallel (Vaswani et al., 2017; Zhou et al., 2021).

Transformer with an **LSTM** output layer **(Transformer_LSTM)**: as shown in Fig. 3, the forward dense layer in Transformer_F is changed to an LSTM layer, to infer the future trajectory recurrently.

In addition to comparing the results from different model architectures, the input features of each model will be respectively set as 3D position sequences, 3D differentiated velocity sequences, and 3D differentiated acceleration sequences.

(c) Implementation details

We trained the proposed model and the baselines by utilising the AdamW optimizer with a Mean Square Error (MSE) loss function and a batch size of 128 (Loshchilov and Hutter, 2018). Moreover, we applied an exponential learning rate decay schedule with an initial rate of 0.001 and decaying every 6000 steps with a base of 0.9, alongside a fixed weight decay with 0.001. All the experiments were conducted on the platform equipped with an AMD Ryzen 7950x CPU and a single NVIDIA 4090 GPU.

(d) Evaluation metrics

In this work, two kinds of measurements are considered to evaluate the prediction errors of the proposed model and compare its performance with baselines. The first type covers errors in 3 dimensions together based on Euclidean distance (L2-norm), including the Average Displacement Error (ADE) and Final Displacement Error (FDE). They provide a more straightforward indication of how closely the predicted trajectory aligns with the ground truth in 3D space, and have been widely used in trajectory prediction (Alahi et al., 2016; Mohamed et al., 2022; Salzmann et al., 2020). Referring to the notations defined in Section 3.2, they can be expressed as follows:

ADE: the average Euclidean distance between the predicted 3D positions $\widehat{\boldsymbol{x}}$ and the corresponding ground truth $\boldsymbol{x}$ over the entire prediction horizon, expressed as:

$$ADE = \frac{1}{N \times T_{pred}} \sum_{i=1}^{N} \sum_{t=1}^{T_{pred}} \left\| \widehat{\boldsymbol{x}}_t^i - \boldsymbol{x}_t^i \right\|_2 \tag{16}$$

FDE: the Euclidean distance between the predicted 3D position $\widehat{\boldsymbol{x}}_{T_{pred}}$ and the corresponding ground truth $\boldsymbol{x}_{T_{pred}}$ at the last timestamp T_{pred}.

$$FDE = \frac{1}{N} \sum_{i=1}^{N} \left\| \widehat{\boldsymbol{x}}_{T_{pred}}^i - \boldsymbol{x}_{T_{pred}}^i \right\|_2 \tag{17}$$

Another category of evaluation metrics, including the Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Median Relative Error (MedRE, i.e., the mean absolute error without percentage scaling), three criteria used in previous works (Jia et al., 2022; Yang et al., 2020), is employed to evaluate the proposed method and baselines on the performance variance in each axis $j, j \in \{1, 2, 3\}$ in 3D space at the last timestamp T_{pred}. Following the notations stated above, the three metrics are defined as:

$$MAE_j = \frac{1}{N} \sum_{i=1}^{N} \left| \widehat{\boldsymbol{x}}_{T_{pred}}^{i,j} - \boldsymbol{x}_{T_{pred}}^{i,j} \right| \tag{18}$$

$$RMSE_j = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(\widehat{\boldsymbol{x}}_{T_{pred}}^{i,j} - \boldsymbol{x}_{T_{pred}}^{i,j} \right)^2} \tag{19}$$

$$MedRE_j = Med\left(\left|\frac{\hat{\boldsymbol{x}}^{i,j}_{T_{pred}} - \boldsymbol{x}^{i,j}_{T_{pred}}}{\boldsymbol{x}^{i,j}_{T_{pred}}}\right|\right), i \in [1, N] \tag{20}$$

Since some trajectory points have zero velocity along with one of three axes, leading to zero relative movement to the trajectory segment origin as described in Section 3.1 (b), we decided to use median relative errors instead of the mean values. If the relative error is calculated globally, the results will be affected by the absolute coordinates.

5.2. Results from differential sequences

Based on ADE and FDE, we first conducted experiments to quantitatively evaluate the impact of introducing differential sequences into trajectory prediction on the prediction performance based on MavicAir2 and Payload datasets and presented the results in Table 2. It is evident that all velocity-oriented trajectory prediction models significantly improve accuracy in the two datasets, leveraging the first-order differential flight trajectory positions. However, the introduction of second-order differentials shows a moderate drop in ADEs for some models but a considerable increase in FDEs across most cases, indicating the error increases rapidly over longer prediction periods. Additionally, models utilising dynamically decoded LSTM and attention mechanisms tend to achieve slightly more precise trajectory predictions compared to those employing fully connected layers for predicting the flattened 3D trajectories (models with 'F' suffix vs. 'SA' suffix). Moreover, models with Transformer-based encoders outperform those with LSTM-based encoders, showcasing their superior ability in handling trajectory prediction tasks.

To explore the performance variation of different input features along with the increase of the prediction period, the displacement errors (DEs) of each model at each timestamp are compared in Fig. 5. It can be found that acceleration-based predictions achieve the best results in the initial period while the corresponding DEs increase dramatically to be significantly higher than the counterparts of the other two models at the later stage. Far more significant errors, by contrast, are observed in the initial stage for position-based predictions. In comparison, predicting the future velocities and then inferring the trajectory can obtain a more balanced result that seems to combine the strengths of both acceleration and position-based predictions, achieving accurate prediction in the early stage while mitigating the drastic growth of DEs.

5.3. Results from LIMM model

Given the superior prediction accuracy of the Transformer_LSTM model as shown in Table 2, we choose it as the upstream predictor to generate pseudo measurements in our subsequent experiment. This experiment aims to evaluate the effectiveness of the proposed LIMM model across 15 entirely new trajectory datasets, in addition to the two identically distributed test sets. To facilitate comparison and demonstrate the benefits of feature fusion, we also evaluated a single velocity-based model across all. Additionally, to assess the performance of the LIMM model in integrating multiple features, we have prepared a baseline model that takes positions, differential velocities, and accelerations as inputs (concatenating them along the feature dimension). All three models under comparison are designed to predict future velocity series and subsequently derive the trajectory, building upon the findings from our previous experiment indicating that the velocity-based model is superior.

Based on the quantitative comparisons presented in Table 3, it is evident that the LIMM model demonstrates the best performance overall, achieving significant improvements on all shifted datasets except for Spark, where the velocity-based model performs better. On the identically distributed dataset Payload, the LIMM model shows slightly inferior performance compared to the mixed features-

Table 2
The quantitative results of all models with different input features in the two identically distributed test sets (T_{obs} = 40 s and T_{pred} = 20 s).

Dataset	Model	Inputting position		Inputting velocity		Inputting acceleration	
		ADE (m)	FDE (m)	ADE (m)	FDE (m)	ADE (m)	FDE (m)
MavicAir2	LSTM_F	24.91	42.20	8.61	13.11	20.23	43.79
	GRU_F	27.10	46.82	10.81	16.51	20.36	43.85
	BiLSTM_F	22.63	39.17	7.91	12.19	16.07	34.55
	ConvLSTM1D_F	21.04	35.34	7.78	12.07	21.97	47.67
	LSTM_SA	19.37	32.50	8.23	12.67	20.93	45.18
	GRU_SA	25.94	45.07	12.28	19.62	26.25	57.11
	BiLSTM_SA	18.33	29.97	7.49	11.58	19.26	41.60
	Transformer	14.79	22.87	7.08	11.35	20.72	45.89
	Transformer_LSTM	**13.25**	**21.09**	**6.97**	**11.23**	**15.62**	**33.31**
Payload	LSTM_F	7.93	11.53	4.61	7.83	7.69	15.50
	GRU_F	7.52	11.61	4.61	7.66	7.63	15.79
	BiLSTM_F	6.96	10.44	4.07	7.06	5.92	11.98
	ConvLSTM1D_F	7.79	11.79	4.98	8.44	9.03	18.68
	LSTM_SA	7.30	11.77	4.40	7.59	7.23	14.58
	GRU_SA	7.21	11.70	4.32	7.40	7.54	15.36
	BiLSTM_SA	7.24	11.05	3.92	6.62	6.32	12.63
	Transformer	6.31	9.25	3.59	6.27	6.33	13.34
	Transformer_LSTM	**5.69**	**8.79**	**3.32**	**5.99**	**5.54**	**11.51**

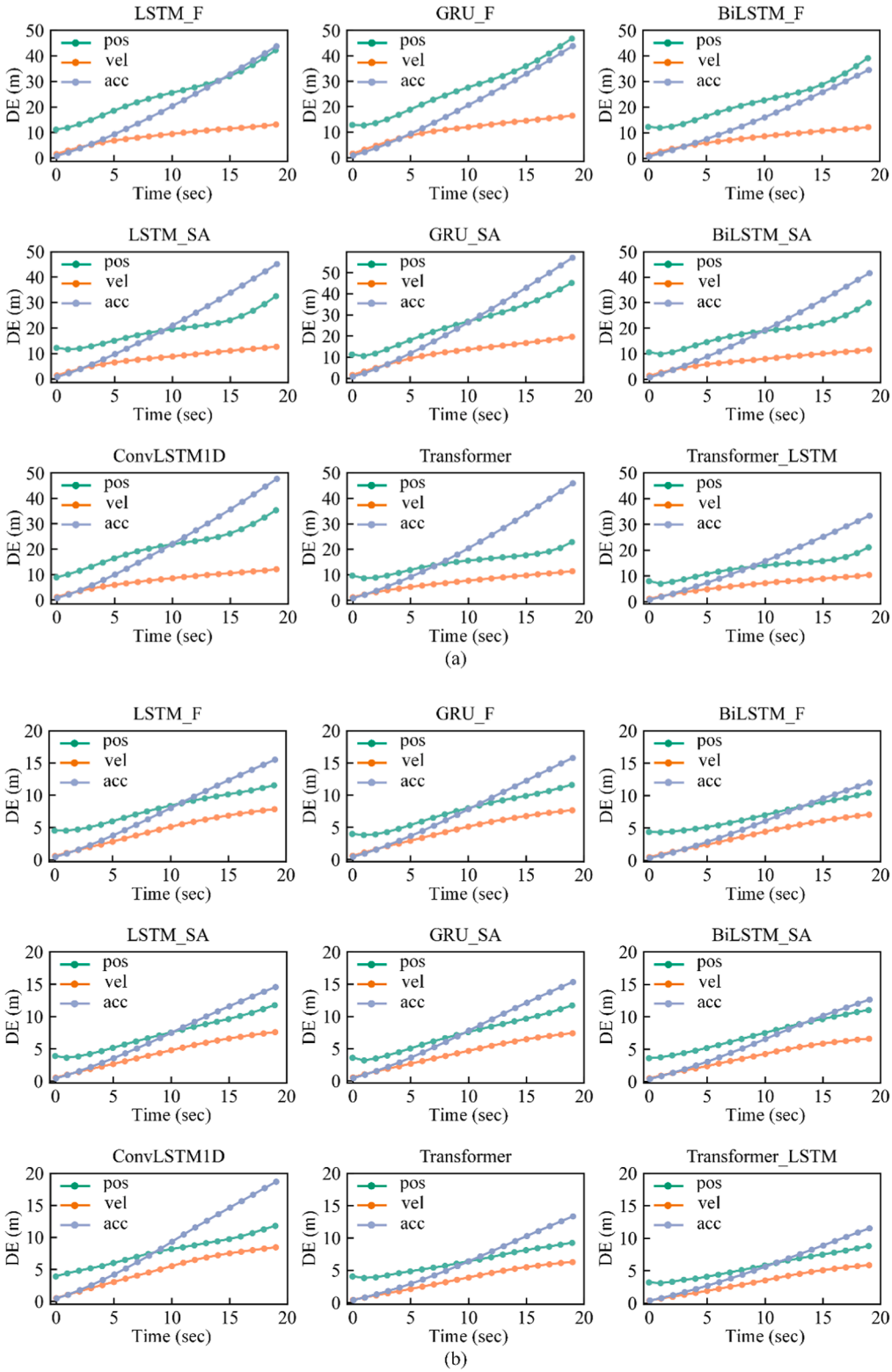

Fig. 5. The displacement error of each model along with increasing prediction time horizon on (a) MavicAir2 and (b) Payload dataset.

Table 3
The comparison of the LIMM model with baselines in both identically distributed and shifted datasets (T_{obs} = 40 s and T_{pred} = 20 s).

Type	Dataset	Only velocity		Mixed features		LIMM	
		ADE (m)	FDE (m)	ADE (m)	FDE (m)	ADE (m)	FDE (m)
Identical	MavicAir2	6.97	11.23	7.54	12.27	**6.51**	**10.30**
	Payload	3.32	5.99	**3.26**	**5.63**	3.29	5.84
Shifted	Inspire	69.59	135.98	69.49	134.43	**63.12**	**125.68**
	Inspire2	35.42	74.74	36.76	74.74	**29.95**	**64.06**
	Matrice600Pro	67.03	133.36	68.31	134.65	**60.84**	**126.58**
	Mavic2Enterprise	32.12	70.06	36.51	79.63	**24.37**	**55.21**
	Mavic2Pro	34.26	72.70	36.98	74.37	**30.80**	**66.16**
	Mavic2Zoom	60.03	114.29	56.64	109.66	**50.48**	**97.54**
	MavicAir	74.60	153.34	70.87	147.77	**57.47**	**120.32**
	MavicPro	29.71	58.73	30.48	62.44	**24.43**	**50.14**
	Phantom3	38.78	80.94	40.05	81.66	**32.41**	**68.95**
	Phantom4Pro	41.31	82.78	42.68	86.06	**34.56**	**69.71**
	Phantom4ProV2	55.16	114.08	51.91	111.42	**44.13**	**92.35**
	Spark	**6.75**	**13.26**	7.21	13.44	7.09	14.85
	Farm	57.90	116.50	54.74	110.01	**52.69**	**109.10**
	UMCD	12.78	24.59	13.91	26.24	**8.19**	**16.87**
	ZUMAVD	6.38	12.09	8.25	15.32	**5.18**	**10.32**

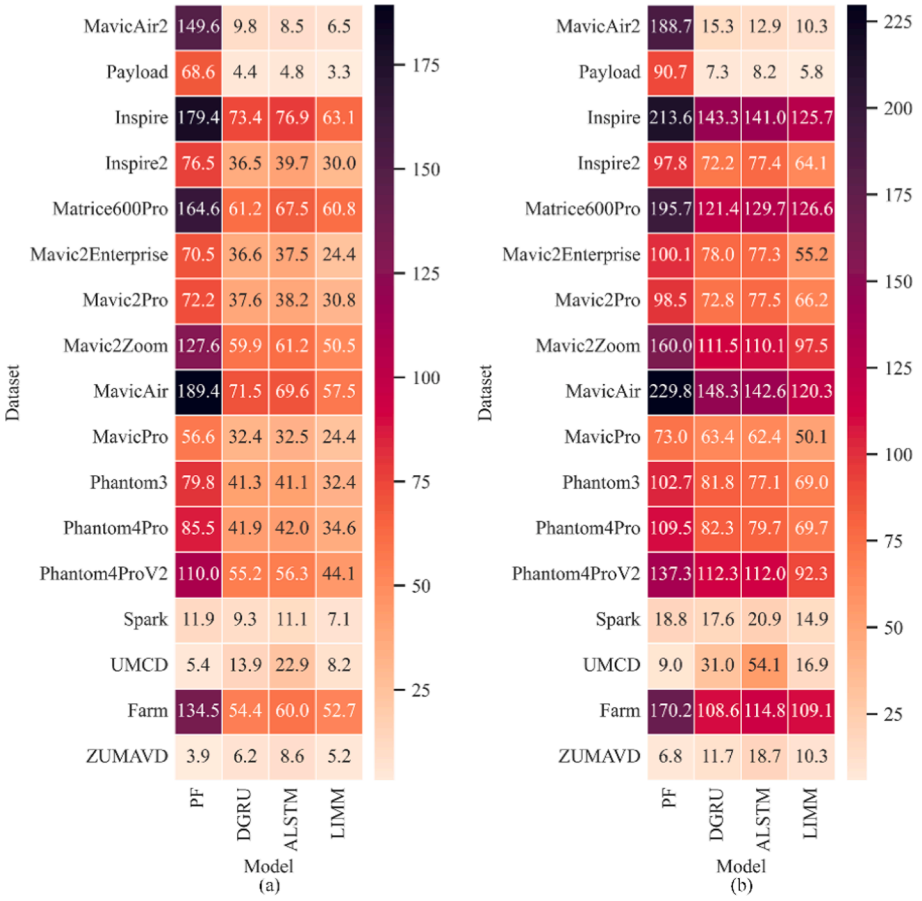

Fig. 6. The comparison of the proposed LIMM model with baselines on (a) ADE (m) and (b) FDE (m) metrics.

based model. These findings confirm the effectiveness of our proposed method, which adaptively integrates multiple predictions to forecast future trajectories in unknown scenarios, making it highly suitable for real-world applications.

Moreover, the results suggest that employing mixed features does not guarantee higher prediction accuracy than models that only utilise velocities as inputs. Additionally, we observed that accurately predicting shifted trajectories faces great challenges. For instance, the ADE and FDE on the MavicPro dataset are significantly larger than those on the MavicAir2 dataset, despite both datasets having similar flight speeds.

5.4. Comparison with the state-of-the-art

In this subsection, we compare our LIMM model with advanced drone trajectory prediction models, including both machine learning-based and kinematic modelling-based approaches (PF), under the same experimental settings as described in Section 5.3. The machine learning-based models include: 1) DGRU, designed for multi-rotor trajectory prediction by leveraging dual intervals of velocity series (Yang et al., 2020); and 2) Attention-LSTM (ALSTM), which utilises a dot-attention mechanism to weight sequential hidden states in 4D flight trajectory prediction (Jia et al., 2022). In addition to the metrics ADE and FDE, we also include MAE, RMSE, and MedRE, defined in Section 5.1, along each axis to evaluate and compare the models from multiple perspectives. It is important to note that ADE covers the entire prediction horizon, while the remaining four metrics focus on the performance at the final timestamp.

Given the comparison of 4 models across 5 metrics for 17 datasets, we first present the ADE and FDE results in Fig. 6. The results indicate that the proposed LIMM model outperforms the three baseline models, except for two low-speed trajectory datasets (UMCD and ZUMAVD), where the PF provides more accurate predictions compared to the machine learning-based models. For the two identically distributed datasets, the errors of trajectories predicted by learning-based models are significantly lower than those of the PF. However, for the remaining 15 datasets, the performance of all models drops considerably due to distribution shifts, which introduce unseen data patterns that challenge the pre-trained models.

Next, Fig. 7 and Fig. 8 compare these models based on MAE and RMSE metrics along each axis, respectively. Similarly, the PF model proves superior to machine learning-based models in predicting trajectories for slowly flying drones (UMCD and ZUMAVD), as evidenced by both MAE and RMSE comparisons. In terms of longitude and altitude predictions of the two datasets, DGRU achieves the second-best performance. Nevertheless, the proposed LIMM outperforms all baseline models in the majority of both identically and shifted-distributed datasets, particularly for those involving faster-moving drones, as described in Table 1 and Fig. 1. The only exceptions are its inferior performance to ALSTM and DGRU in the Matrice600Pro dataset along the longitude and latitude, respectively,

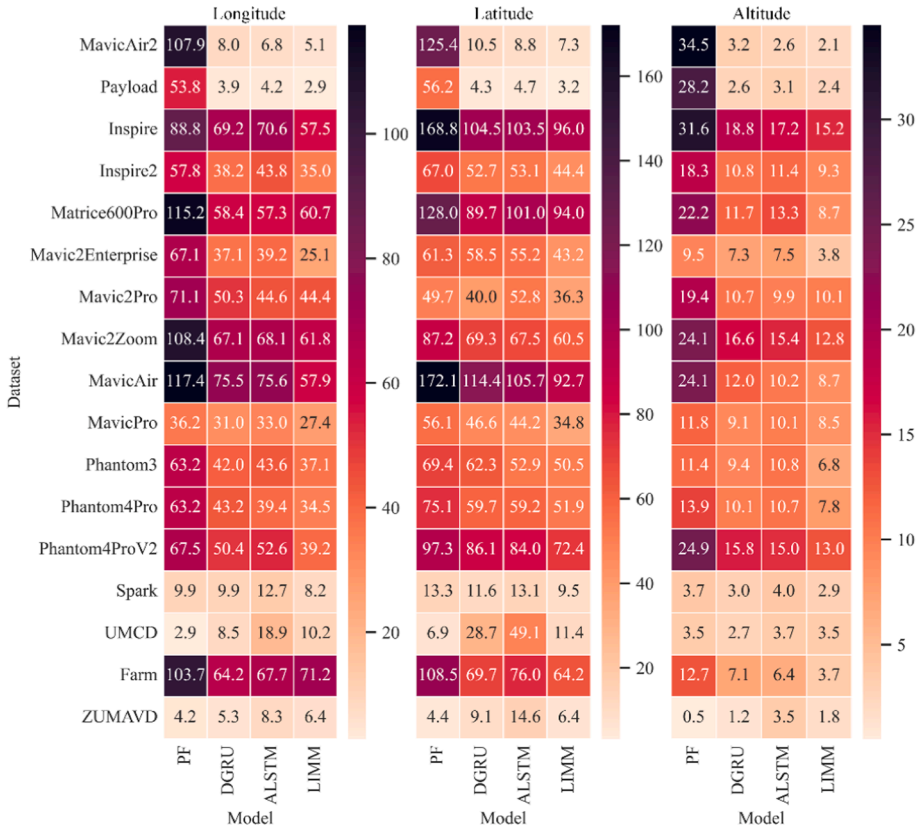

Fig. 7. The comparison of the proposed LIMM model with baselines on MAE (m) in three axes.

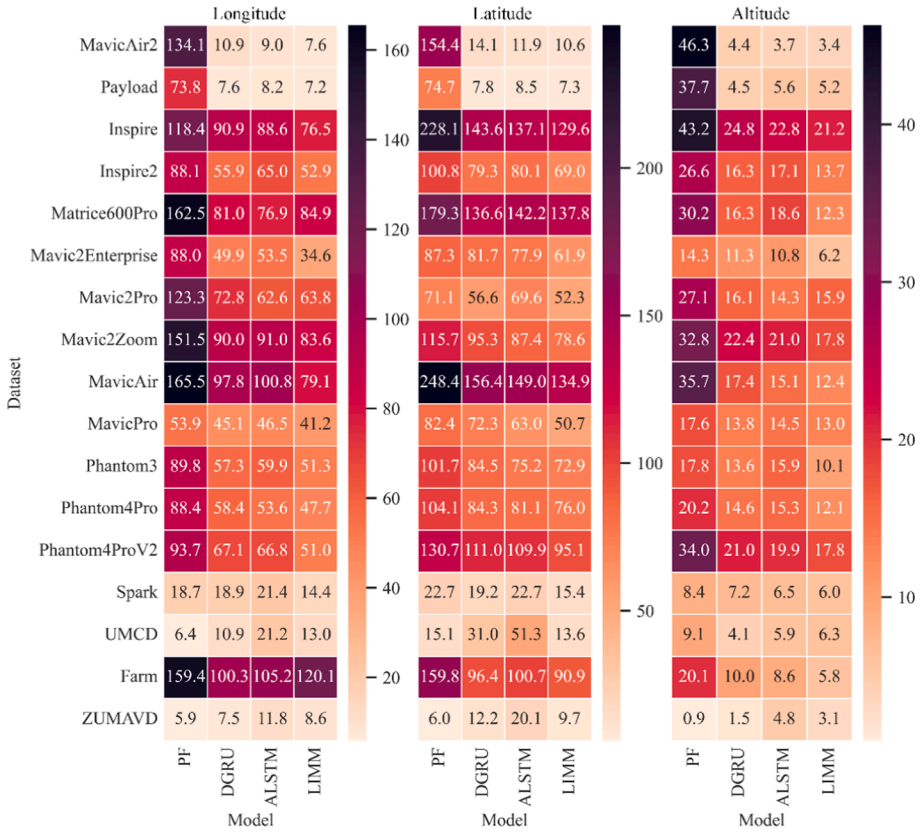

Fig. 8. The comparison of the proposed LIMM model with baselines on RMSE (m) in three axes.

and its marginally worse altitude prediction compared to ALSTM in the Mavic2Pro dataset.

Again, the comparison results for MedRE are presented in Fig. 9, providing further evidence that the proposed LIMM outperforms the three baseline models across most datasets. However, it is outperformed by PF (in all three axes) and DGRU (in longitude) in the two low-speed datasets. Additionally, MedRE in altitude is notably higher than in the lateral axes, particularly in the ZUMAVD dataset, which contains many segments with near-zero vertical velocity, as illustrated in Fig. 1. Consequently, we excluded the altitude MedRE results for the ZUMAVD dataset, which exceed 90 for all models, in Fig. 9.

These results reveal that accurate trajectory prediction for non-cooperative drones over extended horizons remains a significant challenge for both machine learning and kinematic modelling-based methods, given that only historical trajectory data is assumed to be available from the management platform.

5.5. *The transition between multiple predictions*

To further validate the effectiveness of the proposed LIMM, we extracted the learned transition probability matrix from the trained model and illustrated the results at 5 different timestamps in Fig. 10. It is observed that the probability of transition to velocity-oriented upstream prediction is consistently higher than the other two modes, especially in long-term predictions. This suggests that the velocity-based model plays a prominent role in the combination process within the IMM framework. In contrast, the acceleration-based model shows a higher contribution in the initial stages, as depicted in the first subplot of Fig. 10. This observation aligns with our expectations, given that models predicting future velocities demonstrate significantly better performance over longer prediction horizons, albeit slightly inferior to acceleration-based models in the initial stages, as shown in Table 2 and Fig. 5.

To validate whether the IMM model effectively achieves the transition function (i.e., initially relying more on the prediction from the acceleration-based model and gradually incorporating more weight from the velocity-based model), we compared the sequential predictions from the acceleration, velocity, position, and LIMM-oriented models using the MavicAir2 dataset. The results for the first 5 s are presented in Table 4, as predictions for the remaining time horizons are predominantly determined by the velocity-based model.

The results indicate that the LIMM model provides a balanced prediction in the early stage, outperforming both the velocity and acceleration-based models. However, it does not achieve the ideal weight configuration, where the LIMM model would rely solely on acceleration-based predictions during the first two seconds. Nevertheless, for the full trajectory prediction sequence, the model learns the optimal transition to minimize the total loss across the entire horizon. The transition probabilities shown in Fig. 10 are continuous rather than binary, which accounts for the observed gap between the LIMM model and the acceleration-based prediction in the early stage.

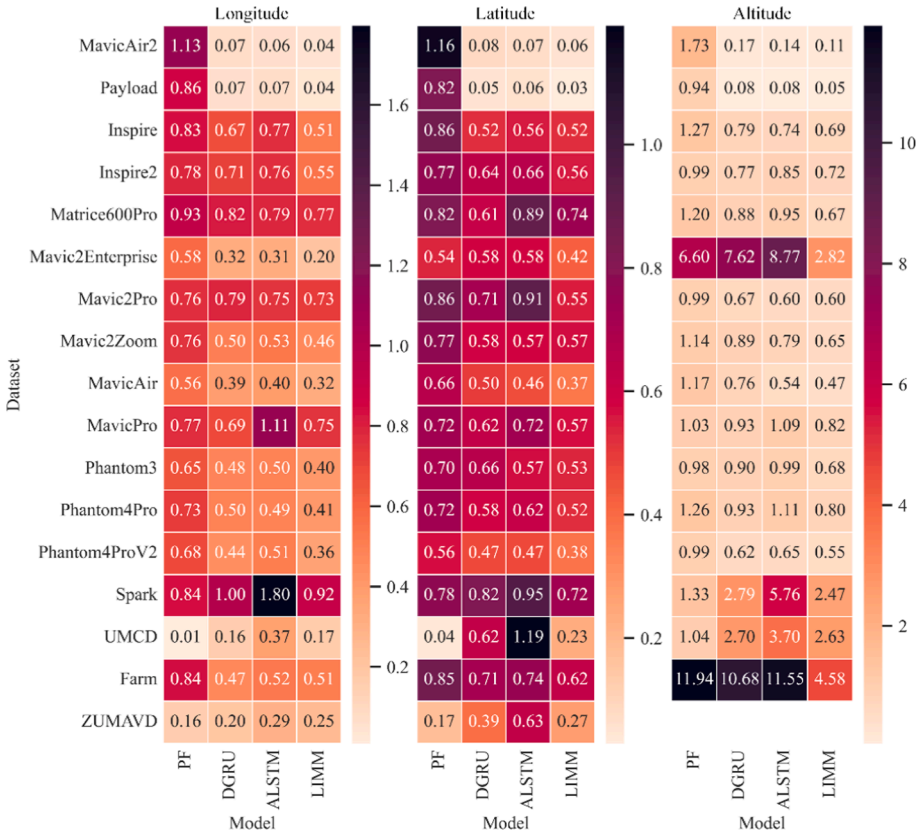

Fig. 9. The comparison of the proposed LIMM model with baselines on MedRE (ratio) in each axis.

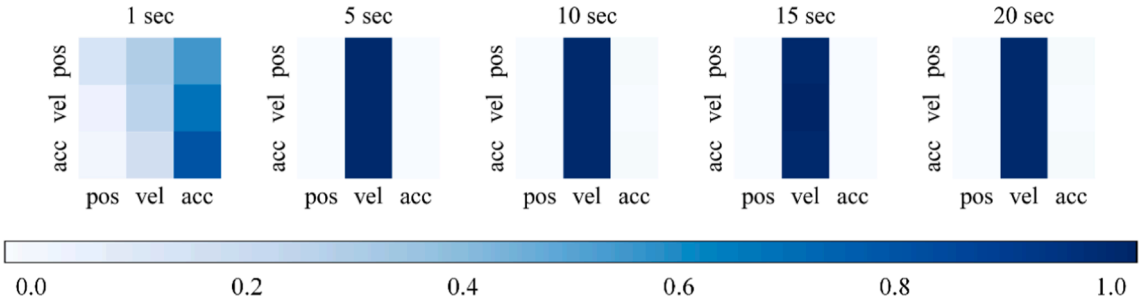

Fig. 10. The learned transition probability matrices at different prediction timestamps.

Table 4
The comparison of the 4 models in the early stage based on DE (m) in MavicAir2 dataset.

Model	1 s	2 s	3 s	4 s	5 s
Position	7.95	6.95	7.66	8.59	9.65
Velocity	1.35	2.41	3.24	3.85	4.44
Acceleration	**0.74**	**1.84**	3.09	4.43	5.85
LIMM	1.07	2.03	**2.88**	**3.60**	**4.23**

To further investigate how the interplay between the models is controlled by the transition probability matrix in the LIMM model under different scenarios, we explicitly compare the prediction errors in samples with sharp turns and those with straight flight paths. Each group consists of 10 samples manually selected to ensure that the samples represent either rapid turns or straight flight segments. The displacement errors for the first 5 s are presented in Table 5.

It is observed that the mean prediction error, represented by the displacement error of samples with sharp turns, is significantly larger than that of the samples with straight flight paths and also exceeds the global average value presented in Table 4. In both cases,

Table 5
The comparison of the 4 models in the early stage for different cases based on DE (m).

Group	Model	1 s	2 s	3 s	4 s	5 s
Turn	Position	6.68	7.03	7.75	9.30	11.18
	Velocity	1.70	3.65	5.50	7.34	9.09
	Acceleration	**1.18**	3.38	6.07	8.89	11.93
	LIMM	1.34	**3.09**	**4.56**	**5.80**	**6.84**
Straight	Position	7.36	5.35	7.09	9.59	11.30
	Velocity	0.75	1.34	1.51	2.26	3.03
	Acceleration	**0.35**	**0.89**	1.51	2.26	3.03
	LIMM	0.74	1.13	**1.24**	**1.36**	**1.64**

the LIMM model provides a balanced prediction in the early stage by combining the three base predictions according to the learned transition probabilities. However, the model encounters challenges when predicting rapid turns.

5.6. Analysing the predicted trajectories

In addition to demonstrating the superior performance of the proposed model through quantitative evaluations in the above subsections, we aim to gain deeper insights into the qualitative performance of our model in scenarios involving manoeuvring drones. In Fig. 11, for several test samples from the MavicAir2 dataset, the predicted trajectories from the LIMM model and the corresponding three types of upstream estimates are visualised. It's noted that only the last 10-second historical trajectory is presented for visualization, while the entire 40-second trajectory is used as input in our model. The plots illustrate that our model effectively integrates multiple upstream predictions even when they generate divergent future trajectories, particularly when the target sharply changes heading.

Importantly, the results from the LIMM are not simply averaged values of the three upstream predictions. The examples in Fig. 11 demonstrate that our model successfully performs adaptive fusion among the three resources, such as disregarding the position-based model's prediction in the upper left example. In addition, the drone in the middle-left example climbed in the initial stage and was predicted accurately, as evidenced by the multiple dots clustered in the area connecting the historical trajectory and the predictions. We also note significant initial shifts in trajectory predictions from the position-based model, while the acceleration-based model shows substantial deviations in long-term predictions. These observations align with the quantitative results shown in Fig. 5.

Additionally, we present some failure cases where our model does not achieve more accurate results than the best upstream prediction (typically velocity-oriented), illustrated in the last row of Fig. 11. A significant factor contributing to these outcomes is that the LIMM model integrates all three upstream predictions to some extent. However, trajectories predicted based on acceleration can deviate greatly from the ground truth, as seen in the upper right example in Fig. 12.

Finally, we illustrate some challenging examples in Fig. 13 to shed light on the potential limitations of our model and even other machine learning-based approaches. Our IMM model can struggle when there is significant divergence among the three upstream base models, often leaning towards the velocity-based result due to its higher average accuracy. In the cases depicted in the first row of Fig. 13, for instance, the predictions from velocity and acceleration-based models are nearly opposite in terms of flight direction, with the fact that the acceleration-based model performs better than the velocity-based one. Interestingly, while the trained models may not accurately predict the exact trajectory in some instances, they do capture the general movement tendencies of the target, as seen in the example in the lower left. Moreover, unexpected manoeuvres pose a considerable challenge for accurate prediction, as illustrated in the lower right example where all models initially predicted the drone would continue turning left, whereas it made a slight manoeuvre to the right.

6. Discussion

6.1. Differential trajectory

The primary aim of utilising the velocity and acceleration series for drone trajectory prediction, derived from the position series, is to explicitly capture the divergence in trajectories predicted from different kinematic information (position, velocity, and acceleration). Some may question whether the velocity and acceleration series offer more comprehensive information compared to the position series, which serves as the foundation for the former two. However, akin to ARIMA models that employ differencing mechanisms to transform non-stationary time series into stationary ones with a relatively constant mean, the prediction of future trajectories theoretically benefits from differencing the original series. Generally, the velocity and acceleration series fluctuate within a much narrower scope due to mechanical constraints. This restriction allows models trained on differential data to forecast future positions more accurately, aligning with kinematic characteristics. Additionally, there have been intriguing explorations into using differential trajectories as prediction targets to reduce embedding space and mitigate the influence of outliers (Guo et al., 2024).

However, it is not implied that the differencing mechanism universally enhances the accuracy of time series prediction, especially for data lacking pronounced seasonal trends. Differencing data can indeed improve short-term forecasting accuracy but tends to perform poorly over longer horizons (Fildes et al., 1998; Makridakis and Hibon, 1997). Unfortunately, UAV trajectory data typically lacks discernible seasonal trends. Moreover, differential sequences exhibit higher frequency fluctuations compared to the original data,

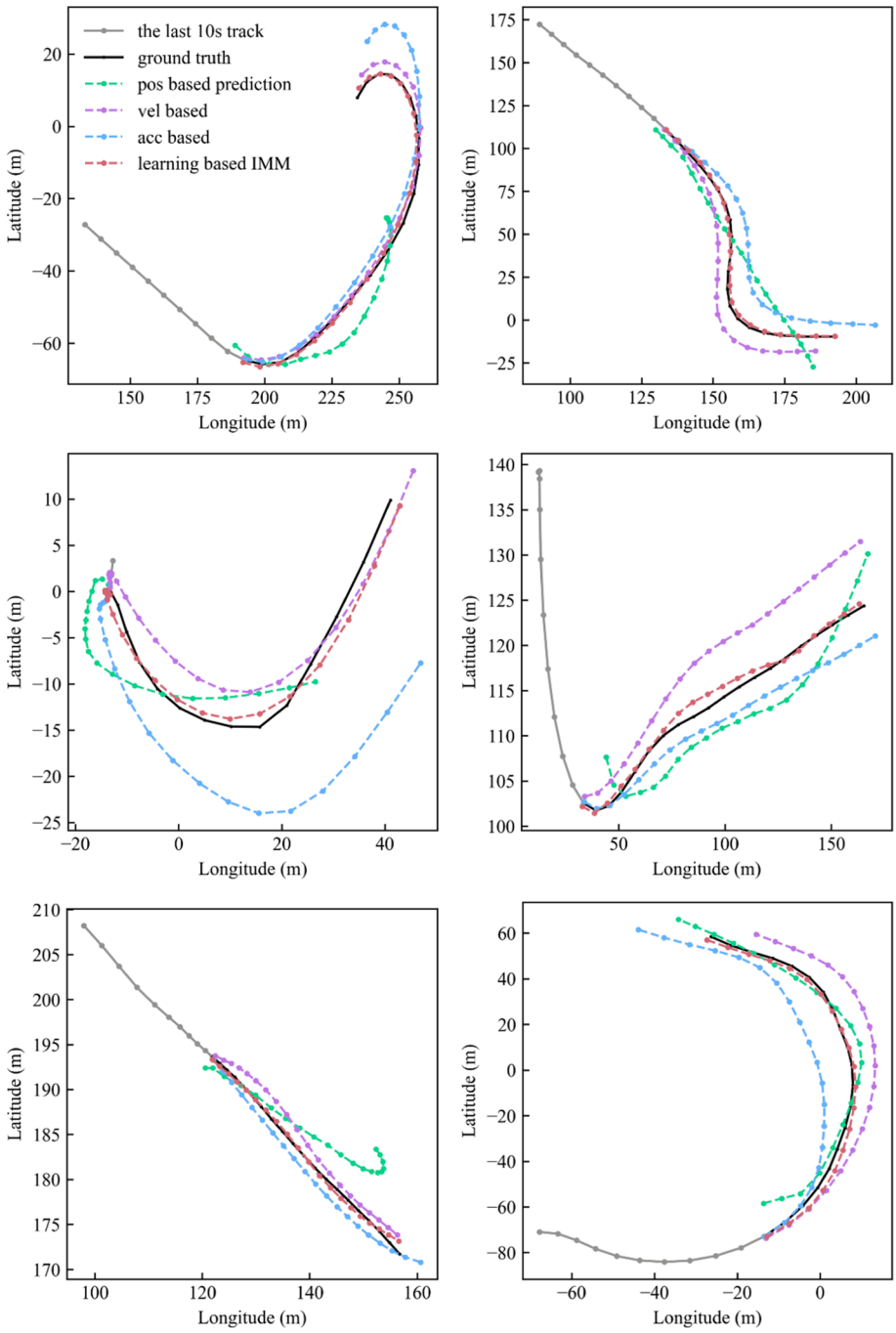

Fig. 11. The examples in which the LIMM model is superior to the three upstream models.

which intuitively appears less smooth. In efforts to minimize overall prediction loss, such as MSE, models trained on differential series often tend to forecast the mean value (the most probable state), for instance, when a quadcopter is manoeuvring unpredictably. These prediction models are particularly sensitive to shifts in differential series, such as abrupt changes in direction or hovering, leading to cumulative increases in prediction errors, as illustrated in the examples shown in Figs. 11 - 13. This sensitivity explains why prediction errors for models using acceleration as input dramatically increase over longer terms, as observed in Table 2 and Fig. 5.

To assess whether finer sampling can mitigate the drift from the ground-truth acceleration and velocity sequences, thereby improving prediction accuracy, we conducted an experiment to examine the performance gap between coarse and relatively refined discretisation methods based on the MavicAir2 dataset and the Transformer-LSTM model. Specifically, we increased the temporal resolution to 0.5 s, resulting in input and output sequence lengths of 80 and 40, respectively, when predicting a 20-second trajectory.

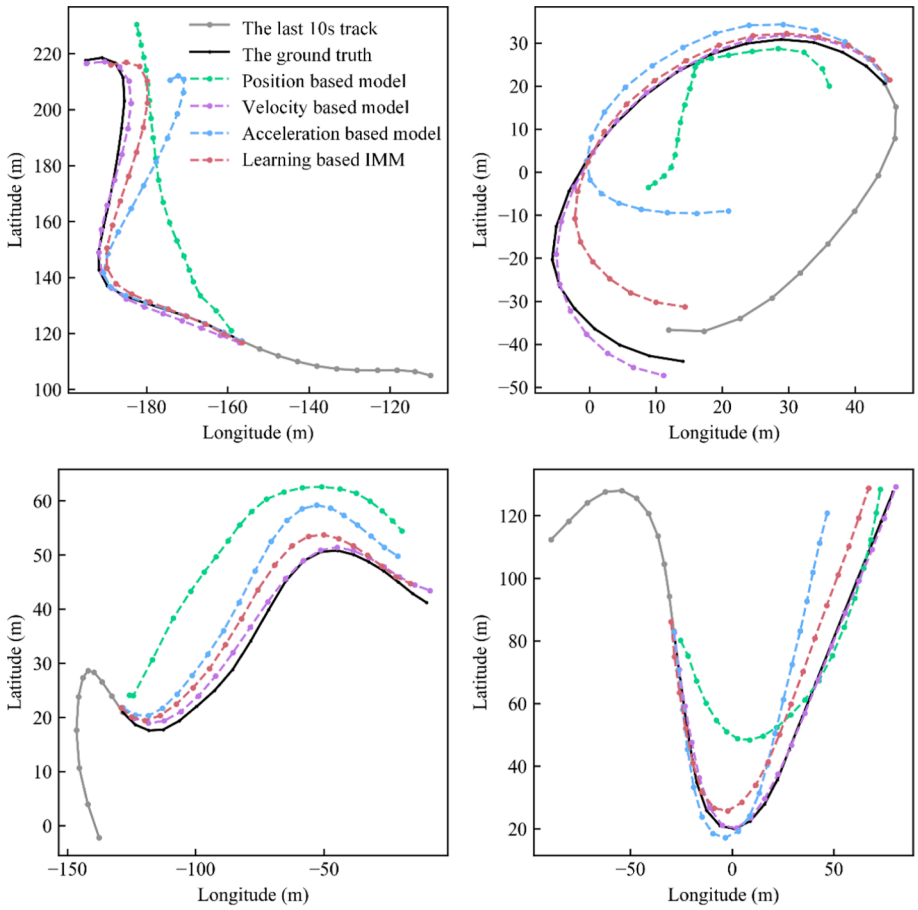

Fig. 12. The examples in which the LIMM model is inferior to the velocity-based model.

Additionally, we sampled trajectory points at 2-second intervals to investigate the impact of sampling resolution on model performance further. The results, presented in Table 6, include comparisons of the mean absolute error (MAE) in each dimension and the displacement errors at the first two seconds and the final timestamp (DE_1, DE_2, and DE_{20}). It should be noted that only the MavicAir2 dataset was used as the training pool in this experiment, rather than a mix of MavicAir2 and Payload data, due to limitations in trajectory frequency. As a result, the errors from the acceleration-based model in this experiment differ from those reported in Section 5.

In general, finer velocity or acceleration discretization (0.5 s & 1.0 s vs. 2.0 s) improves early-stage trajectory prediction but negatively impacts longer-horizon performance in the acceleration-based model, as indicated by the DE_{20} values in Table 6. However, increasing the sampling resolution from 1.0 s to 0.5 s does not improve prediction accuracy; instead, it degrades performance and increases training complexity, particularly for the acceleration-based model. A finer resolution imposes stricter constraints on machine learning models to minimize overall loss, making training more challenging (e.g., 40-step vs. 20-step sequence prediction). Conversely, reducing the resolution from 1.0 s to 2.0 s obscures manoeuvring details due to averaging, leading to reduced accuracy in the initial stage (2.0 s). Thus, a balance between learning complexity and resolution should be considered.

6.2. *The ratio of FDE to ADE*

Table 3 reveals an approximately linear relationship between ADE and FDE, following the relation FDE ≈ 2.0·ADE. To determine whether this ratio remains consistent across different prediction horizons and can serve as a diagnostic tool for identifying deviations in model performance, we extend the prediction horizon to 30 and 40 s. Fig. 14 illustrates the FDE-to-ADE ratios across datasets from Table 3, with each colour representing a different dataset (dataset labels are omitted as the focus is on overall ratio variation). The two leftmost small circles correspond to the identical MavicAir2 and Payload datasets, exhibiting lower ratios (approximately 1.6) compared to the other 15 distribution-shifted datasets.

The results in Fig. 14 illustrate the varying FDE-to-ADE ratio across different prediction horizons, which remains approximately between 1.8 and 2.4 for the 15 newly introduced datasets. This suggests that the ratio, to some extent, can serve as a diagnostic index to validate the consistency of predicted trajectories through cross-validation between ADE and FDE. Notably, an increase in ADE does not

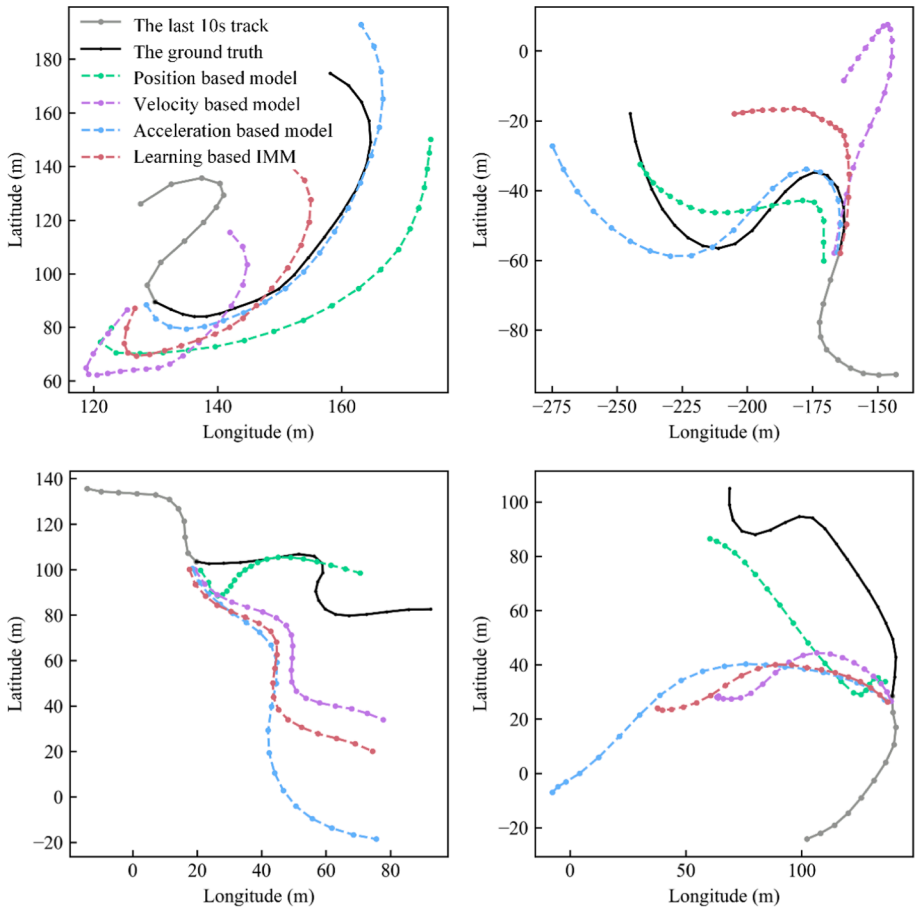

Fig. 13. The examples that all models failed to predict accurately.

Table 6
The impact of sampling resolution in discretization on the accuracy of 20-second trajectory prediction for each axis in MavicAir2 dataset.

Feature	Resolution (s)	LongitudeMAE (m)	LatitudeMAE (m)	AltitudeMAE (m)	DE_1 (m)	DE_2 (m)	DE_{20} (m)
Velocity	0.5	3.52	5.35	1.24	0.86	1.56	7.27
	1.0	3.19	4.43	1.18	0.79	1.38	6.27
	2.0	3.54	5.18	1.29	–	2.54	7.17
Acceleration	0.5	18.06	23.95	9.24	0.54	1.44	35.71
	1.0	10.44	12.43	9.10	0.53	1.22	21.58
	2.0	8.67	13.36	7.28	–	1.34	20.17

necessarily lead to a strictly proportional rise in FDE. For instance, in the Matrice600Pro dataset (orange circle in Fig. 14), the FDE-to-ADE ratio decreases from 2.1 to approximately 1.8 as the prediction horizon extends to 30 and 40 s. However, due to the substantial increase in ADE over longer horizons, this slight reduction in the ratio does not undermine the overall positive correlation between ADE and FDE, despite the lack of strict proportionality. Moreover, the significant rise in average prediction error, as indicated by ADE, constrains the feasibility of using a 40-second historical input to predict future trajectory in more than 20 s, similar to the ratio adopted in a recent study (Nacar et al., 2025).

6.3. *Real-world applicability*

In this study, UAVs that do not voluntarily participate in flight activities, provide identification, or share their data with management authorities for position distribution are classified as non-cooperative targets. In such cases, accessing onboard sensors and cameras to determine their intentions is challenging. To address this issue, we propose leveraging existing platforms and potentially connected UAVs equipped with radars and cameras to capture the positional information of these non-cooperative targets. This data is then transmitted to a ground-based system, where trajectory prediction algorithms operate using cloud-based databases and flight

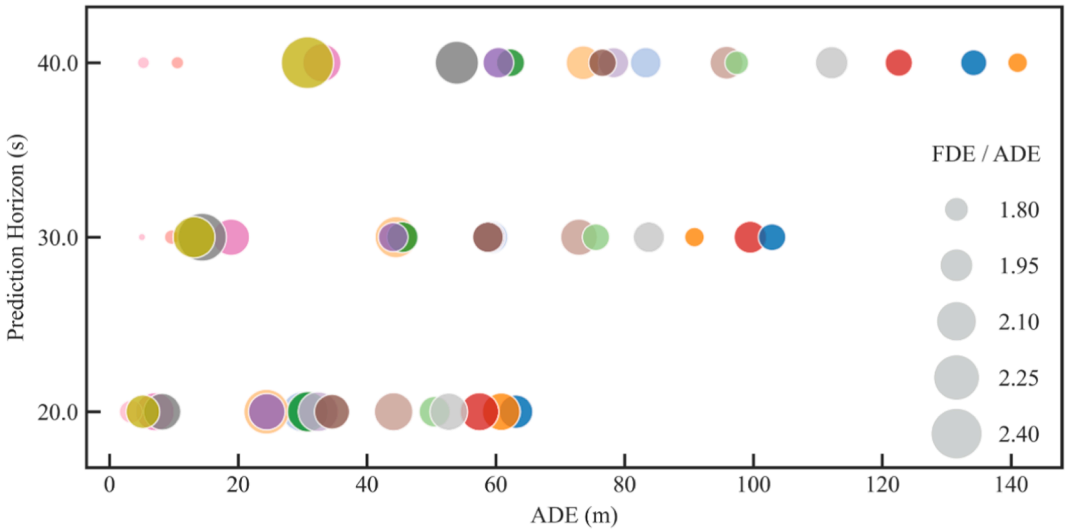

Fig. 14. The ratio of FDE to ADE, represented with the circle size, across different prediction horizons in all coloured datasets.

information management systems. This approach enhances the management of non-cooperative UAVs, aligning with the objectives of Conflict Detection and Resolution (CDR) and DAA systems (Ren et al., 2017; Wu et al., 2019).

In light of recent advancements in data storage and transfer technologies, we evaluated the average data load and inference time of the proposed method on our platform, emphasizing the critical importance of real-time performance in practical applications. The process includes reading and normalising input data, performing trajectory forecasting, and applying inverse normalisation to generate the final predicted trajectory. The results indicate that machine learning-based methods offer significant potential for real-time prediction, particularly when leveraging pre-training and efficient inference mechanisms. Specifically, the Transformer_LSTM model achieves an average inference time of less than 1.5e-4 s, regardless of whether the input consists of a single velocity series or multiple features. Even when incorporating additional upstream predictions, the proposed model requires approximately 8.0e-3 s per execution. In contrast, traditional filtering-based models, such as the PF and KF, exhibit an average prediction time exceeding 0.05 and 0.02 s, respectively, under the conditions outlined in Section 5.4. A recent study has explored the feasibility of integrating the pre-trained enhanced GRU model into a real-time drone trajectory prediction module by utilising velocity estimates (Nacar et al., 2025).

Furthermore, by integrating hardware acceleration and leveraging distributed computation techniques to parallelize operations across multiple processing units, inference speed and scalability can be significantly enhanced, allowing the system to simultaneously handle multiple drones. Additionally, we observed that preparatory tasks, such as model loading and encapsulation for on-device deployment, can be relatively time-intensive in real-world scenarios. However, these tasks are typically one-time operations or can be preemptively executed to mitigate their impact on real-time performance. Consequently, such a hybrid architecture, offloading computationally intensive tasks to cloud-based systems while utilising edge devices for lightweight inference, can ensure the system remains responsive, even in congested airspaces.

Another key development in the application of learning-based trajectory prediction models is their potential to incorporate multi-source data, such as environmental wind conditions, and integrate anti-interference techniques. However, the scope of this study is primarily focused on historical trajectory-driven prediction, evaluated across multiple real-world application datasets. Looking ahead, it is expected that the collection of more comprehensive and matched data, encompassing both trajectory and environmental factors, will be facilitated through contributions from open-source initiatives and the ongoing development of our experimental conditions. This will support the refinement and enhancement of the proposed trajectory prediction method, ensuring greater robustness and reliability for future applications.

7. Conclusion

This study proposes an integrated flight trajectory prediction method for multirotor drones, aimed at providing decision support for early alerting of abnormal behaviour and scientific assessment of traffic conditions at low altitudes. The method comprises two main parts: upstream parallel trajectory prediction leveraging Transformer architecture and a LIMM model to fuse multiple predictions for enhanced accuracy and reliability. In the upstream module, we additionally introduce differential trajectory prediction (velocity and acceleration) to capture diverse flight patterns. Experimental results indicate that velocity-based prediction models achieve superior accuracy compared to position and acceleration-oriented approaches. To integrate multiple predictions from the upstream module, we incorporate LSTM components into the traditional IMM framework. This creates a learning-based model that harnesses neural networks' predictive capabilities to replace conventional filters and dynamically learn optimal time-invariant transition probabilities between different predictions. Test results across multiple datasets validate the method's superiority, particularly on datasets not included in the training phase. Remarkably, this study utilises 17 trajectory datasets from industry and academia to support validation

and inform the relevant future research in this field.

Despite the potential demonstrated by the proposed LIMM approach for drone trajectory prediction, several avenues warrant further exploration in future studies. First, effectively utilising pre-trained machine learning models based on extensive historical data to predict the future trajectory of the interested target with varying flight patterns remains challenging, as highlighted in Section 5.3. Exploring techniques such as transfer learning and few-shot learning to mitigate performance degradation could significantly enhance applicability in real-world applications. Secondly, this study focuses on deterministic trajectory prediction under circumstances where only positional information is available. Addressing meteorological uncertainties, such as dynamic wind fields at low altitudes affected by terrains and buildings, through distributional prediction methods could yield more robust results. This improvement also represents a promising solution for mitigating the challenges posed by unpredictable manoeuvres in practical applications. Finally, this work concentrates on predicting future trajectories for individual drones. As urban air traffic evolves, modelling interactions between different drones, whether cooperative or unidentified, to better understand joint trajectory prediction scenarios will be crucial.

CRediT authorship contribution statement

Rong Tang: Writing – original draft, Visualization, Software, Resources, Methodology, Formal analysis, Data curation, Conceptualization. **Kam K.H. Ng:** Writing – review & editing, Supervision, Resources, Funding acquisition, Conceptualization. **Lishuai Li:** Writing – review & editing, Supervision, Resources. **Zhao Yang:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The research reported in this work was supported by General Research Fund (PolyU15201423), Research Institute for Sustainable Urban Development (BBG5), Research Centre on Unmanned Autonomous Systems, The Hong Kong Polytechnic University (CE1W), Research Funding Scheme for Supporting Intra-Faculty Interdisciplinary Projects (WZ8D), Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong SAR (RMK1), National Natural Science Foundation of China (52172328, 72301229), Fundamental Research Funds for the Central Universities (NS2024067) and National Key R&D Programme of China (2022YFB3104502).

References

Abbas, M.T., Jibran, M.A., Afaq, M., Song, W.-C., 2020. An adaptive approach to vehicle trajectory prediction using multimodel Kalman filter. Trans. Emerg. Telecommun. Technol. 31, e3734.

Abebe, M., Noh, Y., Kang, Y.-J., Seo, C., Kim, D., Seo, J., 2022. Ship trajectory planning for collision avoidance using hybrid ARIMA-LSTM models. Ocean Eng. 256, 111527. https://doi.org/10.1016/j.oceaneng.2022.111527.

Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S., 2016. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In: Presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp. 961–971.

Alzyout, M., Alsmirat, M., Al-Saleh, M.I., 2019. Automated ARIMA Model Construction for Dynamic Vehicle GPS Location Prediction. In: In: 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS). Presented at the 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS). IEEE, pp. 380–386. https://doi.org/10.1109/IOTSMS48152.2019.8939197.

Avola, D., Cinque, L., Foresti, G.L., Martinel, N., Pannone, D., Piciarelli, C., 2020. A UAV video dataset for mosaicking and change detection from low-altitude flights. IEEE Trans. Syst., Man, and Cybernetics: Syst. 50, 2139–2149. https://doi.org/10.1109/TSMC.2018.2804766.

Ayhan, S., Samet, H., 2016. Aircraft Trajectory Prediction Made Easy with Predictive Analytics, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16. Association for Computing Machinery, New York, NY, USA, pp. 21–30. Doi: 10.1145/2939672.2939694.

Bahdanau, D., Cho, K., Bengio, Y., 2016. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv:1409.0473 [cs, stat].

Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X., Tian, Q., 2023. Accurate medium-range global weather forecasting with 3D neural networks. Nature 619, 533–538. https://doi.org/10.1038/s41586-023-06185-3.

Brandenburger, A., Hoffmann, F., Charlish, A., 2023. Learning IMM Filter Parameters from Measurements using Gradient Descent, in: 2023 26th International Conference on Information Fusion (FUSION). Presented at the 2023 26th International Conference on Information Fusion (FUSION), pp. 1–7. Doi: 10.23919/FUSION52260.2023.10224219.

CFReDS Portal [WWW Document], 2024. URL https://cfreds.nist.gov/ (accessed 7.7.24).

Cho, K., van Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. Conference on Empirical Methods in Natural Language Processing (EMNLP 2014).

Chu, C.-S.-J., 1995. Time series segmentation: a sliding window approach. Inf. Sci. 85, 147–173. https://doi.org/10.1016/0020-0255(95)00021-G.

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

Cone, A.C., Wu, M.G., Lee, S.M., 2019. Detect-and-Avoid Alerting Performance for High-Speed UAS and Non-Cooperative Aircraft, in: AIAA Aviation 2019 Forum. American Institute of Aeronautics and Astronautics. Doi: 10.2514/6.2019-3313.

Corbetta, M., Banerjee, P., Okolo, W., Gorospe, G., Luchinsky, D.G., 2019. Real-time UAV Trajectory Prediction for Safety Monitoring in Low-Altitude Airspace. In: AIAA Aviation 2019 Forum, AIAA AVIATION Forum. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2019-3514.

Coskun, H., Achilles, F., DiPietro, R., Navab, N., Tombari, F., 2017. Long Short-Term Memory Kalman Filters: Recurrent Neural Estimators for Pose Regularization. Presented at the Proceedings of the IEEE International Conference on Computer Vision, pp. 5524–5532.

Dalmau, R., Pérez-Batlle, M., Prats, X., 2018. Real-time Identification of Guidance Modes in Aircraft Descents Using Surveillace Data. IEEE, pp. 1–10. https://doi.org/10.1109/DASC.2018.8569811.

EUROCONTROL, 2017. EUROCONTROL Specification for Trajectory Prediction, 2.0. ed. EUROCONTROL, Brussels.

Fern, L., Rorie, R.C., Pack, J., Shively, J., Draper, M., 2015. An Evaluation of Detect and Avoid (DAA) Displays for Unmanned Aircraft Systems: The Effect of Information Level and Display Location on Pilot Performance. In: 15th AIAA Aviation Technology, Integration, and Operations Conference. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2015-3327.
Fildes, R., Hibon, M., Makridakis, S., Meade, N., 1998. Generalising about univariate forecasting methods: further empirical evidence. Int. J. Forecast. 14, 339–358. https://doi.org/10.1016/S0169-2070(98)00009-0.
García, J., Besada, J.A., Molina, J.M., de Miguel, G., 2015. Model-based trajectory reconstruction with IMM smoothing and segmentation. Inf. Fusion 22, 127–140. https://doi.org/10.1016/j.inffus.2014.06.004.
Geng, M., Li, J., Xia, Y., Chen, X., 2023. A physics-informed transformer model for vehicle trajectory prediction on highways. Transp. Res. Part C Emerging Technol. 154, 104272. https://doi.org/10.1016/j.trc.2023.104272.
Graves, A., 2013. Generating Sequences With Recurrent Neural Networks.
Gu, T., Chen, G., Li, J., Lin, C., Rao, Y., Zhou, J., Lu, J., 2022. Stochastic Trajectory Prediction via Motion Indeterminacy Diffusion. Presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 17113–17122.
Guan, L., Shi, J., Wang, D., Shao, H., Chen, Z., Chu, D., 2023. A trajectory prediction method based on bayonet importance encoding and bidirectional LSTM. Expert Syst. Appl. 223, 119888. https://doi.org/10.1016/j.eswa.2023.119888.
Guo, D., Zhang, Z., Yan, Z., Zhang, J., Lin, Y., 2024. FlightBERT++: a non-autoregressive multi-horizon flight trajectory prediction framework. Proc. AAAI Conf. Artif. Intell. 38, 127–134. https://doi.org/10.1609/aaai.v38i1.27763.
Hrastovec, M., Solina, F., 2016. Prediction of aircraft performances based on data collected by air traffic control centers. Transp. Res. Part C Emerging Technol. 73, 167–182. https://doi.org/10.1016/j.trc.2016.10.018.
Huang, K., Shao, K., Zhen, S., Sun, H., Yu, R., 2016. A novel approach for trajectory tracking control of an under-actuated quad-rotor UAV. IEEE/CAA J. Autom. Sin. 1–10. https://doi.org/10.1109/JAS.2016.7510238.
Jia, P., Chen, H., Zhang, L., Han, D., 2022. Attention-LSTM based prediction model for aircraft 4-D trajectory. Sci. Rep. 12, 15533. https://doi.org/10.1038/s41598-022-19794-1.
Jiang, C., Fang, Y., Zhao, P., Panneerselvam, J., 2020. Intelligent UAV identity authentication and safety supervision based on behavior modeling and prediction. IEEE Trans. Ind. Inf. 16, 6652–6662. https://doi.org/10.1109/TII.2020.2966758.
Jilkov, V.P., Ledet, J.H., Li, X.R., 2019. Multiple model method for aircraft conflict detection and resolution in intent and weather uncertainty. IEEE Trans. Aerosp. Electron. Syst. 55, 1004–1020. https://doi.org/10.1109/TAES.2018.2867698.
Jin, Z., Ng, K.K.H., Zhang, C., Wu, L., Li, A., 2024. Integrated optimisation of strategic planning and service operations for urban air mobility systems. Transp. Res. A Policy Pract. 183, 104059. https://doi.org/10.1016/j.tra.2024.104059.
Jo, K., Chu, K., Sunwoo, M., 2012. Interacting multiple model filter-based sensor fusion of GPS with in-vehicle sensors for real-time vehicle positioning. IEEE Trans. Intell. Transport. Syst. 13, 329–343. https://doi.org/10.1109/TITS.2011.2171033.
Khaledian, H., Prats Menéndez, X., Vilà-Valls, J., 2020. Real-time Identification of High-Lift Devices Deployment in Aircraft Descents (An Interacting Multiple Model Filtering Application Validated with Simulated Trajectories). Presented at the 10th SESAR Innovation Days: 7th of December-10th of December, 2020, virtual event, Single European Sky ATM Research (SESAR), pp. 1–8.
Khaledian, H., Sáez, R., Vilà-Valls, J., Prats, X., 2023. Interacting multiple model filtering for aircraft guidance modes identification from surveillance data. J. Guid. Control Dynam. 46, 1580–1595. https://doi.org/10.2514/1.G007139.
Khaledian, H., Sáez, R., Vilà-Valls, J., Prats, X., 2022. On the Impact of Guidance Commands Mismatch in IMM-Based Guidance Modes Identification for Aircraft Trajectory Prediction, in: 2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC). Presented at the 2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC), pp. 1–6. Doi: 10.1109/DASC55683.2022.9925846.
Khodarahmi, M., Maihami, V., 2023. A review on kalman filter models. Arch. Comp. Methods Eng. 30, 727–747. https://doi.org/10.1007/s11831-022-09815-7.
Kumar, R., Agrawal, A.K., 2021. Drone GPS data analysis for flight path reconstruction: a study on DJI, Parrot & Yuneec make drones. Forensic Sci. Int.: Digital Invest. 38, 301182. https://doi.org/10.1016/j.fsidi.2021.301182.
Li, H., Jiao, H., Yang, Z., 2023a. Ship trajectory prediction based on machine learning and deep learning: a systematic review and methods analysis. Eng. Appl. Artif. Intel. 126, 107062. https://doi.org/10.1016/j.engappai.2023.107062.
Li, R., Qin, Y., Wang, J., Wang, H., 2023b. AMGB: trajectory prediction using attention-based mechanism GCN-BiLSTM in IOV. Pattern Recogn. Lett. 169, 17–27. https://doi.org/10.1016/j.patrec.2023.03.006.
Li, X.R., Bar-Shalom, Y., 1993. Design of an interacting multiple model algorithm for air traffic control tracking. IEEE Trans. Control Syst. Technol. 1, 186–194. https://doi.org/10.1109/87.251886.
Ligas, M., Banasik, P., 2011. Conversion between Cartesian and geodetic coordinates on a rotational ellipsoid by solving a system of nonlinear equations. Geodesy and Cartography 60, 145–159. https://doi.org/10.2478/v10277-012-0013-x.
Lin, Y., Zhang, J., Liu, H., 2018. An algorithm for trajectory prediction of flight plan based on relative motion between positions. Front. Inf. Technol. Electronic. Eng. 19, 905–916. https://doi.org/10.1631/FITEE.1700224.
Liu, S., Wang, Y., Sun, J., Mao, T., 2022. An efficient Spatial–Temporal model based on gated linear units for trajectory prediction. Neurocomputing 492, 593–600. https://doi.org/10.1016/j.neucom.2021.12.051.
Liu, W., Hwang, I., 2011. Probabilistic trajectory prediction and conflict detection for air traffic control. J. Guid. Control Dynam. 34, 1779–1789. https://doi.org/10.2514/1.53645.
Liu, Y., Ng, K.K.H., Chu, N., Hon, K.K., Zhang, X., 2023. Spatiotemporal image-based flight trajectory clustering model with deep convolutional autoencoder network. J. Aerospace Inform. Syst. 20, 575–587. https://doi.org/10.2514/1.I011194.
Loshchilov, I., Hutter, F., 2018. Decoupled Weight Decay Regularization. Presented at the International Conference on Learning Representations.
Luo, C., McClean, S.I., Parr, G., Teacy, L., De Nardi, R., 2013. UAV position estimation and collision avoidance using the extended kalman filter. IEEE Trans. Veh. Technol. 62, 2749–2762. https://doi.org/10.1109/TVT.2013.2243480.
Lykou, G., Moustakas, D., Gritzalis, D., 2020. Defending airports from UAS: a survey on cyber-attacks and counter-drone sensing technologies. Sensors 20, 3537. https://doi.org/10.3390/s20123537.
Maeder, U., Morari, M., Baumgartner, T.I., 2011. Trajectory prediction for light aircraft. J. Guid. Control Dynam. 34, 1112–1119. https://doi.org/10.2514/1.52124.
Majdik, A.L., Till, C., Scaramuzza, D., 2017. The Zurich urban micro aerial vehicle dataset. Int. J. Robot. Res. 36, 269–273. https://doi.org/10.1177/0278364917702237.
Makridakis, S., Hibon, M., 1997. ARMA Models and the Box–Jenkins Methodology. J. Forecast. 16, 147–163. https://doi.org/10.1002/(SICI)1099-131X(199705)16:3<147::AID-FOR652>3.0.CO;2-X.
Malviya, V., Kala, R., 2022. Trajectory prediction and tracking using a multi-behaviour social particle filter. Appl. Intell. 52, 7158–7200. https://doi.org/10.1007/s10489-021-02286-6.
Maza, I., Caballero, F., Capitán, J., Martínez-de-Dios, J.R., Ollero, A., 2011. Experimental results in multi-UAV coordination for disaster management and civil security applications. J. Intell. Robot. Syst. 61, 563–585. https://doi.org/10.1007/s10846-010-9497-5.
Mazor, E., Averbuch, A., Bar-Shalom, Y., Dayan, J., 1998. Interacting multiple model methods in target tracking: a survey. IEEE Trans. Aerosp. Electron. Syst. 34, 103–123. https://doi.org/10.1109/7.640267.
Mohamed, A., Zhu, D., Vu, W., Elhoseiny, M., Claudel, C., 2022. Social-Implicit: Rethinking Trajectory Prediction Evaluation and The Effectiveness of Implicit Maximum Likelihood Estimation. Springer Nature Switzerland, Cham, pp. 463–479. https://doi.org/10.1007/978-3-031-20047-2_27.
Mokhtarzadeh, H., Colten, T., 2015. Small UAV Position and Attitude, Raw Sensor, and Aerial Imagery Data Collected over Farm Field with Surveyed Markers.
Mondoloni, S., Rozen, N., 2020. Aircraft trajectory prediction and synchronization for air traffic management applications. Prog. Aerosp. Sci. 119, 100640. https://doi.org/10.1016/j.paerosci.2020.100640.

Nacar, O., Abdelkader, M., Ghouti, L., Gabr, K., Al-Batati, A., Koubaa, A., 2025. VECTOR: velocity-enhanced GRU neural network for real-time 3D UAV trajectory prediction. Drones 9, 8. https://doi.org/10.3390/drones9010008.
Ogasawara, E., Martinez, L.C., de Oliveira, D., Zimbrão, G., Pappa, G.L., Mattoso, M., 2010. Adaptive Normalisation: A novel data normalisation approach for non-stationary time series, in: The 2010 International Joint Conference on Neural Networks (IJCNN). Presented at the The 2010 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. Doi: 10.1109/IJCNN.2010.5596746.
Pang, Y., Yao, H., Hu, J., Liu, Y., 2019. A Recurrent Neural Network Approach for Aircraft Trajectory Prediction with Weather Features From Sherlock. In: AIAA Aviation 2019 Forum, AIAA AVIATION Forum. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2019-3413.
Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., Iosifidis, A., 2020. Deep adaptive input normalisation for time series forecasting. IEEE Trans. Neural Networks Learn. Syst. 31, 3760–3765. https://doi.org/10.1109/TNNLS.2019.2944933.
Pyrgies, J., 2019. The UAVs threat to airport security: risk analysis and mitigation. J. Airline Airport Manag. 9, 63–96. https://doi.org/10.3926/jairm.127.
Qiao, S., Shen, D., Wang, X., Han, N., Zhu, W., 2015. A self-adaptive parameter selection trajectory prediction approach via hidden markov models. IEEE Trans. Intell. Transp. Syst. 16, 284–296. https://doi.org/10.1109/TITS.2014.2331758.
Ren, J., Wu, X., Liu, Y., Ni, F., Bo, Y., Jiang, C., 2023. Long-term trajectory prediction of hypersonic glide vehicle based on physics-informed transformer. IEEE Trans. Aerosp. Electron. Syst. 59, 9551–9561. https://doi.org/10.1109/TAES.2023.3322977.
Ren, L., Castillo-Effen, M., Yu, H., Yoon, Y., Nakamura, T., Johnson, E.N., Ippolito, C.A., 2017. Small Unmanned Aircraft System (sUAS) Trajectory Modeling in Support of UAS Traffic Management (UTM), in: 17th AIAA Aviation Technology, Integration, and Operations Conference. American Institute of Aeronautics and Astronautics. Doi: 10.2514/6.2017-4268.
Ri, S., Ye, J., Toyama, N., Ogura, N., 2024. Drone-based displacement measurement of infrastructures utilising phase information. Nat. Commun. 15, 395. https://doi.org/10.1038/s41467-023-44649-2.
Rodrigues, Thiago A., Patrikar, Jay, Choudhry, Arnav, Feldgoise, Jacob, Arcot, Vaibhav, Gahlaut, Aradhana, Lau, Sophia, Moon, Brady, Wagner, Bastian, Matthews, H Scott, Scherer, Sebastian, Samaras, Constantine, 2021. Data Collected with Package Delivery Quadcopter Drone. Doi: 10.1184/R1/12683453.V1.
Rong Li, X., Jilkov, V.P., 2005. Survey of maneuvering target tracking. part v. multiple-model methods. IEEE Trans. Aerosp. Electron. Syst. 41, 1255–1321. https://doi.org/10.1109/TAES.2005.1561886.
Rudenko, A., Palmieri, L., Herman, M., Kitani, K.M., Gavrila, D.M., Arras, K.O., 2020. Human motion trajectory prediction: a survey. Int. J. Robot. Res. 39, 895–935. https://doi.org/10.1177/0278364920917446.
Ruseno, N., Lin, C.-Y., 2024. Real-time UAV trajectory prediction for UTM surveillance using machine learning. In: Un. Sys., pp. 1–15. https://doi.org/10.1142/S230138502550030X.
Sahadevan, D., M, H.P., Ponnusamy, P., Gopi, V.P., Nelli, M.K., 2022. Ground-based 4d trajectory prediction using bi-directional LSTM networks. Appl. Intell. 52, 16417–16434. https://doi.org/10.1007/s10489-022-03309-6.
Salzmann, T., Ivanovic, B., Chakravarty, P., Pavone, M., 2020. Trajectron++: Dynamically-Feasible Trajectory Forecasting with Heterogeneous Data. Springer International Publishing, Cham, pp. 683–700. https://doi.org/10.1007/978-3-030-58523-5_40.
Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W., Woo, W., 2015. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. Advances in Neural Information Processing Systems. Curran Associates, Inc.
Shi, Z., Xu, M., Pan, Q., Yan, B., Zhang, H., 2018. LSTM-based Flight Trajectory Prediction, in: 2018 International Joint Conference on Neural Networks (IJCNN). Presented at the 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. Doi: 10.1109/IJCNN.2018.8489734.
Solodov, A., Williams, A., Al Hanaei, S., Goddard, B., 2018. Analyzing the threat of unmanned aerial vehicles (UAV) to nuclear facilities. Secur J 31, 305–324. https://doi.org/10.1057/s41284-017-0102-5.
Sun, J., Ellerbroek, J., Hoekstra, J.M., 2019. WRAP: An open-source kinematic aircraft performance model. Transp. Res. Part C Emerging Technol. 98, 118–138. https://doi.org/10.1016/j.trc.2018.11.009.
Sutskever, I., Vinyals, O., Le, Q.V., 2014. Sequence to sequence learning with neural networks. Adv. Neural Inform. Proc. Syst. 27.
Swinney, C.J., Woods, J.C., 2022. A review of security incidents and defence techniques relating to the malicious use of small unmanned aerial systems. IEEE Aerosp. Electron. Syst. Mag. 37, 14–28. https://doi.org/10.1109/MAES.2022.3151308.
Tastambekov, K., Puechmorel, S., Delahaye, D., Rabut, C., 2014. Aircraft trajectory forecasting using local functional regression in Sobolev space. Transp. Res. Part C Emerging Technol. 39, 1–22. https://doi.org/10.1016/j.trc.2013.11.013.
Ulyanov, D., Vedaldi, A., Lempitsky, V., 2017. Instance Normalisation: The Missing Ingredient for Fast Stylization. Doi: 10.48550/arXiv.1607.08022.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Curran Associates Inc., Red Hook, NY, USA, pp. 6000–6010.
Verdonk Gallego, C.E., Gómez Comendador, V.F., Amaro Carmona, M.A., Arnaldo Valdés, R.M., Sáez Nieto, F.J., García Martínez, M., 2019. A machine learning approach to air traffic interdependency modelling and its application to trajectory prediction. Transp. Res. Part C Emerging Technol. 107, 356–386. https://doi.org/10.1016/j.trc.2019.08.015.
Verdonk Gallego, C.E., Gómez Comendador, V.F., Sáez Nieto, F.J., Orenga Imaz, G., Arnaldo Valdés, R.M., 2018. Analysis of air traffic control operational impact on aircraft vertical profiles supported by machine learning. Transp. Res. Part C Emerging Technol. 95, 883–903. https://doi.org/10.1016/j.trc.2018.03.017.
Wang, Q., Huang, J., 2014. A VB-IMM filter for ADS-B data, in: 2014 12th International Conference on Signal Processing (ICSP). Presented at the 2014 12th International Conference on Signal Processing (ICSP), pp. 2130–2134. Doi: 10.1109/ICOSP.2014.7015371.
Wiest, J., Höffken, M., Kreßel, U., Dietmayer, K., 2012. Probabilistic trajectory prediction with Gaussian mixture models, in: 2012 IEEE Intelligent Vehicles Symposium. Presented at the 2012 IEEE Intelligent Vehicles Symposium, pp. 141–146. Doi: 10.1109/IVS.2012.6232277.
Wu, M.G., Lee, S., Cone, A.C., 2019. Detect and Avoid Alerting Performance with Limited Surveillance Volume for Non-Cooperative Aircraft. In: AIAA Scitech 2019 Forum. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2019-2073.
Xie, G., Gao, H., Qian, L., Huang, B., Li, K., Wang, J., 2018. Vehicle trajectory prediction by integrating physics- and maneuver-based approaches using interactive multiple models. IEEE Trans. Ind. Electron. 65, 5999–6008. https://doi.org/10.1109/TIE.2017.2782236.
Yang, Z., Tang, R., Bao, J., Lu, J., Zhang, Z., 2020. A real-time trajectory prediction method of small-scale quadrotors based on gps data and neural network. Sensors 20, 7061. https://doi.org/10.3390/s20247061.
Zeng, W., Chu, X., Xu, Z., Liu, Y., Quan, Z., 2022. Aircraft 4D trajectory prediction in civil aviation: a review. Aerospace 9, 91. https://doi.org/10.3390/aerospace9020091.
Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., Eickhoff, C., 2021. A Transformer-based Framework for Multivariate Time Series Representation Learning, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, KDD '21. Association for Computing Machinery, New York, NY, USA, pp. 2114–2124. Doi: 10.1145/3447548.3467401.
Zhang, Z., Guo, D., Zhou, S., Zhang, J., Lin, Y., 2023. Flight trajectory prediction enabled by time-frequency wavelet transform. Nat. Commun. 14, 5258. https://doi.org/10.1038/s41467-023-40903-9.
Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W., 2021. Informer: beyond efficient transformer for long sequence time-series forecasting. Proc. AAAI Conf. Artif. Intell. 35, 11106–11115.
Zhu, W., Wang, W., Yuan, G., 2016. An improved interacting multiple model filtering algorithm based on the cubature kalman filter for maneuvering target tracking. Sensors 16, 805. https://doi.org/10.3390/s16060805.
Zubača, J., Stolz, M., Seeber, R., Schratter, M., Watzenig, D., 2022. Innovative interaction approach in IMM filtering for vehicle motion models with unequal states dimension. IEEE Trans. Veh. Technol. 71, 3579–3594. https://doi.org/10.1109/TVT.2022.3146626.