Information Fusion 107 (2024) 102326

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

FAST-CA: Fusion-based Adaptive Spatial–Temporal Learning with Coupled Attention for airport network delay propagation prediction

Chi Li [a,d], Xixian Qi [b], Yuzhe Yang [b], Zhuo Zeng [b], Lianmin Zhang [d], Jianfeng Mao [b,c,*]

[a] *School of Science and Engineering, The Chinese University of Hong Kong, No. 2001 Longxiang Avenue, Longgang District, Shenzhen, 518172, Guangdong, China*
[b] *School of Data Science, The Chinese University of Hong Kong, No. 2001 Longxiang Avenue, Longgang District, Shenzhen, 518172, Guangdong, China*
[c] *Guangdong Provincial Key Laboratory of Big Data Computing, The Chinese University of Hong Kong, Shenzhen, Guangdong, China*
[d] *Shenzhen Research Institute of Big Data, No. 2001 Longxiang Avenue, Longgang District, Shenzhen, 518172, Guangdong, China*

ARTICLE INFO

Keywords:
Graph neural networks
Flight delay prediction
Delay propagation
Dynamic graph
Adaptive learning

ABSTRACT

The issue of delay propagation prediction in airport networks has garnered increasing global attention, particularly due to its profound impact on operational efficiency and passenger satisfaction in modern air transportation systems. Despite research advancements in this domain, existing methodologies often fall short of comprehensively addressing the challenges associated with predicting delay propagation in airport networks, especially in terms of handling complex spatial–temporal dependencies and sequence couplings. In response to the complex challenge of predicting delay propagation in airport networks, we introduce the Fusion-based Adaptive Spatial–Temporal Learning with Coupled Attention (FAST-CA) framework. FAST-CA is an innovative model that integrates dynamic and adaptive graph learning, coupled attention mechanisms, periodicity feature extraction, and multifaceted information fusion modules. This holistic approach enables a thorough analysis of the interplay between flight departure and arrival delays and the spatial–temporal correlations within airport networks. Rigorously evaluated on two extensive real-world datasets, our model consistently outperforms current state-of-the-art baseline models, showcasing superior predictive performance and the effective learning capabilities of its intricately designed modules. Our research highlights the criticality of analyzing spatial–temporal relationships and the dynamics of flight coupling, offering significant theoretical and practical contributions to the advancement and management of air transportation systems.

1. Introduction

The burgeoning demand for air travel and the growth of intricately connected air transportation networks have precipitated heightened congestion and resultant flight delays. Over recent decades, such delays have ascended the ranks as a pivotal concern in both airport management and flight scheduling. These inefficiencies not only compromise the streamlined operations of air transportation systems but also sway passenger decisions. Within the U.S. context, the annual aggregate cost of delays is staggering, exceeding an estimated $30 billion [1]. In 2018 alone, the U.S. grappled with close to 2 million flight delays, inducing significant disruptions within its air traffic systems [2]. A predominant catalyst for these flight delays is the propagation behavior observed in preceding and succeeding flights [3]. Given the complex interdependencies inherent to air traffic networks, such delays invariably proliferate throughout the entire aviation system, creating a cascade of disruptions. Therefore, unraveling the intricacies of delay propagation within aviation networks and accurately forecasting airport delay magnitudes is of utmost importance. Insights gleaned from this research are crucial for mitigating the extensive economic impacts of delays, enhancing the operational efficiency of air transportation systems, and significantly improving passenger satisfaction.

As elucidated in [4], predicting delay propagation in airport networks is not a straightforward time series problem but rather one influenced by intricate spatial correlations and numerous external factors. Building upon the four challenges outlined in [4], we delve into a practical case study to illustrate the comprehensive range of factors impacting this issue. Furthermore, we identify and articulate five key challenges that we believe are critical in influencing the prediction of airport network delay propagation. As depicted in Fig. 1, the propagation of airport delays is influenced by a myriad of factors, including geographic proximity, weather conditions, and airline schedules. Illustratively, adverse weather conditions in the Eastern United States, due

[*] Corresponding author at: School of Data Science, The Chinese University of Hong Kong, No. 2001 Longxiang Avenue, Longgang District, Shenzhen, 518172, Guangdong, China.
E-mail address: jfmao@cuhk.edu.cn (J. Mao).

[1] In this study, we utilize International Air Transport Association (IATA) codes to represent the names of airports.

https://doi.org/10.1016/j.inffus.2024.102326
Received 30 November 2023; Received in revised form 23 February 2024; Accepted 26 February 2024
Available online 29 February 2024

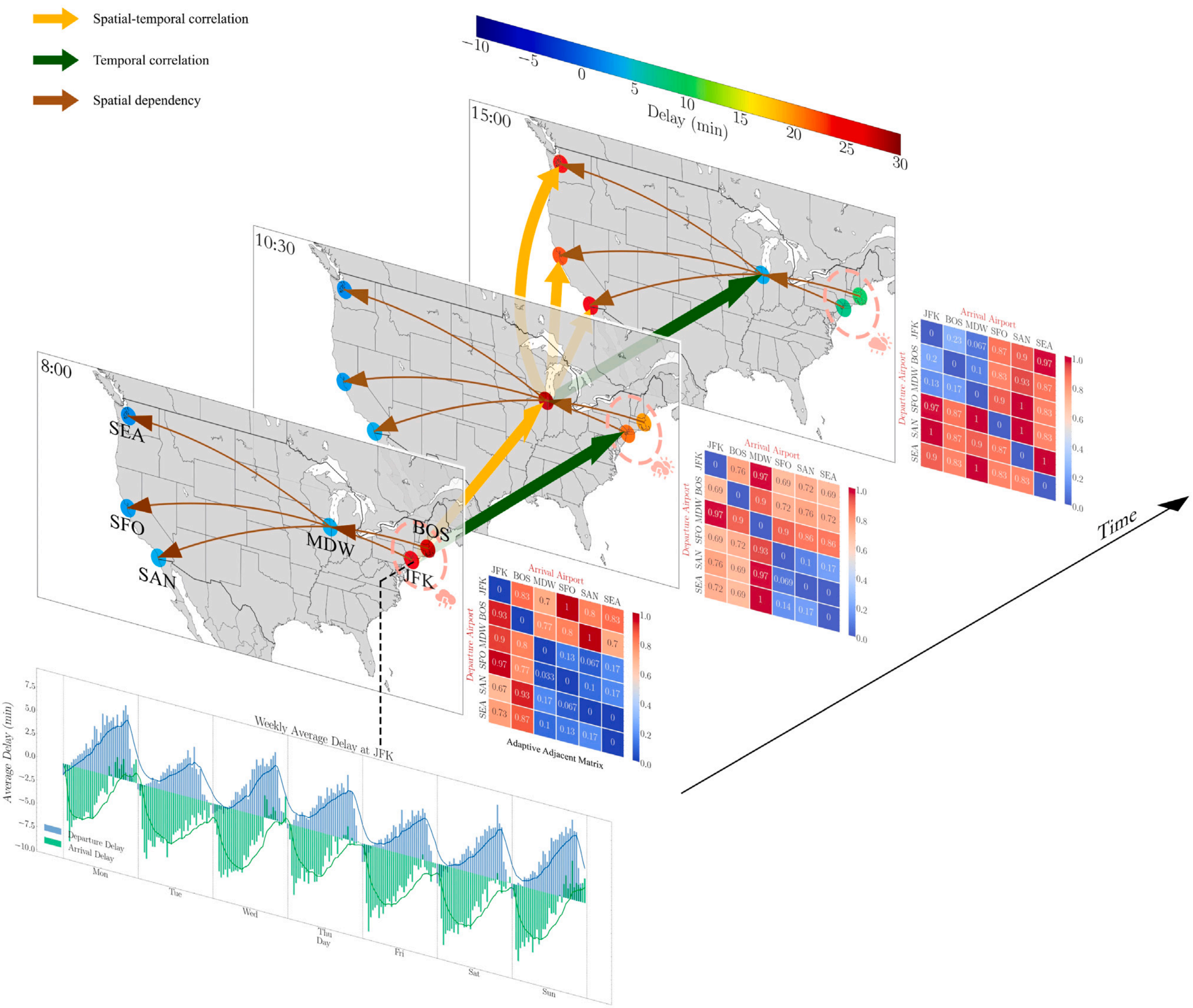


Fig. 1. Depicting the complex spatial–temporal relationships in airport network delay propagation. Circles represent six major international airports: BOS, JFK, MDW, SEA, SFO, and SAN, with varied colors indicating the level of delay at each airport. Three geographically distributed diagrams illustrate the delay status of airports at different time instances and their corresponding flight schedules, denoted by fine gray arrows. These diagrams also capture three distinct spatial–temporal dependencies prevalent during delay propagation. The heatmap portrays the pairwise relationships between airports, which dynamically evolve over time. Proximity in airport locations reveals similarities in delay levels, often influenced by shared weather conditions. At the bottom, a subplot delineates the time-series variation in both departure and arrival delays at JFK airport over a week, highlighting the interdependency and periodic nature between the two curves.

to their geographic proximity, are causing flight departure delays at JFK[1] and BOS airports. This results in a positive correlation of delays within the Eastern region. Over time, these departure delays contribute to increasing arrival delays at a central region airport, such as MDW. This cascade effect gradually extends nationwide, ultimately impacting Western coastal airports like SEA, SFO, and SAN. In Fig. 1, this progression is observable as delays move from east to west, culminating at three distinct time points. This scenario underscores five primary challenges in predicting airport network delay propagation:

- **External Influences:** Air travel operations are notably susceptible to external conditions, particularly severe weather events like thunderstorms, dense fog, and hurricanes. These elements are primary contributors to aviation delays. As Fig. 1 illustrates, a thunderstorm can significantly disrupt flight schedules.
- **Coupled Spatial–temporal Dependencies:** Delay propagation is a product of the interconnected influences of both time and location. Delays at one site can ripple through the network over time, impacting subsequent flights at the original location and other connected nodes. This phenomenon underscores the intertwined nature of space and time in aviation operations. In Fig. 1, the three arrows represent spatial–temporal correlation, temporal correlation, and spatial dependency, respectively.
- **Departure–Arrival Delay Relationship:** The interrelation between arrival and departure delays is evident, as shown by the Weekly Average Delay at JFK in Fig. 1. A delay in departure at one airport can lead to arrival delays at a connected airport, and vice versa.
- **Dynamic and Adaptive Spatial Dependencies:** The adjacency matrices depicted in Fig. 1 are dynamic, reflecting the evolving nature of airport delay propagation over time. Our model captures these intricate relationships adaptively, offering a nuanced understanding of the dynamic spatial factors influencing delays.
- **Periodicity in Airport Delays:** Airport delay time series data frequently display periodic characteristics, marked by daily and weekly patterns. This cyclical nature of delays is vividly illustrated in Fig. 1, which showcases the weekly average delay metrics at JFK airport.

Table 1
Comparison of airport delay prediction methodologies.

Feature/method	Traditional statistical	Data-driven	Deep learning	FAST-CA
Theoretical basis	Mathematical models	Statistical learning	(Graph) Neural networks	Graph neural networks
Key characteristics	Robust principles	Feature engineering	Feature learning	Fusion-based and adaptive and dynamic graph learning
Data scalability	Limited	Moderate	High	High
Temporal dynamics	Basic models	Improved with machine learning	Recurrent neural networks, attention-based	Context-aware attention
Spatial dynamics	Simplistic	Improved with machine learning	Convolutional neural networks, graph-based	Adaptive graph learning
Predictive performance	Good interpretability	Better accuracy	High potential	State-of-the-art
Computational efficiency	High	Moderate	Varies	Optimized and acceptable
Application scalability	Limited	Moderate	High	High

Given the aforementioned significance and challenges of airport network delay propagation, the past two decades have witnessed a surge in research efforts aimed at modeling this issue. These efforts have evolved from early mathematical and statistical methods to data-driven approaches, and more recently, to deep learning methodologies that specifically address spatial–temporal dependencies. Traditional statistical approaches are designed to analyze the inherent mechanisms of delay propagation and identify key factors influencing delays using mathematical models. These include queueing theory [5,6], survival models [7], and regression models [8]. While these methods typically offer robust principle-based explanations, they may encounter challenges when dealing with high-dimensional, large-scale delay propagation data, due to inherent constraints in processing complex datasets. Data-driven methods typically employ machine learning algorithms, such as random forest [9–11], to select influencing factors and represent features for predicting specific airport or network states. While effective, these approaches often rely on domain knowledge from experts for feature construction and tend to utilize shallow representations. In recent years, deep learning techniques have gained widespread adoption for modeling delay propagation in airport networks [12–14]. However, these methods often do not fully address the complex challenges involved in delay propagation modeling, such as the dynamic nature of spatial–temporal dependencies, the coupling of spatial–temporal factors, the intertwined nature of departure and arrival delay sequences, and the impact of periodic factors. In Table 1, we provide a detailed description of the comparison between various types of airport network delay prediction methods and FAST-CA across different feature dimensions.

In an effort to address the five challenges mentioned earlier and recognize the limitations of existing deep learning methodologies, we propose the Fusion-based Adaptive Spatial–Temporal Learning with Coupled Attention (FAST-CA) framework for airport network delay propagation prediction. Our approach first considers the dynamic graph characteristics and integrates weather features as inputs for the adaptive graph learning module. We then employ coupled attention mechanisms to fuse features of both temporal and spatial dependencies, as well as the interlinked departure and arrival delay sequences. Subsequently, we incorporate context-aware positional encoding combined with a self-attention mechanism to model temporal dependencies and extract periodic features. By fusing the outputs of these modules, we achieve predictions of future delay scenarios in airport networks. Tested on two types of real-world delay datasets, FAST-CA demonstrates superior performance, outperforming existing state-of-the-art baseline models. Overall, the contributions of our research can be summarized as follows:

1. Our proposed FAST-CA framework delivers an exhaustive analysis of spatial–temporal dependencies, effectively addressing the identified challenges. By employing advanced information fusion techniques, we achieve a profound understanding of the dynamics of airport network delay propagation. This comprehensive approach ensures meticulous modeling of multiple facets of airport delays, resulting in predictions that are both accurate and rich in insights.
2. Incorporating a dynamic and adaptive graph learning module, FAST-CA adeptly captures the evolving relationships between airport nodes. This module's ability to dynamically extract and adapt to complex relationships in continuously changing airport networks allows for a nuanced understanding of inter-node interactions, essential for precise delay predictions in such a dynamic setting.
3. We recognize the importance of periodic patterns and the interrelation between departure and arrival sequences in delay propagation. FAST-CA integrates these critical elements to significantly enhance prediction accuracy. By incorporating these cyclical patterns and coupled relationships, our framework excels in forecasting delays, particularly in recurring operational scenarios where traditional models may struggle.
4. Extensive testing of FAST-CA on two large-scale datasets has validated its effectiveness in predicting both arrival and departure delays. This thorough testing not only confirms the robustness of our model but also underscores its applicability and reliability in real-world settings. The model's proven efficacy in handling large-scale datasets positions it as a valuable asset for managing airport delays.

The remainder of this article is organized as follows. Section 2 presents a comprehensive review of the related works, encompassing delay propagation modeling and prediction methodologies ranging from classical statistical methods to data-driven approaches and deep learning techniques. Section 3 delves into the intricate construction of our FAST-CA framework, detailing its unique components and operational mechanisms. In Section 4, we provide an exhaustive account of the experiments conducted to validate the efficacy of our proposed model. This section also includes an ablation study and demonstrates the learning capabilities of our framework's key modules. We conclude the article in Section 5, where we reflect on our findings and explore potential avenues for future research.

2. Related works

2.1. Classical statistical methods

The problem of modeling flight delay propagation has been extensively studied over the past two decades, with the earliest research tracing back to a simple delay multiplier index [15]. This index measures the ratio between the initial delay and the summation of the downstream delay. Subsequent to the initial studies, survival models are proposed to examine patterns of flight delay and its propagation, as well as to assess the potential impact of various factors on departure and arrival delays [7]. For analyzing the network effects of delay propagation, a multivariate simultaneous regression model is introduced. This model aims to identify contributing factors and examine delay propagation interactions emanating from a single airport to the rest of the network [8]. In addition to examining delay propagation resulting from aircraft, several studies have also investigated the impacts of crew and passenger connectivity [16].

Furthermore, the queueing theory has gained popularity in modeling flight delay propagation [5,6]. In these studies, flights are considered as customers, and runways serve as servers processing the flights as departure and arrival flows. The objective is to understand how delays at one airport can propagate through the network, affecting the performance of other airports and flights, using various evaluation metrics. These methods, grounded in mathematical principles, typically offer good interpretability. However, they may struggle in dealing with large-scale datasets, high-dimensional problems, and the complexities of extensive network delay propagation.

2.2. Data-driven methods

In recent years, a variety of data-driven approaches have been employed to analyze and predict flight delays and the overall state of airport networks. In [9], the K-means clustering algorithm is integrated with the random forest method to construct temporal and spatial explanatory variables for predicting future network delay scenarios. An innovative framework that combines a deep belief network (DBN) with support vector regression (SVR) has been explored to uncover intrinsic patterns of flight delays and identify micro-level key factors influencing these delays [17]. Additionally, the concept of chained flight delay prediction has been examined by merging the strengths of mathematical models and machine learning methods [10]. This approach facilitates the analysis of large-scale datasets while capturing the intrinsic relationships between airports.

From a network perspective, studies have integrated network metrics like betweenness centrality with airport delay series to predict flight delays based on fitting performance [18]. In [19], various machine learning models have been proposed for predicting flight delay propagation and analyzing airport network dynamics. A recent study [11] introduces a spatial–temporal model that combines spatial features constructed from network metrics with temporal states to predict the status of flight delays. While these data-driven methods have achieved anticipated performance improvements, they are heavily reliant on feature engineering and expert knowledge and tend to offer only a superficial representation of feature characteristics.

2.3. Spatial–temporal methods

Graph neural networks (GNNs) have shown remarkable capabilities in traffic flow forecasting, effectively capturing and modeling the dynamic and uncertain aspects of spatial–temporal traffic flows. Within this spatial–temporal graph neural network (ST-GNN) modeling framework, GNN-based models are typically employed to extract spatial relationships between nodes, while recurrent neural networks (RNNs), temporal convolutional networks (TCNs), and self-attention mechanisms are often integrated to model the temporal dependencies in time series data.

In [20], the STGCN framework combines graph convolutional networks (GCNs) with temporal gated convolution to extract spatial–temporal dependencies in time series. ASTGCN [21] integrates an attention mechanism to learn the spatial–temporal dependencies of traffic flow, further considering periodic features like recent, daily, and weekly patterns. These early approaches to spatial–temporal modeling often rely on predefined adjacency matrices. Acknowledging network dynamics, Graph WaveNet [22] proposes the generation of adaptive adjacency matrices through node embeddings. This concept is also applied in AGCRN [23], which additionally considers heterogeneous parameter learning for nodes through matrix decomposition in a scalable manner. Following this, Ada-STNet [23] introduces a two-stage training approach for learning adaptive adjacency matrices, a methodology further extended in AdapGL [24], where the prediction network module and the graph learning module are optimized through alternate training. While these learned adjacency matrices represent an optimal measurement of node relationships, they often fail to capture the dynamic nature of actual spatial relationships due to continual temporal changes. To address this, STCGAT [25] models spatial information extraction for each node at every moment through dynamic graph inputs, coupled with TCNs for capturing temporal dependencies in traffic flow. Similarly, LATFPM [26] considers a multi-relational graph structure and dynamic graph inputs to tackle the prediction of airport arrival flow.

Moreover, as illuminated by [27], graph signal processing offers a powerful toolkit for examining flight delay propagation. In this context, airport delays can be conceptualized as node signals within a graph, facilitating the delineation and quantification of diverse spatial–temporal patterns through graph spectral analysis. As summarized in Table 2, there have been six notable publications dedicated to employing spatial–temporal graph neural networks for predicting airport network delay propagation. A deep graph-embedded LSTM model (DG-LSTM) is initially proposed to leverage a diffusion convolution kernel to encapsulate delay propagation characteristics and long short-term memory (LSTM) to unravel temporal dependencies [12]. AG2S-Net [13], a pioneering graph-to-sequence learning architecture, incorporates attention mechanisms and emphasizes adaptive adjacency matrix construction. To accommodate the time-varying and periodic nature of airport networks, MSTAGCN [14] is proposed, anchored in a meticulously designed adaptive graph convolutional block. In [19], DST-GAT is proposed to use a spatial–temporal graph attention neural network, employing the graph attention networks (GAT) to capture the dynamic adjacency matrices. Subsequently, the GOGCN model is proposed to simultaneously extract geographical and operational spatial–temporal dependencies of airport network nodes [28]. In a recent contribution, STPN is suggested in [4], a space–time separable multi-graph convolutional network framework, facilitating independent extraction of temporal and spatial dependencies within the airport delay network. Their experimental outcomes underscored state-of-the-art performance across diverse datasets and multi-step prediction horizons.

Nevertheless, as Table 2 elucidates, these studies still overlook certain practical aspects inherent in airport network delay propagation, such as the dynamic nature of inter-airport relationships and the inability of existing matrices to precisely quantify specific influence relationships, as well as the intricate coupling between departure and arrival delays. Consequently, our research introduces a more comprehensive spatial–temporal graph analysis framework to model the complex and dynamic delay propagation process within airport networks.

3. Methodology

In our methodology, as outlined in Fig. 3, the proposed FAST-CA framework integrates dynamic graph inputs with weather data using a fusion-based adaptive spatial–temporal learning approach. This framework skillfully merges spatial and temporal dynamics through an adaptive graph learning module and a dual attention mechanism, which includes both self and cross-attention components. These features equip the framework with the capability to capture complex interactions between flight departure and arrival sequences, the coupling of spatial and temporal dependencies, and the influence of weather and periodic features. Consequently, this enhances the precision of airport network delay propagation predictions.

3.1. Problem formulation

In this study, the airport network delay propagation prediction problem is conceptualized as follows. Given a set of N airports, we represent these airports as a weighted graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, encapsulating the topological structure of the airport network. Here, $\mathcal{V}$ denotes the set of airport nodes, with $|\mathcal{V}| = N$, and $\mathcal{E}$ signifies the set of connecting edges amongst all nodes within graph $\mathcal{G}$. We demonstrate the historical

Table 2
Summary of the characteristics and description of ST-GNN models for flight delay propagation prediction.

Literature	Data scope	Spatial correlation	Temporal correlation	Adjacency matrix generation	Weather	Temporal periodicity	Dynamic graph	Spatial–temporal coupling	Departure-arrival correlation
DG-LSTM [12]	U.S.	GCN	LSTM	Predefined	–	–	–	–	–
AG2S-Net [13]	China	GCN	Bi-LSTM; Attention	Predefined	✓	–	–	–	–
MSTAGCN [14]	China	GCN	R-GCN	Predefined	–	–	–	–	–
DST-GAT [19]	Europe	GAT	LSTM	Predefined	–	–	–	–	–
GOGCN [28]	China	GCN	–	Predefined	–	–	–	–	–
STPN [4]	U.S.; China	GCN	Self attention; Positional encoding	Predefined	✓	✓	–	–	✓
Our work	U.S.; China	GAT	Self attention; Context-aware positional encoding	Predefined; Adaptive learning	✓	✓	✓	✓	✓

The symbol "✓" indicates that the aspect is considered in the study, while "–" denotes that it is not considered.

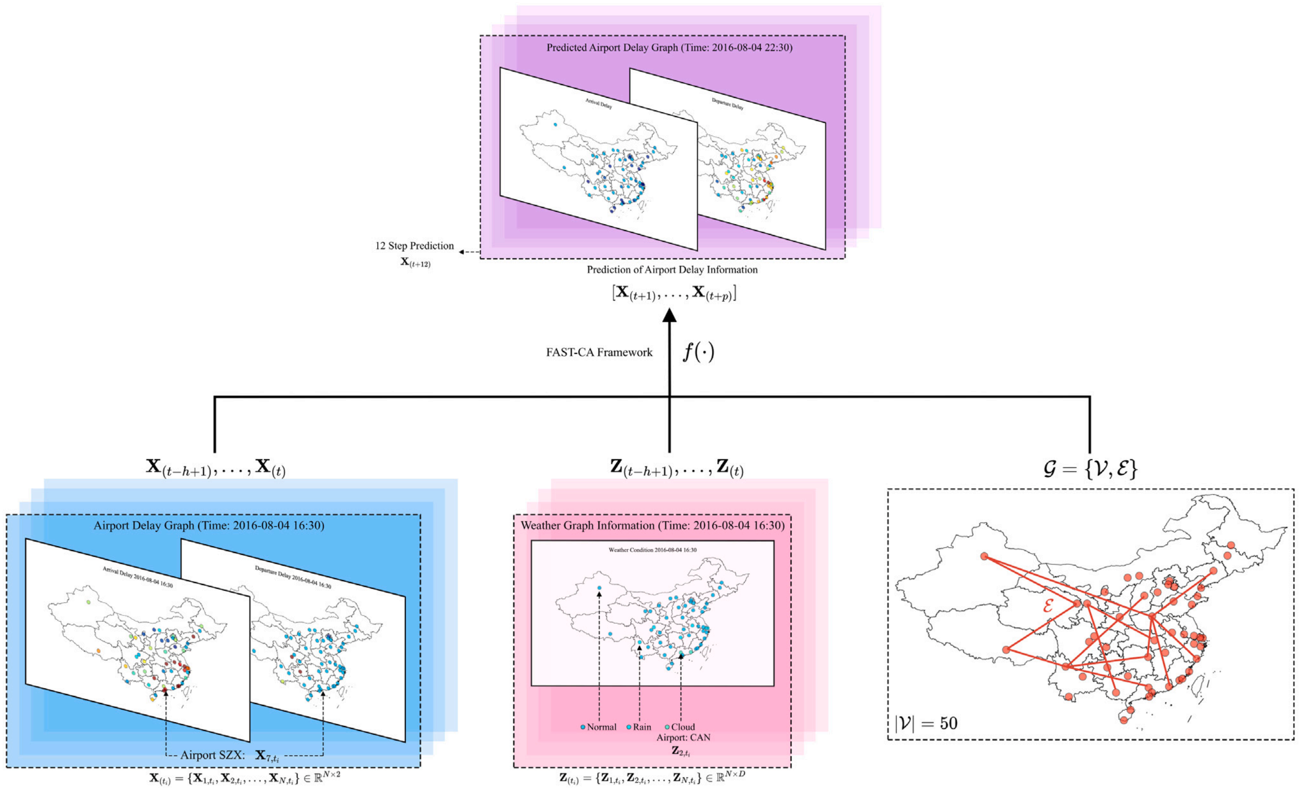

Predicted Airport Delay Graph (Time: 2016-08-04 22:30)
Arrival Delay
Departure Delay
12 Step Prediction
$\mathbf{X}_{(t+12)}$
Prediction of Airport Delay Information
$[\mathbf{X}_{(t+1)}, \dots, \mathbf{X}_{(t+p)}]$
FAST-CA Framework
$f(\cdot)$
$\mathbf{X}_{(t-h+1)}, \dots, \mathbf{X}_{(t)}$
$\mathbf{Z}_{(t-h+1)}, \dots, \mathbf{Z}_{(t)}$
$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$
Airport Delay Graph (Time: 2016-08-04 16:30)
Arrival Delay 2016-08-04 16:30
Departure Delay 2016-08-04 16:30
Airport SZX: $\mathbf{X}_{7,t_i}$
$\mathbf{X}_{(t_i)} = \{\mathbf{X}_{1,t_i}, \mathbf{X}_{2,t_i}, \dots, \mathbf{X}_{N,t_i}\} \in \mathbb{R}^{N\times 2}$
Weather Graph Information (Time: 2016-08-04 16:30)
Weather Condition 2016-08-04 16:30
● Normal ● Rain ● Cloud
Airport: CAN
$\mathbf{Z}_{2,t_i}$
$\mathbf{Z}_{(t_i)} = \{\mathbf{Z}_{1,t_i}, \mathbf{Z}_{2,t_i}, \dots, \mathbf{Z}_{N,t_i}\} \in \mathbb{R}^{N\times D}$
$\mathcal{E}$
$|\mathcal{V}| = 50$

Fig. 2. Visualization of the process of problem formulation.

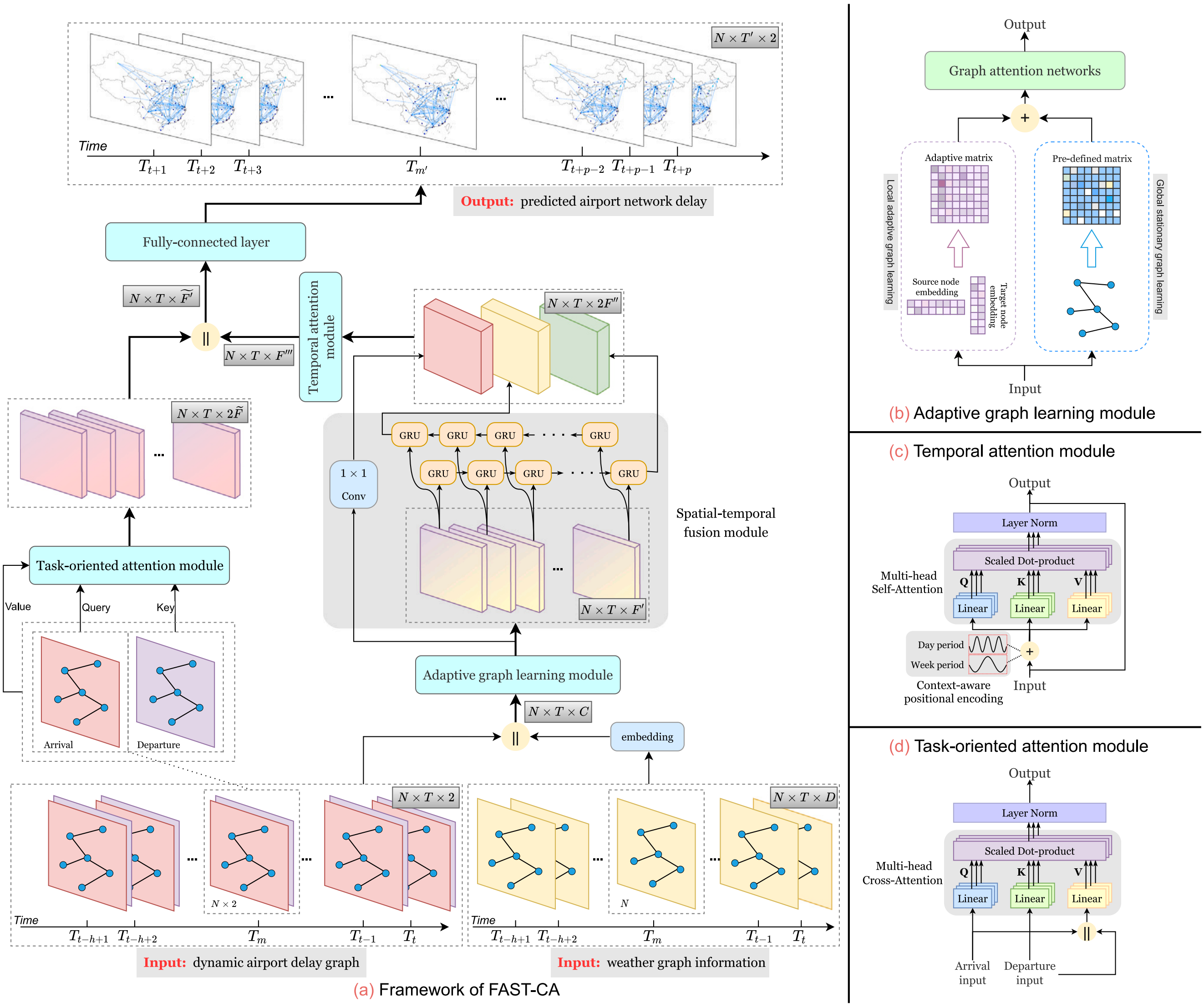
delay information of the airport network over a time span of T by a feature matrix $\mathbf{X} \in \mathbb{R}^{N\times T\times 2}$, where "2" reflects the feature dimension, accounting for both departure and arrival delay series. Specifically, arrival and departure delays are represented by vectors $\mathbf{X}_{i,j} \in \mathbb{R}^2$ indicating the delays at the airport i at time j, and covariate vectors $\mathbf{Z}_{i,j} \in \mathbb{R}^D$ indicating the type of weather at the airport i at time j, with D being the number of weather categories. Additionally, we denote $\mathbf{X}_{(t)} = \{\mathbf{X}_{1,t}, \mathbf{X}_{2,t}, \dots, \mathbf{X}_{N,t}\} \in \mathbb{R}^{N\times 2}$ and $\mathbf{Z}_{(t)} = \{\mathbf{Z}_{1,t}, \mathbf{Z}_{2,t}, \dots, \mathbf{Z}_{N,t}\} \in \mathbb{R}^{N\times D}$ as the sets of delay and weather information for all airport nodes at time t, respectively. We will present a real-world scenario to visually demonstrate the notation used in our modeling process and how the FAST-CA modules predict delay propagation. In Fig. 2, using the Chinese dataset as an example, we illustrate the graphical information at a specific moment, t_i: August 4, 2016, at 16:30. From $\mathbf{X}_{(t_i)}$ and $\mathbf{Z}_{(t_i)}$, we can also derive data regarding each airport's departure, arrival, and weather conditions at that moment. For instance, the weather condition at CAN airport at this time is noted as cloudy, denoted by Z_{2,t_i}. Based on the historical data, the model predicts future delay values, including both arrival and departure delays. The predictions for 12 timesteps later are displayed in Fig. 2. In this context, the task of predicting airport network delay propagation is summarized as follows:

$$\left[\left(\mathbf{X}_{(t-h+1)}, \dots, \mathbf{X}_{(t)}\right); \left(\mathbf{Z}_{(t-h+1)}, \dots, \mathbf{Z}_{(t)}\right); \mathcal{G}\right] \xrightarrow{f(\cdot)} \left[\mathbf{X}_{(t+1)}, \dots, \mathbf{X}_{(t+p)}\right] \quad (1)$$

Here, a function $f(\cdot)$ is devised to learn from h historical delay observations and covariates based on graph $\mathcal{G}$, aiming to predict future p delay states within the network. The subsequent section will elucidate how we utilize the proposed FAST-CA framework to model this mapping $\xrightarrow{f(\cdot)}$. For enhanced clarity, we delineate the essential notations used in the methodological description and subsequent data processing sections in Table 3.

3.2. Adaptive graph learning module

In the spatial dependence modeling, the delay states of each airport are interrelated and subject to dynamic changes. For instance, under certain severe weather conditions, widespread delays in flights from Beijing to Shanghai can exacerbate the delay states of both airports at


N × T′ × 2
Time
T_{t+1} T_{t+2} T_{t+3} T_{m′} T_{t+p−2} T_{t+p−1} T_{t+p}
Output: predicted airport network delay
Fully-connected layer
N × T × F̃′
||
N × T × F‴
Temporal attention module
N × T × 2F″
N × T × 2F̃
1 × 1 Conv
GRU
Spatial-temporal fusion module
N × T × F′
Task-oriented attention module
Value
Query
Key
Arrival
Departure
Adaptive graph learning module
N × T × C
embedding
N × T × 2
N × 2
N × T × D
N
T_{t−h+1} T_{t−h+2} T_m T_{t−1} T_t
Input: dynamic airport delay graph
Input: weather graph information
Output
Graph attention networks
+
Adaptive matrix
Pre-defined matrix
Local adaptive graph learning
Global stationary graph learning
Source node embedding
Target node embedding
Input
Layer Norm
Scaled Dot-product
Multi-head Self-Attention
Q
K
V
Linear
Day period
Week period
Context-aware positional encoding
Multi-head Cross-Attention
Arrival input
Departure input

(a) Framework of FAST-CA

(b) Adaptive graph learning module

(c) Temporal attention module

(d) Task-oriented attention module

Fig. 3. Architecture of the FAST-CA Framework. (a) This schematic illustrates the FAST-CA framework, highlighting its integration of dynamic graphs and weather data inputs. Central to the framework is the adaptive graph learning module paired with a dual attention mechanism, adeptly capturing the complex interplay between spatial–temporal factors and flight departure and arrival sequences for accurate airport network delay predictions. (b)–(d) provide detailed representations of three core modules within the FAST-CA framework: the adaptive graph learning module, the context-aware temporal attention module, and the task-oriented attention module, respectively.

Table 3
Summary of representative symbols and their descriptions.

Symbol	Description
N	The number of airport nodes in a network.
$\mathcal{G}$	The graph structure.
$\mathbf{A}^{\mathrm{glb}} \in \mathbb{R}^{N\times N}$	The global distance adjacency matrix.
$\mathbf{X} \in \mathbb{R}^{N\times T\times 2}$	The historical delay information of the airport network over a time span of T.
$\mathbf{X}^{\mathrm{arr}} \in \mathbb{R}^{N\times T\times 1}$	The network-wide arrival delay series.
$\mathbf{X}^{\mathrm{dep}} \in \mathbb{R}^{N\times T\times 1}$	The network-wide departure delay series.
$\hat{\mathbf{X}} \in \mathbb{R}^{N\times T\times 2}$	The predicted delay values of the airport network over a time span of T.
$\widetilde{\mathbf{X}} \in \mathbb{R}^{N\times T\times C}$	The fused feature vector from delay and embedded weather features.
$\mathbf{Z} \in \mathbb{R}^{N\times T\times D}$	The weather values of the airport network over a time span of T.
$\mathbf{X}_{(t)} \in \mathbb{R}^{N\times 2}$	The sets of delay information for all airport nodes at time t.
$\mathbf{Z}_{(t)} \in \mathbb{R}^{N\times D}$	The sets of weather information for all airport nodes at time t.
$E_{At} \in \mathbb{R}^{N\times d}$	The embedding dictionary for each airport node.
$W_{\mathcal{G}} \in \mathbb{R}^{d\times C\times F}$	The shared weight pool.
$\alpha_{ik}^{t,q}$	The normalized coefficient between node i and node k in qth attention group at time t.
$\widetilde{\mathbf{X}}''_{(t)} \in \mathbb{R}^{N\times F'}$	The feature vectors of all nodes at time t output by the adaptive graph learning module.
$H' \in \mathbb{R}^{N\times T\times 2F''}$	The spatial–temporal fusion module's output feature matrix over T time steps.
$H'' \in \mathbb{R}^{N\times T\times 2F''}$	The feature vector matrix over T time steps, integrated with periodic positional encoding.
$H''' \in \mathbb{R}^{N\times T\times F'''}$	The context-aware temporal attention module's output feature matrix over T time steps.
$\tilde{H} \in \mathbb{R}^{N\times T\times 2\tilde{F}}$	The task-oriented attention module's output feature matrix over T time steps.

that time. Consequently, the relationship between these two airports might become less tight due to reduced flights. However, as the weather gradually improves over time, leading to an amelioration in delays, the relationship between the two airports tightens once again, driven by the importance of the flight routes connecting them. Previous literature on modeling airport network delay propagation predominantly relied on predefined matrices [13,14], failing to account for the dynamic nature of the airport network. Moreover, multi-relational graphs constructed based on domain expertise, such as distance graphs, origin–destination graphs, and similarity graphs, have not fully captured the dynamic interplay of relationships exhibited by airport networks at different times. They also fall short in portraying the optimal relationship matrix between airport nodes [29]. To address these issues, as depicted in Fig. 3(b), our study adopts an adaptive graph learning module.

Similar to the approaches in [23,25], our proposed adaptive graph learning module first learns the embedding vectors of airport nodes to effectively discern the dynamic correlation information among these nodes at different moments, thereby generating an adaptive adjacency matrix specific to each corresponding moment. However, we recognize that while the delay states of airports, which are reflected through the node embedding vectors, dynamically influence their adjacency relationships, the distance relationships between airports are static from a global perspective and cannot be captured through adaptive learning alone. Therefore, we introduce the Adaptive-Fixed Matrix Integration (AFMI) module, which innovatively combines the adaptively generated adjacency matrix with a predefined distance matrix. This approach is designed to more comprehensively reflect the local dynamic and global stationary characteristics of the adjacency relationships between airport nodes. Through the AFMI model, we generate an adjacency matrix for each moment t, as illustrated by the following formula:

$$\tilde{A}^t = \mathrm{softmax}\left(\mathrm{ReLU}\left(E_{At} \cdot E_{At}^T\right) + A^{\mathrm{glb}}\right) \tag{2}$$

where $E_{At} \in \mathbb{R}^{N \times d}$ serves as the embedding dictionary for each airport node, with d representing the dimension of these node embeddings. The transpose form E_{At}^T, is used alongside the predefined distance matrix A^{glb}. The model incorporates the ReLU activation function for introducing non-linearity and employs softmax for normalization, ensuring well-scaled outputs.

Subsequently, we have developed a module, termed AFMI–GAT, which inputs the generated adaptive adjacency matrices at each time step into the GAT framework. This facilitates the dynamic aggregation of neighboring node features for each airport node by calculating attention coefficients, in conjunction with the node's feature vectors. It is important to note that we are not considering the original delay features of airport nodes. Given the significant impact of weather on delays and the interdependence of weather conditions across different airports, we initially embed the original weather features of each airport. These embedded weather features are then concatenated and integrated with the delay features, forming a composite feature vector $\widetilde{\mathbf{X}}_{i,j} \in \mathbb{R}^C$ for airport node i at time j. Additionally, diverging from traditional GAT networks where parameters are uniformly shared across all nodes, our approach, inspired by the node adaptive parameter learning from [23], recognizes the distinct patterns of different airport nodes. To balance model complexity and specificity, we utilize a shared weight pool $W_{\mathcal{G}} \in \mathbb{R}^{d \times C \times F}$. This weight pool dynamically interacts with each node's embedding vector E_{At}, generating unique parameter matrices $\Theta = E_{At} \cdot W_{\mathcal{G}} \in \mathbb{R}^{N \times C \times F}$ that reflect the individual characteristics of each node, without excessively enlarging the model's parameter space. Summarizing the above, the AFMI–GAT module we employ is depicted by the following formula:

$$\begin{aligned} e_{ij}^t &= \mathrm{LeakReLu}\left(\vec{a}^T\left[\left(E_{At} \cdot W_{\mathcal{G}}\right)_i \cdot \widetilde{\mathbf{X}}_{i,t} \parallel \left(E_{At} \cdot W_{\mathcal{G}}\right)_j \cdot \widetilde{\mathbf{X}}_{j,t}\right]\right) \\ \alpha_{ij}^t &= \frac{\exp\left(e_{ij}^t\right)}{\sum_{k \in \mathcal{N}_i^{\tilde{A}^t}} \exp\left(e_{ik}^t\right)} \end{aligned} \tag{3}$$

where $\vec{a} \in \mathbb{R}^{2F}$ is the weight vector, $\left(E_{At} \cdot W_{\mathcal{G}}\right)_i \in \mathbb{R}^{C \times F}$ represents the parameter matrix at node i, $\parallel$ denotes the concatenation operation, and LeakReLU is the nonlinear activation function. $\mathcal{N}_i^{\tilde{A}^t}$ is defined as the neighbor nodes of node i at time t in the generative dynamic graph $\tilde{A}^t$. e_{ij}^t and α_{ij}^t are the attention coefficients and normalized attention coefficients between node i and its neighbor node j at time t, respectively.

As the efficacy of multi-head attention in stabilizing the learning process of self-attention has been substantiated [30], we also employ this mechanism to more profoundly extract feature information pertinent to modeling spatial dependencies. Specifically, Q attention mechanisms independently carry out the transformation delineated in Eq. (3). The features extracted by each mechanism are then concatenated, culminating in the following representation of the output features:

$$\widetilde{\mathbf{X}}'_{i,t} = \parallel_{q=1}^{Q} \mathrm{LeakReLu}\left(\sum_{k \in \mathcal{N}_i^{\tilde{A}^t}} \alpha_{ik}^{t,q}\left(E_{At} \cdot W_{\mathcal{G}}^q\right)_k \cdot \widetilde{\mathbf{X}}_{i,t}\right) \tag{4}$$

where $\alpha_{ik}^{t,q}$ is the normalized coefficient computed by the attention mechanism of the qth group at time t, $\left(E_{At} \cdot W_{\mathcal{G}}^q\right)_k \in \mathbb{R}^{C \times F}$ represents the weight matrix of the corresponding group at the neighbor node k, and $\widetilde{\mathbf{X}}'_{i,t} \in \mathbb{R}^{QF}$ is the new feature representation obtained for node i at time t through the multi-head graph attention layer.

Given that the dimensionality of each node's new feature vector becomes substantially large after parallel aggregation of neighboring node information via the multi-head graph attention network, which will complicate the training process, we implement an independent self-attention mechanism layer. This layer reduces the dimensionality of each node's feature vector from $\widetilde{\mathbf{X}}'_{i,t} \in \mathbb{R}^{QF}$ to $\widetilde{\mathbf{X}}''_{i,t} \in \mathbb{R}^{F'}$ for the output of the multi-head attention layer. Upon completing the graph attention representation process for all nodes using the aforementioned method, we obtain the new feature vectors for all nodes at time t as $\widetilde{\mathbf{X}}''_{(t)} = \left\{\widetilde{\mathbf{X}}''_{1,t}, \widetilde{\mathbf{X}}''_{2,t}, \ldots, \widetilde{\mathbf{X}}''_{N,t}\right\} \in \mathbb{R}^{N \times F'}$.

To highlight the advantages of our adaptive graph learning module over existing methods in terms of spatial information modeling and comprehensive problem consideration, we will conduct a multifaceted comparison in Table 4. Our AFMI–GAT module not only accounts for the local dynamics and global stationary characteristics of distance relationships between airport nodes but also integrates external weather information, reflecting the mutual impact of weather conditions across different airports. Furthermore, the multi-head attention graph network and dynamic graph framework enhance our ability to extract complex delay propagation characteristics. The performance of our novel AFMI–GAT module, when replacing the corresponding spatial information extraction models within the STCGAT and STPN frameworks, will be evaluated in terms of efficiency, accuracy, and scalability. Detailed comparisons and analyses will be presented in the ablation study of Section 4.5.

3.3. Spatial–temporal fusion module

Considering that delays propagate through the aviation network from one airport to another, following the flight schedule connections, and given that the delay time series of different airports are interrelated, we have designed a spatial–temporal fusion module. This module is specifically tailored to model the spatial–temporal dependencies inherent in the delay propagation process. As illustrated in Fig. 4, we have replaced the gating unit in the original gate recurrent unit (GRU) structure with the AFMI–GAT framework. This adaptation allows us to extract temporal features from the delay sequences while simultaneously considering the spatial dependencies identified by the adaptive graph learning module.

Specifically, the output from the adaptive graph learning module at time t serves as the input to the GRU model. The operational workflow

Table 4
Analysis and feature summary of existing methods in spatial information modeling.

Literature	Target	Local adaptive modeling	Global stationary modeling	External features fusion	Dynamic graph structure	Spatial modeling methods	Distinct nodel parameters
AG2S-Net [13]	Airport network delay prediction	–	✓	–	–	GCN	–
MSTAGCN [14]	Airport network delay prediction	–	✓	–	–	GCN	–
AGCRN [23]	Traffic flow prediction	✓	–	–	–	GCN	✓
STCGAT [25]	Traffic flow prediction	✓	–	–	✓	Multi-head GAT	✓
STPN [4]	Airport network delay prediction	–	✓	✓	–	GCN	–
FAST-CA	Airport network delay prediction	✓	✓	✓	✓	Multi-head GAT	✓

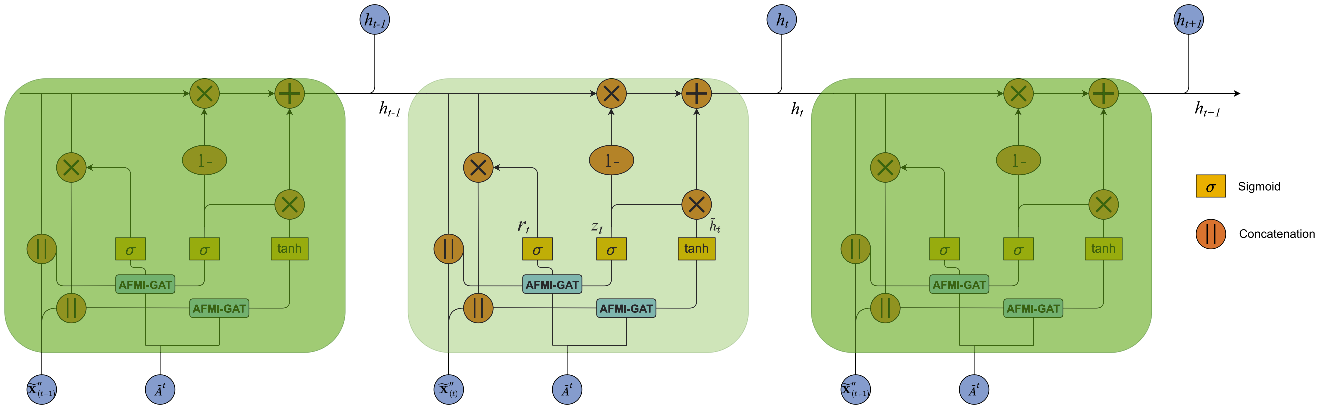


Fig. 4. The illustration of the spatial–temporal fusion module.

of our spatial–temporal fusion module is delineated as follows in the equations:

$$\begin{aligned}
z_t &= \sigma\left(\tilde{A}^t\left[\tilde{\mathbf{X}}''_{(t)}, h_{t-1}\right]\left(E_{At}\cdot W_{\mathcal{G}}^{z}\right)\right)\\
r_t &= \sigma\left(\tilde{A}^t\left[\tilde{\mathbf{X}}''_{(t)}, h_{t-1}\right]\left(E_{At}\cdot W_{\mathcal{G}}^{r}\right)\right)\\
\tilde{h}_t &= \tanh\left(\tilde{A}^t\left[\tilde{\mathbf{X}}''_{(t)}, r_t\odot h_{t-1}\right]\left(E_{At}\cdot W_{\mathcal{G}}^{\tilde{h}_t}\right)\right)\\
h_t &= z_t\odot h_{t-1} + \left(1-z_t\right)\odot\tilde{h}_t
\end{aligned} \tag{5}$$

where h_{t-1} is the output at the previous moment, $\tilde{h}_t$ is the candidate hidden layer state, $[\cdot]$ represents the concatenation operation in the feature dimension, and $\odot$ denotes element-wise multiplication. In addition, $E_{At}, W_{\mathcal{G}}^{z}$, $W_{\mathcal{G}}^{r}$, and $W_{\mathcal{G}}^{\tilde{h}_t}$ are the learnable parameters, and $h_t \in \mathbb{R}^{N\times F''}$ is the output at time t. Upon completing the described procedures over T time steps, we successfully generate sequence data that encapsulates the fused spatial–temporal dependencies. This data is represented as $H \in \mathbb{R}^{N\times T\times F''}$, effectively capturing the complex interactions within the given time frame.

Moreover, drawing insights from [25,31], we have learned that in many traffic data modeling scenarios, the data relationships extend beyond mere sequences to encompass complex contextual dynamics. Therefore, adopting a similar approach, we utilize a bidirectional GRU framework to learn the intricate spatial–temporal dependencies in the airport network delay propagation process. The reverse operation mirrors the aforementioned steps, and we concatenate the outputs of the forward and reverse GRU processes to obtain the final output $H' \in \mathbb{R}^{N\times T\times 2F''}$.

3.4. Context-aware temporal attention module

In the FAST-CA framework, the context-aware temporal attention module is a key element, functioning on the outputs from the spatial–temporal fusion module. This module initially utilizes context-aware positional encoding, infusing the representation with daily and weekly periodic features. This enhancement significantly boosts the model's proficiency in capturing temporal dependencies inherent in time series data. These encodings are amalgamated to create an exhaustive time representation, subsequently processed through the self-attention mechanism to yield the final output representation.

As demonstrated in Fig. 1, we observe distinct daily and weekly periodic characteristics in the time series of both departure and arrival delays at airports. Therefore, diverging from the classical positional encoding method proposed in [32], we distinguish between daily and weekly periodic positional information. This differentiation allows for a deeper semantic understanding of the time series data. The context-aware positional encoding is delineated in the following formula:

$$\begin{cases}
PE^{d}\left(t_d, 2s_d\right) = \sin\left(\dfrac{t_d}{10000^{2s_d/d_{\text{model}}}}\right)\\
PE^{d}\left(t_d, 2s_d+1\right) = \cos\left(\dfrac{t_d}{10000^{2s_d/d_{\text{model}}}}\right)
\end{cases} \tag{6}$$

$$\begin{cases}
PE^{w}\left(t_w, 2s_w\right) = \sin\left(\dfrac{t_w}{10000^{2s_w/d_{\text{model}}}}\right)\\
PE^{w}\left(t_w, 2s_w+1\right) = \cos\left(\dfrac{t_w}{10000^{2s_w/d_{\text{model}}}}\right)
\end{cases} \tag{7}$$

where PE^{d} and PE^{w} represent the daily and weekly positional encodings, respectively. $t_d \in \{0,1,\ldots,T_d-1\}$ denotes the time of day, with T_d being the maximum daily time, determined by the temporal resolution of the delay data. Similarly, $t_w \in \{0,1,\ldots,T_w-1\}$ indicates the time in the week, and T_w is the maximum weekly time, also determined by the temporal resolution of the delay data. The scales s_d and s_w fall within the range $[0, d_{\text{model}}/2-1]$. It is important to note that the dimension of

d_{model} is equal to F''. Finally, we concatenate the outputs of the two types of positional encodings to obtain $PE = PE^d \parallel PE^w \in \mathbb{R}^{T\times 2F''}$, which is then merged with the output H' from the spatial–temporal fusion module. This results in a temporally representative sequence $H'' \in \mathbb{R}^{N\times T\times 2F''}$ with integrated periodic features, as shown in the following formula:

$$H'' = H' + PE \tag{8}$$

Multi-head attention operates by concurrently learning various pattern dependencies using multiple sets of queries, keys, and values, with each set functioning as an independent attention head. The learned relationships from these multiple heads are then concatenated to form the output. In our approach, we employ a self-attention mechanism targeting the temporal feature representation H'' to construct the multi-head attention module. The formulation of the multi-head self-attention module is shown below:

$$\begin{aligned} &\text{MHSelfAtt} = \text{Concat}\left(\text{head}_1, \dots, \text{head}_{h_1}\right) \\ &\text{head}_i = \text{Att}\left(Q_i, K_i, V_i\right) = \text{softmax}\left(\frac{Q_i K_i^{\mathrm{T}}}{\sqrt{d_1}}\right) V_i \end{aligned} \tag{9}$$

where $Q_i = H'' W_i^Q \in \mathbb{R}^{N\times T\times d_1}$, $K_i = H'' W_i^K \in \mathbb{R}^{N\times T\times d_1}$, and $V_i = H'' W_i^V \in \mathbb{R}^{N\times T\times d_1}$. The matrices $W_i^Q \in \mathbb{R}^{2F''\times d_1}$, $W_i^K \in \mathbb{R}^{2F''\times d_1}$, and $W_i^V \in \mathbb{R}^{2F''\times d_1}$ are the learned weights. The parameter h_1 represents the number of heads in this multi-head self-attention mechanism, and we have $d_1 \times h_1 = F'''$, where F''' is the final output dimension of our context-aware temporal attention module. After completing the above calculations, the final output of the module is obtained as $H''' \in \mathbb{R}^{N\times T\times F'''}$.

3.5. Task-oriented attention module

In the process of flights taking off and landing, an aircraft follows a pre-determined flight itinerary, executing multiple flights and thereby propagating delays along this flight chain. Consequently, departure delays at a specific airport can lead to arrival delays at airports connected via the flight schedule, and vice versa. Moreover, as our study focuses on simultaneously predicting average departure and arrival delays across the airport network, we have innovatively designed a task-oriented attention module. This module is tailored to more effectively integrate the coupling relationship between departure and arrival delay sequences.

Specifically, our proposed task-oriented attention module is implemented via a multi-head cross-attention mechanism, as illustrated in the following formula:

$$\begin{aligned} &\text{MHTaskAtt} = \text{Concat}\left(\widetilde{\text{head}}_1, \dots, \widetilde{\text{head}}_{h_2}\right) \\ &\widetilde{\text{head}}_i = \text{Att}\left(\tilde{Q}_i, \tilde{K}_i, \tilde{V}_i\right) = \text{softmax}\left(\frac{\tilde{Q}_i \tilde{K}_i^{\mathrm{T}}}{\sqrt{d_2}}\right) \tilde{V}_i \end{aligned} \tag{10}$$

where $\tilde{Q}_i = \mathbf{X}^{\text{arr}} \tilde{W}_i^Q \in \mathbb{R}^{N\times T\times d_2}$, $\tilde{K}_i = \mathbf{X}^{\text{dep}} \tilde{W}_i^K \in \mathbb{R}^{N\times T\times d_2}$, and $\tilde{V}_i = \tilde{Q}_i \parallel \tilde{K}_i \in \mathbb{R}^{N\times T\times 2d_2}$. The matrices $\tilde{W}_i^Q \in \mathbb{R}^{1\times d_2}$ and $\tilde{W}_i^K \in \mathbb{R}^{1\times d_2}$ are the learned weights. $\mathbf{X}^{\text{arr}} \in \mathbb{R}^{N\times T\times 1}$ and $\mathbf{X}^{\text{dep}} \in \mathbb{R}^{N\times T\times 1}$ represent the corresponding arrival and departure delay sequences within the airport network. The parameter h_2 denotes the number of heads in this multi-head cross-attention mechanism, and we have $2d_2 \times h_2 = 2\tilde{F}$, where $2\tilde{F}$ is the final output dimension of our task-oriented attention module. After completing the above calculations, the final output of the module is obtained as $\tilde{H} \in \mathbb{R}^{N\times T\times 2\tilde{F}}$.

3.6. Fully-connected layer

Finally, we concatenate the outputs from the context-aware attention module H''' and the task-oriented attention module $\tilde{H}$ along the feature dimension to obtain the integrated sequence representation $\tilde{H}' = H''' \parallel \tilde{H} \in \mathbb{R}^{N\times T\times \tilde{F}'}$. Subsequently, $\tilde{H}'$ is fed into a two-layer fully connected network, culminating in the final output of our model. The formula is shown below:

$$\hat{\mathbf{X}} = W_2 \cdot \varphi\left(W_1 \cdot \tilde{H}' + b_1\right) + b_2 \tag{11}$$

where $\hat{\mathbf{X}} \in \mathbb{R}^{N\times T'\times 2}$ represents the predicted future T' time steps of airport network departure and arrival delay sequences. The parameters W_1, W_2, b_1, and b_2 are learnable weights and biases in the model. $\varphi()$ is the activation function. Afterward, our model is trained and optimization is performed using the L1 loss function in order to minimize the error between the predicted values $\hat{\mathbf{X}}$ and the labeled values $\mathbf{X}$:

$$\mathcal{L} = \frac{1}{N \cdot T' \cdot 2} \sum_{i=1}^{N} \sum_{j=1}^{T'} \sum_{c=1}^{2} \left| \mathbf{X}_{i,j,c} - \hat{\mathbf{X}}_{i,j,c} \right| \tag{12}$$

4. Experiments

4.1. Datasets

To assess the model's generalization ability under various conditions, we evaluate its performance on two public datasets: the U.S. dataset and the China dataset, as referenced in [4]. Each dataset comprises two components: delay information and weather data. The U.S. delay dataset, sourced from the U.S. Bureau of Transportation Statistics,[2] includes flight records from January 1, 2015, to December 31, 2021, across 360 airports. In our analysis, we select only 70 high-capacity airports to minimize outlier data from remote locations. The U.S. weather dataset, adopted from [33], classifies weather into eight categories based on specific thresholds: normal, severe cold, fog, hail, rain, snow, storm, and other precipitation. The China delay and weather dataset, obtained from Xiecheng,[3] encompasses delay and weather information from a network of Chinese airports from April 30, 2015, to May 1, 2017. We focus on 50 airports with higher traffic volumes. This dataset categorizes weather into seven types: thunderstorms, showers, torrential rain, heavy rain, light rain, cloudy, sunny, and fog. To align the data more closely with peak flight hours, we exclude flight records between 12 A.M. and 6 A.M. from both datasets.

We employ a window size of 30 min and aggregate flight records into corresponding time slots based on their originally scheduled take-off and landing times. Similar to the data preprocessing in [4], we impose a maximum delay limit of 30 min. Let $\mathbf{X}^{\text{arr}} \in \mathbb{R}^{N\times T\times 1}$ represent the arrival values and $\mathbf{X}^{\text{dep}} \in \mathbb{R}^{N\times T\times 1}$ represent the departure values after aggregation from the raw data. They are defined as follows:

$$\begin{aligned} \mathbf{X}_{i,j}^{\text{dep}} &= \frac{\sum_{i\in\mathbf{V}} \sum_{j\in\mathbf{S}} \min\left(\mathbf{r}_{i,j}^{\text{dep}}, 30\right)}{|\mathbf{V} \cap \mathbf{S}|} \\ \mathbf{X}_{i',j'}^{\text{arr}} &= \frac{\sum_{i'\in\mathbf{V}'} \sum_{j'\in\mathbf{S}'} \min(\mathbf{r}_{i',j'}^{\text{arr}}, 30)}{|\mathbf{V}' \cap \mathbf{S}'|} \end{aligned} \tag{13}$$

Here, $\mathbf{r}_{i,j}^{\text{dep}}$ and $\mathbf{r}_{i',j'}^{\text{arr}}$ represent the departure and arrival delay of a flight departing from airport i at time j and arriving at airport i' at time j'. $\mathbf{V}$ and $\mathbf{V}'$ are the sets of flights departing from and arriving at airport v, respectively, while $\mathbf{S}$ and $\mathbf{S}'$ are the sets of flights departing from and arriving during the timestamps $[\mathbf{s}, \mathbf{s} + 30)$. During the calculation of aggregated delay information, two issues arise. Firstly, we exclude all records with missing values or unusual statuses, such as flight cancellations. Secondly, each flight record is split into two independent data ($\mathbf{r}_{i,j}^{\text{dep}}$ and $\mathbf{r}_{i',j'}^{\text{arr}}$). Consequently, a single flight may be included in the arrival delay calculations but excluded from the departure delay calculations if it departs before 12 P.M. and arrives after that time. Additionally, it is important to note that negative delays are present in the data, as some flights may depart or arrive earlier than scheduled.

[2] https://www.transtats.bts.gov/DL_SelectFields.aspx?gnoyr_VQ=FGJ&QO_fu146_anzr=b0-gvzr

[3] https://pan.baidu.com/s/1dEPyMGh

It is crucial to note that both datasets, particularly the China dataset, contain a substantial number of missing values, which significantly increases the complexity of the prediction task.

The adaptive graph learning module is composed of two key components: an adaptive matrix derived from node embeddings and a predefined matrix obtained from the input. Similar to other spatial–temporal prediction models with adjacency matrices [4], the predefined matrix can be computed using a Gaussian kernel based on the distances between airports. Let $\mathbf{d}_{i,j}$ represent the distance between airport i and airport j. Then, the global adjacency matrix $\mathbf{A}^{\text{glb}} \in \mathbb{R}^{N\times N}$ can be computed as follows:

$$\mathbf{A}^{\text{glb}}_{i,j} = \begin{cases} \exp\left(-\frac{\mathbf{d}_{i,j}{}^2}{\sigma^2}\right) & \text{if } \mathbf{A}^{\text{glb}}_{i,j} > 0.1 \\ 0 & \text{if } \mathbf{A}^{\text{glb}}_{i,j} \le 0.1 \end{cases} \tag{14}$$

4.2. Evaluation metrics

In all experiments, we predict network-wide arrival and departure delays using the China dataset. We employ Mean Absolute Error (*MAE*) and Root Mean Square Error (*RMSE*) as our evaluation metrics. Let $\widehat{\mathbf{X}} \in \mathbb{R}^{N\times T'\times 2}$ denote the predicted network-wide airport delay over a time span, while $\mathbf{X} \in \mathbb{R}^{N\times T'\times 2}$ denote the observed values. These metrics are defined as follows:

$$\begin{aligned} RMSE(\mathbf{X},\widehat{\mathbf{X}}) &= \sqrt{\frac{1}{N\cdot T'\cdot 2}\sum_{i=1}^{N}\sum_{j=1}^{T'}\sum_{c=1}^{2}\left(\mathbf{X}_{i,j,c}-\widehat{\mathbf{X}}_{i,j,c}\right)^2} \\ MAE(\mathbf{X},\widehat{\mathbf{X}}) &= \frac{1}{N\cdot T'\cdot 2}\sum_{i=1}^{N}\sum_{j=1}^{T'}\sum_{c=1}^{2}\left|\mathbf{X}_{i,j,c}-\widehat{\mathbf{X}}_{i,j,c}\right| \end{aligned} \tag{15}$$

4.3. Experimental setups

To evaluate the predictive accuracy of our proposed model and other models, we conduct multi-step ahead forecasting on both datasets. Each dataset is partitioned into three subsets: a 60% training set, a 20% validation set, and a remaining 20% test set. Following the setup used in [4], we adopt different time span configurations for the two datasets. The China dataset, which has more missing values and requires a larger input window, utilizes a 36-time-point historical sequence to predict the subsequent 12-time-point future sequence. In contrast, the U.S. dataset employs a 12-time-point historical sequence to predict the subsequent 12-time-point future sequence.

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

- HA: This model utilizes historical delay data, taking its mean value for prediction purposes.
- VAR [34]: It captures the dynamic behavior of temporal patterns, offering superior forecasts based on potential future trajectories of the delay.
- ARIMA [35]: This model employs varying parameters across different locations and times to provide short-term traffic data forecasts.
- SVR [36]: It identifies delay patterns by using a kernel function to map non-linear delay data onto an optimal hyperplane in high-dimensional space.
- GAT [30]: It captures spatial dependencies through a cross-attention mechanism between different nodes.
- GRU [37]: This model leverages advanced recurrent units to represent both short-term and long-term temporal dependencies effectively.
- ASTGCN [21]: This model extracts multi-level periodic patterns by integrating a spatial–temporal attention mechanism with convolution techniques.
- STCGAT [25]: This model generates spatial adjacency subgraphs using node embedding techniques in a step-wise manner, dynamically modeling a changing graph.
- STPN [4]: It employs a multi-graph convolution model alongside a self-attention mechanism to predict delays in large-scale airport networks.

The baseline models are open-sourced with optimized hyper-parameters. Our model operates on the PyTorch 1.13.1 framework. Key settings include a batch size of 64 and a hidden dimension for the GRU cell set at 192. Other critical hyper-parameters, such as the node embedding dimension, the weather embedding dimension, and the number of attention heads in the temporal module and the task-oriented modules, are detailed in Fig. 8. Training is executed using the Adam optimizer with an initial learning rate of 0.001. The best model is selected based on minimizing the MAE value.

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).

4.4. Main results and analysis

The performance of our proposed model and other baseline models are shown in Table 5 for the China dataset and Table 6 for the U.S. dataset. We compare their performance in a 3-step (1.5 h), 6-step (3 h), and 12-step (6 h) length that reflects learning ability on short-term and long-term patterns. Overall, our model demonstrates the lowest values for both MAE and RMSE on China and U.S. datasets, underscoring its superior predictive accuracy compared to actual values. Looking deeper into the result, we can make several observations:

1. Models utilizing spatial–temporal approaches, including ASTGCN, STPN, STCGAT, and FAST-CA, have demonstrated significant superiority over conventional statistical and machine learning models in performance. These spatial–temporal models excel at capturing the intrinsic spatial and temporal dynamics present in delay data. In contrast, other models typically concentrate on either spatial or temporal features exclusively, which limits their overall effectiveness in prediction.
2. In the U.S. dataset, an unexpected outcome is observed where the GRU model surpasses some spatial–temporal models, a phenomenon not mirrored in the China dataset. This discrepancy may stem from the more distinctly defined spatial correlations within the U.S. data, which GRU effectively exploits to discern spatial–temporal relationships. Conversely, the complexity and diversity of spatial–temporal patterns in the China dataset render spatial–temporal models more adaptable, and thus more efficacious.
3. In our study, FAST-CA emerges as the most effective among the spatial–temporal models, outperforming STPN,[4] which itself exhibits superior performance over STCGAT and ASTGCN. STPN's efficacy is attributed to its integration of multi-graph convolution, self-attention mechanisms, and advanced feature extraction, which collectively enable it to comprehensively capture spatial–temporal dependencies in complex, large-scale aviation networks. This integration significantly bolsters its understanding of intricate network characteristics and adaptability. Conversely, STCGAT, despite emphasizing node adaptability and dynamic graph modeling, encounters challenges in managing the myriad of factors inherent in airport networks. ASTGCN,

[4] While using identical datasets, STPN results presented in our table differ from those in the original paper. This variation is attributable to differences in the dataset split ratio, batch size, and evaluation metrics used in our experimental setup.

Table 5
Results on the China delay dataset.

		1.5 h		3 h		6 h	
	Method	*MAE*	*RMSE*	*MAE*	*RMSE*	*MAE*	*RMSE*
Arrival delay	HA	10.720(+3.237)	13.123(+3.072)	10.720(+2.351)	13.123(+2.148)	10.720(+1.678)	13.123(+1.502)
	VAR	9.300(+1.817)	11.621(+1.570)	9.819(+1.450)	12.136(+1.161)	10.571(+1.529)	12.848(+1.227)
	ARIMA	11.085(+3.602)	13.239(+3.188)	11.032(+2.663)	13.185(+2.210)	11.099(+2.057)	13.263(+1.642)
	SVR	9.904(+2.421)	12.457(+2.406)	10.315(+1.946)	12.871(+1.896)	10.528(+1.486)	13.113(+1.492)
	GAT	9.314(+1.831)	11.969(+1.918)	9.552(+1.183)	12.220(+1.245)	9.986(+0.944)	12.629(+1.008)
	GRU	9.144(+1.661)	11.769(+1.718)	9.419(+1.050)	12.070(+1.095)	9.860(+0.818)	12.455(+0.834)
	ASTGCN	8.912(+1.429)	11.601(+1.550)	9.128(+0.759)	11.898(+0.923)	9.465(+0.423)	12.184(+0.563)
	STCGAT	8.468(+0.985)	11.209(+1.158)	8.776(+0.407)	11.527(+0.552)	9.154(+0.112)	11.898(+0.277)
	STPN	8.214(+0.731)	10.695(+0.644)	8.797(+0.428)	11.347(+0.372)	9.348(+0.306)	11.988(+0.367)
	FAST-CA	**7.483**	**10.051**	**8.369**	**10.975**	**9.042**	**11.621**
Departure delay	HA	10.441(+2.205)	12.929(+1.995)	10.441(+1.978)	12.929(+1.611)	10.441(+1.633)	12.929(+1.295)
	VAR	9.718(+1.482)	12.065(+1.861)	10.324(+1.861)	12.635(+1.317)	10.918(+2.110)	13.160(+1.526)
	ARIMA	11.099(+2.863)	13.263(+2.329)	11.593(+3.130)	13.658(+2.340)	11.639(+2.831)	13.687(+2.053)
	SVR	10.185(+1.949)	12.805(+1.871)	10.474(+2.011)	13.128(+1.810)	10.620(+1.812)	13.308(+1.674)
	GAT	9.735(+1.499)	12.506(+1.572)	9.904(+1.441)	12.771(+1.453)	10.761(+1.953)	13.426(+1.792)
	GRU	9.311(+1.075)	12.037(+1.103)	9.574(+1.111)	12.337(+1.019)	9.940(+1.132)	12.747(+1.113)
	ASTGCN	9.151(+0.915)	11.870(+0.936)	9.354(+0.891)	12.069(+0.751)	9.609(+0.801)	12.317(+0.683)
	STCGAT	8.291(+0.055)	11.197(+0.263)	8.546(+0.083)	11.469(+0.151)	8.871(+0.063)	11.792(+0.158)
	STPN	8.656(+0.420)	11.205(+0.271)	8.858(+0.395)	11.444(+0.126)	9.112(+0.304)	11.756(+0.122)
	FAST-CA	**8.236**	**10.934**	**8.463**	**11.318**	**8.808**	**11.634**

reliant mainly on shallow attention mechanisms, demonstrates relatively less proficiency in capturing complex spatial–temporal patterns. In the subsequent sections, we will delve into the distinct advantages of FAST-CA in handling the complexities of spatial–temporal dynamics within aviation networks.

4. A standout feature of FAST-CA is its adaptive learning capacity for generating adjacency matrices, a significant departure from the static, predefined matrices utilized in STPN and similar models. This distinctive capability not only strengthens the theoretical framework of our approach but is also substantiated by extensive numerical analyses. Our findings demonstrate consistent results across datasets from both the U.S. and China, with particularly pronounced performance enhancements observed in the China dataset. In terms of short-term predictions, FAST-CA exhibits remarkable improvements in the China dataset when compared to STPN, the current benchmark model. Employing an identical partition ratio and batch size, our model achieves a reduction in MAE by 9.2% for the 3-step arrival delay forecast and 4.8% for the 3-step departure delay forecast. Conversely, in the U.S. delay dataset, the predictive edge of FAST-CA is less pronounced, showing a relative decrease in MAE of less than 1% for the 3-step arrival delay forecast and 4.2% for the 3-step departure delay forecast. This capability proves especially advantageous in addressing challenges posed by significant portions of missing values, as observed in the China dataset. This aspect highlights FAST-CA's robustness and adaptability in diverse data environments.

In summary, our FAST-CA model showcases exceptional performance in both the China and U.S. datasets, outperforming traditional approaches and several existing spatial–temporal models. The particularly significant improvements observed in the China dataset underscore the model's preeminence in spatial–temporal prediction tasks. This performance highlights FAST-CA's robustness and versatility, making it a valuable tool for complex delay prediction scenarios in diverse geographical contexts.

As depicted in Fig. 5, our study not only demonstrates the efficacy of the FAST-CA model in predicting arrival delays across datasets from the United States and China but also conducts a comparative analysis with four other models: STPN, ASTGCN, GAT, and GRU. FAST-CA adeptly merges spatial and temporal dynamics via an adaptive graph learning module complemented by a dual attention mechanism, which includes both self and cross-attention. These features empower FAST-CA to discern complex interactions among flight departure and arrival sequences, alongside considering the impact of external factors such as weather conditions, thereby significantly enhancing the precision of predictions in airport network delay propagation.

In Fig. 5, each visualization step, representing a 30 minute interval and covering a total of 160 time steps, showcases FAST-CA's superior data fitting capabilities across all four airport cases. The model's efficacy is particularly marked in two categories of airports within each dataset: those with substantial delays and those with prevalent missing data. The airports include:

- Boston Logan International Airport (BOS)
- Will Rogers World Airport (OKC)
- Shanghai Hongqiao International Airport (SHA)
- Beijing Nanyuan Airport (NAY)

Notably, FAST-CA demonstrates exceptional predictive accuracy, particularly at operationally complex airports such as BOS and SHA. In contrast to GRU, which focuses solely on time, and GAT, which considers only spatial relationships, FAST-CA's adaptive graph learning module dynamically refines the graph structure, capturing nuanced spatial–temporal dependencies among flights. Furthermore, its dual attention mechanism enhances the model's ability to interpret complex interrelations across different time points and spatial locations in the time series. This multidimensional capability enables FAST-CA to accurately capture fluctuations, even amidst significant data variability. It outperforms the unidimensional focus of models like GRU and GAT, offering a more robust solution than other spatial–temporal models like ASTGCN and STPN.

Moreover, compared to road traffic datasets, the airport delay dataset presents a more complex and erratic nature, amplifying the research challenges. FAST-CA, with its advanced learning mechanisms, demonstrates remarkable adaptability and robustness, especially in handling airports with significant instances of missing data, such as OKC and NAY. This adaptability, surpassing the limitations inherent in models like GRU and GAT, and even outperforming ASTGCN and STPN, plays a pivotal role in FAST-CA's ability to effectively manage missing data and identify intricate delay patterns. Thus, in scenarios with extensive missing data and severe airport delays, FAST-CA maintains higher accuracy, showcasing its advanced technology in managing complex situations and incomplete data.

In Figs. 6 and 7, we illustrate spatial visualizations of the predictions generated by the FAST-CA, STPN, and ASTGCN models across 12 time steps for datasets from China and the U.S. Each circle on the map denotes an airport's location, with its delay time represented

Table 6
Results on the U.S. delay dataset.

		1.5 h		3 h		6 h	
	Method	*MAE*	*RMSE*	*MAE*	*RMSE*	*MAE*	*RMSE*
Arrival delay	HA	9.089(+2.470)	11.847(+2.191)	9.089(+1.778)	11.847(+1.859)	9.089(+1.507)	11.847(+1.571)
	VAR	7.795(+0.753)	10.468(+0.812)	8.123(+0.812)	10.824(+0.836)	8.479(+0.897)	11.237(+0.961)
	ARIMA	10.508(+3.466)	13.894(+4.238)	10.481(+3.170)	13.863(+3.875)	10.599(+3.017)	14.020(+3.744)
	SVR	8.175(+1.133)	10.947(+1.291)	8.487(+1.176)	11.271(+1.283)	8.736(+1.154)	11.560(+1.284)
	GAT	7.595(+0.553)	10.222(+0.566)	7.856(+0.545)	10.492(+0.504)	8.337(+0.755)	10.995(+0.719)
	GRU	7.768(+0.726)	9.981(+0.325)	7.986(+0.675)	10.659(+0.671)	8.371(+0.789)	11.035(+0.759)
	ASTGCN	7.396(+0.354)	10.130(+0.474)	7.557(+0.246)	10.316(+0.660)	8.011(+0.429)	10.727(+0.451)
	STCGAT	7.416(+0.374)	10.065(+0.409)	7.648(+0.337)	10.292(+0.304)	7.874(+0.292)	10.507(+0.231)
	STPN	7.077(+0.035)	9.743(+0.087)	7.333(+0.022)	10.061(+0.073)	7.669(+0.087)	10.461(+0.185)
	FAST-CA	**7.042**	**9.656**	**7.311**	**9.988**	**7.582**	**10.276**
Departure delay	HA	6.519(+1.916)	8.631(+1.757)	6.519(+1.824)	8.631(+1.668)	6.519(+1.629)	8.631(+1.532)
	VAR	5.561(+0.958)	7.656(+0.782)	5.817(+1.122)	7.925(+0.962)	6.165(+1.275)	8.304(+1.205)
	ARIMA	7.607(+3.004)	10.549(+3.675)	7.587(+2.892)	10.551(+3.588)	7.653(+2.763)	10.643(+3.544)
	SVR	5.962(+1.359)	8.131(+1.257)	6.240(+1.545)	8.413(+1.450)	6.429(+1.539)	8.651(+1.552)
	GAT	4.854(+0.251)	6.989(+0.115)	5.050(+0.355)	7.121(+0.158)	5.362(+0.472)	7.373(+0.274)
	GRU	4.875(+0.272)	6.990(+0.116)	4.983(+0.288)	7.099(+0.136)	5.287(+0.397)	7.338(+0.239)
	ASTGCN	4.829(+0.226)	7.052(+0.178)	4.898(+0.203)	7.109(+0.146)	5.357(+0.467)	7.422(+0.323)
	STCGAT	4.835(+0.232)	6.977(+0.103)	4.839(+0.144)	7.035(+0.072)	4.973(+0.083)	7.189(+0.090)
	STPN	4.804(+0.201)	6.974(+0.100)	4.927(+0.232)	7.101(+0.138)	5.109(+0.219)	7.299(+0.200)
	FAST-CA	**4.603**	**6.874**	**4.695**	**6.963**	**4.890**	**7.099**

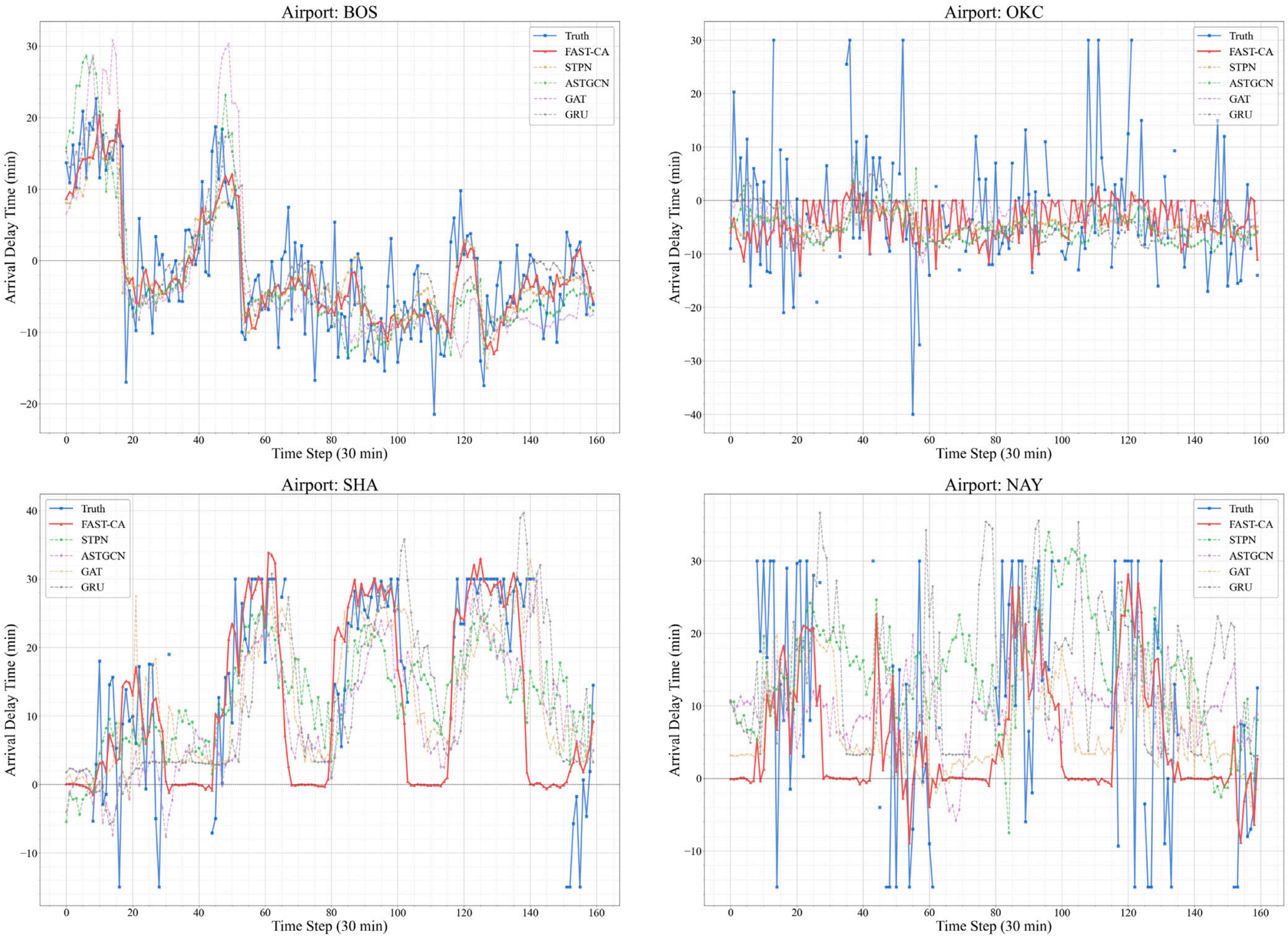


Fig. 5. Arrival delay prediction visualization on U.S. and China dataset.

by a color map. For improved clarity, the size of each circle indicates the respective airport's traffic volume, thus vividly depicting the spatial spread of delays. Although all three spatial–temporal models incorporate attention mechanisms, FAST-CA distinguishes itself with its adaptive learning capabilities and dynamic graph properties. These attributes allow FAST-CA to more accurately reflect real-world airport delay scenarios, demonstrating its superior proficiency in capturing intricate spatial–temporal dependencies.

From Fig. 6, it becomes clear that the Yangtze River Delta and Bohai Bay areas in China are significantly impacted by delays. The FAST-CA

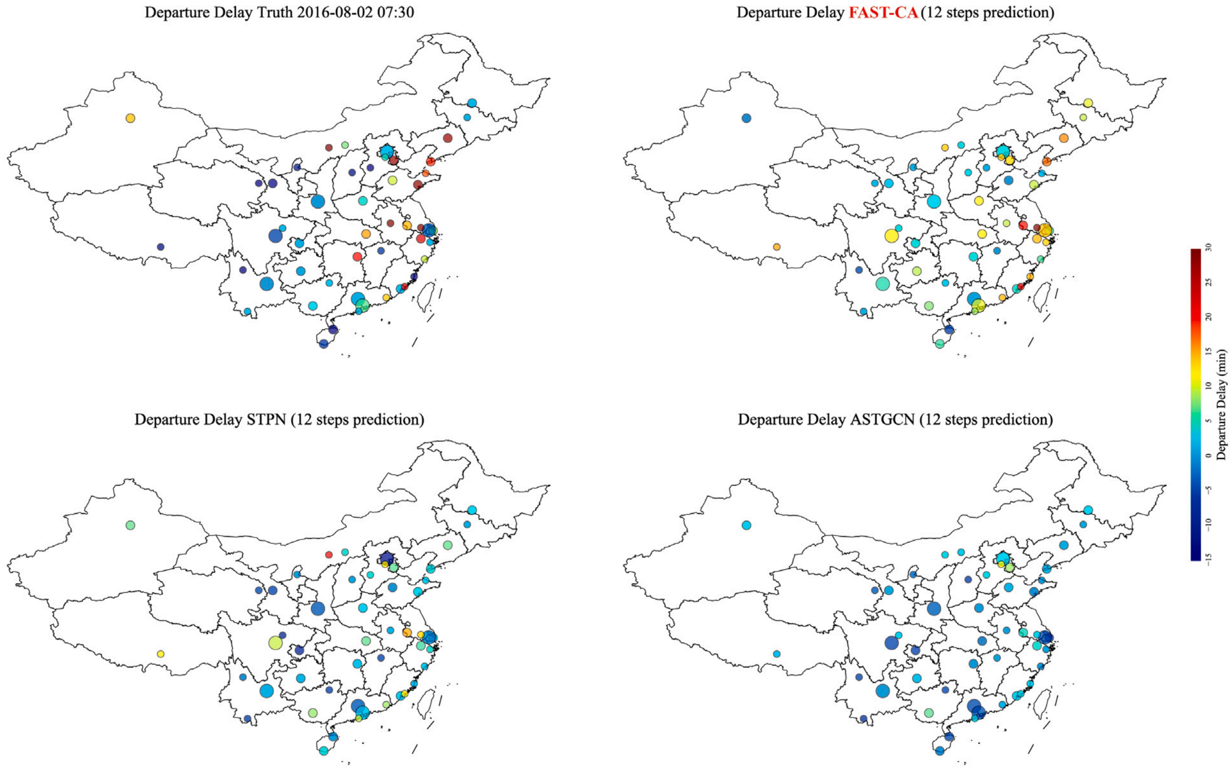

Departure Delay Truth 2016-08-02 07:30
Departure Delay FAST-CA (12 steps prediction)
Departure Delay STPN (12 steps prediction)
Departure Delay ASTGCN (12 steps prediction)
30
25
20
15
10
5
0
−5
−10
−15
Departure Delay (min)

Fig. 6. Spatial visualization of 12-step ahead departure delay prediction on China dataset.

model's schematic representation accurately depicts these high-delay regions, aligning well with real-world scenarios in areas exhibiting lower delays. Conversely, the performance of the other two models, particularly the STPN, reveals notable discrepancies. For instance, the STPN model predominantly captures extreme delays at specific airports but falls short of reflecting broader regional delay patterns. Moreover, as shown in Fig. 7, the delay dynamics across various regions of the United States are effectively illustrated. The central United States, for example, is characterized by higher delays, with the bustling Denver International Airport exhibiting significant congestion.

In practical terms, the FAST-CA model's exceptional performance offers substantial support for air traffic flow management. Its capacity to accurately forecast arrival and departure delays hours in advance is particularly beneficial for managing intricate situations and incomplete datasets. Such comprehensive effectiveness establishes FAST-CA as a valuable asset in tackling real-world challenges in airport delay prediction.

Finally, we have compiled statistics on the parameters and training costs for each model, as detailed in Table 7. The FAST-CA model notably leads in parameter count, a complexity that usually signifies a greater ability to capture data features, potentially enhancing prediction accuracy. Despite the longest training time among the models, this duration is deemed reasonable against the backdrop of the model's extensive parameterization. Moreover, when considering the ratio of parameter quantity to runtime, our training duration is highly acceptable. This holds particularly true for the China dataset, where FAST-CA's tailored optimization achieves faster training speeds without sacrificing complexity. This efficiency suggests that the additional training time FAST-CA requires is a worthwhile trade-off for its superior predictive performance, especially evident in its handling of the China dataset. Here, FAST-CA not only demonstrates relative efficiency in training time but also excels in predictive outcomes due to its specific optimization, striking an optimal balance between performance and efficiency. Thus, considering the parameters, training time, and potential for enhanced prediction accuracy, FAST-CA showcases an outstanding cost-performance ratio in our performance assessment.

Table 7
Models parameters and training cost statistics.

Model	U.S. dataset		China dataset	
	Parameters	Train time (epoch)	Parameters	Train time (epoch)
GAT	1788	27.1 s	4092	9.7 s
GRU	219660	30.6 s	256524	8.8 s
STCGAT	700088	144.5 s	2110064	160.2 s
ASTGCN	450031	181.3 s	450031	111.7 s
STPN	94580	232.1 s	94576	80.9 s
FAST-CA	3976292	633.9 s	3976090	205.1 s

4.5. Ablation study

To gain deeper insights into how each module extracts specific patterns, we conduct six experiments on FAST-CA-based models, with a particular component either removed or modified in each variant. Additionally, we conduct four crossover experiments by substituting key components of the STPN with modules from FAST-CA, illustrating the effective applicability of FAST-CA modules across various scenarios. Moreover, recognizing the pivotal role of AFMI–GAT in feature aggregation, we replace the spatial feature extraction modules in different models with AFMI–GAT and undertake four ablation studies. These studies are aimed at evaluating its contributions towards enhancing efficiency, accuracy, and scalability. The initial two tasks utilize the China dataset, which poses a more complex forecasting challenge due to its unique characteristics. The final task employs both the China and U.S. datasets, which differ significantly in flight volume, providing a comprehensive assessment of the models' scalability and performance across diverse operational contexts.

Regarding task 1, the variants are detailed as follows:

1. FAST-CA_W: We remove the weather inputs along with the fused module that combines weather and delay inputs.
2. FAST-CA_TO: We exclude the task-oriented module responsible for extracting relationships between departure and arrival delays.

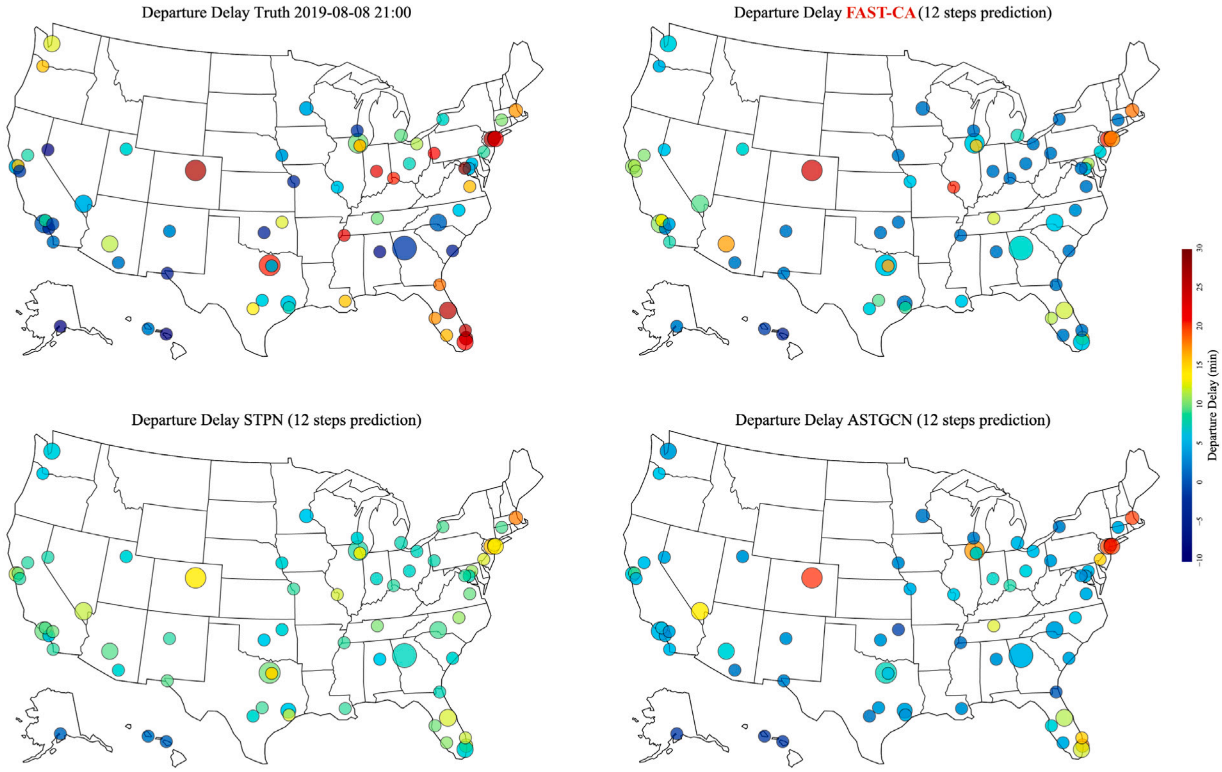

Departure Delay Truth 2019-08-08 21:00
Departure Delay FAST-CA (12 steps prediction)
Departure Delay STPN (12 steps prediction)
Departure Delay ASTGCN (12 steps prediction)
Departure Delay (min)
30
25
20
15
10
5
0
−5
−10

Fig. 7. Spatial visualization of 12-step ahead departure delay prediction on U.S. dataset.

Table 8
Results for FAST-CA and its ablations on arrival and departure delays with feature dimensions.

	Average		Arrival delay		Departure delay	
Variants	*MAE*	*RMSE*	*MAE*	*RMSE*	*MAE*	*RMSE*
FAST-CA	8.346	11.069	8.240	10.870	8.463	11.283
FAST-CA_W	8.393	11.161	8.277	10.967	8.520	11.372
FAST-CA_TO	8.374	11.110	8.280	10.922	8.477	11.313
FAST-CA_ST	8.600	11.314	8.453	11.099	8.762	11.547
FAST-CA_G	8.405	11.125	8.334	10.940	8.483	11.326
FAST-CA_L	8.447	11.164	8.336	10.971	8.570	11.373
FAST-CA_PE	8.434	11.118	8.315	10.903	8.566	11.351

3. FAST-CA_ST: The spatial–temporal fusion module is replaced with a standard GAT module, which only identifies spatial patterns within the graph.
4. FAST-CA_G: We omit the predefined matrix as inputs, limiting the model to learning only local adaptive graph structures.
5. FAST-CA_L: We remove the entire adaptive learning module. Consequently, the model evaluates the relationships between different airports solely based on the global adjacency matrix.
6. FAST-CA_PE: We exclude the positional encoding module, which is applied to the context-aware temporal attention module.

The results of the first six experiments are concisely summarized in Table 8. Upon comparing our proposed models with the other variants, several key observations emerge: (1) It is evident that our proposed models outperform all similar variants. This underscores the harmonious integration of each module, which efficiently captures and fuses different types of patterns within the extensive airport network graph. (2) The slight advantage of FAST-CA over FAST-CA_W suggests that the model benefits minimally from weather information, as some patterns can already be inferred from the embedded delay information. (3) The significant performance drop in the FAST-CA_ST variant, compared to FAST-CA_TO, reveals the critical role of the spatial–temporal module. It captures spatial and temporal dependencies, which is essential for recognizing factors in airport delay propagation. In contrast, the task-oriented module primarily extracts relationships between arrival and departure delays. (4) The inferior performance of FAST-CA_L compared to FAST-CA_G indicates that local information (i.e., the dynamic and adaptive adjacency matrix) is more crucial than global information (i.e., the predefined adjacency matrix based on airport distances). The model can effectively learn global patterns from weather and delay inputs. (5) The poor predictive ability of FAST-CA_PE highlights the significance of the flight schedule's explicit daily and weekly periodicity. The multi-level positional encoding module adeptly represents this periodicity in the attention mechanism, as further detailed in Section 4.7. Notably, the variant model performs worse in predicting departure delays, suggesting a higher sensitivity to periodicity representation in departure delays.

As depicted in Fig. 8, several observations can be made when comparing the results under different hyper-parameters: (1) Both MAE and RMSE display relatively stable variations across different experiments, with optimal settings achieving a balance in the middle. This pattern indicates that the model exhibits robustness, maintaining consistent performance despite changes in these hyper-parameters. (2) Despite the model's overall robustness, the selection of appropriate hyper-parameters emerges as a more critical factor influencing the model's predictive capability than the integration of different modules. For instance, setting the node embedding dimension to 10 results in the worst MAE outcome and the third-worst RMSE outcome when compared to all other ablation configurations.

In task 2, we enhance the original STPN model by incorporating our mechanisms and developing four distinct ablations, as follows:

1. STPN_TO: The ablation introduces a task-oriented attention module, effectively capturing the interdependencies between departure and arrival delay sequences, thereby integrating temporal dynamics more cohesively.
2. STPN_ST: The original graph convolution network is replaced with a more advanced spatial–temporal fusion module, aimed

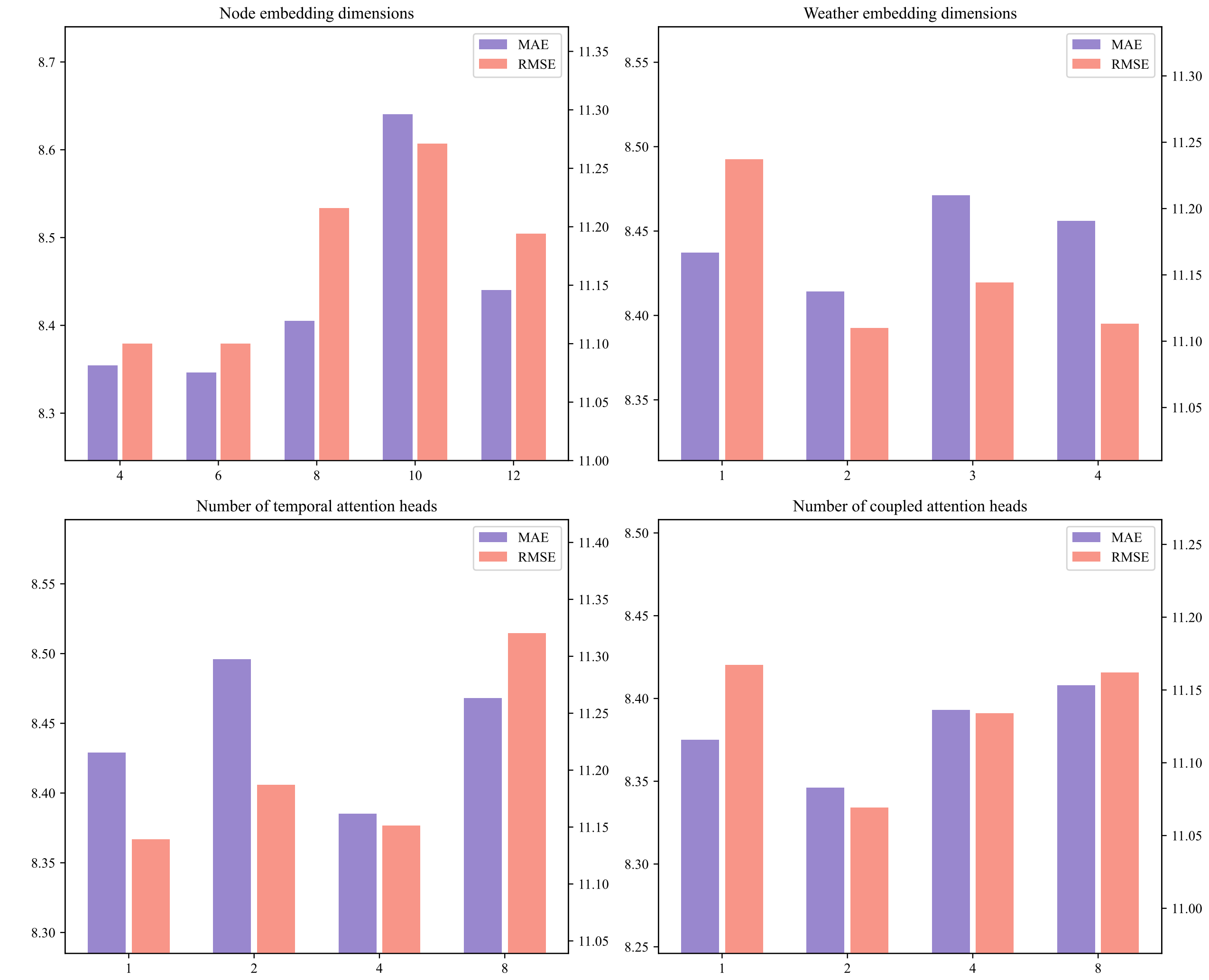


Fig. 8. Experimental results with different hyper-parameters settings.

Table 9
Results for integrated models on arrival and departure delays with feature dimensions.

	Average		Arrival delay		Departure delay	
Variants	*MAE*	*RMSE*	*MAE*	*RMSE*	*MAE*	*RMSE*
STPN	8.914	11.420	8.916	11.351	8.912	11.496
STPN_TO	8.809	11.266	8.800	11.204	8.818	11.333
STPN_ST	8.578	11.046	8.398	10.838	8.776	11.270
STPN_L	8.746	11.307	8.721	11.232	8.773	11.388
STPN_PE	8.404	11.176	8.268	10.963	8.553	11.406

at improving the model's ability to capture complex spatio-temporal interactions.

3. STPN_L: This variant integrates a predefined matrix with an adaptive learning module within our framework, allowing the model to better represent both stable and dynamic patterns observed in the data.
4. STPN_PE: In this ablation, periodic representation is substituted with a context-aware positional encoding mechanism, designed to capture multi-level periodicity with enhanced precision and context sensitivity.

The outcomes of the four experiments conducted in task 2 are presented in Table 9. A comparative analysis between the original model and the modified models reveals several key observations: (1) It is apparent that all modified models demonstrate superior performance across various metrics compared to the original model. This improvement underscores the scalability and effectiveness of the FAST-CA modules in diverse scenarios. (2) Notably, the STPN_PE model exhibits the most significant discrepancy in error compared to the original model. This can be attributed to the multi-level periodic embedding's ability to intricately represent the interrelations among different positions within a time sequence. Such a representation is pivotal for the effective application of the temporal attention mechanism, thereby enhancing predictive accuracy.

To evaluate the effectiveness of the Adaptive Graph Learning module, we execute four sets of one-to-one comparison experiments in task 3. This involves integrating the module with both the STPN and STCGAT models and conducting a comparative analysis focusing on running time and validation loss. The specific modifications made are as follows:

1. STPN_L: Unlike the original model, which relies solely on three types of static adjacency matrices, we incorporate a dynamic embedding module designed to extract local features more effectively. This addition aims to enhance the model's ability to adapt to changing data patterns by capturing dynamic interactions.
2. STCGAT_G: In the original STCGAT model, the weights of the graph attention module are determined based on node embeddings. We augment this mechanism with a predefined matrix, enabling the model to capture global representations alongside local interactions.

Table 10
Ablation results for STPN variants.

Dataset	Variants	Running time	Average		Arrival delay		Departure delay	
			MAE	*RMSE*	*MAE*	*RMSE*	*MAE*	*RMSE*
China	STPN	**7792 s**	8.914	11.420	8.916	11.351	8.912	11.496
	STPN_L	7905 s	**8.796**	**11.372**	**8.811**	**11.293**	**8.780**	**11.458**
U.S.	STPN	**10230 s**	6.223	8.704	7.344	**9.999**	5.043	7.093
	STPN_L	10510s	**6.101**	**8.691**	**7.312**	10.041	**4.826**	**6.995**

Table 11
Ablation results for STCGAT variants.

Dataset	Variants	Running time	Average		Arrival delay		Departure delay	
			MAE	*RMSE*	*MAE*	*RMSE*	*MAE*	*RMSE*
China	STCGAT	**16807 s**	8.681	11.439	8.820	11.501	8.527	11.370
	STCGAT_G	17421 s	**8.638**	**11.414**	**8.763**	**11.462**	**8.499**	**11.360**
U.S.	STCGAT	**22063 s**	6.235	8.655	7.630	10.260	4.767	**6.967**
	STCGAT_G	22271 s	**6.206**	**8.643**	**7.577**	**10.231**	**4.764**	6.972

As depicted in Table 10, STPN_L exhibits a slight increase in running time but achieves superior prediction performance across all metrics. These results highlight the module's efficiency and accuracy, particularly with the models that only involve a predefined adjacency matrix. With regard to the variants of STCGAT, results presented in Table 11 show a similar trend. The results demonstrate higher accuracy with a marginal increase in running time for nearly all metrics. Given that STCGAT dynamically computes the relationships between nodes, integrating it with prior static knowledge enhances the model's explanatory power, thereby improving its predictive capabilities. In conclusion, the AFMI–GAT framework enhances model performance in terms of accuracy with minimal additional running time across various baselines and datasets. This underscores the framework's efficiency, accuracy, and scalability. Such results demonstrate the potential of AFMI–GAT in improving prediction models without imposing substantial computational overheads.

4.6. Visualization of dynamic and adaptive adjacency matrix

In this section, the concept of dynamic and adaptive adjacency matrix is introduced. Fig. 9(a) illustrates the average departure delay on January 11, 2016, for 50 airports in China. Four airports were selected for detailed analysis: TNA, TYN, WNZ, and SJW, corresponding to the indexes 25, 28, 32, and 35, respectively. The four airports are interconnected. These airports are highlighted with yellow boxes in Fig. 9(b), which depicts the normalized distance adjacency matrix representing the distances between the 50 Chinese airports.

The adaptive adjacency matrix, being dynamic, more accurately reflects the relationships between airports. Figs. 9(d), (e), and (f) clearly show the dynamic changes within the data highlighted in yellow boxes. Focusing on these four airports, the variations in their interrelationships are distinctly noticeable at different times on January 11, 2016. The actual departure delay data for these airports on the specified date is plotted in Fig. 9(c).

Focusing on delay data at 16:00, 18:00, and 20:00 (highlighted in yellow rectangles), we observe that the delay trends at both TNA and WNA airports are increasing. This indicates a similarity and interrelation in the delay patterns of these two airports. This is reflected in the data points corresponding to Figs. 9(d), (e), and (f) (marked with red circles), where the color changes from dark blue to light blue, signifying an increase in the correlation between the two airports. The correlation between TNA and WNA airports is depicted by the red line in Fig. 9(a). Similarly, between 16:00 and 22:00, the delay data for TYN and SJW airports show comparable fluctuations (highlighted in Fig. 9(c), indicating a mutual correlation in their delay patterns. This is depicted in Figs. 9(d), (e), and (f) (marked with purple circles), where the color transition from dark to light blue indicates a strengthening correlation. The correlation between TYN and SJW airports is depicted by the purple line in Fig. 9(a). However, a static distance adjacency matrix fails to capture these dynamically changing correlations.

This indicates that the continually learning and dynamically updating adaptive adjacency matrix provides a more accurate representation of the actual conditions between airports than the distance adjacency matrix. In the time series graph presented in Fig. 9(c), some data is missing in the morning. This is attributed to the greater irregularity of aviation delay datasets compared to road traffic datasets, making the research more challenging.

4.7. Periodicity learning analysis

In this section, we describe the multi-level periodicity observed in delay data and how our model reflects this pattern using context-aware positional encoding. As shown in Fig. 10, we exemplify this with a 3-week departure delay at the CAN airport in China. Fig. 10(a) visualizes the ground truth and predicted delay, while Fig. 10(b) shows the context-aware positional encoding for each timestamp. The delay data reveals two levels of periodicity—daily and weekly. The red boxes, highlighting the daily period, show that delays typically drop to negative values after 12 P.M. and surge to positive values on a day-to-day basis within a week. The purple boxes, indicating the weekly period, reveal a significant peak in delays around 3 P.M. every Sunday.

By applying Eqs. (6) and (7), each timestamp is assigned a unique phase shift within two-level periodic spans, which then repeats in the subsequent period. The upper part of the encoding repeats daily, while the lower part does so weekly. By combining this unique encoding with delay features, our proposed model effectively captures periodic delay patterns. Each encoding corresponds to its respective periodic pattern. As revealed in the ablation study, the model is capable of mitigating unpredictable interventions at specific times by fusing multi-level periodic insights with delay embedding. This fusion of periodic information and delay data highlights the model's effectiveness in accurately predicting delays.

4.8. Case study

To comprehend how the model effectively learns in different scenarios, we analyze the prediction error across various airports over a period, correlating it with their factual characteristics to provide plausible explanations. Then, we analyze the model's performance across various conditions, including different airport networks, and during peak and off-peak hours.

The first case study is conducted using the China dataset. We specifically track the departure and arrival volumes over time and use the overall MAE as our accuracy metric. The results of this experiment

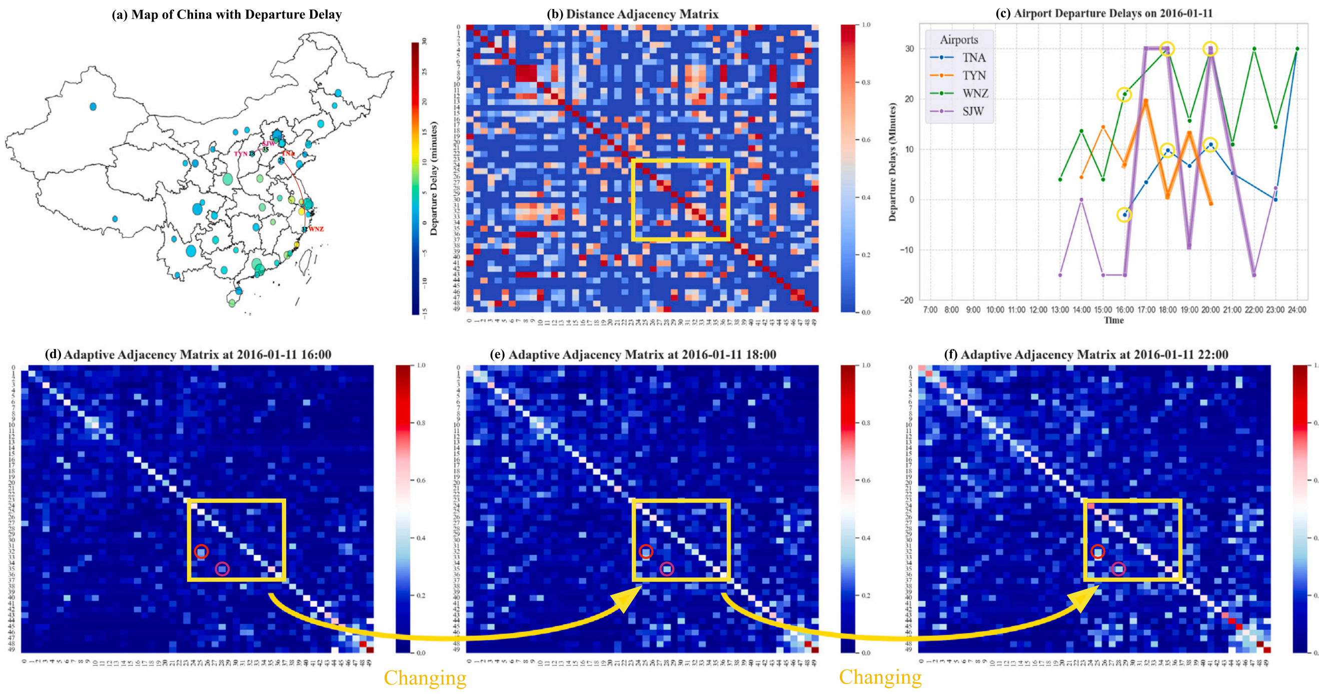
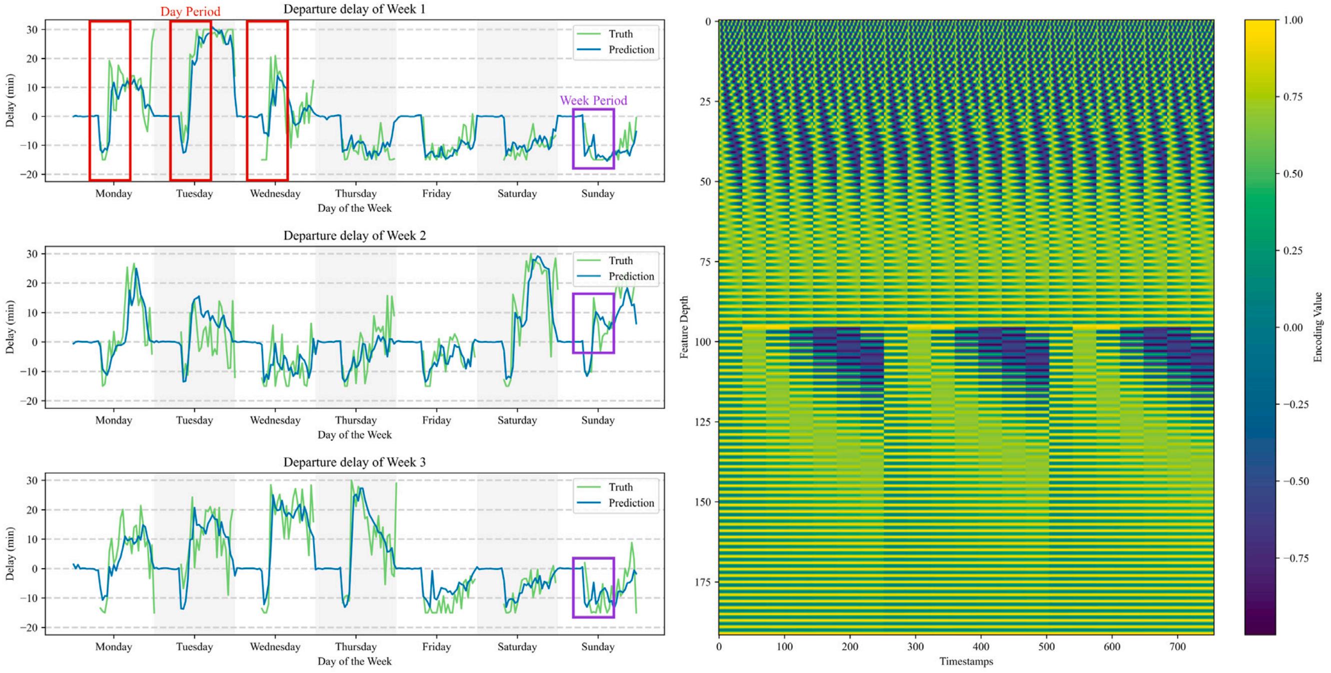


Fig. 9. Visualization of the dynamic and adaptive adjacency matrix.



(a) Visualization of departure delay prediction on CAN airport from Chinese dataset

(b) Visualization of the periodic positional encoding

Fig. 10. Visualization of multi-level periodic delay and its related context-aware positional encoding.

are illustrated in Fig. 11. Airports are listed in descending order of departure volume since the number of departures and arrivals at each airport varies slightly.

It is observed that the prediction error tends to increase as the airport volume decreases. This trend can be attributed to two possible reasons. Firstly, the enhanced accuracy at busier airports may be partly attributed to the larger samples associated with them. These samples typically have fewer missing values, making it easier for the model to extract and learn patterns specific to these airports. Secondly, airports with lower volumes often have smaller handling capacities and are more susceptible to disruption from extreme conditions, which are inherently more challenging to forecast. To further elucidate this phenomenon with detailed examples, we select four typical airports with distinct characteristics.

- Beijing Capital International Airport (PEK): Located in northern China, Beijing experiences a temperate monsoon climate. Extreme weather conditions like dust storms and gales occur rarely during spring and summer. Beijing International Airport, serving as the central hub of the northern Chinese aviation network, records the

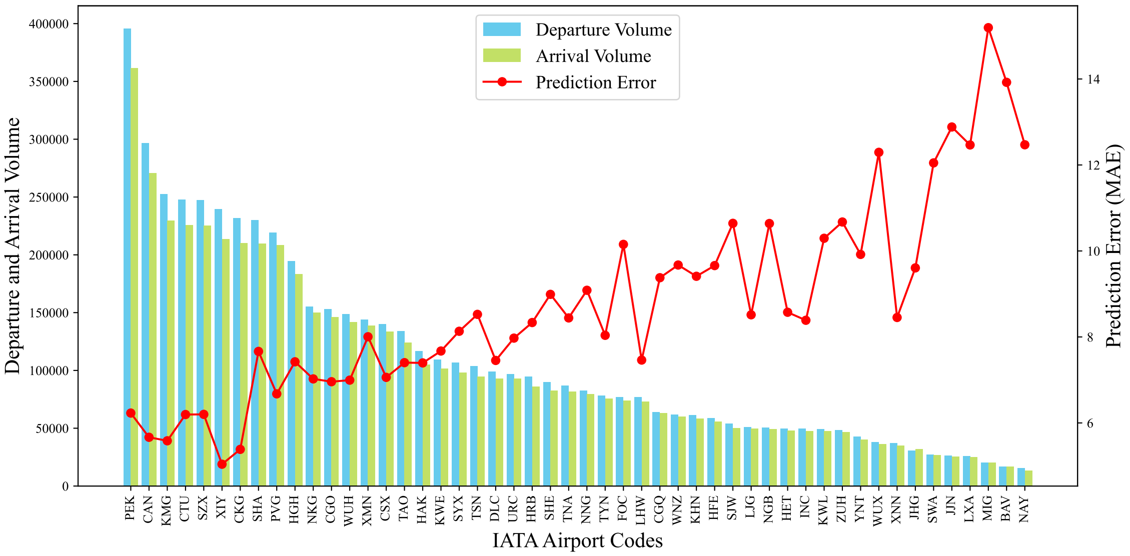


Fig. 11. Departure and arrival volumes and model prediction error at each Chinese airport.

highest volume in the period in our study. This dominant role facilitates the propagation of delays to and from this airport. The proposed model achieves accurate forecasts here, benefiting from moderate weather conditions and a substantial dataset.

- Shanghai Hongqiao International Airport (SHA): Situated on China's east coast, Shanghai is characterized by a humid subtropical climate. The city frequently faces extreme weather events such as typhoons and heavy rain, which significantly impact flight punctuality and increase delay unpredictability. Although Hongqiao International Airport is a major domestic and regional hub in eastern China, it faces operational constraints due to its location at the city's edge. The combination of unforeseen weather and operational disruptions makes it challenging for the model to predict delays accurately, leading to the poorest accuracy among all airports with over 200,000 departures and arrivals.
- Sunan Shuofang International Airport (WUX): Located 100 km west of Shanghai, Suzhou often experiences similar extreme weather conditions in summer and autumn. Sunan Shuofang International Airport is situated in the densely populated Jiangsu region, in close proximity to two major hubs. This proximity not only results in fewer samples for model training but also leads to a lower priority in delay management. Consequently, the combination of unpredictable extreme weather, small datasets, and limited flight operations contributes to a higher prediction error at this airport.
- Lanzhou Zhongchuan International Airport (LHW): Lanzhou, located in Northwest China, is characterized by its semi-arid climate, which typically experiences less extreme weather than other regions. This stable weather pattern contributes to the accuracy of flight delay predictions, as weather-related disruptions are less severe and more predictable. Additionally, the airport plays a crucial role in the region's aviation network though it is not a large hub. This balance ensures sufficient data to train the model while avoiding the complexities and congestion typically found in larger airports. Therefore, Lanzhou's moderate weather conditions and mid-sized flight records result in the best forecasting ability among all airports with a comparable volume of traffic.

In the subsequent analysis, we evaluate the performance of FAST-CA across different airport networks. Building upon the original dataset from China, we have selected airports based on their traffic volume, organizing them into three new networks comprising 10, 30, and 50 airports, respectively, ranked from highest to lowest traffic volume. Experimental data presented in Table 12 reveals that in networks composed of high-traffic airports, the ASTGCN, STPN, and FAST-CA all demonstrate reduced error rates. Notably, within networks of increasingly higher-traffic airports, where delay characteristics are more pronounced, the FAST-CA exhibits a markedly superior predictive capability.

Furthermore, we analyze the performance of the model during peak and off-peak hours. Utilizing spring festival travel data from Civil Aviation Administration of China[5] and the operational timetable of CAN Airport,[6] we categorize the hours of 6, 7, 8, 10, 11, and 12 as peak hours, and 13, 14, 15, 21, and 23 as off-peak hours, thereby splitting the original dataset into two distinct subsets. Figs. 12 and 13 illustrate the performance of our model at the PEK and CAN airports under these two conditions. Across these two newly created datasets, the FAST-CA consistently demonstrates superior predictive capabilities, especially for peak hours. This can be attributed to the higher traffic volume, reduced instances of missing data, and stronger regularity during peak hours, resulting in lower MAE and RMSE error values.

5. Conclusion and future works

This study introduces the FAST-CA framework, specifically designed for predicting delay propagation in airport networks. The FAST-CA model encompasses several key components: a fusion of dynamic graph information inputs, an adaptive graph learning module, a spatial–temporal fusion module, a context-aware temporal attention module, and a task-oriented attention module. These components synergistically work to learn adaptive dynamic relationships between airport nodes, and coupled spatial–temporal dependencies, and to extract information on periodicity and temporal dependencies, while considering the coupling between departure and arrival sequences. Our model's efficacy is rigorously evaluated using two real-world datasets, demonstrating its robustness and applicability. The FAST-CA model achieves state-of-the-art performance, with each module significantly contributing to the overall predictive accuracy improvement. More specifically, the adaptive graph learning module in our FAST-CA framework uncovers dynamic and adaptive relationships between airport nodes. The adjacency matrices learned through this module more accurately reflect

[5] http://www.cadas.com.cn/news/20191220174812000001.html

[6] https://www.csair.com/eu/de/tourguide/airport_service/airports_info/domestic/18id40mhhm94.shtml

Table 12
Results on the China delay dataset with different airport networks.

		1.5 h		3 h		6 h	
	Method	*MAE*	*RMSE*	*MAE*	*RMSE*	*MAE*	*RMSE*
Arrival delay	ASTGCN_10	6.436	8.571	6.872	9.147	7.448	9.851
	STPN_10	6.384	8.381	6.944	9.059	7.712	10.035
	FAST-CA_10	**5.003**	**6.936**	**5.828**	**7.929**	**6.977**	**9.294**
	ASTGCN_30	7.788	10.130	8.105	10.511	8.496	10.942
	STPN_30	7.415	9.476	7.958	10.096	8.599	10.873
	FAST-CA_30	**6.480**	**8.652**	**7.205**	**9.494**	**8.187**	**10.461**
	ASTGCN_50	8.881	11.624	9.124	11.869	9.459	12.203
	STPN_50	8.908	11.607	9.369	12.150	9.877	12.735
	FAST-CA_50	**7.468**	**10.066**	**8.416**	**11.048**	**9.209**	**11.865**
Departure delay	ASTGCN_10	8.854	12.518	8.874	12.522	9.022	12.673
	STPN_10	6.889	8.863	7.208	9.303	7.662	9.933
	FAST-CA_10	**5.862**	**7.950**	**6.334**	**8.594**	**6.950**	**9.298**
	ASTGCN_30	8.738	12.215	8.760	12.367	8.882	12.414
	STPN_30	7.942	10.298	8.267	10.697	8.699	11.260
	FAST-CA_30	**7.263**	**9.548**	**7.626**	**9.996**	**8.078**	**10.470**
	ASTGCN_50	8.804	12.617	8.822	12.698	8.918	12.568
	STPN_50	9.050	11.693	9.252	12.011	9.590	12.404
	FAST-CA_50	**8.254**	**11.053**	**8.623**	**11.435**	**8.996**	**11.841**

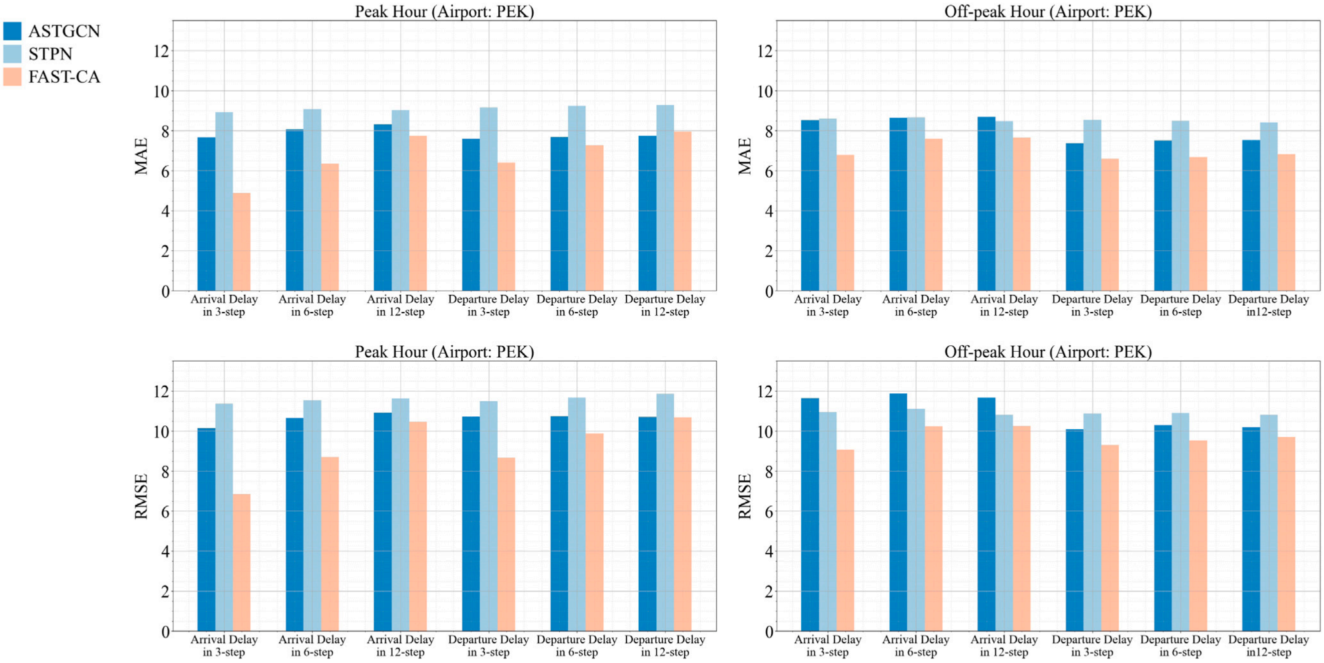


Fig. 12. Experimental results with peak hour and off-peak hour on PEK airport.

the temporal relationships of the respective nodes compared to static distance matrices. The temporal attention module effectively captures the daily and weekly semantic information inherent in the delay time series. Meanwhile, the task-oriented attention module efficiently extracts coupled features of departure and arrival sequences. The fusion of this information as output significantly enhances the accuracy of our predictions, showcasing the efficacy of our approach in capturing complex delay patterns in airport networks.

These findings not only underscore the efficacy of our FAST-CA framework in handling the complexities of airport delay prediction but also highlight its potential to provide comprehensive insights into delay propagation dynamics. This research paves the way for more accurate and reliable delay management strategies in air transportation systems, contributing substantially to the field of delay prediction and management. Our work, while pioneering, is not without its limitations. Firstly, despite the model's robust performance in leveraging operational flight data and weather information, it may encounter challenges with data sparsity, particularly in smaller airports or during certain periods, suggesting a need for further exploration into handling missing data more effectively. Secondly, although our framework achieves superior predictive accuracy, its extensive parameter set necessitates a longer runtime. This reveals an opportunity for optimizing computational efficiency to enable faster, yet equally accurate, predictions. Lastly, while the model demonstrates commendable adaptability across various network sizes, its generalization to specific airport networks with unique traffic volumes or geographical characteristics is an area that could benefit from focused improvement. Addressing these limitations will not only enhance the model's applicability but also its operational efficiency in diverse airport environments.

In our future research endeavors, we aim to delve into micro-level modeling by considering the propagation of delays along flight chains, integrating this aspect with the propagation process within airport networks for a comprehensive and detailed examination of the mechanisms underlying flight delay propagation. Furthermore, we plan to explore more advanced methods of information fusion to enhance both the predictive performance and interpretability of our models.

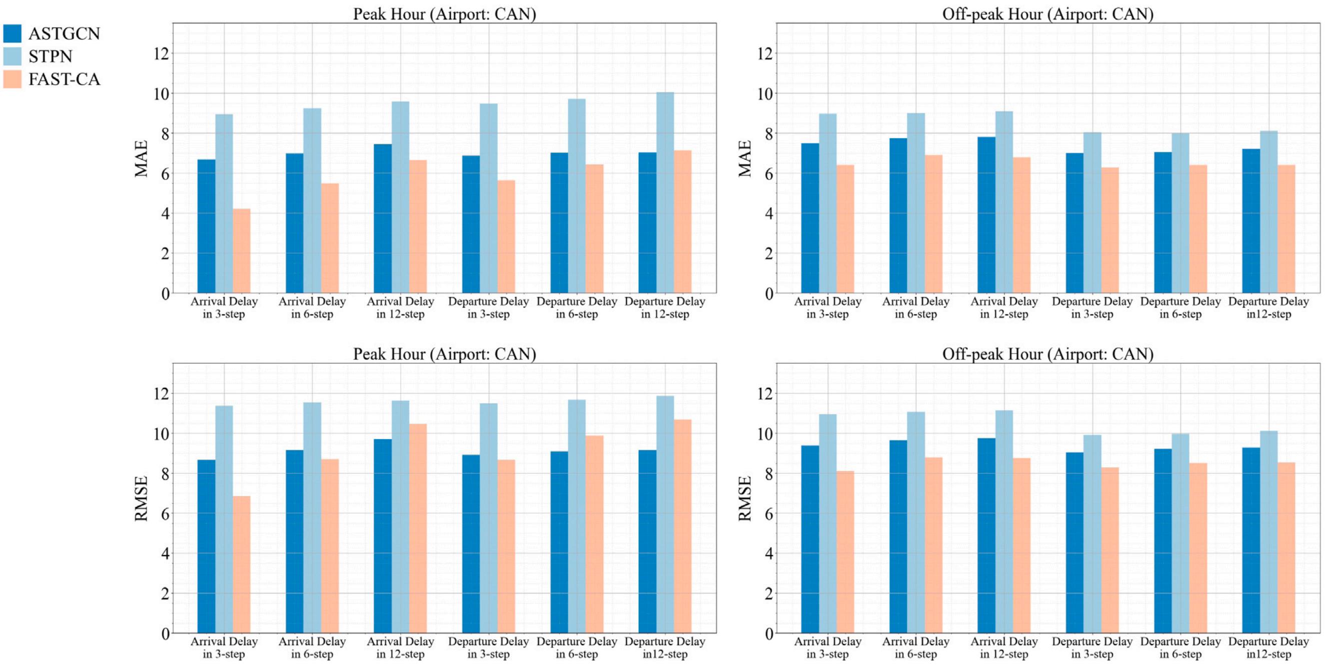


Fig. 13. Experimental results with peak hour and off-peak hour on CAN airport.

Additionally, we intend to investigate new algorithmic optimization techniques or leverage more efficient computational frameworks to reduce the time required for model training and prediction. This endeavor would necessitate examining more sophisticated graph neural network architectures or developing innovative parallel processing techniques. Finally, it would be highly desirable to explore how to effectively integrate real-time data, such as weather changes and live flight statuses, into our models to improve the accuracy and timeliness of predictions. This approach needs to involve developing new data processing workflows to rapidly respond to changes in real-time data.

CRediT authorship contribution statement

Chi Li: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization, Writing – review & editing. **Xixian Qi:** Writing – original draft, Visualization, Validation, Software, Methodology, Data curation, Conceptualization, Writing – review & editing. **Yuzhe Yang:** Writing – original draft, Visualization, Validation, Data curation, Writing – review & editing. **Zhuo Zeng:** Writing – original draft, Visualization, Data curation, Writing – review & editing. **Lianmin Zhang:** Supervision, Resources. **Jianfeng Mao:** Supervision, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have provided a link to access our dataset in the Datasets section of our manuscript.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under grant U1733102, in part by the National Natural Science Foundation of China (NSFC) under grant 72394362, in part by the Guangdong Provincial Key Laboratory of Big Data Computing, The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen) under grant 2024SC0003, in part by Shenzhen Science and Technology Innovation Committee under grant ZDSYS20170725140921348, in part by Shenzhen Science and Technology Innovation Committee under Shenzhen Stability Science Program grant 2024SC0010, and in part by The Chinese University of Hong Kong, Shenzhen (CUHK-Shenzhen) under grant PF.01.000404.

References

[1] A.M. Churchill, D.J. Lovell, M.O. Ball, Flight delay propagation impact on strategic air traffic flow management, Transp. Res. Rec. 2177 (1) (2010) 105–113.

[2] Y. Wang, M.Z. Li, K. Gopalakrishnan, T. Liu, Timescales of delay propagation in airport networks, Transp. Res. E 161 (2022) 102687.

[3] K. Gopalakrishnan, H. Balakrishnan, Control and optimization of air traffic networks, Annu. Rev. Control Robot. Auton. Syst. 4 (2021) 397–424.

[4] Y. Wu, H. Yang, Y. Lin, H. Liu, Spatiotemporal propagation learning for network-wide flight delay prediction, IEEE Trans. Knowl. Data Eng. (2023).

[5] N. Pyrgiotis, K.M. Malone, A. Odoni, Modelling delay propagation within an airport network, Transp. Res. C 27 (2013) 60–75.

[6] I. Simaiakis, H. Balakrishnan, A queuing model of the airport departure process, Transp. Sci. 50 (1) (2016) 94–109.

[7] J.-T. Wong, S.-C. Tsai, A survival model for flight delay propagation, J. Air Transp. Manag. 23 (2012) 5–11.

[8] N. Nayak, Y. Zhang, Estimation and comparison of impact of single airport delay on national airspace system with multivariate simultaneous models, Transp. Res. Rec. 2206 (1) (2011) 52–60.

[9] J.J. Rebollo, H. Balakrishnan, Characterization and prediction of air traffic delays, Transp. Res. C 44 (2014) 231–241.

[10] J. Chen, M. Li, Chained predictions of flight delay using machine learning, in: AIAA Scitech 2019 Forum, 2019, p. 1661.

[11] Q. Li, R. Jing, Flight delay prediction from spatial and temporal perspective, Expert Syst. Appl. 205 (2022) 117662.

[12] W. Zeng, J. Li, Z. Quan, X. Lu, A deep graph-embedded LSTM neural network approach for airport delay prediction, J. Adv. Transp. 2021 (2021) 1–15.

[13] J. Bao, Z. Yang, W. Zeng, Graph to sequence learning with attention mechanism for network-wide multi-step-ahead flight delay prediction, Transp. Res. C 130 (2021) 103323.

[14] K. Cai, Y. Li, Y.-P. Fang, Y. Zhu, A deep learning approach for flight delay prediction through time-evolving graphs, IEEE Trans. Intell. Transp. Syst. (2021).

[15] R. Beatty, R. Hsu, L. Berry, J. Rome, Preliminary evaluation of flight delay propagation through an airline schedule, Air Traffic Control Q. 7 (4) (1999) 259–270.

[16] P. Fleurquin, J.J. Ramasco, V.M. Eguiluz, Systemic delay propagation in the US airport network, Sci. Rep. 3 (1) (2013) 1159.

[17] B. Yu, Z. Guo, S. Asian, H. Wang, G. Chen, Flight delay prediction for commercial air transport: A deep learning approach, Transp. Res. E 125 (2019) 203–221.

[18] M. Güvercin, N. Ferhatosmanoglu, B. Gedik, Forecasting flight delays using clustered models based on airport networks, IEEE Trans. Intell. Transp. Syst. 22 (5) (2020) 3179–3189.

[19] J. Sun, T. Dijkstra, C. Aristodemou, V. Buzetelu, T. Falat, T. Hogenelst, N. Prins, B. Slijper, Designing recurrent and graph neural networks to predict airport and air traffic network delays, in: 10th International Conference for Research in Air Transportation, FAA & Eurocontrol, 2022, pp. 1–8.

[20] B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting, 2017, arXiv preprint arXiv:1709.04875.

[21] S. Guo, Y. Lin, N. Feng, C. Song, H. Wan, Attention based spatial–temporal graph convolutional networks for traffic flow forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 922–929.

[22] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, Graph wavenet for deep spatial–temporal graph modeling, 2019, arXiv preprint arXiv:1906.00121.

[23] L. Bai, L. Yao, C. Li, X. Wang, C. Wang, Adaptive graph convolutional recurrent network for traffic forecasting, Adv. Neural Inf. Process. Syst. 33 (2020) 17804–17815.

[24] W. Zhang, F. Zhu, Y. Lv, C. Tan, W. Liu, X. Zhang, F.-Y. Wang, AdapGL: An adaptive graph learning algorithm for traffic prediction based on spatiotemporal neural networks, Transp. Res. C 139 (2022) 103659.

[25] W. Zhao, S. Zhang, B. Wang, B. Zhou, Spatio-temporal causal graph attention network for traffic flow prediction in intelligent transportation systems, PeerJ Comput. Sci. 9 (2023) e1484.

[26] Z. Yan, H. Yang, D. Guo, Y. Lin, Improving airport arrival flow prediction considering heterogeneous and dynamic network dependencies, Inf. Fusion 100 (2023) 101924.

[27] M.Z. Li, K. Gopalakrishnan, K. Pantoja, H. Balakrishnan, Graph signal processing techniques for analyzing aviation disruptions, Transp. Sci. 55 (3) (2021) 553–573.

[28] C.A.I. Kaiquan, L.I. Yue, Z.H.U. Yongwen, F. Quan, Y. Yang, D.U. Wenbo, A geographical and operational deep graph convolutional approach for flight delay prediction, Chin. J. Aeronaut. 36 (3) (2023) 357–367.

[29] X. Ta, Z. Liu, X. Hu, L. Yu, L. Sun, B. Du, Adaptive spatio-temporal graph neural network for traffic forecasting, Knowl.-Based Syst. 242 (2022) 108199.

[30] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, et al., Graph attention networks, Stat 1050 (20) (2017) 10–48550.

[31] Z. Cui, R. Ke, Z. Pu, Y. Wang, Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction, 2018, arXiv preprint arXiv:1801.02143.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[33] S. Moosavi, M.H. Samavatian, A. Nandi, S. Parthasarathy, R. Ramnath, Short and long-term pattern discovery over large-scale geo-spatiotemporal data, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2905–2913.

[34] E. Zivot, J. Wang, Vector autoregressive models for multivariate time series, in: Modeling Financial Time Series with S-PLUS®, Springer, 2006, pp. 385–429.

[35] M.S. Ahmed, A.R. Cook, Analysis of Freeway Traffic Time-Series Data By using Box–Jenkins Techniques, SAGE, 1979, p. 722,

[36] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Stat. Comput. 14 (2004) 199–222.

[37] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014, arXiv preprint arXiv:1412.3555.