Graph-Based Learning for Modeling Delay Propagation in Airline Networks

October 30, 2024

Keywords: Delay Propagation, Graph-Based Learning, Air Transportation Networks, Network Resilience, Delay Absorption Capacity, Neural Operators

1 INTRODUCTION

Airline operations are pivotal to global transportation and economies, with efficiency directly impacting passenger satisfaction, economic performance, and environmental sustainability. A key challenge in airline operations is *delay propagation*, particularly within *tail networks* where aircraft assignments and crew schedules are intricately linked. Delay propagation refers to the phenomenon where initial flight delays cause subsequent flights to be delayed due to rotational dependencies, crew assignments, and passenger connections. This cascading effect not only undermines operational efficiency and customer satisfaction but also leads to increased fuel consumption and greenhouse gas (GHG) emissions, exacerbating environmental impacts. Studies clearly indicate that a significant percentage of flight delays are *reactionary*, caused by preceding delays rather than initial exogenous factors. Traditional statistical models, such as linear regression and time-series analysis (Erdem & Bilgiç, 2024, Pineda-Jaramillo *et al.*, 2024), have been employed to predict flight delays. While effective in identifying linear relationships and temporal trends, these models often fail to capture the nonlinear interactions and network effects inherent in delay propagation within tail networks.

Advanced machine learning techniques, including graph neural networks (GNNs) and neural operators (Kovachki *et al.*, 2023), have shown promise in modeling complex systems, including for airport situational awareness (Shao *et al.*, 2022). Neural operators can learn mappings between function spaces and have been applied in physics-informed modeling and fluid dynamics (Li *et al.*, 2024). GNNs effectively capture dependencies and learn representations in applications such as vehicle routing, supply chain management, and energy network optimization. However, their application to airline delay propagation remains under-explored. Addressing this gap is critical not only for enhancing operational efficiency but also for reducing aviation's environmental footprint and decarbonizing the industry (EUROCONTROL, 2023). Delay propagation contributes to increased emissions due to extended flight times and inefficient resource utilization. Therefore, innovative solutions that mitigate delays can play a significant role in decarbonizing air travel and supporting climate adaptation and mitigation efforts.

In this paper, we propose a novel graph-based learning framework that integrates neural operators (Kovachki *et al.*, 2023) with GNNs to accurately model local and global interactions within airline tail networks. Our approach captures dynamic, nonlinear relationships between flights, enabling more precise prediction of delay propagation. Additionally, we introduce a method for calculating *Delay Absorption Capacity* (DAC) using our advanced models, providing valuable insights into network resilience. This offers an opportunity to offer actionable recommendations for improving operational planning, resource allocation, and ultimately contributing to emission reductions and sustainability in airline operations. To the best of our knowledge, this is the **first** extension of neural operators to both operations research and specifically to delay propagation modeling and mitigation in the aviation industry. The remainder of this paper is organized as follows. Section 2 details the data collection process and modeling methodology. Section 3 presents the experimental results and evaluates the model's performance. Section 4 discusses the implications of our findings, potential, future research directions, and concludes.

2 METHODOLOGY

We propose a graph-based learning framework utilizing *Graph Neural Operators (GNOs)* to model and predict delay propagation in airline networks. Our methodology encompasses data preparation, construction of a flight network graph, modeling delay propagation using GNOs, model training, and calculation of DAC. Below, we detail each component.

2.1 Mathematical Formulation and Graphical Representation

We consider a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ of N flights, where each flight i is associated with a feature vector $\mathbf{x}_i$ and target variable y_i (e.g., delay prediction). Each flight i is characterized by **(1)** Temporal Attributes: Flight date, scheduled (CRS) and actual departure (DEP) and arrival (ARR) times, day of the week. **(2)** Identifiers: Carrier code, flight number, tail number (aircraft identifier, TAIL_NUM). **(3)** Route Information: Origin and destination airports. **(4)** Delay Metrics: Departure and arrival delays, causes of delay (e.g., carrier, weather, NAS, security, late aircraft). **(5)** Operational Details: Cancellation/diversion status, flight duration, distance.

Sequential Flight Connections To model delay propagation through aircraft rotations, we examine the sequence of flights for each aircraft $a \in \mathcal{A}$, where $\mathcal{A}$ is the set of all aircraft identified by their tail numbers. For each aircraft a, we define an ordered sequence of flights $\{f_k^{(a)}\}_{k=1}^{n_a}$. For consecutive flights $f_k^{(a)}$ and $f_{k+1}^{(a)}$ $(k = 1, \ldots, n_a - 1)$, scheduled ground time - $\mathrm{SGT}_k^{(a)} = \texttt{CRS\_DEP\_TIME}_{k+1}^{(a)} - \texttt{CRS\_ARR\_TIME}_k^{(a)}$; the actual ground time (AGT) as $\mathrm{AGT}_k^{(a)} = \texttt{DEP\_TIME}_{k+1}^{(a)} - \texttt{ARR\_TIME}_k^{(a)}$; the slack time (Slack) as $\mathrm{Slack}_k^{(a)} = \mathrm{SGT}_k^{(a)} - T_{\min}$, where $T_{\min}$ is the minimum required turnaround time; and the propagated delay as $\Delta_{\mathrm{prop}}^{(a)}(k) = \max\left\{0,\ \mathrm{AGT}_k^{(a)} - \mathrm{SGT}_k^{(a)}\right\}$.

Normalization and Encoding Continuous numerical features are normalized using z-score normalization $x_i^{\mathrm{norm}} = \frac{x_i - \mu_x}{\sigma_x}$, where μ_x and σ_x are the mean and standard deviation of feature x. Each categorical variable (feature) with high cardinality (e.g. `TAIL_NUM`) is mapped to an embedding vector $\mathbf{e}_c \in \mathbb{R}^d$, where d is the embedding dimension.

Construction of the Flight Network Graph We represent the airline operations as a directed graph $G = (V, E)$, capturing the structural and operational dependencies crucial for modeling delay propagation. Each node $v_i \in V$ corresponds to flight i and carries a feature vector $\mathbf{x}_i$ containing flight features described earlier. These features encapsulate the state of each flight, including scheduling, delays, and operational details. An edge $e_{ij} \in E$ exists from node v_i to node v_j if flight j immediately follows flight i in the sequence operated by the same aircraft $E = \left\{(i,j) \mid \text{flight } j = f_{k+1}^{(a)},\ \text{flight } i = f_k^{(a)},\ a \in \mathcal{A},\ k = 1, \ldots, n_a - 1\right\}$. Edge features $\mathbf{e}_{ij}$ represent the relationship between i and j, including connection type (aircraft rotation), time between flights (ground time between arrival of i and departure of j), and resource sharing.

Graph Representation Matrices To facilitate the application of GNOs, we define the following matrices: **(1) Adjacency Matrix** $\mathbf{A} \in \mathbb{R}^{N \times N}$ as $A_{ij} = 1$, if there is an edge from flight i to flight j; 0 otherwise. **(2) Node Feature Matrix**: $\mathbf{X} \in \mathbb{R}^{N \times D}$: Each row corresponds to the feature vector $\mathbf{x}_i$ of node i, with D being the feature dimension. **(3) Edge feature matrix**: $\mathbf{E} \in \mathbb{R}^{|E| \times F}$: Each row corresponds to the feature vector $\mathbf{e}_{ij}$ of edge e_{ij}, with F being the edge feature dimension. The constructed graph $G = (V, E)$, along with these matrices, serves as the foundation for applying GNOs. This framework enables the model to capture both the individual flight features and the relational information indicative of delay propagation pathways.

2.2 Modeling Delay Propagation Using Graph Neural Operators

We employ GNOs to model delay propagation by leveraging the constructed graph G. GNOs extend the capabilities of GNNs by learning operators that map between function spaces, allowing them to capture complex spatial and temporal patterns in graphs.

Graph Neural Operator Architecture Our GNO framework combines local message-passing mechanisms with global operator learning. The update rule for node i at layer l is given by $\mathbf{h}_i^{(l)} = \sigma\left(\mathbf{W}^{(l)}\mathbf{h}_i^{(l-1)} + \int_\Omega \kappa^{(l)}(v_i, v)\mathbf{h}^{(l-1)}(v)\, dv\right)$, where: $\mathbf{h}_i^{(l)} \in \mathbb{R}^{d_l}$ is the hidden representation of node i at layer l,σ is an activation function (e.g., ReLU),$\mathbf{W}^{(l)} \in \mathbb{R}^{d_l \times d_{l-1}}$ is a learnable weight matrix,$\kappa^{(l)}(v_i, v)$ is a kernel function learned by the operator, capturing interactions over Ω. In practice, we approximate the integral by aggregating over neighboring nodes in Eq 1, where $\mathcal{N}(i)$ denotes the set of neighbors of node i, and $\kappa^{(l)}(v_i, v_j)$ can be parameterized as in Eq 2, with $\Psi^{(l)}$ being a learnable function (e.g., a neural network) that depends on node and edge features.

$$\mathbf{h}_i^{(l)} = \sigma\left(\mathbf{W}^{(l)}\mathbf{h}_i^{(l-1)} + \sum_{j \in \mathcal{N}(i)} \kappa^{(l)}(v_i, v_j)\mathbf{h}_j^{(l-1)}\right), \tag{1}$$

$$\kappa^{(l)}(v_i, v_j) = \Psi^{(l)}\left(\mathbf{x}_i, \mathbf{x}_j, \mathbf{e}_{ij}\right), \tag{2}$$

Incorporating Global Interactions To capture global patterns, we integrate spectral methods using the *Fourier Neural Operator (FNO)* within the GNO framework. The FNO component operates in the frequency domain, enabling the modeling of long-range dependencies. The FNO component applies Fourier transforms: $\hat{\mathbf{H}}^{(l-1)} = \mathcal{F}\left(\mathbf{H}^{(l-1)}\right)$, where $\mathbf{H}^{(l-1)} \in \mathbb{R}^{N \times d_{l-1}}$ and $\mathcal{F}$ denotes the discrete Fourier transform. The model (spectral convolution) then learns a multiplication operator in the Fourier space $\hat{\mathbf{H}}^{(l)} = \hat{\mathbf{P}}^{(l)} \odot \hat{\mathbf{H}}^{(l-1)}$, where $\hat{\mathbf{P}}^{(l)}$ is a learned filter in the frequency domain, and $\odot$ denotes element-wise multiplication. After applying the inverse Fourier transform, we obtain $\mathbf{H}_{\text{FNO}}^{(l)} = \mathcal{F}^{-1}\left(\hat{\mathbf{H}}^{(l)}\right)$.

Combined Update Rule The final update for node representations combines the GNO and FNO components, where $\mathbf{h}_{i,\text{FNO}}^{(l)}$ is the i-th row of $\mathbf{H}_{\text{FNO}}^{(l)}$. This architecture enables the model to learn from both local graph structures and global patterns.

$$\mathbf{h}_i^{(l)} = \sigma\left(\mathbf{W}^{(l)}\mathbf{h}_i^{(l-1)} + \sum_{j \in \mathcal{N}(i)} \kappa^{(l)}(v_i, v_j)\mathbf{h}_j^{(l-1)} + \mathbf{h}_{i,\text{FNO}}^{(l)}\right), \tag{3}$$

Training the Model The GNO model parameters—including $\mathbf{W}^{(l)}$, $\kappa^{(l)}$, and $\hat{\mathbf{P}}^{(l)}$—are learned by minimizing a loss function over the training data. We define the loss function : $\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{\Delta}_i - \Delta_i^{\text{prop}}\right)^2 + \lambda\|\Theta\|^2$, where $\hat{\Delta}_i$ is the predicted propagated delay for flight i, Δ_i^{prop} is the actual propagated delay, Θ represents all trainable parameters, λ is a regularization coefficient. We use stochastic gradient descent-based optimizers (Adam) to minimize $\mathcal{L}$, updating parameters across multiple epochs with early stopping based on validation loss.

Calculation of delay absorption capacity (DAC) For each flight i, the DAC is computed as $\text{DAC}_i = \text{Slack}_i - \hat{\Delta}_i$, where Slack_i is the slack time available and $\hat{\Delta}_i$ is the predicted propagated delay. Positive DAC indicates resilience, while negative indicates vulnerability. The overall network resilience is assessed via $\text{DAC}_{\text{network}} = \sum_{i=1}^{N} \text{DAC}_i$. Flights with low or negative DAC_i are critical nodes where delays are likely to propagate, informing operational decisions.

3 COMPUTATIONAL RESULTS

Dataset Description We conducted experiments using the US Flight Delay Dataset for 2016-2020, comprising $2M$ flight records. The dataset includes a comprehensive set of features essential for modeling delay propagation, including temporal attributes, operational metrics, carrier information, airport information, and various delay-related features, with specific delay causes.

Experimental setup Experiments were run on a standard CPU configuration with monitored memory usage, using Python 3.10, Scikit-learn, PyTorch, PyTorch Geometric, NetworkX, Pandas, NumPy. We evaluated several traditional models—Linear Regression, Ridge Regression, Random Forests, and Neural Networks—and compare their performance to the proposed

Graph Neural Operators (GNOs). The model performance was evaluated using the following metrics Mean Absolute Error: $\text{MAE} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \left| \hat{\Delta}_i - \Delta_i^{\text{prop}} \right|$; Root Mean Square Error: $\text{RMSE} = \sqrt{\frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \left(\hat{\Delta}_i - \Delta_i^{\text{prop}} \right)^2}$; R-squared Score: $R^2 = 1 - \frac{\sum_{i=1}^{N_{\text{test}}} \left(\hat{\Delta}_i - \Delta_i^{\text{prop}} \right)^2}{\sum_{i=1}^{N_{\text{test}}} \left(\Delta_i^{\text{prop}} - \bar{\Delta}^{\text{prop}} \right)^2}$.

Table 1 summarizes the performance of each model on the test set. The GNOs model outperformed traditional machine learning models, achieving the lowest MAE and RMSE, and the highest R^2 score. The model's ability to capture both local interactions through the graph structure and global patterns via the integrated FNO contributed significantly to its superior performance. Ensemble methods like Random forests showed strong performance, indicating the importance of capturing nonlinear relationships. Linear models had higher errors, suggesting that delay propagation exhibits complex patterns not well captured by simple linear relationships. By analyzing the model's predictions and the calculated DAC, we provide actionable insights, identifying vulnerable flights (with low DAC for targeted interventions), optimizing scheduling and resource deployment to enhance delay absorption, and reducing delay propagation contributes to lower fuel consumption and GHG emissions, supporting sustainability goals.

Table 1 – *Model Performance Comparison on Test Set*

Model	MAE (↓ better)	RMSE (↓ better)	R^2 (↑ better)
Linear Regression	5.685	8.637	0.623
Ridge Regression	4.799	9.986	0.496
Random Forests	4.097	5.511	0.846
Neural Networks	4.093	5.514	0.846
GNOs	**2.339**	**2.900**	**0.957**

4 CONCLUSION

We propose a graph-based learning framework using graph neural operators to model delay propagation in airline networks, integrating local and global interactions for superior predictive accuracy. By identifying critical flights with low delay absorption capacity, our model offers actionable insights for proactive interventions, enhancing operational efficiency, and reducing environmental impacts through minimized delays and fuel consumption. Future work will incorporate stochastic modeling, real-time data integration, and external factors like weather and air traffic control constraints to improve model robustness and practical applicability.

References

Erdem, Furkan, & Bilgiç, Taner. 2024. Airline delay propagation: Estimation and modeling in daily operations. *Journal of Air Transport Management*, **115**, 102548.

EUROCONTROL. 2023 (August). *Think Paper #21: Long-haul flight decarbonisation: When can cutting-edge energies & technologies make a difference?* Tech. rept. 21. EUROCONTROL.

Kovachki, Nikola, Li, Zongyi, Liu, Burigede, Azizzadenesheli, Kamyar, Bhattacharya, Kaushik, Stuart, Andrew, & Anandkumar, Anima. 2023. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, **24**(89), 1–97.

Li, Zongyi, Kovachki, Nikola, Choy, Chris, Li, Boyi, Kossaifi, Jean, Otta, Shourya, Nabian, Mohammad Amin, Stadler, Maximilian, Hundt, Christian, Azizzadenesheli, Kamyar, *et al.* 2024. Geometry-informed neural operator for large-scale 3d pdes. *NeurIPS*, **36**.

Pineda-Jaramillo, Juan, Munoz, Claudia, Mesa-Arango, Rodrigo, Gonzalez-Calderon, Carlos, & Lange, Anne. 2024. Integrating multiple data sources for improved flight delay prediction using explainable machine learning. *Research in Transportation Business & Management*, **56**, 101161.

Shao, Wei, Prabowo, Arian, Zhao, Sichen, Koniusz, Piotr, & Salim, Flora D. 2022. Predicting flight delay with spatio-temporal trajectory convolutional network and airport situational awareness map. *Neurocomputing*, **472**, 280–293.