Journal of Network Intelligence ©2025 ISSN 2414-8105
Taiwan Ubiquitous Information Volume **10**, Number **4**, November **2025**

Convention Visitor Flow Prediction Based on Spatio-Temporal Graph Neural Network

Wen-Li Cheng

Guangzhou Vocational College of Technology & Business
Guangzhou 511442, P. R. China
chengwenli1984@126.com

Shao-Chi Li*

Guangzhou Administration Institute
Guangzhou 510070, P. R. China
Institute of Systems Engineering
Macau University of Science and Technology
Macau 999078, P. R. China
bluechallenge@126.com

Zhu-Wang Tsai

Nakhon Phanom University
Nakhon Phanom 48000, Thailand
zj6905@163.com

*Corresponding author: Shao-Chi Li
Received August 8, 2024, revised March 4, 2025, accepted September 3, 2025.

Abstract. *Convention and exhibition visitor flow prediction is of great significance in modern city management and large-scale event planning. Traditional methods of people flow prediction, such as linear regression and time series models, are often difficult to deal with complex spatio-temporal dependencies, resulting in insufficient prediction accuracy. In order to solve this problem, this paper proposes a convention visitor flow prediction method based on Spatio-Temporal Constant Differential Reformer Graph Neural Network (ST-EDRGNN). Firstly, the spatial relationship between different areas of the convention centre is captured by the Graph Neural Network (GNN) and combined with the spatial feature learning module based on ordinary differential equations, which enables the propagation process of spatial information to be modelled as a continuous dynamic process, thus describing the dependency relationship between the visitor flows of each area more accurately. Then, the combination of LSH Attention and Reversible Residual Network in Reformer model is used to process long time series data, which effectively reduces the computational complexity and memory consumption, and improves the training efficiency and prediction speed of the model. The experimental results show that the Root Mean Square Error (RMSE) of the ST-EDRGNN model is 1.750 and 1.770, the Mean Absolute Error (MAE) is 1.480 and 1.500, and the Mean Absolute Percentage Error (MAPE) is 3.85% and 3.90%, which are significantly better than the other baseline models on the two datasets.*
Keywords: visitor flow prediction; spatio-temporal graph neural network; reformer; self-attention mechanism; deep learning

---

1. **Introduction.** Pedestrian flow prediction is of great importance in modern city management and business decision-making, especially in high-density public places such as convention and exhibition centres [1, 2]. With the acceleration of urbanisation, more

and more large-scale conventions, exhibitions and events are held in major cities, and it becomes increasingly important to accurately predict the flow of people at these events. Accurate visitor flow prediction can help convention and exhibition managers make reasonable resource allocation before the event, such as security personnel, guide signs, emergency facilities, etc., to ensure the smooth running of the event. In addition, accurate visitor flow prediction can improve service quality, enhance visitors' experience, and avoid congestion and safety problems caused by too many people [3, 4]. The ability to respond to unexpected events and emergencies can also be improved, thus ensuring the safe conduct of convention and exhibition activities. Therefore, the study of crowd prediction for exhibition visitor scenarios not only has significant practical application value, but also has far-reaching significance for urban management and public safety.

Neural networks, especially deep learning models, show great potential in processing complex data patterns and capturing nonlinear relationships in data. With the improvement of computing power and the development of big data technology, neural networks have been widely used in various prediction tasks. In the field of people flow prediction, traditional time series models such as ARIMA and Long Short-Term Memory Network (LSTM) perform well in some scenarios, but they are often incompetent in dealing with large-scale and complex people flow data [5]. Graph Neural Networks (GNN) [6, 7] are able to effectively utilise the connectivity information between nodes and edges to capture the spatio-temporal dependencies in data by representing the data as a graph structure. Graph neural networks can handle complex spatial relationships and improve prediction accuracy through multi-level information transfer and feature aggregation [8, 9]. In addition, the Transformer [10, 11] model is able to model long-range temporal dependencies through the self-attention mechanism, improving the accuracy of time series prediction. Combining graph neural networks and Transformer models can make full use of their advantages in both spatial and temporal dimensions to provide a more accurate and reliable solution for people flow prediction. The aim of this paper is to combine the advantages of graph neural networks and Transformer models to propose a convention visitor flow prediction method, which improves the accuracy and efficiency of prediction by comprehensively utilising spatial and temporal features.

1.1. **Related work.** Pedestrian flow forecasting has been an important research direction in urban management and business decision-making. Traditional methods of people flow forecasting mainly include statistical modelling and time series analysis. Firstly, statistically based methods have been widely used in early pedestrian flow forecasting. Nassar [12] proposed a simple linear regression model for predicting people flow in buildings. Their study showed that although the linear regression model can reflect the trend of people flow to a certain extent, its prediction accuracy and applicability are limited when facing complex and dynamically changing people flow data. Gutiérrez et al. [13] proposed a multivariate linear regression model to improve the prediction accuracy by introducing multiple independent variables, such as weather, holidays and other factors. However, with the increase in data dimensionality and complexity, the multiple linear regression model faces the problems of dimensional catastrophe and overfitting.

Time series analysis methods are another common type of people flow forecasting methods. The ARIMA model proposed by Mondal et al. [14] achieves effective forecasting of time series by smoothing, differencing, and estimating model parameters for time series data. Although the ARIMA model performs well in univariate time series forecasting, it has limited ability to handle multivariate time series data. To overcome this problem, Dabral and Murry [15] proposed the Seasonal ARIMA (SARIMA) model, which improves the accuracy of forecasting by taking seasonality into account. However, the SARIMA model is still deficient in handling nonlinear and non-stationary data.

In general, the traditional methods of people flow prediction are able to capture the changing law of people flow to a certain extent, but their prediction accuracy and applicability are limited when facing complex, nonlinear and dynamically changing data. Therefore, seeking more advanced and flexible prediction methods has become a hot spot of current research.

With the increase in computing power and the development of data science, machine learning and neural network based methods have become the main research direction for people flow prediction in recent years. These methods are capable of handling complex nonlinear relationships and extracting useful features from large amounts of data. However, current research still faces some challenges. Firstly, methods based on traditional neural networks have been widely used in early research. Dudek [16] proposed a prediction model based on neural networks and linear regression. Subsequently, Wang et al. [17] proposed the Long Short-Term Memory (LSTM) network, which can effectively capture long-range dependencies in time series by introducing memory units and gating mechanisms in neural networks. LSTM performs well in several application scenarios, but its computational complexity and training time in dealing with large-scale data are still a challenge.

To further improve the prediction performance, researchers have started to combine deep learning with graph structure data. Graph Convolutional Network (GCN) [18] is able to effectively capture the relationships between nodes and their neighbours by performing convolutional operations on graph structures. GCN performs well in the fields of social networks, traffic networks, etc., but it has limitations in dealing with dynamically changing graph structure data. Hu et al. [19] proposed the Graph Attention Network (GAT), which enhances the model's ability to perceive the importance of different neighbouring nodes by introducing the attention mechanism, further improving the prediction accuracy. However, GAT still has high computational complexity when dealing with long time series and large-scale graph data. The introduction of the Transformer model provides a new idea for people flow prediction. The Transformer model proposed by Ding et al. [20] achieves efficient modelling of long-range dependencies through the self-attention mechanism. The Transformer is used in the natural language processing and time series prediction, but it lacks the ability to explicitly model node relationships when dealing with graph-structured data.

1.2. **Motivation and contribution.** Existing methods for people flow prediction mainly focus on static and simple scenarios, and show certain limitations when facing complex dynamic changes and spatially dependent visitor flow prediction for conventions and exhibitions. Traditional time series models and neural network models are inadequate in capturing the spatial and temporal dependencies, which makes it difficult to effectively cope with the complex fluctuations of visitor flow in convention and exhibition scenarios. Traditional models are prone to overfitting and computational complexity problems when dealing with multivariate time series and high-dimensional data, leading to insufficient prediction accuracy. In addition, most of the existing studies ignore the important role of spatial relationships in forecasting and cannot make full use of the spatial dependency information between regions.

In order to solve the above problems, this paper proposes a convention visitor flow prediction method based on Spatio-Temporal Ordinary Differential Reformer Graph Neural Network (ST-EDRGNN), which aims to improve the accuracy and efficiency of the prediction. The main innovations and contributions of this work include:

1. **Spatial feature learning module based on ordinary differential equations:** Aiming at the limitations of traditional models in dealing with spatial relationships, this paper proposes a spatial feature learning module based on ordinary differential equations [21]. By continuously modelling the propagation process of node features, this module avoids the excessive smoothing problem in traditional graph neural networks due to the increase in the number of layers, and thus captures the complex dependencies between the visitor flows in each region more accurately.
2. **Efficiency improvement via Reformer architecture:** Existing methods tend to have high computational complexity and long training time when dealing with multivariate time series and high-dimensional data. In this paper, by introducing the Locally Sensitive Hash (LSH) Attention mechanism and the Reversible Residual Network (RRN) in the Reformer model, the computational complexity and memory occupation are effectively reduced, which improves the training efficiency and prediction speed of the model. This improvement is especially significant when dealing with large-scale and dynamically changing convention visitor traffic data.

2. Related Technical Studies.

2.1. Principles of Graph Neural Networks.

Graph Neural Networks (GNN) are a class of neural network models designed to process graph-structured data [22]. Since many real-world data, such as social networks and traffic networks, can be represented as graphs, GNNs are widely applied in these fields.

2.1.1. *Basic structure of GNN.*

The basic structure of a GNN consists of *nodes*, *edges*, and an *adjacency matrix*. Nodes represent entities in the graph (e.g., users in a social network or intersections in a traffic network), edges represent connections or relationships between nodes, and the adjacency matrix describes the connectivity between nodes.

In GNN, information is propagated between nodes and edges, and features in the graph structure are captured by iteratively updating the states of nodes. At each layer, a node receives information from its neighbours and updates its own state based on the aggregated neighbour information. Mathematically, this process can be formulated as:

$$h_i^{(k)} = \mathrm{UPDATE}\left(h_i^{(k-1)}, \mathrm{AGGREGATE}\left(\left\{h_j^{(k-1)} \mid j \in \mathcal{N}(i)\right\}\right)\right), \tag{1}$$

where $h_i^{(k)}$ denotes the feature vector (state) of node i at the k-th layer, and $\mathcal{N}(i)$ represents the set of neighbouring nodes of i. AGGREGATE$(\cdot)$ is the aggregation function, while UPDATE$(\cdot)$ is the update function.

A typical aggregation function can be expressed as:

$$\mathrm{AGGREGATE}\left(\left\{h_j^{(k-1)} \mid j \in \mathcal{N}(i)\right\}\right) = \sum_{j \in \mathcal{N}(i)} h_j^{(k-1)}, \tag{2}$$

and the update step is often performed as:

$$h_i^{(k)} = \sigma\left(W \cdot \mathrm{AGGREGATE}\left(\left\{h_j^{(k-1)} \mid j \in \mathcal{N}(i)\right\}\right) + b\right), \tag{3}$$

where W and b denote the weight matrix and bias vector, respectively, and $\sigma(\cdot)$ represents a nonlinear activation function such as ReLU or sigmoid.

To prevent numerical instability and overfitting, node features are usually normalised in GNNs. Specifically, by normalising the adjacency matrix A using the degree matrix D, we obtain the symmetric normalised adjacency matrix $\hat{A}$:

$$\hat{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}, \tag{4}$$

where D is the degree matrix, and the diagonal element D_{ii} denotes the degree of node i.

2.1.2. *Graph Convolutional Networks.* Graph Convolutional Networks (GCN) are one of the most widely used variants of GNNs, which generalise the convolution operation from Euclidean to non-Euclidean domains [23]. While traditional Convolutional Neural Networks (CNNs) operate on regular grids (e.g., images), GCNs are capable of processing graph-structured data with arbitrary topology.

In a GCN, each node updates its feature representation by aggregating the features of its neighbouring nodes via a graph convolution operation, formulated as:

$$H^{(k+1)} = \sigma\left(\hat{D}^{-\frac{1}{2}} \hat{A} \hat{D}^{-\frac{1}{2}} H^{(k)} W^{(k)}\right), \tag{5}$$

where $\hat{A} = A + I$ is the adjacency matrix with added self-loops, $\hat{D}$ is its corresponding degree matrix, $H^{(k)}$ denotes the node feature matrix at the k-th layer, and $W^{(k)}$ is the trainable weight matrix of that layer.

The first task of GCN is *feature propagation*, which transmits a node's features to its neighbours. This propagation process can be expressed as:

$$H' = AH, \tag{6}$$

where H is the current node feature matrix and H' is the propagated feature matrix.

The propagated features are then transformed linearly and passed through a nonlinear activation function [24]:

$$H^{(k+1)} = \sigma\left(H' W^{(k)}\right). \tag{7}$$

To maintain numerical stability during propagation, a normalised adjacency matrix $\tilde{A}$ is employed:

$$H' = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} H. \tag{8}$$

Through iterative propagation and aggregation, GNNs can efficiently capture complex dependencies and relationships in graph-structured data. As a representative model, GCN enables effective feature extraction and representation learning by performing convolution operations directly on graphs.

2.2. **Principles of Transformer.** The Transformer model, as an innovative deep learning architecture, is particularly suitable for processing sequential data, such as Natural Language Processing (NLP) tasks. Compared with traditional Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) models, Transformer achieves more efficient parallel computation and longer sequence-dependent modelling through the self-attention mechanism [24, 25].

2.2.1. *The basic architecture of Transformer.* The Transformer model consists of an Encoder and a Decoder, each composed of several identical layers stacked on top of each other. Each Encoder and Decoder layer contains the following two main parts.

(1) Multi-head self-attention mechanism: the self-attention mechanism captures the dependencies within a sequence by calculating the similarity of the features at each position

in the input sequence to the features at all other positions [26]. The computation of the self-attention mechanism is as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{9}$$

where Q, K, and V denote the Query matrix, Key matrix and Value matrix respectively; d_k is the dimension of the key vector.

The computation of the multi-head self-attention mechanism is shown as follows:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(head_1, head_2, \dots, head_h)W^O, \tag{10}$$

$$head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V), \tag{11}$$

where W_i^Q, W_i^K, W_i^V and W^O are learnable parameter matrices.

(2) Feed-Forward Neural Network (FFN): the feed-forward neural network in each encoder and decoder layer is a fully connected network applied independently to each position.

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2, \tag{12}$$

where W_1, W_2 are weight matrices and b_1, b_2 are bias vectors.

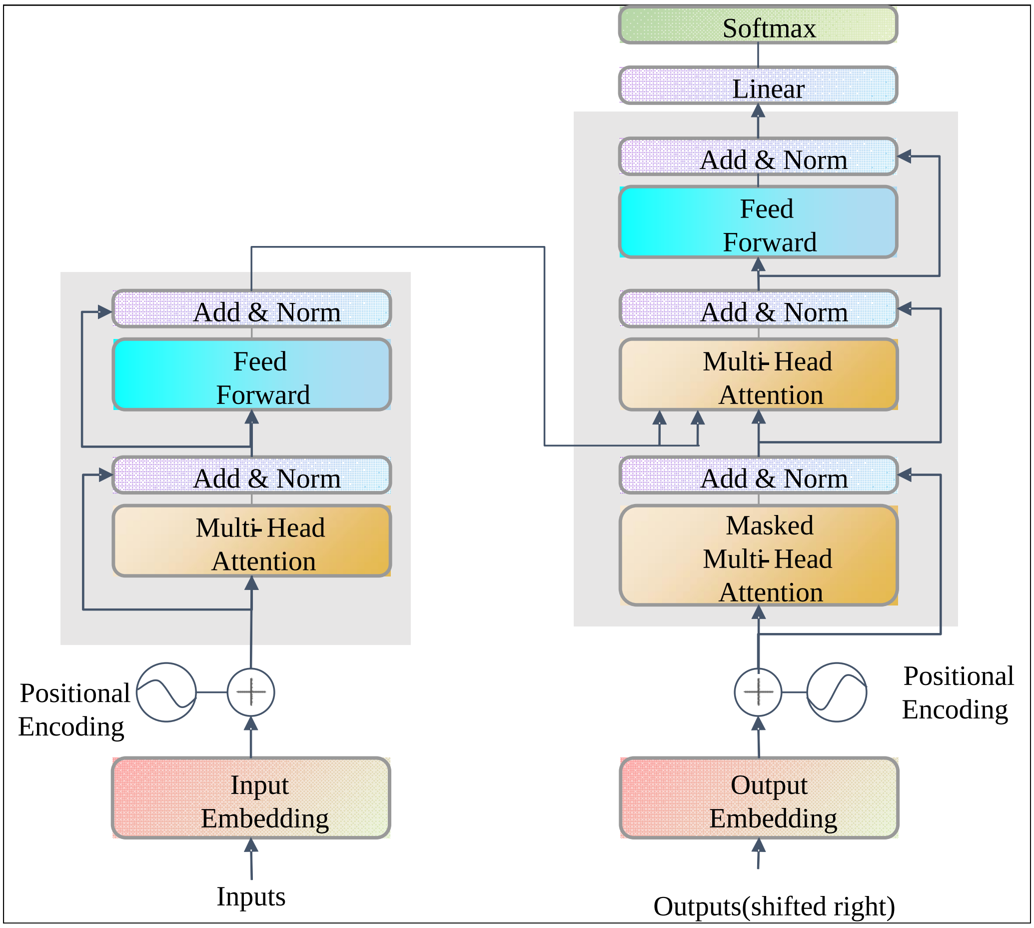

Figure 1. Structure of the Transformer.

2.2.2. *Position encoding.* Since the Transformer model does not have inbuilt sequential information, position encoding is required to provide information about each position in the sequence [27]. Position Encoding (PE) can be generated by sine and cosine functions:

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right), \tag{13}$$

$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right), \tag{14}$$

where pos denotes the position, i denotes the dimension index, and d_{model} is the dimension of the model.

2.2.3. *Encoder-decoder architecture.* The encoder maps the input sequence to a set of consecutive representations, and the decoder generates the output sequence based on these representations. Information is transferred between the encoder and decoder through the self-attention mechanism and the interactive attention mechanism. The output of the encoder is used as one of the inputs to the decoder. The decoder also uses the self-attention mechanism to process the outputs that have been generated earlier, ensuring that each step of the generation takes into account the entire input sequence and the previous outputs. The structure of the Transformer is shown in Figure 1.

3. **ST-EDRGNN based convention visitor flow prediction.**

3.1. **Overall framework design.** The spatio-temporal constant differential Reformer graph neural network (ST-EDRGNN) proposed in this paper consists of two key components: the S-GNN and the Reformer layer. The S-GNN layer is used to capture spatial relationships between different areas of the convention centre, while the Reformer layer is used to capture the long-range temporal dependencies in visitor flow data. The overall framework is shown in Figure 2.

First, the S-GNN layer is responsible for node representation and initialisation. Each node represents a region of a convention centre. The initialised node feature matrix X contains visitor flow data, historical data and other relevant features for each region. Then, the node feature representation H is updated by capturing the spatial relationship between each region through GCN. The core of the graph convolution operation is to aggregate and transform the features of each node so that the node features contain information about its neighbouring nodes, as in Equation (5).

The Reformer layer is responsible for long-range temporal dependency modelling. The Reformer layer captures long-range temporal dependencies in visitor traffic data through the Local Sensitive Hash (LSH) Attention mechanism and reversible residual networks. Compared with the traditional Transformer model, Reformer has higher efficiency and lower memory footprint in processing long sequence data.

Finally, the prediction layer consists of a fully connected network and output. The features processed by the S-GNN and Reformer layers are input to the fully connected layer, which outputs the predicted visitor traffic. The fully connected network maps the features to the target output space to achieve the prediction of visitor traffic at future moments.

3.2. **Data preprocessing.** Firstly, the visitor flow data of the convention centre needs to be transformed into graph structure data and feature matrix. It is assumed that the convention centre can be divided into several regions, and the visitor flow of each region can be represented as a node. The connection between nodes indicates the direct access or spatial proximity between regions.

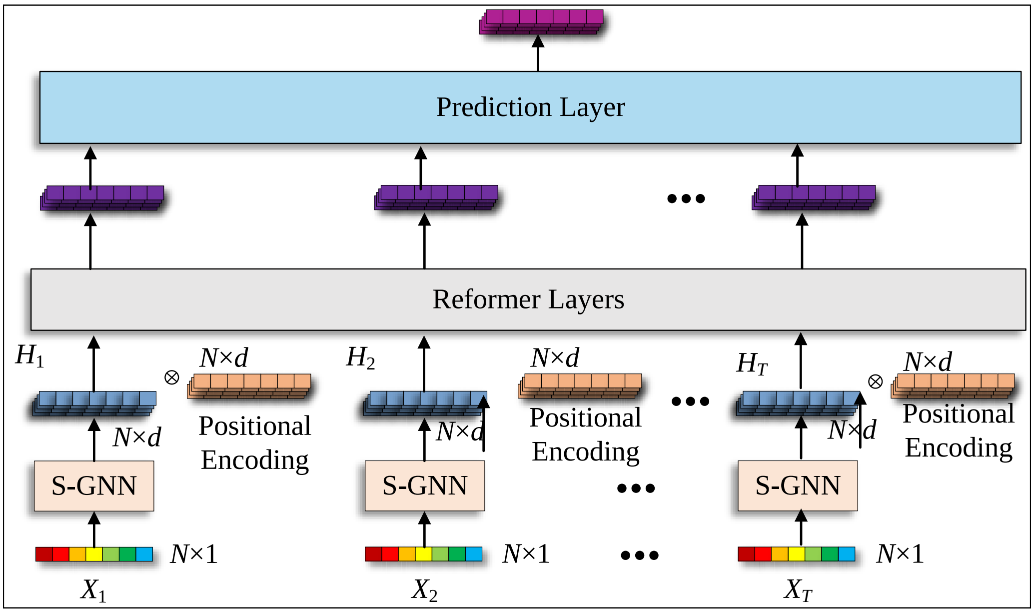

Figure 2. Framework of the ST-EDRGNN.

The neighbourhood matrix $A \in \mathbb{R}^{N\times N}$ is used to describe the connectivity relationship between nodes, where N denotes the number of regions. If there is a direct channel or spatial proximity relationship between region i and region j, then $A_{ij} = 1$; otherwise, $A_{ij} = 0$.

The feature matrix $X \in \mathbb{R}^{N\times F}$ describes the features of each node, where F denotes the dimension of the feature. For example, the features of each node may include visitor traffic at the current time step, historical visitor traffic, weather information, and so on.

Through the data preprocessing steps described above, the raw visitor flow data is transformed into an input form suitable for GNN, enabling the model to make full use of spatial and temporal information for prediction.

3.3. **Spatial feature learning module based on ordinary differential equation.** The spatial feature learning module is designed based on Ordinary Differential Equation (ODE) to capture the complex spatial relationships between different areas of the convention centre. By modelling ODE, we are able to model the propagation of spatial information as a continuous dynamic process, and thus describe the dependency of visitor flow between different zones more accurately.

First, given a graph $G = (V, E)$ representing the regions of the convention centre, where V denotes the set of nodes, each representing a region, and E denotes the set of edges representing the connections between regions. Let the adjacency matrix be $A \in \mathbb{R}^{N\times N}$ and the node feature matrix be $X \in \mathbb{R}^{N\times F}$, where N is the number of nodes, and F is the dimension of the features.

Based on GCN, we introduce ODE to model the change of node features. First, we define the rate of change of node features at time t:

$$\frac{dH(t)}{dt} = H(t) \times (A - I) + H(t) \times (U - I) + H(0), \tag{15}$$

where $H(t)$ denotes the node identity matrix at time t, I is the unit matrix, U is the time transformation matrix, and $H(0)$ is the initial node identity matrix.

The above ordinary differential equation can be further converted into an integral expression to describe the evolution of nodal features in the time dimension:

$$H(t) = \int_0^t \left(H(0) \times \hat{A}_\tau \times U_\tau \times W_\tau \right) d\tau, \tag{16}$$

where $\hat{A}_\tau$, U_τ and W_τ denote the representations of the regularised adjacency matrix, the temporal transformation matrix and the eigen-transformation matrix at time τ, respectively. By further simplification and transformation, we obtain:

$$\frac{dH(t)}{dt} = H(t) \times (\hat{A} - I) + H(t) \times (U - I) + H(0). \tag{17}$$

Based on the above ordinary differential equations, we define the graph ordinary differential equation learning framework:

$$H(t) = ODESolver\left(\frac{dH(t)}{dt}, H(0), t \right), \tag{18}$$

where *ODESolver* denotes the numerical solver used to solve the ordinary differential equations. In our model, the Eulerian solver is used to compute the evolution of node features over time.

The spatial feature learning module is able to effectively capture the spatial dependencies between the various regions of the convention centre. By continuous modelling of ordinary differential equations, we avoid the problem of excessive smoothing due to the increase in the number of layers in the traditional graph neural network, thus improving the ability to capture complex spatial relationships. By modelling the propagation process of node features as a continuous dynamical system, we are able to reflect more accurately the dynamic changes of visitor flow in each region.

3.4. **Reformer-based temporal feature learning module.** The temporal feature learning module uses the Reformer model to model time-series data of convention visitor traffic. The Reformer model is an improved version of the Transformer model designed to handle long-series data, and reduces computational complexity and memory consumption by means of the LSH Attention mechanism and Reversible Residual Network (RRN).

The core of the Reformer model lies in its improvement on the traditional Transformer model, which consists of two main aspects.

(1) LSH Attention: With LSH Attention, Reformer can effectively reduce the computational complexity of long sequence data, reducing the complexity of the attention mechanism in the original Transformer from $O(L^2)$ to $O(L \log L)$, where L is the sequence length.

LSH Attention simplifies the computation of the attention mechanism by splitting the input sequence into different hash buckets through a hash function. The specific steps are as follows:

(a) Sequence partitioning: the input sequences Q and K are partitioned into different hash buckets by a hash function such that similar elements are grouped into the same bucket [28].

(b) Bucket sorting: within each hash bucket, elements are sorted according to their position in the original sequence.

(c) Chunking: the sorted elements are divided into multiple chunks for parallel processing.

(d) Calculate attention: for each element in the block, calculate the attention weights and perform a weighted sum.

The specific calculation for LSH Attention is as follows:

$$\text{Attention}(q) = \frac{\sum_{i:h(k_i)=h(k_q)} v_i}{\sum_{j:h(k_j)=h(k_q)} 1}. \tag{19}$$

(2) Reversible Residual Network: Reversible Residual Network allows recalculation of the intermediate activation values during backpropagation, thus significantly reducing the memory footprint and improving the training efficiency of the model. In the Reformer model, the Reversible Residual Network allows the recalculation of intermediate activation values during backpropagation, thus reducing the memory footprint. Its forward propagation and backpropagation are computed respectively as follows:

Forward propagation:

$$y_1 = x_1 + \text{Att}(x_2), \tag{20}$$

$$y_2 = x_2 + \text{FFN}(y_1), \tag{21}$$

Backward propagation:

$$x_1 = y_1 - \text{Att}(x_2), \tag{22}$$

$$x_2 = y_2 - \text{FFN}(y_1), \tag{23}$$

where Att denotes the attention mechanism and FFN denotes the feed-forward neural network.

In convention visitor traffic forecasting, the Reformer model is used to capture long-range temporal dependencies in time-series data. Firstly, the convention visitor traffic data is transformed into serial form and input into the Reformer model for processing. With LSH Attention and the reversible residual network, the model is able to efficiently capture the long-range dependencies and change patterns in the visitor flow data.

The module implementation steps are as follows.

Step 1: Transform the preprocessed convention visitor traffic data into an input format suitable for the Reformer model, containing time-series features of visitor traffic.

Step 2: Encode the input data and extract temporal features using the Reformer model. Specifically, it includes the parallel computation of LSH Attention and the efficient processing of reversible residual network.

Step 3: The encoded temporal features are fed into the subsequent prediction layer to enable prediction of unvisited traffic.

Through the above steps, the Reformer model is able to effectively capture the dynamic change patterns in the convention visitor flow data in the temporal feature learning module, thus providing accurate temporal features for subsequent flow prediction.

4. Analysis of experimental results.

4.1. Experimental environment and dataset. In order to verify the effectiveness of ST-EDRGNN in convention visitor traffic prediction, we designed a series of experiments. The experiments were conducted in well-configured hardware and software environments, and representative real datasets were selected for evaluation.

For the experimental environment, we used a powerfully configured computer. The operating system is Windows 10 with an Intel Core i7-8750H @ 2.20GHz central processing unit (CPU) and an NVIDIA GeForce RTX 3090 Ti 24GB graphics processor (GPU). The whole machine has 16GB RAM and 1TB hard drive, which provides sufficient computing resources and storage space for data processing and model training. The programming language chosen is Python 3.10, and deep learning frameworks such as TensorFlow 1.14.0

and PyTorch are used. The integrated development environment (IDE) is Anaconda3-Windows-x86_64, which facilitates package management and environment configuration. The experimental environment is shown in Table 1.

Table 1. Experimental environment

Hardware/Software	**Configuration**
Operating system	Windows 10
Central Processing Unit (CPU)	Intel Core i7-8750H @ 2.20GHz
Graphics Processor (GPU)	NVIDIA GeForce RTX 3090 Ti 24GB
Random access memory (RAM)	16GB
Hard drive	1TB
Programming language	Python 3.10
Deep learning frameworks	TensorFlow 1.14.0, PyTorch
Integrated Development Environment (IDE)	Anaconda3-Windows-x86_64

In terms of datasets, we chose two main datasets for our experiments. Dataset A is derived from the visitor flow records of a large convention centre, spanning from 1 January 2022 to 30 June 2022, a period of 6 months, with data recorded every 5 minutes. The features of this dataset include visitor flow, weather information, holiday information, etc., and the total number of records is 51,840 (288 time slices per day for 6 months). Dataset B is derived from the visitor flow records of another large convention centre, spanning the period from 1 July 2021 to 31 December 2021, for a period of 6 months, again with data recorded every 5 minutes, and features such as visitor flow, weather information, holiday information, etc., with a total record count of 51,840 as well.

Before the experiment, we carried out a comprehensive pre-processing of the data. Firstly, missing data points were filled in using linear interpolation to ensure the continuity and completeness of the data. Secondly, in order to eliminate the difference in magnitude between different features, all numerical features were normalised so that their value domains were between 0 and 1. Finally, the adjacency matrix A is constructed based on the actual connectivity between the various areas of the convention and exhibition centre, and the feature matrix X is constructed based on the visitor flow data.

4.2. **Assessment indicators.** In order to ensure the objectivity and accuracy of the final experimental results, the following three assessment metrics were used in this study to evaluate the performance of the model: Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). These evaluation metrics are widely used in time series forecasting tasks and can effectively measure the forecasting performance of the model.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{24}$$

where y_i denotes the actual visitor flow data, $\hat{y}_i$ denotes the predicted visitor flow data, and n is the sample size.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{25}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%. \tag{26}$$

4.3. **Comparison of the performance of different models.** In order to validate the effectiveness of the ST-EDRGNN based model, we compared it with several common baseline models. These baseline models include the ARIMA, LSTM, the Transformer model, as well as the traditional GCN and the GCN-LSTM. We performed a detailed performance comparison on both datasets and the results are shown in Table 2 and Table 3.

Table 2. Dataset A

Model	**RMSE**	**MAE**	**MAPE (%)**
ARIMA	2.310	1.870	4.51
LSTM	2.105	1.725	4.25
Transformer	2.050	1.690	4.15
GCN	2.300	1.860	4.45
GCN-LSTM	1.980	1.640	4.00
ST-EDRGNN	1.750	1.480	3.85

Table 3. Dataset B

Model	**RMSE**	**MAE**	**MAPE (%)**
ARIMA	2.350	1.900	4.60
LSTM	2.150	1.750	4.30
Transformer	2.080	1.700	4.20
GCN	2.320	1.880	4.50
GCN-LSTM	1.990	1.660	4.05
ST-EDRGNN	1.770	1.500	3.90

On dataset A, the ST-EDRGNN model has an RMSE of 1.750, which is significantly better than the other models, and the closest is the GCN-LSTM model with an RMSE of 1.980, which is 0.230 better than the ST-EDRGNN. On dataset B, the ST-EDRGNN has an RMSE of 1.770, which is also better than the other models. On dataset A, ST-EDRGNN has a MAE of 1.480, which is significantly lower than the other models, with the closest being GCN-LSTM at 1.640, which is 0.160 higher than ST-EDRGNN. On dataset B, ST-EDRGNN has a MAE of 1.500. On dataset A, ST-EDRGNN has a MAPE of 3.85%, which is significantly lower than the other models, and the closest is 4.00% for GCN-LSTM, which is 0.15 percentage points higher than ST-EDRGNN. On dataset B, ST-EDRGNN has a MAPE of 3.90%, again better than the other models. The lower MAPE value indicates that ST-EDRGNN also has a significant advantage in terms of the relative error of the prediction results.

It can be seen that the ST-EDRGNN model significantly outperforms other models in all three metrics, RMSE, MAE and MAPE, especially in capturing spatio-temporal features. This is attributed to the fact that the ST-EDRGNN model combines ordinary differential equations and the Reformer model, which is able to capture the spatio-temporal dependencies in the visitor flow data more effectively.

5. **Conclusion.** In this paper, a convention and exhibition visitor flow prediction method based on ST-EDRGNN is proposed, which effectively solves the limitations of traditional prediction models in dealing with complex spatio-temporal dependencies and long time dependencies. By introducing a graph neural network, the model is able to capture the spatial relationships between different areas of the convention and exhibition centre more accurately, and combined with a spatial feature learning module based on ordinary

differential equations, it improves the dynamic modelling capability of spatial information. In addition, the Reformer model utilises LSH Attention and Reversible Residual Network (RRN) to further enhance its ability to capture long time series data, ensuring the stability and robustness of the prediction results. The following conclusions can be drawn from the experiments conducted on two real convention datasets:

(1) The spatial feature learning module for graph neural networks combined with ordinary differential equations significantly improves the ability to model complex spatial dependencies.

(2) The ST-EDRGNN model significantly outperforms other baseline models in the three indicators of RMSE, MAE and MAPE, reflecting its superiority in the prediction of convention visitor flow. Specific experimental results show that the RMSE of the ST-EDRGNN model is 1.750 and 1.770, the MAE is 1.480 and 1.500, and the MAPE is 3.85% and 3.90%, which are all superior to the traditional models such as ARIMA, LSTM and GCN-LSTM.

Although the Reformer model has high computational efficiency and memory utilisation when dealing with long time series data, this paper failed to demonstrate this in detail in the experimental part. Future research should verify the computational efficiency advantage of the Reformer model on large-scale datasets through more experiments.

REFERENCES

[1] Y. Zhou, H. Chen, J. Li, Y. Wu, J. Wu, and L. Chen, "Large-scale station-level crowd flow forecast with ST-UNet," *ISPRS International Journal of Geo-Information*, vol. 8, no. 3, 140, 2019.

[2] Z. Guo, X. Zhao, Y. Chen, W. Wu, and J. Yang, "Short-term passenger flow forecast of urban rail transit based on GPR and KRR," *IET Intelligent Transport Systems*, vol. 13, no. 9, pp. 1374–1382, 2019.

[3] Y. Jin, Y. Gao, P. Wang, J. Wang, and L. Wang, "Improved manpower planning based on traffic flow forecast using a historical queuing model," *IEEE Access*, vol. 7, pp. 125101–125112, 2019.

[4] D. Yang, K. Chen, M. Yang, and X. Zhao, "Urban rail transit passenger flow forecast based on LSTM with enhanced long-term features," *IET Intelligent Transport Systems*, vol. 13, no. 10, pp. 1475–1482, 2019.

[5] T.-Y. Wu, H. Li, S. Kumari, and C.-M. Chen, "A spectral convolutional neural network model based on adaptive Fick's law for hyperspectral image classification," *Computers, Materials & Continua*, vol. 79, no. 1, pp. 19–46, 2024.

[6] Y. Ma, Y. Peng, and T.-Y. Wu, "Transfer learning model for false positive reduction in lymph node detection via sparse coding and deep learning," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 2, pp. 2121–2133, 2022.

[7] F. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G. Ding, P. Mei, and L. Liu, "Application of quantum genetic optimization of LVQ neural network in smart city traffic network prediction," *IEEE Access*, vol. 8, pp. 104555–104564, 2020.

[8] W. Fan, Y. Ma, Q. Li, J. Wang, G. Cai, J. Tang, and D. Yin, "A graph neural network framework for social recommendations," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 5, pp. 2033–2047, 2020.

[9] N. A. Asif, Y. Sarker, R. K. Chakrabortty, M. J. Ryan, M. H. Ahamed, D. K. Saha, F. R. Badal, S. K. Das, M. F. Ali, and S. I. Moyeen, "Graph neural network: A comprehensive review on non-Euclidean space," *IEEE Access*, vol. 9, pp. 60588–60606, 2021.

[10] E. I. Amoiralis, M. A. Tsili, and A. G. Kladas, "Transformer design and optimization: a literature survey," *IEEE Transactions on Power Delivery*, vol. 24, no. 4, pp. 1999–2024, 2009.

[11] S. Luo, S. Li, S. Zheng, T.-Y. Liu, L. Wang, and D. He, "Your transformer may not be as powerful as you expect," *Advances in Neural Information Processing Systems*, vol. 35, pp. 4301–4315, 2022.

[12] K. Nassar, "A model for assessing occupant flow in building spaces," *Automation in Construction*, vol. 19, no. 8, pp. 1027–1036, 2010.

[13] J. Gutiérrez, O. D. Cardozo, and J. C. García-Palomares, "Transit ridership forecasting at station level: an approach based on distance-decay weighted regression," *Journal of Transport Geography*, vol. 19, no. 6, pp. 1081–1092, 2011.

[14] P. Mondal, L. Shit, and S. Goswami, "Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices," *International Journal of Computer Science, Engineering and Applications*, vol. 4, no. 2, pp. 13, 2014.

[15] P. Dabral, and M. Z. Murry, "Modelling and forecasting of rainfall time series using SARIMA," *Environmental Processes*, vol. 4, no. 2, pp. 399–419, 2017.

[16] G. Dudek, "Pattern-based local linear regression models for short-term load forecasting," *Electric Power Systems Research*, vol. 130, pp. 139–147, 2016.

[17] T. Wang, R. G. Qiu, and M. Yu, "Predictive modeling of the progression of Alzheimer's disease with recurrent neural networks," *Scientific Reports*, vol. 8, no. 1, pp. 9161, 2018.

[18] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, no. 1, pp. 1–23, 2019.

[19] B. Hu, K. Guo, X. Wang, J. Zhang, and D. Zhou, "RRL-GAT: Graph attention network-driven multilabel image robust representation learning," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9167–9178, 2021.

[20] Y. Ding, M. Jia, Q. Miao, and Y. Cao, "A novel time–frequency transformer based on self–attention mechanism and its application in fault diagnosis of rolling bearings," *Mechanical Systems and Signal Processing*, vol. 168, 108616, 2022.

[21] A. Lou, D. Lim, I. Katsman, L. Huang, Q. Jiang, S. N. Lim, and C. M. De Sa, "Neural manifold ordinary differential equations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 17548–17558, 2020.

[22] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, "Graph neural networks in recommender systems: a survey," *ACM Computing Surveys*, vol. 55, no. 5, pp. 1–37, 2022.

[23] V. P. Dwivedi, C. K. Joshi, A. T. Luu, T. Laurent, Y. Bengio, and X. Bresson, "Benchmarking graph neural networks," *Journal of Machine Learning Research*, vol. 24, no. 43, pp. 1–48, 2023.

[24] C. Gao, Y. Zheng, N. Li, Y. Li, Y. Qin, J. Piao, Y. Quan, J. Chang, D. Jin, and X. He, "A survey of graph neural networks for recommender systems: challenges, methods, and directions," *ACM Transactions on Recommender Systems*, vol. 1, no. 1, pp. 1–51, 2023.

[25] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, and Y. Xu, "A survey on vision transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 87–110, 2022.

[26] N. Engel, V. Belagiannis, and K. Dietmayer, "Point transformer," *IEEE Access*, vol. 9, pp. 134826–134840, 2021.

[27] C. Zhu, W. Ping, C. Xiao, M. Shoeybi, T. Goldstein, A. Anandkumar, and B. Catanzaro, "Long-short transformer: efficient transformers for language and vision," *Advances in Neural Information Processing Systems*, vol. 34, pp. 17723–17736, 2021.

[28] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: point cloud transformer," *Computational Visual Media*, vol. 7, pp. 187–199, 2021.