Code “TRAJECTORY”

Aircraft Trajectory Generation Method Based on Visual Transformer

ANNOTATION

This work contains: introduction, 3 chapters, conclusions, list of references, which contains 41 references. Total work size – 24 pages (excluding appendices and list of references). Contains 1 table, 9 pictures.

Relevance

The generation of accurate and reliable aircraft trajectories plays a crucial role in modern aviation, as it directly impacts the efficiency, safety, and sustainability of air transportation systems. With the continuous growth of global air traffic, airspace management requires advanced trajectory prediction methods to prevent conflicts, optimize routes, and reduce delays. Moreover, trajectory generation supports the integration of emerging aviation technologies, including unmanned aerial vehicles (UAVs), urban air mobility platforms, and next-generation air traffic management systems.

Purpose of the Work

Development of a novel approach for generating aircraft flight trajectories, integrating multiple methodologies to more accurately capture the complex dynamics and constraints of real-world flight.

Tasks

1. comprehensive analysis of existing approaches to aircraft trajectory generation, highlighting their strengths and limitations;
2. investigation of modern models and methodologies applied to aircraft trajectory generation;
3. design and development of a custom hybrid model tailored for aircraft trajectory generation tasks;
4. pre-processing and refinement of the aircraft trajectory dataset to ensure data quality and consistency;
5. training and optimization of the proposed model on the prepared dataset;
6. evaluation and analysis of training outcomes, including performance metrics and model behavior;

7. assessment of model performance under diverse scenarios, with emphasis on adaptability and robustness.

Method

This paper presents a method for generating aircraft trajectories that integrates deep learning with aerodynamic principles to ensure physically consistent predictions. The model is designed to capture the temporal patterns of aircraft motion and produce smooth, realistic trajectories through a multi-level approach that combines historical flight data with strict physical constraints, including speed, acceleration, and motion smoothness. The model was trained with a multi-component loss function that considers derivatives of multiple orders – position, velocity, acceleration, and jerk, enabling it to learn full flight dynamics and generate trajectories consistent with physical constraints and realistic flight scenarios.

Key words

air traffic, machine learning, deep learning, trajectory generation, transformer, cross-attention, physics-based approach.

CONTENTS

CHAPTER 1 INTRODUCTION

1.1 Motivation and relevance

Currently, the main achievements in the application of AI in the field of air transport are due to the development of new technologies, the availability of powerful computing resources, and the ability to use large training samples [1-4]. Modern AI uses a combination of new neural network architectures and advanced methods of machine/deep/semi-supervised/transfer learning, which significantly improves the quality of processing different types of data [5-11].

Today, artificial intelligence allows for the comprehensive solution of a huge range of tasks with high accuracy and speed. First and foremost, this applies to flight route optimization, which ultimately leads to reduced fuel consumption.

Air traffic management, covering route/area control, approach control, and airport control.

A lot of researches are being conducted in the field of trajectory prediction based on machine learning [12, 13] and its generation [14-17], where learning-based models generate or refine continuous arrival trajectories that match the planned sequence, minimizing deviations and ensuring energy efficiency (e.g., a continuous descent trajectory can save fuel and reduce emissions).

The relevance of aircraft trajectory generation research lies in its direct impact on the efficiency, safety, and sustainability of air transport. Accurate trajectory generation enables better planning and coordination in crowded airspace, reduces fuel consumption and emissions, minimizes delays, making it a crucial area of study in modern aviation.

1.2 Analysis of existing methods

Aircraft trajectory generation has evolved significantly over the past decades, with research approaches broadly categorized into model-driven and data-driven methodologies. Each approach presents distinct advantages and limitations that have shaped the current landscape of trajectory prediction systems.

1.2.1 Model-Driven Approaches

Traditional model-driven methods generate aircraft trajectories by leveraging kinetic and kinematic models with various parameters including initial aircraft states, performance coefficients, weather conditions, and flight intentions [18, 19]. These physics-based approaches inherently guarantee flyability since they are grounded in aerodynamic principles and aircraft performance constraints [20]. However, they face significant challenges in balancing fidelity and diversity requirements.

Early model-driven systems adopted free-flight concepts to enhance trajectory diversity by allowing aircraft to deviate from published flight routes [21, 22]. While this approach successfully generated varied trajectories, it resulted in poor fidelity as the synthetic paths often exhibited distributions that differed substantially from actual operational data. To address these concerns, subsequent research focused on optimization-based methods that search for optimal flight paths among predefined routes, including Standard Terminal Arrival Routes and associated deviation patterns [23]. Although these methods improved fidelity, they suffered from limited diversity due to the long-tail effect inherent in route selection, leading to homogenized trajectory generation that failed to capture the full spectrum of real-world operations.

1.2.2 Optimization-Based Approaches

Optimization frameworks have been widely employed to bridge the gap between flyability and operational realism. Techniques such as dynamic programming, mixed-integer linear programming, and evolutionary algorithms have been utilized to compute fuel-optimal, time-optimal, or conflict-free trajectories [24, 25]. These approaches are particularly effective in structured environments, where clear performance objectives, such as fuel efficiency or minimal delay, can be formalized as cost functions subject to operational constraints. Typical constraints include separation minima, airspace structure, weather avoidance, and compliance with Standard Instrument Departures (SIDs) or Standard Terminal Arrival Routes (STARs). Multi-objective optimization has also been introduced to balance competing goals (e.g., minimizing emissions while reducing conflict risk), further aligning generated trajectories with environmental and operational needs.

Despite these advantages, optimization-based approaches face notable challenges. Their reliance on heavy computation makes them less suitable for large-scale generation or real-time applications, especially when modeling complex environments with multiple interacting aircraft. Furthermore, their outputs are often deterministic and concentrated around optimal solutions, which limits diversity and fails to capture the variability observed in real-world operations. Attempts to introduce randomness or relax constraints to enhance diversity often compromise fidelity or violate feasibility. As a result, while optimization provides strong guarantees for flyability and compliance, it struggles to simultaneously ensure scalability and realism in diverse traffic scenarios.

1.2.3 Data-Driven Methods

The emergence of machine learning has revolutionized trajectory generation, particularly in ground transportation applications [26, 27]. These successes have motivated researchers to extend data-driven approaches to aircraft trajectory generation. Early implementations utilized traditional neural architectures including Multilayer Perceptrons [28], Long Short-Term Memory networks [29], and Convolutional Neural Networks [30] to extract spatial-temporal features from historical trajectory data.

More sophisticated generative models have since been developed, incorporating Variational Auto-Encoders (VAEs) [31] and Generative Adversarial Networks (GANs) to synthesize novel trajectories. Recent aviation-specific implementations include VampPrior TCVAE [32], Conv1D-GAN [33], Gaussian mixture model-based generators [34], and Principal Component Analysis approaches [35]. These methods demonstrate improved fidelity and diversity compared to traditional model-driven approaches by incorporating controllers' operational experience extracted from historical data.

1.2.4 Reinforcement Learning Approaches

Reinforcement learning (RL) has emerged as a promising direction for trajectory generation, particularly in environments where sequential decision-making and adaptation to dynamic conditions are required. In RL-based frameworks, the aircraft is modeled as an agent that learns optimal maneuvers through interaction with a simulated air traffic environment. Reward functions can encode efficiency, safety, and compliance with air traffic regulations [36, 37]. This formulation allows RL to naturally capture the trade-offs between conflicting objectives, such as minimizing fuel consumption while avoiding conflicts or adhering to sector capacity constraints.

Several RL paradigms have been investigated in this context. Value-based methods such as Q-learning [38] have been applied to simplified two-dimensional navigation problems, while policy gradient methods and actor–critic architectures have been adopted for continuous control tasks relevant to aircraft dynamics. More advanced approaches leverage deep reinforcement learning (DRL) [39], where deep neural networks approximate the value or policy functions, enabling the handling of high-dimensional state spaces that include weather fields, air traffic density, and aircraft performance envelopes. Multi-agent reinforcement learning (MARL) [40] has also gained attention, with multiple aircraft modeled as interacting agents that must coordinate to avoid conflicts while maintaining efficiency.

Despite these advantages, RL-based methods face several critical challenges. Training typically requires millions of simulated interactions, making them computationally intensive and dependent on high-fidelity simulators. Moreover, the resulting policies may overfit to the training environment and fail to generalize under real-world uncertainties, such as unexpected controller interventions, varying weather patterns, or unmodeled aircraft dynamics. The design of reward functions is another key limitation: overly simplistic rewards can lead to unrealistic or unsafe behaviors, while overly complex reward shaping may hinder convergence. Safety and explainability remain significant concerns, as RL policies often behave like black boxes, making it difficult to certify them for safety-critical domains such as aviation.

Nevertheless, reinforcement learning provides a flexible foundation for trajectory generation, particularly when combined with domain knowledge or hybridized with physics-based models. Integrating explicit flight dynamics constraints into the training loop, or using imitation learning to pre-train policies from operational data, are promising directions that could help RL achieve more reliable and physically consistent trajectory generation.

1.3 Current limitations

Although a wide range of methods have been explored, several fundamental limitations remain across existing approaches. Model-driven and optimization-based frameworks, while ensuring flyability, often lack diversity and fail to replicate the operational complexity of real-world trajectories. Their deterministic nature tends to produce homogenized paths that do not capture the variability introduced by human controllers, weather deviations, and airspace constraints.

On the other hand, data-driven and reinforcement learning approaches excel at capturing statistical patterns from historical operations but frequently generate trajectories that violate physical or operational constraints. Their reliance on differentiable loss functions is a key bottleneck, as many flight performance and safety requirements are inherently non-differentiable. Consequently, generated trajectories may compromise feasibility despite appearing realistic. Furthermore, probabilistic models of deviation are insufficient to represent the structured and rule-based nature of air traffic control interventions, which are deterministic rather than purely stochastic.

As a result, current research struggles to simultaneously achieve fidelity, diversity, flyability, and scalability-highlighting the need for hybrid approaches that integrate the strengths of physics-based models, data-driven learning, and operational knowledge.

1.4 Proposed direction

In this work, we aim to overcome the above limitations by introducing a hybrid trajectory generation framework that combines the strengths of deep learning with the guarantees of physically consistent modeling. Specifically, we integrate physics-based constraints directly into the learning pipeline, ensuring that generated trajectories remain operationally feasible while still benefiting from the flexibility and pattern recognition capabilities of modern neural networks. This design allows the model to respect aerodynamic principles and flight performance limits, while simultaneously learning realistic variations from historical operational data. By embedding physical correctness into the generative process itself, our approach seeks to achieve a more balanced trade-off between fidelity, diversity, flyability, and scalability than existing methods.

Unlike conventional data-driven methods that rely on post-processing corrections or differentiable approximations of constraints, our framework enforces physical feasibility during the generation phase. This eliminates the need for heuristic adjustments or surrogate models and enables the direct incorporation of non-differentiable aerodynamic rules into the trajectory synthesis process. At the same time, the use of deep generative models ensures that the system can capture complex patterns of controller interventions, route deviations, and operational variability that purely model-driven techniques fail to represent.

CHAPTER 2 THEORETICAL METHOD

2.1 Dataset description

The dataset employed for training our model is ATFMTraj (Aircraft Trajectory Classification Data for Air Traffic Management) [41]. ATFMTraj provides a large-scale collection of flight trajectory data specifically curated to support research in air traffic management, trajectory prediction, and classification tasks. It contains approximately 70,000 distinct aircraft trajectories, each with an average duration of 1200 time steps, sampled at a frequency of one second. This high temporal resolution ensures that the dataset captures fine-grained variations in aircraft dynamics and operational patterns. Examples of dataset trajectories illustrated on Fig. 1.

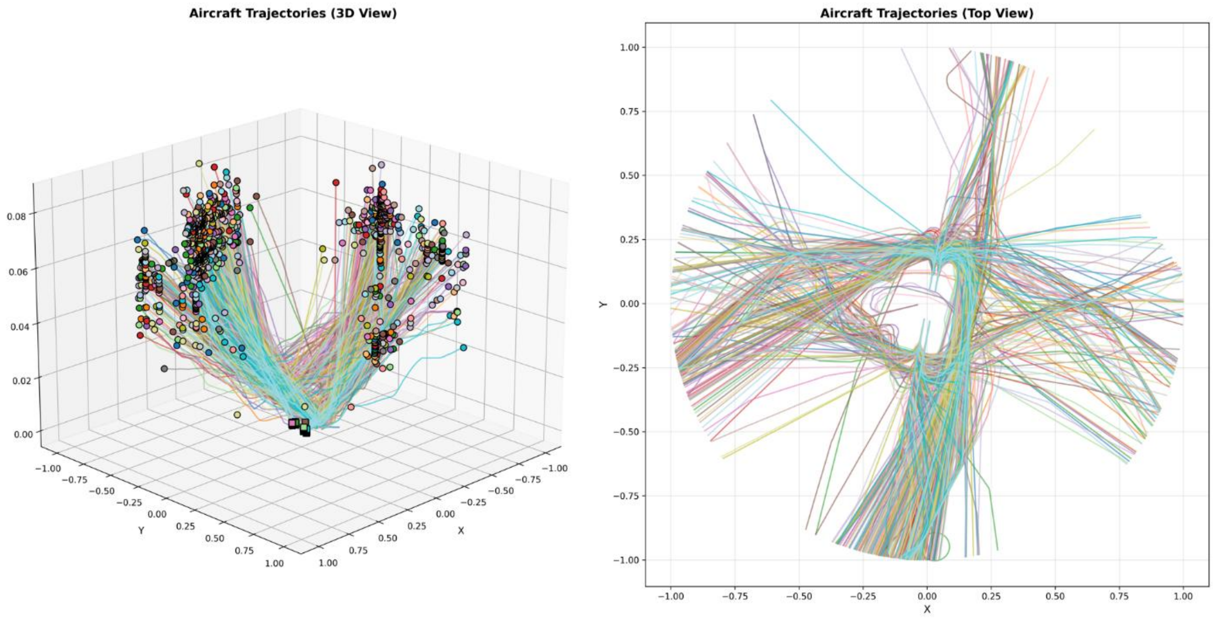

Fig. 1. Examples of dataset trajectories in 3d space and projected on 2d space

The trajectories were collected from a diverse range of airports across the globe, including major hubs such as Incheon International Airport (ICN), Stockholm Arlanda Airport (ARN), and Zurich Airport (ZRH). By encompassing different airspaces, traffic densities, and operational environments, the dataset introduces substantial variability in flight paths. This diversity enhances the generalizability of models trained on

ATFMTraj, enabling them to learn not only common movement patterns but also rare or complex behaviors observed in real-world operations.

To ensure consistency and usability, the dataset undergoes a rigorous preprocessing pipeline. First, incomplete or corrupted trajectories are discarded. The remaining flight paths are then transformed from geographic coordinates (latitude, longitude, altitude) into a local East–North–Up (ENU) Cartesian coordinate system, bounded by an airport-specific radius. Each trajectory is resampled at 1-second intervals, and noise is reduced through outlier removal and Savitzky–Golay filtering to obtain smoother and more physically consistent paths. Afterward, trajectories are normalized, bringing all values into a comparable scale across airports.

For our specific task of aircraft trajectory generation, we further adapt the dataset to match the requirements of our model. Instead of using the full one-second frequency, trajectories are downsampled to one point every five seconds, which reduces excess points while retaining essential motion dynamics. In addition, longer sequences are sliced into smaller trajectory segments that are more suitable for model training, ensuring that the inputs remain consistent with the design and capacity of our generation framework.

2.2 General structure information

The core architecture has structure, designed to capture temporal patterns and ensure physically realistic predictions Fig. 2.

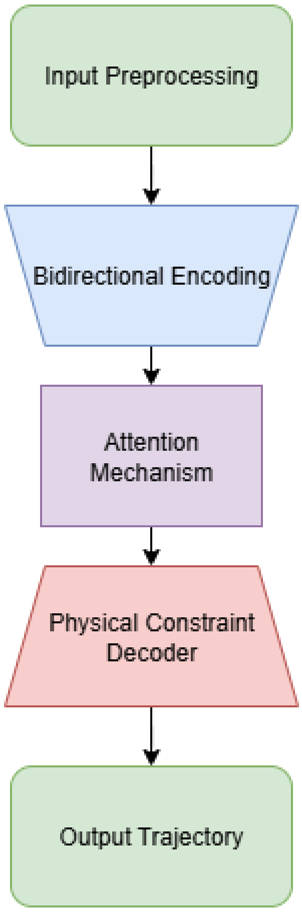

Fig. 2. General model structure

What makes our approach unique is the integration of learned patterns with physical constraints throughout the entire pipeline. While the model learns complex relationships from data, it simultaneously enforces velocity limits, acceleration constraints, and smoothness requirements that reflect real aircraft capabilities. By combining data-driven learning with physics-based validation, we achieve predictions that are both accurate and operationally feasible for real-world aviation applications.

2.3 Feature extraction

Aircraft movement follows specific patterns related to velocity, acceleration, and direction changes, so our preprocessing pipeline extracts these kinematic properties and creates rich feature representations that help the model understand the underlying physics of flight.

The model begins by transforming raw three-dimensional coordinates into a comprehensive 16-dimensional feature space that captures the essential physics of aircraft motion Fig. 3. Rather than expecting the neural network to learn motion dynamics from scratch, this approach provides it with physically meaningful features from the outset.

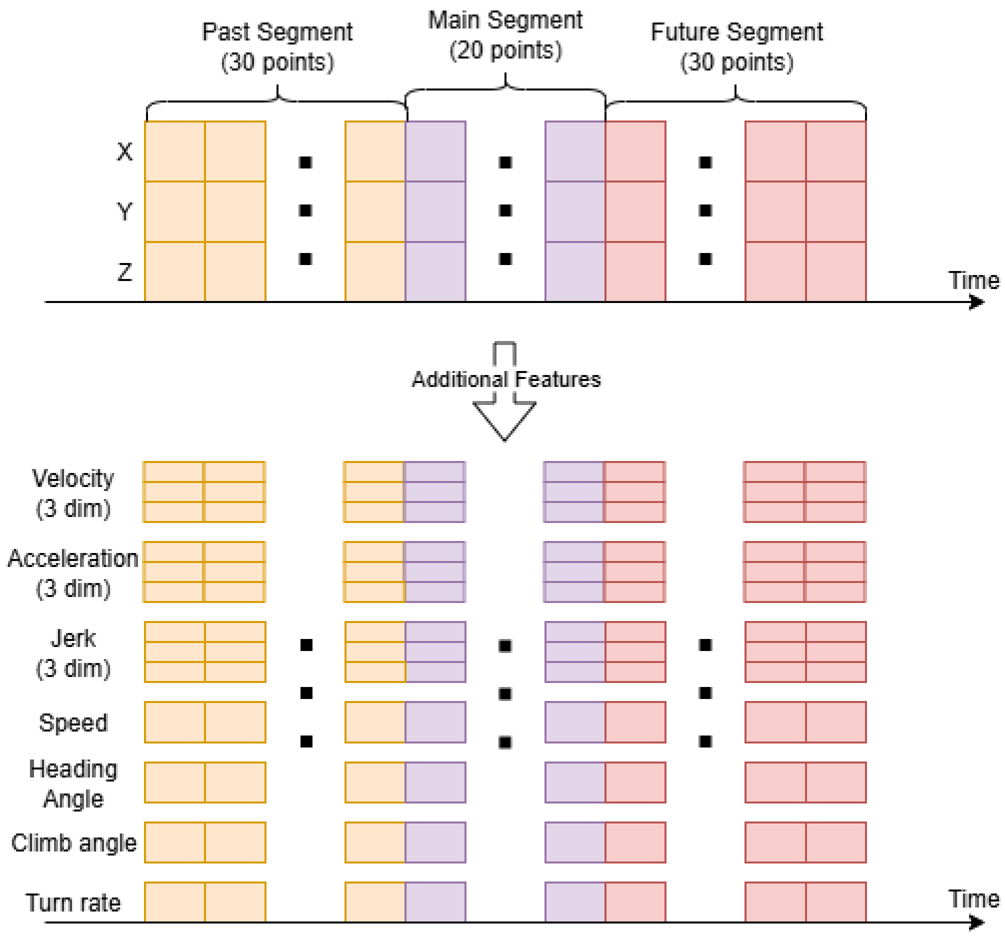

Fig. 3. Visualization of data and feature engineering

The feature extractor computes multiple derivatives of position to capture different aspects of motion. Starting with the original x, y, and z coordinates, it calculates velocity using central differences. This same approach extends to acceleration and jerk, the third derivative that measures how smoothly acceleration changes over time.

Beyond these derivatives, the extractor computes specialized aviation-relevant features. Speed represents the magnitude of the velocity vector, while heading angle captures the horizontal direction of flight. The climb angle indicates whether the aircraft is ascending or descending relative to the horizontal plane. Together, these features provide a multi-scale temporal representation of motion, from instantaneous position to the rate of acceleration change.

The use of central differences throughout the feature extraction process ensures numerical stability and reduces noise in the computed derivatives. This is particularly important for aircraft trajectories, where measurement noise can significantly affect derivative calculations. By starting with physics-aware features, the model gains an immediate advantage in understanding motion patterns. Also, we handle the fact that velocity and acceleration cannot be computed at trajectory boundaries by using zero-padding, ensuring consistent input dimensions while explicitly marking these boundary conditions for the model.

2.4 Encoder structure

After adding new features to input data we are using encoder to further process it. Rather than treating trajectory generation as a simple sequence-to-sequence problem, we recognize that past and future segments contain fundamentally different types of information that require specialized processing before they can be effectively combined.

The encoder employs separate Gated Recurrent Unit networks for past and future sequences, allowing each to specialize in its temporal direction. GRUs were chosen over simpler RNNs for their superior gradient flow properties and over transformers for their computational efficiency with sequential data. These parallel encoders process their respective sequences independently before engaging in sophisticated information exchange. Fig. 4.

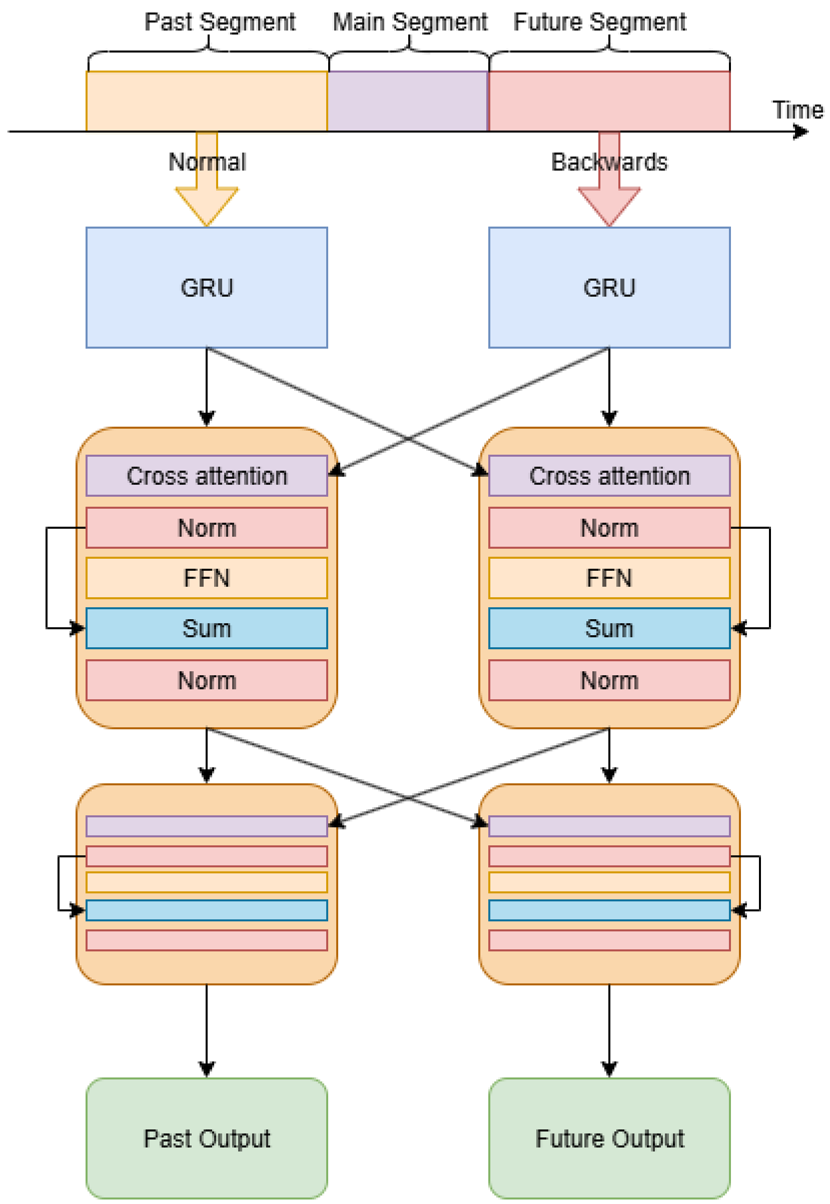

Fig. 4. Trajectory encoding process

The key innovation comes through multi-layer cross-attention mechanisms. In these layers, the past trajectory representation attends to the future trajectory and vice versa, enabling the model to identify relationships between where the aircraft was and where it will be. This bidirectional attention is applied twice, with each iteration followed by feed-forward networks and layer normalization, creating increasingly refined representations that capture complex temporal dependencies.

Learnable position embeddings enhance the model's understanding of temporal relationships within each sequence. These learned embeddings can adapt to the specific temporal patterns present in aircraft trajectories, such as the characteristic time scales of different maneuvers.

The encoder pays special attention to boundary conditions-the points where the predicted segment must connect to the known past and future. It extracts the last two points from the past trajectory and the first two from the future, capturing both position and velocity information at these critical junctions. A dedicated fusion network combines these boundary features, ensuring the model maintains awareness of the connection constraints throughout processing.

2.5 Physics-constrained decoder

Unlike traditional sequence decoders that generate outputs based solely on learned patterns, our decoder integrates aerodynamic principles and operational constraints throughout the generation process.

The decoder component represents a critical stage in the trajectory prediction pipeline, transforming the abstract encoded context representation into a smooth, physically plausible trajectory that seamlessly connects the past and future segments. This transformation is achieved through a sophisticated multi-stage process that strategically combines classical analytical methods with modern learned components, balancing mathematical guarantees with data-driven flexibility Fig. 5.

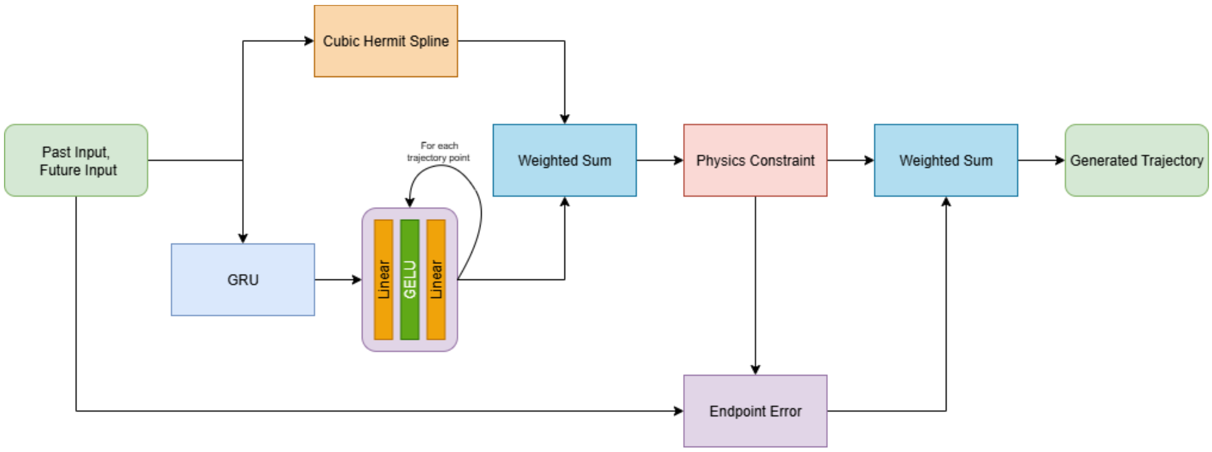

Fig. 5. Structure of model decoder

The first step of trajectory generation relies on cubic Hermite spline interpolation, a well-established mathematical technique from numerical analysis. This approach creates a smooth parametric curve between the end point of the past trajectory and the beginning point of the future trajectory, leveraging not just the spatial positions at these critical boundary points but also the instantaneous velocities.

The choice of Hermite splines is particularly advantageous because they provide mathematical guarantees that both position and velocity change continuously without abrupt jumps or discontinuities. This property is a critical requirement for realistic aircraft motion modeling, as real aircraft cannot instantaneously change velocity due to physical constraints like inertia and aerodynamic forces.

The cubic formulation provides sufficient flexibility to match both position and velocity boundary conditions while maintaining computational efficiency, striking an optimal balance between complexity and expressiveness.

While the Hermite spline provides a mathematically sound baseline, it cannot capture the complex, scenario-specific patterns present in real flight data. To address this limitation, a Gated Recurrent Unit (GRU)-based trajectory generator operates on top of the analytical foundation to produce context-aware adjustments.

This recurrent neural network processes the encoded context representation along with explicit boundary condition information to generate trajectory modifications that reflect the specific characteristics of the current flight scenario.

This design choice is crucial: it maintains the inherent smoothness and physical plausibility of the base Hermite trajectory while adding the necessary fine-grained adjustments needed to improve prediction accuracy and capture real-world flight patterns that pure analytical methods would miss.

The model incorporates multiple complementary smoothing techniques to ensure that the generated trajectories remain physically plausible and free from artifacts. We use filter on the trajectory by computing weighted averages of each point with its temporal neighbors, effectively reducing high-frequency noise and oscillations that could arise from the neural network components.

Unlike simple moving averages, this filtering approach is designed to preserve important features like peaks and overall trajectory shape while suppressing unrealistic fluctuations. Additionally, the decoder implements explicit acceleration constraints that enforce physically realistic limits on the rate of velocity change. These constraints prevent the generation of impossible maneuvers that would exceed the performance envelope of real aircraft-such as instantaneous direction changes or excessive g-forces-ensuring that all predicted trajectories respect fundamental physical laws and aircraft operational limitations.

To ensure proper geometric connection with the known future trajectory segment, the decoder applies a sophisticated soft endpoint constraint mechanism. Rather than rigidly forcing exact endpoint matching through hard constraints-which could create artificial discontinuities or unrealistic sharp corrections near the boundary-the system gradually corrects the trajectory toward the target endpoint using a smooth linear interpolation of the accumulated position error over the predicted time window.

This approach also implements that early in the predicted segment, the trajectory has more freedom to deviate based on learned patterns, but as it approaches the boundary with the future segment, corrections become progressively stronger. This gradual blending maintains overall smoothness and physical realism while simultaneously ensuring that the predicted segment connects properly to the known future, avoiding visible seams or discontinuities that would compromise the quality and usability of the complete trajectory reconstruction.

2.6 Loss function

Defining loss function is crucial element of our trajectory generation framework. Instead of relying solely on a mean squared error (MSE) between generated and ground-truth trajectories. we introduce a multi-objective loss function, sophisticated loss function to ensure the model learns to generate physically realistic trajectories. This multi-component loss function supervises the model at different scales of motion.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{pos}} + \lambda_1 \mathcal{L}_{\text{vel}} + \lambda_2 \mathcal{L}_{\text{acc}} + \lambda_3 \mathcal{L}_{\text{jerk}}$$

where $\mathcal{L}_{\text{pos}}$ – weighted MSE loss; $\mathcal{L}_{\text{vel}}$ – velocity loss; $\mathcal{L}_{\text{acc}}$ – acceleration loss; $\mathcal{L}_{\text{jerk}}$ – jerk penalty; λ_i – weights.

The primary component measures position accuracy through mean squared error between predicted and ground truth trajectories. However, matching positions alone isn't sufficient for smooth, realistic motion. The loss function therefore includes velocity matching with a weight of 0.3, ensuring the model learns correct speed profiles throughout the trajectory. Acceleration matching, weighted at 0.2, maintains proper dynamics and helps the model understand how aircraft change speed and direction.

Furthermore, the loss includes a jerk penalty weighted at 0.1. By penalizing large values of jerk, the training process naturally encourages the generation of smooth trajectories. This reflects the physical reality that aircraft, due to their mass and control system limitations, cannot change acceleration instantaneously.

This multi-scale supervision provides training signals at different derivative orders, helping the model learn the full dynamics of motion rather than just position sequences. The carefully chosen weights balance these objectives, preventing any single component from dominating while ensuring all aspects of trajectory quality are maintained.

CHAPTER 3 PRACTICAL RESULTS

3.1 Training parameters

For the training process, the dataset was divided into three subsets: 70% for training, 20% for validation, and 10% for testing. This split ensured that the model had sufficient data for learning while still being evaluated on unseen samples during and after training. The training was conducted for a total of 70 epochs, allowing the model to gradually refine its parameters and improve predictive accuracy.

The training configuration included the following parameters:

- Batch size: 64.
- Learning rate decay: 0.1.
- Optimizer: Adam with parameters (weight decay=; ;).

During the training phase, the model's performance was evaluated on the validation subset at the end of each epoch. This continuous monitoring allowed us to detect overfitting or underfitting trends and ensured that the model's generalization ability was preserved.

3.2 Model results and results analysis

Fig. 6 demonstrates the results of trajectory generation for several representative examples, illustrating how the proposed model captures the overall flight dynamics and produces realistic trajectories.

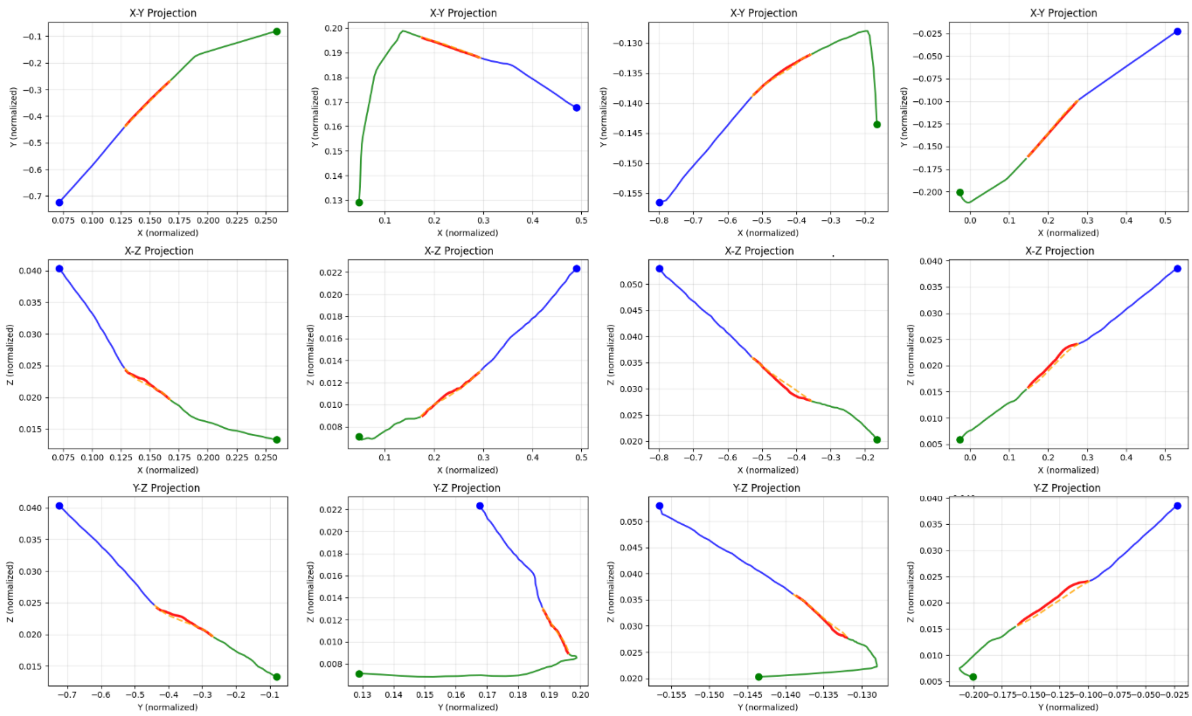

Fig. 6. Examples of generated trajectories, projected on 2d space

In this image you can see examples of trajectory prediction of our model. Each column refers to one example, projected on every possible 2d plane. The blue line represents the past segment of the trajectory, the green line corresponds to future segment, the orange line denotes the model prediction, and the red line indicates the true trajectory for comparison.

Our model produces reasonably accurate predictions. While it does not replicate the target path with complete precision, it generates its own smooth and realistic trajectories that remain close to the ground truth, which was our primary task. This behavior is particularly valuable, since exact replication of future motion is neither feasible nor always desirable due to inherent stochasticity in real-world trajectories.

Moreover, the generated paths exhibit continuity and physical plausibility, avoiding unrealistic jumps or discontinuities that are common in naive approaches. The slight deviations from the true trajectory can also be interpreted as the model's ability to generalize rather than memorize training examples. Overall, these results demonstrate that the model captures both short-term dynamics and long-term motion trends effectively, producing trajectories that are consistent, feasible, and aligned with actual aircraft behavior.

Another, more distinct example of the model's capabilities is its ability to generate entirely new paths that do not simply mirror the ground truth but still maintain realism and adherence to physical constraints Fig. 7. In such cases, model produces smooth and coherent trajectory that could plausibly represent an alternative evolution of the aircraft's movement. This is particularly important in scenarios where future trajectories are inherently uncertain, as the model can propose multiple feasible continuations rather than overfitting to a single deterministic path.

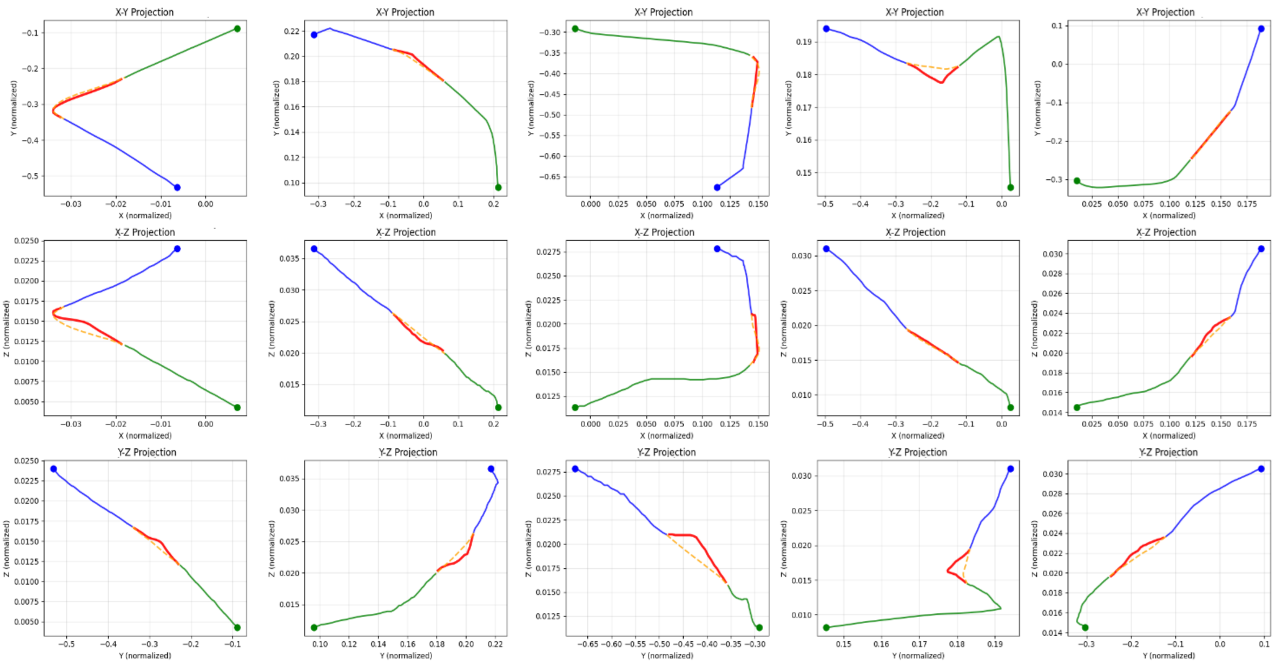

Fig. 7. Examples of model generating completely new trajectories, projected on 2d space

However, despite these strengths, certain limitations remain. In some instances, the model tends to produce unnecessarily complex trajectories Fig. 8. These cases suggest that while the physics constraints embedded in the architecture mitigate most unrealistic behaviors, additional regularization or refinement could further reduce such artifacts. Addressing this challenge is essential for improving reliability in safety-critical applications such as air traffic management.

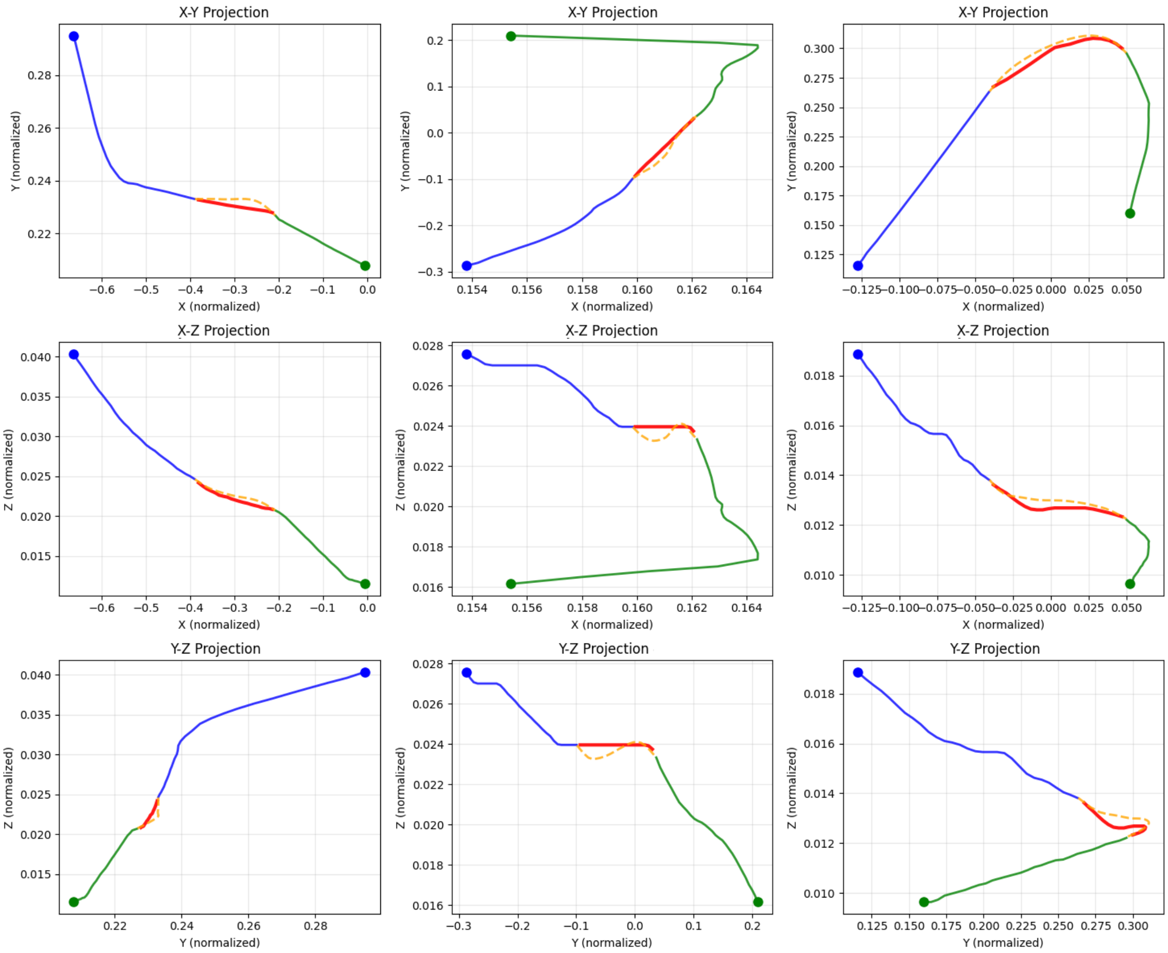

Fig. 8. Examples of model unnecessarily complex trajectories, projected on 2d space

Also, to evaluate objective quality of generated trajectories, we evaluate our model on test dataset and compute metrics that include confidence scores of the model, Mean Square Error and Mean Absolute Error. Table 1 illustrates this metrics.

Table 1. Metrics values for test dataset.

Metric	**Value**
Mean Squared Error	0.000004
Mean Absolute Error	0.001211
Root Mean Squared Error	0.002042

The metrics confirm the effectiveness of the developed model for aircraft trajectory prediction. Despite fluctuations in some specific cases, the model consistently generates trajectories that remain within acceptable operational limits. This indicates that the approach not only achieves high accuracy on average but also maintains robustness under varying conditions. The stability of results across diverse scenarios suggests strong generalization ability, which is crucial for practical deployment in real-world air traffic management systems.

3.3 Model testing in new environment

We also explored a more challenging setting by combining past and future segments originating from different trajectories and using them as input to generate completely new trajectories Fig. 9.

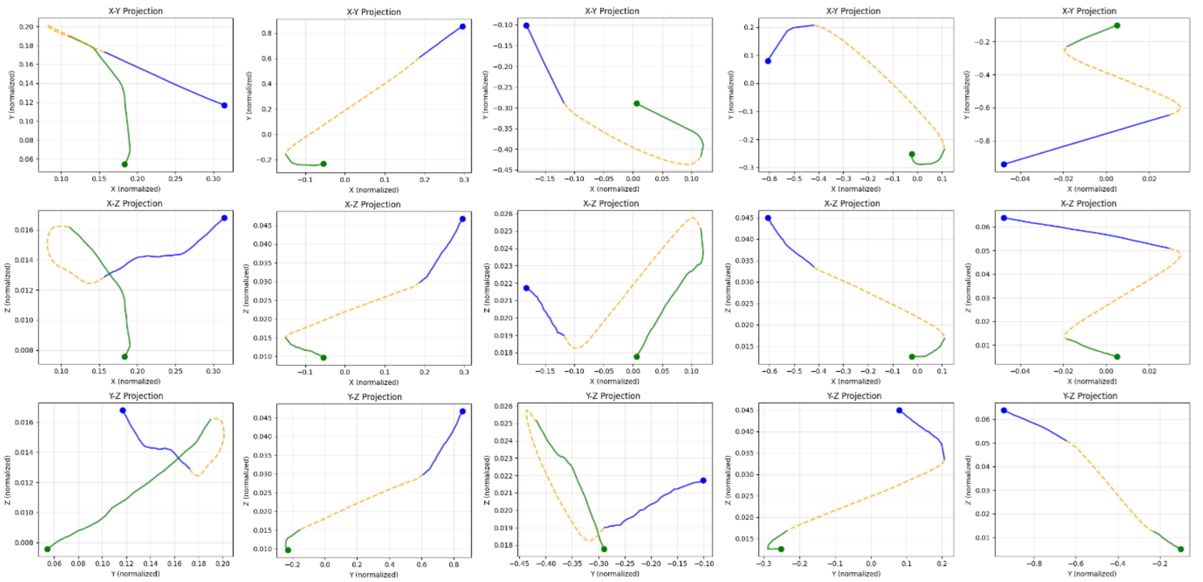

Fig. 9. Examples of model generating trajectories for different dataset paths, projected on 2d space

The model still managed to generate coherent transitions between mismatched segments, producing trajectories that different trajectories into a consistent motion path. This outcome underscores the robustness and adaptability of the approach, highlighting its potential for trajectory generation in more complex operational scenarios.

CONCLUSION

In this work, we presented a hybrid framework for aircraft trajectory generation that integrates data-driven learning with physics-based constraints. Unlike traditional methods that either oversimplify flight dynamics or rely solely on black-box neural models, our approach leverages the strengths of both domains. By embedding aerodynamic principles, velocity and acceleration constraints, and smoothness requirements directly into the pipeline, the proposed model generates trajectories that are not only accurate but also operationally feasible.

The key innovation lies in the seamless integration of multiple specialized components: separate transformer encoders for past and future trajectory segments, cross-modal attention mechanisms for temporal relationship modeling, and a physics-constrained decoder with multiple prediction heads. This architecture enables the model to capture complex flight dynamics while maintaining adherence to fundamental physical constraints including velocity limits, acceleration bounds, and trajectory smoothness requirements.

The experimental evaluation on the ATFMTraj dataset demonstrated that our method consistently produces smooth, continuous, and physically plausible paths, even in complex and uncertain scenarios. The results highlight the model's ability to generalize beyond exact replication, offering realistic alternatives when deterministic predictions are neither possible nor desirable. Importantly, the framework maintains robustness across diverse flight conditions, reinforcing its suitability for safety-critical applications such as air traffic management and decision support systems.

Dataset of 70,000 trajectories from airports worldwide was used to train our model. Validation demonstrates the effectiveness of our approach. The model achieves quantitative performance with MSE of 0.000004 and MAE of 0.001211 and RMSE 0.002042. Visual assessment confirms that generated trajectories exhibit realistic flight patterns and smooth transitions between past and future segments.

While the approach shows strong performance, challenges remain in further reducing overly complex outputs and optimizing for efficiency in large-scale deployments. Future work will focus on refining regularization strategies, exploring multi-modal prediction to handle uncertainty more explicitly, and integrating real-time operational constraints. With these extensions, the framework has the potential to become a core component in next-generation air traffic management systems, supporting safer, more efficient, and more reliable aviation operations.

LIST OF REFERENCES

1. Abduljabbar, R., Dia, H., Liyanage, S., & Bagloee, S. A. (2019). Applications of arti cial intelligence in transport: An overview. Sustainability, 11(1), 189.

2. Geske, A. M., Herold, D. M., & Kummer, S. (2024). Integrating AI support into a framework for collaborative decision-making (CDM) for airline disruption management. Journal of the Air Transport Research Society, 3, Article 100026. http://dx.doi.org/10.1016/j.jatrs.2024.100026

3. Kar, S., Kar, A. K., & Gupta, M. P. (2021). Modeling drivers and barriers of arti cial intelligence adoption: Insights from a strategic management perspective. Intelligent Systems in Accounting, Finance and Management, 28(4), 217238.

4. Torens, C., Durak, U., & Dauer, J. C. (2022). Guidelines and regulatory framework for machine learning in aviation. In AIAA scitech 2022 forum (p. 1132).

5. Li, Y. (2017). Deep reinforcement learning: An overview. arXiv preprint arXiv:1701.07274.

6. Mathew, A., Amudha, P., & Sivakumari, S. (2021). Deep learning techniques: an overview. Advanced Machine Learning Technologies and Applications: Proceedings of AMLTA 2020, 599608

7. Milani, S., Topin, N., Veloso, M.,& Fang, F. (2024). Explainable reinforcement learning: A survey and comparative review. ACM Computing Surveys, 56(7), 136.

8. ...

...

...

9. ...

...

...

10. ...

...

...

11. ...

...

...

12. Choi, H.-C., Deng, C., & Hwang, I. (2021). Hybrid machine learning and estimation-based ight trajectory prediction in terminal airspace. IEEE Access, 9, 151186151197.

13. Gallego, C. E. V., Comendador, V. F. G., Carmona, M. A. A., Valds, R. M. A., Nieto, F. J. S., & Martnez, M. G. (2019). A machine learning approach to air traf c interdependency modelling and its application to trajectory prediction. Trans portation Research Part C: Emerging Technologies, 107, 356386.

14. Gui, X., Zhang, J., Tang, X., & Bao, J. (2025). Improved data-driven trajectory optimization method utilizing deep trajectory generation. Journal of Aerospace Information Systems, 22(1), 2842. http://dx.doi.org/10.2514/1.I011418.

15. Gui, X., Zhang, J., Tang, X., Bao, J., & Wang, B. (2022). A data driven trajectory optimization framework for terminal maneuvering area operations. Aerospace Science and Technology, 131, Article 108010. http://dx.doi.org/10.1016/j.ast.2022.108010.

16. Gui, X., Zhang, J., Tang, X., Delahaye, D., & Bao, J. (2024). A novel aircraft trajectory generation method embedded with data mining. Aerospace, 11(8), 648. http://dx.doi.org/10.3390/aerospace11080648

17. ...

...

...

18. McGee, T. G., & Hedrick, J. K. (2007). Optimal path planning with a kinematic airplane model. Journal of guidance, control, and dynamics, 30(2), 629-633.

19. Sun, J., Ellerbroek, J., & Hoekstra, J. M. (2019). WRAP: An open-source kine matic aircraft performance model. Transportation Research Part C: Emerging Technologies, 98, 118-138.

20. Morris, S. J., & Kroo, I. (1990). Aircraft design optimization with dynamic per formance constraints. Journal of Aircraft, 27(12), 1060-1067.

21. Shahzad, M., Mora-Camino, F., Slama, J. G., & Achaibou, K. (2013). Trajectory generation and display for free ight. arXiv preprint arXiv:1302.4858.

22. Danecker, F. (2023). A Discrete-Continuous Algorithm for Globally Optimal Free Flight Trajectory Optimization. Freie Universitaet Berlin (Germany).

23. Zhou, J., Ca eri, S., Delahaye, D., & Sbihi, M. (2017). Optimization-based design of departure and arrival routes in terminal maneuvering area. Journal of Guidance, Control, and Dynamics, 40(11), 2889-2904.

24. Calvo-Fernández, E., Perez-Sanz, L., Cordero-García, J. M., & Arnaldo-Valdés, R. M. (2017). Con ict-free trajectory planning based on a data-driven con ict resolution model. Journal of Guidance, Control, and Dynamics, 40(3), 615-627.

25. Guitart, A., Delahaye, D., Camino, F. M., & Feron, E. (2023). Collaborative gen eration of local con ict free trajectories with weather hazards avoidance. IEEE Transactions on Intelligent Transportation Systems, 24(11), 12831-12842.

26. Zhang, J., Wang, F. Y., Wang, K., Lin, W. H., Xu, X., & Chen, C. (2011). Data driven intelligent transportation systems: A survey. IEEE transactions on intelli gent transportation systems, 12(4), 1624-1639.

27. Chazal, F., Chen, D., Guibas, L., Jiang, X., & Sommer, C. (2011, November). Data-driven trajectory smoothing. In proceedings of the 19th ACM SIGSPATIAL international conference on advances in geographic information systems (pp. 251 260).

28. Zhang, Z., Zhang, J., Lin, Y., Zhang, K., Zheng, X., & Xiang, D. (2025). Spa tial Temporal Physics-Constrained Multilayer Perceptron for Aircraft Trajectory Prediction. Applied Sciences, 15(18), 9895.

29. Shi, Z., Xu, M., Pan, Q., Yan, B., & Zhang, H. (2018, July). LSTM-based ight trajectory prediction. In 2018 International joint conference on neural networks (IJCNN) (pp. 1-8). IEEE.

30. Li, Y., Ge, R., Ji, Y., Gong, S., & Liu, C. (2017). Trajectory-pooled spatial temporal architecture of deep convolutional neural networks for video event de tection. IEEE Transactions on Circuits and Systems for Video Technology, 29(9), 2683-2692.

31. Xie, L., Guo, T., Chang, J., Wan, C., Hu, X., Yang, Y., & Ou, C. (2023). A novel model for ship trajectory anomaly detection based on Gaussian mixture variational autoencoder. IEEE Transactions on Vehicular Technology, 72(11), 13826-13835.

32. Krauth, K., Bonilla, E. V., Cutajar, K., & Filippone, M. (2016). AutoGP: Ex ploring the capabilities and limitations of Gaussian process models. arXiv preprint arXiv:1610.05392.

33. Wang, Z., Ren, W., & Qiu, Q. (2018). Lanenet: Real-time lane detection networks for autonomous driving. arXiv preprint arXiv:1807.01726.

34. Wiest, J., Hö ken, M., Kreÿel, U., & Dietmayer, K. (2012, June). Probabilistic tra jectory prediction with Gaussian mixture models. In 2012 IEEE Intelligent vehicles symposium (pp. 141-146). IEEE.

35. Gariel, M., Srivastava, A. N., & Feron, E. (2011). Trajectory clustering and an ap plication to airspace monitoring. IEEE Transactions on Intelligent Transportation Systems, 12(4), 1511-1524.

36. Xie, J., Shao, Z., Li, Y., Guan, Y., & Tan, J. (2019). Deep reinforcement learning with optimized reward functions for robotic trajectory planning. IEEE Access, 7, 105669-105679.

37. Naveed, K. B., Qiao, Z., & Dolan, J. M. (2021, September). Trajectory planning for autonomous vehicles using hierarchical reinforcement learning. In 2021 IEEE International Intelligent Transportation Systems Conference (ITSC) (pp. 601-606). IEEE

38. Kosuru, V. S. R., & Venkitaraman, A. K. (2022). Developing a deep Q-learning and neural network framework for trajectory planning. European Journal of Engi neering and Technology Research, 7(6), 148-157.

39. Chai, R., Niu, H., Carrasco, J., Arvin, F., Yin, H., & Lennox, B. (2022). Design and experimental validation of deep reinforcement learning-based fast trajectory planning and control for mobile robot in unknown environment. IEEE Transactions on Neural Networks and Learning Systems, 35(4), 5778-5792.

40. Wang, L., Wang, K., Pan, C., Xu, W., Aslam, N., & Hanzo, L. (2020). Multi-agent deep reinforcement learning-based trajectory planning for multi-UAV assisted mo bile edge computing. IEEE Transactions on Cognitive Communications and Net working, 7(1), 73-84.

41. Phisannupawong, T., Damanik, J. J., & Choi, H.-L. (2024). ATFMTraj: Aircraft trajectory classi cation data for air tra c management [Dataset]. Hugging Face. https://huggingface.co/datasets/petchthwr/ATFMTraj