A logo of a school

Description automatically generated

**General Aviation Trajectory Prediction Research:**

**Out-of-Distribution and Non-IID Data Considerations**

By Tsung-Tse Tu, 47801873

School of Electrical Engineering and Computer Science,

The University of Queensland.

Submitted for Master of Data Science Capstone 2

On 17th October

**Executive Summary**

General aviation (GA) operates in a dynamic and unpredictable environment, where flights are influenced by varying weather conditions, non-standard routes, and fluctuating altitudes. Unlike commercial aviation, which adheres to strict schedules and flight paths, GA flights often display a high degree of variability. This unpredictability challenges current trajectory prediction models, which often struggle to handle data that falls outside their training scope (Out-of-Distribution or OOD) or data that lacks consistent patterns (non-IID). The growing demand for accurate GA flight predictions, driven by the need for improved air traffic management and flight safety, calls for models that can more effectively handle these challenges.

This project is motivated by the limitations of traditional models in predicting the complex paths of GA flights. Many existing models, such as the Kalman Filter (Im, 2024) and Constant Velocity (CV) (Baisa, 2020), struggle when faced with unpredictable factors like diverse weather conditions, irregular flight routes, and fluctuating altitudes. To address these challenges, my team developed an improved version of the pre-trained model. We incorporated synthetic data to enhance the model’s ability to handle unpredictable flight scenarios and better manage OOD data. The goal of this project is to generate more reliable predictions and provide valuable insights, ultimately making GA flight planning safer and more efficient.

To enhance the pre-trained model, I introduced uncertainty estimation techniques, such as Maximum Softmax Probability (MSP). This method helped the model capture its prediction confidence when dealing with unfamiliar data, allowing it to recognize when the input differed from what it had been trained on. Additionally, synthetic data, created by my teammate Andryian Saputra, was integrated into the model’s training process. The synthetic data was designed to mimic real GA flight conditions, and it exposed the model to a wider range of scenarios. This helped the model improve its performance in unpredictable situations.

After multiple rounds of training, fine-tuning, and testing, the model showed a marked improvement in predicting complex GA flight paths. By incorporating synthetic data and addressing both familiar and unfamiliar flight conditions, the model gradually increased its path prediction accuracy, especially in challenging or unusual circumstances.

Future works could focus on improving the model’s computational efficiency and exploring advanced machine learning techniques like ensemble learning to further enhance its performance in even more complex GA environments. This project has made improvements in GA flight path prediction, offering valuable insights for enhancing air traffic control and flight safety through better predictive models.

**Contents**

[**1.** **Introduction** 4](#_Toc180421715)

[**1.1** **Methods and their Limitations** 4](#_Toc180421716)

[**1.1.1** **Constant Velocity (CV) Model** 5](#_Toc180421717)

[**1.1.2** **Kalman Filter** 6](#_Toc180421718)

[**1.1.3** **Particle Filter** 8](#_Toc180421719)

[**1.1.4** **Long Short-Term Memory (LSTM) network** 9](#_Toc180421720)

[**1.1.5** **Spatio-Temporal Graph Neural Networks (ST-GNNs)** 10](#_Toc180421721)

[**1.2** **Environmental Factors** 11](#_Toc180421722)

[**1.3** **OOD and non-IID** 12](#_Toc180421723)

[**1.4** **Motivation** 13](#_Toc180421724)

[**2.** **Objective – The Originality** 15](#_Toc180421725)

[**2.1** **Key Achievements** 15](#_Toc180421726)

[**2.2 Workflow** 16](#_Toc180421727)

[**3.** **Project Preliminary** 18](#_Toc180421728)

[**3.1** **Data Overview** 19](#_Toc180421729)

[**3.1.1** **Preprocessing** 20](#_Toc180421730)

[**3.2** **Pre-trained Model Overview** 21](#_Toc180421731)

[**3.2.1** **Architecture** 21](#_Toc180421732)

[**3.3** **Evaluation Criteria** 22](#_Toc180421733)

[**3.3.1** **Training and Testing Metrics** 22](#_Toc180421734)

[**3.3.2** **Uncertainty Estimation Metrics** 24](#_Toc180421735)

[**3.4** **Ethics and Privacy** 25](#_Toc180421736)

[**3.5** **Time Schedule (need update)** 25](#_Toc180421737)

[**4.** **Pre-trained Model Enhancement (Phase 1)** 26](#_Toc180421738)

[**4.1** **Trial Run** 26](#_Toc180421739)

[**4.2** **Adding Visualization** 28](#_Toc180421740)

[**4.2.1** **Supervised Visualization: Real vs. Predicted Trajectories** 28](#_Toc180421741)

[**4.2.2** **Unsupervised Visualization: 3D Clustering** 29](#_Toc180421742)

[**4.3** **Addressing Overfitting and Underfitting (Loss Curve)** 30](#_Toc180421743)

[**4.4** **Addressing OOD (Uncertainty estimation)** 31](#_Toc180421744)

[**4.5** **Final Test (After Fine-tuning)** 31](#_Toc180421745)

[**5.** **Synthetic Data Implementation (Phase 2)** 35](#_Toc180421746)

[**5.1** **Integration** 36](#_Toc180421747)

[**5.2** **Results Showcase** 36](#_Toc180421748)

[**6.** **Conclusion** 41](#_Toc180421749)

[**7.** **Drawbacks and Future Work** 42](#_Toc180421750)

[**7.1** **drawbacks** 42](#_Toc180421751)

[**7.2** **Future Works** 42](#_Toc180421752)

[**References** 44](#_Toc180421753)

[**Appendix** 47](#_Toc180421754)

# **Introduction**

General aviation (GA) involves a wide range of flight operations that do not fall under commercial air transport, as defined by the International Civil Aviation Organization (ICAO). This broad category consists of everything from private leisure flights to business jet operations, all of which operate under diverse and often unpredictable conditions. Accurate trajectory prediction is essential for the safety and efficiency of GA flights, but it remains challenging due to the variability in altitudes, weather conditions, and flight paths. Unlike commercial flights that follow more fixed schedules and routes, GA flights are less predictable, which limits the effectiveness of traditional trajectory prediction methods. For example, as Wang et al. (2021) explain, technologies like Automatic Dependent Surveillance-Broadcast (ADS-B) often struggle to predict low-altitude GA flight paths accurately, due to the changing operational environments.

To address these challenges, this project focused on developing a more robust predictive model for GA flight trajectories, particularly in handling out-of-distribution (OOD) and non-independent, non-identically distributed (non-IID) data. Recognizing the limitations of traditional models, such as the Kalman Filter (Im, 2024) and Constant Velocity models (Baisa, 2020), I implemented a refined version of the pre-trained "Trajairnet" model originally developed by Patrikar, Moon, Ghosh, Oh, and Scherer (2021). Previous research by Le et al. (2020) highlights the potential of deep learning techniques for spatio-temporal tasks, such as trajectory prediction. Building on these insights, I incorporated synthetic data integration and uncertainty estimation techniques to enhance the model’s ability to generalize across a wide range of flight conditions, improving its handling of OOD and non-IID scenarios.

Our project utilized a blend of real-world trajectory data—incorporating variables such as wind speed and direction—alongside synthetic data to train our predictive models. Earlier studies, such as Le et al. (2020) and Schimpf et al. (2023), highlight how integrating both synthetic and real datasets with advanced machine learning approaches can significantly improve prediction accuracy. Drawing from these insights, we applied a similar methodology, combining diverse datasets to enhance the adaptability of our models to the complex, dynamic nature of general aviation. My teammate, Andryian Saputra, made a significant contribution by developing a model for generating synthetic data that replicated real-world aviation scenarios, thereby adding data variability crucial for training.

As for my role, I concentrated on evaluating the performance of the pre- in predicting GA trajectories, with particular attention to out-of-distribution (OOD) and non-IID data. To address the OOD challenge, I implemented uncertainty estimation methods, allowing the model to capture its prediction confidence and identify cases where the input data diverged from the training set. Furthermore, I introduced a dataset with varied aircraft IDs to simulate OOD scenarios, facilitating a direct comparison with the in-distribution data.

## **Methods and their Limitations**

As outlined in the introduction, general aviation (GA) involves diverse and unpredictable flight operations that challenge traditional trajectory prediction models. The following part will review a detailed exploration of the state of the art, assessing how different methodologies align with the operational demands of general aviation. Start by giving an overview of the traditional approaches to trajectory prediction in general aviation or other similar fields.

### **Constant Velocity (CV) Model**

One of the most frequently adopted models for predicting trajectories is the Constant Velocity (CV) model. This approach operates under the assumption that the object maintains a steady speed and direction over time, making it an attractive option for trajectory prediction due to its simplicity and efficiency. The CV model's straightforward design has led to its widespread application in fields like visual object tracking, forecasting pedestrian movement, and monitoring aerial activities. Its ease of implementation is one of its major strengths (Baisa, 2020).

Despite its practicality, the CV model falls short in more dynamic and unpredictable contexts, such as general aviation (GA). Aircraft operating in GA often undergo fluctuations in speed, altitude, and direction, driven by external variables like weather shifts, air traffic, or required course adjustments. In these instances, the assumption of a constant velocity proves inadequate, as the model struggles to reflect rapid changes in the aircraft’s movement.

Additionally, the CV model exhibits weaknesses when dealing with out-of-distribution (OOD) or non-independent and identically distributed (non-IID) data. General aviation flights frequently encounter irregular flight patterns or novel situations that are not present in the model’s training data. This can lead to significant prediction errors, especially in scenarios like low-altitude flight, where factors such as wind shear or terrain navigation come into play. Baisa (2020) also mentioned that the CV model's overly simplistic approach can result in inaccurate trajectory forecasts, limiting its effectiveness in real-world GA applications.

However, despite these drawbacks, the CV model remains a valuable baseline for comparing the performance of more advanced prediction models. Its simplicity offers a minimum standard of predictive accuracy that other models must surpass, especially when balancing computational demands and the need for real-time responsiveness. Despite these limitations, the CV model serves as a useful baseline for comparing more advanced models. It provides a minimal level of predictive accuracy that more complex models are expected to surpass. The challenge lies in developing models that can handle the dynamic and unpredictable nature of GA, without compromising computational efficiency or real-time performance.

The Kalman Filter assumes that both the system's motion and observation models are linear. The equations used in these models can be expressed as follows:

Position Update Equation:

Where:

* is the position of the object at time step .
* is the position of the object at the previous time step .
* is the time difference between two consecutive steps.
* is the velocity of the object at time .

Velocity Update Equation:

Where:

* is the velocity of the object at time step , which remains constant over time.

### **Kalman Filter**

The Kalman Filter (KF) is a widely used recursive method designed to estimate the state of a dynamic system based on noisy observations. Its strength lies in systems where linear assumptions hold and the noise is Gaussian. The KF works through two main steps: first, during the prediction phase, it generates an estimate of the system's future state using the previous state and any control inputs. Then, in the correction phase, this estimate is adjusted by incorporating new measurements to reduce the overall uncertainty (Im, 2024).

Although the Kalman Filter is popular in many applications, it has notable drawbacks, especially when dealing with non-linear systems, such as those found in general aviation. According to Im (2024), the KF relies on assumptions that both the system dynamics and observation models are linear, and that noise follows a Gaussian distribution—assumptions that are often violated in real-world settings. General aviation flights, for instance, typically involve non-linear behavior, such as abrupt turns, changes in altitude, and unpredictable weather patterns. In such cases, the filter’s performance declines, resulting in less precise predictions.

To address these challenges, variants like the Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) were developed. These models are designed to better handle non-linear systems: the EKF uses a linear approximation around the current estimate, while the UKF applies sampling techniques to approximate non-linear dynamics more accurately. Although these improvements make the filters more suitable for complex environments like general aviation, they also increase computational demands and may still fall short in highly dynamic scenarios.

Despite these limitations, the Kalman Filter remains an essential tool in trajectory prediction tasks. It serves as a baseline against which more advanced models are often compared, thanks to its simplicity and computational efficiency.

The Kalman Filter equations are split into two phases:

1. Prediction Phase:

Where:

* is the predicted state.
* is the predicted covariance matrix.
* is the state transition matrix.
* is the process noise covariance.

1. Correction Phase:

Where:

* is the Kalman Gain.
* is the observation.
* is the observation matrix.
* is the observation noise covariance.

### **Particle Filter**

The Particle Filter is a non-parametric Bayesian method that estimates the posterior distribution of a system's state by employing a set of weighted particles. Unlike parametric filters, this approach excels in scenarios where the noise is non-Gaussian and the system's dynamics are non-linear, offering greater flexibility compared to filters such as the Kalman Filter (Jasra et al., 2024)​. The Particle Filter works by propagating a set of particles, with each particle representing a potential state of the system. These particles are then sequentially updated, and their importance weights—indicative of how well each particle fits the observed data—are used to re-sample the particle set at every time step.

One of the major strengths of the Particle Filter is its ability to handle non-linear state transitions and measurement models, which is particularly useful in general aviation (GA). GA flights frequently encounter non-linear conditions, such as turbulence, sudden shifts in wind, or rapid changes in altitude, where linear assumptions break down. As noted by Jasra et al. (2024), the Particle Filter can manage multi-dimensional diffusion processes, making it an appropriate choice for modeling complex systems like GA.

Despite its strengths, the Particle Filter is not without its challenges, especially in the domain of trajectory prediction for general aviation. A key limitation is that achieving high accuracy often requires a substantial number of particles, particularly in high-dimensional systems, which significantly raises computational demands. Another issue is particle degeneracy, where over time, most particles receive very low weights, diminishing the filter's ability to accurately track the system's true state.

The Particle Filter involves two main steps:

1. Prediction: Propagate the particles based on the system's dynamic model.

Where:

* is the -th particle at time step .  
   represents process noise.

1. Update: Compute the importance weights of each particle based on the observation model.

Where:

* is the weight of a particle at time .
* is the likelihood of the observation given the particle .

### **Long Short-Term Memory (LSTM) network**

The Long Short-Term Memory (LSTM) network is a specialized form of recurrent neural network (RNN) designed to handle sequential data by effectively capturing long-term dependencies. This makes it particularly advantageous for tasks like trajectory prediction. Traditional RNNs often struggle with the vanishing gradient problem, which hinders their ability to learn from long-term sequences. LSTMs, however, overcome this challenge through a gating mechanism that allows them to retain important information across time steps and discard less relevant data. This mechanism significantly enhances their ability to predict outcomes in complex sequential datasets, such as flight trajectories (Burgueño et al., 2021)​.

As highlighted by Burgueño et al. (2021), LSTM networks have been successfully applied in various spatio-temporal tasks, including trajectory forecasting, time series analysis, and even speech recognition. In the context of general aviation (GA), where flights experience highly variable conditions and frequent changes in velocity or direction, LSTMs provide a powerful tool for analyzing sequential flight data. Their capacity to capture non-linear relationships and long-term dependencies gives them a distinct advantage over simpler methods like the Constant Velocity (CV) model in trajectory prediction​.

However, despite their strengths, LSTM networks are computationally intensive and demand significant amounts of training data. The complexity inherent in their architecture, with multiple layers and memory cells, also increases the risk of overfitting, unless appropriate regularization techniques are used. Additionally, LSTM models can struggle with out-of-distribution (OOD) data, where novel flight patterns or unexpected environmental conditions may lead to reduced performance. Addressing these challenges is essential to enhance their effectiveness in real-world general aviation trajectory predictions.

A diagram of a network

Description automatically generated

***Figure 1:*** *A generic LSTM-based neural network architecture (Burgueño, 2021).*

### **Spatio-Temporal Graph Neural Networks (ST-GNNs)**

Spatio-Temporal Graph Neural Networks (ST-GNNs) represent a class of models designed to capture both spatial relationships and temporal patterns within data, making them particularly useful for analyzing time series data that can be represented as graphs. These networks have proven effective in tasks such as traffic forecasting, where the road network forms a natural graph and traffic conditions evolve over time (Yu et al., 2018)​.

ST-GNNs employ graph convolutional layers to model the spatial dependencies between nodes, while temporal layers—either through temporal convolutions or recurrent structures—track changes over time. This hybrid architecture enables the network to learn intricate spatio-temporal interactions, resulting in more precise predictions compared to conventional time series approaches. For instance, in traffic prediction, ST-GNNs can forecast future traffic states by analyzing both the current status of neighboring roads (spatial information) and the progression of traffic over time (temporal information). As noted by Yu et al. (2018), this integrated approach allows the model to outperform traditional methods like ARIMA and LSTM, particularly in capturing complex, non-linear spatio-temporal patterns​.

Nevertheless, despite their powerful capabilities, ST-GNNs still have certain challenges. These models require significant computational resources and large datasets for effective training. Additionally, they may struggle with out-of-distribution (OOD) data, where previously unseen spatial configurations or temporal patterns lead to reduced performance.

A diagram of a computer system

Description automatically generated

***Figure 2****: Architecture of spatio-temporal graph convolutional networks (Yu et al., 2018).*

## **Environmental Factors**

After discussing all the methods in previous sections, we must understand the importance of environmental factors when talking about trajectory predictions. For predicting trajectories, we cannot purely predict it based on historical data

Environmental factors play a critical role in trajectory prediction for general aviation (GA). Unlike commercial airline operations, GA flights frequently operate in diverse and unpredictable conditions. These flights often navigate through varying altitudes, weather patterns, and airspaces that are not as standardized or controlled as commercial routes. This introduces substantial challenges for predictive models, as environmental variables such as wind speed, turbulence, visibility, and precipitation directly influence the trajectory of an aircraft (Gultepe et al., 2019)​

Weather conditions are among the most significant environmental factors affecting trajectory prediction. Changes in weather—such as sudden storms, shifts in wind direction, and low visibility—can drastically alter a flight path, making accurate predictions difficult. Turbulence and wind shear, for example, can result in sudden altitude or directional changes that standard predictive models like Constant Velocity (CV) or Kalman Filters cannot easily accommodate (Kim et al., 2024)​. These environmental conditions often occur unexpectedly, leading to out-of-distribution (OOD) situations where the model is forced to make predictions based on data it has not encountered during training.

Furthermore, GA operations are often subject to non-independent and identically distributed (non-IID) data conditions. In contrast to commercial aviation, where flight paths and environmental factors are relatively more predictable, GA involves highly varied routes and weather patterns. For instance, flights at lower altitudes may face terrain-induced wind shifts and other microclimatic conditions, while flights across mountainous regions may experience sudden updrafts or downdrafts (Gultepe et al., 2019)​. These factors result in data that is not uniformly distributed, causing challenges for traditional machine learning models that expect IID data. As a result, the predictive accuracy can drop when these models face data that deviates from the conditions encountered during their training phase.

Given these complexities, addressing environmental factors becomes crucial for the development of reliable predictive models. One approach involves weather data integration. By incorporating real-time weather information, such as wind velocity, precipitation, and temperature into the model, predictive accuracy can be improved, allowing models to adjust predictions based on real-time weather inputs (Kim et al., 2024)​.

## **OOD and non-IID**

After explaining the importance of environmental factors in trajectory prediction, it becomes clear that one of the greatest challenges predictive models face in general aviation (GA) is dealing with out-of-distribution (OOD) and non-independent and identically distributed (non-IID) data.

In trajectory prediction models, particularly in domains such as general aviation (GA), the ability to handle out-of-distribution (OOD) and non-independent and identically distributed (non-IID) data is crucial. OOD refers to scenarios where the model encounters input data during testing that deviates significantly from the data it was trained on, while non-IID data reflects real-world conditions where data points are not identically distributed or independent from one another (Zhao & Cao, 2024)​. In GA, aircraft frequently operate in conditions or routes that are not presented in the training data, such as varying weather patterns, unexpected altitudes, and sudden course changes. These factors can significantly affect the accuracy of predictions, as models trained on typical scenarios may fail to generalize to these new situations.

Handling OOD data is essential for enhancing the reliability of predictive models. In practical applications, models that cannot identify and appropriately handle OOD data may make overly confident and inaccurate predictions, resulting in potential safety risks (Lu et al., 2024)​. For instance, in low-altitude general aviation flights, unpredictable environmental conditions such as wind shear or sudden terrain obstacles may lead to situations that deviate from the trained data distribution. To mitigate these risks, OOD detection techniques are employed to flag such scenarios, allowing the system to defer to more cautious decision-making, such as requesting human intervention or adjusting the prediction model's behavior (Rubinstein et al., 2024)​.

Another related challenge is the handling of non-IID data, which occurs when the data points within the dataset do not follow the assumption of independence and identical distribution. In GA, flights often experience sequential dependency in their flight patterns, which violates the IID assumption commonly made by traditional machine learning models (Zhang et al., 2023). This issue is particularly pronounced when flights operate in highly dynamic environments where consecutive data points are influenced by external factors like weather conditions or air traffic. As noted by Zhao and Cao (2024), treating non-IID data as IID can cause models to misinterpret these dependencies, leading to inaccurate predictions during test phases.

Addressing these challenges requires incorporating uncertainty estimation techniques or ensemble learning methods. These methods enhance model robustness by allowing predictions to account for unknown data points or variations. For example, weighting techniques like Weighted non-IID Batching (WNB) adjust the training process to account for the discrepancies between batch samples and the full dataset, improving model sensitivity to both OOD and non-IID data (Zhao & Cao, 2024)​. Additionally, diversified ensemble learning, which encourages models to disagree on hard samples, has proven effective in improving generalization across OOD test cases and complex data structures, particularly in the aviation domain (Rubinstein et al., 2024)​.

## **Motivation**

Building on this motivation, it is essential now that we understand the direct impact of environmental factors on the predictability of general aviation trajectories. The unpredictable nature of weather conditions, such as wind shear, turbulence, and sudden altitude shifts, further complicates the trajectory prediction process. These environmental variables not only introduce out-of-distribution (OOD) situations, where models encounter data that deviates from the training set, but also highlight the presence of non-independent and identically distributed (non-IID) data patterns in sequential flights.

To enhance the safety and efficiency of air traffic management systems, predicting flight trajectories is the key component. Although trajectory prediction models have been widely implemented in commercial aviation and urban traffic management, their application in general aviation (GA) remains limited due to the unique challenges posed by GA operations. GA covers a broad range of aircraft, such as private planes and business jets, which often navigate non-standardized airspaces and experience varying weather and altitude conditions (Wang et al., 2021). These characteristics make traditional models, like the Constant Velocity (CV) model and Kalman Filter (KF), less effective, as these methods are designed for linear dynamics and struggle with the sudden shifts in speed, altitude, and direction that are common in GA flights (Im, 2024).

The shortcomings of existing models are worsened by their inability to handle out-of-distribution (OOD) and non-independent and identically distributed (non-IID) data. In GA, unanticipated events such as turbulence, sharp turns, or abrupt environmental shifts often challenge conventional predictive models. This is particularly problematic during low-altitude operations, where obstacles like terrain must be carefully navigated. As a result, there is an urgent need for a more sophisticated model that can move beyond linear assumptions and address the complexities inherent in GA trajectories. Solving these issues is crucial for advancing safety and operational efficiency in GA.

While machine learning models like Long Short-Term Memory (LSTM) networks and Spatio-Temporal Graph Neural Networks (ST-GNNs) have shown promise in capturing complex, non-linear patterns and spatio-temporal relationships, they face significant challenges when confronted with out-of-distribution (OOD) and non-independent and identically distributed (non-IID) scenarios. LSTMs are designed to capture long-term dependencies in sequential data, which improves prediction accuracy in many domains. However, in the context of general aviation (GA), where flight conditions can vary significantly due to factors such as weather anomalies or emergency maneuvers, LSTMs struggle to generalize effectively. Because they rely on previously observed sequential patterns, these models often fail when faced with unfamiliar conditions not present in the training data, leading to inaccurate or unreliable predictions (Burgueño et al., 2021).

Similarly, ST-GNNs, which are adept at modeling dynamic spatial and temporal relationships—such as those between aircraft—also encounter limitations with OOD data. These models rely on the assumption that the spatio-temporal dependencies learned during training will remain consistent in real-world applications. However, this assumption is often violated in general aviation, where non-IID factors, such as sudden changes in air traffic or unpredictable environmental conditions, undermine the model’s ability to generalize (Yu et al., 2018). As a result, ST-GNNs may generate suboptimal predictions when confronted with these unexpected situations, highlighting the need for more robust models that can adapt to OOD data and effectively manage the inherent unpredictability of GA.

This project aimed to address these challenges by applying advanced machine-learning techniques from the pre-trained model to research flight trajectory prediction in general aviation. By focusing on the model's robustness and addressing the issues related to OOD and non-IID data, the project sought to contribute to safer and more efficient flight management systems. This project intended to offer inspiration to the aviation industry, helping to improve air traffic control and safety through real-world applications, and advancing the current capabilities of trajectory prediction models in the future based on the findings.

# **Objective – The Originality**

In the previous section, I identified the limitations of traditional models in handling complex and unpredictable GA flight paths. To address these issues, this project focused on improving trajectory prediction by confronting challenges related to out-of-distribution (OOD) and non-independent, non-identically distributed (non-IID) data. Traditional models like the Kalman Filter (Im, 2024) and Constant Velocity (CV) model (Baisa, 2020) often perform well in controlled environments. However, they struggle to adapt to the highly variable conditions of GA operations. GA flights frequently encounter diverse weather patterns, fluctuating altitudes, and irregular flight paths, which makes accurate prediction difficult for such models (Wang et al., 2021).

My approach focused on refining a pre-trained trajectory prediction model. One of the main improvements involved the implementation of uncertainty estimation techniques, which allowed the model to assess its confidence in predictions. This made it possible to identify OOD instances and adjust the model's predictions accordingly. As a result, the model’s capacity to generalize to unfamiliar conditions and flight patterns significantly improved. Uncertainty estimation played a critical role in distinguishing between in-distribution and OOD data, reducing the chances of incorrect predictions.

Additionally, my teammate Andryian Saputra led the development of synthetic data generation. This synthetic data simulated the unpredictable conditions of GA operations by introducing controlled variations. Incorporating synthetic data into the training process exposed the model to a wider range of flight scenarios. This enhanced its ability to generalize when confronted with non-IID data, making the model more adaptable to real-world aviation conditions.

## **Key Achievements**

Throughout the course of this project, several key milestones were achieved, each of which played an essential role in improving the model’s ability to predict complex General Aviation (GA) flight trajectories. One of the most important achievements was the refinement of the pre-trained models. By fine-tuning these models, I ensured that they could better handle out-of-distribution (OOD) and non-independent, non-identically distributed (non-IID) data, which are prevalent in GA operations. This fine-tuning enabled the model to capture the dynamic and unpredictable nature of real-world GA flight paths more accurately.

The successful integration of synthetic data was another pivotal achievement. Synthetic flight data introduced controlled variations, simulating a wide range of GA flight scenarios that were difficult to capture with real-world data alone. This significantly enhanced the model’s generalization capabilities, improving its performance when dealing with OOD data. By exposing the model to a broader array of flight conditions, I was able to ensure greater predictive accuracy, even in unfamiliar situations.

Additionally, I implemented uncertainty estimation techniques, particularly Maximum Softmax Probability (MSP), to quantify the model’s confidence when making predictions in both in-distribution (ID) and OOD situations. This provided a reliable means for the model to express uncertainty, particularly when encountering data outside its training distribution. The inclusion of this technique reduced the risk of overconfident but incorrect predictions, supporting the model’s robustness when faced with unexpected inputs.

Visualizations such as training and test loss curves, alongside 3D scatter plots of predicted flight paths, were critical in assessing the model’s learning progression and its ability to handle both ID and OOD data. These visual tools helped identify overfitting or underfitting, providing valuable insight into how the model could be further refined for real-world GA applications. Ultimately, these achievements significantly contributed to the model's ability to adapt to GA’s complex, ever-changing environments.

## **2.2 Workflow**

This project followed a detailed workflow, consisting of several critical phases aimed at refining the pre-trained model and integrating synthetic data to maximize its effectiveness. Each phase played a pivotal role in constructing a robust trajectory prediction model capable of addressing out-of-distribution (OOD) and non-independent and identically distributed (non-IID) data challenges.

1. Data Processing:  
   The initial phase involved gathering and processing the real-world general aviation dataset. This included data cleaning, reformatting, and preprocessing to ensure the dataset was suitable for model integration. Handling missing or inconsistent data points was crucial to ensure the dataset’s reliability for the model’s trial run.
2. Pre-trained Model Trial Run:  
   Once the data was prepared, I ran the pre-trained model as an initial test. This trial run generated baseline performance metrics, providing us with an early evaluation of the model’s strengths and limitations using the real-world dataset. The results guided us on areas for improvement before further tuning.
3. Fine-tuning and Additional Enhancements:  
   Using the insights from the trial run, I fine-tuned the model to improve its performance. Both supervised and unsupervised learning techniques were incorporated to help the model better handle complex patterns in the data. We also introduced uncertainty estimation techniques to enhance robustness, especially in OOD cases. To aid understanding of how the model processed unseen data, we developed visualization methods for the unsupervised learning outcomes.
4. First Evaluation:  
   Following the fine-tuning process, I conducted the first formal evaluation. This involved a detailed comparison between the updated model’s performance and the baseline results from the trial run. The analysis, based on quantitative metrics and visualized graphs, showed improvements in predictive accuracy, model confidence, and generalization to unfamiliar flight conditions.
5. Synthetic Data Integration:  
   After the first evaluation, I began incorporating synthetic data generated by my partner, Andryian Saputra. This phase involved modifying the model to effectively utilize the synthetic data and integrate it with the real-world dataset. The synthetic data introduced new, previously unseen variations, testing the model’s adaptability to novel scenarios and its capacity to handle non-IID data.
6. Second Evaluation:  
   With the synthetic data incorporated, I carried out a second evaluation to assess how well the model performed with the expanded dataset. This phase focused on analyzing how the model adapted to the synthetic data alongside the real-world information. The results demonstrated the model's improved ability to predict under unfamiliar or OOD conditions, highlighting the positive impact of synthetic data on enhancing prediction reliability in general aviation contexts.

Workflow


**Figure 3:** Project Workflow

# **Project Preliminary**

Before diving into the implementation phase of our project, it is essential to lay out the foundational aspects that construct our approach. This section offered a comprehensive overview of the dataset, pre-trained models, evaluation metrics, and ethical considerations that shape the trajectory of our work. Understanding these elements is crucial as they provide insight into the resources, methods, and assessment tools we use to explore trajectory prediction for general aviation.

The Data Overview section (3.1) outlined the dataset used in the project, including the raw flight data and the preprocessing techniques applied to prepare it for modeling. Preprocessing involves cleaning, normalization, and integrating relevant environmental factors such as weather data. Following this, an exploration of the processed data is conducted to extract meaningful data columns, which helped guide the training process.

In Pre-trained Model Overview (3.2), we leap into the architecture of the “Trajairnet” pre-trained model, which served as the baseline for our work. This included a detailed discussion of the components of the model, such as Temporal Convolutional Networks (TCN), Graph Attention Networks (GAT), and Conditional Variational Autoencoders (CVAE). These components are crucial for handling the spatio-temporal complexity of general aviation trajectory prediction.

The Evaluation Criteria section (3.3) explained the metrics and methods used to assess model performance and to measure estimated uncertainty for possible OOD elements. I introduced metrics such as Average Displacement Error (ADE) and Final Displacement Error (FDE), which measure the accuracy of predicted trajectories, and distance.

Lastly, in Ethics and Privacy (3.4), I addressed the ethical considerations of working with flight data, particularly ensuring privacy and compliance with relevant regulations. This section emphasizes the importance of using the data and pre-trained models responsibly, for demonstration and research purposes only. The final part, Expected Schedule (3.5), outlines a timeline for completing the project tasks, from data processing and model fine-tuning to evaluation and report writing.

## **Data Overview**

The dataset used in this project is sourced from “TrajAir”, a publicly available dataset designed for the prediction and analysis of general aviation (GA) flight trajectories (Patrikar et al., 2021). The dataset offered both real-world GA trajectories and weather data that play a critical role in the performance of my predictive models. The dataset contains a total of 111 days of flight data, providing a broad range of general aviation trajectories.

The raw data consists of flight trajectories recorded through the Automatic Dependent Surveillance-Broadcast (ADS-B) system. Each record in the raw data includes the following key attributes:

* Aircraft ID: A unique identifier for each aircraft.
* Position Coordinates (x, y, z): These represent the aircraft’s location in 3D space, recorded in kilometers.
* Wind Speed (wind\_x, wind\_y): Wind conditions are recorded in meters per second, providing insight into the environmental factors affecting the aircraft's trajectory.
* Frame Number: A sequential number representing each time step at 1 Hz intervals.

Additionally, the dataset includes weather data, which provides environmental context for each flight. This information includes:

* Temperature
* Wind Direction and Speed
* Precipitation Levels

These environmental variables are integrated, which can be useful in the synthetic data generation process and enhances the model’s ability to predict the behavior of aircraft in a dynamically changing environment, which is particularly relevant for general aviation operating under unpredictable conditions.

### **Preprocessing**

After reviewing the raw data, a series of preprocessing steps were undertaken before being used in model training. These steps included:

* Data Cleaning: Removing any missing or inconsistent values to maintain data integrity.
* Normalization: Transforming and scaling the x, y, z coordinates and wind speed values to ensure uniformity across all features.
* Weather Data Integration: The weather data, captured in separate files, includes environmental factors such as temperature, precipitation, and wind conditions. I eventually chose wind speed of x and y as factors because of their consistency. These factors are integrated into the trajectory data to account for the influence of environmental conditions on flight patterns.

A black and white image of a speed limit

Description automatically generated

**Figure 4:** Preprocessing the data

Some data exploration on the whole processed data was done. See the appendix for “Processed Data Exploration”. Due to the extended training time required for processing such a large dataset, the focus has primarily been on the first two weeks, represented by “7days1” and “7days2”. This allows me to efficiently train and fine-tune the model while ensuring it is exposed to a representative sample of the broader dataset.

## **Pre-trained Model Overview**

The “Trajairnet” pre-trained model served as the baseline for this project, and its design incorporated both historical aircraft trajectory data and environmental factors to make accurate predictions. Below is a detailed overview of its architecture.

### **Architecture**

The model integrates several advanced machine learning components to predict future trajectories of aircraft operating in general aviation airspace. Its architecture includes the following key components:

1. Temporal Convolutional Network (TCN):

This module processes time-series data from previous trajectory points. The TCN ensures that the model captures temporal dependencies in the movement of aircraft, such as speed, direction, and altitude changes over time.

1. Graph Attention Network (GAT):

The GAT is used to model interactions between multiple aircraft within a shared airspace. It allows the model to consider the influence of nearby aircraft on a particular aircraft’s trajectory, improving predictions in crowded airspace situations.

1. Conditional Variational Autoencoder (CVAE):

This part of the model handles uncertainty in trajectory predictions by generating multiple possible future trajectories, rather than a single deterministic path. The CVAE’s encoder-decoder structure helps model the inherent variability in aircraft behavior, particularly under unpredictable conditions.

1. Environmental Data Integration (CNN for Wind Components):

The CNN processes environmental data, specifically wind speed and direction, which are integrated into the TCN and GAT. CNNs are particularly effective at extracting patterns from grid-like data, such as wind maps. This modular architecture allows the model to process various inputs, including temporal, spatial, and environmental factors, providing a comprehensive prediction of future trajectories in general aviation contexts.

A diagram of a graph

Description automatically generated

**Figure 5**: Structure of TrajAirNet model (Patrikar et al., 2022).

In summary, the model integrates several key components to provide robust trajectory predictions. By combining Temporal Convolutional Networks (TCN) for time-series data and Graph Attention Networks (GAT) for spatial relationships, the model captures both temporal and spatial patterns in the data. Additionally, the Conditional Variational Autoencoder (CVAE) handles uncertainty, ensuring the model remains adaptable to unpredictable situations. This architecture ensures the model can capture complex patterns in general aviation trajectories while accounting for real-world variability and uncertainty. This integrated approach enables the model to adapt to the unpredictable nature of general aviation, making it a suitable foundation for our project’s trajectory prediction tasks.

## **Evaluation Criteria**

To accurately assess the performance of our trajectory prediction model and compare it with the baseline, we must apply a variety of evaluation metrics that consider both spatial accuracy and the model's ability to generalize across different scenarios. These metrics are crucial in understanding how well the model performs when encountering familiar as well as out-of-distribution (OOD) data. Below, we discussed the key metrics selected for this project and their relevance to the model’s evaluation.

### **Training and Testing Metrics**

One of the key metrics I used is Mean Squared Error (MSE), which measures the average squared difference between predicted and actual values during training. MSE was chosen because it effectively penalizes large errors more than smaller ones, providing a smooth gradient for optimization and helping the model focus on reducing larger errors. In the context of trajectory prediction, minimizing MSE helps the model learn to align predicted and actual trajectory points more accurately. This metric is also well-suited for continuous, numerical data, which is the nature of our problem.

Another important metric used is Average Displacement Error (ADE), which measures the average Euclidean distance between predicted and actual points over the entire trajectory. ADE captures the overall accuracy of the predicted path, making it an ideal choice for trajectory prediction tasks where every point along the sequence is important. By focusing on the entire trajectory, ADE offers insight into the model's ability to track flight paths consistently across multiple time steps.

In addition to ADE, I used Final Displacement Error (FDE), which focuses specifically on the distance between the predicted final point and the actual final point of the trajectory. While ADE gives an overview of the model’s accuracy across the sequence, FDE provides crucial information about the endpoint prediction, which is critical for real-world applications such as landing prediction in aviation. FDE ensures that the model not only predicts intermediate steps accurately but also lands near the correct location.

Several common metrics, such as Accuracy, were considered but ultimately assumed unsuitable for our task. Accuracy is a standard metric in classification problems, but trajectory prediction is a regression problem where the model outputs continuous coordinates rather than class labels. Accuracy would not adequately capture the nuance of predicting precise flight paths, particularly in out-of-distribution scenarios where fine deviations matter more than overall classification correctness.

Similarly, ROC-AUC (Receiver Operating Characteristic - Area Under Curve), another widely used metric for evaluating classification models, was not appropriate for this task. ROC-AUC measures the model's ability to distinguish between classes, which is not relevant for continuous trajectory prediction. While valuable in binary or multi-class classification tasks, it does not offer useful insights when dealing with spatial predictions and time-series data.

I also evaluated the potential use of Precision and Recall, which are commonly used in classification models to measure the rate of correct positive predictions. However, like accuracy and ROC-AUC, these metrics focus on classification tasks and do not align well with our objective of predicting precise coordinates over time. Precision and recall are more suited for binary outcomes, whereas our task requires continuous and accurate trajectory estimation.

A diagram of a graph

Description automatically generated

***Figure 6:*** *FDE and ADE demonstration (Kumichev, 2024).*

### **Uncertainty Estimation Metrics**

The key metric I employed for uncertainty estimation was Maximum SoftMax Probability (MSP), which measures the highest SoftMax probability output from the model’s predictions. MSP was chosen for its simplicity and effectiveness in capturing the model’s confidence in the prediction test. In trajectory prediction tasks, where the output involves forecasting future positions, MSP acts as a proxy for uncertainty by indicating how certain the model is about its most confident prediction. High SoftMax probabilities suggest the model is confident, while lower values indicate uncertainty in the prediction, signaling that the model may be encountering unfamiliar or out-of-distribution (OOD) data.

MSP was also selected for its computational efficiency. Unlike more complex uncertainty estimation techniques—such as Bayesian neural networks or ensemble methods—MSP does not require major changes to the model architecture or significant computational overhead. Since MSP is directly computed from the softmax output layer, it is a lightweight and practical solution for uncertainty estimation. This makes it ideal for real-time applications like trajectory prediction, where computational speed and cost are crucial.

While other uncertainty estimation techniques, such as Monte Carlo Dropout or Deep Ensembles, can offer more robust uncertainty estimates, MSP was chosen for its simplicity and ease of integration into the existing model. These more advanced techniques typically come with higher computational costs, which can be a bit excessive for large-scale or time-sensitive tasks like ours. MSP strikes a balance by offering valuable insights into uncertainty without compromising the computational efficiency needed for real-time trajectory prediction.

## **Ethics and Privacy**

The development of machine learning models for trajectory prediction in general aviation (GA) requires careful consideration of ethical and privacy concerns. Working with sensitive flight data and generating predictions about real-world movements necessitates responsible data use, storage, and sharing, as well as attention to the ethical implications associated with the model’s predictions.

It is important to emphasize that the data and pre-trained models used in this project are for demonstration purposes only and are not intended for real-world applications. The data and model were originally developed by Patrikar, J., Moon, B., Ghosh, S., Oh, J., and Scherer, S. (2021), in their research on aircraft trajectory prediction. Their work forms the foundation of this project, and we fully acknowledge their ownership of the original dataset and the pre-trained model.

**Privacy** is a critical concern, even in a demonstration context. All data used in this project has been anonymized to ensure that no identifiable information about individual pilots, aircraft, or operators can be traced. As Riyana (2024) points out, privacy-preserving methods are essential when handling sensitive or potentially identifiable data. While this project used anonymized and demonstrative data, it still follows industry best practices for privacy protection and data security.

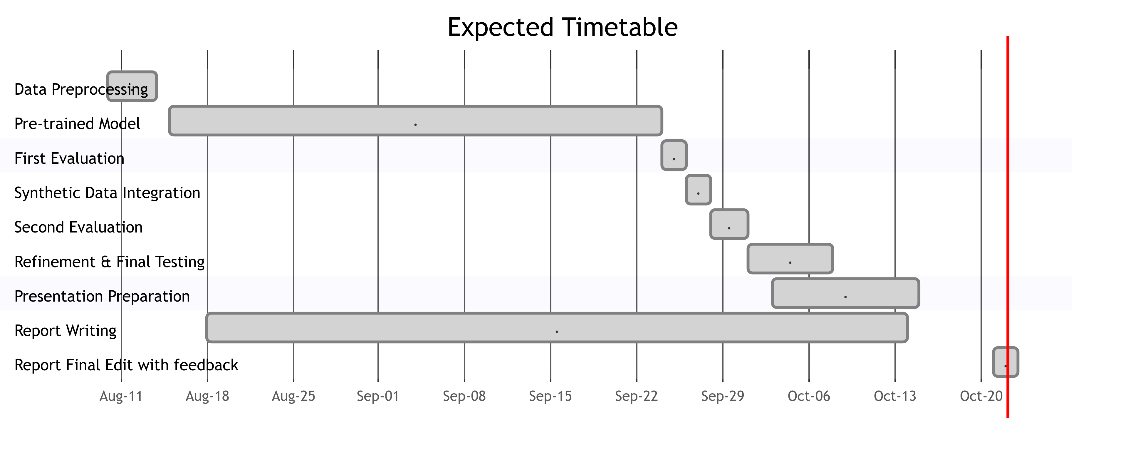
From an **ethical** standpoint, transparency and fairness are key considerations in the development of predictive models. In aviation, models must provide interpretable and reliable predictions to enable human operators to make informed decisions. As Xi et al. (2024) highlight, privacy-preserving techniques must strike a balance between data utility and ethical usage, ensuring the accuracy of predictions while safeguarding privacy (Xi et al., 2024). In this project, the model is designed to aid decision-making, not to replace human judgment. The responsibility for final decisions remains with pilots and air traffic controllers, maintaining full human oversight over flight operations.

In summary, the data and pre-trained model used in this project are strictly for demonstration purposes and have no real-world application. Although the project is based on the work of Patrikar et al. (2021), it adheres to the correct ethical standards by anonymizing data, ensuring transparency, and protecting the privacy and safety of all aviation-related operations.

## **Time Schedule (need update)**

The timeline illustrates the key phases of the project from August 10th to November 6th, highlighting the progression of tasks involved in developing the trajectory prediction model and report preparation. Each phase builds on the previous one, starting with data preprocessing, followed by the evaluation of the pre-trained model, the integration of synthetic data, and culminating in model refinement and final testing, before completing the report and final editing process.

**Table 1:** Timetable (created by Draw.io)



# **Pre-trained Model Enhancement (Phase 1)**

Phase 1 of this project centered on improving the predictive capabilities of the pre-trained "Trajairnet" model, specifically focusing on the non-linear and non-IID (non-Independent and Identically Distributed) nature of GA trajectories. I began by fine-tuning key hyperparameters to help the model adapt better to the unpredictable nature of GA flights, such as varying altitudes and weather conditions.

To further strengthen its predictive accuracy, I introduced uncertainty estimation techniques, which allowed the model to express its confidence in each prediction. This feature proved particularly useful in cases involving out-of-distribution (OOD) data, where the model might encounter unfamiliar flight patterns. These enhancements provided a baseline understanding of the model’s capacity to handle the complexities of GA flight trajectories, laying the groundwork for subsequent phases where more complex data would be introduced.

## **Trial Run**

For the initial trial run, I utilized the pre-trained model provided by the “Trajnet” project team (Patrikar, Moon, Ghosh, Oh, and Scherer, 2021), making minimal adjustments to ensure seamless integration with my operating system. In this initial phase, I focused on assessing the model's baseline performance to see what the model would output without any implements and evaluate the model’s inherent predictive capabilities.

The trial run lasted approximately 3 hours, with each epoch out of two requiring just over a half hour. The loss metrics captured during this phase serve as a reference for evaluating future iterations and enhancements to the model.

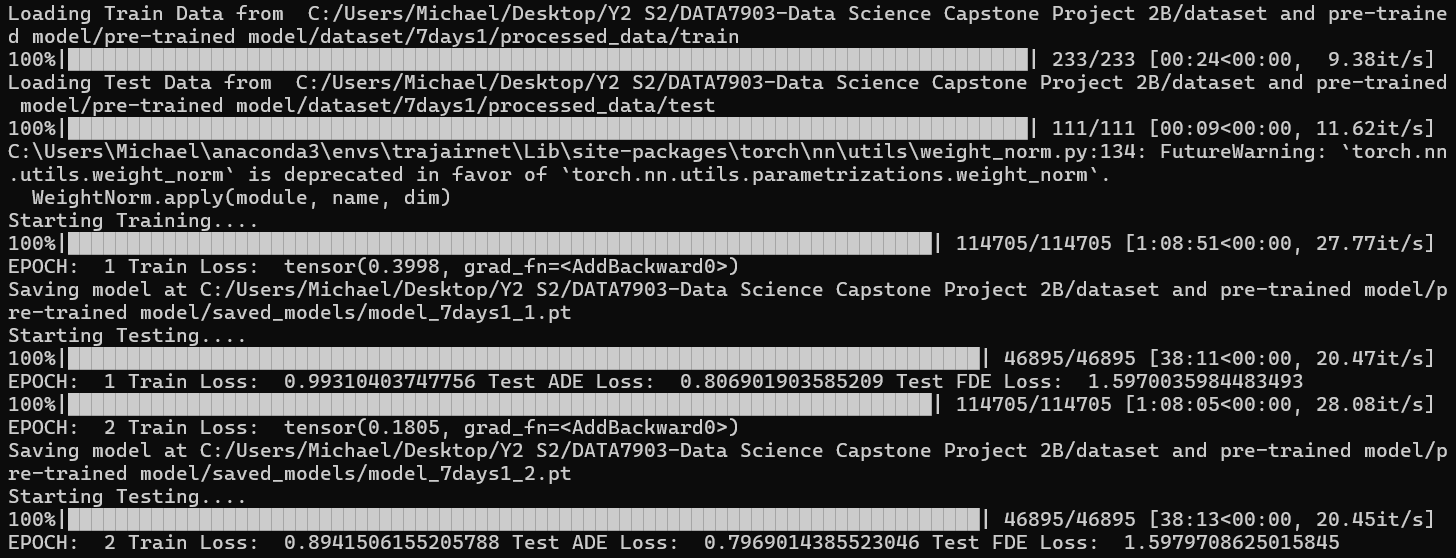
As we have seen in Table 2, the model’s performance exhibited some improvement across the two epochs during the training. Throughout the training process, instantaneous loss values (e.g., tensor (0.3998)) were recorded as the model processed mini-batches of data. These values provided real-time feedback on the model's learning progress but did not reflect the overall learning outcome for the epoch. Instead, the epoch-level loss is the key indicator of performance. For example, after the first epoch, the average training loss across all batches was 0.9931, which decreased to 0.8941 by the end of the second epoch. This reduction in training loss signifies the model’s ability to progressively minimize the discrepancy between the predicted and actual values in the dataset, indicating that it is learning the complex patterns present in the flight data.

When evaluating the model’s performance using the test set, two crucial metrics were examined: Average Displacement Error (ADE) and Final Displacement Error (FDE). ADE, which measures the average error across all points of the predicted trajectory, initially stood at 0.8069 after the first epoch and showed a slight improvement, decreasing to 0.7969 by the second epoch. Although the improvement in ADE was marginal, it still demonstrates that the model is making gradual progress in accurately predicting flight paths.

In contrast, the FDE, which focuses on the error at the final predicted point of the trajectory, remained unchanged across the two epochs. After the first epoch, the FDE was recorded at 1.5970, and this value persisted after the second epoch. This suggests that while the model is becoming more accurate in predicting the overall trajectory, it faces challenges in refining the prediction of the trajectory’s endpoint. This limitation highlights the need for further model fine-tuning or the introduction of advanced methods to improve accuracy at critical trajectory points.

In summary, these initial results established a baseline for future improvements. While the model demonstrated modest performance gains over two epochs, it exhibited the capacity to learn from the data and reduce errors. As we move forward, my efforts will be directed toward incorporating advanced techniques like uncertainty estimation and visualizations to address out-of-distribution (OOD) scenarios. These strategies should highlight the limitations observed in the baseline and ultimately result in more reliable and precise trajectory predictions.

**Table 2:** Summary table (Trial Run)



## **Adding Visualization**

The same trained “Trajairnet” model from the baseline section was employed for visualization, with no changes in the training process. Since the original model only shows training and testing loss, I added functions for supervised (real vs. predicted trajectories) and unsupervised (3D clustering) visualization. These functions help interpret the model’s prediction performance and cluster results in both supervised and unsupervised contexts.

### **Supervised Visualization: Real vs. Predicted Trajectories**

Figure 7 illustrates the real trajectories in green alongside the predicted trajectories in red for four different samples. While the predicted trajectories generally follow a linear pattern, slight deviations can be observed, especially towards the final points of each path. These initial samples were generated during the early stages of the implementation phase and are intended to demonstrate how the results appear at this stage.

The current model exhibited a tendency to produce predicted trajectories that closely follow a linear path. This linear behavior may be attributed to the limited timeframe set for the baseline model or the lack of sufficient training epochs. Additionally, there were noticeable deviations in the predicted endpoints when compared to the real trajectories, indicating that the model struggled to accurately capture the endpoints of the flight paths. These gaps suggest the model could benefit from further training adjustments to improve its precision in trajectory prediction, especially towards the final stages of the flight paths.

A graph of a graph of a function

Description automatically generated with medium confidence

**Figure 7:** Examples of 3D trajectory comparison samples

### **Unsupervised Visualization: 3D Clustering**

Figures 8 illustrate the clustering of real and predicted trajectories using 3D K-means clustering. The clustering was performed on a flattened version of the trajectory data. Interestingly, the clustering of the real trajectories appears to display a more structured spread, while the predicted trajectories are more condensed around a central point.

The clustering of the real trajectories reveals a wider distribution in 3D space, indicating that real flight paths exhibit greater variability. In contrast, the predicted trajectories form tighter, more compact clusters, highlighting the model’s tendency to produce more linear predictions. This suggests that the model, in its current state, may not fully capture the complexity and diversity present in real flight paths, likely due to its training limitations. The linear nature of the predictions reflects the need for further refinement to better handle diverse trajectory patterns.

To improve the model's performance, I planned to fine-tune it by increasing the number of training epochs, which allowed the model to capture more variability in trajectory patterns. These changes are expected to produce predictions that more closely resemble real trajectories, addressing the current limitations of overly tight clusters and linear predictions.

A graph of a graph of a graph

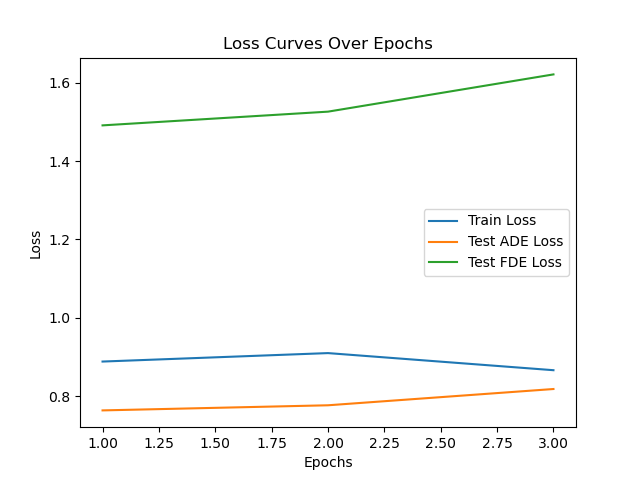
Description automatically generated with medium confidence

**Figure 8:** Example output of 3D trajectory clustering (pred vs real)

## **Addressing Overfitting and Underfitting (Loss Curve)**

I implemented the loss curve to illustrate how the model's training process was designed to handle overfitting and underfitting concerns. The curve provides a clear view of the training loss, as well as the test ADE and FDE losses, as training progresses across epochs. Observing the differences between the training and test losses allowed us to monitor whether the model was learning effectively without overfitting the training data. We will also add the loss curve comparison of in-distribution and OOD test datasets in the following result section.

The graph shows a relatively steady trend in both train and test losses in the pre-trained model, indicating that the model is not overfitting as there are no significant divergences between the training and test losses. Similarly, the flat progression of the test losses suggests that underfitting is also minimized, as the model consistently improves with each epoch. This analysis was vital in guiding further adjustments to the model to ensure a balanced fit to the data.

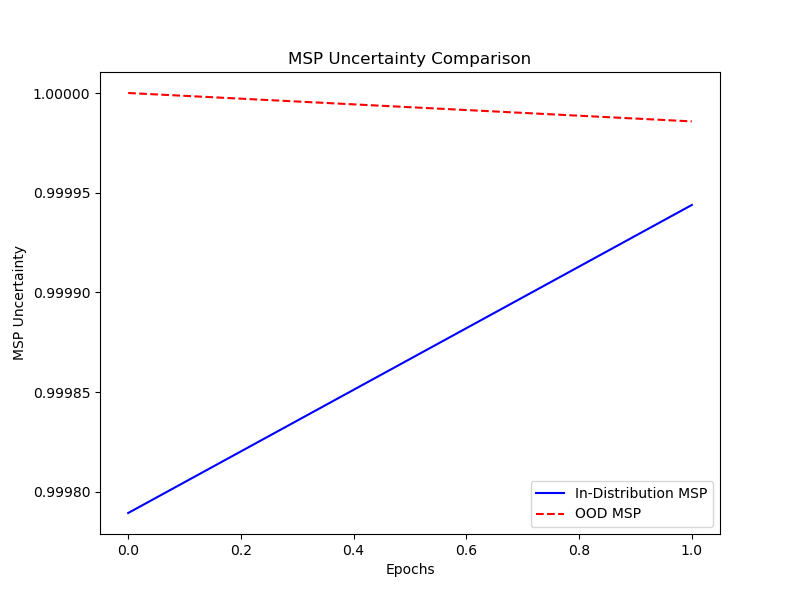


**Figure 9:** Example output of the training and test losses over epochs for trajectory prediction

## **Addressing OOD (Uncertainty estimation)**

To address the out-of-distribution (OOD) challenge, I trained the model using a specific airplane ID set (7days1) for the in-distribution data. To evaluate the model's performance on OOD data, I used another dataset with a different set of airplane IDs (7days2). This approach allowed the model to learn from one set of aircraft operations while testing its ability to generalize to unseen IDs.

Since the data was collected from an airport that handles a variety of general aviation flights, using different airplane IDs was a natural method to simulate OOD scenarios. This strategy helped demonstrate how well the model adapts to new and unfamiliar flight patterns, ensuring that it could effectively generalize beyond the training data in real-world applications.



**Figure 10:** Example output of the comparison between in-distribution (ID) and OOD uncertainty (MSP)

## **Final Test (After Fine-tuning)**

After fine-tuning the hyperparameters, I made several adjustments to improve the model's performance. The TCN kernel size was increased from 4 to 5, the preds\_step was reduced from 10 to 5, the number of attention heads was increased from 16 to 20, and the number of epochs was raised from 2 to 5. Additionally, I increased the batch size from 4 to 16 to optimize the training process. The fine-tuning process required significant computational resources, with training taking approximately 2 hours per epoch and testing requiring around 1 hour for each epoch, resulting in a total of roughly 15 hours to complete the entire process. These changes aimed to enhance the model's ability to capture more complex patterns and improve prediction accuracy.

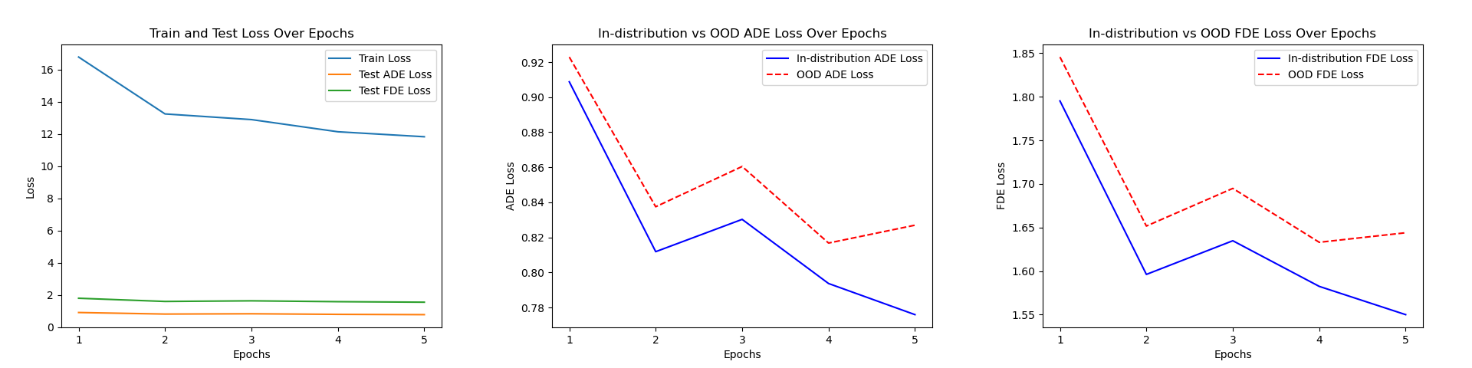
**Table 3:** Summary table (Final prediction adjustment)

A screenshot of a computer

Description automatically generated

In Figure 10, the Average Displacement Error (ADE) and Final Displacement Error (FDE) loss curves over five epochs show notable improvements for both in-distribution and out-of-distribution (OOD) data. The in-distribution ADE loss starts at 0.9087 in Epoch 1 and decreases steadily to 0.7758 by Epoch 5. Similarly, the in-distribution FDE loss begins at 1.7952 and drops to 1.5499 over the same period. This consistent reduction demonstrates the model's increasing accuracy as it learns from the training data.

For the OOD data, the ADE loss initially starts higher at 0.9229 in Epoch 1 and, despite some fluctuations, decreases to 0.8269 by Epoch 5. The OOD FDE loss follows a similar trend, starting at 1.8455 in Epoch 1 and dropping to 1.6439 by the fifth epoch. While these values are higher than the in-distribution losses, they show improvement over time, though with more variability. This indicates the model is making progress in handling OOD scenarios but still faces challenges compared to its performance on in-distribution data.



***Figure 10:*** *In-distribution vs OOD ADE and FDE Loss Over Epochs with Train and Test Loss Comparison (final output)*

Figure 11 presents the comparison of Maximum Softmax Probability (MSP) between in-distribution (ID) and out-of-distribution (OOD) data over five epochs. The solid blue line represents the MSP for ID data, while the red dashed line illustrates the MSP for OOD data.

Throughout the epochs, the MSP for ID data remains relatively stable, fluctuating between 0.78 and 0.79. This consistency reflects the model's high level of confidence when making predictions with familiar, in-distribution data. The narrow range of variation suggests that the model reliably maintains strong confidence in these predictions across all five epochs.

In contrast, the MSP for OOD data shows more fluctuation and starts at a lower value, ranging from 0.74 to 0.76. Initially, the OOD MSP begins at 0.74 in Epoch 1, rises to a peak of 0.758 by Epoch 3, and then slightly decreases toward Epoch 5. This variation shows the model’s reduced confidence when working with unfamiliar OOD data, which is expected since the model encounters more uncertainty in these scenarios. The persistent gap between ID and OOD MSPs across all epochs indicates that the model effectively distinguishes between in-distribution and out-of-distribution data, consistently showing lower confidence in OOD predictions.

Overall, the results suggest that while the model maintains higher uncertainty for OOD data, it achieves a relatively stable and strong confidence level for in-distribution data. This clear distinction between ID and OOD MSPs is a positive outcome, as it demonstrates the model's ability to signal when it is less certain about its predictions, particularly in unfamiliar contexts.

A graph with blue lines and red dots

Description automatically generated

***Figure 11:*** *MSP comparison between in-distribution (ID) and OOD uncertainty (MSP)*

In Figure 12, it is noticable that the model generally performs well in aligning predicted trajectories (in red) with the real trajectories (in green). However, some variations are observed. In Sample 0, there is a noticeable deviation between the predicted and real trajectories, particularly towards the endpoint, suggesting that the model struggles with accurately predicting certain flight patterns, likely due to its limitations in handling more complex or non-linear trajectories. The deviation might also indicate insufficient training epochs or hyperparameter tuning.

In contrast, Samples 1, 2, and 3 demonstrate a much closer alignment between the predicted and real trajectories. The strong alignment in these cases reflects the model's strength in predicting simpler, more linear flight paths. Despite this, the consistent alignment for these samples could suggest that the model is still over-relying on linear predictions, potentially missing the complexity of real-world GA flight paths. Fine-tuning the model further to capture the nuances of more variable trajectories could improve performance, particularly in cases like Sample 0.

A group of graphs showing different types of graphs

Description automatically generated with medium confidence

***Figure 12:*** *3D Trajectory Comparison between real and predicted flight paths for four samples*

Figure 13 contrasts the 3D clustering of predicted and actual flight trajectories. On the left, the predicted trajectories exhibit a more condensed distribution along the Z-axis, with most paths converging towards the center. This suggests the model may struggle to accurately represent the wider range of altitude variations seen in real flight data, possibly underestimating the complexity of actual aviation flight patterns, particularly with respect to altitude changes in general aviation (GA).

On the right, the clustering of real trajectories presents a broader and more evenly distributed spread, both horizontally and vertically, reflecting the inherent diversity in real-world flight paths. This highlights that GA flights naturally involve more intricate and variable spatial movements. The contrast between the predicted and actual clustering points to the model’s limitations, suggesting that adjustments—such as incorporating more varied datasets or refining the training process—are necessary to enhance the model's ability to generalize across the full spectrum of real-world flight trajectories.

A graph of a graph on a grid

Description automatically generated with medium confidence

***Figure 13****: 3D Trajectory Clustering comparison between predicted and real data*

# **Synthetic Data Implementation (Phase 2)**

Building upon the findings from Phase 1, the second phase introduced synthetic data to further enhance the model's generalization capabilities. The synthetic data, generated to mimic real-world GA flight scenarios, was integrated to expose the model to a wider range of trajectories and environmental factors. This phase aimed to address gaps identified in the first phase, particularly the model's handling of Out-of-Distribution (OOD) scenarios and non-standard flight patterns. The integration of synthetic data was crucial in extending the model’s versatility and accuracy when predicting previously unseen or rare flight conditions.

## **Integration**

Following the results from Phase 1, the second phase implemented synthetic data to further enhance the model's ability to generalize. The synthetic data generated by my teammate was 2D-based, lacking the z-axis (altitude) information. To address this, I added a constant value for the z-axis, derived from the average altitude of the real flight data collected over 111 days. This adjustment was made for demonstration purposes. Additionally, I transformed the synthetic data from CSV format into individual text files to match the format I had previously used for training and testing in Python.

The generated synthetic data contained approximately 1 million data points per aircraft ID, spread across multiple trajectories. Due to this large volume, I downsampled the data to align more closely with the index and size of the original real data. The real data consisted of 343 trajectories, and I selected 321 synthetic trajectories, maintaining a near 50/50 split between real and synthetic data for the training process. This ensured a balanced training set, comparable to the 7days1 dataset used previously.

## **Results Showcase**

After integrating the synthetic data into the training phase, the summary table 3 presented details of both the training and test loss over five epochs. The training loss shows a steady decrease from 18.54 at epoch 1 to 13.29 at epoch 5, indicating that the model is learning effectively during the training phase. The Test ADE (Average Displacement Error) and Test FDE (Final Displacement Error) also follow a decreasing trend, reflecting improvements in prediction accuracy across epochs. For example, the Test ADE decreases from 0.98 in epoch 1 to 0.81 by epoch 5, while the Test FDE decreases from 2.02 to 1.60 over the same period. The model’s uncertainty, measured using Maximum Softmax Probability (MSP), remains relatively stable, indicating that the model maintains consistent confidence in its predictions. For Out-of-Distribution (OOD) data, the OOD ADE and OOD FDE losses also improve, showing a decline in prediction error from 0.96 and 1.97 in epoch 1 to 0.84 and 1.65 by epoch 5, respectively. The MSP for OOD data is lower than for in-distribution data but remains stable, suggesting that the model recognizes the unfamiliar data but maintains reasonable confidence. Overall, we could say that the integration of synthetic data helps improve the model's generalization and performance on both in-distribution and OOD data.

**Table 3:** Summary table (Final prediction with synthetic data)

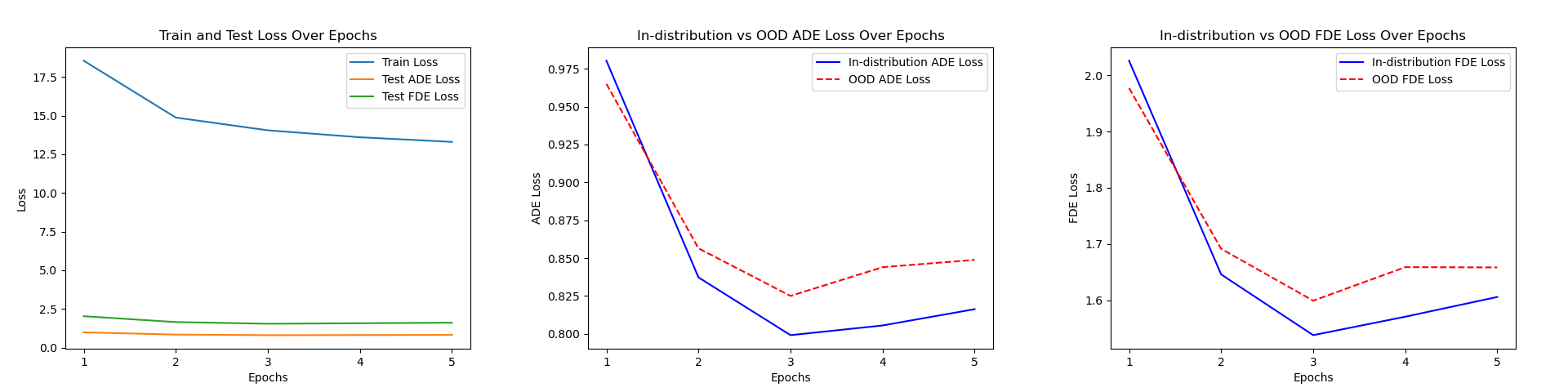
A screenshot of a computer program

Description automatically generated

Figure 14 illustrates the training and test losses across five epochs for both in-distribution and out-of-distribution (OOD) data after the integration of synthetic data. The first graph on the left shows the overall trend of training and test losses. The training loss starts at around 17.5 in epoch 1 and steadily decreases to 12.5 by epoch 5, demonstrating effective learning. The Test ADE (Average Displacement Error) and Test FDE (Final Displacement Error) remain stable, showing no significant sign of overfitting as the gap between the test and training losses is minimal.

The second graph compares the ADE loss for in-distribution and OOD data. Both in-distribution and OOD ADE losses decrease significantly after the first epoch. However, from epoch 3 onward, OOD ADE loss starts to plateau and slightly increases by epoch 5, whereas the in-distribution ADE loss continues to decline, indicating better performance in familiar (in-distribution) scenarios.

The third graph shows the FDE loss for in-distribution and OOD data. Similar to the ADE results, in-distribution FDE decreases steadily, whereas OOD FDE loss plateaus after epoch 3, indicating the model's limitations in handling unfamiliar scenarios. This suggests that while the model performs well on familiar data, the synthetic data integration, though helpful, has room for improvement in handling more complex or unpredictable OOD cases.

**

***Figure 14:*** *Comparison of Loss Curves for In-Distribution and Out-of-Distribution (OOD) Data with Synthetic Data Integration.*

Figure 15 compares the uncertainty levels of the model using Maximum Softmax Probability (MSP) between in-distribution and out-of-distribution (OOD) data over five epochs after integrating synthetic data. The MSP is a measure of the model’s confidence, with higher values indicating greater confidence in its predictions.

The blue line, representing in-distribution MSP, shows consistent performance across epochs. Initially, the model demonstrates relatively high confidence with an MSP of 0.772, and this gradually increases, peaking at 0.784 in epoch 3, before slightly decreasing to 0.781 by epoch 5. This upward trend in in-distribution MSP, compared to results without synthetic data, shows that the model has gained more confidence in handling familiar data after the synthetic data was incorporated.

In contrast, the red dashed line representing OOD MSP reveals a lower confidence in predictions for unfamiliar data. The OOD MSP starts at around 0.745 in epoch 1, rising to approximately 0.756 in epoch 2 before falling back to 0.751 by epoch 5. Although the OOD MSP is consistently lower than the in-distribution MSP, the integration of synthetic data appears to reduce the gap between the two, reflecting the model's improved ability to generalize across unfamiliar data. However, the OOD MSP fluctuates more across the epochs, indicating that while synthetic data integration enhances performance, the model still encounters challenges when handling OOD scenarios. The uncertainty remains higher for OOD data, as expected, but the overall reduction in the gap suggests progress in making the model more robust in diverse environments.

The trend in this figure indicates that, while the model can now express greater confidence in both familiar and unfamiliar data compared to the previous iteration, its ability to handle OOD data still needs further improvement. Regularization techniques or additional fine-tuning could further minimize the OOD uncertainty in future iterations.

A graph with a line and a line

Description automatically generated

***Figure 15:*** *Comparison of Uncertainty Between In-Distribution and Out-of-Distribution (OOD) Data with Synthetic Data*

Figure 16 provides a 3D comparison between actual and predicted flight paths for four different samples. In each subplot, the green line represents the real trajectory, while the red dashed line shows the model’s predicted path after incorporating synthetic data. While the predicted paths generally follow the same direction as the real ones, noticeable deviations occur, particularly toward the end of the trajectories. In Samples 0 and 3, the predicted paths diverge from the real ones, especially along the Z-axis, suggesting the model struggles with capturing precise altitude changes.

Samples 1 and 2 show better alignment between the predicted and real paths, particularly in the X and Y axes, but still have some discrepancies in the Z-axis. These inconsistencies indicate that while the model has improved in predicting general flight patterns, it still needs fine-tuning to better capture vertical movements and accurately forecast the complete trajectory.

In summary, the inclusion of synthetic data has strengthened the model’s ability to generalize and predict GA flight paths, but further optimization is needed to improve performance, particularly in handling complex 3D spatial variations.

A graph of different types of graphs

Description automatically generated with medium confidence

***Figure 16:*** *Comparison of Predicted Trajectories (Red Dashed Line) with Real Trajectories (Green Solid Line) for Four Sample.*

Figure 17 presents a 3D comparison of clustering patterns between predicted and actual general aviation (GA) flight trajectories. While both clusters represent GA flight paths, there are clear differences in how these trajectories are distributed. The predicted trajectories show more vertical dispersion along the Z-axis, with some paths extending farther from the cluster's center. This suggests that the model is overestimating altitude variation in certain cases, leading to a broader vertical spread. In contrast, the real trajectories form a tighter, more compact cluster, particularly along the X-Y plane, indicating that actual flights tend to follow more consistent and constrained patterns both horizontally and vertically.

The broader spread in the predicted clustering points to limitations in the model’s ability to fully capture the complex spatial dynamics of GA flight paths, especially when it comes to realistic altitude behavior. However, the use of synthetic data has enhanced the model's ability to predict a wider variety of flight paths, as shown by the more extensive spatial coverage in the predicted cluster. Further refinement of the model, particularly in how it handles altitude, could improve its trajectory predictions and help reduce the over-dispersion seen in the predictions.

A graph of a graph of a graph

Description automatically generated with medium confidence

***Figure 17:*** *Comparison of 3D Trajectory Clustering for Predicted (Left) and Real (Right) Trajectories, Displaying the Spatial Distribution of Multiple Flight Paths.*

# **Conclusion**

This project successfully undertook the challenges of predicting general aviation (GA) flight paths, particularly when dealing with Out-of-Distribution (OOD) and non-Independent and Identically Distributed (non-IID) data. By refining the pre-trained "Trajairnet" model and incorporating synthetic data, we significantly enhanced the model's ability to predict both familiar and unfamiliar flight trajectories. Training with synthetic data helped the model generalize better, enabling it to manage a wider range of GA flight scenarios and boosting its robustness against OOD data.

The results showed that the model performed well on standard trajectory predictions and also gained the ability to handle more challenging OOD cases with greater accuracy, supported by uncertainty estimation techniques like Maximum Softmax Probability (MSP). This progress demonstrates the model's potential in real-world settings where GA flights often operate in unpredictable conditions.

Despite these advancements, challenges remain. The training process was computationally intensive, and the model’s ability to capture complex flight behaviors, particularly in terms of vertical movements, could benefit from further refinement. Future efforts could focus on optimizing computational efficiency and exploring advanced machine learning methods, such as ensemble learning, to enhance performance in more complex scenarios.

In summary, this project made significant progress in GA trajectory prediction, providing valuable insights into managing OOD data. The outcomes offer a promising foundation for further development, with potential benefits for improving air traffic control and flight safety through more reliable predictive modeling.

# **Drawbacks and Future Work**

This project showed visible improvement in trajectory prediction for general aviation (GA) using synthetic data and advanced uncertainty estimation techniques. However, certain limitations were encountered, which present opportunities for further enhancement.

## **drawbacks**

One notable drawback was the limitation of the synthetic data. The data generated for this project lacked the z-axis (altitude), which is crucial for accurate 3D trajectory prediction. While we compensated by adding a constant z-value based on the average altitude of the real flight data, this solution introduced potential inaccuracies. A more realistic approach to generating synthetic data with varying altitudes is necessary to fully capture the spatial complexity of GA flight paths.

Another major challenge was related to computational resource constraints. The complexity of the hyperparameter configurations, including an increase in attention heads to 20, TCN channels to 512, and GAT hidden layers to 512, placed significant demands on the computing hardware. Running the model with these settings, especially with 16 batches per dataset, required over five hours per epoch. This high computational load was unsustainable on the available hardware, highlighting the need for more advanced computational resources to handle the intensive fine-tuning and larger-scale experiments efficiently.

The selection of out-of-distribution (OOD) data also presented limitations. In this project, the primary difference between in-distribution and OOD datasets was based on the aircraft ID. While this provided useful insights, stronger candidates for OOD data could include flight data from UAVs or even birds, which exhibit distinctly different flight patterns. However, integrating such datasets poses a challenge, as they must be formatted to include environmental factors, such as wind speed, which were central to this project.

## **Future Works**

Looking ahead, there are several promising directions for future work. One key area for improvement involves the generation of more accurate synthetic data. Ensuring that future synthetic datasets include realistic altitude variations will allow for more precise training and testing, leading to better 3D trajectory predictions. The model's generalization capabilities can be further enhanced by developing synthetic data models that account for a wider range of flight conditions.

To address the limitations of computational efficiency, future efforts could benefit from leveraging high-performance computing platforms, such as cloud-based services that offer access to advanced GPUs and TPUs. This would allow for faster model training, enabling more extensive experimentation with complex hyperparameter configurations. Additionally, techniques like model compression, pruning, and quantization could be explored to reduce computational requirements without compromising model performance.

Expanding the scope of OOD data is another area for potential growth. Incorporating a broader range of OOD flight data, such as UAVs or birds, would offer a more comprehensive evaluation of the model's ability to generalize across diverse flight patterns. Although formatting and integrating such data may be challenging, this would provide a more rigorous test of the model’s robustness in unfamiliar conditions.

Furthermore, the application of advanced machine learning techniques, such as ensemble learning or reinforcement learning, could further enhance the model's predictive accuracy and adaptability. Ensemble methods could combine multiple models to mitigate weaknesses and improve overall performance, while reinforcement learning could enable the model to adjust dynamically to real-time changes in flight conditions.

In conclusion, while the project successfully addressed some key challenges in GA trajectory prediction, there is significant potential for further development. By addressing the identified drawbacks and pursuing these future directions, the model can continue to evolve and contribute to safer and more efficient GA flight planning and air traffic management.

# **References**

Baisa, N. L. (2020, May 2). Derivation of a constant velocity motion model for visual tracking. arXiv.org. https://arxiv.org/abs/2005.00844

Burgueño, L. (2021, May 21). A Generic LSTM-Based Neural Network Architecture to infer Heterogeneous Model Transformations. Modeling Languages. https://modeling-languages.com/lstm-neural-network-model-transformations/

Gultepe, Ismail & Sharman, Robert & Williams, Paul & Zhou, Binbin & Ellrod, Gary & Minnis, Patrick & Trier, S. & Griffin, S. & Yum, Seong & Gharabaghi, Bahram & Feltz, W. & Temimi, Marouane & Dimri, A P & Dietz, Sebastian & França, Gutemberg & Almeida, M. & Albuquerque, Francisco & Pu, Zhaoxia & Kneringer, Philipp & Thobois, Ludovic. (2019). A Review of High Impact Weather for Aviation Meteorology. Pure and Applied Geophysics. 176. 1869–1921. 10.1007/s00024-019-02168-6.

Im, Gyubeom. (2024). Notes on Kalman Filter (KF, EKF, ESKF, IEKF, IESKF). 10.48550/arXiv.2406.06427.

Jasra, Ajay & Maama, Mohamed & Ombao, Hernando. (2024). Antithetic multilevel particle filters. Advances in Applied Probability. 1-40. 10.1017/apr.2024.12.

Kim, Sang-Il & Jin, Donghyun & Kim, Jiyeon & Ahn, Do-Seob & Han, Kyung-Soo. (2024). Seamless Weather Data Integration in Trajectory-Based Operations Utilizing Geospatial Information. Remote Sensing. 16. 3573. 10.3390/rs16193573.

Kumichev, G. (2024, July 30). Why ADE and FDE are not the best metrics to score motion prediction. *Medium*. https://towardsdatascience.com/why-ade-and-fde-might-not-be-the-best-metrics-to-score-motion-prediction-model-performance-and-what-1980366d37be

Le, Thanh-Ha & Tran, Phu & Pham, Duc-Thinh & Schultz, Michael & Alam, Sameer. (2020). Short-Term Trajectory Prediction Using Generative Machine Learning Methods

Lu, Shuo & Wang, YingSheng & Sheng, LuJun & Zheng, AiHua & He, LinXiao & Liang, Jian. (2024). Recent Advances in OOD Detection: Problems and Approaches. 10.48550/arXiv.2409.11884.

Patrikar, J., Moon, B., Ghosh, S., Oh, J., & Scherer, S. (2021). TrajAir: A General Aviation Trajectory Dataset (Version 1). *Carnegie Mellon University*. <https://doi.org/10.1184/R1/14866251.v1>

Patrikar, J., Moon, B., Oh, J., & Scherer, S. (2022). Predicting like a pilot: Dataset and method to predict socially-aware aircraft trajectories in non-towered terminal airspace. *2022 International Conference on Robotics and Automation (ICRA)*. https://doi.org/10.1109/icra46639.2022.9811972

Riyana, Surapon. (2024). A Privacy Preservation Model for Dynamic Trajectory Datasets. 10.21203/rs.3.rs-4678898/v1.

Rubinstein, Alexander & Scimeca, Luca & Teney, Damien & Oh, Seong Joon. (2024). Scalable Ensemble Diversification for OOD Generalization and Detection. 10.48550/arXiv.2409.16797.

Schimpf, N., Wang, Z., Li, S., Knoblock, E. J., Li, H., & Apaza, R. D. (2023). A generalized approach to aircraft trajectory prediction via supervised deep learning. *IEEE Access*, *11*, 116183–116195. https://doi.org/10.1109/access.2023.3325053

Wang, B., Zhai, Z., Xiong, R., & Gao, B. (2021). Flight trajectory prediction of General Aviation aircraft based on LSTM model. *2021 IEEE 4th International Conference on Information Systems and Computer Aided Education (ICISCAE)*. https://doi.org/10.1109/iciscae52414.2021.9590656

Xi, Ji & Shi, Meiyu & Zhang, Weiqi & Xu, Zhe & Liu, Yanting. (2024). Trajectory Privacy-Protection Mechanism Based on Multidimensional Spatial–Temporal Prediction. Symmetry. 16. 1248. 10.3390/sym16091248.

Yu, B., Yin, H., & Zhu, Z. (2018). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence* (pp. 3634–3640). AAAI Press.

Zhang, Zhuo & Li, Yang & Gong, Yicheng & Yang, Yue & Ma, Shukun & Guo, Xiaolan & Ercisli, Sezai. (2023). Dataset and Baselines for IID and OOD Image Classification Considering Data Quality and Evolving Environments. International Journal of Interactive Multimedia and Artificial Intelligence. In Press. 1. 10.9781/ijimai.2023.01.007.

Zhao, Zhilin & Cao, Longbing. (2024). Weighting non-IID batches for out-of-distribution detection. Machine Learning. 10.1007/s10994-024-06605-z.

# **Appendix**

**Processed Data Exploration:**

A graph of a line drawn on a white background

Description automatically generatedA screenshot of a graph

Description automatically generated

A diagram of a plane

Description automatically generated with medium confidence