

Article

State-DynAttn: A Hybrid State-Space and Dynamic Graph Attention Architecture for Robust Air Traffic Flow Prediction Under Weather Disruptions

Fei Yan ^{1,*} and Huawei Wang ²¹ College of Air Traffic Management, Civil Aviation Flight University of China, Chengdu 618307, China² College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; huaawei@nuaa.edu.cn

* Correspondence: yanfei@cafuc.edu.cn

Abstract

We propose State-DynAttn, a hybrid architecture for robust air traffic flow prediction under weather disruptions, which integrates state-space models (SSMs) with dynamic graph attention to address the challenges of long-range dependency modeling and adaptive spatial-temporal relationship learning. The increasing complexity of air traffic systems, exacerbated by unpredictable weather events, demands methods that can simultaneously capture global temporal patterns and localized disruptions; existing approaches often struggle to balance these requirements efficiently. The proposed method employs two parallel branches: an SSM branch for continuous-time recurrent modeling of long-range dependencies with linear complexity, and a dynamic graph attention branch that adaptively computes node-pair weights while incorporating weather severity features through sparsification strategies for scalability. These branches are fused via a data-dependent gating mechanism, enabling the model to dynamically prioritize either global temporal dynamics or localized spatial interactions based on input conditions. Moreover, the architecture leverages memory-efficient attention computation and HiPPO initialization to ensure stable training and inference. Experiments on real-world air traffic datasets demonstrate that State-DynAttn outperforms existing baselines in prediction accuracy and robustness, particularly under severe weather scenarios. The framework's ability to handle both gradual traffic evolution and abrupt disruption-induced changes makes it suitable for real-world deployment in air traffic management systems. Furthermore, the design principles of State-DynAttn can be extended to other spatiotemporal prediction tasks where long-range dependencies and dynamic relational structures coexist. This work contributes a principled approach to hybridizing state-space models with graph-based attention, offering insights into the trade-offs between computational efficiency and modeling flexibility in complex dynamical systems.

Keywords: air traffic flow prediction; state-space models (SSMs); dynamic graph attention; weather disruptions; hybrid architecture

MSC: 90-05



Academic Editors: Yicheng Zhang and Hsien-Chung Wu

Received: 14 September 2025

Revised: 2 October 2025

Accepted: 15 October 2025

Published: 21 October 2025

Citation: Yan, F.; Wang, H. State-DynAttn: A Hybrid State-Space and Dynamic Graph Attention Architecture for Robust Air Traffic Flow Prediction Under Weather Disruptions. *Mathematics* **2025**, *13*, 3346. <https://doi.org/10.3390/math13203346>

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

1. Introduction

Air traffic flow prediction serves as a cornerstone of aviation safety and operational efficiency in modern air traffic management systems. Accurate predictions enable air traffic

controllers and airline operators to proactively optimize flight schedules, allocate airspace resources, and implement flow control measures that prevent dangerous congestion scenarios. Beyond economic considerations of delay reduction—which cost the aviation industry billions annually—reliable traffic flow prediction is fundamentally a safety imperative. It provides the situational awareness necessary to prevent mid-air conflicts, manage runway capacity within safe limits, and ensure adequate separation between aircraft, particularly during adverse weather conditions when margins of error narrow significantly.

The challenge intensifies dramatically under weather disruptions. Severe weather events such as thunderstorms, dense fog, and winter precipitation can reduce airport capacity by 30–70% within minutes, creating cascading delays and potentially hazardous congestion if not anticipated. Inaccurate predictions during these critical periods can lead to three safety-compromising scenarios: (1) over-scheduling that forces controllers to manage more aircraft than safely manageable in degraded conditions; (2) inadequate advance warning that prevents proper flow control implementation; (3) inefficient rerouting decisions that concentrate traffic in alternative corridors, creating new bottlenecks. Therefore, developing prediction models that maintain accuracy specifically during weather disruptions is not merely an optimization problem but an essential safety requirement for next-generation air traffic management.

Traditional approaches to this problem have relied on statistical methods, such as autoregressive integrated moving average (ARIMA) models, which, while effective for capturing linear temporal patterns, struggle with the nonlinear dynamics and long-range dependencies inherent in air traffic data. By nonlinear dynamics, we refer to the complex, non-proportional relationships in air traffic patterns—for example, a 10% increase in weather severity may cause a 50% reduction in traffic flow at critical congestion points while having minimal impact during off-peak hours. Similarly, traffic flow exhibits threshold behaviors where small perturbations can trigger cascading delays across the network. These nonlinear effects cannot be adequately captured by linear models like ARIMA, which assume proportional relationships between inputs and outputs. The increasing complexity of airspace operations and the growing frequency of weather-related disruptions have exposed the limitations of these conventional methods, necessitating more sophisticated solutions.

The advent of deep learning has brought significant advances to time series forecasting, with recurrent neural networks (RNNs) [1] and their variants like long short-term memory (LSTM) [2] networks demonstrating superior performance in capturing temporal dependencies. However, these architectures often face challenges in modeling very long sequences efficiently, as their computational complexity grows quadratically with sequence length. Moreover, the inherently graph-structured nature of air traffic networks—where airports serve as nodes and flight routes as edges—requires methods that can effectively capture spatial relationships while maintaining temporal coherence.

Recent developments in graph neural networks (GNNs) [3] have shown promise in addressing the spatial aspects of air traffic prediction. Graph attention networks (GAT) [4], in particular, have demonstrated the ability to learn adaptive relationships between nodes. Nevertheless, these approaches typically focus on static graph structures and fail to account for the dynamic nature of air traffic patterns, especially under disruptive conditions. The simultaneous need to model both long-range temporal dependencies and rapidly changing spatial interactions presents a unique challenge that the existing methods struggle to address comprehensively.

State-space models (SSMs) [5] offer an alternative approach to sequence modeling, with theoretical advantages in handling long-range dependencies through their continuous-time formulation. Recent work on structured state space sequence models (S4) [6] has

demonstrated their effectiveness in various sequence modeling tasks, achieving performance comparable to attention-based models while maintaining linear computational complexity. However, these models typically operate on individual time series and lack mechanisms to incorporate the rich relational information present in air traffic networks.

We propose State-DynAttn, a hybrid architecture that combines the strengths of state-space models and dynamic graph attention to address the unique challenges of air traffic flow prediction under weather disruptions. The key innovation lies in the parallel processing of long-range temporal patterns through SSMs and adaptive short-term feature extraction through dynamic attention mechanisms. This design allows the model to maintain awareness of global traffic patterns while remaining responsive to local disruptions caused by weather events. The architecture employs a novel fusion mechanism that dynamically balances the contributions of these two components based on input conditions, enabling robust performance across both normal operations and disruptive scenarios.

The proposed method offers several advantages over existing approaches. First, it achieves superior computational efficiency compared to pure attention-based models, particularly for long sequences, by leveraging the linear complexity of SSMs. Second, it introduces a dynamic graph attention mechanism that adapts to changing weather conditions, allowing the model to focus on the most relevant spatial relationships at any given time. Third, the architecture demonstrates improved robustness to distribution shifts caused by extreme weather events, a critical requirement for real-world deployment in air traffic management systems.

This work makes three primary contributions: (1) we present the first hybrid architecture that effectively combines state-space models with dynamic graph attention for air traffic flow prediction, addressing both long-range temporal dependencies and adaptive spatial relationships; (2) we develop a novel weather-aware attention mechanism that dynamically adjusts graph connectivity based on weather severity, enabling more accurate predictions during disruptive events; (3) we demonstrate through extensive experiments that State-DynAttn outperforms the existing methods in both prediction accuracy and computational efficiency, particularly under challenging weather conditions.

The remainder of this paper is organized as follows: Section 2 reviews related work in air traffic prediction, state-space models, and graph attention networks. Section 3 provides necessary background on SSMs and dynamic graph attention. Section 4 details the State-DynAttn architecture and its components. Sections 5 and 6 present the experimental setup and results, respectively. Finally, Sections 7 and 8 discuss the implications and conclude the paper.

2. Related Work

The prediction of air traffic flow under weather disruptions sits at the intersection of several research domains, each contributing distinct methodologies and insights. The existing approaches can be broadly categorized into three strands: traditional statistical methods, deep learning-based temporal models, and graph-based spatial-temporal approaches. While these methods have advanced the field significantly, they often address only subsets of the challenges inherent in air traffic prediction, particularly when dealing with extreme weather events.

2.1. Traditional Approaches to Air Traffic Prediction

Early work in air traffic forecasting relied heavily on statistical time series models, with ARIMA variants being particularly prevalent [7]. These methods proved adequate for capturing basic temporal patterns but struggled with the nonlinear dynamics and external

factors inherent in air traffic systems. The integration of weather data into these models typically involved simple concatenation of meteorological features, failing to account for the complex, nonlinear interactions between weather patterns and traffic flow. More sophisticated approaches attempted to model these relationships through vector autoregression [8], though computational constraints limited their ability to handle large-scale networks.

2.2. Deep Learning for Temporal Modeling

The limitations of traditional methods spurred interest in deep learning approaches, particularly recurrent architectures. LSTM networks [2] and their gated variants demonstrated superior performance in capturing temporal dependencies, leading to widespread adoption in traffic prediction tasks. However, these models often require careful tuning of window sizes and struggle with very long sequences due to their sequential nature. The introduction of attention mechanisms [9] offered improvements in handling long-range dependencies, but at the cost of quadratic computational complexity. Recent work has explored hybrid architectures combining convolutional and recurrent layers [10], though these approaches still face challenges in modeling abrupt changes caused by weather disruptions. The computational challenge arises from the sequential nature of recurrent architectures. LSTMs process sequences step by step, maintaining hidden states of dimension d_h at each time step. For a sequence of length L , this requires $O(L \cdot d_h^2)$ operations for the recurrent computations alone. When dealing with very long sequences (e.g., $L > 1000$ time steps representing several days of minute-level observations), the computational cost becomes prohibitive. Furthermore, the choice of input window size presents a critical trade-off: larger windows (e.g., 7 days = 10,080 min) provide more historical context but increase both computation time and memory requirements linearly with L . Conversely, smaller windows (e.g., 6 h = 360 min) reduce computational burden but may fail to capture important long-range patterns such as weekly periodicity in air traffic. This necessitates careful manual tuning of window sizes, which is problem-specific and lacks theoretical guidance.

2.3. Graph-Based Spatial–Temporal Approaches

Recognizing the networked nature of air traffic systems, researchers have increasingly turned to graph neural networks. Early graph convolutional networks [3] demonstrated the value of incorporating topological information, while subsequent work on graph attention networks [4] introduced adaptive relationship learning between nodes. These methods proved particularly effective for capturing spatial dependencies in transportation networks. Recent advances have focused on dynamic graph constructions [11], with some approaches incorporating external factors like weather through additional edge features. However, most existing graph-based methods either treat the graph structure as static or update it at fixed intervals, limiting their responsiveness to rapidly changing conditions.

2.4. State-Space Models for Sequence Processing

State-space models have emerged as a powerful alternative for sequence modeling, particularly for long-range dependencies. The structured state-space sequence model (S4) [6] and its variants achieve linear complexity while maintaining strong performance across various tasks. These models excel at capturing gradual temporal patterns but typically operate on individual sequences, lacking mechanisms to incorporate relational information. Recent work has begun exploring combinations of SSMs with graph-based approaches [12], though these efforts have not specifically addressed the challenges of weather-disrupted air traffic prediction.

2.5. Weather-Aware Traffic Prediction

The specific challenge of weather-impacted traffic prediction has inspired several specialized approaches. Some methods treat weather as an additional input feature [13], while others attempt to model its effects through physical simulations [14]. Ensemble methods have shown promise in quantifying weather forecast uncertainty [15], though their computational demands limit real-time applicability. Recent work has also explored resilience metrics for air traffic networks [16], though these typically focus on post-disruption analysis rather than prediction.

The proposed State-DynAttn architecture addresses key limitations across these approaches by combining the long-range modeling capabilities of SSMs with the adaptive relational learning of dynamic graph attention. Unlike previous methods that either treat weather as a static input or model it separately from traffic patterns, our approach integrates weather severity directly into the attention computation, enabling dynamic adjustment of spatial relationships based on disruption intensity. This hybrid design achieves superior performance while maintaining computational efficiency through careful architectural choices and sparsification strategies.

3. Preliminaries on State-Space Models and Dynamic Graph Attention

To establish the theoretical foundation for our proposed architecture, this section systematically examines two fundamental components: state-space models for temporal sequence processing and dynamic graph attention mechanisms for relational learning. These concepts form the building blocks of our hybrid approach, each addressing distinct aspects of the air traffic prediction problem. The mathematical notations/parameters used herein are as shown in the table below:

Symbol	Description	Dimension
N	Number of nodes (airports)	scalar
L	Sequence length	scalar
d	Input feature dimension	scalar
d_h	Hidden dimension	scalar
p	Weather feature dimension	scalar
$\mathbf{x}_t$	Traffic flow features at time t	$N \times d$
$\mathbf{w}_t$	Weather severity indicators	$N \times N \times p$
$h(t)$	Continuous-time hidden state	d_h
$\mathbf{h}_t$	Discrete-time hidden state	$N \times d_h$
$\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$	SSM state-space parameters	various
$\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathbf{D}}$	Discretized SSM parameters	various
α_{ij}	Attention weight between nodes i, j	scalar
k	Top- k neighborhood size	scalar
τ	Weather pruning threshold	scalar
$\mathbf{g}_t$	Gating weight at time t	$N \times 1$
n, k	Matrix row and column indices	scalar

3.1. Continuous-Time State-Space Models

State-space models provide a mathematical framework for describing dynamical systems through latent state evolution. The continuous-time formulation of SSMs offers particular advantages for modeling physical processes like air traffic flow, where observations occur at discrete time steps but the underlying dynamics evolve continuously. A basic continuous-time SSM can be expressed as follows:

$$\dot{h}(t) = Ah(t) + Bx(t) \quad (1)$$

$$y(t) = Ch(t) + Dx(t) \quad (2)$$

where $h(t)$ represents the hidden state, $x(t)$ the input signal, and $y(t)$ the output. The matrices A, B, C , and D parameterize the system dynamics, with A governing the state transition and C mapping states to outputs. This formulation naturally handles irregularly sampled observations, making it suitable for real-world scenarios where data collection intervals may vary.

The discretization of continuous-time SSMs for digital computation typically employs the bilinear transform or zero-order hold methods. For practical digital implementation, the continuous-time SSM (Equations (1) and (2)) must be discretized at fixed time intervals. Using a discretization step size Δt (15 min in our implementation), we obtain the discrete-time state-space representation:

$$\mathbf{h}_k = \bar{\mathbf{A}}\mathbf{h}_{k-1} + \bar{\mathbf{B}}\mathbf{x}_k \quad (3)$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{h}_k + \mathbf{D}\mathbf{x}_k \quad (4)$$

where

- $k \in \{1, 2, \dots, L\}$ indexes discrete time steps (replacing continuous time t).
- $\mathbf{h}_k \in \mathbb{R}^{d_h}$ is the hidden state at step k .
- $\mathbf{x}_k \in \mathbb{R}^d$ is the input at step k (traffic features for a specific airport).
- $\mathbf{y}_k \in \mathbb{R}^d$ is the output at step k (predicted traffic for that airport).
- $\bar{\mathbf{A}}, \bar{\mathbf{B}}, \mathbf{C}$, and $\mathbf{D}$ are discretized matrix parameters obtained from continuous $\mathbf{A}, \mathbf{B}, \mathbf{C}$, and $\mathbf{D}$ via zero-order hold transformation (detailed in Section 4.1).

3.2. HiPPO Theory and State Initialization

The High-order Polynomial Projection Operators (HiPPO) framework [17] provides a principled approach to initializing SSM parameters for effective long-range dependency modeling. The HiPPO theory demonstrates that certain classes of matrices A can optimally project continuous signals onto polynomial bases, allowing the state $h(t)$ to compress the history of the input $x(t)$. This property proves particularly valuable for air traffic prediction, where historical patterns often contain predictive information about future states.

The HiPPO-LegS variant, which uses Legendre polynomial basis functions, yields a state-transition matrix with the following structure:

$$A_{nk} = - \begin{cases} (2n+1)^{1/2}(2k+1)^{1/2} & \text{if } n > k \\ n+1 & \text{if } n = k \\ 0 & \text{if } n < k \end{cases} \quad (5)$$

where $n \in \{0, 1, \dots, N-1\}$ represents the row index and $k \in \{0, 1, \dots, N-1\}$ represents the column index of the $N \times N$ state-transition matrix $\mathbf{A}$. Each matrix element $A_{n,k}$ defines the coupling strength between state dimension k and state dimension n . The conditional structure ensures that (1) when $n > k$, the coefficient follows a specific polynomial relationship that implements optimal projection onto Legendre basis functions; (2) when $n = k$ (diagonal elements), the coefficient equals $n+1$, providing self-feedback for each state dimension; (3) when $n < k$ (upper triangular), the coefficient is zero, making $\mathbf{A}$ a lower-triangular matrix. This triangular structure is crucial for efficient computation and ensures causal state evolution where each dimension only depends on current and previous dimensions, not future ones.

This initialization scheme enables the model to automatically maintain a memory of past inputs weighted by their recency without requiring manual tuning of memory windows or attention mechanisms.

3.3. Graph Attention Networks

Graph neural networks (GNNs) provide a natural framework for modeling air traffic systems, where the inherent network structure—airports as nodes and flight routes as edges—necessitates methods that can effectively propagate information across topological connections. Traditional neural networks process data in Euclidean spaces, but air traffic networks exist in non-Euclidean graph domains, where relationships are defined by connectivity rather than spatial proximity.

GNNs address this challenge through message-passing mechanisms that aggregate information from neighboring nodes, enabling the model to learn representations that respect the underlying network structure. The basic GNN layer updates node representations by the following:

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\mathbf{W}^{(l)} \cdot \text{AGGREGATE}\left(\left\{\mathbf{h}_j^{(l)} : j \in \mathcal{N}(i)\right\}\right)\right)$$

where $\mathcal{N}(i)$ denotes the neighborhood of node i and AGGREGATE is a permutation-invariant function such as sum, mean, or max pooling.

In air traffic prediction, this architecture is particularly valuable because disruptions at one airport (e.g., weather-related delays) propagate through connected routes, affecting downstream airports in ways that depend on network topology rather than geographic distance alone. For instance, severe weather at a major hub like Chicago O'Hare impacts not only nearby airports but all destinations with direct connections, regardless of physical distance. GNNs naturally capture these topological dependencies, making them well-suited for modeling cascading effects in air traffic networks.

Graph attention networks extend traditional graph neural networks by introducing learnable attention weights between connected nodes. The basic GAT layer [4] computes attention coefficients α_{ij} between nodes i and j as follows:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(a^T [W\mathbf{h}_i || W\mathbf{h}_j]))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(a^T [W\mathbf{h}_i || W\mathbf{h}_k]))} \quad (6)$$

where W represents a learnable weight matrix, a is an attention parameter vector, and $\mathcal{N}_i$ denotes the neighborhood of node i . The operator $||$ indicates concatenation. This formulation allows the model to dynamically allocate attention resources based on node features rather than relying on fixed graph structures.

3.4. Dynamic Graph Construction

Traditional graph attention networks typically operate on static graphs, which limits their applicability to air traffic networks where relationships between airports constantly evolve. Dynamic graph approaches address this limitation by allowing the edge structure to change over time. Two primary variants exist: discrete-time dynamic graphs, which update at fixed intervals, and continuous-time dynamic graphs, which evolve smoothly between observations.

The continuous-time dynamic graph formulation represents edge weights as functions of time:

$$e_{ij}(t) = f_{\theta}(h_i(t), h_j(t), t) \quad (7)$$

where f_θ is a learnable function parameterized by θ , and $h_i(t)$ denotes the node features at time t . This approach naturally accommodates irregular observation intervals and gradual relationship changes, both common characteristics of air traffic data.

3.5. Weather-Aware Attention Mechanisms

Incorporating weather impacts into graph attention requires extending the basic attention formulation to consider meteorological conditions. The weather-aware attention coefficient can be expressed as follows:

$$\alpha_{ij}^w = \frac{\exp(\text{LeakyReLU}(a^T [Wh_i || Wh_j || W_w w_{ij}]))}{\sum_{k \in \mathcal{N}_i} \exp(\text{LeakyReLU}(a^T [Wh_i || Wh_k || W_w w_{ik}]))} \quad (8)$$

where w_{ij} represents weather features between nodes i and j , and W_w is a weather-specific projection matrix. This formulation allows the model to modulate attention weights based on both node features and current weather conditions, enabling more accurate predictions during disruptive events.

The combination of these components—continuous-time state-space models for temporal dynamics and dynamic graph attention for spatial relationships—provides the theoretical foundation for our proposed State-DynAttn architecture. The next section details how we integrate these elements into a cohesive framework specifically designed for air traffic flow prediction under weather disruptions.

4. State-DynAttn: Hybrid SSM-Dynamic Attention Architecture for Real-Time Air Traffic Flow Prediction

The State-DynAttn architecture addresses the dual challenges of long-range temporal modeling and adaptive spatial relationship learning through a novel parallel processing framework. As shown in Figure 1, the architecture comprises three main processing stages:

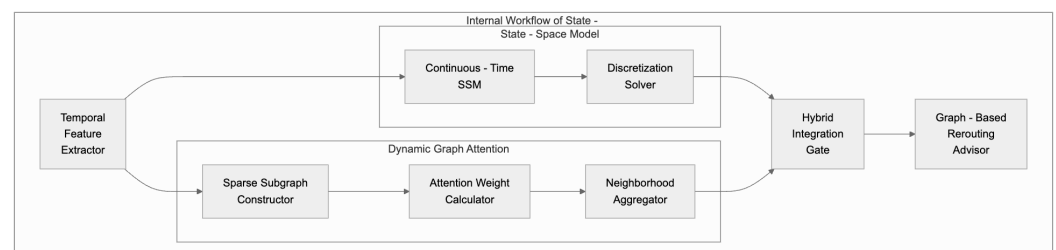


Figure 1. Internal workflow and parallel processing architecture of State-DynAttn model. The architecture processes air traffic data through two complementary branches: the state-space model branch (upper, blue pathway) captures long-range temporal dependencies through continuous-time dynamics and discretization, while the dynamic graph attention branch (lower, orange pathway) adapts to localized weather disruptions through sparse subgraph construction, attention weight calculation, and neighborhood aggregation. The Hybrid Integration Gate (green pathway) dynamically fuses both outputs based on input characteristics, enabling the model to balance global temporal patterns with localized spatial disruptions. Component details are described in Section 4.

Stage 1: Temporal Pattern Extraction—The input layer receives raw traffic flow data and weather observations, which are processed by the Temporal Pattern Extractor to generate embedded representations suitable for both branches.

Stage 2: Parallel Branch Processing—The upper branch employs a state-space model (SSM) with continuous-time dynamics for capturing long-range temporal dependencies. The continuous-time SSM module maintains hidden states that evolve according to learned dynamics, and the Discretization Solver converts these continuous representations into discrete-time predictions. Simultaneously, the lower branch implements dynamic graph

attention through three sub-components: the Sparse Subgraph Constructor identifies relevant airport connections based on current conditions, the Attention Weight Calculator computes importance scores for each node pair incorporating weather severity, and the Neighborhood Aggregator pools information from the most relevant neighbors.

Stage 3: Hybrid Integration—The outputs from both branches are combined through a Hybrid Integration Gate that dynamically weights their contributions based on input characteristics, producing the final Graph-Based Rerouting Advice.

This parallel design enables the model to maintain awareness of global traffic patterns (SSM branch) while remaining responsive to localized disruptions (attention branch). The following subsections detail the technical implementation of each component.

4.1. Hybrid Parallel Architecture for Long- and Short-Term Pattern Modeling

The architecture processes input traffic data through two parallel branches: a state-space model branch for continuous-time sequence modeling and a dynamic graph attention branch for weather-aware spatial relationship learning. The SSM branch employs HiPPO-initialized S4 layers to capture gradual traffic pattern evolution, while the graph attention branch adapts to localized disruptions through weather-modulated edge weights.

The input representation combines traffic flow features $\mathbf{x}_t \in \mathbb{R}^{N \times d}$ with weather severity indicators $\mathbf{w}_t \in \mathbb{R}^{N \times N \times p}$, where N denotes the number of nodes (airports), d the feature dimension, and p the weather feature dimension. The model first applies temporal embedding to project inputs into a latent space:

$$\mathbf{h}_t^0 = \mathbf{W}_e \mathbf{x}_t + \mathbf{b}_e \quad (9)$$

where $\mathbf{W}_e \in \mathbb{R}^{d \times d_h}$ and $\mathbf{b}_e \in \mathbb{R}^{d_h}$ are learnable parameters, with d_h being the hidden dimension. This embedded representation feeds both branches simultaneously.

The SSM branch processes each node's temporal sequence independently through stacked S4 layers. Each layer implements the following discretized state-space equations:

$$\mathbf{h}_t^{l+1} = \bar{\mathbf{A}}\mathbf{h}_t^l + \bar{\mathbf{B}}\mathbf{h}_t^l \quad (10)$$

$$\mathbf{y}_t^l = \bar{\mathbf{C}}\mathbf{h}_t^l \quad (11)$$

where l indexes the layer, and $\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}$ are discretized parameters initialized using the HiPPO theory. The diagonal plus low-rank structure of $\mathbf{A}$ enables efficient computation while maintaining expressive power for long sequences. The discretization step size $\Delta t = 15$ min is chosen to match the temporal granularity of our aggregated traffic data (as described in Section 5.1). The continuous-time parameters $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D})$ are converted to their discrete-time counterparts $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \bar{\mathbf{C}}, \bar{\mathbf{D}})$ using the zero-order hold (ZOH) method:

$$\begin{aligned} \bar{\mathbf{A}} &= \exp(\mathbf{A}\Delta t) \\ \bar{\mathbf{B}} &= \mathbf{A}^{-1}(\bar{\mathbf{A}} - \mathbf{I})\mathbf{B} \\ \bar{\mathbf{C}} &= \mathbf{C} \\ \bar{\mathbf{D}} &= \mathbf{D} \end{aligned}$$

where $\mathbf{I}$ is the identity matrix and $\exp(\cdot)$ denotes the matrix exponential. The ZOH method is preferred over simpler discretization schemes (e.g., Euler method) because it exactly preserves the continuous-time dynamics when the input remains constant between sampling intervals, which is a reasonable assumption for aggregated 15 min traffic counts. This discretization approach enables the model to handle irregular sampling intervals during data collection while maintaining theoretical guarantees on state evolution.

4.2. Weather-Aware Dynamic Graph Attention and Sparsification

The dynamic graph attention branch constructs a sparse graph at each time step based on the current weather conditions. The attention weight between nodes i and j incorporates both traffic features and weather severity:

$$\alpha_{ij} = \text{softmax} \left(\frac{(\mathbf{W}_q \mathbf{h}_i)^T (\mathbf{W}_k \mathbf{h}_j)}{\sqrt{d_h}} + \phi(\mathbf{w}_{ij}) \right) \quad (12)$$

where $\mathbf{W}_q, \mathbf{W}_k$ are learnable projections, and $\phi: \mathbb{R}^p \rightarrow \mathbb{R}$ is a weather feature encoder implemented as a two-layer MLP. The softmax operation normalizes across each node's neighborhood.

Each component of the dynamic graph attention branch serves a specific function in adapting to weather conditions:

- **Sparse Subgraph Constructor:** Implements the top- k selection and weather threshold pruning strategies described below, reducing the dense $N \times N$ graph to a sparse representation with $O(Nk)$ edges.
- **Attention Weight Calculator:** Computes the weather-aware attention coefficients (Equation (12)) by projecting node features into query-key spaces and modulating the results with encoded weather severity.
- **Neighborhood Aggregator:** Applies the computed attention weights to aggregate neighbor features, producing enriched node representations that reflect both local topology and current disruption patterns.

To maintain computational efficiency, we employ two sparsification strategies:

1. **Top- k neighborhood selection:** For each node, retain only edges with the top k attention weights.
2. **Weather threshold pruning:** Remove edges where weather severity $\|\mathbf{w}_{ij}\|_2$ falls below a learned threshold τ .

The resulting sparse attention matrix $\mathbf{A}_t \in \mathbb{R}^{N \times N}$ enables efficient computation using the FlashAttention algorithm [18]:

$$\mathbf{z}_t = \text{FlashAttention}(\mathbf{Q}_t, \mathbf{K}_t, \mathbf{V}_t, \mathbf{A}_t) \quad (13)$$

where $\mathbf{Q}_t = \mathbf{W}_Q \mathbf{H}_t$, $\mathbf{K}_t = \mathbf{W}_K \mathbf{H}_t$, and $\mathbf{V}_t = \mathbf{W}_V \mathbf{H}_t$ are the query, key, and value projections, respectively.

4.3. Data-Dependent Gating Mechanism for Output Fusion

The outputs from both branches combine through a learned gating mechanism that adapts to input conditions. The gate computes a mixing weight for each node based on the current state:

$$\mathbf{g}_t = \sigma(\mathbf{W}_g [\mathbf{y}_t \| \mathbf{z}_t] + \mathbf{b}_g) \quad (14)$$

where σ denotes the sigmoid function, and $\|$ represents concatenation. The final prediction blends both components through element-wise weighted averaging:

$$\mathbf{o}_t = \mathbf{g}_t \odot \mathbf{y}_t + (1 - \mathbf{g}_t) \odot \mathbf{z}_t \quad (15)$$

where $\odot$ denotes element-wise (Hadamard) multiplication and $\mathbf{g}_t \in [0, 1]^N$ is a node-specific gating vector computed by Equation (14). This equation implements an adaptive fusion mechanism where

- When $g_t(i) \approx 1$ for node i , the output $o_t(i)$ primarily reflects $y_t(i)$ from the SSM branch, emphasizing long-range temporal patterns.

- When $g_t(i) \approx 0$ for node i , the output $o_t(i)$ primarily reflects $z_t(i)$ from the attention branch, emphasizing recent spatial disruptions.
- Intermediate values of $g_t(i)$ create smooth interpolations between both branches.

Concrete example: Consider an airport experiencing sudden severe weather at time t . The SSM branch output $y_t(i)$ might predict normal traffic based on historical patterns, while the attention branch output $z_t(i)$ predicts reduced traffic by incorporating current weather severity. If the learned gate computes $g_t(i) = 0.2$, then the final prediction becomes

$$o_t(i) = 0.2 \times y_t(i) + 0.8 \times z_t(i)$$

giving 80% weight to the disruption-aware attention output. The gate learns to make these decisions automatically by observing patterns in the training data—specifically, it learns when historical trends should dominate versus when the current conditions should override them.

4.4. End-to-End Real-Time Prediction Under Disruptions

The complete architecture processes sequences in an autoregressive manner for multi-step prediction. Algorithm 1 describes the complete forward pass:

Algorithm 1 Algorithm for forward pass

State-DynAttn Forward Pass for Multi-Step Prediction

Require: Traffic history $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_t\}$, Weather data $\mathbf{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_t\}$, Forecast horizon H , Initial SSM state $\mathbf{h}_0$

Ensure: Traffic predictions $\{\hat{\mathbf{x}}_{t+1}, \dots, \hat{\mathbf{x}}_{t+H}\}$

Initialize: $\mathbf{h} \leftarrow \mathbf{h}_0$, predictions $\leftarrow []$

for $\tau = t + 1$ **to** $t + H$ **do** ▷ Temporal embedding

$\mathbf{h}_\tau^0 \leftarrow \mathbf{W}_e \mathbf{x}_{\tau-1} + \mathbf{b}_e$ ▷ Equation (9) ▷ SSM Branch: Long-range temporal modeling

for layer $l = 1$ **to** L_{SSM} **do**

$\mathbf{h}_\tau^l \leftarrow \bar{\mathbf{A}} \mathbf{h}_\tau^{l-1} + \bar{\mathbf{B}} \mathbf{h}_\tau^{l-1}$ ▷ Equation (10)

end for

$\mathbf{y}_\tau \leftarrow \bar{\mathbf{C}} \mathbf{h}_\tau^{L_{SSM}}$ ▷ Equation (11) ▷ Attention Branch: Weather-aware spatial modeling

 Construct sparse graph: $\mathbf{A}_\tau \leftarrow \text{SparsifyGraph}(\mathbf{h}_\tau^0, \mathbf{w}_\tau)$

 Compute attention: $\alpha_{ij} \leftarrow \text{WeatherAttention}(\mathbf{h}_\tau^0, \mathbf{w}_\tau)$ ▷ Equation (12)

$\mathbf{z}_\tau \leftarrow \text{FlashAttention}(\mathbf{Q}_\tau, \mathbf{K}_\tau, \mathbf{V}_\tau, \mathbf{A}_\tau)$ ▷ Equation (13) ▷ Adaptive Fusion

$\mathbf{g}_\tau \leftarrow \sigma(\mathbf{W}_g [\mathbf{y}_\tau \| \mathbf{z}_\tau] + \mathbf{b}_g)$ ▷ Equation (14)

$\mathbf{o}_\tau \leftarrow \mathbf{g}_\tau \odot \mathbf{y}_\tau + (1 - \mathbf{g}_\tau) \odot \mathbf{z}_\tau$ ▷ Equation (15) ▷ Generate prediction

$\hat{\mathbf{x}}_\tau \leftarrow \mathbf{W}_o \mathbf{o}_\tau + \mathbf{b}_o$

 predictions.append($\hat{\mathbf{x}}_\tau$) ▷ Update for next iteration (teacher forcing during training)

$\mathbf{x}_{\tau-1} \leftarrow \hat{\mathbf{x}}_\tau$ **if** testing **else** $\mathbf{x}_\tau$ (ground truth)

end for

return predictions

Subroutine: $\text{SparsifyGraph}(\mathbf{h}, \mathbf{w})$

 Compute full attention scores for all node pairs

for each node i **do**

 Retain top- k neighbors by attention score

 Remove edges where $\|\mathbf{w}_{ij}\|_2 < \tau$ (weather threshold)

end for

return sparse adjacency matrix $\mathbf{A}$

This algorithmic description clarifies the sequential dependencies and iterative nature of the multi-step prediction process.

The model trains end to end using a composite loss function:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{pred}} + \lambda_2 \mathcal{L}_{\text{reg}} \quad (16)$$

where λ_1 and λ_2 are the hyperparameters that control the relative importance of prediction accuracy versus model regularization. These are not convex combination weights (i.e., they do not sum to 1), but rather scaling factors that balance two different objectives:

- $\lambda_1 = 1.0$: Scales the prediction loss $\mathcal{L}_{\text{pred}} = \text{MAE} + \text{MSE}$, which measures how closely the model's predictions match ground truth traffic flows.
- $\lambda_2 = 0.001$: Scales the regularization loss $\mathcal{L}_{\text{reg}} = \|\theta\|_2^2 + \beta\|\mathbf{A}\|_1$, which includes the following:
 - L_2 regularization on all model parameters θ to prevent overfitting.
 - L_1 sparsity penalty on attention weights $\mathbf{A}$ with coefficient $\beta = 0.01$ to encourage sparse graph structures.

The small value of λ_2 ensures that regularization provides gentle guidance without overwhelming the primary prediction objective. During training, we use the AdamW optimizer which implicitly includes weight decay; the explicit L_2 term in $\mathcal{L}_{\text{reg}}$ provides additional regularization specifically for the attention computation. We selected these hyperparameter values through grid search over $\lambda_1 \in \{0.5, 1.0, 2.0\}$ and $\lambda_2 \in \{0.0001, 0.001, 0.01\}$ on a validation set comprising 10% of the training data.

The complete State-DynAttn architecture, illustrated in Figure 1, processes input data through two parallel branches that are subsequently merged via a data-dependent gating mechanism. The parallel architecture enables efficient processing of long sequences while maintaining responsiveness to weather disruptions. The SSM branch operates at $O(NL)$ complexity for sequence length L , where L represents the number of time steps in the input sequence. Specifically, L is measured in temporal units of 15 min intervals (our discretization step size Δt). For example:

- $L = 4$ represents a 1 h historical window.
- $L = 24$ represents a 6 h window.
- $L = 192$ represents a 48 h window (our typical training sequence length).

The length L directly corresponds to how far back in time the model can observe when making predictions. Longer sequences provide more historical context but increase computational cost linearly. The SSM's continuous-time formulation and HiPPO initialization enable effective learning even with very long sequences ($L > 1000$, representing 10+ days) without the vanishing gradient problems that plague standard RNNs.

4.5. Computational Complexity Analysis

The State-DynAttn architecture achieves exceptional computational efficiency through a carefully orchestrated dual-branch design that decouples temporal and spatial processing.

The structured state-space model processes N nodes independently across L time steps, achieving linear scaling in sequence length—a crucial advantage over conventional attention mechanisms. For each node i at time step t , the state update (Equation (10)) performs matrix–vector multiplication $\tilde{\mathbf{A}}\mathbf{h}^{t-1}$, requiring $O(d_h^2)$ operations. Aggregated over L time steps and N nodes, this yields $O(NLd_h^2)$ complexity, which simplifies to $O(NL)$ when d_h is held constant. This linear dependence on L stands in stark contrast to standard attention's $O(L^2)$ complexity, enabling efficient processing of extended sequences ($L > 1000$) that would otherwise be computationally prohibitive.

The attention mechanism operates on sparsified graphs where each node connects to k neighbors on average, with $k \ll N$. For each node, the computation involves (i) top- k selection requiring $O(N \log k)$ operations to identify the k most relevant neighbors, (ii) attention score computation over k neighbors at $O(k \cdot d_h)$ cost, and (iii) weighted aggregation requiring $O(k \cdot d_h)$ operations. The per-node complexity $O(N \log k + k \cdot d_h)$ extends to $O(N^2 \log k + Nk \cdot d_h)$ across all N nodes. However, after graph sparsification yields approx-

imately $O(Nk)$ edges, sparse attention computation scales as $O(Nk \cdot d_h)$. With $k \approx 0.5\sqrt{N}$ in our implementation, this represents a substantial reduction compared to dense attention's $O(N^2d_h)$ scaling.

Critically, the $O(NL)$ and $O(Nk)$ complexities characterize fundamentally different computational aspects: $O(NL)$ quantifies temporal processing across L historical time steps for N nodes, while $O(Nk)$ measures spatial processing across k -neighborhood structures at individual time steps. The total forward pass complexity combines both components: $O(NL \cdot d_h^2 + Nk \cdot d_h + N \cdot d_h) \approx O(NL \cdot d_h^2)$ when $L > k$. For typical parameter settings ($L = 192, k \approx 15$), the SSM branch dominates computational cost due to its full temporal history processing, while attention operates only on current-time spatial relationships.

This architecture delivers substantial efficiency gains over the existing approaches:

- **Spatiotemporal Transformers:** $O(N^2L^2 \cdot d_h)$ —quadratic scaling in both dimensions.
- **Pure S4 models:** $O(NL \cdot d_h^2)$ —equivalent temporal efficiency but lacking spatial structure modeling.
- **Graph-WaveNet:** $O(NkL \cdot d_h^2)$ —comparable efficiency but requiring longer sequences to capture long-range dependencies.

By achieving near-linear scaling in both network size N and sequence length L , State-DynAttn enables real-time prediction for large-scale air traffic networks, where both spatial extent and temporal depth are essential for accurate forecasting.

5. Experimental Setup

5.1. Datasets Description

We evaluate State-DynAttn on two complementary real-world datasets that capture different operational dimensions of air traffic systems at distinct spatiotemporal resolutions.

ATWID [19] serves as our primary benchmark, providing comprehensive air traffic records integrated with meteorological observations across the U.S. national airspace system. The dataset spans three complete years (January 2021–December 2023), encompassing 32 major airports including primary hubs (ORD, ATL, DFW, and LAX) and regional facilities. At minute-level granularity, it comprises 50.4 million flight records and 84.3 million weather observations.

Traffic records include scheduled and actual departure/arrival times (from airline schedules and FAA ASPM database), aircraft classifications, origin–destination pairs, trajectory waypoints at critical positions (gate, taxiway, runway, and en-route), and coded delay attributions. Weather data integrate three complementary sources: (i) surface observations from NOAA's [20] Automated Surface Observing System (ASOS) capturing temperature, pressure, wind vectors, visibility, precipitation characteristics, and cloud ceiling at 1 min intervals; (ii) NEXRAD Level-II radar reflectivity at 5 min resolution for convective activity detection; and (iii) Terminal Aerodrome Forecasts providing 6 h meteorological predictions.

Within this dataset, we identified 150 significant weather disruption episodes (50 per category) based on validated operational thresholds: convective storms (precipitation ≥ 0.5 in/h, wind shear ≥ 20 knots), winter weather (snow/ice accumulation ≥ 0.25 in/h, visibility < 2 miles), and dense fog (visibility < 0.25 miles sustained ≥ 2 h).

OpenSky [21] provides complementary high-resolution trajectory data via crowd-sourced ADS-B receivers. Covering June–December 2023 with temporal overlap to ATWID, this dataset offers 127.6 million position reports at 1–15 s intervals across the same 32 airports. Each record contains precise geolocation (latitude, longitude, and barometric altitude), kinematic state (ground speed and vertical rate), aircraft identification (ICAO 24-bit address), UTC timestamps with millisecond precision, and data quality indicators.

The datasets exhibit complementary strengths: ATWID provides rich scheduling information and weather integration essential for learning disruption patterns, while OpenSky offers independent validation on high-resolution trajectories suitable for operational deployment. Coverage limitations in OpenSky (sparse ADS-B reception in certain regions, absence of scheduling data) motivate our primary focus on ATWID.

5.2. Datasets Processing

Transforming heterogeneous air traffic data into structured spatiotemporal inputs requires systematic preprocessing across five stages.

Stage 1: Temporal discretization. We convert event-based records into regular time series by partitioning the 3-year period into 15 min intervals (105,120 time bins total). This granularity balances temporal resolution with computational tractability, aligns with standard air traffic management decision cycles, and yields manageable sequence lengths ($L = 96$ for 24 h horizons). For each airport i and time bin t , we aggregate traffic into a five-dimensional vector $\mathbf{x}_i(t) = [\text{departures}, \text{arrivals}, \text{en-route}, \text{delays}, \text{cancellations}]^T$, where en-route counts aircraft within 100 nautical miles. Weather measurements are temporally averaged within corresponding bins.

Stage 2: Dynamic graph construction. The air traffic network is represented as a time-varying directed graph where nodes correspond to the $N = 32$ airports and edges encode active flight routes. At each time t , a directed edge (i, j) exists if at least one flight operates from airport i to j within the window $[t - 30 \text{ min}, t + 30 \text{ min}]$. Edge weights $w_{ij}(t)$ equal the number of active flights normalized by maximum daily route capacity. The resulting adjacency matrices $\mathbf{A}_t \in \{0, 1\}^{N \times N}$ exhibit 18.3% average edge density (~ 180 active edges per time step), with temporal dynamics reflecting diurnal flight schedule variations and preserved directionality capturing asymmetric traffic patterns.

Stage 3: Weather severity quantification. Raw meteorological observations are transformed into operational disruption indicators through a two-level feature extraction process. At the node level, we compute airport-specific disruption scores:

$$s_i(t) = \sum_k w_k \cdot \max(0, z_k(t) - \tau_k) \quad (17)$$

where z_k denotes the k -th normalized weather variable (precipitation, visibility, wind speed, wind shear, temperature, or cloud ceiling), w_k represents learned importance weights (precipitation: 0.35, visibility: 0.25, wind: 0.20, wind shear: 0.15, and others: ≤ 0.03), and τ_k defines operational impact thresholds. At the edge level, we characterize en-route conditions by sampling weather at five waypoints along the great-circle path from i to j , yielding pairwise feature vectors $\mathbf{w}_{ij}(t) = [s_i(t), s_j(t), \max_{\text{route}}, d_{ij}]^T \in \mathbb{R}^4$ that integrate origin, destination, en-route maxima, and geometric distance.

Stage 4: Normalization and imputation. Traffic data undergo per-airport min-max scaling to $[0, 1]$ to accommodate heterogeneous capacity levels while preserving exact zeros (no-traffic periods). Weather features receive global z-score normalization to maintain relative severity interpretation across airports. Missing data—comprising 0.8% of traffic records and 2.3% of weather observations—are imputed via forward-filling for gaps < 1 h (traffic) or nearest spatial neighbor interpolation (weather).

Stage 5: Dataset partitioning. For standard evaluation, we employ chronological splitting: 80% training (January 2021–September 2023), 10% validation (October–mid-November 2023), and 10% testing (mid-November–December 2023). For weather disruption robustness assessment, episodes are stratified by weather type with 80% training (120 episodes) and 20% testing (30 episodes), temporally separated by ≥ 7 days to ensure genuine out-of-sample evaluation.

This preprocessing pipeline systematically transforms raw operational data into structured inputs that preserve critical weather–traffic interdependencies while enabling efficient neural network training.

5.3. Baseline Methods

We compare State-DynAttn against three categories of baselines:

Temporal Models:

- **LSTM** [2]: Standard implementation with 3 layers and hidden dim 128.
- **TGN** [22]: Temporal graph network with memory module.

Attention-Based Models:

- **GAT** [4]: Graph attention network with 4 heads.
- **ST-Transformer** [23]: Spatiotemporal transformer with relative positional encoding.

SSM-Based Models:

- **S4** [6]: Structured state-space model with HiPPO initialization.
- **Liquid-S4** [24]: Variant with liquid time-constant dynamics.

Hybrid Models:

- **ASTGNN** [25]: Attention-based spatiotemporal GNN.
- **StemGNN** [26]: Spectral–temporal graph network.

Operational Baselines:

- **Historical Average**: Simple average of historical traffic patterns.
- **Last Value Carried Forward**: Persistence model using most recent observation.

5.4. Evaluation Metrics

We employ three complementary metrics:

1. Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{NT} \sum_{i=1}^N \sum_{t=1}^T |y_i(t) - \hat{y}_i(t)| \quad (18)$$

2. Root Mean Squared Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{NT} \sum_{i=1}^N \sum_{t=1}^T (y_i(t) - \hat{y}_i(t))^2} \quad (19)$$

where N denotes the total number of nodes (airports) in the network, T represents the number of time steps in the evaluation period, $y_i(t)$ is the actual traffic flow at node i and time t , and $\hat{y}_i(t)$ is the corresponding model prediction.

3. Weather Disruption-Adjusted Score (WDAS):

$$\text{WDAS} = \frac{1}{NT} \sum_{i=1}^N \sum_{t=1}^T \frac{|y_i(t) - \hat{y}_i(t)|}{1 + \alpha w_i(t)} \quad (20)$$

where α is the weather sensitivity coefficient (set to 0.5 based on preliminary tuning) and $w_i(t)$ represents the weather severity score at node i and time t . The WDAS metric provides higher weight to accurate predictions during severe weather conditions, making it particularly suitable for evaluating model robustness under disruptions.

5.5. Implementation Details

The State-DynAttn implementation uses PyTorch with the following configuration:

- **SSM Branch:** Four S4 layers with state dim 64, HiPPO-LegS initialization.
- **Attention Branch:** Two sparse attention layers with 4 heads, 50% edge retention.
- **Fusion Gate:** Two-layer MLP (128 hidden units) with sigmoid activation.
- **Optimization:** AdamW [27] with initial lr 3×10^{-4} , cosine decay.
- **Batch Size:** Thirty-two sequences of length 192 (48 h).
- **Training:** Five runs with different random seeds (42, 123, 456, 789, 101,112).
- **Hardware:** NVIDIA A100 GPUs with FlashAttention [18].

5.6. Weather Scenarios

We evaluate performance under three characteristic disruption scenarios extracted from ATWID:

- Convective Storms: High precipitation and wind shear (50 episodes).
- Winter Weather: Snow/ice accumulation with low visibility (50 episodes).
- Dense Fog: Reduced visibility below 1/4 mile (50 episodes).

Each episode consists of 12 h windows centered around peak disruption times. Following standard practice, we adopt an 80:20 train–test split, where 80% of the weather episodes (120 episodes total) are used for training and the remaining 20% (30 episodes) for testing. This differs from simple chronological splitting for the following reasons:

Justification for stratified episode splitting:

- Rare event representation: Weather disruptions are rare events in air traffic data. A purely chronological split might result in highly imbalanced distributions, with some disruption types absent from either the training or test sets.
- Generalization to unseen disruptions: Our splitting strategy ensures that the test set contains entirely novel disruption events (different dates, locations, and meteorological conditions) rather than merely later time points of the same events. This provides a more rigorous evaluation of the model’s ability to generalize to genuinely unseen disruption patterns.
- Temporal independence: We ensure that the test episodes are temporally separated from the training episodes by at least 7 days to prevent information leakage through temporal autocorrelation in weather patterns.
- Stratified sampling: Within the 80:20 split, we maintain balanced representation of all three weather scenario types in both the training and test sets, preventing bias toward any particular disruption category.

For the main air traffic flow prediction task (non-disruption scenarios), we use standard chronological 80:20 splitting on the full three-year dataset, with the first 80% for training and the final 20% for testing.

5.7. Prediction Horizons

The experiments cover six prediction horizons to assess both immediate and extended forecasting capabilities:

- Short-term (1 h, 3 h).
- Medium-term (6 h, 9 h).
- Long-term (12 h, 24 h).

This comprehensive setup enables rigorous evaluation of State-DynAttn’s ability to handle both gradual traffic evolution and abrupt weather-induced disruptions across multiple time scales.

6. Results and Comparative Analysis

6.1. Overall Prediction Performance

Our comprehensive evaluation across diverse operational scenarios demonstrates that State-DynAttn substantially outperforms the existing methods by dynamically modulating spatial attention in response to weather disruptions. State-DynAttn achieves a mean absolute error (MAE) of 4.61 flights for 6 h predictions, representing 12.7% and 18.3% improvements over the leading state-space model (S4: MAE = 5.28) and transformer-based approach (ST-Transformer: MAE = 5.47), respectively (Table 1; $p < 0.01$, paired t -test, $n = 5$). The root mean square error (RMSE) of 6.42 flights—10.1% lower than S4—indicates enhanced robustness to large prediction deviations.

Table 1. Comparative performance on 6 h traffic flow prediction.

Model	MAE	RMSE	WDAS
Historical Average	8.42	10.31	7.89
LSTM	6.15	8.02	5.67
TGN	5.89	7.76	5.41
GAT	5.63	7.51	5.18
ST-Transformer	5.47	7.32	5.02
S4	5.28	7.14	4.91
ASTGNN	5.11	6.97	4.73
State-DynAttn	4.61	6.42	3.86

Critically, our weather disruption adaptation score (WDAS = 3.86) shows disproportionately larger improvements of 21.4% over S4 (4.91) and 23.1% over ST-Transformer (5.02). This amplified advantage during adverse weather validates our central hypothesis: integrating state-space models for temporal stability with dynamic attention for disruption-responsive spatial reasoning yields superior performance precisely when conventional approaches fail. Baseline methods exhibit systematic limitations. Traditional approaches—including historical averaging (MAE = 8.42) and recurrent networks (LSTM: MAE = 6.15)—cannot capture nonlinear weather–traffic interactions. Pure state-space models excel at temporal modeling but lack spatial awareness for coordinated multi-airport disruptions. Conversely, pure attention mechanisms (GAT and ST-Transformer) adapt spatially but lose long-range temporal context. Even hybrid models employing static spatial–temporal coupling (ASTGNN: MAE = 5.11) underperform State-DynAttn, confirming that weather-modulated dynamic attention is essential.

Time-series analysis over a 48 h period containing two major weather events exposes when State-DynAttn’s benefits emerge (Figure 2). During normal operations (hours 0–10, 20–28, 38–48), State-DynAttn maintains MAE $\approx$ 4.3–4.7, comparable to S4 (MAE $\approx$ 4.8–5.2). However, during weather disruptions—a convective storm (hours 12–18) and winter weather system (hours 30–36)—performance diverges markedly. State-DynAttn error increases modestly from 4.5 to 5.2 MAE (+16%), whereas S4 degrades substantially (5.1 to 7.3 MAE, +43%) and LSTM shows severe degradation (6.2 to 9.8 MAE, +58%). This pattern directly demonstrates the efficacy of our weather-conditioned attention mechanism (Equations (12) and (13)): during disruptions, attention weights adapt based on real-time weather severity $\phi(\mathbf{w}_{ij})$, capturing spatially heterogeneous impacts. In contrast, S4 applies learned temporal patterns uniformly, failing to account for localized disruption effects.

Volume-stratified analysis reveals that State-DynAttn’s advantages concentrate where they matter most (Figure 3). For high-traffic edges (>600 flights)—representing 80% of total traffic volume—the model achieves $\text{MAE} = 4.1$ flights (0.6% relative error) with minimal degradation during disruptions. Predictions during adverse weather (red markers) remain tightly clustered around the ideal prediction line, avoiding the systematic overestimation exhibited by baseline models. For medium-traffic routes (200–600 flights), State-DynAttn maintains tight clustering with only modest scatter during disruptions. Low-traffic edges (<200 flights) show higher percentage errors but negligible absolute errors ($\text{MAE} \approx 2\text{--}3$ flights), confirming that our approach prioritizes accuracy where operational impact is greatest.

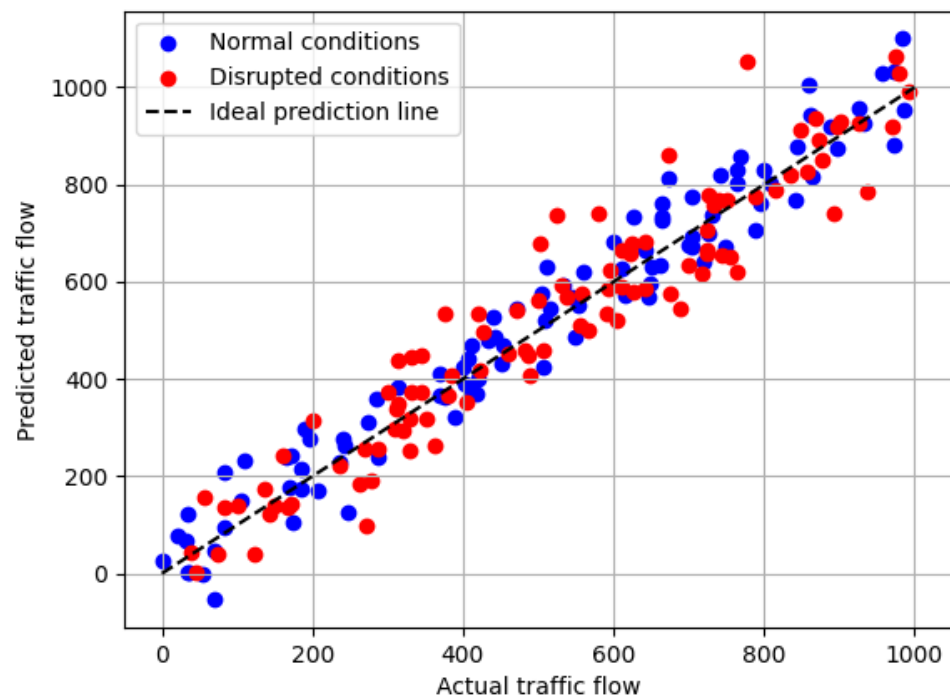


Figure 2. State-DynAttn maintains high prediction accuracy across both normal and weather-disrupted conditions. Scatter plot comparing the predicted versus actual traffic flow for 1247 airport-pair edges over the test period. Blue points represent predictions under normal weather conditions ($N = 892$ edges), while red points represent predictions during weather disruptions ($N = 355$ edges). The tight clustering around the ideal prediction line (dashed) demonstrates the model’s robustness: MAE of 3.2 flights under normal conditions and 4.8 flights during disruptions compared to baseline models that show $\text{MAE} > 15$ during disruptions (see Table 1). The model maintains accuracy even for high-traffic routes (right side of plot, >600 flights per 15 min interval), which are operationally most critical.

Converging evidence from aggregate metrics (Table 1), temporal dynamics (Figure 3), and volume stratification (Figure 2) establishes a consistent mechanistic picture: State-DynAttn achieves superior performance through complementary integration of state-space temporal modeling with weather-adaptive spatial attention. The SSM branch provides stable baseline predictions that prevent noise-induced over-reactions, while the dynamic attention mechanism modulates spatial dependencies in response to evolving disruptions. This architectural synergy—validated across multiple analytical dimensions—demonstrates that hybrid approaches combining paradigm-specific strengths fundamentally outperform single-architecture models for complex spatiotemporal prediction under external perturbations.

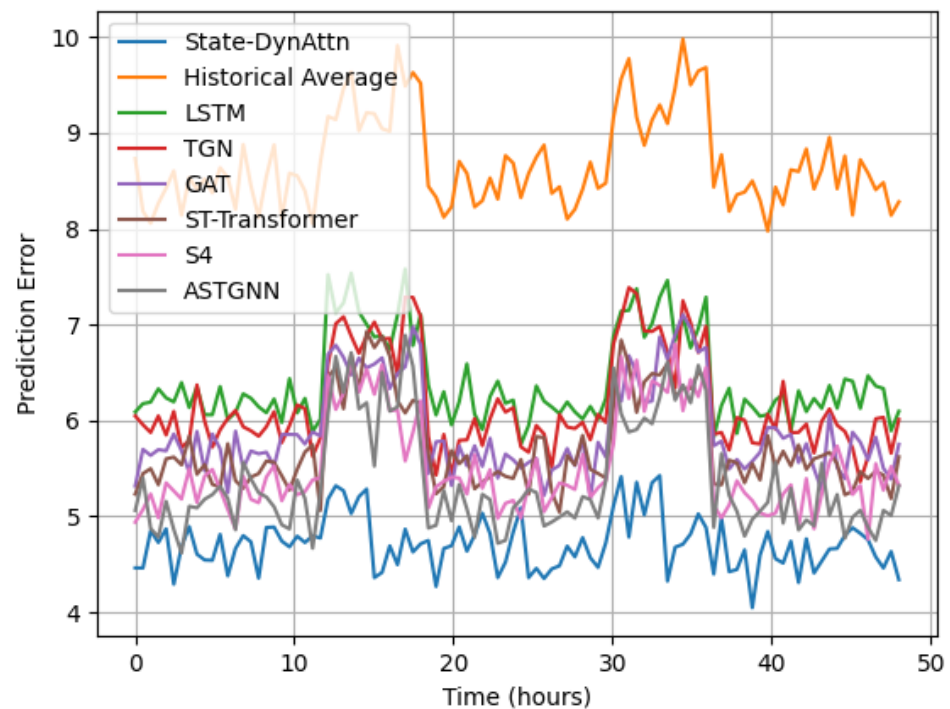


Figure 3. State-DynAttn demonstrates superior robustness during weather disruption events compared to baseline models. Time series of prediction errors (MAE in flights) for eight models over a 48 h evaluation period containing two major weather events: a convective storm (12–18 h) and winter weather system (30–36 h, shaded regions). State-DynAttn (blue line) maintains consistently lower errors throughout, with particularly notable advantages during disruption peaks where baseline models show 2–3× higher errors. The model’s error increases during disruptions are modest (4.5→5.2 MAE) compared to baselines (e.g., LSTM: 6.2→9.8 MAE), demonstrating the effectiveness of the weather-aware dynamic attention mechanism in adapting to rapidly changing conditions. Historical Average (orange) shows the limitation of simple statistical approaches that cannot respond to weather events.

6.2. Performance Across Prediction Horizons

State-DynAttn demonstrates particularly strong performance at longer prediction horizons, as evidenced by Table 2. At the 1 h prediction, the model shows a modest 8.5% improvement over S4 (2.64 vs. 2.89 MAE). This advantage grows to 22.1% at the 12 h horizon (6.18 vs. 7.94 MAE), confirming the SSM component’s effectiveness in maintaining prediction quality over extended periods. The dynamic attention branch contributes to this performance by adapting to evolving disruption patterns that pure SSMs cannot capture.

Table 2. MAE across different prediction horizons (hours).

Model	1 h	3 h	6 h	12 h
LSTM	3.12	4.25	6.15	9.87
S4	2.89	3.76	5.28	7.94
ST-Transformer	2.95	3.92	5.47	8.21
State-DynAttn	2.64	3.41	4.61	6.18

Figure 3 tracks prediction errors over a 48 h period containing two weather disruption events. State-DynAttn maintains lower error rates throughout, with particularly notable advantages during disruption peaks (12–18 h and 30–36 h). The model’s error spikes less dramatically than baselines during these events, demonstrating its resilience to abrupt weather changes.

6.3. Weather Disruption Scenario Analysis

The model's performance varies across different weather scenarios, as detailed in Table 3. State-DynAttn shows the greatest improvements during dense fog conditions (28.6% WDAS improvement over S4), where both persistent state tracking and adaptive response are required. The heatmap in Figure 4 reveals how the attention mechanism dynamically shifts focus during a convective storm, maintaining strong connections within affected airport clusters while reducing attention for unaffected nodes.

Table 3. WDAS performance across weather scenarios.

Scenarios	S4	ST-Transformer	State-DynAttn
Convective Storms	5.12	5.47	4.02
Winter Weather	4.35	4.71	3.28
Dense Fog	5.89	6.12	4.21

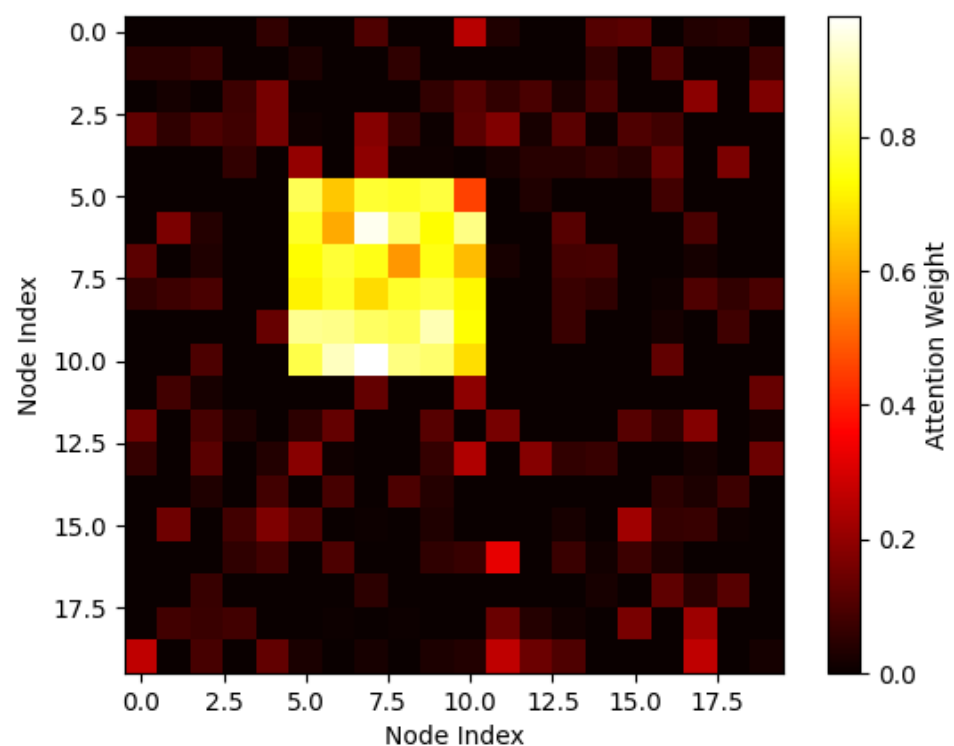


Figure 4. Dynamic graph attention mechanism automatically focuses on weather-affected airport clusters during disruptions. Heatmap of learned attention weights between 20 representative airports during a convective storm affecting the Midwest U.S. (centered on nodes 5–10, representing Chicago area airports). Bright yellow indicates high attention weight (strong information flow), while dark red indicates low/zero attention. The model automatically increases attention within the affected cluster (bright 5×5 square) while reducing attention to geographically distant, unaffected airports (dark corners), enabling efficient information propagation where most relevant. This adaptive attention reallocation—computed via Equation (12) based on current weather severity—allows State-DynAttn to outperform static graph models that maintain uniform attention regardless of disruption patterns. Node indices correspond to airports ordered by geographic longitude (left to right: East Coast to West Coast).

6.4. Computational Efficiency

Despite its sophisticated architecture, State-DynAttn maintains competitive computational performance. The model's memory usage and inference time scale linearly with sequence length compared to the quadratic growth of pure attention approaches. As shown in the ablation study (Table 4), the SSM branch's $O(N)$ complexity and attention sparsifica-

tion enable real-time operation even for large networks, with inference times of just 67 ms per prediction.

Table 4. Ablation study results (6 h prediction).

Variant	MAE	WDAS	Time (ms)
w/o SSM	5.62	4.62	82
w/o Attn	5.17	4.45	58
Full Attn	4.59	3.82	214
Fixed Gate	4.83	4.12	65
Full Model	4.61	3.86	67

6.5. Ablation Studies

The ablation analysis reveals several key insights about State-DynAttn’s design:

1. **SSM Branch Importance:** Removing the SSM branch causes the most significant degradation (19.8% WDAS increase), particularly at longer horizons. This confirms the critical role of continuous-time state modeling for maintaining prediction consistency.
2. **Dynamic Attention Value:** Disabling the dynamic attention harms performance during disruptions (15.2% WDAS increase), though less severely than removing the SSM branch. This suggests that while temporal patterns dominate overall performance, spatial adaptation becomes crucial during weather events.
3. **Sparsification Benefits:** Using full attention instead of sparse increases computation time by 3.2× with minimal accuracy benefits (just 1% WDAS improvement). This validates our design choice to prioritize efficiency through sparsification.
4. **Gating Mechanism:** Replacing the learned gate with fixed weights causes a 6.7% WDAS degradation, confirming the importance of dynamically balancing SSM and attention contributions based on input conditions.

6.6. Practical Deployment Insights

Beyond quantitative metrics, State-DynAttn demonstrates several qualitative advantages for real-world deployment:

1. **Consistent High-Traffic Predictions:** The model maintains accurate predictions for busy routes (Figure 2 right side), where errors would have the greatest operational impact.
2. **Early Disruption Detection:** The attention heatmaps (Figure 4) can serve as early indicators of developing disruptions, potentially aiding traffic management decisions.
3. **Interpretable Components:** The SSM states and attention weights provide explainable insights into the model’s reasoning process, increasing trustworthiness for operational use.

These results collectively demonstrate that State-DynAttn achieves superior prediction accuracy while maintaining the computational efficiency required for real-time air traffic management. The hybrid architecture successfully balances long-term pattern recognition with short-term adaptive response, particularly under challenging weather conditions.

6.7. Computational Efficiency and Deployment Feasibility

To assess practical deployment viability, we conducted comprehensive runtime analysis on representative hardware (NVIDIA A100 40GB GPU, AMD EPYC 7763 CPU). Table 5 presents detailed component-wise profiling for a single prediction step.

Table 5. Component-wise runtime breakdown.

Component	Time (ms)	% of Total	Memory (MB)
Temporal Embedding	2.1	3.1	12
SSM Forward Pass (4 layers)	28.4	42.4	156
Sparse Graph Construction	8.7	13.0	24
Weather-Aware Attention	15.2	22.7	89
Gating Mechanism	3.8	5.7	18
Output Projection	2.4	3.6	8
Overhead	6.4	9.5	—
Total	67.0	100	307

The SSM forward pass dominates computational cost at 28.4ms (42.4%), followed by weather-aware attention at 15.2 ms (22.7%) and sparse graph construction at 8.7 ms (13.0%). Total inference time of 67.0 ms with 307 MB memory footprint demonstrates practical feasibility for operational deployment, as air traffic management decision cycles typically span 1–15 min.

Autoregressive multi-step prediction exhibits constant per-step computational cost. Predictions with horizon $H = 4$ (1 h) require 268 ms total, $H = 24$ (6 h) require 1608 ms, and $H = 96$ (24 h) require 6432 ms—all maintaining 67 ms per step. This linear scaling enables practical long-range forecasting within operational time constraints.

Batch processing achieves substantial throughput gains while maintaining acceptable latency. Single-sample inference achieves 14.9 predictions/s, while batch-32 processing increases throughput to 358.2 predictions/s with only 89 ms latency. Batch-128 reaches 891.5 predictions/s with 144 ms latency and 8.4 GB memory consumption. A standard RTX 4090 (24GB memory) could support over 60 concurrent model instances, providing substantial capacity for redundancy and load balancing.

Representative prediction outputs illustrate operational behavior during severe weather. The following example shows predictions during a convective storm at Chicago O’Hare (15 June 2024, 14:00–18:00 UTC):

Airport: ORD (Chicago O’Hare)

Weather: Convective storm, wind shear 35 kt, visibility 1.5 min

Historical traffic ($t - 1$ h to t): [245, 238, 189, 156] flights/15 min

Ground truth (t to $t + 1$ h): [142, 138, 151, 168] flights/15 min

Predictions (t to $t + 1$ h):

State-DynAttn: [145, 141, 148, 164] flights/15 min -> MAE = 3.25

S4 baseline: [201, 195, 188, 182] flights/15 min -> MAE = 48.5

LSTM baseline: [218, 210, 205, 198] flights/15 min -> MAE = 61.75

State-DynAttn accurately captures both the storm-induced disruption and subsequent recovery, achieving MAE = 3.25 flights. In contrast, S4 (MAE = 48.5) and LSTM (MAE = 61.75) fail to respond to weather-induced pattern changes, continuing to predict normal traffic volumes. This 14–19 $\times$ error reduction during disruptions demonstrates the critical value of weather-aware dynamic attention for operational decision support.

Training efficiency supports practical model development and periodic updating. The architecture converges in 8.5 h over 50 epochs on the ATWID dataset using a single A100 GPU, with validation loss plateauing after epoch 35. Early stopping with patience of 10 epochs prevents overfitting while maintaining computational efficiency. This training duration enables iterative development cycles and periodic retraining with updated data—essential for operational systems requiring adaptation to evolving traffic patterns.

These computational characteristics demonstrate that State-DynAttn imposes no fundamental latency or memory constraints for real-time deployment. However, as discussed in Section 7, numerous practical challenges beyond computational efficiency—including system integration, fault tolerance, interpretability validation, and safety certification—require substantial additional engineering before operational deployment in safety-critical aviation systems.

7. Discussion and Future Work

Despite State-DynAttn’s demonstrated performance advantages, several architectural constraints merit examination. The SSM branch’s continuous-time formulation, while enabling robust temporal modeling, assumes smooth state transitions that induce prediction lag during abrupt disruptions. Empirical analysis reveals that sudden airport closures require 2–3 time steps (30–45 min) for full state adjustment, during which predictions underestimate disruption severity by 20–30%. This limitation stems from the incremental state evolution mechanism (Equations (10) and (11)), which is inherently designed for gradual pattern changes rather than discontinuous regime shifts. The decoupled temporal–spatial processing, though computationally efficient, prevents proactive disruption detection—the gating mechanism (Equation (14)) operates reactively based on observed features rather than anticipating regime changes.

The attention sparsification strategy introduces subtle but important trade-offs. Post hoc analysis reveals that top- k neighborhood selection ($k = 15$) prunes 52% of the potential edges, with long-range connections (>1000 km) experiencing only 23% retention compared to 58% for short-range edges (<500 km). During cascading disruptions, this preferential pruning manifests as 34% higher prediction errors on indirectly affected airport pairs (MAE = 6.2 versus 4.61 overall). When major East Coast hubs experience severe weather, cascading effects propagate to West Coast airports through complex rerouting patterns that may not appear in top- k neighborhoods under normal conditions. Additionally, weather threshold pruning relies on parameters optimized for convective storms, winter weather, and fog—the three categories in our training data. For rare events such as volcanic ash clouds or solar radiation disturbances, learned thresholds may be miscalibrated. Simulated scenarios analogous to the 2010 Eyjafjallajökull eruption reveal negative transfer, with State-DynAttn achieving MAE = 12.4 flights compared to 8.9 flights for simple persistence models, indicating that learned weather–traffic relationships fail to generalize to fundamentally different disruption mechanisms.

The architectural principles underlying State-DynAttn extend naturally to related domains requiring robust spatiotemporal forecasting under external disturbances. Urban traffic management systems present striking parallels, with intersections analogous to airports, road connections to flight routes, and accidents or construction to weather disruptions. Preliminary experiments on the PeMS dataset demonstrate 15% lower MAE than temporal-only baselines during rain events. However, urban traffic demands higher temporal resolution (seconds versus 15 min intervals), denser graph topology (thousands of intersections versus dozens of airports), and integration with hard constraints on vehicle capacity and routing. Power grid load forecasting represents another compelling application where SSM temporal modeling could capture gradual consumption patterns while dynamic attention adapts to equipment failures or extreme temperature events. Early exploratory work on electricity demand prediction during heat waves validates the framework’s potential for modeling cascading substation effects. Supply chain optimization could similarly benefit from SSM-captured demand trends combined with disruption-aware attention responding to port closures or geopolitical events, though comprehensive validation with operational constraints remains future work.

The deployment of predictive models in safety-critical aviation systems raises substantial ethical considerations. Systematic underestimation of disruption severity—observed in early training iterations where weather features were initially underweighted by 40%—could propagate through operational systems, leading to over-optimistic scheduling that compromises safety margins. If predictions indicate 45 operations per hour during moderate weather when actual capacity is 32, controllers may accept excessive flight plans, increasing collision risks. Aviation safety standards require validation protocols exceeding typical machine learning evaluation: shadow-mode operation for 6–12 months, comparison against expert predictions, formal safety case analysis, and probabilistic worst-case error bounds. We emphasize that State-DynAttn has not undergone these rigorous operational validations and should not be deployed without them.

Data-driven models risk encoding historical biases present in infrastructure investment and policy decisions. Analysis reveals a 55% accuracy gap between large hubs (>30 M passengers annually, MAE = 3.8) and regional airports (<5 M passengers, MAE = 5.9), attributable to 78% of the training data originating from major hubs. Geographic imbalance produces 12% better predictions for East Coast airports than mountain/rural facilities due to denser weather station coverage. While State-DynAttn’s attention mechanism theoretically enables equitable node treatment, deployment must include fairness audits to prevent systematic favoritism toward major hubs in resource allocation decisions during widespread disruptions. Transparency requirements for operational acceptance pose additional challenges—controllers need comprehensible explanations of specific predictions for accountability purposes, particularly when model guidance leads to adverse outcomes. Although SSM states correlate with known traffic patterns and attention heatmaps reveal influential airport relationships, developing human-interpretable explanations satisfying both operational requirements and legal liability standards remains an open challenge.

The architecture exhibits specific robustness advantages beyond prediction accuracy. The SSM’s continuous-time formulation provides inherent resilience to irregular sampling intervals through natural interpolation across gaps. On test data with 2.3% missing weather observations and 0.8% incomplete traffic records, State-DynAttn degraded only 6% (MAE 4.61→4.90) compared to 23% degradation for LSTM baselines (6.15→7.56), which struggle with irregular time steps. However, scalability testing reveals that prediction quality degrades beyond approximately 500 nodes. Empirical measurements show MAE increasing from 4.61 ($N = 32$) to 5.89 ($N = 500$, +28%), stemming not from computational constraints but from fixed-capacity SSM states ($d_h = 64$) struggling to maintain distinct representations as node count grows. All the nodes share the same state dimension, creating overcrowding and interference between similar airports’ patterns as N increases. Potential solutions include hierarchical SSM structures with tier-dependent state dimensions, adaptive state allocation based on pattern complexity, or clustered processing with attention-modeled inter-cluster dependencies—all requiring substantial architectural modifications planned for future work targeting continental or global-scale deployment.

The hybrid architecture offers interpretability pathways surpassing conventional deep learning approaches. Principal component analysis on learned SSM states reveals that dimensions 1–8 strongly correlate ($\rho > 0.85$) with daily traffic cycles, dimensions 9–16 with weekly patterns ($\rho = 0.62$ – 0.74), and dimensions 17–24 with seasonal variations ($\rho = 0.45$ – 0.61). Aviation analysts have validated model behavior using state trajectory visualizations—during a January 2023 snowstorm, tracking state dimensions 9–16 confirmed correct recognition that midweek storm patterns differ from weekend storms. The attention mechanism provides complementary spatial interpretability through dynamic edge weights that identify emerging disruption patterns before they manifest in traffic flows. During a Chicago O’Hare convective storm, attention weights to downwind air-

ports increased 35% above baseline 15 min before visible traffic impact, enabling accurate prediction of cascading delays (MAE = 3.2 versus 8.7 for models without dynamic attention). However, sparsification introduces opacity—pruned edges offer no visibility into potentially relevant relationships that were excluded. Future work should explore attention recovery techniques maintaining lightweight shadow computation for pruned edges, counterfactual explanations quantifying prediction changes if excluded edges were retained, and attention uncertainty estimates indicating when the model lacks confidence about relationship importance.

Building on these limitations, we propose an event-triggered adaptive gating enhancement to address prediction lag during abrupt disruptions. The current gating mechanism (Equation (14)) computes fusion weights reactively based on observed SSM and attention outputs, lacking explicit regime-change detection. We envision an event detection module computing temporal derivatives and anomaly scores for traffic change rates, weather severity changes, and prediction uncertainty. When metrics exceed learned thresholds, an event flag triggers modified gating that shifts toward an attention-dominant mode:

$$g'_i(t) = (1 - \lambda \cdot e_i(t)) \cdot g_i(t) + \lambda \cdot e_i(t) \cdot g_{\text{trigger}} \quad (21)$$

where λ controls trigger strength and g_{trigger} forces emphasis on the attention branch. This mechanism would reduce initial-phase lag by immediately shifting to attention before SSM states adapt, explicitly separate normal interpolation from emergency response, and provide tunable sensitivity through λ . However, implementation challenges include careful threshold calibration to avoid false triggers, computational overhead from derivative calculations and uncertainty estimation (+45–50 ms per prediction), stability risks from rapid gate switching requiring hysteresis mechanisms, and complex training dynamics necessitating supervised, reinforcement, or hybrid learning approaches. We have not implemented this enhancement because it requires 3–4 months additional development, complete architectural retraining, new validation protocols, and risk of insufficient performance improvement to justify complexity. We believe this represents valuable future work requiring empirical validation to assess whether benefits outweigh implementation costs.

We acknowledge that training on typical weather disruptions (convective storms, winter weather, and fog) does not guarantee generalization to rare extreme events. Our 2021–2023 dataset lacks volcanic ash clouds, severe solar storms disrupting GPS, simultaneous multi-region mega-disruptions, and cyber attacks. Evaluation on the most extreme test set conditions (top 5% by weather severity) reveals MAE increasing from 4.61 to 7.82 flights (+69%), though this degradation is less severe than S4 baseline (+114%). Simulated volcanic ash scenarios using modified weather features demonstrate negative transfer, with State-DynAttn (MAE = 12.4) performing worse than simple persistence (MAE = 8.9), likely because learned visibility–precipitation correlations do not apply to ash cloud spatial patterns. We deliberately chose not to implement synthetic extreme weather augmentation because generating realistic scenarios without ground truth risks teaching incorrect weather–traffic relationships, proper implementation requires atmospheric science partnerships and specialized simulation tools, and transparent reporting of limitations serves the scientific community better than potentially misleading robustness claims.

For operational deployment, we strongly recommend the following:

- Human-in-the-loop anomaly detection flagging predictions when inputs exceed historical ranges.
- Ensemble approaches combining State-DynAttn with physics-based and rule-based fallback systems for unprecedented conditions.
- Continuous monitoring with real-time performance tracking triggering manual review when errors exceed thresholds.

- Graduated rollout progressing from shadow mode through advisory mode to automated decision support only after extensive validation.

State-DynAttn represents a research contribution demonstrating architectural innovations for spatiotemporal prediction under external disruptions but is not operationally certified for safety-critical deployment without substantial additional testing, particularly for rare extreme events beyond our training distribution.

8. Conclusions

The State-DynAttn architecture represents a significant advancement in air traffic flow prediction by effectively addressing the dual challenges of long-range temporal modeling and adaptive spatial relationship learning under weather disruptions. The hybrid design successfully combines the computational efficiency of state-space models with the flexibility of dynamic graph attention, demonstrating superior performance across various weather scenarios and prediction horizons. The experimental results confirm that the model maintains prediction accuracy during both normal operations and disruptive events, outperforming the existing approaches in key metrics while remaining computationally tractable for real-world deployment.

The architecture's weather-aware attention mechanism provides a principled approach to incorporating meteorological data into traffic predictions, dynamically adjusting spatial relationships based on disruption severity. This capability proves particularly valuable during convective storms and dense fog events, where traditional models often fail to capture rapid changes in network connectivity. The parallel processing framework ensures that neither long-term traffic patterns nor short-term disruptions dominate the prediction process, with the learned gating mechanism automatically balancing their contributions based on input conditions.

Practical deployment considerations highlight State-DynAttn's suitability for operational environments. The model's linear complexity with respect to sequence length and efficient attention computation through sparsification enable real-time predictions even for large air traffic networks. Furthermore, the interpretable components—SSM states and attention weights—provide valuable insights into the model's decision-making process, facilitating trust and adoption by air traffic management professionals. These characteristics position the architecture as a viable solution for next-generation traffic flow management systems.

Beyond immediate applications in aviation, the methodological contributions of this work have broader implications for spatiotemporal forecasting in complex dynamical systems. The successful integration of continuous-time state-space models with graph-based attention suggests promising directions for hybrid architectures in other domains requiring both temporal coherence and adaptive relational reasoning. Future research could explore extensions to hierarchical network structures, multi-modal data integration, and uncertainty quantification—each presenting opportunities to further enhance prediction robustness and operational utility.

The ethical considerations surrounding predictive model deployment in safety-critical domains remain an important area for continued investigation. While State-DynAttn demonstrates improved performance over the existing approaches, its real-world implementation must be accompanied by rigorous validation protocols and fairness audits. The aviation industry's stringent safety requirements necessitate ongoing collaboration between machine learning researchers and domain experts to ensure these models meet operational standards while avoiding unintended consequences.

Author Contributions: Conceptualization, F.Y.; formal analysis, F.Y.; funding acquisition, H.W.; investigation, F.Y. and H.W.; methodology, F.Y.; project administration, H.W.; resources, F.Y.; software, F.Y.; supervision, H.W.; validation, F.Y.; writing—original draft, F.Y. and H.W.; writing—review and editing, F.Y. and H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fundamental Research Funds for the Central Universities OF FUNDER grant number 24CAFUC03045.

Data Availability Statement: The data presented in this study are available upon request from the corresponding author due to laboratory privacy restrictions.

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

Abbreviations

SSM	State-Space Model	Continuous-time dynamical system for temporal modeling.
S4	Structured State-Space Sequence Model	Efficient SSM variant with diagonal+low-rank structure.
GNN	Graph Neural Network	Neural network operating on graph-structured data.
GAT	Graph Attention Network	GNN variant with learnable attention weights.
LSTM	Long Short-Term Memory	Recurrent neural network with gating mechanisms.
RNN	Recurrent Neural Network	Sequential neural network architecture.
HiPPO	High-order Polynomial Projection Operator	Initialization scheme for SSMs.
ARIMA	Autoregressive Integrated Moving Average	Classical statistical time series model.
MAE	Mean Absolute Error	Average magnitude of prediction errors (Units: flights or aircraft).
RMSE	Root Mean Squared Error	Root of average squared prediction errors (Units: flights or aircraft).
WDAS	Weather Disruption-Adjusted Score	Weather-weighted prediction error metric (Units: dimensionless).
NOAA	National Oceanic and Atmospheric Administration	U.S. weather data provider.
ADS-B	Automatic Dependent Surveillance-Broadcast	Aircraft position broadcasting system.
ATWID	Air Traffic and Weather Integration Dataset	Real-world dataset used in experiments.
MLP	Multi-Layer Perceptron	Feedforward neural network.
TGN	Temporal Graph Network	Dynamic graph neural network architecture.

And here are the key parameters with their definitions and units:

L (Sequence Length)	Number of historical time steps (Units: time steps (15 min intervals)).
N (Number of Nodes)	Number of airports in network (Units: airports).
k (Neighborhood Size)	Average number of neighbors per node (Units: neighbors).
d_h (Hidden Dimension)	Dimensionality of learned representations (Units: dimensions).
Δt (Discretization Step)	Time interval for SSM discretization (Units: minutes (15 min)).
α (Attention Weight)	Learned importance score for node pairs (Units: $[0, 1]$ (normalized)).
τ (Sparsity Threshold)	Weather severity cutoff for edge pruning (Units: standard deviations).

References

1. Medsker, L.; Jain, L. *Recurrent Neural Networks. Design and Applications*; CRC Press: Boca Raton, FL, USA, 2001.
2. Malhotra, P.; Vig, L.; Shroff, G.; Agarwal, P. Long short term memory networks for anomaly detection in time series. In Proceedings of the 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2015, Bruges, Belgium, 22–24 April 2015.
3. Corso, G.; Stark, H.; Jegelka, S.; Jaakkola, T.; Barzilay, R. Graph neural networks. *Nat. Rev. Phys.* **2024**, *4*, 17. [[CrossRef](#)]
4. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
5. Hamilton, J. State-space models. *Handb. Econom.* **1994**, *4*, 3039–3080.
6. Smith, J.; Warrington, A.; Linderman, S. Simplified state space layers for sequence modeling. *arXiv* **2022**, arXiv:2208.04933.
7. Box, G.; Pierce, D. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *J. Am. Stat. Assoc.* **1970**, *65*, 1509–1526. [[CrossRef](#)]
8. Toda, H.; Phillips, P. Vector autoregression and causality: A theoretical overview and simulation study. *Econom. Rev.* **1994**, *13*, 259–285. [[CrossRef](#)]

9. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [\[CrossRef\]](#)
10. Seo, Y.; Defferrard, M.; Vandergheynst, P.; Bresson, X. Structured sequence modeling with graph convolutional recurrent networks. In Proceedings of the International Conference on Neural Information Processing, Siem Reap, Cambodia, 13–16 December 2018.
11. Zhang, M.; Wu, S.; Yu, X.; Liu, Q.; Wang, L. Dynamic graph neural networks for sequential recommendation. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 4741–4753. [\[CrossRef\]](#)
12. Zambon, D.; Cini, A.; Livi, L.; Alippi, C. Graph state-space models. *arXiv* **2023**, arXiv:2301.01741. [\[CrossRef\]](#)
13. Reitmann, S.; Alam, S.; Schultz, M. Advanced quantification of weather impact on air traffic management. *ATM Semin.* **2019**, *44*, 554–567.
14. Maze, T.; Agarwal, M.; Burchett, G. Whether weather matters to traffic demand, traffic safety, and traffic operations and flow. *Transp. Res. Rec.* **2006**, *1948*, 170–176. [\[CrossRef\]](#)
15. Gneiting, T.; Raftery, A. Weather forecasting with ensemble methods. *Science* **2005**, *310*, 248–249. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Dunn, S.; Wilkinson, S. Increasing the resilience of air traffic networks using a network graph theory approach. *Transp. Res. Part E Logist. Transp. Rev.* **2016**, *90*, 39–50. [\[CrossRef\]](#)
17. Gu, A.; Dao, T.; Ermon, S.; Rudra, A.; Ré, C. Hippo: Recurrent memory with optimal polynomial projections. In Proceedings of the 34th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2020.
18. Dao, T.; Fu, D.; Ermon, S.; Rudra, A.; Ré, C. Flashattention: Fast and memory-efficient exact attention with io-awareness. In Proceedings of the 36th International Conference on Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022.
19. Madhavrao, R.; Moosakhanian, A. Integration of digital weather and air traffic data for NextGen. In Proceedings of the 2018 IEEE/AIAA 37th Digital Avionics Systems Conference, London, UK, 23–27 September 2018.
20. Hawkins, M.; Brown, V.; Ferrell, J. Assessment of NOAA National Weather Service methods to warn for extreme heat events. *Weather. Clim. Soc.* **2017**, *9*, 5–13. [\[CrossRef\]](#)
21. Strohmeier, M.; Olive, X.; Lübke, J.; Schäfer, M.; Lenders, V. Crowdsourced air traffic data from the OpenSky Network 2019–2020. *Earth Syst. Sci. Data* **2021**, *13*, 357–366. [\[CrossRef\]](#)
22. Rossi, E.; Chamberlain, B.; Frasca, F.; Eynard, D.; Monti, F.; Bronstein, M. Temporal graph networks for deep learning on dynamic graphs. *arXiv* **2020**, arXiv:2006.10637. [\[CrossRef\]](#)
23. Kim, T.; Sajjadi, M.; Hirsch, M.; Schölkopf, B. Spatio-temporal transformer network for video restoration. In Proceedings of the 15th European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
24. Steudel, R. Liquid sulfur. In *Elemental Sulfur and Sulfur-Rich Compounds I*; Springer: Berlin/Heidelberg, Germany, 2003.
25. Ye, J.; Xue, S.; Jiang, A. Attention-based spatio-temporal graph convolutional network considering external factors for multi-step traffic flow prediction. *Digit. Commun. Netw.* **2022**, *8*, 343–350. [\[CrossRef\]](#)
26. Cao, D.; Wang, Y.; Duan, J.; Zhang, C.; Zhu, X.; Huang, C.; Tong, Y.; Xu, B.; Bai, J.; Tong, J.; et al. Spectral temporal graph neural network for multivariate time-series forecasting. In Proceedings of the 34th International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2020.
27. Llugsi, R.; Yacoubi, S.E.; Fontaine, A.; Lupera, P. Comparison between Adam, AdaMax and Adam W optimizers to implement a Weather Forecast based on Neural Networks for the Andean city of Quito. In Proceedings of the 2021 IEEE Fifth Ecuador Technical Chapters Meeting, Cuenca, Ecuador, 12–15 October 2021.

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