DiffMoE: Differential-Encoded MoE Predictor for Variable-Length Flight Trajectory Prediction

Lindong Wang, Hongya Tuo, Zhongliang Jing*
School of Aeronautics and Astronautics
Shanghai Jiao Tong University
Shanghai, China

***Abstract*—Flight trajectory prediction (FTP) is a essential task for future air traffic management (ATM), enabling conflict detection and traffic flow control. Unlike fixed-window sliding methods that require sequence truncation or padding, variable-length prediction preserves complete historical information from takeoff to landing, reducing information loss and improving accuracy. However, variable-length prediction must balance accuracy across both short sequences (takeoff phase) and long sequences (landing phase). Additionally, dynamic transitions among flight modes, including climbing, turning, cruising, and descending, further complicate the prediction task. To address these challenges, we propose DiffMoE, a framework incorporating differential normalization and encoding to convert absolute coordinates into relative displacements, effectively handling scale variations in variable-length sequences. We further introduce an MoE regression head with Top-K routing for multi-expert soft-threshold learning, enabling adaptive handling of flight mode transitions. Experiments on a real-world ADS-B dataset demonstrate that DiffMoE outperforms all comparison methods and achieves distance errors as low as 154 meters. Ablation studies further confirm the effectiveness of both differential encoding and MoE modules.**

***Index Terms*—trajectory prediction, mixture of experts, differential encoding, ADS-B, deep learning**

I. Introduction

With the rapid growth of global air traffic, Air Traffic Management (ATM) systems face unprecedented challenges in airspace capacity, flight safety, and operational efficiency. Flight trajectory prediction, as a core technology in ATM, plays a critical role in conflict detection, traffic flow control, and resource optimization. The widespread deployment of Automatic Dependent Surveillance-Broadcast (ADS-B) technology [1] has enabled the acquisition of massive high-precision trajectory data, providing new opportunities for data-driven prediction methods. Deep learning techniques, particularly Recurrent Neural Networks (RNNs), Transformers, and their variants, have demonstrated remarkable performance in time-series prediction and are becoming mainstream approaches in trajectory forecasting.

Most existing trajectory prediction methods adopt a fixed-length sliding window mechanism, using a fixed number of historical trajectory points to predict future positions. However, this approach has two fundamental limitations: (1) Information truncation: When historical trajectories exceed the window size, early information is discarded; when trajectories are shorter, zero-padding introduces invalid data. (2) Sensitivity to signal missing: Extended ADS-B signal outages leave the sliding window empty, leading to unreliable predictions.

These limitations motivate the study of variable-length trajectory prediction, which presents two key challenges: (1) Inconsistent features across sequence lengths: Short sequences lack sufficient information, while long sequences respond slowly to mode transitions; additionally, normalization strategies are difficult to unify across different lengths. (2) Mode transition problem: Prediction accuracy often degrades during flight mode transitions (e.g., from climbing to cruising), requiring models that can dynamically adapt to various motion patterns.

Existing methods for variable-length sequences have notable limitations. LSTM and GRU [3], while theoretically capable of handling variable lengths, suffer from gradient vanishing in long sequences. Transformer-based methods [4], [5] capture long-range dependencies but incur $O(n^2)$ computational complexity for long sequences. Causal CNN methods typically require fixed-length inputs, and padding for variable lengths introduces invalid information and reduces efficiency. More critically, all these methods employ a single prediction head, failing to design differentiated strategies for different flight modes (climbing, cruising, descending), thus struggling to capture the complex dynamics during mode transitions.

To address these challenges, we propose DiffMoE (Differential-Encoded Mixture of Experts), a deep learning framework specifically designed for variable-length trajectory prediction.

We observe that trajectory data exhibits low-frequency dominant, slowly-varying characteristics: aircraft typically undergo uniform climbing or descending, with velocity change rates close to zero or constant. Based on this observation, we design a differential encoding strategy that converts absolute coordinates to relative displacements between adjacent frames. This design significantly reduces learning difficulty while making the model more sensitive to velocity changes, thereby improving the capture of flight mode transitions.

To handle dynamic mode transitions, we introduce a Mixture of Experts (MoE) regression head. Through Top-K routing, different experts specialize in different mode features, and soft-threshold weighting enables smooth mode transitions

*Corresponding author: Zhongliang Jing (zljing@sjtu.edu.cn).

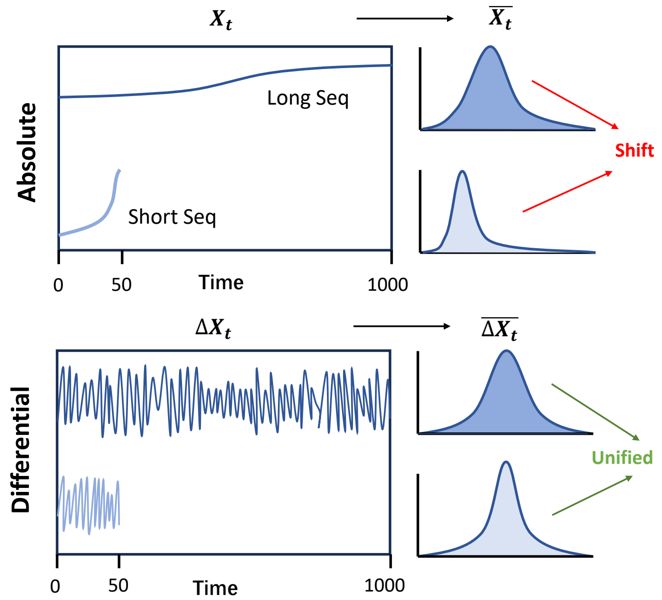

Fig. 1. Comparison of absolute-domain and differential-domain normalization for variable-length trajectory prediction. Absolute-domain methods (Global/Local Z-Score, Min-Max) are hard to adapt to all varying sequence lengths. In contrast, differential-domain normalization exploits the slowly-varying nature of trajectories, resulting in more consistent normalization across variable-length sequences.

without prediction discontinuities.

DiffMoE adopts a concise cascaded architecture: Differential Encoding $\rightarrow$ Backbone Feature Extraction $\rightarrow$ MoE Head Prediction. To validate the effectiveness of our proposed modules, we intentionally adopt a simple overall structure that can flexibly accommodate various backbones. Our design goal is to provide general-purpose building blocks for variable-length trajectory prediction. Experimental results demonstrate that our modules can improve the prediction performance of various methods on this task.

Our main contributions are summarized as follows:

- We propose differential normalization and encoding, a strategy adapted to variable-length sequences that converts absolute coordinates into relative displacements, reducing prediction difficulty and facilitating mode feature learning.
- We introduce an MoE residual prediction head that achieves smooth dynamic mode switching through soft-threshold weighting, enabling faster response to flight mode transitions.
- We conduct experiments on a real-world ADS-B dataset. DiffMoE outperforms various baseline methods and achieves distance prediction errors as low as 154 meters. Ablation studies further validate the effectiveness of each component.

II. Related Work

A. Time Series Forecasting Models

Time series forecasting has been extensively studied in deep learning. Long Short-Term Memory (LSTM) networks [3] effectively mitigate the gradient vanishing problem of traditional RNNs through gating mechanisms, becoming a classical approach for sequential modeling. Transformer [4] introduces the self-attention mechanism to enable parallelized modeling of long-range dependencies, though its $O(n^2)$ computational complexity limits applications to long sequences. To address this, Informer [5] proposes the ProbSparse self-attention mechanism, significantly reducing computational overhead.

Recently, state space models have attracted considerable attention: Mamba [7] achieves linear-time sequence modeling through selective state spaces; RWKV-TS [8] combines the efficiency of RNNs with the expressive power of Transformers. Additionally, SegRNN [9] proposes a segment-wise recurrent strategy to reduce iterations for long sequences, while TimeFilter [10] captures patch-level spatial-temporal correlations through graph filtration. However, these general-purpose time series models are not specifically optimized for the characteristics of trajectory data, such as low-frequency slowly-varying dynamics and flight mode transitions.

B. Flight Trajectory Prediction

Deep learning methods have achieved significant progress in flight trajectory prediction. Shi et al. [11] first applied LSTM to trajectory prediction and further proposed 4D-CLSTM [12] incorporating flight dynamics constraints. Wu et al. [13] employed generative adversarial networks for long-term trajectory generation. FlightBERT [14] introduced binary encoding and pre-training paradigms to enhance feature representation. Shafienya et al. [15] proposed a CNN-LSTM hybrid architecture for ADS-B trajectory data. Zhang et al. [16] published a time-frequency wavelet transform method in Nature Communications. FT-TF [17] applied Transformer to 4D long-term prediction.

However, existing methods predominantly adopt fixed-length sliding window mechanisms, suffering from information truncation or zero-padding issues. Moreover, most employ a single prediction head, struggling to adaptively handle dynamic transitions among different flight modes such as climbing, cruising, and descending. The proposed DiffMoE framework addresses these challenges through differential encoding and mixture-of-experts mechanisms.

III. Methodology

By analyzing the collected real-world ADS-B data, we identify the following characteristics of flight trajectory time series: (1) sequence lengths vary significantly with flight progress (20-2000 frames); (2) data presents low-frequency, slowly-varying temporal dynamics; (3) trajectories undergo dynamic transitions among multiple flight modes, including climbing, cruising, descending, and turning. Based on these characteristics, our design employs three strategies:

- **Differential normalization** addresses scale unification for variable-length sequences by converting absolute coordinates into relative sequences using frame-to-frame increment normalization.

- **Differential encoding** extracts local temporal patterns from increment sequences through causal convolution, reducing prediction complexity.
- **MoE residual prediction head** adaptively handles different flight modes through multi-expert soft routing and outputs predictions in residual form.

DiffMoE adopts a simple modular cascaded architecture, which means our method can be easily adapted to other FTP models. We conduct extensive ablation experiments to demonstrate the effectiveness of our proposed modules.

A. Problem Formulation

The goal of variable-length trajectory prediction is to predict the future state of an aircraft based on historical observations. Given an input sequence $\mathbf{X}_{1:t} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t\}$ (abbreviated as $\mathbf{X}_t$ hereafter), where $\mathbf{x}_i \in \mathbb{R}^d$ represents the trajectory state vector at time step i, containing position and velocity information (for $d = 6$: latitude, longitude, altitude, speed, vertical speed, heading). The sequence length t varies significantly with flight progress (20-2000 frames).

The model aims to learn a mapping function $\mathcal{F}$ that predicts the next state:

$$\hat{\mathbf{x}}_{t+1} = \mathcal{F}(\mathbf{X}_t; \theta) \tag{1}$$

where θ denotes the model parameters.

B. Overall Framework

The DiffMoE architecture consists of three core modules in cascade: differential normalization and encoding, feature extraction backbone, and MoE residual prediction head, as illustrated in Fig. 2.

Input sequences $\mathbf{X}_t$ first undergo the **differential normalization and encoding** module, which converts absolute coordinates into frame-to-frame displacement increments $\Delta\mathbf{X}$, scales them with pre-computed increment standard deviation $\boldsymbol{\sigma}_\Delta$ to obtain $\bar{\Delta}\mathbf{X}$, and extracts local temporal patterns via single-scale causal convolution to produce encoded features $\mathbf{F}_{embed}$. The encoded sequence is then fed into the **backbone network** (replaceable with LSTM, Mamba, etc.) to extract temporal features, outputting the hidden representation $\mathbf{H}_t$ corresponding to the last valid frame $\mathbf{x}_t$. Finally, the **MoE residual prediction head** activates multiple expert networks through Top-K routing, fuses expert outputs via weighted summation to predict the normalized increment $\bar{\Delta}\hat{\mathbf{x}}_{t+1}$, which is then denormalized and added to $\mathbf{x}_t$ to obtain the final prediction $\hat{\mathbf{x}}_{t+1}$.

C. Differential Normalization and Encoding

Variable-length sequence normalization faces unique challenges. We analyze the limitations of common normalization methods:

- **Global Z-Score**: Normalizes using mean and standard deviation from the entire training set. However, absolute coordinate ranges differ drastically across flights (transcontinental vs. regional), making global statistics unsuitable for all samples.

Algorithm 1 Differential Normalization and Denormalization

Require: Input sequence $\mathbf{X} \in \mathbb{R}^{t \times d}$, Global increment std $\boldsymbol{\sigma}_\Delta \in \mathbb{R}^d$
Ensure: Normalized increment sequence $\bar{\boldsymbol{\Delta}}\mathbf{X}$, Reference $\mathbf{x}_t$
// Normalization
$\Delta\mathbf{x}_1 \leftarrow \mathbf{0}$
for $i = 2$ to t **do**
  $\Delta\mathbf{x}_i \leftarrow \mathbf{x}_i - \mathbf{x}_{i-1}$
end for
$\bar{\Delta}\mathbf{x}_i \leftarrow \Delta\mathbf{x}_i / \boldsymbol{\sigma}_\Delta$ for all i
return $\bar{\boldsymbol{\Delta}}\mathbf{X}$, $\mathbf{x}_t$
// Denormalization
$\Delta\hat{\mathbf{x}}_{t+1} \leftarrow \bar{\Delta}\hat{\mathbf{x}}_{t+1} \cdot \boldsymbol{\sigma}_\Delta$

- **Local Z-Score**: Normalizes using current sequence statistics. For short sequences (e.g., 20 frames) that sometimes exhibit minimal or no variation, the standard deviation becomes very small, easily causing numerical explosion after normalization.
- **Min-Max Normalization**: Scales data to [0,1] range, but for slowly-varying or stationary short sequences, numerical explosion similarly occurs, and is sensitive to outliers.

To address these issues, we propose **differential normalization**. The core idea is to convert absolute coordinates to frame-to-frame increments, then scale using global increment statistics. Since frame-to-frame increment magnitudes are independent of sequence length, the statistics are more stable.

After differential normalization, the model's learning objective becomes:

$$\bar{\Delta}\hat{\mathbf{x}}_{t+1} = \mathcal{F}(\bar{\boldsymbol{\Delta}}\mathbf{X}; \theta) \tag{2}$$

where the input is the normalized increment sequence $\bar{\boldsymbol{\Delta}}\mathbf{X}$ instead of the original absolute coordinates. The final prediction $\hat{\mathbf{x}}_{t+1}$ is recovered from the normalized space via:

$$\hat{\mathbf{x}}_{t+1} = \mathbf{x}_t + \bar{\Delta}\hat{\mathbf{x}}_{t+1} \cdot \boldsymbol{\sigma}_\Delta \tag{3}$$

This transformation shifts the model's learning space from absolute coordinates space $\mathbf{X}$ to the differential space $\bar{\boldsymbol{\Delta}}\mathbf{X}$, which is more stable across varying sequence lengths and easier to learn.

Differential normalization offers several advantages: (1) increment representations have small and stable numerical ranges; (2) velocity changes are more prominent in the differential domain; (3) the incremental form naturally corresponds to the residual prediction paradigm. Fig. 3 demonstrates that differential normalization effectively reveals hidden periodic dependencies in ADS-B sequences, facilitating network learning.

The differential encoding step then extracts features from the normalized increment sequence using single-scale causal convolution:

$$\mathbf{F}_{conv} = \text{Conv1D}(\bar{\Delta}\mathbf{X}, k) \tag{4}$$

where k is the kernel size (default k=3). Causal convolution only uses information from current and previous time steps,

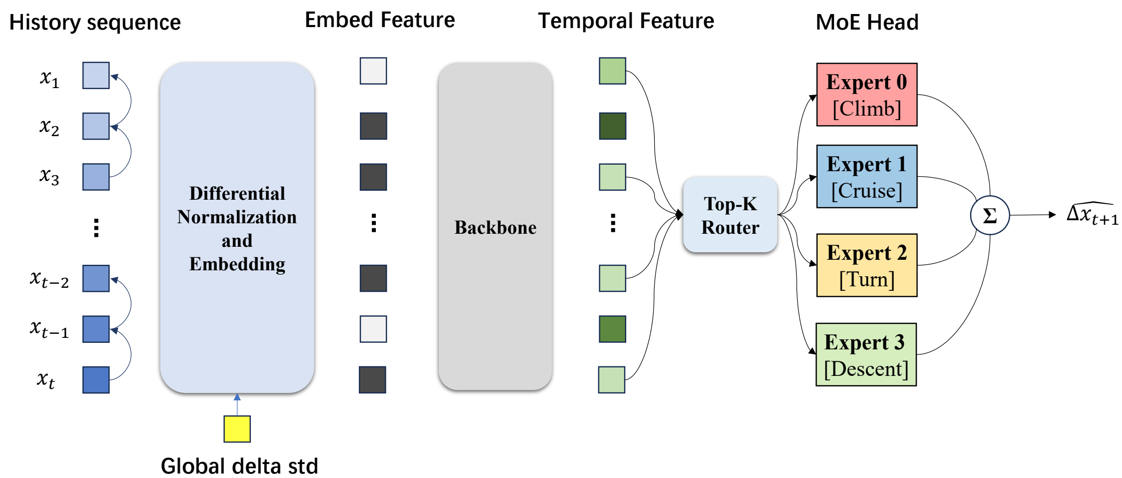

Fig. 2. Overall architecture of DiffMoE. The framework consists of three core modules: (1) Differential Normalization and Encoding module converts absolute coordinates to normalized increments and extracts local temporal patterns via causal convolution; (2) Feature Extraction Backbone captures temporal dependencies; (3) MoE Residual Prediction Head uses Top-K routing to activate multiple experts for adaptive flight mode handling.

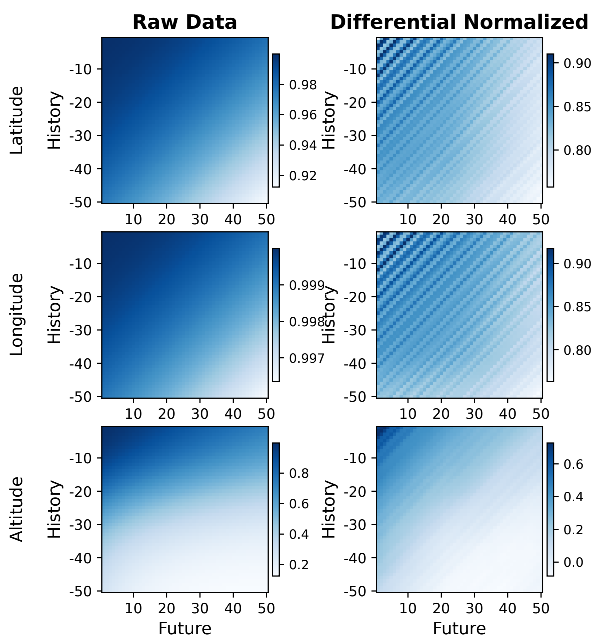

Fig. 3. Temporal correlation comparison before and after differential normalization. Differential normalization reveals periodic correlation patterns in trajectory data, facilitating network learning.

maintaining temporal causality and capturing short-term motion patterns (acceleration, deceleration, turning).

D. MoE Residual Prediction Head

Flight trajectories contain multiple motion modes (climbing, cruising, descending, turning, etc.) with distinct dynamic characteristics. To address this, we design the MoE residual prediction head, unifying the multi-expert mechanism with residual prediction in one module.

Expert Networks. Each expert is an MLP:

$$\text{Expert}_i(\mathbf{h}) = \mathbf{W}_2^{(i)} \cdot \text{GELU}(\mathbf{W}_1^{(i)} \cdot \mathbf{h} + \mathbf{b}_1^{(i)}) + \mathbf{b}_2^{(i)} \quad (5)$$

Top-K Router. The router dynamically selects K experts based on input features:

$$s_i = \mathbf{W}_{gate} \cdot \mathbf{h}, \quad g_i = \frac{\exp(s_i/\tau)}{\sum_{j \in \text{TopK}} \exp(s_j/\tau)} \quad (6)$$

Residual Output. Selected expert outputs are weighted and summed, then added to the last frame in incremental form:

$$\Delta\hat{\mathbf{y}} = \sum_{i=1}^{N} g_i \cdot \text{Expert}_i(\mathbf{h}), \quad \mathbf{x}_{t+1} = \mathbf{x}_{last} + \sigma(\alpha) \cdot \Delta\hat{\mathbf{y}} \quad (7)$$

where $\sigma(\cdot)$ is the Sigmoid function and α is a learnable parameter constraining the residual weight to $(0, 1)$.

Load Balancing Loss. To encourage uniform expert utilization:

$$\mathcal{L}_{balance} = N \cdot \sum_{i=1}^{N} f_i \cdot p_i, \quad \mathcal{L} = \mathcal{L}_{pred} + \lambda \cdot \mathcal{L}_{balance} \quad (8)$$

where λ=0.01 is the balancing coefficient.

IV. Experiments and Analysis

A. Dataset

The dataset used in this study is derived from real ADS-B data collected via the Flightradar24 platform [2], covering two major airports: Boston Logan International Airport (BOS) and Frankfurt Airport (EDDF). The data collection spans approximately 12 days, from October 20 to October 31, 2023, encompassing flight trajectories of all arriving and departing flights during this period. After data cleaning and preprocessing, a total of 22,374 valid trajectories were obtained. We employed random sampling to partition the data into training and test sets, with the training set containing 17,882

TABLE I
MODEL PERFORMANCE COMPARISON

No	Model	Year	Pub	EoD	LAT (°)		LON (°)		ALT (km)		SPEED (m/s)		V-SPEED (m/s)		HEADING (°)		Param	FPS
				(km)	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	(M)	
A1	LSTM	1997	NC	0.242	0.0013	0.0026	0.0019	0.0037	0.0289	0.0649	4.45	11.78	0.61	1.36	1.46	8.16	0.67	19.9
A2	Transformer	2017	NeurIPS	0.832	0.0040	0.0067	0.0075	0.0115	0.0182	0.0385	4.41	11.37	0.60	1.36	1.47	8.16	1.90	4.0
A3	Informer	2021	AAAI	0.623	0.0030	0.0055	0.0057	0.0098	0.0139	0.0322	3.95	11.03	0.61	1.36	1.46	8.16	2.09	3.0
A4	Koopa	2023	NeurIPS	2.143	0.0101	0.0141	0.0188	0.0254	0.0545	0.0808	5.26	14.63	0.70	1.46	2.73	8.64	**0.03**	45.5
A5	Mamba	2023	arXiv	0.893	0.0046	0.0073	0.0077	0.0118	0.0274	0.0484	4.42	11.58	0.62	1.36	2.50	8.40	0.12	21.9
A6	RWKV-TS	2024	arXiv	1.096	0.0058	0.0083	0.0091	0.0128	0.0266	0.0505	4.41	11.68	0.60	1.36	1.48	8.15	0.56	44.3
A7	TimeFilter	2025	ICML	2.118	0.0113	0.0161	0.0205	0.0277	0.0501	0.0790	5.87	13.37	0.90	1.59	3.61	9.28	0.60	8.5
A8	SegRNN	2025	IoTJ	0.349	0.0018	0.0036	0.0029	0.0058	0.0176	0.0357	4.42	10.89	0.69	1.46	1.73	8.06	0.40	**75.4**
A9	DiffMoE	2025	Ours	**0.154**	**0.0008**	**0.0019**	**0.0013**	**0.0027**	**0.0080**	**0.0235**	**3.42**	**10.11**	**0.46**	**1.09**	**1.38**	**7.54**	2.68	22.9

[1] For fair comparison, all models use a feature dimension of 256 with default hyperparameters.
[2] Inference speed (FPS) and parameter counts are measured on a single RTX 3090 GPU with input shape $[256, 700, 6]$ over 1000 recurrent predictions.

TABLE II
ABLATION STUDY ON NORMALIZATION AND CORE COMPONENTS

No	Norm Method	Diff Enc	MoE	EoD (km)	Params (M)	FPS
B1	Global Z-Score	✓	✓	4.164	2.68	**25.8**
B2	Local Z-Score	✓	✓	2.466	2.68	**25.8**
B3	Min-Max	✓	✓	0.866	2.68	22.1
B4	Diff Norm	×	✓	0.179	2.68	22.4
B5	Diff Norm	✓	×	0.164	**1.11**	22.7
A9	Diff Norm	✓	✓	**0.154**	2.68	22.9

TABLE III
ABLATION STUDY ON BACKBONE AND EMBEDDING LAYER

No	Backbone	Embedding	EoD (km)	Params (M)	FPS
B6	Mamba	Diff Embed	0.193	2.88	8.0
B7	LSTM	MLP	0.205	2.68	26.1
B8	LSTM	Linear	0.195	2.67	**27.6**
B9	LSTM	Multi Conv	0.170	2.68	25.6
B10	GRU	Diff Embed	0.164	**2.55**	24.1
A9	LSTM	Diff Embed	**0.154**	2.68	22.9

TABLE IV
PARAMETER STABILITY ANALYSIS

No	d_{model}	#Experts	Top-K	EoD (km)	Params (M)	FPS
C1	128	4	2	0.177	**0.68**	**31.8**
C2	512	4	2	**0.157**	10.67	5.3
C3	256	2	1	0.163	1.63	16.0
C4	256	6	3	0.159	3.74	15.8
C5	256	8	4	0.192	4.79	25.5
A9	256	4	2	0.154	2.68	22.9

trajectories (80%) and the test set containing 4,492 trajectories (20%).

During the preprocessing stage, all raw trajectory data underwent regular sampling and cubic spline interpolation, resulting in 6-dimensional feature vectors comprising longitude, latitude, altitude, ground speed, vertical speed, and heading. Regarding sequence length settings, training set sequences are constrained to 20-700 frames with a sampling rate of 0.2 to reduce computational overhead. The test set retains complete sequences ranging from 20 to 2000 frames, progressively sampled from takeoff to landing to simulate the full-flight inference prediction scenario during actual flight operations.

B. *Experimental Setup*

1) Training Configuration: All models are trained using the Adam optimizer with an initial learning rate of 1e-3 and a CosineAnnealingLR learning rate scheduling strategy. Training is conducted for 50 epochs with a batch size of 512. To ensure gradient stability during training, we employ gradient clipping with a maximum gradient norm of 1.0. Experiments are performed on a single NVIDIA RTX 3090 GPU (24GB memory), with each experimental run requiring approximately 6 hours of training time. Regarding loss functions, all models utilize Mean Squared Error (MSE) loss as the primary optimization objective; for the DiffMoE model, we additionally introduce a load balancing loss to ensure uniform utilization of experts within the MoE module.

2) Experimental Groups: The experimental design in this study is organized into three groups. The first group is the comparative experiment group (Group A), comprising 9 models: the proposed DiffMoE, traditional RNN methods (LSTM, SegRNN), state space models (Mamba, RWKV), Koopman operator method (Koopa), Transformer variants (Transformer, Informer, TimeFilter). The second group is the ablation study group (Group B), designed to validate the effectiveness of core components, including removal of differential encoding, removal of MoE prediction head, comparison of different normalization methods, and comparison of different backbone networks. The third group is the parameter stability experiment group (Group C), aimed at analyzing the scaling law of model performance with respect to key parameter variations, including embedding layer variants (multi-layer convolution, linear layer, MLP), MoE parameter settings (number of experts, Top-K values), and feature dimension d_{model} (128, 512) comparisons.

3) Evaluation Metrics: We employ multiple metrics for comprehensive model evaluation, including Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) for each dimension, as well as Euclidean distance error measured in meters, where the distance error serves as the primary metric

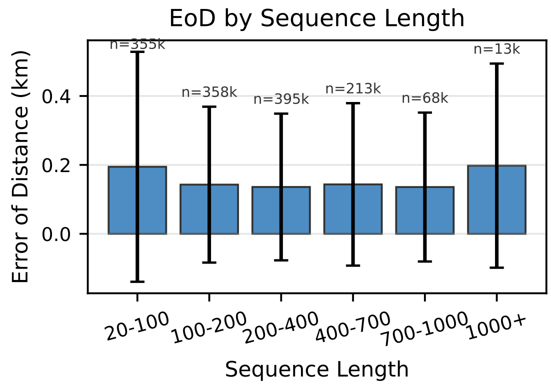

Fig. 4. Distribution of Error of Distance (EoD) versus length.

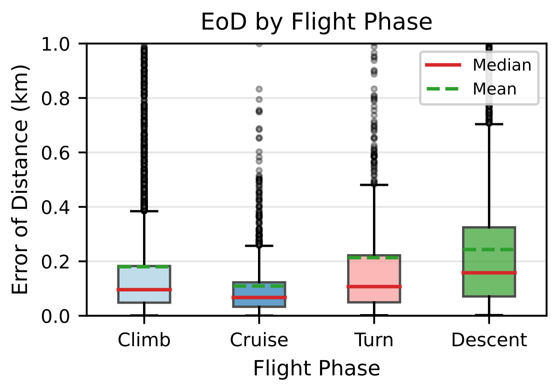

Fig. 5. Error distribution across different flight phases (takeoff, climbing, cruising, descending, landing). Cruising shows the lowest error, followed by climbing and turning, while descending exhibits the highest error.

for assessing trajectory prediction accuracy. Additionally, we report the parameter count (in millions) and inference speed (FPS) for each model.

C. Quantitative Analysis

The main experimental results are presented in Table I, including parameter counts, per-dimension MAE/RMSE, and Error of Distance (EoD) metrics for each model. DiffMoE achieves an EoD of 0.154 km, significantly outperforming the second-best LSTM. Among other methods, only SegRNN achieves competitive results, while Transformer-based and state space model-based approaches show less satisfactory performance on this task.

1) Error Analysis by Sequence Length and Flight Phase: Fig. 4 shows the relationship between sequence length and Error of Distance (EoD). As expected, errors are slightly higher for very short and very long sequences. Fig. 5 displays the error distribution across different flight phases (takeoff, climbing, cruising, descending, landing). The descending phase exhibits relatively higher errors due to more complex maneuvers, while errors across the other three phases show minimal differences.

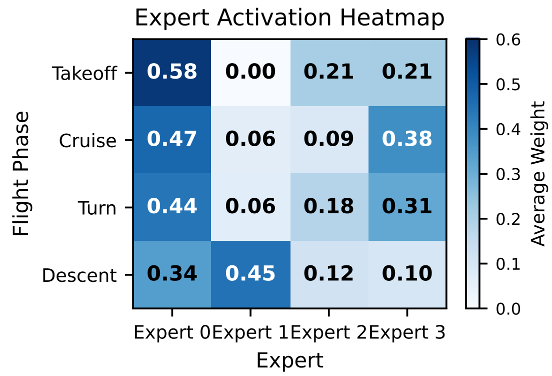

Fig. 6. Expert activation heatmap across flight phases. Expert 0 is the dominant expert active across all four modes. Expert 1 primarily handles descending mode, while Experts 2 and 3 collaborate with Expert 0 to handle cruising and turning.

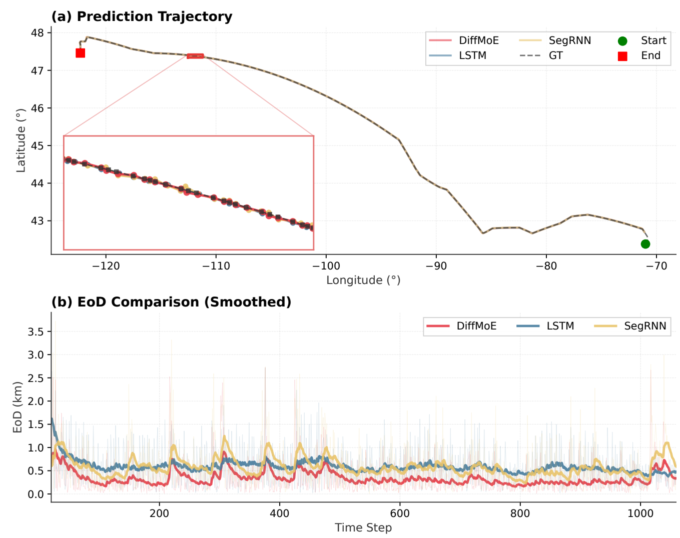

Fig. 7. 2D trajectory comparison between DiffMoE, LSTM, and SegRNN on a representative flight sample. Fig.(a) shows detailed view of trajectory segments where DiffMoE (blue) closely follows the ground truth (green) while LSTM (orange) and SegRNN exhibit larger deviations. The error curves in Fig.(b) show that prediction errors increase at mode transition points, but DiffMoE recovers faster after transitions, demonstrating better ability to capture mode changes.

2) Expert Routing Analysis: To understand the behavior of the MoE module, we visualize the expert activation patterns across different flight phases in Fig. 6. Expert 0 serves as the dominant expert, contributing significantly to all four modes. Expert 1 primarily handles the descending phase. Since cruising and turning modes frequently alternate, Experts 2 and 3 collaboratively manage these two modes. Notably, we do not include phase labels during training—this specialization is learned autonomously by the model, demonstrating the effectiveness of MoE guidance.

D. Ablation Study and Parameter Stability Analysis

1) Ablation Study: Table II compares different normalization methods and core components. All three alternative

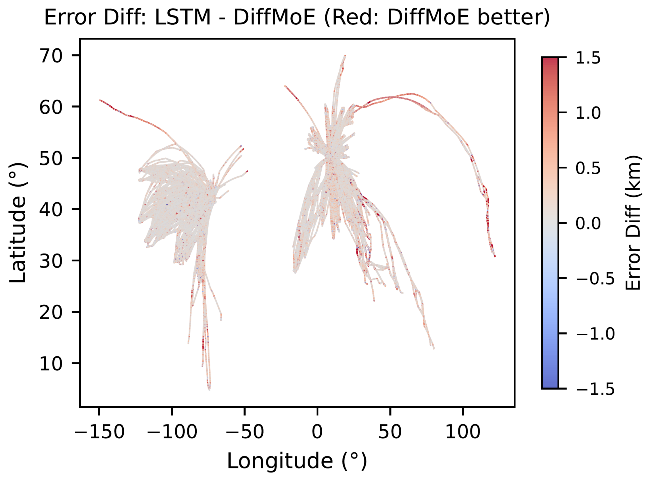

Fig. 8. Spatial distribution of prediction error differences between DiffMoE and baseline methods across BOS and EDDF airports. Red points are mainly distributed at turning and landing phases of long trajectories, indicating that DiffMoE also improves long sequence handling capability.

normalization methods perform significantly worse than Diff Norm. Removing Diff Encode and MoE decreases model performance by 16.2% and 6.5% respectively, effectively demonstrating the contribution of our proposed modules. Table III evaluates different backbone networks and embedding layers. Replacing Diff Embedding with other layers yields inferior results compared to causal convolution. Replacing the backbone with Mamba or GRU achieves competitive performance, demonstrating the transferability of our approach.

2) Parameter Stability Analysis: Table IV analyzes the impact of model scale and MoE expert count. Experiments C1 and C2 show that increasing model dimension to 512 does not improve performance, suggesting that excessive complexity may lead to overfitting and is unsuitable for this task. Experiments C3-C5 demonstrate that more experts are not always better, with 4-6 experts being optimal.

E. Qualitative Analysis

1) Trajectory Prediction Visualization: Fig. 7 presents a 2D map visualization comparing the predicted trajectories of DiffMoE against LSTM and SegRNN on a representative flight sample. The visualization clearly demonstrates that DiffMoE predictions closely follow the ground truth trajectory, while baseline methods exhibit larger deviations, particularly during turning phases. Fig. 8 shows that DiffMoE's improvements are mainly concentrated in the mid-to-late segments of long trajectories, indicating enhanced prediction capability for long sequences.

2) Multi-step Inference Analysis: We also evaluate DiffMoE under multi-step autoregressive inference. The results show approximately linear cumulative error growth, indicating that DiffMoE can effectively perform multi-step prediction without causing trajectory deviation.

V. Conclusion

This paper presents DiffMoE, a framework for variable-length flight trajectory prediction. Through differential normalization and encoding, absolute coordinates are converted into relative displacements to address scale unification across variable-length sequences. The MoE residual prediction head adaptively handles flight mode transitions via multi-expert soft routing. Experiments on real-world ADS-B data show that DiffMoE outperforms all baselines with a distance error as low as 154 meters.

This work has limitations in dataset scale (∼20K trajectories) and single-task validation. Future work will focus on intent information integration (e.g., flight routes) and multi-task learning for trajectory prediction, ETA estimation, and trajectory imputation.

References

[1] M. Strohmeier, M. Schäfer, V. Lenders, and I. Martinovic, "Realities and challenges of NextGen air traffic management: The case of ADS-B," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 111–118, 2014.

[2] Flightradar24, "Flightradar24: Live flight tracker," 2023. [Online]. Available: https://www.flightradar24.com. [Accessed: Oct. 31, 2023].

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

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[5] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 12, 2021, pp. 11106–11115.

[6] Y. Liu, C. Li, J. Wang, and M. Long, "Koopa: Learning non-stationary time series dynamics with Koopman predictors," *Advances in Neural Information Processing Systems*, vol. 36, pp. 12271–12290, 2023.

[7] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," in *First Conf. Language Modeling*, 2024.

[8] H. Hou and F. R. Yu, "RWKV-TS: Beyond traditional recurrent neural network for time series tasks," *arXiv preprint arXiv:2401.09093*, 2024.

[9] S. Lin, W. Lin, W. Wu, F. Zhao, R. Mo, and H. Zhang, "SegRNN: Segment recurrent neural network for long-term time series forecasting," *IEEE Internet of Things Journal*, 2025.

[10] Y. Hu, G. Zhang, P. Liu, D. Lan, N. Li, D. Cheng, T. Dai, S.-T. Xia, and S. Pan, "TimeFilter: Patch-specific spatial-temporal graph filtration for time series forecasting," *arXiv preprint arXiv:2501.13041*, 2025.

[11] Z. Shi, M. Xu, Q. Pan, B. Yan, and H. Zhang, "LSTM-based flight trajectory prediction," in *2018 Int. Joint Conf. Neural Networks (IJCNN)*, 2018, pp. 1–8.

[12] Z. Shi, M. Xu, and Q. Pan, "4-D flight trajectory prediction with constrained LSTM network," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 7242–7255, 2020.

[13] X. Wu, H. Yang, H. Chen, Q. Hu, and H. Hu, "Long-term 4D trajectory prediction using generative adversarial networks," *Transp. Res. Part C: Emerg. Technol.*, vol. 136, p. 103554, 2022.

[14] D. Guo, E. Q. Wu, Y. Wu, J. Zhang, R. Law, and Y. Lin, "FlightBERT: Binary encoding representation for flight trajectory prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 1828–1842, 2022.

[15] H. Shafienya and A. C. Regan, "4D flight trajectory prediction using a hybrid deep learning prediction method based on ADS-B technology: A case study of Hartsfield–Jackson Atlanta International Airport (ATL)," *Transp. Res. Part C: Emerg. Technol.*, vol. 144, p. 103878, 2022.

[16] Z. Zhang, D. Guo, S. Zhou, J. Zhang, and Y. Lin, "Flight trajectory prediction enabled by time-frequency wavelet transform," *Nature Communications*, vol. 14, no. 1, p. 5258, 2023.

[17] S. Luo, M. Zhao, Z. Zhao, L. Li, S. Zhang, and X. Zhang, "FT-TF: A 4D long-term flight trajectory prediction method based on Transformer," in *2023 42nd Chinese Control Conf. (CCC)*, 2023, pp. 4616–4621.