

Article

A Lightweight LSTM Model for Flight Trajectory Prediction in Autonomous UAVs

Disen Jia, Jonathan Kua *  and Xiao Liu 

School of Information Technology, Deakin University, Geelong, VIC 3220, Australia;
disen.jia@deakin.edu.au (D.J.); xiao.liu@deakin.edu.au (X.L.)

* Correspondence: jonathan.kua@deakin.edu.au

Abstract

Autonomous Unmanned Aerial Vehicles (UAVs) are widely used in smart agriculture, logistics, and warehouse management, where precise trajectory prediction is important for safety and efficiency. Traditional approaches require complex physical modeling including mass properties, moment of inertia measurements, and aerodynamic coefficient calculations, which creates significant barriers for custom-built UAVs. Existing trajectory prediction methods are primarily designed for motion forecasting from dense historical observations, which are unsuitable for scenarios lacking historical data (e.g., takeoff phases) or requiring trajectory generation from sparse waypoint specifications (4–6 constraint points). This distinction necessitates architectural designs optimized for spatial interpolation rather than temporal extrapolation. To address these limitations, we present a segmented LSTM framework for complete autonomous flight trajectory prediction. Given target waypoints, our architecture decomposes flight operations, predicts different maneuver types, and outputs the complete trajectory, demonstrating new possibilities for mission-level trajectory planning in autonomous UAV systems. The system consists of a global duration predictor (0.124 MB) and five segment-specific trajectory generators (~1.17 MB each), with a total size of 5.98 MB and can be deployed in various edge devices. Validated on real Crazyflie 2.1 data, our framework demonstrates high accuracy and provides reliable arrival time predictions, with an Average Displacement Error ranging from 0.0252 m to 0.1136 m. This data-driven approach avoids complex parameter configuration requirements, supports lightweight deployment in edge computing environments, and provides an effective solution for multi-UAV coordination and mission planning applications.

Keywords: UAV trajectory prediction; data-driven modeling; segmented LSTM; complete flight trajectory prediction; lightweight deployment; edge computing



Academic Editors: Andrey V. Savkin
and Hailong Huang

Received: 11 November 2025

Revised: 8 December 2025

Accepted: 17 December 2025

Published: 20 December 2025

Copyright: © 2025 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the [Creative Commons
Attribution \(CC BY\) license](https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Autonomous Unmanned Aerial Vehicles (UAVs) are important to modern applications spanning smart agriculture, logistics, warehouse management, and autonomous delivery systems. The success of these applications requires precise trajectory prediction to enable safe, efficient, and autonomous flight operations [1].

Traditional trajectory prediction approaches use simplified mathematical models. They assume the UAV can perfectly track the planned path. This neglects the complex dynamics and real-world disturbances of actual flight [2]. These limitations become obvious in high-precision tasks such as autonomous navigation in dense environments, precision agriculture operations, and coordinated multi-UAV missions. In these scenarios, trajectory

prediction errors can lead to collision risks, mission failures, and reduced operational efficiency [3].

The emerging paradigm of edge computing offers new opportunities for UAV trajectory prediction systems [4]. Due to limitations of onboard computing resources, offloading computational tasks to edge environments enables more sophisticated trajectory prediction while preserving battery life [5]. This motivates the development of lightweight models suitable for distributed deployment across edge devices.

Recent advances in machine learning, particularly deep learning methods, have helped address these challenges. LSTM-based approaches have demonstrated great capability in capturing temporal dependencies in UAV trajectories [6,7], while trajectory segmentation techniques have emerged as effective strategies for managing complex flight patterns [8]. Despite these advances, several limitations persist in current UAV trajectory prediction research. Existing trajectory prediction methods are primarily designed for motion forecasting from dense historical observations, which are unsuitable for scenarios lacking historical data (e.g., takeoff phases) or requiring complete trajectory generation from sparse waypoint specifications (4–6 constraint points). Additionally, state-of-the-art architectures exhibit computational challenges: Transformer-based models demonstrate quadratic complexity ($O(T^2)$), while probabilistic generative models require multiple sampling iterations for diverse trajectory generation. As noted in recent surveys, these computational characteristics can “impact the ability of the drone to make rapid trajectory adjustments” [9]. Conversely, simpler lightweight models such as basic LSTM or small feedforward networks lack the representational capacity to generate complete multi-segment trajectories spanning extended durations with diverse maneuver types.

This paper addresses these limitations by proposing a data-driven UAV trajectory prediction framework based on segmented LSTM architecture (Implementation available at: <https://github.com/JudsonJia/drone-trajectory-lstm>, accessed on 16 December 2025). Our framework decomposes autonomous missions into five specialized maneuver types, enabling complete flight trajectory prediction through a combination of global duration estimation and segment-specific trajectory generation. The system is designed for lightweight deployment in edge computing environments, with a total model size of only 5.98 MB (0.124 MB duration predictor + 5×1.17 MB trajectory generators), making it suitable for integration with UAV systems.

The key contributions of this paper are:

- Introduce a trajectory prediction approach that generates complete spatiotemporal sequences (5–25 s) from sparse waypoint specifications (4–6 constraint points), addressing a distinct problem from existing motion forecasting methods designed for dense historical observations.
- Develop a segmented LSTM architecture where each segment specializes in distinct flight maneuvers. The architecture incorporates physics-informed constraints to ensure realistic trajectory generation while preserving computational efficiency. Each maneuver-specific module requires only 1.17 MB, enabling flexible deployment either onboard UAVs or on edge devices.
- Demonstrate the practical applicability of our approach through validation with real Crazyflie 2.1 UAV (<https://www.bitcraze.io/products/old-products/crazyflie-2-1/>, accessed on 16 December 2025) flight data collection (Data collection framework available at: <https://github.com/JudsonJia/crazyfly-data-collection>, accessed on 16 December 2025). Our validation shows prediction accuracy with Average Displacement Error ranging from 0.0252 m to 0.1136 m across different maneuver types.

The proposed framework has broad implications for UAV applications requiring predictive planning capabilities. In smart agriculture, the system can enable autonomous crop monitoring missions with precise timing and coverage predictions. For logistics and warehouse management, it supports automated delivery systems with reliable arrival time estimation and path optimization. The lightweight architecture makes it suitable for deployment in multi-UAV coordination scenarios. Furthermore, the data-driven approach avoids the need for complex parameter configuration requirements typical of physics-based methods. This accessibility is valuable for the growing ecosystem of custom-built UAV systems where traditional modeling approaches face significant implementation barriers.

The remainder of this paper is organized as follows. Section 2 reviews current UAV trajectory prediction approaches and identifies research gaps. Section 3 presents the experimental setup and framework design. Section 4 details the data collection and segmented LSTM training framework. Section 5 provides evaluation results using real flight data. Section 6 presents our insights and discussion on limitations and threats to validity. Section 7 concludes the paper and presents future research directions.

2. Background and Related Work

The field of UAV trajectory prediction has evolved from traditional mathematical approaches to advanced machine learning methodologies. This section reviews current methods and identifies gaps motivating our work.

2.1. Traditional and Machine Learning Approaches

UAV trajectory prediction has traditionally relied on physics-based models incorporating kinematic and dynamic constraints. A recent comprehensive survey by Shukla et al. [9] reveals that UAV trajectory prediction methods have evolved from traditional kinematic models to diverse paradigms encompassing machine learning, deep learning, and reinforcement learning. The survey emphasizes tight integration between trajectory prediction and planning systems, and identifies key future directions: multi-UAV cooperative prediction for efficiency optimization, generative AI (e.g., GANs, diffusion models) for diverse trajectory generation, and lightweight deployment solutions for resource-constrained environments. Traditional models, while computationally efficient, suffer from oversimplified assumptions, poor environmental adaptability, and require detailed system parameters including mass properties, moment of inertia, and aerodynamic coefficients. This poses practical obstacles for rapid deployment across diverse platforms.

Deep learning models, particularly Long Short-Term Memory (LSTM) networks [10], have emerged as dominant solutions for capturing temporal dependencies in sequential prediction tasks. Foundational trajectory prediction works such as Social LSTM [11] demonstrated LSTM's effectiveness for learning complex motion patterns from observed sequences in crowded pedestrian scenarios, establishing the viability of data-driven approaches over hand-crafted social force models. These early successes motivated subsequent applications of LSTM architectures to diverse trajectory forecasting domains, including autonomous vehicles and aerial platforms. Shu et al. [7] achieved centimeter-level accuracy with Stacked Bidirectional and Unidirectional LSTM (SBULSTM), while Zhang et al. [6] demonstrated 6.25 m average error for 3D trajectories using Recurrent LSTM with ADS-B data integration. Peng et al. [12] proposed sliding window LSTM (9 time steps), which required an average inference time of 1.219 s to complete 500 prediction steps, outperforming polynomial fitting for hovering and turning trajectories. Chen et al. [13] developed EC-Bessel-BiLSTM deployed on ground stations via 5G (transmission delay < 400 μ s), achieving RMSE below 1 m in 0.1 s with 30,000–40,000 parameters through Bessel coordinate transformation and PID-based error compensation.

Wang et al. [14] combined LSTM with Particle Swarm Optimization for path planning, achieving 3.05% path length reduction. Palamas et al. [15] introduced multi-task learning for simultaneous state identification (98.51% accuracy) and trajectory prediction (<3.44 m error at 3 s). Zhang et al. [16] transformed prediction into a classification problem by segmenting the reachable airspace and predicting grid cell probabilities via an Attention-based ConvLSTM (AttConvLSTM) network. On a high-speed aircraft dataset, their model's error increased with the prediction horizon, reaching a mean absolute error of 45.73 m at 2.5 s, while maintaining much higher accuracy (e.g., 2.09 m) at shorter horizons (e.g., 0.5 s). Despite these advances, nearly all methods focus on short-term horizons, insufficient for complete flight trajectory planning.

2.2. Trajectory Segmentation and Advanced Strategies

To improve prediction accuracy, researchers have explored trajectory segmentation and hierarchical strategies. Some approaches also attempt to extend prediction horizons beyond typical 1–3 s windows.

Conte et al. [8] introduced trajectory segmentation for time-of-flight prediction, decomposing paths into “corner” sub-paths and achieving 3.2% error. However, their work produced only temporal predictions, not complete spatiotemporal trajectories.

The effectiveness of trajectory segmentation has been demonstrated in aviation contexts. Phisannupawong et al. [17] proposed Aircraft Trajectory Segmentation-based Contrastive Coding (ATSCC), which uses the Ramer-Douglas-Peucker algorithm to automatically identify turning points and partition trajectories into segments. Through contrastive learning on these segments, ATSCC achieves superior performance in trajectory classification and clustering across multiple airport datasets without requiring labeled data or predefined configurations.

While ATSCC applies segmentation to discriminative tasks (representation learning for classification), our work applies it to trajectory generation from sparse waypoints. Both demonstrate that decomposing trajectories into semantically coherent segments enables more effective flight dynamics modeling, though the specific objectives differ: ATSCC learns representations for analysis, while our framework generates complete spatiotemporal trajectories for planning.

Xie et al. [18] proposed hierarchical prediction combining Ada-AE-DeepESN for long-term maneuver unit prediction with GWSPSO-LSTM for short-term refinement. Their maneuver library contains 21 basic types (horizontal, vertical, spatial patterns), achieving 0.002 s per step with 6–7 m horizontal errors. This approach extends prediction horizons to 10–20 s but still falls short of complete trajectory coverage. Zhang et al. [19] proposed a framework that first classifies UAV flight into five states (climb, level flight, turn, descent, circle) using PCA-SVM, then applies state-specific BP neural networks for trajectory prediction. This flight state recognition approach achieved mean prediction error of 0.214 m, with maximum error of 0.41 m in the challenging circling state, substantially outperforming direct neural network prediction without state recognition. While demonstrating segmentation's value, these methods lack complete spatiotemporal trajectory generation.

For multi-UAV scenarios, Sun et al. [20] combined LSTM with attention-based GCN for air combat prediction (0.713 km average displacement error across 1v1, 2v2, 2v3 engagements, outperforming LSTM-only methods at 0.826 km). Wang et al. [21] used GAN with cooperative information interaction (GAN-CI) for eight UAVs, achieving 0.397 km average error and 0.619 km final error—45% improvement over non-cooperative GANs and 38% over LSTM with pooling. Ma et al. [22] proposed LSTM-based framework for swarm trajectory prediction with intent recognition, focusing on enemy cluster prediction for defense applications. Together, these studies highlight how trajectory prediction techniques

can capture interaction, cooperation, and intent within UAV groups, demonstrating the potential of such methods to enhance collective intelligence in multi-UAV scenarios.

2.3. Control-Oriented and Physics-Informed Methods

Recent research emphasizes integrating control considerations into prediction frameworks. Liu et al. [2] presented control-oriented trajectory planning using Trajectory-Mapping Network (TMN) based on TSCNN within an MPC framework, demonstrating strong obstacle avoidance performance and revealing the necessity of integrating UAV tracking characteristics.

As noted in recent surveys [9], future research is expected to combine control-oriented, physics-informed, and data-driven paradigms. In line with this trend, our method integrates lightweight physics-informed components by introducing physical loss constraints during training, enabling trajectory predictions consistent with UAV motion dynamics while retaining computational efficiency suitable for real-time, edge-deployed systems.

2.4. Deployment Constraints and Existing Solutions

Computational demands present significant challenges for real-time UAV operations. Edge computing solutions have been explored to address these limitations. Sankaranarayanan et al. [5] proposed PACED-5G architecture combining edge computing with predictive control to compensate for 5G communication delays. Their approach achieved RMS estimation error of 0.5 cm or less in all dimensions and mean tracking error below 6 cm. Chen et al. [13] demonstrated practical deployment by offloading to ground stations (transmission delay $<400 \mu\text{s}$), enabling 0.1 s inference with sub-meter accuracy using single-layer BiLSTM (60 hidden units, 30,000–40,000 parameters).

Recent UAV research has emphasized lightweight orchestration frameworks for edge deployment. Conti et al. [23] developed Twinflie, a digital twin UAV orchestrator and simulator that seamlessly integrates real and virtual UAV platforms for coordinated mission execution. Their framework demonstrates the feasibility of distributed AI-based navigation systems across heterogeneous edge nodes, particularly for indoor environments where GPS is unavailable. Similarly, Soliman et al. [24] proposed an AI-based UAV navigation framework combining deep reinforcement learning with digital twin technology for energy-efficient mobile target visitation, demonstrating how simulated training environments can reduce real-world deployment costs. Together, these works align with the growing trend toward flexible deployment strategies that balance computational efficiency with mission requirements.

Digital twin technology has been applied mainly for real-time control. Xiao et al. [25] achieved 0.127 m average error between virtual simulation and physical execution through continuous synchronization. Yang et al. [26] implemented digital twin-based obstacle avoidance using feedforward-feedback MPC (maximum error 0.18 m), offloading planning to virtual space via offline EGO-Planner computation.

However, these solutions are highly dependent on network/system infrastructure. Bandwidth-constrained environments and highly variable network conditions present significant challenges for real-time control in autonomous UAVs. Offloading approaches require reliable connectivity—empirical study on micro-UAVs [27] has shown that network bandwidth below 0.225–0.25 kbps causes control instability and oscillations. This bandwidth sensitivity limits autonomous operation in resource-constrained or intermittent connectivity environments. Digital twins require detailed physics-based modeling, tight synchronization, and high computational overhead. Moreover, they address control-level tracking rather than complete trajectory prediction. These edge-oriented solutions emphasize communication architectures to mitigate latency (particularly for latency-sensitive

applications such as real-time media streaming [28] and teleoperation [29]) rather than designing inherently lightweight models optimized for onboard deployment.

Application-specific optimizations achieve strong domain performance but lack generalizability. Karam et al. [30] developed OptiFly for agriculture (17.6% path reduction), Kong et al. [31] created attention-based pointer networks for delivery (18% path reduction versus genetic algorithms), and Suhartono et al. [32] introduced time compensation for cargo operations (<1 s error). These frameworks rely on domain-specific assumptions limiting cross-platform applicability.

In contrast, our complete trajectory generation offers inherent tolerance to computational delays. Since entire trajectories are produced in a single inference, minor latency does not disrupt closed-loop control, mitigating strict real-time requirements of step-wise short-horizon methods. Furthermore, our segmented framework enables distributed deployment: each maneuver-specific prediction module requires only 1.17 MB, allowing independent onboard execution on individual UAVs or edge devices without centralized computation infrastructure.

2.5. Research Gaps and Positioning

Four critical gaps emerge from this review:

Complete trajectory Perspective: Most methods predict short-term ahead [6,7,15], serving immediate safety monitoring but limiting applicability for autonomous mission planning spanning complete flight operations from takeoff to landing. Existing long-horizon attempts either predict time only [8] or operate on limited windows [18].

Lightweight Deployment: While deep learning approaches achieve high accuracy, few are optimized for resource-constrained edge computing environments. Existing solutions rely on offloading to ground stations [13] or edge servers [5], creating communication dependencies. Frameworks enabling complete trajectory generation with flexible deployment on local resources and independence from continuous communication infrastructure remain absent.

Segmented Modeling: Trajectory segmentation has been explored for flight state classification [19] and time-of-flight prediction [8]. However, comprehensive segmented approaches that generate complete spatiotemporal trajectory sequences for different maneuver types through end-to-end learning remain underdeveloped.

Generalizability: Application-specific methods [30–32] require reconfiguration for new platforms. Physics-based approaches [9] need detailed parameters. Data-driven frameworks with lightweight physics constraints that adapt across platforms without extensive tuning are needed.

This research addresses these limitations through a segmented LSTM framework specifically designed for complete mission trajectory prediction with lightweight deployment characteristics. Our approach combines data-driven learning with lightweight physics-informed constraints, avoiding complex modeling while enabling scalable complete trajectory prediction capabilities for autonomous UAV operations.

3. System Design and Experimental Methodology

Our system consists of two main phases: Flight data collection using repeatable Crazyflie 2.1 missions, followed by segmented LSTM model training for trajectory prediction. This two-stage approach enables us to capture real-world flight behaviors and learn complete trajectory prediction capabilities.

3.1. Phase 1: Flight Data Collection

Figure 1 shows our experimental environment. It includes the 3D visualization interface of the Loco positioning system and the physical test environment.

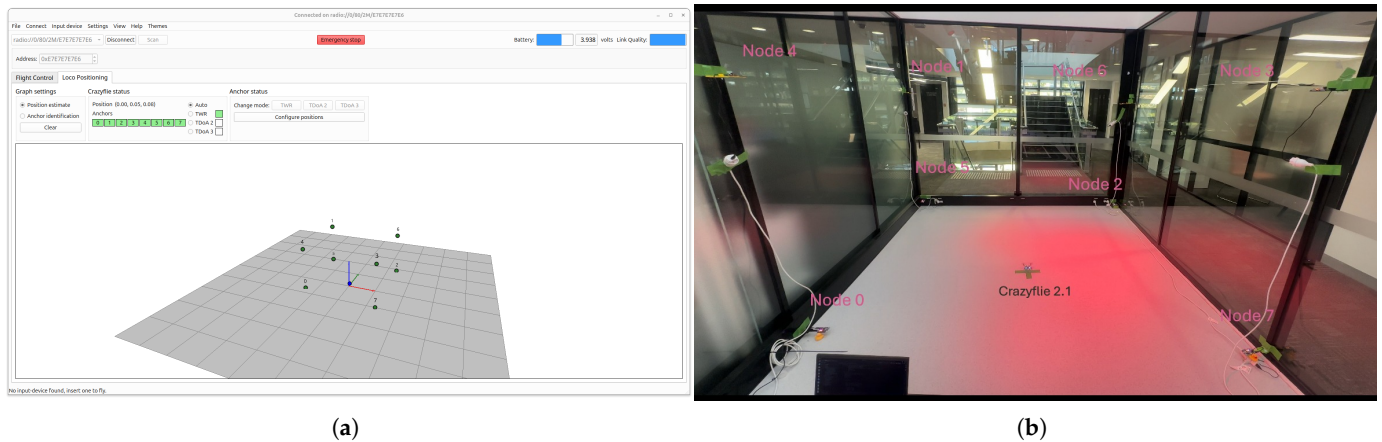


Figure 1. Experimental environment: (a) Software interface showing 3D anchor configuration and real-time tracking; (b) Physical test environment with Crazyflie UAV and UWB anchors. (a) The Loco positioning system interface displays 8 UWB anchor points and tracks the Crazyflie drone in real time. Battery voltage reads 3.938 V. Link quality is good. Anchors are numbered 0 to 7 across the experimental space. (b) The physical setup shows the Crazyflie 2.1 UAV at the center of a red flight zone. Eight UWB nodes (Nodes 0–7) surround the test area. A glass room encloses the space to maintain controlled conditions.

3.1.1. Experimental Testbed Setup

To collect real flight trajectory data of unmanned aerial vehicles with different trajectory lengths across various trajectory types, we established an experimental platform for data collection in a controlled environment. The hardware configuration consists of one Crazyflie 2.1 quadrotor, weighing 27 g, a Crazyradio PA USB dongle operating in the 2.4 GHz band with 20 dBm power amplification [33], and an 8-anchor Loco Positioning System providing sub-10 cm positioning accuracy.

For experimental repeatability, we used one Crazyflie drone equipped with STM32F405 microcontrollers, nRF51822 radio and power management MCUs, and BMI088 3-axis accelerometer/gyroscope with BMP388 high-precision pressure sensor [34]. The drone is also fitted with a DWM1000 module to interface with the positioning system for precise localization.

Accurate position tracking is essential for our experiments. While the theoretical minimum for 3D positioning is 4 anchors, we used the maximum supported 8 anchors for improved redundancy and accuracy. Each Loco Positioning Node is based on the Decawave DWM1000 module implementing IEEE 802.15.4 UWB standard [35]. The system provides ± 10 cm positioning accuracy in three dimensions with a tested range of up to 10 m. Each node is controlled by an STM32F072 MCU and can be configured to work in Anchor, Tag, or Sniffer mode [35].

The control station consists of a Linux-based computer with an Intel Core i7 processor, 16 GB RAM, and USB 3.0 ports for Crazyradio connection. This hardware setup creates a controlled environment where flight data can be collected while maintaining accurate position tracking, allowing us to isolate the effects of environment conditions on flight performance.

3.1.2. Flight Mission Design and Execution

We designed nine distinct flight patterns to systematically capture diverse flight behaviors across vertical and horizontal maneuvers. These patterns were carefully selected to represent fundamental UAV movement primitives that occur in real-world applications. Table 1 presents the set of mission patterns used in our data collection.

Throughout mission execution, the system continuously logs multi-sensor data at 100 Hz sampling rate, capturing position, velocity, attitude, and acceleration measurements with precise temporal synchronization.

Table 1. Flight Mission Waypoint Coordinates.

Mission Pattern	Waypoint Sequence (x, y, z)
vertical_small	(0, 0, 0.3) → (0, 0, 0.6) → (0, 0, 0.9) → (0, 0, 0.6) → (0, 0, 0.3)
vertical_med	(0, 0, 0.4) → (0, 0, 0.8) → (0, 0, 0.4)
vertical_large	(0, 0, 0.5) → (0, 0, 1.0) → (0, 0, 0.5)
square_small	(0, 0, 0.3) → (0.3, 0, 0.3) → (0.3, 0.3, 0.3) → (0, 0.3, 0.3) → (0, 0, 0.3)
square_med	(0, 0, 0.4) → (0.5, 0, 0.4) → (0.5, 0.5, 0.4) → (0, 0.5, 0.4) → (0, 0, 0.4)
square_large	(0, 0, 0.5) → (0.7, 0, 0.5) → (0.7, 0.7, 0.5) → (0, 0.7, 0.5) → (0, 0, 0.5)
triangle_small	(0, 0, 0.6) → (0.4, 0, 0.6) → (0.2, 0.4, 0.6) → (0, 0, 0.6)
triangle_med	(0, 0, 0.6) → (0.5, 0, 0.6) → (0.25, 0.5, 0.6) → (0, 0, 0.6)
triangle_large	(0, 0, 0.6) → (0.8, 0, 0.6) → (0.4, 0.6, 0.6) → (0, 0, 0.6)

Mission Execution Characteristics: The vertical patterns capture pure z-axis movements at fixed horizontal positions. Square trajectories involve alternating single-axis horizontal movements (x-only or y-only changes between waypoints), while triangle patterns combine single-axis movements with diagonal segments requiring simultaneous x-y coordinate changes. This design systematically covers fundamental movement primitives: vertical-only, x-only, y-only, and combined x-y movements.

3.2. Phase 2: Segmented LSTM Model Training

All models were trained using an NVIDIA GeForce RTX 3050 Laptop GPU (NVIDIA Corporation, Santa Clara, CA, USA) with sufficient memory to accommodate the specified batch sizes.

3.2.1. Problem Formulation

We formulate trajectory prediction as a two-tier supervised learning problem. The first tier involves global duration prediction, mapping start positions, target positions, and maneuver types to mission duration, average velocity, and trajectory sampling points. The second tier generates detailed trajectories, combining position boundary conditions and predicted durations to produce complete spatiotemporal trajectory sequences.

3.2.2. Data Preprocessing and Segmentation

Raw flight data undergoes automated segmentation using our trajectory segmentation tool to extract individual autonomous missions from continuous data streams. The segmentation process categorizes trajectory segments into five fundamental maneuver types: takeoff sequences, vertical ascent, vertical descent, horizontal straight movements, and horizontal diagonal movements. Quality filtering removes incomplete or erroneous trajectory segments, while normalization processes trajectory coordinates, average velocities, durations, and other parameters to ensure training consistency.

3.2.3. Segmented LSTM Architecture

Our framework employs specialized neural networks optimized for each maneuver type. The Global Duration Predictor utilizes a 3-layer feed-forward network with segment embeddings, processing geometric features including start/end positions, direction vectors, distance, and segment type information to predict mission timing parameters.

Five segment-specific trajectory generators implement 2-layer LSTM architectures with time encoding and boundary constraint capabilities. Each generator processes start/end positions, initial velocity, and segment type embeddings to output complete spatiotemporal trajectories (positions and velocities indexed over time, given the predicted duration and sampling points). Direction vectors are computed internally from the position boundaries. The velocity output primarily serves to ensure physics consistency during training through velocity-position coherence constraints rather than as a direct prediction target.

3.3. Evaluation Methodology

3.3.1. Model Training Process Assessment

Global Duration Predictor Training Metrics:

(1) Duration Prediction Mean Absolute Error: Evaluates deviation between predicted mission duration and actual flight time; (2) Average Speed Prediction Mean Absolute Error: Validates predicted average flight speed accuracy; (3) Trajectory Sampling Points Prediction Mean Absolute Error: Assesses precision of predicted trajectory points.

Segment Trajectory Generator Training Metrics:

(1) Training Process ADE Monitoring: Tracks average displacement error convergence during training for each segment type; (2) Physics Consistency Loss: Mean squared error between velocities calculated from position differences and model-output velocities, ensuring trajectory physical plausibility; (3) Boundary Condition Loss: Evaluates satisfaction of start/end position and velocity constraints; (4) Acceleration Smoothness Loss: Ensures trajectory motion continuity and reasonableness.

Training Optimization Strategies:

Employs differentiated early stopping mechanisms (duration predictor patience = 20, trajectory generators patience = 30), adaptive learning rate scheduling (ReduceLROnPlateau), and gradient clipping (max_norm = 1.0) to ensure training stability.

3.3.2. Final Model Performance Evaluation

Spatial Accuracy Metrics:

Average Displacement Error (ADE) evaluation for five maneuver types (takeoff, vertical_up, vertical_down, horizontal_straight, horizontal_diagonal) to validate specialized modeling effectiveness.

Temporal Accuracy Metrics:

Mission duration prediction error: Evaluates absolute deviation between global duration predictor output and actual execution time.

System Integration Assessment:

Through 3D trajectory comparison and X-Y plane projection visualization analysis, geometric consistency between predicted and actual trajectories is evaluated. Combined with quantitative ADE metrics, validates overall prediction accuracy and trajectory shape fidelity when the global duration predictor and segment trajectory generators work jointly.

4. Proposed Architecture for Data-Driven UAV Trajectory Prediction Modeling

4.1. Trajectory Prediction System Architecture

Figure 2 provides an architectural overview of the complete UAV trajectory prediction system, illustrating the integration between five key components: data collection, data processing, model training, prediction inference, and evaluation/validation.

4.1.1. Data Collection Pipeline

The data collection pipeline addresses Crazyflie's packet size constraints through a two-stream acquisition strategy operating at 100 Hz. Under the hardware limit of 30 bytes for LogConfig payload, the 12-dimensional state vector must be divided into separate position-velocity and attitude-IMU log groups. System timing optimization relies on a delay measurement tool that measures LogConfig transmission delays and recommends optimal time slot configurations. The PureContinuousCollector implements 10 ms time slots to reconstruct complete flight states from temporally-aligned sensor streams. Waypoint commands are transmitted via Crazyradio PA to the onboard autopilot for autonomous trajectory following. Meanwhile, two independent LogConfig channels simultaneously capture position-velocity estimates and attitude-acceleration measurements. The delay-calibrated synchronization framework ensures temporal alignment within 10 ms windows, enabling reconstruction of complete flight states. Quality validation confirms measurement fusion completeness and consistency before structured data storage.

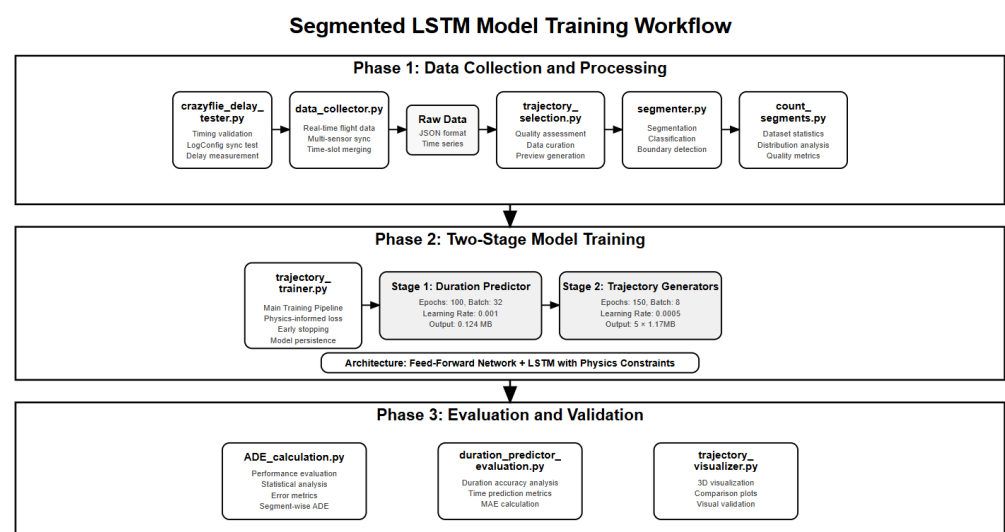


Figure 2. Three-phase architecture of the UAV trajectory prediction system: data collection and processing, two-stage model training, and evaluation and validation.

4.1.2. Trajectory Data Processing and Segmentation

The trajectory data processing pipeline employs a three-stage workflow to transform raw flight data into structured training segments for machine learning models.

Stage 1: Trajectory Selection

The initial processing phase employs trajectory selection script to curate high-quality flight data from the collected dataset. This component implements automated quality filtering based on temporal constraints, accepting complete flight trajectories with total durations between 5–25 s to ensure the quality of trajectory data. The script applies velocity smoothing using a 3-point moving average window to reduce sensor noise that could compromise training data quality.

Interactive visualization generates grid-based preview images showing either height-versus-time plots for vertical trajectories or X-Y plane projections for horizontal patterns. At this stage, we remove trajectories with insufficient motion characteristics due to system limitations. All square_small and triangle_small patterns are filtered out as the Crazyflie LPS positioning accuracy constraints result in poor horizontal flight trajectory features during short-distance movements. Additionally, vertical flight trajectories exhibiting excessive hovering durations caused by waypoint positioning judgment errors are excluded to maintain trajectory quality standards.

Stage 2: Trajectory Segmentation

The core segmentation script divides continuous flight data into semantically meaningful phases. The system recognizes five distinct segment types through rule-based classification: TAKEOFF (initial ground-to-flight transition), VERTICAL_UP (ascending flight phases), VERTICAL_DOWN (descending flight phases), HORIZONTAL_STRAIGHT (linear horizontal movements), and HORIZONTAL_DIAGONAL (angular horizontal transitions). The segmentation algorithm employs several key processing techniques. Takeoff phase splitting uses sustained vertical velocity analysis to separate initial transitions into preparation (TAKEOFF) and ascent (VERTICAL_UP) phases, improving dataset balance. Apex detection and correction identify true trajectory peaks independent of waypoint targets for vertical patterns. Multi-criteria waypoints arrival detection combines 0.08 m positional tolerance with 0.2 m/s velocity thresholding to determine segment boundaries accurately. Figure 3 demonstrates the effectiveness of the segmentation algorithm across different trajectory patterns, showing clear phase boundaries and velocity transitions for each flight category.

Stage 3: Dataset Analysis

The final processing stage employs dataset analysis tools to generate distribution statistics and validation results. The analysis component processes all segmented trajectory files to extract trajectory type distributions, segment type counts, and duration/distance characteristics.

The dataset contains 1939 trajectory segments divided into five flight categories, as detailed in Table 2. The distribution is well-balanced across categories. This gives us enough samples in each category for training the LSTM models. The dataset covers all main flight patterns, from takeoff to complex horizontal maneuvers.

Table 2. Final Dataset Composition After Trajectory Selection And Segmentation.

Segment Type	Samples	Duration Range (s)	Distance Range (m)
TAKEOFF	313	3.30–3.48	0.0055–0.0522
VERTICAL_UP	463	0.94–5.20	0.1921–0.6947
VERTICAL_DOWN	463	0.99–5.48	0.0332–0.5995
HORIZONTAL_STRAIGHT	400	1.00–7.63	0.4148–0.8308
HORIZONTAL_DIAGONAL	300	1.47–7.91	0.4338–0.8252
Total	1939	0.94–7.91	0.0055–0.8308

4.1.3. Segmented LSTM Training Framework

The training framework implements a two-stage optimization process with specialized handling for different trajectory types. The Global Duration Predictor employs a multi-layer feed-forward architecture with segment type embedding layers, trained using batch sizes of 32 samples with early stopping mechanisms (patience = 20 epochs) and learning rate scheduling to prevent overfitting. The Global Duration Predictor will be used in the prediction stage.

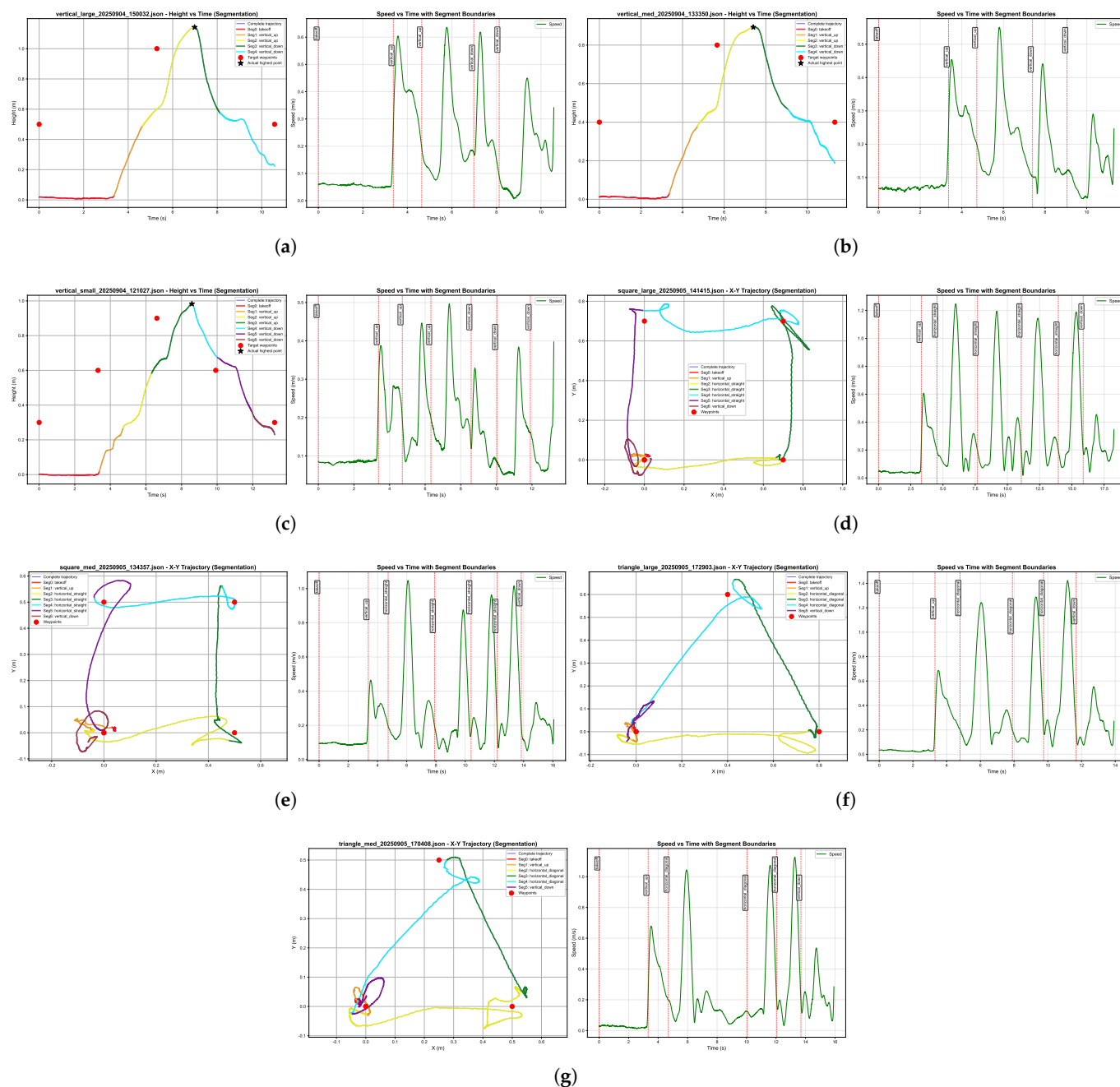


Figure 3. Trajectory Segmentation Results Across Different Flight Patterns. Left panels show trajectory paths with color-coded segments, right panels display speed profiles with segment boundaries. The segmentation algorithm successfully identifies distinct flight phases including takeoff preparation, vertical motion, and horizontal maneuvering patterns. (a) Vertical Large Trajectory. (b) Vertical Medium Trajectory. (c) Vertical Small Trajectory. (d) Square Large Trajectory. (e) Square Medium Trajectory. (f) Triangle Large Trajectory. (g) Triangle Medium Trajectory.

The segment-specific trajectory generators utilize 2-layer vertically stacked LSTM architectures. The first LSTM layer processes concatenated inputs of time embeddings (32 dimensions) and context vectors (128 dimensions), producing 128-dimensional hidden state sequences. The second LSTM layer receives the first layer’s output as input and generates the final 128-dimensional representations, which are subsequently mapped to 6-dimensional trajectory predictions through a linear output layer. Training is conducted with reduced batch sizes (8 samples) and extended training duration (150 epochs, patience = 30) to

accommodate the complexity of spatiotemporal sequence modeling. The generators optimize trajectory reconstruction through combined loss functions including position accuracy, velocity consistency, physics-informed constraints, and boundary condition losses.

For horizontal trajectory segments (straight and diagonal patterns), the training process incorporates intermediate point correction mechanisms. This enhancement applies soft constraints at strategic trajectory positions (1/4, 1/2, 3/4 points along the path) using weighted correction factors (weight = 0.3) that blend predicted and target positions. Additionally, horizontal segments use boundary loss weight of 15.0 versus 10.0 for vertical segments to improve geometric path adherence during training.

4.1.4. Physics-Informed Loss Function Implementation

The training process incorporates multiple loss components to ensure physical plausibility and geometric accuracy. Position reconstruction loss employs mean squared error between predicted and target trajectory points, representing the primary prediction objective. Velocity outputs serve primarily as auxiliary constraints to enhance position prediction quality through physics consistency constraints, ensuring coherence between model-predicted velocities and velocities derived from position differences, rather than as independent prediction targets.

Physics consistency constraints verify that velocities calculated from position differences match model velocity predictions:

$$\mathcal{L}_{physics} = \frac{1}{N-1} \sum_{i=0}^{N-2} \text{MSE} \left(\frac{\mathbf{p}_{i+1} - \mathbf{p}_i}{0.01}, \mathbf{v}_{pred,i} \right) \quad (1)$$

Boundary condition enforcement ensures trajectory start and end points match specified position and velocity constraints:

$$\mathcal{L}_{boundary} = \text{MSE}(\mathbf{p}_0, \mathbf{p}_{start}) + \text{MSE}(\mathbf{p}_{N-1}, \mathbf{p}_{end}) + \text{MSE}(\mathbf{v}_0, \mathbf{v}_{start}) + \text{MSE}(\mathbf{v}_{N-1}, \mathbf{v}_{target}) \quad (2)$$

Acceleration regularization prevents unrealistic motion discontinuities:

$$\mathcal{L}_{smooth} = \frac{1}{N-1} \sum_{i=0}^{N-2} |\mathbf{v}_{i+1} - \mathbf{v}_i| \quad (3)$$

The complete loss function combines these components with segment-specific weighting:

$$\mathcal{L}_{total} = \mathcal{L}_{position} + \mathcal{L}_{velocity} + \alpha \mathcal{L}_{physics} + \beta \mathcal{L}_{boundary} + \gamma \mathcal{L}_{smooth} \quad (4)$$

where $\alpha = 5.0$, $\beta = 10.0$ (15.0 for horizontal segments), and $\gamma = 0.02$ based on empirical optimization. These coefficients were empirically optimized to achieve a balanced trade-off between positional accuracy, physical consistency, and trajectory smoothness.

4.2. Trajectory Prediction Inference Pipeline

Two-Stage Prediction Process

The inference system accepts high-level mission commands specifying start position, target position, initial velocity, and maneuver type. The system automatically computes mission distance and direction vectors required for duration prediction. Mission execution prediction follows the trained two-tier architecture. The Global Duration Predictor first estimates mission timing parameters including total duration, average velocity, and number of sampling points. These temporal parameters are then provided to the appropriate segment-specific trajectory generator for detailed spatiotemporal sequence synthesis.

5. Results and Analysis

5.1. Experimental Setup and Evaluation Metrics

5.1.1. Evaluation Framework

The empirical evaluation uses a comprehensive assessment framework that evaluates both component-level performance and end-to-end trajectory prediction accuracy. The evaluation encompasses two primary dimensions: temporal accuracy through duration prediction assessment and spatial accuracy via trajectory reconstruction analysis.

5.1.2. Performance Metrics

Trajectory prediction accuracy is quantified using Average Displacement Error (ADE), calculated as the mean Euclidean distance between predicted and ground truth trajectory points:

$$\text{ADE} = \frac{1}{N} \sum_{i=1}^N \|\mathbf{p}_{\text{pred},i} - \mathbf{p}_{\text{true},i}\|_2 \quad (5)$$

Duration prediction accuracy employs Mean Absolute Error (MAE) to evaluate temporal estimation performance:

$$\text{Duration MAE} = \frac{1}{M} \sum_{j=1}^M |t_{\text{pred},j} - t_{\text{true},j}| \quad (6)$$

where N represents trajectory points, M denotes trajectory segments, $\mathbf{p}$ indicates position vectors, and t represents duration values.

5.2. Component-Level Performance Analysis

5.2.1. Global Duration Predictor Performance

The Global Duration Predictor demonstrates variable accuracy across different maneuver types (Figure 4), with performance characteristics reflecting both typical execution patterns and platform variability. Duration prediction achieves median MAE of 0.028 s (mean: 0.13 ± 0.42 s, $n = 313$) for takeoff, 0.102 s (mean: 0.19 ± 0.43 s, $n = 463$) for vertical_up, 0.204 s (mean: 0.32 ± 0.34 s, $n = 463$) for vertical_down, 0.216 s (mean: 0.39 ± 0.49 s, $n = 400$) for horizontal_straight, and 0.371 s (mean: 0.59 ± 0.57 s, $n = 300$) for horizontal_diagonal.

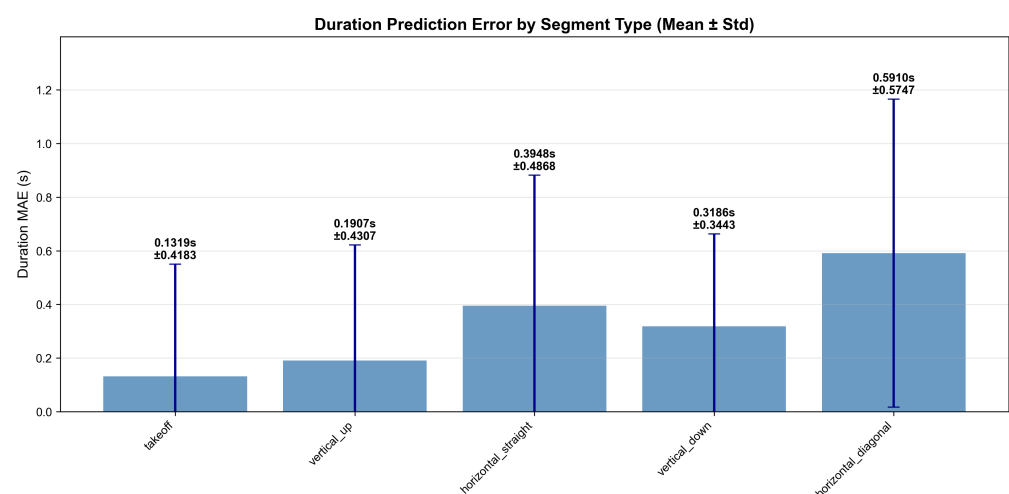


Figure 4. Duration Prediction Error by segment type.

The substantial differences between median and mean values indicate right-skewed error distributions, as illustrated in Figure 5 for representative segment types. Distribution analysis reveals that approximately 75–85% of predictions fall below the statistical outlier threshold ($Q3 + 1.5 \times IQR$), with the remaining 15–25% experiencing elevated errors that elevate mean values. For instance, takeoff segments (Figure 5a) show median MAE of 0.028 s representing typical performance, while the mean of 0.132 s reflects the influence of outliers beyond the 0.102 s threshold. This bimodal behavior—where most executions are consistent but a minority exhibit large deviations—reflects the inherent execution variability of the Crazyflie platform, including inconsistent hover durations at waypoints and variable flight speeds. For vertical trajectories, samples with excessive hovering were filtered during data preprocessing. However, for horizontal trajectories, such filtering was not feasible as abnormal hovering is difficult to distinguish from normal deceleration in the horizontal plane, so all collected samples were retained for training, resulting in higher variability in horizontal segment predictions.

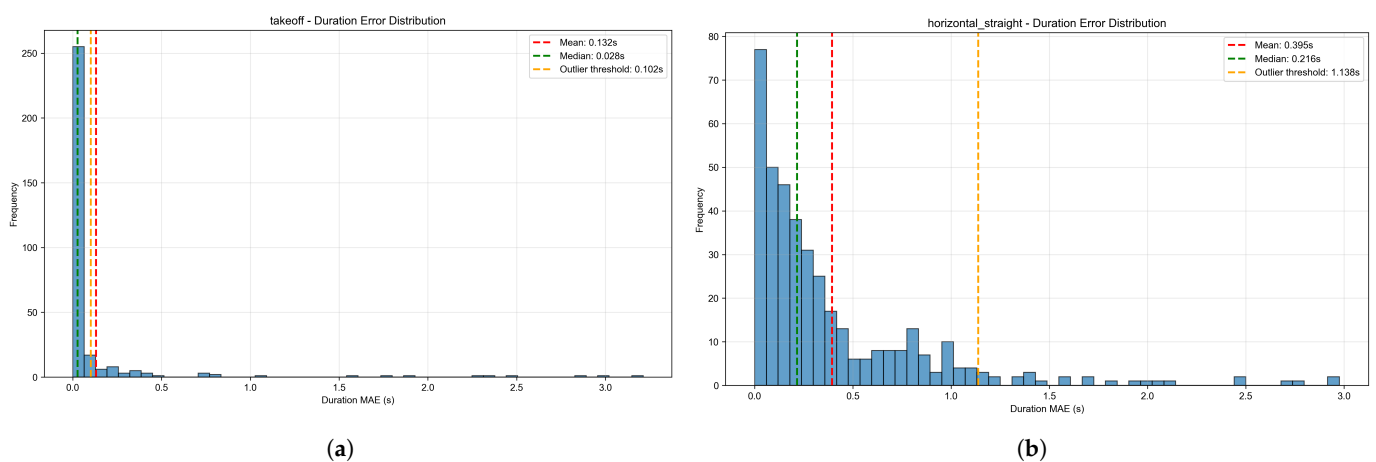


Figure 5. Duration prediction error distributions for representative segment types illustrating right-skewed patterns. The distributions demonstrate that while median MAE values (green dashed line) represent typical performance, mean values (red dashed line) are elevated by outlier errors beyond the $Q3 + 1.5 \times IQR$ threshold (orange dashed line). For takeoff segments, approximately 82% of predictions achieve errors below 0.102 s, with median MAE of 0.028 s indicating high typical accuracy despite mean MAE of 0.132 s reflecting outlier influence. Similarly, horizontal segments show 80% of predictions below 1.138 s threshold, with median MAE of 0.216 s versus mean of 0.395 s. This bimodal behavior reflects platform execution variability where most flight executions are consistent, but waypoint arrival detection inconsistencies produce occasional large timing deviations. (a) Takeoff Duration Error Distribution (Median: 0.028 s, Mean: 0.132 s). (b) Horizontal Straight Duration Error Distribution (Median: 0.216 s, Mean: 0.395 s).

Similarly, horizontal segments (Figure 5b) demonstrate median MAE of 0.216 s versus mean of 0.395 s, with outliers beyond 1.138 s accounting for approximately 20% of samples. These extended execution times occur when the UAV requires additional stabilization cycles to satisfy waypoint arrival criteria, a platform-specific characteristic independent of the prediction model.

The temporal variability observed across trajectory segments reflects inherent limitations in small-scale quadrotor flight execution consistency. Despite identical geometric constraints and control commands, the Crazyflie platform exhibits variable acceleration patterns, maximum velocities, and deceleration behaviors. This variation stems from multiple factors including battery voltage fluctuations, sensor noise, environmental disturbances, PID controller response variations, and most importantly, inconsistent waypoint arrival detection times. The stabilization verification process requires sustained positioning

accuracy within tolerance bounds, but the time needed to achieve this stability varies across identical waypoints due to platform-specific factors. This hardware-level execution variance explains why segments with similar spatial characteristics can exhibit substantial temporal differences.

Despite these platform-specific limitations, the proposed model achieves reasonable predictive accuracy for mission planning purposes. The median values represent typical flight behavior, while mean values account for platform-specific delays, providing users with both optimistic (median) and conservative (mean) timing estimates.

Future work could explore performance improvements under standardized flight conditions, where consistent flight behaviors might enable more precise temporal predictions for autonomous trajectory planning applications.

5.2.2. Segment-Specific Trajectory Generator Performance

Segment-specific trajectory generators demonstrate strong reconstruction accuracy across all maneuver types (Figure 6). Horizontal trajectory training excludes small patterns to ensure feature quality under LPS positioning constraints, while vertical trajectories include all pattern scales.

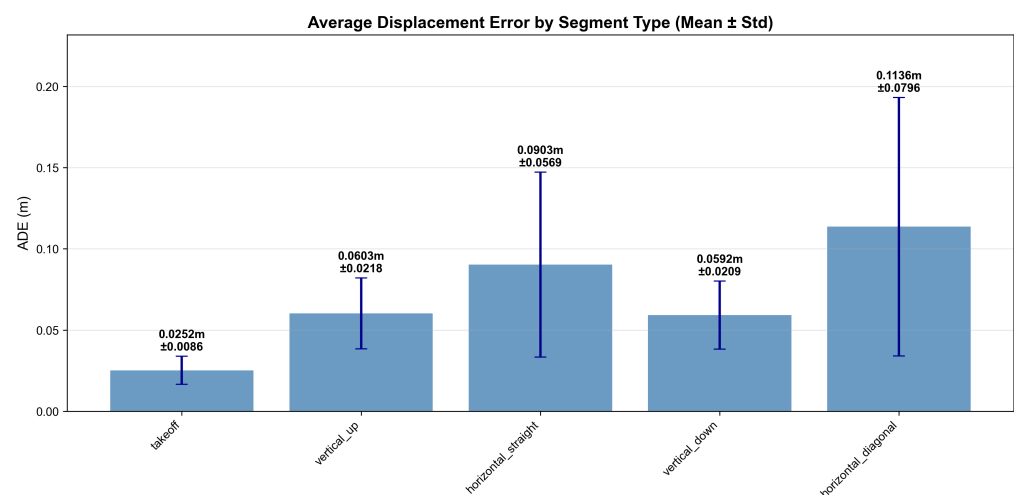


Figure 6. Average Displacement Error by segment type.

Takeoff sequences achieve exceptional precision with mean ADE of 0.0252 ± 0.0086 m ($n = 313$), reflecting the controlled vertical motion during initial stabilization. Vertical maneuvers maintain strong performance with mean ADE of 0.0603 ± 0.0218 m for vertical_up ($n = 463$, 30–50 cm segments) and 0.0592 ± 0.0209 m for vertical_down ($n = 463$), representing 12–20% relative error well within the LPS positioning baseline (± 10 cm).

Horizontal trajectory reconstruction demonstrates mean ADE of 0.0903 ± 0.0569 m for straight movements ($n = 400$, 50–70 cm segments, medium and large patterns only) and 0.1136 ± 0.0796 m for diagonal patterns ($n = 300$, 56–72 cm segments, medium and large patterns only), corresponding to 13–18% and 16–20% relative error respectively. These results approach the LPS positioning system's inherent accuracy (± 10 cm), indicating that spatial prediction errors are primarily sensor-bound rather than model-limited. The elevated standard deviations in horizontal segments (63–70% coefficient of variation) reflect two factors: (1) accumulated positioning uncertainty in the horizontal plane where LPS noise is more pronounced over longer flight distances, and (2) endpoint hovering phases where position drift contributes to ADE measurements.

All segment types achieve mean ADE at or below the 11 cm threshold, demonstrating effective learning of flight dynamics within the constraints of the positioning system accuracy.

5.3. End-to-End Trajectory Prediction Validation

5.3.1. Vertical Flight Pattern Analysis

Vertical trajectory prediction demonstrates strong performance across different scales, with the system effectively capturing ascending and descending flight dynamics. Representative prediction results for vertical missions are presented in Figure 7.

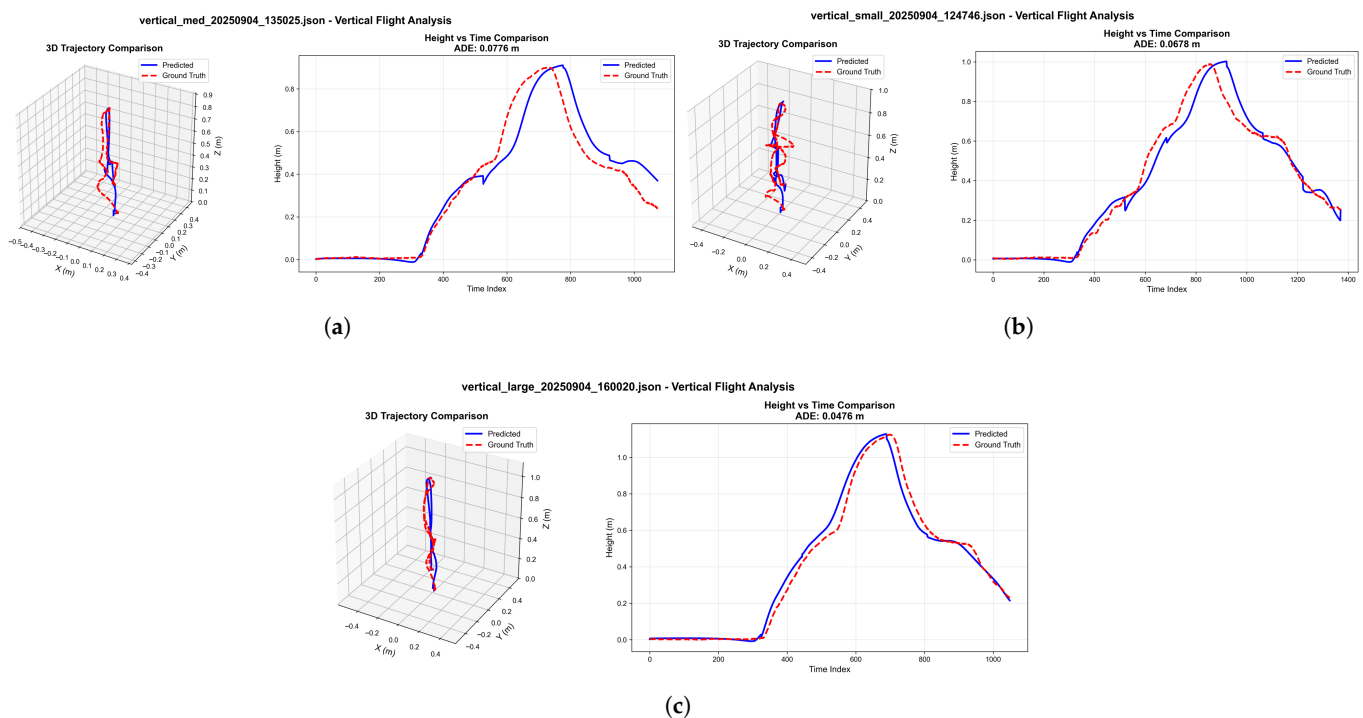


Figure 7. Vertical Flight Pattern Prediction Results. Left panels show 3D trajectory comparisons, right panels display height versus time profiles eliminating perspective distortion. (a) Vertical Medium Trajectory (ADE: 0.0776 m). (b) Vertical Small Trajectory (ADE: 0.0678 m). (c) Vertical Large Trajectory (ADE: 0.0476 m).

The vertical flight analysis reveals consistent prediction accuracy across trajectory scales, with ADE values ranging from 0.0476 m to 0.0776 m. The system demonstrates particular strength in modeling altitude changes, accurately reconstructing the vertical displacement patterns during ascent and descent phases. Height versus time profiles show strong correlation between predicted and ground truth trajectories, indicating effective integration of temporal and spatial prediction components.

5.3.2. Horizontal Flight Pattern Analysis

Horizontal trajectory prediction exhibits greater complexity due to the geometric constraints of maintaining specific path shapes while accommodating UAV dynamics. Representative results for square and triangular flight patterns are shown in Figure 8.

Horizontal trajectory analysis demonstrates the system's capability to maintain geometric path characteristics while accommodating UAV flight dynamics. Square patterns achieve ADE values ranging from 0.0673 m to 0.1607 m, with larger trajectories generally exhibiting better prediction accuracy due to reduced relative impact of positioning errors. Triangular patterns show ADE values from 0.0868 m to 0.1131 m, indicating consistent performance across different geometric configurations.

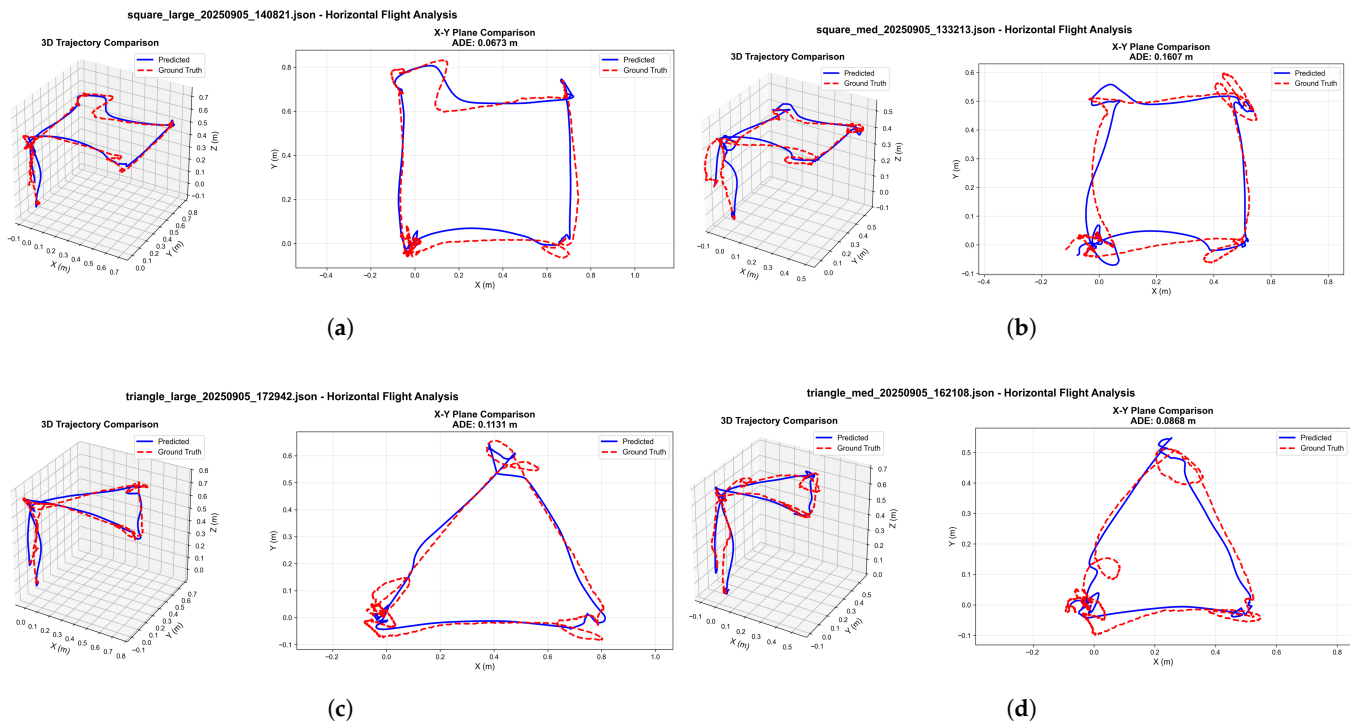


Figure 8. Horizontal Flight Pattern Prediction Results. Left panels show 3D trajectory comparisons, right panels display X-Y plane projections eliminating perspective distortion. (a) Square Large Trajectory (ADE: 0.0673 m). (b) Square Medium Trajectory (ADE: 0.1607 m). (c) Triangle Large Trajectory (ADE: 0.1131 m). (d) Triangle Medium Trajectory (ADE: 0.0868 m).

6. Discussion and Threats

6.1. Segmented Architecture Effectiveness

The two-stage segmented LSTM architecture demonstrates clear advantages for UAV trajectory prediction. By decomposing complex spatiotemporal prediction into duration estimation and segment-specific generation, the system achieves specialized modeling for different flight maneuvers. The Global Duration Predictor successfully captures temporal patterns across segment types, while dedicated trajectory generators effectively learn the unique spatial characteristics of each maneuver category.

The segmented approach proves valuable for understanding model reasoning capabilities across different maneuver types. Under the limitation of the system positioning accuracy of Crazyflie LPS, the take-off sequence achieved best reconstruction accuracy, while the vertical and horizontal maneuvers maintained good performance despite the increased geometric complexity.

6.2. Performance Relative to System Constraints

The spatial prediction results (2.5–11.4 cm reconstruction errors) demonstrate effective learning within the LPS positioning accuracy baseline (± 10 cm). This performance indicates the model successfully captures essential flight dynamics while accommodating measurement noise. The model achieves sub-positioning-system accuracy for takeoff and vertical segments (2.5–6.0 cm), and near-system-accuracy for horizontal segments (9.0–11.4 cm), with narrow standard deviations (± 0.9 –8.0 cm) confirming consistent spatial reconstruction.

Vertical trajectories include all pattern scales (30–50 cm segments), achieving 12–20% relative error well within the LPS baseline. For horizontal segments, small patterns were excluded from training to ensure trajectory feature quality under positioning constraints. The retained medium and large patterns (straight: 50–70 cm, diagonal: 56–72 cm) achieve

mean ADE of 9–11 cm, with relative errors of 13–18% for straight segments and 16–20% for diagonal segments respectively, which is reasonable given the LPS positioning baseline (± 10 cm, or ± 14 –20% for 50–70 cm flights).

In contrast, temporal prediction exhibits substantially higher uncertainty (std: ± 0.34 – 0.57 s) compared to spatial prediction. This disparity arises from fundamental differences in prediction targets: spatial trajectories depend primarily on geometric constraints (start/end positions) which remain consistent across executions, while temporal prediction must account for platform-dependent factors—battery state, control system response, and waypoint arrival criteria—all of which exhibit high variability in micro-UAV platforms. The large temporal standard deviations quantify this execution inconsistency: identical mission commands can result in 0.3–0.6 s duration variations depending on flight conditions. While this limits precise temporal scheduling applications, the duration predictor still provides valuable approximate timing for mission planning scenarios.

6.3. Practical Application Potential

With a total size of only 5.98 MB (0.124 MB duration predictor + 5×1.17 MB trajectory generators), the model is lightweight enough for potential deployment in edge computing environments and directly onboard UAV platforms, enabling real-time trajectory planning applications.

Our “high accuracy” refers to achieving physically plausible trajectories within system constraints (ADE: 0.0252–0.1136 m approaching ± 10 cm LPS positioning accuracy), demonstrating effective learning of flight dynamics rather than claiming superiority over existing methods. Our “lightweight” claim is based on absolute model size suitable for edge deployment, rather than empirically measured deployment performance. Actual edge device profiling (inference latency, memory usage, energy consumption) remains future work.

The system’s dual temporal-spatial prediction capabilities make it suitable for multi-agent coordination scenarios requiring accurate timing and path prediction for collision avoidance and mission synchronization. This segmentation method enables the model to naturally extend to swarm robot applications, where agents can focus on different segmentation types while sharing the prediction results. The physics-informed trajectory generation provides a foundation for autonomous navigation systems requiring predictive path planning from high-level waypoint commands.

6.4. Hardware Platform Limitations

As demonstrated in our analysis, despite the same geometric constraints and control commands, the crazyfly platform exhibits variable acceleration modes, maximum velocities, and deceleration behaviors. Possible causes include battery voltage fluctuations, sensor noise, environmental interference, changes in PID controller response, and inconsistent waypoint arrival detection times. The stability verification process needs to be maintained within the waypoint arrival determination conditions, but due to platform-specific factors, the time required to achieve such stability may vary at the same waypoint. These are the fundamental reasons for the significant differences in duration in the figure chart.

6.5. Controlled Environment Limitations

Data collection in a controlled indoor environment may not capture environmental disturbances encountered in real-world operations. Wind effects, temperature variations, and electromagnetic interference typical of outdoor operations were not systematically evaluated, potentially overestimating model robustness.

6.6. Trajectory Type and Scale Generalization Limitation

The training process only uses a limited number of trajectory segment types and lengths. When the model encounters unseen trajectory patterns or exceeds the scale of the training distribution, it may lead to a decrease in prediction accuracy or even prediction failure. The current framework is confined to five basic segment types (take-off, vertical up and down, horizontal straight line/diagonal) within a specific distance range, which limits its ability to handle trajectory sequences of different lengths from the training dataset and different flight actions.

6.7. UAV Platform Generalizability

The current model is trained on a specific UAV (Crazyflie 2.1) and is therefore not directly transferable to platforms with different aerodynamic and inertial properties. However, the underlying training methodology is general and can be applied to other UAV platforms with appropriate data.

6.8. Evaluation Metric Limitations

This model does not impose any restrictions on the accessibility of the target. For instance, whether the remaining battery power of the current drone can support the drone to eventually reach the target.

7. Conclusions and Future Work

This research addresses limitations in existing approaches. We designed a segmented LSTM architecture for complete flight trajectory prediction. The framework breaks down complex missions into five maneuver types. This allows complete spatiotemporal trajectory prediction for full missions.

We validated our approach using real Crazyflie 2.1 flight data. The results show three main contributions. The segmented LSTM architecture achieves high prediction accuracy. Average Displacement Error ranges from 0.0252 m for takeoff to 0.1136 m for diagonal trajectories. Each segment is effectively specialized for its maneuver type. Our two-stage prediction process combines global duration estimation with segment-specific trajectory generation. This captures both temporal and spatial flight characteristics. Existing short-term methods cannot meet these complete flight trajectory planning requirements. The model architecture is compact at 5.98 MB total (0.124 MB duration predictor + 5×1.17 MB trajectory generators). This size supports deployment in edge computing or onboard environments.

The training framework uses physics-informed loss functions. We add three constraint terms: velocity consistency between adjacent trajectory points, boundary conditions matching the start/end states, and acceleration smoothness penalizing sharp changes. These soft constraints guide the LSTM to generate realistic trajectories without adding computational overhead during inference. The model achieves reconstruction errors within baseline LPS accuracy (± 10 cm). This shows the model learned essential flight dynamics despite measurement noise.

Within the scope of five fundamental maneuver types, the system has practical applications for structured UAV operations. Smart agriculture can use it for autonomous crop monitoring missions with precise timing and coverage predictions. Logistics applications benefit from automated delivery systems with reliable arrival estimates. The lightweight modular architecture (1.17 MB per segment) enables distributed deployment, providing a foundation for multi-UAV coordination in structured scenarios where agents execute similar basic maneuvers.

However, we must acknowledge several limitations. The model is trained on Crazyflie 2.1 data. It needs retraining for other UAV architectures. Our controlled indoor environment may not capture outdoor complexity. Wind disturbances and electromagnetic interference are absent from our tests. The framework currently handles five fundamental maneuvers within specific distance ranges. This limits applicability to missions requiring maneuver types outside these five primitives, or trajectories exceeding the training distance range (0.0055–0.8308 m per segment). Dynamic replanning in response to unexpected obstacles or environmental changes also remains beyond current scope.

Future research directions include extending the segmented modeling approach to incorporate weather-adaptive prediction capabilities and more granular trajectory decomposition to support longer missions (e.g., decomposing horizontal and vertical movements into acceleration, uniform, and deceleration phases for scalable trajectory prediction). Such extensions could expand the maneuver library from five basic types to more refined segments, enabling representation of substantially more complex missions through flexible composition. Further avenues include exploring transfer learning mechanisms for rapid adaptation to different UAV platforms and integrating real-time environmental perception (e.g., computer vision) for dynamic trajectory adjustment to enhance dynamic flight planning capabilities. A critical future direction involves systematic baseline comparisons by adapting existing trajectory prediction methods (Transformer-based, GAN, unified LSTM) for waypoint-based input, establishing standardized benchmarks, and conducting ablation studies to quantify segmentation benefits. Empirical edge deployment validation on resource-constrained platforms with measured inference latency and energy profiling would further strengthen practical deployment claims.

While our current implementation addresses complete flight trajectory prediction from predetermined waypoints using five fundamental maneuver types, the segmented architecture provides a scalable foundation for future mission-level autonomous systems. By demonstrating that segmented approaches can achieve both high prediction accuracy (ADE: 0.0252–0.1136 m) and computational efficiency (5.98 MB total size), we establish the feasibility of this architectural paradigm. The key insight is that as the maneuver library expands—for example, decomposing horizontal flight into acceleration, cruise, and deceleration phases, or adding curved transition segments—the framework can generate increasingly complex trajectories through flexible composition of atomic prediction modules. In this vision, any flight mission decomposable into a sequence of learned maneuver primitives becomes predictable through modular assembly, regardless of overall complexity. This positions our work as a foundational building block demonstrating the viability of segmented prediction architectures for future mission-level UAV planning systems, while acknowledging that realizing comprehensive mission-level autonomy requires expanding the maneuver library and integrating with higher-level planning capabilities beyond our current contribution.

This work connects data-driven methods with mission-level planning needs for autonomous flight. Our segmented LSTM framework maintains deployment efficiency and prediction accuracy. It offers a practical approach for trajectory prediction in intelligent UAV systems under task-level flight control.

Author Contributions: Conceptualization, D.J.; methodology, D.J. and J.K.; software, D.J.; validation, D.J., J.K. and X.L.; formal analysis, D.J.; investigation, D.J. and J.K.; resources, J.K. and X.L.; data curation, D.J. and J.K.; writing—original draft preparation, D.J.; writing—review and editing, J.K. and X.L.; visualization, D.J.; supervision, J.K. and X.L.; project administration, J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in GitHub: <https://github.com/JudsonJia/crazyfly-data-collection> (accessed on 16 December 2025).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Gupta, L.; Jain, R.; Vaszkun, G. Survey of important issues in UAV communication networks. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 1123–1152. [\[CrossRef\]](#)
2. Liu, Y.; Wang, H.; Fan, J.; Wu, J.; Wu, T. Control-oriented UAV highly feasible trajectory planning: A deep learning method. *Aerosp. Sci. Technol.* **2021**, *110*, 106435. [\[CrossRef\]](#)
3. Ladegourdie, M.; Kua, J. Performance analysis of opc ua for industrial interoperability towards industry 4.0. *IoT* **2022**, *3*, 507–525. [\[CrossRef\]](#)
4. Kua, J.; Loke, S.W.; Arora, C.; Fernando, N.; Ranaweera, C. Internet of things in space: A review of opportunities and challenges from satellite-aided computing to digitally-enhanced space living. *Sensors* **2021**, *21*, 8117. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Sankaranarayanan, V.N.; Damigos, G.; Seisa, A.S.; Satpute, S.G.; Lindgren, T.; Nikolakopoulos, G. PACED-5G: Predictive Autonomous Control using Edge for Drones over 5G. In Proceedings of the 2023 International Conference on Unmanned Aircraft Systems, ICUAS 2023, Warsaw, Poland, 6–9 June 2023; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2023; pp. 1155–1161. [\[CrossRef\]](#)
6. Zhang, Y.; Jia, Z.; Dong, C.; Liu, Y.; Zhang, L.; Wu, Q. Recurrent LSTM-based UAV Trajectory Prediction with ADS-B Information. In Proceedings of the Proceedings—IEEE Global Communications Conference, GLOBECOM, Rio de Janeiro, Brazil, 4–8 December 2022; pp. 6475–6480. [\[CrossRef\]](#)
7. Shu, P.; Chen, C.; Chen, B.; Su, K.; Chen, S.; Liu, H.; Huang, F. Trajectory prediction of UAV Based on LSTM. In Proceedings of the Proceedings—2021 2nd International Conference on Big Data and Artificial Intelligence and Software Engineering, ICBASE 2021, Zhuhai, China, 24–26 September 2021; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2021; pp. 448–451. [\[CrossRef\]](#)
8. Conte, C.; de Alteriis, G.; Moriello, R.S.L.; Accardo, D.; Rufino, G. Drone trajectory segmentation for real-time and adaptive time-of-flight prediction. *Drones* **2021**, *5*, 62. [\[CrossRef\]](#)
9. Shukla, P.; Shukla, S.; Singh, A.K. Trajectory-Prediction Techniques for Unmanned Aerial Vehicles (UAVs): A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2025**, *27*, 1867–1910. [\[CrossRef\]](#)
10. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [\[CrossRef\]](#) [\[PubMed\]](#)
11. Alahi, A.; Goel, K.; Ramanathan, V.; Robicquet, A.; Fei-Fei, L.; Savarese, S. Social LSTM: Human Trajectory Prediction in Crowded Spaces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 961–971. [\[CrossRef\]](#)
12. Peng, S.; Sun, S.; Zhao, G.; Yan, Q. A LSTM-Based Trajectory Prediction Method For Aerial Target. In Proceedings of the 2024 4th International Conference on Artificial Intelligence, Robotics, and Communication, ICAIRC 2024, Xiamen, China, 27–29 December 2024; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2024; pp. 47–52. [\[CrossRef\]](#)
13. Chen, S.; Chen, B.; Shu, P.; Wang, Z.; Chen, C. Real-time UAV Flight Path Prediction Using A Bi-directional Long Short-term Memory Network with Error Compensation. *J. Comput. Des. Eng.* **2022**, *10*, 16–35. [\[CrossRef\]](#)
14. Wang, Y.; Li, J.; Zou, P.; Guo, Z.; Li, W. Research on UAV Path Planning based on LSTM. In Proceedings of the International Conference on Integrated Intelligence and Communication Systems, ICIICS 2023, Kalaburagi, India, 24–25 November 2023; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2023. [\[CrossRef\]](#)
15. Palamas, A.; Souli, N.; Panayiotou, T.; Kolios, P.; Ellinas, G. A Multi-Task Learning Framework for Drone State Identification and Trajectory Prediction. In Proceedings of the Proceedings—19th International Conference on Distributed Computing in Smart Systems and the Internet of Things, DCOSS-IoT 2023, Pafos, Cyprus, 19–21 June 2023; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2023; pp. 676–683. [\[CrossRef\]](#)
16. Zhang, A.; Zhang, B.; Bi, W.; Mao, Z. Attention based trajectory prediction method under the air combat environment. *Appl. Intell.* **2022**, *52*, 17341–17355. [\[CrossRef\]](#)
17. Phisannupawong, T.; Damanik, J.J.; Choi, H.L. Aircraft Trajectory Segmentation-Based Contrastive Coding: A Framework for Self-Supervised Trajectory Representation. *IEEE Open J. Intell. Transp. Syst.* **2025**, *6*, 738–757. [\[CrossRef\]](#)
18. Xie, L.; Wei, Z.; Ding, D.; Zhang, Z.; Tang, A. Long and Short Term Maneuver Trajectory Prediction of UCAV Based on Deep Learning. *IEEE Access* **2021**, *9*, 32321–32340. [\[CrossRef\]](#)

19. Zhang, J.; Shi, Z.; Zhang, A.; Yang, Q.; Shi, G.; Wu, Y. UAV Trajectory Prediction Based on Flight State Recognition. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 2629–2641. [[CrossRef](#)]
20. Sun, Y.; Zhou, X.; Yang, Z.; Wang, W.; Shi, Q. Flight Trajectory Prediction Method Based on Attentional Graph Convolutional Network. In Proceedings of the 2023 3rd International Symposium on Computer Technology and Information Science, ISCTIS 2023, Chengdu, China, 16–18 June 2023; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2023; pp. 127–132. [[CrossRef](#)]
21. Wang, Y.; Chen, Y.Y.; Yu, R.; Liu, G.; Liu, T.; Wang, X. Cooperative Trajectory Prediction of UAVs via Generative Adversarial Networks. In Proceedings of the IECON Proceedings (Industrial Electronics Conference), Singapore, 16–19 October 2023; IEEE Computer Society: Washington, DC, USA, 2023. [[CrossRef](#)]
22. Ma, D.; Fu, X.; Huang, X.; Li, Q.; Zhu, X. LSTM-based UAV Swarm Trajectory Prediction and Intent Recognition. In Proceedings of the 2025 8th International Conference on Advanced Algorithms and Control Engineering, ICAACE 2025, Shanghai, China, 21–23 March 2025; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2025; pp. 676–681. [[CrossRef](#)]
23. Conti, F.C.; Santoro, C.; Santoro, F.F. Twinflie: A Digital Twin UAV Orchestrator and Simulator. In Proceedings of the 2023 IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC/PiCom/CBDDCom/CyberSciTech), Abu Dhabi, United Arab Emirates, 14–17 November 2023; pp. 0258–0263. [[CrossRef](#)]
24. Soliman, A.; Al-Ali, A.; Mohamed, A.; Gedawy, H.; Izham, D.; Bahri, M.; Erbad, A.; Guizani, M. AI-Based UAV Navigation Framework With Digital Twin Technology for Mobile Target Visitation. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106318. [[CrossRef](#)]
25. Xiao, H.; Zhang, Y.; Tang, C.; Zhang, L.; Lv, W.; Xu, S. Research on Cooperative Control of UAV Cluster Based on Digital Twin Technology. In Proceedings of the 2023 IEEE International Conference on Signal Processing, Communications and Computing, ICSPCC 2023, Zhengzhou, China, 14–17 November 2023; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2023. [[CrossRef](#)]
26. Yang, H.; Sun, S.; Xia, Y.; Li, P. Digital Twin-Based Obstacle Avoidance for Unmanned Aerial Vehicles Using Feedforward-Feedback Control. *IEEE Trans. Veh. Technol.* **2025**, *74*, 8721–8733. [[CrossRef](#)]
27. Jia, D.; Kua, J.; Liu, X. Performance Analysis of Network-Aware Micro-UAVs in Low-Altitude Applications. In Proceedings of the 2025 IEEE 50th Conference on Local Computer Networks (LCN), Sydney, Australia, 14–16 October 2025; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2025; pp. 1–7.
28. Kua, J.; Armitage, G.; Branch, P. A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1842–1866. [[CrossRef](#)]
29. Luu, T.; Nguyen, Q.; Tran, T.; Tran, M.; Ding, S.; Kua, J.; Hoang, T. Enhancing real-time robot teleoperation with immersive virtual reality in industrial IoT networks. *Int. J. Adv. Manuf. Technol.* **2025**, *139*, 6233–6257. [[CrossRef](#)]
30. Karam, K.; Mansour, A.; Khaldi, M.; Clement, B.; Ammad-Uddin, M. UAV Path Optimization for WSN in Smart Agriculture. *IEEE Access* **2025**, *13*, 87526–87544. [[CrossRef](#)]
31. Kong, F.; Li, J.; Jiang, B.; Wang, H.; Song, H. Trajectory Optimization for Drone Logistics Delivery via Attention-Based Pointer Network. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 4519–4531. [[CrossRef](#)]
32. Suhartono, A.; Sahal, M.; Wong, K.Y.; Mardiyanto, R. Proposed Cargo Drone Model Respect to Actual Trajectory Tracking. In Proceedings of the ICoCSETI 2025—International Conference on Computer Sciences, Engineering, and Technology Innovation, Proceeding, Dubai, United Arab Emirates, 29–30 November 2025; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2025; pp. 318–323. [[CrossRef](#)]
33. Bitcraze. Crazyradio. Product Specification. 2024. Available online: <https://www.bitcraze.io/products/crazyradio-pa/> (accessed on 29 October 2025).
34. Bitcraze. Datasheet Crazyflie 2.1—Rev 3. Technical Datasheet. 2021. Available online: https://www.bitcraze.io/documentation/hardware/crazyflie_2_1/crazyflie_2_1-datasheet.pdf (accessed on 29 October 2025).
35. Bitcraze. Loco Positioning Node. Product Specification. 2024. Available online: <https://www.bitcraze.io/products/loco-positioning-node/> (accessed on 29 October 2025).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.