

# Integrative Mean-Field Epidemic Model and Adaptive Graph Learning for Network-wide Delay Propagation Dynamics Prediction

Chi Li<sup>a,d</sup>, Mingcong Lei<sup>a</sup>, Jingxuan Wu<sup>a</sup>, Yuzhe Yang<sup>a</sup>, Zibin Pan<sup>b</sup>, Xiongwen Qian<sup>d</sup>, Jianfeng Mao<sup>a,c,\*</sup>

<sup>a</sup>School of Data Science, The Chinese University of Hong Kong, Shenzhen, Guangdong, 518172, P.R. China

<sup>b</sup>School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, Guangdong, 518172, P.R. China

<sup>c</sup>Guangdong Provincial Key Laboratory of Big Data Computing, School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, Guangdong, 518172, P.R. China

<sup>d</sup>Shenzhen Research Institute of Big Data, Shenzhen, Guangdong, 518172, P.R. China

---

## Abstract

Flight delay propagation modeling remains a significant challenge in air traffic management. Current modeling approaches generally fall into two categories: data-driven models, which provide high predictive performance but often lack interpretability, and mechanistic models, which are interpretable but struggle to achieve accurate predictions. To address these limitations, we propose the Mean-field Epidemic Graph Adaptive Framework (MEGA) model. MEGA integrates a white-box Mean-field SIS epidemic model (ME-SIS) with an adaptive graph neural network, employing an extended Kalman Filter to enhance model performance. We perform a detailed analysis of the ME-SIS model and evaluate the MEGA framework in both simulated and real-world scenarios, achieving superior predictive performance compared to state-of-the-art baseline models. Additionally, extensive ablation studies and case studies validate the contributions of each module, demonstrating the model's robustness and scalability. Our results provide valuable insights into flight delay propagation modeling and offer implications for improving air traffic management strategies.

**Keywords:** Mean-field approximation, Epidemic SIS model, Adaptive graph learning, Dynamic graph, Flight delay, Delay propagation

---

## 1. Introduction

Flight delays have long posed a significant challenge in air traffic management, leading to considerable disruptions across interconnected flight networks (Rebollo and Balakrishnan, 2014). The Bureau of Transportation Statistics (BTS) reports that approximately 20% of scheduled commercial flights were delayed between 2010 and 2018 (BTS, 2021). Such delays not only affect airline operations but also propagate across the entire network, increasing congestion and complicating air traffic management. Moreover, the Federal Aviation Administration (FAA) estimated that flight delays led to economic losses of approximately \$33 billion in 2019 alone, reflecting the severe financial burden on airlines, passengers, and the economy as a whole (FAA, 2021). Beyond the economic impacts, flight delays negatively affect passenger satisfaction and contribute to increased carbon emissions, as prolonged flight operations consume more fuel and generate additional greenhouse gases (Li et al., 2023, 2024b). These multifaceted impacts make effective delay management a critical priority for the aviation industry.

A core factor contributing to flight delays is the dynamics propagation of delays across the aviation network

---

\*Corresponding author. Email address: lichi@cuhk.edu.cn; mingconglei@link.cuhk.edu.cn; jingxuanwu@link.cuhk.edu.cn; yuzheyang@link.cuhk.edu.cn; zibinpan@link.cuhk.edu.cn; qianxiongwen@sribd.cn; jfmao@cuhk.edu.cn

(Rebollo and Balakrishnan, 2014). Due to the chain-like structure of flight itineraries and the tight interconnectivity of the aviation network, local delays can quickly escalate into widespread network disruptions (Li et al., 2023, 2024a). Additionally, aviation networks have been identified as a major medium for the spread of infectious diseases, such as COVID-19, which further underscores the importance of understanding network dynamics (Li et al., 2024c). Given that aviation networks are often conceptualized as complex networks, employing a complex network perspective to characterize delay propagation mechanisms is both crucial and beneficial. This approach provides valuable insights into the dynamics of delay spread and aids in developing effective strategies for aviation management, ultimately enhancing the resilience and performance of the entire network.

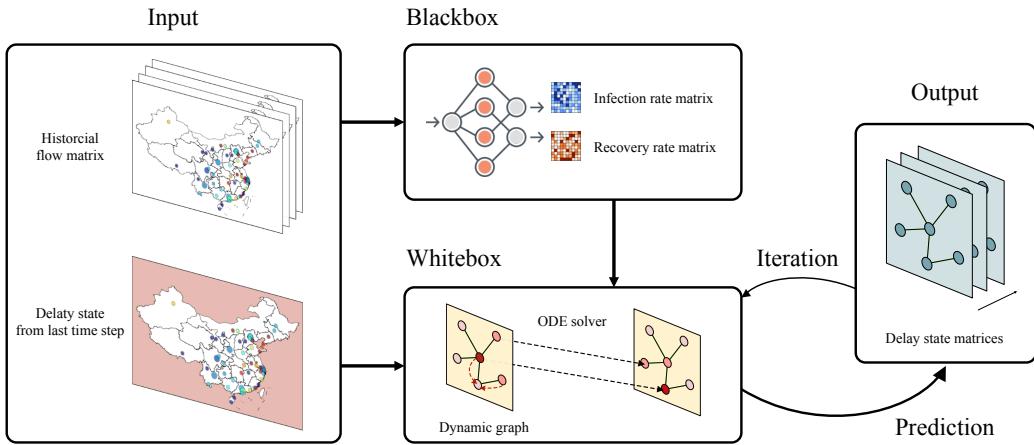
Li et al. (2024a) systematically reviewed current approaches for modeling flight delays from both the flight chain and aviation network perspectives. In particular, network-level modeling approaches frequently involve methods such as epidemic spread models, complex networks, and graph neural networks. Epidemic spreading paradigms constitute mathematical frameworks devised to emulate the propagation of infectious agents within a population. These models classify individuals into discrete states and facilitate transitions between these states based on specific infectious dynamics. Analogously, using epidemic spreading models to analyze flight delay propagation requires establishing parallels between disease transmission dynamics and the spread of delays throughout the aviation network. For instance, Ceria et al. (2021) utilized a heterogeneous SIS spreading process on an airline network to model airport congestion contagion, aiming to replicate airport vulnerability. In this framework, heterogeneous infection and recovery rates are evaluated based on the constructed network. Such models offer a mechanistic and interpretable perspective on delay propagation; however, existing models have yet to accurately replicate the behavior of delay propagation, and their predictive performance still requires improvement.

In recent years, with the rise of Graph Neural Networks (GNNs), a category of deep learning paradigms specifically devised for manipulating graph-structured data, their application has expanded across domains such as transportation, biomedicine, and social networks, owing to their potent representational capabilities for non-Euclidean data and immense potential in information aggregation (Jiang and Luo, 2022). Li et al. (2021) also indicated that graph signal processing can be leveraged to scrutinize flight delay propagation. For example, airport delays can be viewed as node signals in a graph, allowing the identification and quantification of various spatiotemporal patterns using graph spectral analysis. Using GNNs for network-level delay prediction typically requires feature extraction from both spatial and temporal dimensions. The spatial aspect often involves modeling airport relationships using multiple adjacency relations (Wu et al., 2023), while recent advancements have utilized dynamic graph structures to simulate evolving inter-airport relationships (Li et al., 2024b). GNNs have demonstrated effective capabilities in capturing complex inter-airport relationships and achieving state-of-the-art performance. However, due to their black-box nature, these models often fail to provide interpretable insights into the mechanisms underlying delay propagation.

Combining these two approaches—epidemic spreading models and graph neural networks—presents a promising direction for advancing flight delay modeling. This hybrid approach has already been widely adopted in other domains, such as road traffic flow prediction. For example, Yang et al. (2023, 2024) modeled traffic state propagation using a Cell Transmission Model (CTM) under given initial and boundary conditions, while employing LSTMs to estimate the CTM’s initial and boundary conditions, and further utilized an extended Kalman Filter to ensure the consistency of the predictions with traffic flow dynamics. This approach ensures that traffic state propagation

can be accurately predicted while maintaining a high level of interpretability, effectively revealing the underlying propagation patterns. Similarly, integrating epidemic modeling principles with graph neural networks could enhance both the accuracy and interpretability of delay propagation models in aviation networks, providing a more holistic understanding of delay dynamics.

Inspired by these works, we propose the Mean-field Epidemic Graph Adaptive Framework (MEGA) model, which combines the strengths of epidemic spreading models and graph neural networks. Our MEGA model consists of a white-box component, the Mean-field SIS epidemic model (ME-SIS), and a black-box component, an adaptive graph neural network architecture. As illustrated in Fig. 1, the model takes the initial state of the airport network as input and iteratively adapts to predict the future state of the network accurately. This integration allows the MEGA model not only to provide precise predictions of delay propagation but also to reveal the intermediate processes of delay evolution, thereby offering both predictive power and interpretability. By leveraging the complementary strengths of both approaches, MEGA aims to address the limitations of existing models and advance the understanding and management of delay propagation in aviation networks. The main contributions of this paper are as follows:



**Fig. 1.** Demonstration of network-wide delay propagation dynamics prediction using the proposed MEGA framework for airport networks.

- We propose a novel MEGA framework that integrates an adaptive graph neural network with a mean-field SIS epidemic model. This framework effectively captures the relationships between airport delays and accurately predicts future delay propagation trends.
- We improve upon the original epidemic spreading model by incorporating dynamic adjacency matrices and adaptively learning the model parameters, thereby enhancing the accuracy and adaptability of the delay propagation modeling.
- We provide a proposition analysis of the Mean-field SIS model, revealing the steady-state conditions under which delay propagation reaches its peak. This analysis offers valuable insights into the behavior of delay dynamics.
- Our model is evaluated on the Chinese aviation network, demonstrating superior performance compared to state-of-the-art models, with high generalizability across different network configurations.

The remainder of our work is organized as follows. Section 2 presents a detailed literature review, summarizing research on epidemic spreading models in complex networks, flight delay propagation modeling in aviation networks, and the application of deep learning and propagation models in the transportation domain. In Section 3, we outline the preliminaries of the mean-field SIS epidemic model and introduce the problem formulation. Section 4 provides a proposition analysis of the mean-field SIS epidemic model. Section 5 offers an in-depth description of our MEGA framework, elucidating the motivation and functionality of each module. Section 7 presents a comprehensive performance comparison of the proposed model, including an ablation study to evaluate the impact of each module, as well as case study validations. Finally, Section 8 concludes our research and discusses potential future directions.

## 2. Literature Review

### 2.1. Flight delay propagation modeling

Flight delay propagation modeling has long been a focal point in aviation research. To date, numerous methods have been developed to model flight delay propagation, each with distinct characteristics and application scopes. We recommend readers refer to Li et al. (2024a) for an overview of the main approaches, their features, and areas of application. In this section, we primarily focus on two prominent methods for modeling flight delay propagation: epidemic spreading models and graph neural network models.

In the context of epidemic spreading models, airports or flights can be treated as nodes in a network, with the delay propagation acting as the "infection" that spreads among them. To this end, approaches to modeling flight delay propagation via epidemic spreading models can be broadly categorized into two distinct methodologies: airport-based and flight-based methods (Baspinar and Koyuncu, 2016). In their work, the flight-based epidemic model bifurcates flights into susceptible and infected states. Conversely, the airport-based epidemic model characterizes the entire airport as a metapopulation, thus adopting a distinct approach to modeling disease transmission dynamics. Wu et al. (2019) propose a flight-based Susceptible-Infected-Susceptible (FSIS) model to examine the flight delay process within air transportation networks and discern factors that serve to either exacerbate or mitigate the delay propagation. Besides, airport-based models frequently incorporate complex network theory, manifesting in various network configurations as demonstrated by Zhang et al. (2020). Similarly, Dai et al. (2018) formulate a novel congestion propagation model for departure aircraft within a multi-event and multi-stage scheduling framework, grounded in the principles of heterogeneous flight connections and network construction. Li et al. (2020) elucidate the macroscopic manifestation of delay propagation and long-term evolution within airport networks by employing the Susceptible-Infected-Recovered-Susceptible (SIRS) epidemic model. Ceria et al. (2021) utilize a heterogeneous SIS spreading process on an airline network to model airport congestion contagion, with the overarching goal of replicating airport vulnerability. Within this framework, the heterogeneous infection rate and recovery rate are evaluated based on the constructed network. In the realm of parameter estimation, Baspinar and Koyuncu (2016) adopt a data-driven methodology for ascertaining parameter values. This approach entails the computation of transmission rates by leveraging historical flight data and the development of an ODE-solving algorithm to estimate recovery rates. Additionally, there are some econometric methods used to estimate transmission rates, such as the translog model (Wu et al., 2019) and factor analysis (Zhang et al., 2020). Moreover, data-driven approaches may be amalgamated with complex network metrics to facilitate a more precise estimation of network epidemic transmission and recovery rates (Ceria et al., 2021).

In recent years, numerous studies have utilized graph neural networks to predict the time series of flight delays at the airport level, capitalizing on the strong potential of graph neural networks to extract spatial information at this scale (Li et al., 2024a). Zeng et al. (2021) utilized a predefined distance matrix to define adjacency relationships between airports and combined GCN with LSTM to extract spatial and temporal relationships among delay sequences. Bao et al. (2021) built upon this foundation and introduced an LSTM combined with an attention module to alleviate temporal heterogeneities between past data and future predictions and to conduct multi-step-ahead airport delay prediction. Similarly, Cai et al. (2021) proposed a two-stage framework, where the first stage modeled the temporal dynamics of spatial interactions among airports in graph snapshots based on the Markov property, and the second stage captured the hidden patterns in the graph, which could be utilized in emergency cases involving 40 unknown air routes. Sun et al. (2022) introduced a graph attention network to model the delay propagation among the airport network with an updated adjacency matrix to better capture the propagation. Bao et al. (2021) modeled the transit flight process as a graph network, with each stage represented as a node, and predicted departure delays using a model consisting of a GCN layer and a GraphSAGE layer, incorporating sequence information as a temporal feature. Wu et al. (2023) targeted the multiple adjacency relationships and complex coupling relationships within the aviation network delay network, proposing a space-time separable graph neural network model. In a recent study, Li et al. (2024b) considered the dynamic and adaptive nature of the delay network's spatio-temporal relationships and the periodic characteristics of delay sequences, achieving state-of-the-art predictive performance on multiple datasets.

### 2.2. Mean-field epidemic model

### 50 2.3. Deep learning-aided model

## 3. Preliminary

### 3.1. Individual-based mean-field SIS model

The Susceptible-Infected-Susceptible (SIS) model is a classic framework for understanding the spread of infectious diseases within a population. In this model, an infected node can infect its neighbors at an infection rate  $\beta$  and can recover at a recovery rate  $\delta$ . However, once recovered and healthy, the node becomes susceptible to the virus again. The infection and recovery processes are independent of each other. The theory of epidemic spreading through networks can also be applied to spreading email worms and other computer viruses and the propagation of faults in various networks. Naturally, researchers focus on the steady state of spreading problems. Previously, many authors have mentioned the existence of an epidemic threshold  $\tau_c$ . If the effective spreading rate exceeds  $\tau_c$ , the 60 virus persists, and a non-zero proportion of nodes remain infected. The infection decays exponentially for spreading rates below the threshold until it disappears (Chakrabarti et al., 2008).

Researchers often use Markov processes to describe the spreading process. In a network with only two states, the spreading network is defined by  $X_i(t) = 1$ , indicating the infected state, and  $X_i(t) = 0$ , indicating the healthy state. Applying Markov theory directly to this two-state continuous Markov chain, the infinitesimal generator  $Q_i(t)$  is:

$$Q_i(t) = \begin{bmatrix} -q_{1;i} & q_{1;i} \\ q_{2;i} & -q_{2;i} \end{bmatrix} \quad (1)$$

**Table 1**

Notations used in this paper

Parameters	Description
$\beta_{ij}$	Infection rate of node $i$ to node $j$
$\delta_i$	Recovery rate of node $i$
$\tau_i$	Critical effective infection rate of node $i$
$q_{ij}$	Transition rate from state $i$ to state $j$ , representing the rate at which the system transitions between states
$s_i(t)$	Probability that the network is in state $i$ at time $t$
$v_j(t)$	Probability that node $j$ is infected at time $t$
$V(t)$	Probability vector at time $t$ , representing the infection probabilities of all nodes
$w_I$	Infection state vector, indicating the infection status of all nodes
$\Delta$	Diagonal matrix with elements as degree $d_i$
$a_{ij}$	Binary variable equal to 1 if there is a direct connection (e.g., flight) between nodes $i$ and $j$ , and 0 otherwise
$A$	Adjacency matrix of the network
$d_i$	Degree of node $i$
$X_i$	State of node $i$ , where $X_i = 1$ indicates that node $i$ is infected, and $X_i = 0$ indicates that node $i$ is healthy
$Y(t)$	Network state vector at time $t$ , where each entry represents the state of a node, with one node in state 1 (infected) and the rest in state 0 (healthy)
$\lambda_{min}$	Minimum eigenvalue of the matrix, which influences the stability and dynamics of the system

where  $q_{2;i} = \delta$  and  $q_{1;i} = \beta \sum_{j=1}^N a_{ij} 1_{\{X_j(t)=1\}}$ . Here,  $q_{1;i}$  represents the coupling infection rate between node  $i$  and the rest of the network, introducing double randomness to the process, indicating that this process is no longer a pure Markov process:

$$\Pr [X_i(t + \Delta t) = 1 | X_i(t) = 0] = q_{1;i} \Delta t + o(\Delta t) \quad (2)$$

To eliminate the random attribute of  $q_{1;i}$ , we must consider all combinations where adjacent nodes  $X_j(t) = 1$ .

Thus, the number of base states increases dramatically until all possibilities are considered, leading to the Exact  $2^N$ -State Markov Model. Similarly, with  $X_j(t) = 1$  indicating the infected state and  $X_j(t) = 0$  indicating the healthy state, for  $N$  nodes, we use  $Y(t) = [Y_0(t) \ Y_1(t) \ \dots \ Y_{2^N-1}(t)]^T$  to describe all possible states of the network, where:

$$Y_i(t) = \begin{cases} 1, & i = \sum_{k=1}^N X_k(t) 2^{k-1} \\ 0, & i \neq \sum_{k=1}^N X_k(t) 2^{k-1} \end{cases} \quad (3)$$

By establishing the following  $Q$  matrix, the virus infection process can be described as a continuous Markov

chain:

$$q_{ij} = \begin{cases} \delta, & \text{if } i = j + 2^{m-1}; \\ & m = 1, 2, \dots, N; x_m = 1 \\ \beta \sum_{k=1}^N a_{mk} x_k, & \text{if } i = j - 2^{m-1}; \\ & m = 1, 2, \dots, N; x_m = 0 \\ -\sum_{k=1; k \neq j}^N q_{kj}, & \text{if } i = j \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where  $i = \sum_{k=1}^N x_k 2^{k-1}$ . We can then establish a vector  $s(t)$  to describe the probability of each possible state of the network at time  $t$ , where:

$$s_i(t) = \Pr[Y(t) = i] = \Pr[X_1(t) = x_1, X_2(t) = x_2, \dots, X_n(t) = x_n] \quad (5)$$

Letting  $v_j(t) = \Pr[X_j(t) = 1]$ , we have the following equation:  $v^T(t) = s^T(t)M$ , where  $M$  records all possible scenarios for the network. This approach allows us to describe the spreading process and solve for the steady state using differential equations. By analyzing the properties of the Q matrix and applying it to specific topological structures, we can provide the most precise analysis of virus spreading in complex networks. However, due to the exponential growth of state numbers ( $2^N$ ), the state transition matrix  $Q \in \mathbb{R}^{2^N \times 2^N}$  and matrix  $M \in \mathbb{R}^{2^N \times N}$  results in high computational complexity and storage requirements ( $O(2^N)$ ). Van Mieghem et al. (2008) observed that when the same model is precisely described by Markov theory, the Markov chain (with a finite number of states) inherently has absorbing states, contradicting the existence of a threshold. This presents challenges in practical applications for large-scale networks.

To address these issues, we can assume that the infection probability of each node is independent without conditioning. Van Mieghem et al. (2008) proposed the N-Intertwined Markov Chain Model, using global average infection and recovery rates to approximate the system's behavior, which is a mean field approximation:

$$E[q_{1;i}] = E \left[ \beta \sum_{j=1}^N a_{ij} 1_{\{X_j(t)=1\}} \right] = \beta \sum_{j=1}^N a_{ij} \Pr[X_j(t) = 1] \quad (6)$$

Further, we derive the effective infinitesimal generator:

$$\overline{Q_i(t)} = \begin{bmatrix} -E[q_{1;i}] & E[q_{1;i}] \\ \delta & -\delta \end{bmatrix} \quad (7)$$

Applying Markov theory to  $\overline{Q_i(t)}$ , we can describe the state changes of each node with a differential equation over time:

$$\frac{dv_i(t)}{dt} = \beta \sum_{j=1}^N a_{ij} v_j(t) - v_i(t) \left( \beta \sum_{j=1}^N a_{ij} v_j(t) + \delta \right) \quad (8)$$

The state changes of the entire network can be written in vector form:

$$\frac{dV(t)}{dt} = \beta A V(t) - \text{diag}(v_i(t)) (\beta A V(t) + \delta u) \quad (9)$$

This form significantly reduces computational complexity and provides an efficient way to determine the epidemic threshold. Ignoring the nonlinear part, we obtain  $\frac{dV(t)}{dt} = (\beta A - \delta I)V(t)$ . This equation verifies that the threshold proposed by

### 3.2. Problem formulation

In studying epidemic spreading models, the classical SIS (Susceptible-Infected-Susceptible) model assumes that the infection rate  $\beta$  and the recovery rate  $\delta$  are identical for all nodes. Although this assumption simplifies computations, it is overly idealistic for practical applications, especially in complex networks such as airport networks.  
Our research aims to enhance the accuracy and applicability of the SIS model by introducing node-specific infection and recovery rates.

In real-world networks like airport networks, the characteristics of each node (airport) differ significantly. For example, airports vary in terms of flight traffic, causes of delays, recovery capabilities, etc. These differences result in notable variations in the infection rates (flight delay propagation rates) and recovery rates (delay recovery rates) across airports. Using uniform infection and recovery rates fails to accurately reflect these realities, thereby limiting the predictive power and applicability of the model. Heterogeneous networks, which account for diverse node and link types, provide a more realistic framework by allowing for node-specific behaviors. As defined by Shi et al. (2016), Heterogeneous networks capture the complexity of real systems through multiple node and relationship types, making them ideal for modeling such heterogeneous dynamics. By introducing node-specific infection and recovery rates, we can more precisely describe the dynamic behavior of each node and enhance the model's adaptability to various scenarios.  
110

Our project utilizes real airport data to study the propagation of flight delays. This data provides specific information about each airport, allowing us to adjust the parameters for each node based on actual conditions, thereby improving the realism and accuracy of the model. Data-driven models better capture and reflect the complexities of the real world, offering more precise predictions and analyses.

To achieve these goals, we start with the classical SIS model and progressively introduce node-specific infection and recovery rates. In the classical SIS model, the infection rate  $\beta$  and recovery rate  $\delta$  are the same for all nodes. While this model is computationally simple, it fails to capture the differences between nodes. To improve the realism of the model, we introduce node-specific infection rates  $\beta_i$  and recovery rates  $\delta_i$ , with the state transition equation  
120

$$\frac{dV(t)}{dt} = \text{Adiag}(\beta_i)V(t) - \text{diag}(v_i(t))(\text{Adiag}(\beta_i)V(t) + \Delta) \quad (10)$$

where  $A$  is the adjacent matrix.

This improvement allows the model to adjust parameters based on the specific conditions of each node, thereby more accurately describing and predicting the dynamic behavior of the system. The transition from the classical SIS model to a node-specific SIS model enhances its accuracy and applicability and improves its predictive capability in practical applications.

We employ the AdapGL neural network structure to learn the infection rate matrix and recovery vector at time  $t$ , based on a moving time window to capture dynamic changes. We then predict future delay rates using a difference equation approximation derived from the SIS model (Van Mieghem et al., 2008). Notably, this model approximates the spreading threshold using mean-field theory. The use of the spectral radius-based threshold helps explain the

130 steady-state behavior and evolution of virus-spreading:

$$\vec{v}^{(t+1)} = (\mathbf{1} - \vec{\delta}^{(t)}) \otimes \vec{v}^{(t)} + (\mathbf{1} - \vec{v}^{(t)}) \otimes \mathbf{A}^{(t)} \vec{v}^{(t)} \quad (11)$$

This combined black-box and white-box approach allows us to leverage real airport data better, improving our understanding of flight delay propagation patterns and providing theoretical support for developing more effective delay management strategies.

#### 4. Model stability and epidemic threshold analysis

Stability has always been essential in studying epidemic spread within complex networks. Many previous researchers have dedicated themselves to studying various properties of stability. Still, their research has remained at the stage where the SIS model's infection and recovery rates are considered constants. No one has conducted an in-depth study of the properties at the node-specific level. We have made some breakthroughs in this regard, where the infection and recovery rates will appear as vectors in the derivations in this section.

140 This section aims to analyze, through mathematical derivation, the stability of the SIS model and the epidemic threshold under node-specific infection rates and recovery rates. This analysis helps better understand the behavior of epidemic spread in complex networks and provides theoretical support for epidemic prevention and control strategies in practical applications. We will start with the expectation of the number of infected nodes, analyze the range of critical values for the nodes, and finally prove the convergence of the system.

**Proposition 1.** Denoting the vector  $w_I = (1_{\{X_1=1\}}, 1_{\{X_2=1\}}, \dots, 1_{\{X_N=1\}})$ , the average number of infected nodes of the network satisfies:

$$\frac{d}{dt} E \left[ \sum_{j=1}^N 1_{\{X_j=1\}} \right] = \frac{1}{N} E[w_I^T ((\Delta - A) - \text{diag}(\tau_i^{-1})) \text{diag}(\beta_i) w_I] \quad (12)$$

*Proof.* The expectation of the number of infected nodes is computed by summing up all nodes and differentiating them with respect to time  $t$ . Applying the mean-field approximation for this individual-based SIS model, we obtain:

$$\frac{d}{dt} E \left[ \sum_{j=1}^N 1_{\{X_j=1\}} \right] = E \left[ \sum_{k=1}^N 1_{\{X_k=1\}} \beta_k \sum_{j=1}^N a_{kj} 1_{\{X_j=0\}} - \delta_k \sum_{j=1}^N 1_{\{X_j=1\}} \right]. \quad (13)$$

Here,  $\beta_k$  and  $\delta_k$  are the infection and recovery rates, respectively, which are specific to each node. If we denote the 150 first part of the right-hand side of the equation as  $S$ , then for  $S$  and break down the transmission term, we derive:

$$\begin{aligned} S &= \sum_{k=1}^N 1_{\{X_k=1\}} \beta_k \sum_{j=1}^N a_{kj} 1_{\{X_j=0\}} \\ &= \sum_{k=1}^N 1_{\{X_k=1\}} \beta_k \left( \sum_{j=1}^N a_{kj} - \sum_{j=1}^N a_{kj} 1_{\{X_j=1\}} \right) \\ &= \sum_{k=1}^N d_k \beta_k 1_{\{X_k=1\}} - \sum_{k=1}^N \sum_{j=1}^N a_{kj} \beta_k 1_{\{X_k=1\}} 1_{\{X_j=1\}} \\ &= w_I^T \text{diag}(\beta_i) D - w_I^T A \text{diag}(\beta_i) w_I \\ &= D^T \beta_I w_I - w_I^T A \beta_I w_I \\ &= D^T \beta_I w_I - w_I^T \Delta \beta_I w_I + w_I^T \Delta \beta_I w_I - w_I^T A \beta_I w_I \\ &= (u^T - w_I^T) \Delta \beta_I w_I + w_I^T (\Delta - A) \beta_I w_I, \end{aligned} \quad (14)$$

where  $D^T = u^T \Delta$ ,  $\Delta = \text{diag}(d_i)$ ,  $\delta_I = (\delta_1, \delta_2, \dots, \delta_N)$ ,  $\beta_I = (\beta_1, \beta_2, \dots, \beta_N)$ ,  $w_I = (1_{\{X_1=1\}}, 1_{\{X_2=1\}}, \dots, 1_{\{X_N=1\}})$ . Since the indicator function has the property  $1_{\{X_j=1\}} = 1_{\{X_j=1\}} 1_{\{X_j=1\}}$ , we have:

$$\begin{aligned} (u - w_I)^T \Delta \beta_I w_I &= \sum_{j=1}^N (1 - 1_{\{X_j=1\}}) d_j \beta_j 1_{\{X_j=1\}} \\ &= \sum_{j=1}^N (1_{\{X_j=1\}} - 1_{\{X_j=1\}} 1_{\{X_j=1\}}) d_j \beta_j = 0. \end{aligned} \quad (15)$$

Substituting this part equals zero into the equation, we can get the final form:

$$\begin{aligned} \frac{d}{dt} E \left[ \sum_{j=1}^N 1_{\{X_j=1\}} \right] &= E \left[ \sum_{k=1}^N 1_{\{X_k=1\}} \beta_k \sum_{j=1}^N a_{kj} 1_{\{X_j=0\}} - \delta \sum_{j=1}^N 1_{\{X_j=1\}} \right] \\ &= w_I^T (\Delta - A) \beta_I w_I - \delta_I^T w_I \\ &= w_I^T (\Delta - A) \beta_I w_I - w_I^T \text{diag}(\delta_i) w_I \\ &= \frac{1}{N} E[w_I^T ((\Delta - A) - \text{diag}(\tau_i^{-1})) \text{diag}(\beta_i) w_I]. \end{aligned} \quad (16)$$

□

The mathematical formulation underscores the complex interplay between network topology and node-specific epidemiological characteristics. Our model captures nuances that homogeneous models overlook by considering heterogeneous infection and recovery rates. The derived differential equation highlights how network structure and individual differences in disease susceptibility and recovery impact the overall disease dynamics. This work thus contributes to a deeper understanding of epidemic spread in complex networks, providing a foundation for designing targeted interventions based on both network topology and individual node characteristics.

**Proposition 2.** The range that each node's effective infection rate  $\tau$  must satisfy to ensure the stability of the entire network is elaborated below:

$$\begin{aligned} \tau_{\max}^{-1} &\leq \Pr[X_k = 0] \cdot d_{\max} \\ \tau_{\min}^{-1} &\geq \min_{a_{ij}=1} \Pr[X_j = 0 \mid X_i = 1] d_{\min} \end{aligned} \quad (17)$$

*Proof.* When the system has reached equilibrium, there must be points where the derivative equals zero:

$$E[w_I^T (\Delta - A) w_I] = E[w_I^T \text{diag}(\tau^{-1}) w_I], \quad (18)$$

where  $\Delta - A$  is the Laplacian matrix. According to the properties of the Laplacian matrix, we have:

$$w_I^T (\Delta - A) w_I = \sum_{i,j: a_{ij}=1} (1_{\{X_i=1\}} - 1_{\{X_j=1\}})^2. \quad (19)$$

Since both sides are diagonal matrices, when calculating the equilibrium (i.e., when the first derivative is zero), we can simultaneously disregard  $\text{diag}(\beta_i)$ . Omitting  $\text{diag}(\beta_i)$  from the first part of the first conclusion, we can simplify

it as follows:

$$\begin{aligned}
E[w_I^T(\Delta - A)w_I] &= \sum_{i,j:a_{ij}=1} E[(1_{\{X_i=1\}} - 1_{\{X_j=1\}})^2] \\
&= 2 \sum_{i,j:a_{ij}=1} E[1_{\{X_i=1\}} - 1_{\{X_i=1\}}1_{\{X_j=1\}}] \\
&= 2 \sum_{i,j:a_{ij}=1} E[1_{\{X_i=1\}}(1 - 1_{\{X_j=1\}})] \\
&= 2 \sum_{i,j:a_{ij}=1} E[1_{\{X_i=1\}}1_{\{X_j=0\}}] \\
&= 2 \sum_{i,j:a_{ij}=1} \Pr[X_i = 1, X_j = o] \\
&= 2 \sum_{i,j:a_{ij}=1} \Pr[X_i = 1] \Pr[X_j = 0 | X_i = 1] \\
&= \sum_{i=1}^N \Pr[X_i = 1] \sum_{j=1}^N a_{ij} \Pr[X_j = 0 | X_i = 1].
\end{aligned} \tag{20}$$

For the second part, we can handle it as follows:

$$\begin{aligned}
E[w_I^T(\text{diag}(\tau_i^{-1}))w_I] &= E\left[\sum_{i=1}^N 1_{\{X_j=1\}}1_{\{X_j=1\}}\tau_i^{-1}\right] \\
&= E\left[\sum_{i=1}^N 1_{\{X_j=1\}}\tau_i^{-1}\right] \\
&= \sum_{i=1}^N \tau_i^{-1} \Pr[X_i = 1].
\end{aligned} \tag{21}$$

From this equation, we can easily obtain the following range for  $\tau$ :

$$\tau_{\max}^{-1} \sum_{i=1}^N \Pr[X_i = 1] \leq E[w_I^T(\text{diag}(\tau_i^{-1}))w_I] = \sum_{i=1}^N \tau_i^{-1} \Pr[X_i = 1] \leq \tau_{\min}^{-1} \sum_{i=1}^N \Pr[X_i = 1]. \tag{22}$$

<sup>170</sup> For the maximum effective infection rate, we have:

$$\begin{aligned}
\tau_{\max}^{-1} &\leq \frac{\sum_{i=1}^N \Pr[X_i = 1] \sum_{j=1}^N a_{ij} \Pr[X_j = 0 | X_i = 1]}{\sum_{i=1}^N \Pr[X_i = 1]} \\
&\leq \max_{k,l:a_{kl}=1} \Pr[X_k = 0 | X_l = 1] \cdot d_{\max} \\
&\leq \Pr[X_k = 0] \cdot d_{\max}.
\end{aligned} \tag{23}$$

Eq. 23 indicates that the most central airport's expected number of non-infected neighboring airports is greater than the inverse of the maximum effective infection rate. For densely connected airports, the tolerance for effective infection rates is lower, and exceeding a certain value will prevent the network from converging. For the minimum effective infection rate, we have:

$$\begin{aligned}
\tau_{\min}^{-1} &\geq \min \sum_{j=1}^N a_{ij} \Pr[X_j = 0 | X_i = 1] \\
&\geq \min \left( \min_{i,j:a_{ij}=1} \Pr[X_j = 0 | X_i = 1] d_i \right) \\
&= \min_{a_{ij}=1} \Pr[X_j = 0 | X_i = 1] d_{\min}.
\end{aligned} \tag{24}$$

Eq. 24 indicates that the minimum effective infection rate must be below a certain threshold to prevent the epidemic from spreading indefinitely across the network. If the effective infection rate exceeds this threshold, certain remote airports will fail to converge to a steady state. This further suggests that nodes with a smaller degree have a higher upper limit for the minimum effective infection rate, indicating that central airports in the network are more susceptible to infections.  $\square$

<sup>180</sup> *The derivation provides critical insights into how the network's structure and individual node properties affect the stability and spread of delays (or infections) in an aviation network. The results indicate that nodes with higher connectivity have stricter requirements for their effective infection rates to ensure network stability. This understanding is essential for designing targeted control strategies to mitigate the spread of delays and maintain the overall stability of the network. By establishing the range for effective infection rates, the analysis offers a valuable framework for policymakers and network managers to prioritize interventions and resources effectively.*

**Proposition 3.** *The convergence condition for the SIS model is extended to the node-specific context, demonstrating that the system will converge if for each node  $\tau_i = \frac{\beta_i}{\delta_i} \leq \lambda_{\min}$ . Here,  $\lambda_{\min}$  represents the smallest eigenvalue of the matrix  $\text{diag}(\beta_i)A - \text{diag}(\delta_i)$ , and the convergence time is  $\mathcal{O}(n)$ .*

*Proof.* We start by considering the continuous-time Markov process  $\{Z_i\}_{i \in V}$ , with value

$$\begin{aligned} Z_i : k \rightarrow k+1 & \quad \text{at rate } \beta \sum_{(i,j) \in E} Z_j, \\ Z_i : k \rightarrow k-1 & \quad \text{at rate } Z_i. \end{aligned} \tag{25}$$

<sup>190</sup> The coupling argument shows that  $Y(t)$  stochastically dominates  $X(t)$ , which helps in bounding the probability that the system is not in the all-healthy state. This implies that:

$$\mathbf{P}\left(\sum_i X_i(t) = 0\right) \geq \mathbf{P}\left(\sum_i Z_i(t) = 0\right). \tag{26}$$

Moreover, it holds that:

$$\mathbf{P}\left(\sum_i Z_i(t) > 0\right) \leq \sum_i \mathbf{E}(Z_i(t)). \tag{27}$$

However, the transition rates for process  $Y$  are such that:

$$\frac{d}{dt} \mathbf{E}(Z(t)) = (\text{diag}(\beta_i)A - \text{diag}(\delta_i))\mathbf{E}(Z(t)). \tag{28}$$

Since this is a classical ODE equation, we can write its solution directly:

$$\mathbf{E}(Z(t)) = \exp(tM) Z(0), \tag{29}$$

where we denote  $\text{diag}(\beta_i)A - \text{diag}(\delta_i)$  as  $M$ . Note that  $\text{diag}(\beta_i)$ ,  $A$  and  $\delta_i$  are all symmetric matrices, then  $M$  is also a symmetric matrix. And if  $\tau_i = \frac{\beta_i}{\delta_i} \leq \lambda_{\min}$  holds, we can get:

$$\|\mathbf{E}(Z(t))\|_2 \leq e^{-\lambda_{\min}} \|\mathbf{E}(Z(0))\|_2. \tag{30}$$

Applying the Cauchy-Schwarz inequality we have that:

$$\sum_{i \in V} \mathbf{E}(Z_i(t)) \leq \|\mathbf{E}(Z(t))\|_2 \|\mathbf{1}\|_2. \tag{31}$$

By combining several inequalities, we derive a new inequality:

$$\mathbf{P}(X(t) \neq 0) \leq \sum_i \mathbf{E}(Z_i(t)) \leq \|\mathbf{E}(Z(t))\|_2 \|\mathbf{1}\|_2 \leq \sqrt{n} e^{-\lambda_{\min}} \sqrt{\sum_{i=1}^n Z_i(0)^2}. \quad (32)$$

By integrating this equation and applying a piecewise bounding method, we obtain an upper limit for the expected convergence time. This demonstrates that, even in a node-specific network, the convergence time is finite:

$$\begin{aligned} E(\tau) &= \int_0^\infty P(T > t) dt = \int_0^\infty P(X(t) \neq 0) dt \\ &= \int_0^{\frac{\log n}{\lambda_{\min}}} P(X(t) \neq 0) dt + \int_{\frac{\log n}{\lambda_{\min}}}^{+\infty} P(X(t) \neq 0) dt \\ &\leq \int_0^{\frac{\log n}{\lambda_{\min}}} 1 \cdot dt + \int_{\frac{\log n}{\lambda_{\min}}}^\infty n \cdot e^{(-\lambda_{\min})t} dt \\ &= \frac{\log n}{\lambda_{\min}} + n \times \frac{1}{n} = \frac{\log n}{\lambda_{\min}} + 1 = O(\log n). \end{aligned} \quad (33)$$

□

This proof extends Ganesh et al. (2005) convergence condition for the SIS model to node-specific effective infection rates. By demonstrating that the system will converge when  $\tau_i \leq \lambda_{\min}$ , where  $\lambda_{\min}$  is the smallest eigenvalue of  $\text{diag}(\beta_i)A - \text{diag}(\delta_i)$ , we confirm that even with heterogeneous infection and recovery rates, the network can achieve stability. This result is crucial for understanding how individual node properties impact the overall network dynamics and provides a basis for designing effective intervention strategies to control the spread of infections in complex networks.

## 5. Methodology

This section introduces three key components of the MEGA architecture: ME-SIS, the adaptive graph learning module, and EKF. In Section 5.1, we propose the ME-SIS model, which is better suited for flight delay prediction compared to previous models. In Section 5.2, we present an iterative training framework based on GAT for graph neural networks. By combining these two modules, we introduce a framework for airport delay propagation prediction that leverages the high accuracy of graph neural networks and the interpretability of propagation models. In Section 5.3, we propose a method that utilizes the EKF to filter input values, thereby enhancing the model's performance.

### 5.1. Network-wide delay propagation modeling

This section focuses on the ME-SIS model. In Section 5.1.1, we review the disease propagation modeling approaches from previous work. Based on an analysis of the methods proposed in earlier studies, we introduce the ME-SIS model in Section 5.1.2, which is better suited for capturing the delay propagation characteristics of airport networks.

#### 5.1.1. Previous approach

Heterogeneous SIS models (Ceria et al., 2021; Baspinar et al., 2021) have been proposed in previous studies and are described using the following ordinary differential equation (ODE):

$$\frac{dv_i(t)}{dt} = -\delta_i v_i(t) + (1 - v_i(t)) \sum_{j=1}^N \beta_{ij} v_j(t) \quad (34)$$

Here,  $\delta_i$  represents the recovery rate of node  $i$  per unit time  $t$ , and  $v_i(t)$  denotes the infection probability of node  $i$  at time  $t$ .  $\beta_{ij}$  is the probability that node  $i$  infects node  $j$ .

Ceria et al. (2021) computes  $\beta_{ij}$  and  $\delta_i$  using the following expressions :

$$w_{ij} = \frac{w_{ij}^*}{\max_{k,l} w_{kl}^*} \quad (35)$$

$$\beta_{ij} = \beta w_{ij}$$

$$s_i = \sum_j w_{ij} \quad (36)$$

$$\delta_i = \delta \left( c + \left( \frac{s_i}{s_{max}} \right)^\theta \right)$$

Here,  $w_{ij}^*$  is the initial weight in the contagion network, and  $\max_{k,l} w_{kl}^*$  calculates the maximum weight across the network.  $w_{ij}$  represents the normalized network weight.  $\beta$  is a constant that modulates the normalized weights.

Similarly, Baspinar et al. (2021) employs a method akin to that of Ceria et al. (2021) for calculating  $\beta_{ij}$  and  $\delta_i$ , where  $\delta_i$  is derived from historical data using the Euler method for solving differential equations.  $\beta_{ij}$  is derived using a normalized flow matrix.

The methodologies discussed for modeling recovery and infection rates using deterministic functions might not accurately capture the complex interactions among variables due to several limitations. Firstly, employing static functions to model recovery rates or relying exclusively on historical data for solving differential equations may fail to address the dynamic dependencies between variables. Secondly, using normalized flow matrices to represent infection rates for each flight route may not adequately mirror the conditions encountered in real-world scenarios.

### 5.1.2. Individual-based Mean-field SIS approach

Previous models, which employ deterministic functions to model recovery and infection rates, may not adequately reflect reality for several reasons. Firstly, using fixed functions to model recovery rates or merely deriving differential equations from historical data may fail to capture the intricate relationships between variables. Secondly, applying normalized flow matrices as infection rates for individual flight routes may not accurately represent actual conditions. To address these issues, this paper introduces a Mean-field Susceptible-Infected-Susceptible epidemic (ME-SIS) model to simulate the spread of airport network delays. This model categorizes flight delays into two types: those caused by the airport itself and those resulting from delays at other airports. Over time, the probability of flight delays accumulates, and each airport has a specific capacity to handle these delays. We propose that the model for inter-airport delay propagation comprises three factors. Since both the delays caused by the airport itself and its capacity to manage delays are intrinsic attributes of the airport and are in competition, we model the outcome of this competition with a single recovery rate. This parameter, encompassing both elements, allows our model's recovery rate to span all real numbers. We model the propagation of delays between airports using the infection rate between them, asserting that the impact of flights on different routes in their corresponding airports varies; therefore, our model will not adopt a straightforward normalization of flight traffic as the infection rate. This

approach assumes a uniform influence of each flight route on the airports, which is unrealistic. Our model weights the flight traffic differently based on the road.

Building on the above analysis, this paper proposes ME-SIS model. The model treats the entire system as a graph and employs the Individual-based Mean-field SIS disease propagation model to model the top 50 airports in China as a cohesive system. This model conceptualizes the entire system as a dynamic weighted directed graph  $\mathcal{G}^t = (V, E^t, f_w^t, f_\theta^t)$ , where  $V$  is the set of nodes representing each airport. The airports are indexed in descending order of size, denoted as airport  $i (i \in \{1, 2, \dots, 50\})$ , and assumed to be static.  $E^t$ , the edge set, is time-variable, weighted, and directed. If, within any unit of time  $t$ , there is a flight from airport  $i$  to  $j$ , this is denoted as  $e_{ij}^t \in E^t$ .  
260  $f_w^t : E^t \rightarrow [0, 1]$  is a time-varying function mapping any edge  $e_{ij}^t$  in  $E^t$  to its corresponding weight  $w_{ij}^t (w_{ij}^t \in [0, 1])$ . Each edge  $e_{ij}$  corresponds to an infection rate  $\beta_{ij}^t$  computed as:

$$\beta_{ij}^t = \beta w_{ij}^t \quad (37)$$

Where  $w_{ij}^t$  represents the edge weight between airports  $i$  and  $j$  within time  $t$ , and  $\beta (\beta \in (0, 1])$  is a constant, representing the infection rate for that flight route.  $f_\theta^t : V \rightarrow \mathbb{R}^2$  is a time-varying function defined as a two-dimensional real vector  $(v_i^t, \delta_i^t)$ , where  $v_i^t (v_i^t \in [0, 1])$  represents the delay rate of airport  $i$  within time  $t$ , computed as:

$$v_i^t = \frac{D_i^t}{T_i^t} \quad (38)$$

Where  $T_i^t$  is the total number of flights taking off or landing at the airport  $i$  within time  $t$ , and  $D_i^t$  is the number of these flights recorded as delayed.

This paper discreteness the continuous SIS disease propagation model's ordinary differential equations using the explicit Euler method, as shown below:

$$\mathbf{v}^{t+1} = (1 - \delta^t) \otimes \mathbf{v}^t + (1 - \mathbf{v}^t) \otimes \mathbf{B}^t \mathbf{v}^t \quad (39)$$

Here,  $\mathbf{v}^t$  represents the delay rate vector, an  $n$ -dimensional vector where  $v_i^t$  indicates the delay rate of airport  $i$  within time  $t$ .  $\delta^t$  is the recovery rate vector, also  $n$ -dimensional, where  $\delta_i^t$  represents the recovery rate of airport  $i$  within time  $t$ .  $\mathbf{B}^t$  is the infection rate matrix, an  $n$ -dimensional square matrix, where  $\beta_{ij}^t$  represents the infection rate between airports  $i$  and  $j$  within time  $t$ .  
270

It is evident that only two parameters need to be determined in the ME-SIS model: the infection rate matrix  $\mathbf{B}^t$  and the recovery rate vector  $\delta^t$ . Given that both parameters are heterogeneous and time-varying in our modeling, we advocate for developing a graph neural network-based deep learning algorithm to predict the system-wide infection and recovery rates at each unit of time.

## 5.2. Integrative adaptive graph learning

This section introduces the use of the GAT and the Adaptive Graph Learning method to enhance the performance of the ME-SIS model by integrating deep learning-based models with the ME-SIS framework. In Section 5.2.1, we discuss the structure of the Multi-Head GAT model. In Section 5.2.2, we combine the DDP and GAE neural networks, both based on Multi-Head GAT, with the ME-SIS model for airport network delay propagation prediction. In Section 5.2.3, we propose an iterative training framework for the Adaptive Graph Learning module.  
280

### 5.2.1. GAT with multi-head

Graph Attention Networks (GAT) (Veličković et al., 2017) graph introduces a convolutional neural network architecture for graphs based on an attention mechanism. This allows the model to adaptively assign different attention weights to other nodes, thereby learning a weighted combination of node features. Unlike spectral-based methods, GAT operates in the spatial domain, enabling it to handle time-varying weighted directed graphs more effectively than Graph Convolutional Networks (GCN). The input to GAT is an adjacency matrix of dimensions  $\mathbb{R}^{N \times N}$  and a graph signal vector of dimensions  $\mathbb{R}^{N \times F}$ . The output is a node feature matrix of dimensions  $\mathbb{R}^{N \times F'}$ , where  $N$  represents the total number of nodes,  $F$  the feature dimensions per node, and  $F'$  the signal dimensions per node. The feature vector for node  $i$  is denoted as  $h_i$ . The attention mechanism of GAT is described as follows:

For a node  $i$  and its neighboring node  $j$ , the attention coefficient  $e_{ij}$  is computed using the formula:

$$e_{ij} = \text{LeakyReLU}(\mathbf{a}^\top [\mathbf{W}h_i \| \mathbf{W}h_j]) \quad (40)$$

Here,  $\mathbf{W}$  is a weight matrix of dimensions  $\mathbb{R}^{F' \times F}$  that maps the input signals into a hidden space, and  $\mathbf{a}$  is a parameter vector of the attention mechanism with dimensions  $\mathbb{R}^{2F' \times 1}$ . The operator  $\|$  denotes concatenation, and LeakyReLU is an activation function.

Similar to the Transformer model, GAT employs a softmax function to normalize the computed  $e_{ij}$  for all  $j$ :

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})} \quad (41)$$

where  $\mathcal{N}_i$  is the set of neighboring nodes of node  $i$ .

The new feature vector  $h'_i$  for node  $i$  is the weighted average of the feature vectors of all its neighbors, with weights determined by  $\alpha_{ij}$ :

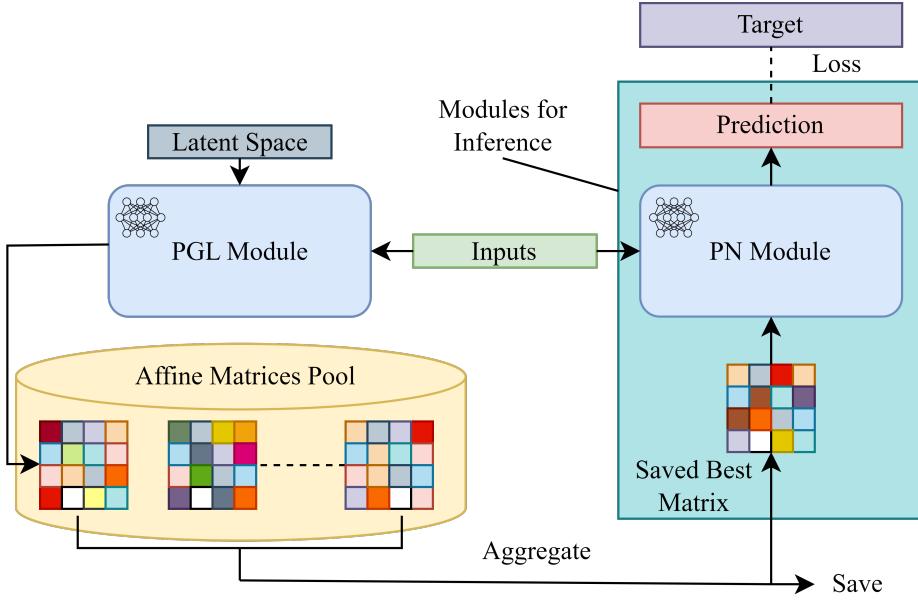
$$h'_i = \sigma \left( \sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}h_j \right) \quad (42)$$

where  $\sigma$  represents a nonlinear activation function.

GAT often employs a multi-head attention mechanism to enhance the model's capability. In this approach, each node has multiple sets of independent  $\mathbf{W}$  and  $\mathbf{a}$ , each producing an output vector. These are then aggregated into a single vector using an arithmetic mean, which combines the multiple attention vectors.

### 5.2.2. Advanced Graph Learning Model

The adaptive graph learning model architecture of MEGA is derived from the AdapGL framework (Zhang et al., 2022) proposed for addressing road traffic flow prediction problems, particularly with the PeMS dataset. The architecture primarily comprises two neural networks and a set of matrices: the Prediction Network (PN), the Parameterized Graph Learning module (PGL), and the Affinity Matrix Set. The PGL is responsible for re-weighting a fixed-topology road adjacency matrix and incorporating it into the Affinity Matrix Set. The PN aggregates the matrices from the Affinity Matrix Set to predict traffic flow at various nodes. The Affinity Matrix Set stores several affinity matrices defined manually or output by the PGL. During training, AdapGL employs a training method akin to the EM algorithm, whereby the PN is fixed while the PGL is trained, and vice versa. The matrix aggregated from the Affinity Matrix Set is then preserved as a fixed matrix for inference. For predictions, given the static graph



**Fig. 2.** Basic model for AdapGL structure

structure, AdapGL exclusively utilizes the PN, which takes the stored adjacency matrix and dynamic input features for predictions (see 2).

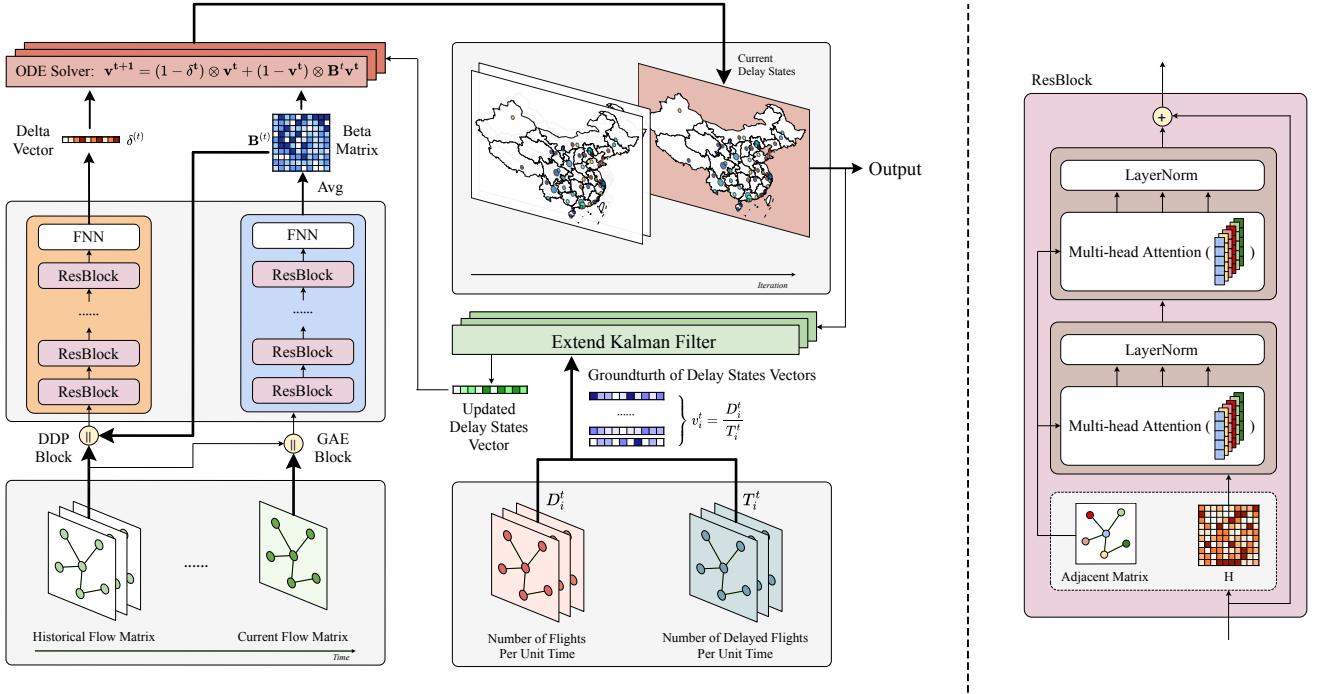
However, the AdapGL architecture faces specific challenges in flight delay prediction tasks. Firstly, the fixed matrix approach of AdapGL could be more suitable since the graph in flight delay predictions is dynamic, weighted, and directed, unlike the static topology in road traffic flow predictions. Secondly, to align with the white-box disease propagation model, our black-box model needs to learn dynamic infection rate matrices  $\hat{\mathbf{B}}^t$  and recovery rate vectors  $\delta^t$ , predicting both dynamic matrices and vectors. To address these issues, we introduce the Adaptive Graph Learning architecture to MEGA (see Fig. 3).  
320

Similar to AdapGL, The adaptive graph learning module in MEGA consists of a Delay Dynamics Prediction Module (DDP) and a Graph-Augmented Epidemic Learning Module (GAE). To accommodate the dynamic, weighted, and directed nature of aviation networks, we utilize a multi-head GAT, a spatial graph convolution layer, as the foundational block. Paired multi-head GAT layers, connected via residual links, form the basic modules of DDP and GAE: the Residual Blocks. DDP and GAE comprise multiple Residual Blocks (see Fig. 3), culminating in a Feed-Forward Neural Network (FNN) layer that adjusts the output signal dimensions to the required levels. Both models participate in the model inference, and due to the dynamic nature of the graph, the Affinity Matrix Set module from AdapGL is omitted in the adaptive graph learning module.  
330

Within the adaptive graph learning module of MEGA, the task of GAE is to dynamically predict the infection rate matrix  $\hat{\mathbf{B}}^t$  for each unit of time. The input to the GAT within the GAE is the graph structure  $\mathbf{B}^t$ , and the signal input comprises a sliding window of length  $n$  for each node  $i$ :

$$x_i = \text{concat}(\mathbf{B}_i^{t-n}, \mathbf{B}_i^{t-n+1}, \dots, \mathbf{B}_i^t) \quad (43)$$

Here,  $\mathbf{B}_i^t$  represents the infection rate matrix's  $i^{th}$  row at time  $t$ . The output of the GAE is a 51-dimensional matrix representing the reweighted infection rates, denoted  $\hat{\mathbf{B}}^t$ . This model is responsible for weighing the infection capabilities of each flight route during each unit of time.



**Fig. 3.** MEGA is a framework consisting of four parts, DDP, GAE, ODE Solver, and EKF. Where GAE is used to re-weight the infection matrix Beta, DDP is used to predict the recovery state vector Delta ODE solver is used to solve the infection propagate process and predict the delay state based on the last state and DDR, GAE’s outcome. EKF filter is used to make the prediction more stable.

The DDP (Vector Learning Network) dynamically predicts the recovery rate vector  $\hat{\delta}^t$  for each time unit. The input structure for the DDP is similar to that of the GAE:

$$x_i = \text{concat}(\mathbf{B}_i^{t-n}, \mathbf{B}_i^{t-n+1}, \dots, \mathbf{B}_i^{t-1}, \hat{\mathbf{B}}_i^t) \quad (44)$$

The output,  $\hat{\delta}_i^t$ , represents the vector of recovery rates, evaluating each airport’s ability to handle delays at time  $t$ .

340 The output of the Adaptive Graph Learning module consists of both GAE and DDP outputs: the infection rate matrix  $\hat{\mathbf{B}}^t$  and the recovery rate vector  $\hat{\delta}^t$ . Leveraging the outputs of the Adaptive Graph Learning module, the ME-SIS model can predict the delay rate  $\hat{\mathbf{v}}^t$  at the current time to estimate the delay rate  $\hat{\mathbf{v}}^{t+1}$  at the next time step.

### 5.2.3. Training Method

To train the DDP and GAE neural networks effectively, considering the unique integration with the ME-SIS framework, we propose an iterative training approach. This method alternates between fixing one neural network while updating the other during each training iteration. Since the outputs of DDP and GAE serve as two critical parameters of the ME-SIS model, they jointly contribute to predicting the delay probability of airports within the aviation network during a given time unit. Consequently, only one neural network—either DDP or GAE—is active in the computational graph during each backpropagation step. This approach ensures better control over the training direction of each neural network. Our method aligns with similar observations reported by Zhang et al.

(2022). Furthermore, due to the iterative nature of the training process, we can employ different optimizer settings tailored to the specific characteristics of each module.

This paper employs the following methodology to construct the Loss Function for training the adaptive graph learning module. Our dataset includes targets  $(\mathbf{v}^t, \mathbf{v}^{t+1})$ . We utilize the Mean Squared Error (MSE) loss function. MSE is calculated by comparing the predicted next-phase delay  $\hat{\mathbf{v}}^{t+1}$  obtained by using the output of DDP, GAE  $\hat{\delta}^t$  and  $\hat{\mathbf{B}}^t$  combined with  $\mathbf{v}^t$  through Eq. 39 against the actual  $\mathbf{v}^{t+1}$ . The formula is as follows:

$$l = \frac{1}{n} \sum_{i=1}^n (\mathbf{v}^{t+1} - \hat{\mathbf{v}}^{t+1})^2 \quad (45)$$

The training methodology adopted in this study is similar to the iterative training method used in the AdapGL architecture. The pseudo-code for training is as Algorithm 1 shows.

---

**Algorithm 1** Training GAE and DDP

---

```

1: time  $\leftarrow 0$ 
2: for each  $(data, graph)$  in datasets do
3:    $mod\_graph \leftarrow GAE(graph)$ 
4:   if time  $\neq 0$  then
5:     Fix GAE parameters
6:     Train DDP with data and mod_graph
7:   else
8:     Fix GAE parameters
9:     Train DDP with data and graph
10:  end if
11:  Fix DDP parameters
12:  Train GAE with data and graph
13:  time  $\leftarrow$  time + 1
14: end for

```

---

360 5.3. Initial condition estimation by Extended Kalman Filter

This section discusses how to accurately estimate the initial condition of the state transition model by integrating field data with mathematical predictions. We detail the role of the Extended Kalman Filter (EKF) in our model, including its principles and integration within our framework. Recall that we use the AdapGL architecture to train PGL and PN models separately to learn the parameters at each time step. These parameters are then incorporated into our ODE equations. However, the predictions derived from our ODE equations are likely to contain noise. We aim to use a filter to eliminate this noise. In Eq. 11, we use  $\bar{v}^{(t)}$  to represent the infection rate at the previous time step, which is theoretically correct. However, our sliding time window in practice inevitably includes randomness and uncertainty. Therefore, we consider using the EKF to absorb and counteract this noise.

The Kalman Filter is a recursive filter used to estimate the state of a dynamic system from a series of incomplete and noisy measurements. It is the optimal sequential data assimilation method developed for linear dynamic systems with Gaussian error statistics. The EKF is a variant of the Kalman Filter that extends it to nonlinear systems by

linearizing the system dynamics around the current estimate.

$$\hat{v}_{t|t-1} = f(\hat{v}_{t-1|t-1}, u_{t-1}) + w_t \quad (46)$$

$$P_{t|t-1} = F_t P_{t-1|t-1} F_t^T + Q_t \quad (47)$$

$$K_t = P_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + R_t)^{-1} \quad (48)$$

$$\hat{v}_{t|t} = \hat{v}_{t|t-1} + K_t(z_t - h(\hat{v}_{t|t-1})) \quad (49)$$

$$P_{t|t} = (I - K_t H_t) P_{t|t-1} \quad (50)$$

where  $F_t$  is the Jacobian of the state transition function,  $Q_t$  is the process noise covariance,  $H_t$  is the Jacobian of the measurement function,  $R_t$  is the measurement noise covariance.  $\hat{x}_{t|t-1}$  is the predicted state and  $\hat{x}_{t|t}$  is the updated one.

The process of implementing EKF is illustrated by Eqs. (46)–(50). Briefly, Eqs. (46) and (47) indicate the prediction step, and EKF uses the dynamic model of the system to predict the state of the next moment and its associated uncertainties. Eqs. (48)–(50) indicates the update step, and the filter updates the state of the prediction with the new measurement, optimally reducing uncertainty by combining the predicted and measured values.

In our project, the EKF is specifically used to filter out noise from the predicted delay rates, providing a more accurate estimate for the next time step. The EKF predicts the next state of the system using the dynamic model. For the ME-SIS model, this involves updating the infection rates using the learned parameters from the DDP and GAE models. The prediction equation is formulated as follows:

$$\hat{v}_{t|t-1} = (1 - \delta)\hat{v}_{t-1|t-1} + (I - \hat{v}_{t-1|t-1})\beta\hat{v}_{t-1|t-1} \quad (51)$$

The EKF then updates the predicted state using the actual observations. The observed delay rates serve as the measurement, and the EKF updates the predicted state to reduce the noise by the update steps Eqs. (48)–(50). By iterating through these steps, the EKF effectively reduces noise and improves the accuracy of state estimates in our delay propagation model. This integration of the EKF within our model ensures robust estimation of delay rates, accounting for inherent noise in the data and providing accurate inputs for subsequent predictions. This makes our model more stable and reliable, even in the presence of noisy measurements.

## 390 6. Propagation analysis and model explanation

In this section, we further explore the propagation of delays both in virtual and real networks. Section 6.1 focuses on analyzing infection and recovery dynamics in a set of predefined scenarios for synthetic networks. Section 6.2 uses real delay and infection rates, illustrating the propagation of airport delays on October 1, 2016, in a real network. The results show that our model provides a good explanation for the observed propagation process.

### 6.1. Experimental analysis on synthetic networks

This section presents the results of experiments conducted on synthetic networks to understand the dynamics of infection and recovery processes. Our experiments focused on four main aspects: the impact of initial infected node probability, the relationship between network size and expected convergence time, the comparison between networks with uniformly and concentrically distributed edges, and the relationship between convergence time and node centrality. The findings from these experiments are illustrated through the accompanying figures(see Fig. 4

and 5), which help elucidate the underlying mechanisms governing the infection and recovery dynamics in complex networks.

#### 6.1.1. Impact of initial infected nodes

The balance between infection spread and recovery processes is crucial. When the initial infection probability is low, the infection spreads slowly while recovery is swift, leading to shorter convergence times. As the initial infection probability increases, the infection spreads faster, resulting in more infected nodes and a prolonged recovery process. At very high initial infection probabilities, nearly all nodes need to be cured, and despite the rapid spread of infection, the recovery time causes the convergence time to stabilize. This trend reflects the fundamental dynamics of network propagation: different initial conditions manifest distinct temporal characteristics in the infection and recovery processes. As the initial infection probability increases from 0 to 1, the network transitions from an almost entirely healthy state to an almost wholly infected state, corresponding to varying dynamics of propagation and recovery (see Fig. 4).

#### 6.1.2. Relationship between airport scale and expected convergence time

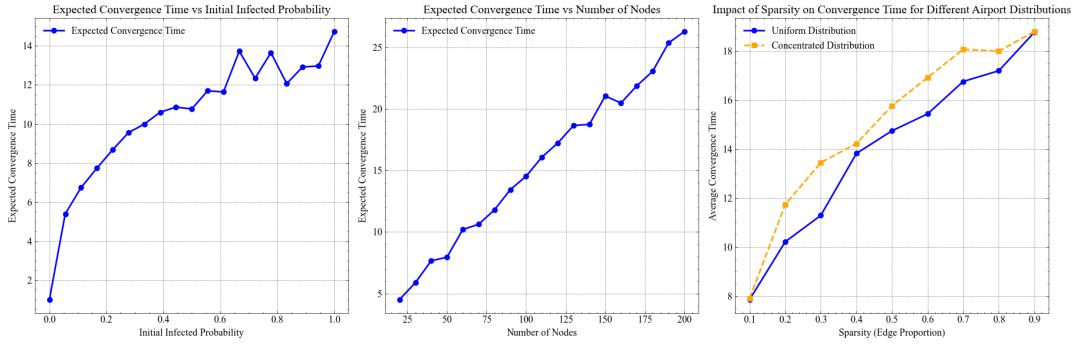
The relationship between network size and complexity is evident. The figures show that as the number of nodes increases, the number of connections and the network's complexity also increase significantly. This leads to longer paths for disease transmission and a more complex recovery process, thereby increasing the convergence time. In small-scale networks, the infection spreads through shorter paths, and recovery is more efficient, resulting in shorter convergence times. In larger networks, the complexity of infection spread and recovery increases significantly, as each node may connect with more neighbors, leading to a wider and faster spread of infection and a longer recovery process to cover the entire network (see Fig. 4).

#### 6.1.3. Comparison between uniformly and concentrically distributed edges

The network structure significantly impacts the infection spread and recovery processes. The figures show that networks with lower sparsity (fewer connections) have shorter transmission paths and convergence times. As sparsity increases, connections grow, transmission paths lengthen, and convergence time increases. Under the same sparsity conditions, networks with uniformly distributed edges across all nodes have longer convergence times than those with edges concentrated in a few nodes. This is because, in concentrated networks, highly connected central nodes can spread and cure infections more quickly, reducing the overall convergence time. Our experiments based on the virtual network model corroborate these findings and validate the proposition outlined in Proposition 2, which provides critical insights into how network structure and individual node properties affect the stability and spread of delays or infections. Specifically, the proposition emphasizes that nodes with higher connectivity impose stricter requirements on their effective infection rates to ensure overall network stability. This understanding is crucial for designing targeted control strategies to mitigate the spread of delays and maintain network stability (see Fig. 4).

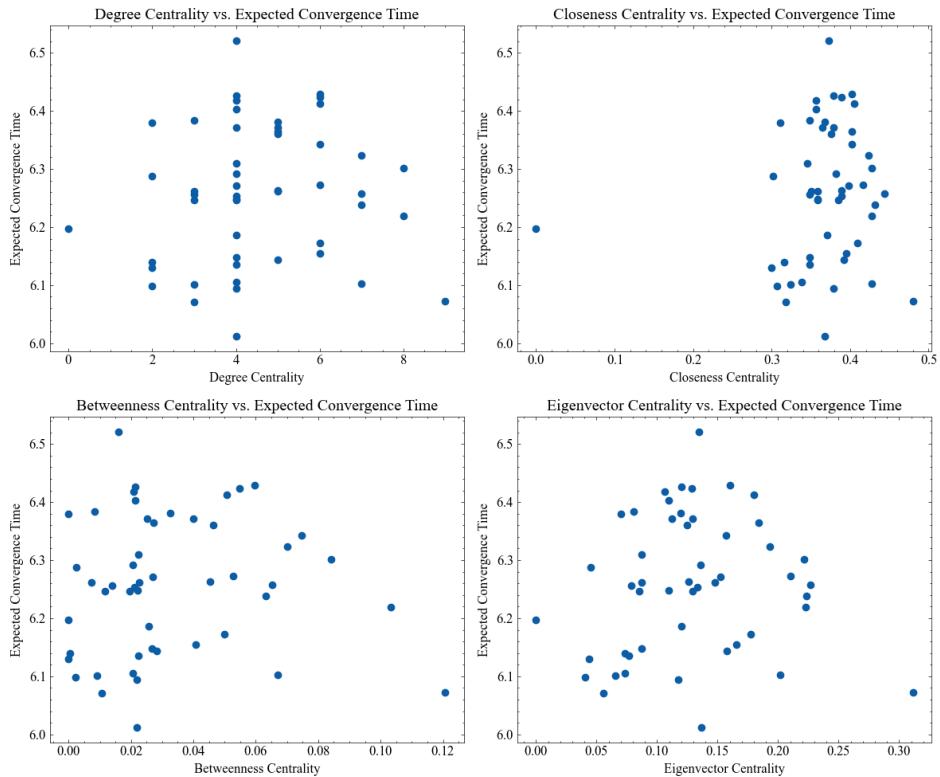
#### 6.1.4. Relationship between convergence time and node centrality

The figures demonstrate that the centrality of individual nodes does not significantly influence the convergence time. In the MEGA framework, both infection and recovery processes are global, with node interactions playing a pivotal role in determining the overall dynamics. As a result, even nodes with lower centrality may experience prolonged convergence times due to reinfection from neighboring nodes. Regardless of centrality, however, the



**Fig. 4.** Expected convergence time analysis for a network propagation model. The left panel shows the expected convergence time across different initial infection probabilities; the middle panel depicts the changes in expected convergence time with an increasing number of nodes; the right panel compares the expected convergence time under uniform, exponential, and concentrated degree distributions across various network sparsities.

convergence time across all nodes remains nearly uniform, indicating that the entire network progresses toward a dynamic equilibrium in the infection and recovery processes, with all nodes approaching a healthy state almost simultaneously. This uniform convergence time is a direct consequence of the concurrent infection and recovery processes in the ME-SIS model, where the interactions among nodes lead to a synchronized resolution of delays across the network (see Fig. 5). Importantly, our experimental setup strictly adheres to the convergence conditions for network propagation, allowing us to observe that, regardless of centrality, delays across all nodes eventually dissipate. This observation supports the proposition presented in Proposition 3, which asserts that, under the current conditions, even when individual nodes exhibit varying infection and recovery rates, the network will ultimately stabilize, and all delays will vanish.



**Fig. 5.** Relationship between various centrality measures and expected convergence time

Our experimental results provide valuable insights into the dynamics of infection and recovery in virtual networks. They highlight the significant influence of initial infection probability, network size, edge distribution, and node centrality on convergence times. These findings can guide the design of more efficient intervention strategies and improve our understanding of disease propagation in complex networks.

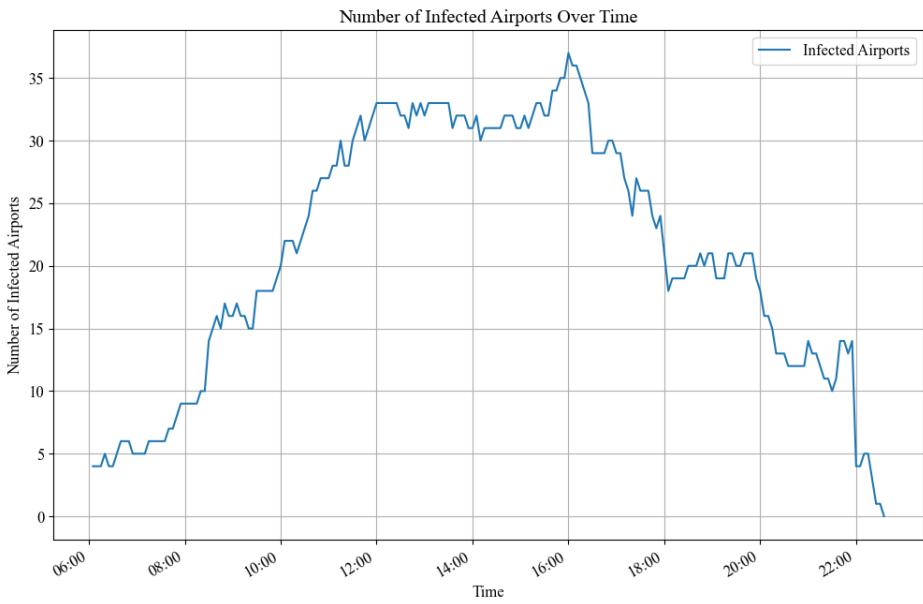
450

## 6.2. Propagation law on real data

We present the process of delay propagation throughout the day by tracking the number of airports experiencing delays. At the start of the day, the number of flights is relatively low, and the infection rate remains at a minimal level, so the number of airports affected by delays is also small. As the day progresses, flight density increases and delays begin to spread across the network, with the number of delayed airports gradually rising. This increase continues through the midday period, reaching a peak in the afternoon as the density of flights remains high. In the evening, the situation improves slightly due to decreased flight density. By 10 p.m., however, the number of flights drops significantly, causing the effective infection rate to fall below the critical threshold mentioned in Proposition 3, which leads to a rapid decline in the number of delayed airports. Within a short period, all delays dissipate completely.

460

The SIS model effectively simulates this real-world delay propagation process, aligning well with our intuitive understanding of how delays spread across a network. Moreover, the model's predictions validate the reasoning behind our Proposition 3, providing deeper insights into the dynamics of delay propagation. By capturing these key patterns, the SIS model not only proves to be a reasonable approach for modeling airport delays but also strengthens the theoretical foundation of our analysis, offering a valuable framework for further research in delay management and optimization strategies.



**Fig. 6.** Propagation of airport delays on October 1, 2016, based on real delay and infection rates. The curves represent the temporal evolution of delays across airports. The number of infected airports increases throughout the daytime as delays spread across the network, reaching a peak in the afternoon. In the evening, the number of infected airports gradually declines, with delays dissipating completely by the end of the day.

## 7. Experiments

In this section, we evaluate our model and present various experimental results on synthetic and real-world networks. Specifically, Section 6.1 focuses on analyzing infection and recovery dynamics in a set of pre-defined scenarios for synthetic networks. Section 7.1 provides an empirical analysis of real-world data, encompassing the preparation and processing of datasets, the experimental setup, the evaluation criteria, and a comprehensive analysis of the results through various comparative and sensitivity analyses, as well as a detailed case study.

### 7.1. Empirical analysis on real data

#### 7.1.1. Datasets and pre-processing

This research examines the spread of delay rates among airports across China using the MEGA framework to forecast delay rates at China’s major airports.

The dataset employed in this study originates from Ctrip and includes flight information spanning from May 1, 2015, to May 31, 2017. The data are aggregated into two-hour intervals, starting from the beginning of each interval. Due to the limited data available from midnight to 6:00 a.m., our analysis considers only the data from 480 6:00 a.m. to midnight the following day. The initial half of the dataset, covering May 1, 2015, to May 31, 2016, is designated as the training set, while the subsequent portion is utilized for testing purposes. This investigation focuses specifically on flights connecting the top fifty airports in China. To ensure a closed system for the study, flights involving other airports are grouped under a hypothetical airport code “XXX.” This approach allows for the examination of interactions between this virtual node and the fifty actual airports, excluding internal flows within each airport. As a result, the aviation network is modeled as a graph consisting of fifty-one nodes, with airports ranked by size and the fictitious airport assigned as node number 51.

The original flight data were transformed into two formats: First, We computed the proportion of delayed flights over total flights for each of the fifty-one airports every two-hour window. Delays under 15 minutes are unacceptable and are not counted as delayed flights—the formula for calculating the delay rate in Eq.38. Second, We defined 490 an adjacency matrix  $\mathbf{W}^t$  for each time interval  $t$ , where nodes represent the 51 airports and the edge weight  $w_{ij}^t$  is defined as the normalized total number of flights from airport  $i$  to airport  $j$ , using Eq. 35. The infection rate  $\mathbf{B}^t$  is calculated by Eq. 37.

This experiment compiles the processed data into the dataset:  $\mathcal{D}(\mathbf{W}, \mathbf{v})$ , where  $\mathbf{W}$  is the input for the DDP and GAE, and  $\mathbf{v}$  is the value to predict.

#### 7.1.2. Experimental setting

Experiments are conducted on a high-performance computing platform. For each experiment, a computing node is allocated, comprising one CPU (Intel Xeon Gold 6152 @ 2.10 GHz, 44 cores) and three GPUs (NVIDIA TITAN RTX, 24 GB memory).

In this experiment, the model employed a Graph Neural Network composed of layers using multi-head attention 500 with 5 GAT layers. Each layer has 5 heads. The training was carried out using the Adam optimizer, with a learning rate of  $10^{-5}$ , a dropout rate of 20%, and over 100 epochs.

### 7.1.3. Evaluation metrics

This paper evaluates the proposed and baseline models using two metrics: Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). The formulas for these metrics are given by:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=0}^N (v_i^t - \hat{v}_i^t)^2} \quad (52)$$

$$MAE = \frac{1}{N} \sum_{i=0}^N |v_i^t - \hat{v}_i^t| \quad (53)$$

Here,  $N$  denotes the total number of airports, and  $v_i^t$  represents the delay rate at airport  $i$  at time  $t$ .

### 7.1.4. Baselines and results comparison

The baseline models encompass typical approaches from statistical analysis, machine learning, and graph neural network methodologies. These include:

- SVR (Drucker et al., 1996): It identifies delay patterns by using a kernel function to map non-linear delay data onto an optimal hyperplane in high-dimensional space.
- GRU (Cho, 2014): A type of recurrent neural network that captures sequential dependencies by using gating mechanisms to control the flow of information.
- GC-LSTM (Chen et al., 2021): This model integrates GCN with Long Short-Term Memory (LSTM) networks to predict time series by learning both spatial and temporal features.
- ASTGCN (Guo et al., 2019): This model extracts multi-level periodic patterns by integrating a spatial-temporal attention mechanism with convolution techniques.
- STGCN (Yu et al., 2017): This is a Spatial-Temporal Graph Convolutional Network. It combines both spatial and temporal information of a graph.
- ME-SIS (Ceria et al., 2021): This model represents the original modeling approach introduced in Ceria's work, utilizing a heterogeneous SIS model framework to depict the system's dynamics.
- AdapGL (Zhang et al., 2022): The AdapGL model combines graph learning modules and time-series prediction modules.
- STPN (Wu et al., 2022): A model that captures both spatial and temporal dependencies through multi-stage propagation, improving predictions for dynamic systems.
- FAST-CA (Li et al., 2024b) This model integrates dynamic graph learning, coupled attention mechanisms, spatial-temporal feature extraction, and incorporates weather data for predicting delays.

We conducted predictions over 12 days (108 time units) between October 1. 2016, and October 12. 2016, achieving a RMSE of 0.141 and a Mean Absolute Percentage Error (MAE) of 0.101 at the 12th time unit.

Table 2 displays the RMSE and MAE losses of the baseline model and the model proposed in this paper

oversampled time units 3, 6, and 12 during the period from October 1, 2016, 06:00 to October 2, 2016, 22:00.

**Table 2**

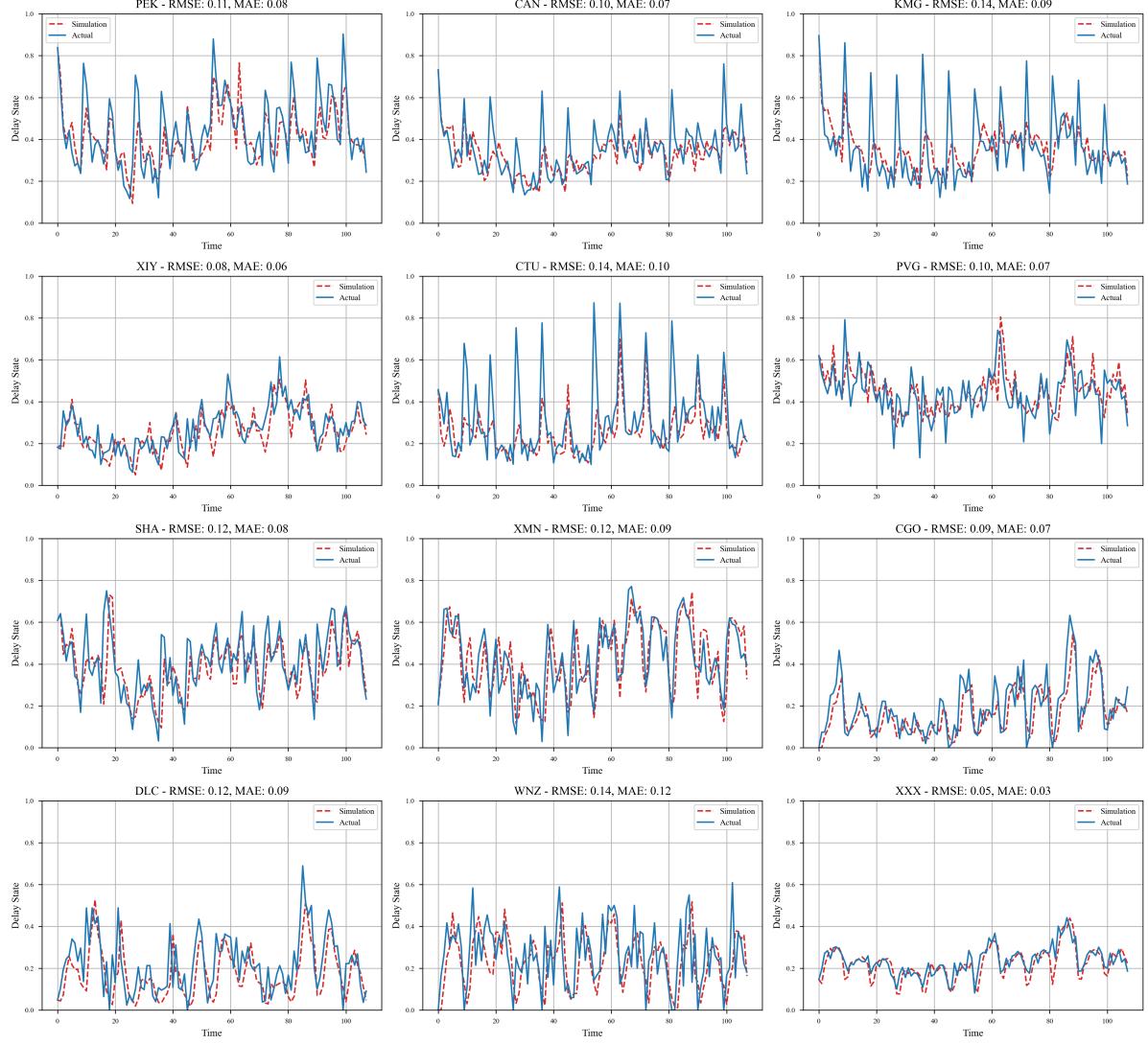
Results on the China delay dataset.

Method	3 steps		6 steps		12 steps	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
SVR	0.190	0.239	0.215	0.250	0.235	0.274
GRU	0.152	0.190	0.127	0.163	0.133	0.173
ME-SIS	0.114	0.168	0.144	0.194	0.173	0.216
GC-LSTM	0.114	0.146	0.117	0.150	0.121	0.155
AdapGL	0.123	0.178	0.128	0.181	0.158	0.215
STGCN	0.120	0.158	0.111	0.145	0.117	0.155
ASTGCN	0.118	0.149	0.123	0.156	0.134	0.169
STPN	0.127	0.163	0.112	0.146	0.112	0.151
FAST-CA	0.097	0.130	0.108	0.135	0.114	0.144
<b>MEGA</b>	<b>0.085</b>	<b>0.127</b>	<b>0.091</b>	<b>0.131</b>	<b>0.101</b>	<b>0.141</b>

Firstly, according to Table 2, the proposed model performs optimally among all models in terms of both RMSE and MAE, with values of 0.141 and 0.101, respectively. Secondly, the fitting losses of each model tend to increase over time, which illustrates the characteristic accumulation of losses in time series forecasting tasks. Lastly, the MAE values for all models are generally lower than the RMSE values, which is due to RMSE magnifying larger losses, thus typically resulting in higher values compared to MAE.

According to Table 2, the SVR model, a non-deep learning model, performs the worst among all considered models, possibly due to its insufficient capability to capture the complex relationships in the dataset. In contrast, the STGCN, ASTGCN, and GC-LSTM models, which are spatiotemporal graph neural networks, perform better because they integrate both temporal and spatial features and possess more robust fitting capabilities due to their deep learning frameworks. The AdapGL model slightly underperforms compared to these three spatiotemporal graph neural network models, possibly due to the original AdapGL model lacking support for dynamic weighted directed graphs like those found in aviation networks. Notably, the ME-SIS model, as a non-deep learning model, performs comparably to the deep learning baseline models in the initial time periods but shows a significant increase in loss in later fittings. This may be due to the disease propagation model aptly modeling delay propagation across airport networks, although the ME-SIS model’s lack of deep learning elements results in weaker fitting capabilities.

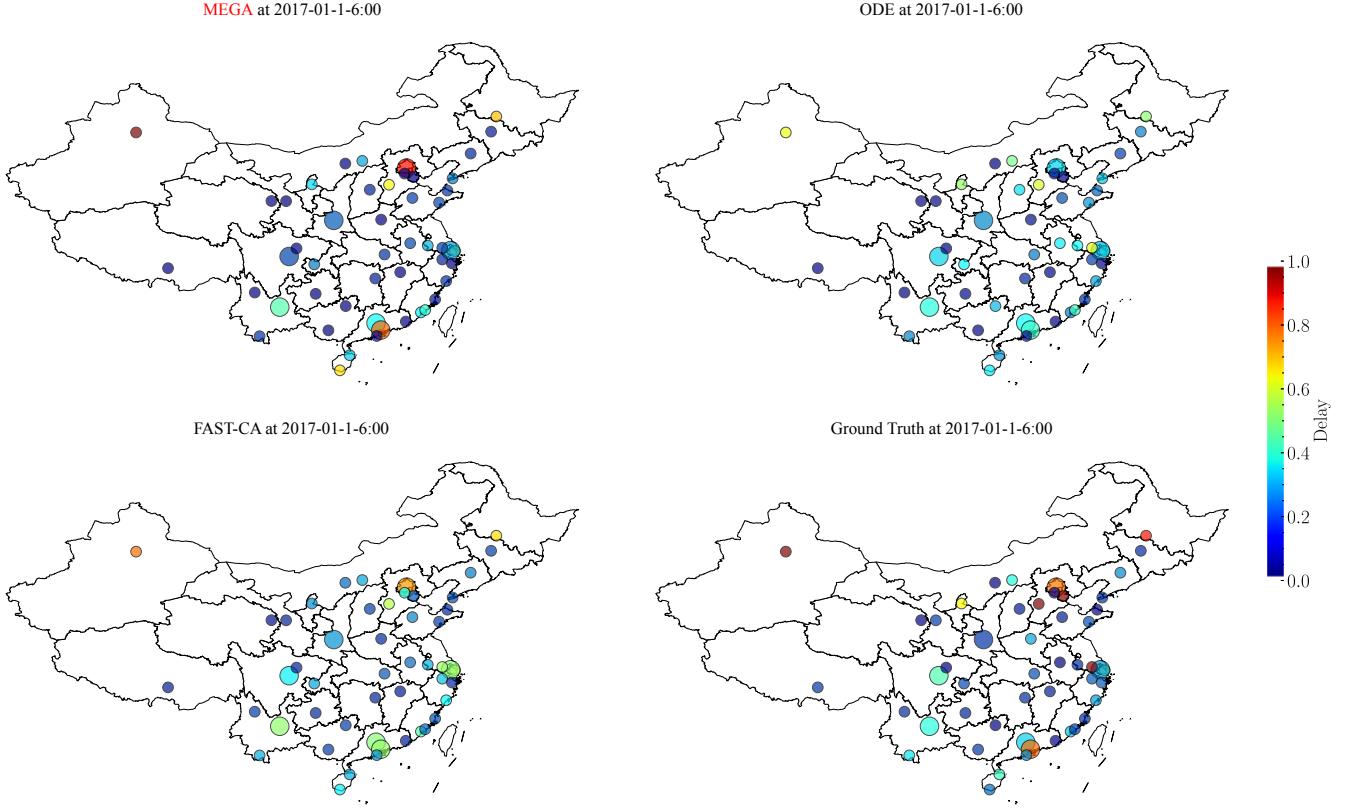
This experiment illustrated the fitting curves of the proposed model for 12 selected airports out of 51, processed in Section 7.1.1, from October 1, 2016, 06:00 to October 12, 2016, 22:00, spanning 108 time units. The selection method involved initially choosing the top five largest airports in China to demonstrate the model’s fitting ability for major airports, then randomly selecting six airports from the remaining 45 to show its capability with smaller-scale airports, and finally, selecting a fictitious 51st airport to demonstrate the validity of the “contraction” operation. Fig. 7 arranges the selected airports in decreasing order of size from left to right, top to bottom. Each subplot is labeled with the corresponding airport’s three-letter code (with XXX airport representing the fictitious airport derived from the “contraction” process in 7.1.1) and displays the average RMSE and MAE loss over the 108 prediction time units. In each subplot, red dashed lines represent the predicted delay rates at each time unit, while blue solid lines represent the actual delay rates, forming line graphs. The proposed model generally exhibits good fitting across all airports, indicating strong fitting capabilities for individual airport delay situations and generalizability across



**Fig. 7.** The visualization of simulated curve comparing with truth curve

the air traffic system. More importantly, the model's effective fitting for the XXX fictitious airport validates the feasibility of contracting out-of-scope airports in the aviation network for the study.

Fig. 8 illustrates the spatial distribution of the top 50 airports in mainland China. Each node represents an airport, with the color of the node indicating the level of delay and the size of the node corresponding to the airport's traffic volume. Fig. 8 (a) shows the actual delay rates for these airports on October 1, 2016, while Fig. 8 (b) presents the predicted delay rates for the same time using the model proposed in this paper. We observe that the top 50 airports are primarily concentrated in eastern China, with significant delays predominantly affecting the Beijing-Tianjin-Hebei region, the Yangtze River Delta, and the Pearl River Delta. Furthermore, the model demonstrates a strong fit across airports in different regions, reflecting its ability to generalize spatially. This robustness is likely since the model is based on the ME-SIS epidemic transmission framework. The extension of the recovery rate in the ME-SIS model, interpreted in this context as a negative recovery rate to represent delays caused by the airports themselves—takes into account multiple factors, including the geographical location of different



**Fig. 8.** This spatial visualization depicts the delay rates, with different colors corresponding to distinct temporal values, and the size of the circles representing the traffic volume at each airport. The figure further demonstrates that prediction errors remain consistently low across various regions.

airports. Additionally, the graph convolutional learning model in MEGA captures how these factors influence both the occurrence and propagation of delays based on historical data.

#### 7.1.5. Ablation study

The black-box component of the model proposed in this paper mainly consists of three parts: the DDP, the GAE, and the EKF Filter. This experiment will validate the role of each part by selectively removing different components.

**Table 3**

Results on ablation study.

Method	3 steps		6 steps		12 steps	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
<b>Our Model</b>	<b>0.085</b>	<b>0.127</b>	<b>0.091</b>	<b>0.131</b>	<b>0.101</b>	<b>0.141</b>
MEGA w/o (DDP)	0.285	0.418	0.315	0.438	0.351	0.474
MEGA w/o GAE	0.096	0.137	0.100	0.139	0.108	0.146
MEGA w/o Sliding Window	0.094	0.135	0.097	0.137	0.103	0.145
MEGA w/o EKF	0.130	0.193	0.168	0.233	0.173	0.231

- MEGA w/o DDP: The Removing DDP experiment will use only the GAE model to calculate the infection rate matrix  $\mathbf{B}$ . For the recovery rate vector, this experiment will employ Eq. 36 to compute recovery states

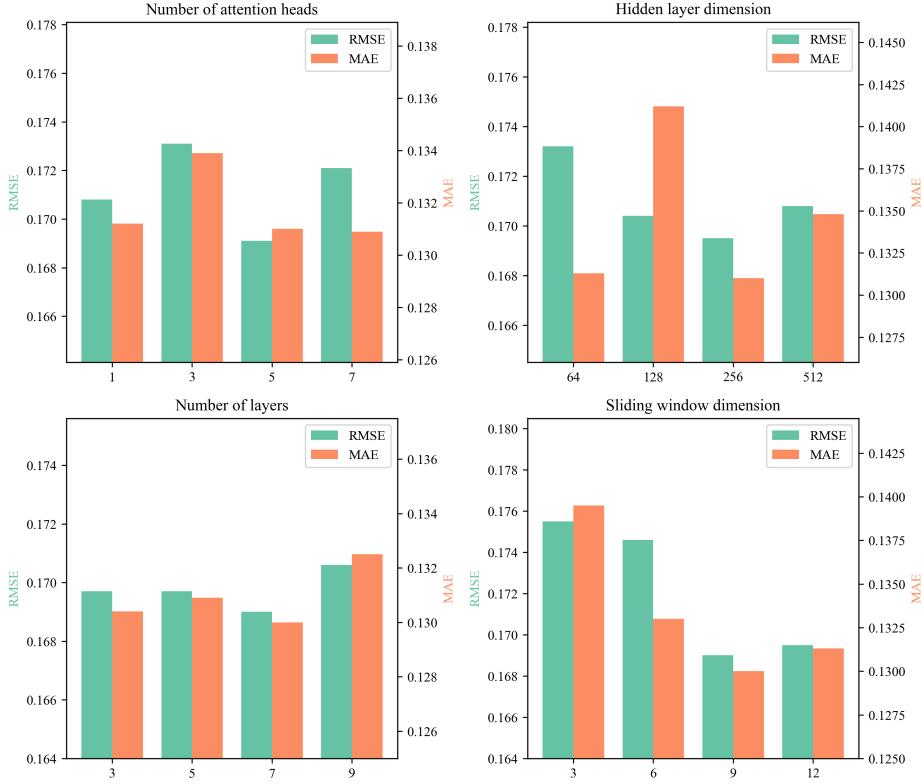
that replace the DDP neural network in the original architecture.

- MEGA w/o GAE: This experiment ignores GAE and investigates the consequences of using only the DDP for prediction. The experiment replaces the GAE with the original flight traffic matrix proposed in formulas 580 2 and 3 for training.
- MEGA w/o Sliding Window: The Removing Sliding Window experiment studies the consequences of removing the sliding window. The experiment will use only the current iteration’s flight traffic data for prediction,
- MEGA w/o EKF: The Removing EKF Filter experiment investigates the consequences of removing the EKF Filter. The experiment will eliminate the EKF Filter and use the prediction from the previous time unit for iteration.

This experiment investigates the performance changes of the proposed model after removing the DDP, GAE, Sliding Windows, and EKF modules, illustrating the contributions of each module to the model’s effectiveness. Similar to Section 7.1.4, this experiment also displays the RMSE and MAE losses sampled over three-time units—3, 6, and 12 hours—during the period from October 1, 2016, at 6:00 AM to October 2, 2016, at 10:00 PM, as shown 590 in Table 3. By comparing the fit losses over time between the proposed model and other models, Table 3 reveals several key observations: (1) Our model exhibits the smallest RMSE and MAE losses among all models tested during the three sampling units, indicating that the interaction between the modules effectively captures the propagation characteristics of delays across the aviation network. (2) The introduction of DDP significantly enhances model performance. This improvement stems from DDP’s ability to predict the delay rates across different airports, which, as described in Section 5.2.2, results from the competing effects of an airport’s inherent delay generation capacity and its ability to manage delays. Thus, DDP’s output integrates these two capacities. Using only Eq. 36 for modeling, as mentioned in Section 5.1.1, would lead to issues. Eq. 36 predicts an airport’s recovery capabilities solely based on its traffic per unit time through a fixed formula, which might not accurately reflect the actual situation at the airport. Therefore, DDP plays a crucial role in forecasting the capabilities of each airport to generate and handle 600 delays per time unit. (3) The introduction of GAE improves model performance, though to a lesser extent than DDP. As stated in Section 5.2.2, GAE determines the rate of delay propagation from the departure to the arrival airport (i.e., the infection rate) per unit of time. The infection rate for each flight route is determined by weighting the route’s traffic volume per unit time. The GAE model assigns varying weights to each flight route over different times, whereas the original method applied a uniform weight to all routes at any given time. This principle underlies the improved predictive ability of the GAE model, although the extent of improvement is relatively minor since DDP expands the definition of “recovery rate” in the traditional disease propagation model to accommodate the network nature of airports, allowing for negative recovery rates where delays can arise due to internal issues at the airport. (4) The introduction of Sliding Windows enhances model performance by incorporating historical flight traffic data, providing additional features for DDP and GAE predictions. (5) Including the EKF module improves 610 the model’s accuracy. As described in Section 5.3, the EKF module filters the predictions from the previous time unit, enhancing the predictability of the input data.

#### 7.1.6. Parameter sensitivity analysis

In our experiment, we fixed the Feed Forward Network that handles output dimensions after the GAT for DDP and GAE with three layers and a hidden layer dimension of 2048. The input dimension is the flattened



**Fig. 9.** Experimental results with different parameter settings.

output dimension of the GAT's hidden layer, and the output dimension corresponds to the overall module's output dimension. The structures for DDP and GAE were kept identical except for the output layer dimension to simplify the discussion.

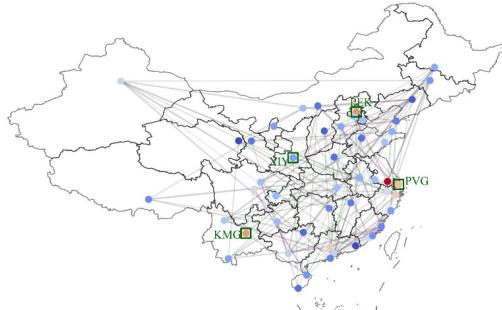
Firstly, increasing the Attention Head Number allows the GAT model to focus on various graph structure features simultaneously, enhancing model performance. However, an excessively high Attention Head Number could lead to training difficulties. Furthermore, if the task does not require attention to many features, increasing the Attention Head Number is unnecessary. Secondly, increasing the Hidden Layer Dimension allows the model to access a higher-dimensional latent space, potentially improving performance. However, a dimension that is too large might also lead to training challenges and an increased risk of overfitting. Thirdly, increasing the Number of Layers enables the GAT model to have a larger receptive field, integrating relationships between flight routes over a more extensive spatial area, but this could result in vanishing or exploding gradients. Lastly, increasing the size of the Sliding Windows provides the model with a broader range of temporal information, enabling it to assess the current state of the air transportation network based on historical data. However, excessively long historical data may not be necessary and could interfere with neural network learning. Experimental results, shown in Fig. 9, demonstrate that setting the DDP and GAE Attention Head Number to 5, Hidden Layer Dimension to 256, Number of Layers to 7, and Sliding Windows Length to 9 units of time yields optimal results.

#### 7.1.7. Case study I: model parameter explanation

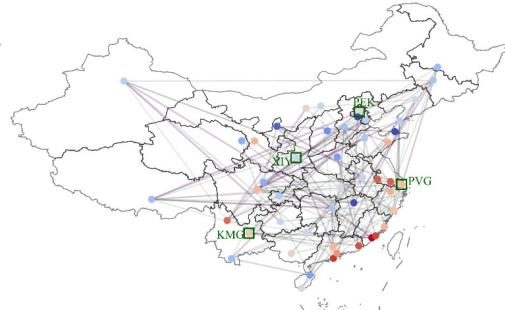
We have visualized the process of delay propagation through three diagrams to highlight our model's insights. Fig. 10 (a) graphically demonstrates the dynamic process of delay propagation between airports across four diagrams, deliberately omitting delays caused directly by poor airport management. Each node in these diagrams represents

(a)

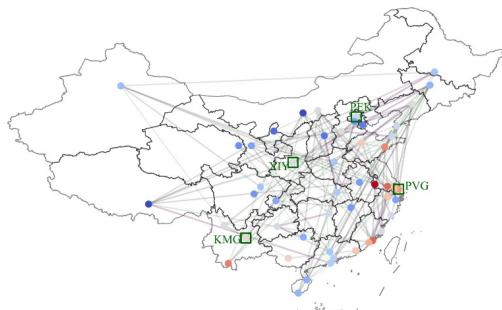
2016-10-01 08:00 Delay propagation



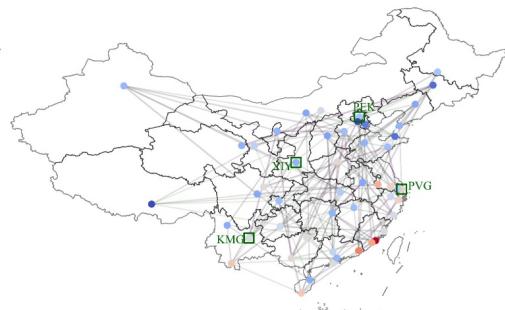
2016-10-01 12:00 Delay propagation



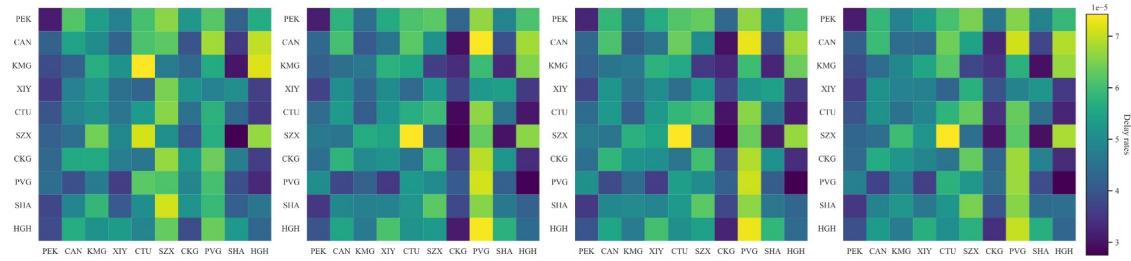
2016-10-01 16:00 Delay propagation



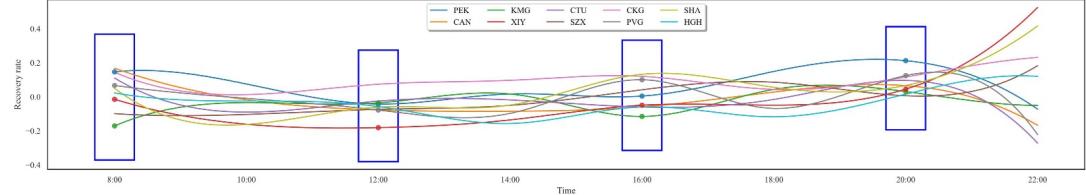
2016-10-01 20:00 Delay propagation



(b)



(c)



**Fig. 10.** Delay propagation and recovery in China's airport network. (a) Maps illustrate delay propagation across Chinese airports throughout the day. Nodes, highlighted for five key airports, show delays with colors ranging from minimal (green) to severe (purple). Red lines indicate significant delay propagation between airports. (b) Heatmaps depict delay rates between the top ten airports at different times, where darker shades suggest lower delays and lighter shades have higher delays. (c) A line graph details the recovery rates from delays at the highlighted airports in (a), with negative values indicating worsening conditions.

an airport, with the color depth of each node reflecting the proportion of flight delays at a specific time. Nodes tending towards purple indicate more severe delays at those airports at that moment, while those leaning towards

green suggest lighter delay conditions. Furthermore, the lines between nodes depict the delay propagation between airports; the closer a line's color is to red, the more significant the propagation, showing how delays at one airport directly affect another via flights, thus spreading throughout the aviation network.

640 Fig. 10 (b) displays a heatmap of the delay matrices for the top ten airports across four different time segments of the day. The color depth of each grid cell reflects the delay rate: darker colors indicate lower delay rates between two airports, whereas lighter colors suggest higher rates. Fig. 10 (c) then presents the variation in recovery rates from 8 AM to 10 PM, where the level of recovery not only reflects an airport's capability to manage delays but may also suggest significant delays were experienced during that period. Suppose a node color shows a negative recovery rate. In that case, it indicates that at that moment, the airport failed to alleviate delays and incurred additional delays due to poor management.

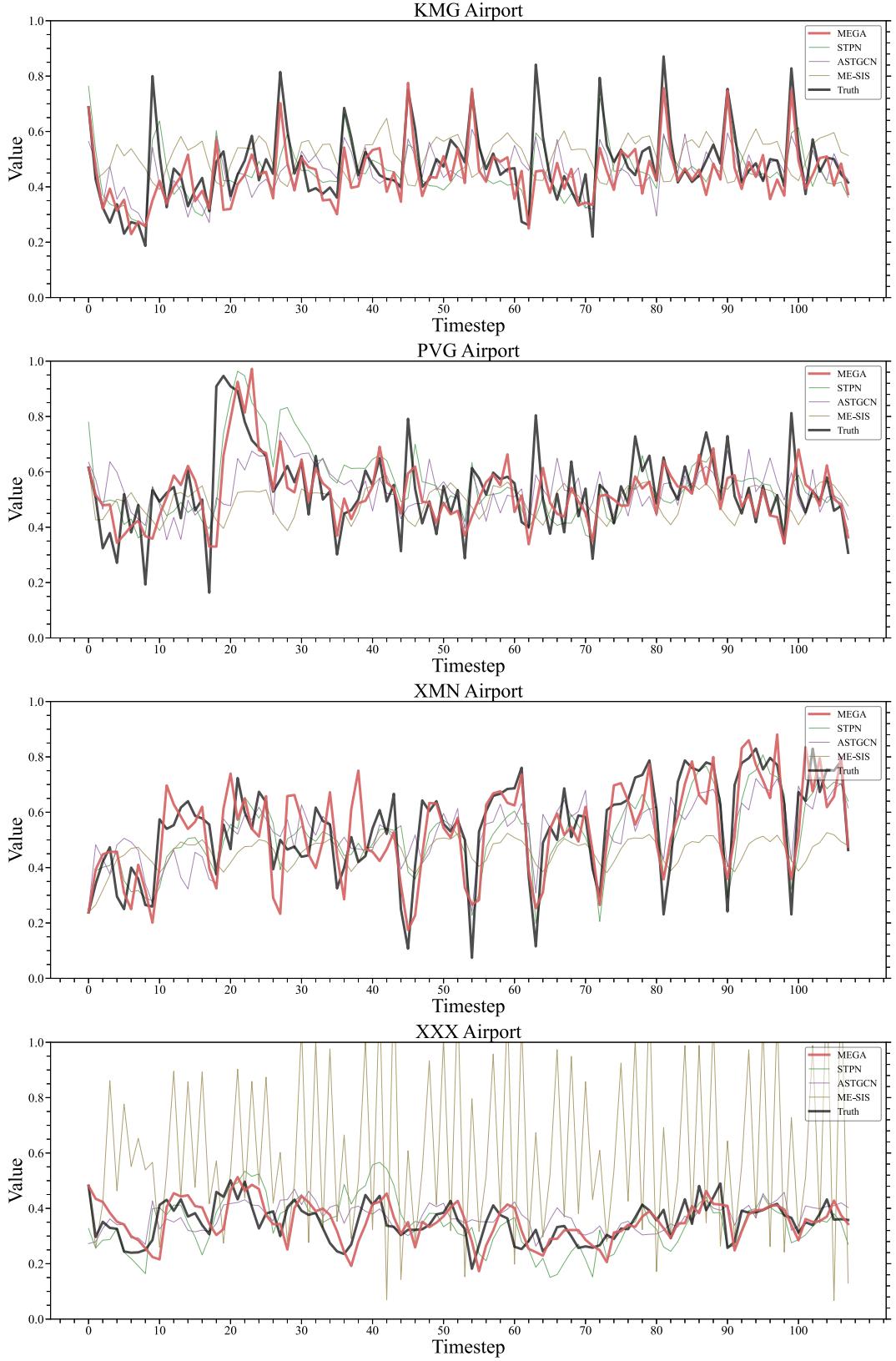
Specifically, in Fig. 10 (a), during the morning rush, some hub airports begin to experience flight delays. Although most other airports do not encounter widespread delays, the propagation initiated by the morning rush is quite significant, highlighting the latency inherent in delay propagation. By noon, influenced by the morning's 650 delays, the central region sees a substantial increase in the proportion of delayed flights, with more frequent delay propagation occurring between airports with high delays, such as Shanghai Pudong International Airport(PVG) experiencing high delays throughout the day, thus causing severe delay propagation to other airports as shown in Fig. 10 (b). By afternoon, many of the previously delayed airports gradually resume normal operations; however, with the onset of a new peak period, new airports with high delay proportions emerge. By evening, most airports in the network have largely returned to normal, and the propagation of delays significantly slows down.

It's important to note that not all airport delays are caused by propagation. Some airports display high proportions of delays without influence from preceding flights, reflecting poor self-management. For instance, Kunming Changshui International Airport(KMG) encountered significant flight delays as early as 8 AM without severe delay propagation from other airports. Its consistently low, even negative, recovery rates throughout the day indicate weak 660 delay management capabilities. In contrast, Beijing Capital International Airport(PEK) maintained a low delay rate throughout the day, as shown in Fig. 10 (c), thanks to its consistently high recovery rate effectively addressing flight delays. Similarly, PVG, another major transport hub, experienced noticeable delay propagation from other airports. Its fluctuating, generally low recovery rate led to a high proportion of flight delays throughout the day, suggesting that its delay management capabilities need improvement. Xi'an Xianyang International Airport(XIY), while maintaining a low recovery rate during the day, kept a low proportion of delays, indicating that the day's flight delays were mild and did not pose a significant challenge to airport operations.

When synthesizing all three diagrams, we observe that the recovery rates depicted in Fig. 10 (c) tend to lag behind the propagation of delays. Most airports experienced their most severe delays around noon but welcomed lower recovery rates. This reflects that some airports chose to conserve their recovery capacity after managing the 670 morning peak delays. However, as more widespread delays arrived at noon, their slow decision responses led to a phenomenon where some airports experienced an inversion between delay propagation and recovery rates.

#### 7.1.8. Case study II: scalability analysis

In Section 7.1.4, the experiments used data from October 2016, which is in autumn. This study further extends the analysis to winter, explicitly using data from January 2017. The model fits are compared with several baseline results. Data spanning 12 days, from January 11 at 06:00 to January 22 at 22:00, covering 200 time units, are



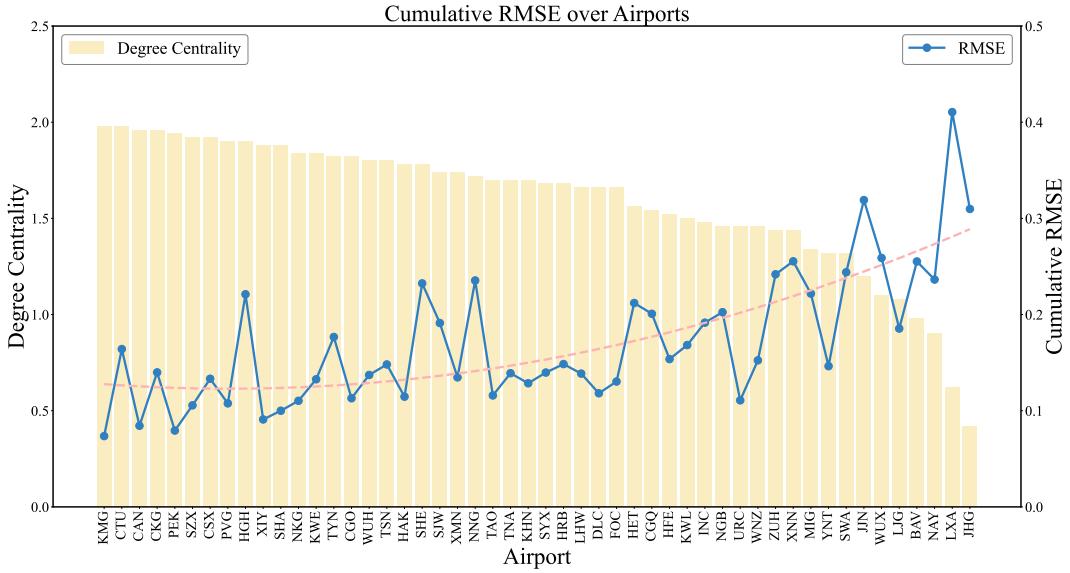
**Fig. 11.** Simulated curve for case study II.

selected for model fitting. We conducted visualizations for four airports randomly chosen from the airports in China. Fig. 11 presents the fitting results. The model proposed in this paper shows superior fitting performance for different airports and significantly outperforms the traditional mathematical models like the original SIS model

and deep learning models like STPN. This experiment shows that the model demonstrates strong generalizability, yielding robust results across different months and for various airports.

#### 7.1.9. Case study III: Prediction loss analysis

As shown in Fig. 12, the average traffic in 2016 and the degree centrality for each airport are presented. The analysis reveals a negative correlation between degree centrality and the cumulative RMSE over a 12-day forecast period. Airports with a higher degree of centrality tend to have lower cumulative prediction losses. This relationship likely arises from the fact that airports with a higher degree of centrality are more interconnected, leading to more stable and predictable traffic patterns, which enhances the forecasting accuracy of the model.



**Fig. 12.** Average traffic in 2016 and degree centrality for each airport.

## 8. Conclusion and future works

In this paper, we proposed the Mean-field Epidemic Graph Adaptive Framework (MEGA) to address the challenging problem of network-wide delay propagation dynamics prediction. By integrating the Mean-field SIS epidemic model (ME-SIS) with an adaptive graph neural network, our framework leverages both mechanistic insights and advanced learning capabilities to accurately capture and predict the spread of delays across the aviation network. The use of an extended Kalman Filter further enhances the model's predictive accuracy by maintaining consistency with system dynamics. Our extensive experiments, conducted on both simulated and real-world datasets, demonstrate that MEGA outperforms existing state-of-the-art models in terms of prediction accuracy and generalizability. Moreover, the ablation studies and case studies confirm the individual contributions of each component in the framework, highlighting the model's robustness and adaptability. These findings suggest that MEGA holds great promise for advancing the understanding and management of delay propagation in air traffic networks, offering a valuable tool for both researchers and practitioners in the field.

Despite the success of the MEGA framework, there are several avenues for future improvements. First, incorporating the heterogeneity of airport nodes, such as differences in airport capacity, operational characteristics, and passenger flow, could further enhance the model's realism and predictive accuracy. Second, expanding the temporal

scope to consider longer-term and seasonal variations in delay propagation would allow for more comprehensive delay management strategies. Third, integrating additional data sources, such as weather conditions, airline schedules, and socio-political events, could further refine the model's predictive capabilities and offer a more holistic view of the factors contributing to delay propagation. These improvements could significantly enhance the applicability and accuracy of MEGA in diverse operational contexts.

## Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (NSFC) under grants U1733102, 72394362, and 72101162; in part by the Guangdong Provincial Key Laboratory of Big Data Computing, 710 The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen) under grant 2024SC0003; in part by the Shenzhen Science and Technology Innovation Committee under the Shenzhen Stability Science Program grant 2024SC0010; in part by CUHK-Shenzhen under grant PF.01.000404; and in part by the Internal Project Fund from Shenzhen Research Institute of Big Data under grant P00120230005.

## References

- Bao, J., Yang, Z., and Zeng, W. (2021). Graph to sequence learning with attention mechanism for network-wide multi-step-ahead flight delay prediction. *Transportation Research Part C: Emerging Technologies*, 130:103323.
- Baspinar, B. and Koyuncu, E. (2016). A data-driven air transportation delay propagation model using epidemic process models. *International Journal of Aerospace Engineering*, 2016(1):4836260.
- Baspinar, B., Tutku Altun, A., and Koyuncu, E. (2021). Event-based air transport network resiliency management 720 with meta-population epidemic model. *Journal of Aerospace Information Systems*, 18(9):632–644.
- BTS (2021). Bts report. understanding the reporting of causes of flight delays and cancellations.
- Cai, K., Li, Y., Fang, Y.-P., and Zhu, Y. (2021). A deep learning approach for flight delay prediction through time-evolving graphs. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):11397–11407.
- Ceria, A., Köstler, K., Gobardhan, R., and Wang, H. (2021). Modeling airport congestion contagion by heterogeneous sis epidemic spreading on airline networks. *Plos one*, 16(1):e0245043.
- Chakrabarti, D., Wang, Y., Wang, C., Leskovec, J., and Faloutsos, C. (2008). Epidemic thresholds in real networks. *ACM Transactions on Information and System Security (TISSEC)*, 10(4):1–26.
- Chen, J., Wang, X., and Xu, X. (2021). Gc-lstm: Graph convolution embedded lstm for dynamic link prediction.
- Cho, K. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.
- Dai, X., Hu, M., Tian, W., and Liu, H. (2018). Modeling congestion propagation in multistage schedule within an airport network. *Journal of Advanced Transportation*, 2018.
- Drucker, H., Burges, C. J., Kaufman, L., Smola, A., and Vapnik, V. (1996). Support vector regression machines. *Advances in neural information processing systems*, 9.

- FAA (2021). Federal aviation administration. benefit-cost analysis.
- Ganesh, A., Massoulié, L., and Towsley, D. (2005). The effect of network topology on the spread of epidemics. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 2, pages 1455–1466. IEEE.
- Guo, S., Lin, Y., Feng, N., Song, C., and Wan, H. (2019). Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, page 922–929.
- Jiang, W. and Luo, J. (2022). Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications*, page 117921.
- Li, C., Mao, J., Li, L., Wu, J., Zhang, L., Zhu, J., and Pan, Z. (2024a). Flight delay propagation modeling: Data, methods, and future opportunities. *Transportation Research Part E: Logistics and Transportation Review*, 185:103525.
- Li, C., Qi, X., Yang, Y., Zeng, Z., Zhang, L., and Mao, J. (2024b). Fast-ca: Fusion-based adaptive spatial-temporal learning with coupled attention for airport network delay propagation prediction. *Information Fusion*, page 102326.
- Li, C., Yu, L., Mao, J., Cong, W., Pan, Z., Du, Y., and Zhang, L. (2024c). How did international air transport networks influence the spread of covid-19? a spatial and temporal modeling perspective. *Transportation Research Part C: Emerging Technologies*, 165:104730.
- Li, M. Z., Gopalakrishnan, K., Pantoja, K., and Balakrishnan, H. (2021). Graph signal processing techniques for analyzing aviation disruptions. *Transportation Science*, 55(3):553–573.
- Li, Q., Jing, R., and Dong, Z. S. (2023). Flight delay prediction with priority information of weather and non-weather features. *IEEE Transactions on Intelligent Transportation Systems*.
- Li, S., Xie, D., Zhang, X., Zhang, Z., and Bai, W. (2020). Data-driven modeling of systemic air traffic delay propagation: an epidemic model approach. *Journal of Advanced Transportation*, 2020.
- Rebollo, J. J. and Balakrishnan, H. (2014). Characterization and prediction of air traffic delays. *Transportation research part C: Emerging technologies*, 44:231–241.
- Shi, C., Li, Y., Zhang, J., Sun, Y., and Philip, S. Y. (2016). A survey of heterogeneous information network analysis. *IEEE Transactions on Knowledge and Data Engineering*, 29(1):17–37.
- Sun, J., Dijkstra, T., Aristodemou, C., Buzetelu, V., Falat, T., Hogenelst, T., Prins, N., and Sliper, B. (2022). Designing recurrent and graph neural networks to predict airport and air traffic network delays. In *10th International Conference for Research in Air Transportation*, pages 1–8. FAA & Eurocontrol.
- Van Mieghem, P., Omic, J., and Kooij, R. (2008). Virus spread in networks. *IEEE/ACM Transactions On Networking*, 17(1):1–14.

- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.
- 770 Wu, W., Zhang, H., Feng, T., and Witlox, F. (2019). A network modelling approach to flight delay propagation: Some empirical evidence from china. *Sustainability*, 11(16):4408.
- Wu, Y., Yang, H., Lin, Y., and Liu, H. (2022). Spatiotemporal propagation learning for network-wide flight delay prediction.
- Wu, Y., Yang, H., Lin, Y., and Liu, H. (2023). Spatiotemporal propagation learning for network-wide flight delay prediction. *IEEE Transactions on Knowledge and Data Engineering*.
- Yang, H., Du, L., Zhang, G., and Ma, T. (2023). A traffic flow dependency and dynamics based deep learning aided approach for network-wide traffic speed propagation prediction. *Transportation research part B: methodological*, 167:99–117.
- 780 Yang, H., Yu, W., Zhang, G., and Du, L. (2024). Network-wide traffic flow dynamics prediction leveraging macroscopic traffic flow model and deep neural networks. *IEEE Transactions on Intelligent Transportation Systems*.
- Yu, B., Yin, H., and Zhu, Z. (2017). Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*.
- Zeng, W., Li, J., Quan, Z., and Lu, X. (2021). A deep graph-embedded lstm neural network approach for airport delay prediction. *Journal of Advanced Transportation*, 2021:1–15.
- Zhang, H., Wu, W., Zhang, S., and Witlox, F. (2020). Simulation analysis on flight delay propagation under different network configurations. *IEEE Access*, 8:103236–103244.
- Zhang, W., Zhu, F., Lv, Y., Tan, C., Liu, W., Zhang, X., and Wang, F.-Y. (2022). Adapgl: An adaptive graph learning algorithm for traffic prediction based on spatiotemporal neural networks. *Transportation Research Part C: Emerging Technologies*, 139:103659.