Completion and augmentation-based spatiotemporal deep learning approach for short-term metro origin-destination matrix prediction under limited observable data

Jiexia Ye1 · Juanjuan Zhao2 · Furong Zheng2 · Chengzhong Xu3

Received: 31 March 2022 / Accepted: 21 September 2022 / Published online: 9 October 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract
Accurate prediction of short-term origin-destination (OD) matrix is crucial for operations in metro systems. Recently, some deep learning-based models have been proposed for OD matrix forecasting in ride-hailing or high-way scenarios. However, the metro OD matrix forecasting receives less attention, and it has different prior knowledge and complex spatiotemporal contextual setting; for example, the sparse destination distribution and the incomplete OD matrices collection in recent time intervals due to unfinished trips before the predicted time interval. This paper designs a deep learning approach for metro OD matrix prediction by addressing the recent destination distribution availability, augmenting the flow presentation for each station, and digging out the global spatial dependency and multiple temporal scale correlations in the mobility patterns of metro passengers. Specifically, it first proposes to complete the recent OD matrices by combining some empirical knowledge including the historical mobility pattern and arrival time distribution. Then, it learns the complementary spatiotemporal contextual features by embedding methods to enrich the station representation. Finally, it captures global mobility trend of metro passengers at each origin station through aggregating the trend of all other origin stations by self-attention mechanism since the mobility synchronizes among stations from spatial perspective. Three temporal convolutional networks are leveraged to extract three temporal trends in passenger mobility data, i.e., recent trend, daily trend, and weekly trend. Smart card data from Shenzhen and Hangzhou metro systems are utilized to demonstrate the superiority of our model over other competitors.

Keywords Origin-destination matrix prediction · Destination distribution availability · Self-Attention mechanism · Temporal convolution network

1 Introduction
OD prediction in urban rail transit (URT) is one of the critical tasks in intelligent transportation systems. The prediction result can benefit both individuals and traffic operators. For operators, such knowledge could help them better understand the real-time passenger flow distribution in whole subway network, thus supporting decisions on network management tasks, such as congestion control and traffic resources deployment. For individuals, OD prediction can provide more accurate suggestions for them to arrange their route plannings, thus improving travel experience. However, the short-term OD flow prediction in metro systems confronts some important challenges as follows:

Juanjuan Zhao
jj.zhao@siat.ac.cn

Jiexia Ye
zumri.jiexiaye@um.edu.mo

Furong Zheng
fr.zheng@siat.ac.cn

Chengzhong Xu
czxu@um.edu.mo

1 Guangdong-Hong Kong-Macao Joint Laboratory of Human-Machine Intelligence-Synergy Systems, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

2 Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

3 State Key Lab of IOTSC, Department of Computer Science, University of Macau, Macao, China
1.1 Flow distribution sparsity

Passenger flow prediction at OD pair level is much larger in scale than that at station level. Take Shenzhen subway with 118 stations for example; only 118 elements need to be predicted if we want to forecast the passenger flow at station level while there are 13,924 (i.e., the square number of 118) predicted elements in OD matrix prediction. The high data dimensionality comes with sparsity. The limited inflow/outflow of each station disperses across the whole network, leading to sparse passenger flow distribution, i.e., many OD pairs with small or even zero values. According to statistics, nearly half of OD pairs in Shenzhen metro dataset have lower than 4 passengers at half-an-hour time granularity, usually attributed to randomly generated trips which significantly decrease the regularity of OD flows. The sparse destination distribution largely increases the difficulty to predict the OD flows accurately.

1.2 Incomplete recent destination distribution

Most of the existing OD prediction methods are based on the premise that the complete traffic data can be obtained in time. For instance, the rail-hailing OD matrices at previous steps can be obtained for prediction. However, the metro OD matrices at latest several time intervals, namely recent OD matrices, are likely to be collected partially and thus incomplete before the predicted time interval. For any passenger trip, there is a time gap between the entrance and exit of the subway stations. For passengers entering the origin station during a past time interval, it’s likely that part of them reach their destinations before the predicted time interval and the remainder finish their journeys after the target time interval. Therefore, destination distributions collected in recent time intervals are likely to be incomplete, and the accurate real-time passenger movement information is unavailable.

1.3 Multiple temporal dependency and global dynamic spatial correlation

Given a station, the passenger mobility is not only related to its own past distributions, which contain various temporal trends (e.g. recent trend, daily trend, weekly trend), but also depends on other stations, showing global spatial dependency. For example, the passenger flows might be abnormally large during an unpredictable event. For any station, the abnormally increasing passenger flow may be regarded as a purely random phenomenon. However, if aggregating the flows of multiple stations, the abnormal passenger mobility pattern comes to the surface, which is difficult to discover from the view of a single station. Purely random fluctuation of OD flow in one station might not have great impact on future prediction, while the abnormal passenger flow in a global scale might last for a long period. We can call this phenomenon as mobility synchronization, which is useful for predicting future passenger mobility of each station more accurately. Note that the correlations among stations are nonlinear and time-evolving, increasing the prediction difficulty.

The OD metro prediction approaches can be roughly divided into two categories, i.e., the traditional methods including, e.g. least-square [1], matrix factorization [2–4], probabilistic model [5] and deep learning-based methods [6–11]. Gong [2] proposed a nonnegative matrix factorization-based model to capture both the stable flows and sudden changes in passenger mobility pattern; however, they only used the information of past observed completed OD values but ignored the incomplete OD values. Noursalehi et al. [8] proposed a CNN-based model to capture local spatial dependencies, exogenous information and evolitional temporal demand in OD flows. They also utilized a squeeze-and-excitation layer to compact the information for final prediction. Zhang et al. [10] proposed a channel-wise attentive split convolutional neural network, which used a user-specific mask to address the data sparsity issue. [7] proposed TS-STN with attention-based LSTM and graph convolution network with temporally shifted correlation graph matrix to capture complex spatiotemporal correlation in OD flows. However, these works are not able to tackle the challenges mentioned in this paper sufficiently. When tackled the sparse flow distribution challenge, Zhang et al. [10] focused only on the large OD values and ignored the small ones, which might not meet the demand of accurate flow control. As to the data complete problem, only the nearly complete OD values in recent OD matrix were utilized and the remainder is abandoned, resulting in a sparse input in [2]. [7] utilized both station entry demand and partially OD flow to complete OD values but simply by a fully connected layer which can’t fully investigate the complex mobility information in the available data. Noursalehi et al. [8] captured the local dependency in OD flow but did not exploit the dynamic spatial dependency in metro OD flow from a global perspective, which is valuable for improving the prediction accuracy.

In order to deal with all the aforementioned challenges of metro OD Matrix prediction, we propose a novel model, called Completion Augmentation based Self-Attention Temporal Convolutional network (i.e., CA-SATCN). First, a fusion gate mechanism is designed to adaptively estimate the unobservable part of recent OD matrices based on two kinds of moving pattern information, i.e., historical stable passenger mobility trend and latest dynamic passenger mobility trend, from various available data (e.g.
inflow, incomplete recent OD matrices, historical destination distribution ratio and statistical arrival time distribution). Then, we augment the sparse flow distribution with compact destination distribution and spatiotemporal context (e.g. connectivity position, region function) by leveraging appropriate embedding methods, e.g. LINE and latent Dirichlet allocation (LDA). Finally, the augmented flow distribution is fed into a variant of self-attention mechanism to produce a global high-level representation of passenger moving pattern for each station through aggregating the mobility trends of all other stations weighted by their dynamic spatial correlations. From these high-level features, three matrices sequences are constructed on different time scales and fed into different temporal convolution networks (TCNs) to extract various temporal trends (i.e., recent trend, daily trend, weekly trend) for prediction. Overall, the main contributions of our paper are as follows:

- We novelly reconstruct the recent OD matrices by designing a fusion gate mechanism to combine the historical stable passenger mobility trend and latest dynamic passenger mobility trend from observable data including inflow, partially available recent OD matrices, historical destination distribution ratio and statistical arrival time distribution.
- To alleviate the flow distribution sparsity problem, we utilize a series of strategies to learn a dense high-level representation for each station by compacting its destination distribution and complementing it with spatiotemporal context. For example, graph embedding model LINE and document topic model LDA are leveraged to learn spatial context for each station.
- A variant of self-attention mechanism is designed to model the dynamic and global spatial dependency in passenger mobility pattern, based on which three TCNs are proposed to capture temporal trends from different perspectives (i.e., recent trend, daily trend, weekly trend) in metro OD flow sufficiently.
- Extensive experiments on two real-world metro datasets demonstrate the superiority of our model.

2 Preliminary

In order to facilitate our discussion, we first introduce some basic definitions and then formulate the problem formally.

Definition 1 (Time interval). We evenly partition each day by a given time granularity \( \delta \) into \( K \) time intervals.

Definition 2 (Inflow). The number of passengers entering station \( i \) at time interval \( t \) is defined as the station inflow, denoted as \( \text{in}_i^t \). Suppose there are \( N \) metro stations in a subway system, the inflow of the subway network at \( t \) is denoted as \( \text{IN}_t = [\text{in}_1^t, \ldots, \text{in}_N^t] \in \mathbb{R}^{1 \times N} \).

Definition 3 (Finished/delayed inflow). The inflow of station \( i \) at an input time interval \( t \) can be divided into two parts according to the predicted time interval \( t' \), i.e., finished inflow and delayed inflow. The former is defined as the number of passengers finishing their trips before the predicted time interval, denoted as \( \text{in}_{i,f}^t \). The latter is defined as the remainder finishing their journeys during or after the predicted time interval, denoted as \( \text{in}_{i,d}^t \).

Note that \( \text{in}_i^t = \text{in}_{i,f}^t + \text{in}_{i,d}^t \). The finished inflow and delayed inflow of the subway are denoted as \( \text{IN}^t_f = [\text{in}_{1,f}^t, \ldots, \text{in}_{N,f}^t] \in \mathbb{R}^{1 \times N} \) and \( \text{IN}^t_d = [\text{in}_{1,d}^t, \ldots, \text{in}_{N,d}^t] \in \mathbb{R}^{1 \times N} \), respectively.

Definition 4 (Destination distribution vector). The destination distribution of passengers entering an origin station \( i \) during a time interval \( t \) is defined as its Destination Distribution Vector, denoted as \( E_i^t = [m_{i,1}^t, \ldots, m_{i,N}^t] \in \mathbb{R}^{1 \times N} \), where \( \sum_{j=1}^{N} m_{i,j}^t = \text{in}_i^t \) and \( m_{i,j}^t \) is the number of passengers entering station \( i \) at time interval \( t \) and exiting at station \( j \).

Definition 5 (OD matrix) The OD matrix of a metro system at time interval \( t \) is denoted as \( M_t = (m_{i,j}^t)_{N \times N} \in \mathbb{R}^{N \times N} \) where \( m_{i,j}^t \) is the number of passengers entering station \( i \) at time interval \( t \) and exiting at station \( j \). Note that OD matrix is composed of destination distribution vectors of all origin stations at time interval \( t \), i.e., \( M_t = [E_1^t, \ldots, E_N^t] \in \mathbb{R}^{N \times N} \).

Definition 6 (Finished/delayed OD matrix). Similar to the definition of finished/delayed inflow, the OD matrix at an input time interval \( t \) can be divided into two parts according to the predicted time interval \( t' \), namely finished OD matrix and delayed OD matrix. Finished OD matrix describes passengers entering subway during the input time interval and exiting the subway before the predicted time interval, denoted as \( M_{t,f}^t \in \mathbb{R}^{N \times N} \). Delayed OD matrix describes passengers entering the subway during the input time interval and exiting the subway during or after the predicted time interval, denoted as \( M_{t,d}^t \in \mathbb{R}^{N \times N} \).

Definition 7 (Recent OD matrices). The OD matrices at previous \( P \) time intervals before the predicted time interval \( t' \) are defined as recent OD matrices, denoted as \( \{M_{t-P}, \ldots, M_{t-1}\} \). As mentioned above, each recent OD matrix can be divided into finished OD matrix and delayed OD matrix according to the predicted time interval, i.e. \( M_t = M_{t,f}^t + M_{t,d}^t \). Note that the recent finished OD matrix \( M_{t,f}^t \) can be collected, while the recent delayed OD matrix
$M_{t,p}^d$ cannot be observed before the predicted time interval. If the unobserved delayed OD matrix $M_{t,p}^d$ has nonzero elements, it refers that not all the passengers exit the subway before the predicted time interval. In such case, the available recent OD matrix before the predicted time interval is incomplete.

### 2.1 Problem formulation

The available OD data in metro scenario is the finished OD matrices at current day and historical OD matrices at previous days or weeks. In this paper, we take full advantage of both the inflow information and the available previous OD data to predict the OD matrix at the predicted time interval. The problem can be formulated as follows:

\[
\hat{M}_t = g([IN_{t-1}, \ldots, IN_{t-P}], [M_{t-1}^f, \ldots, M_{t-P}^f], [M_t, \ldots, M_{t_P}])
\]

where $t'$ is the predicted time interval, $\hat{M}_t$ is the prediction at $t'$, $g$ is the mapping function, $[t' - 1, \ldots, t' - P]$ is the sequence of input time intervals at current day and $[IN_{t-1}, \ldots, IN_{t-P}]$ is the corresponding inflow sequence of all origin stations in the metro network, $[M_{t-1}^f, \ldots, M_{t-P}^f]$ is the corresponding finished OD matrix sequence at these time intervals. $[t_1, \ldots, t_P]$ is the sequence of input time interval at previous days or weeks and $[M_{t_1}, \ldots, M_{t_P}]$ is the corresponding historical OD matrix sequence. $P$ is the number of input time intervals, and it ranges from $[1, \ldots, K-1]$ where $K$ is the total time intervals of a day.

### 3 Methodology

In this section, we introduce CA-SATCN (as shown in Fig. 1), a general solution to forecast the next metro OD Matrix. In Fig. 1, TCN refers to temporal convolution network. FC linear refers to fully connected layer without activation function. FC Layer refers to the fully connected layer with activation function.

In CA-SATCN, first, the recent OD matrices are estimated through merging historical stable passenger mobility information and latest dynamic passenger mobility information by a fusion gate mechanism. Then, we obtain a compact and dense representation of each station at each time interval by compacting its destination distribution and enriching it with its spatiotemporal contexts learned by appropriate embedding approaches such as LINE and LDA. Based on the augmented representation, a variant of self-attention mechanism is designed to extract the dynamic spatial correlations globally for each station at each input step. Finally, three sequential representations at different time scales are fed into three temporal TCN-based components to learn temporal trends (i.e., recent trend, daily trend, weekly trend) for each station. The results of all the trends are fused to generate the predicted OD Matrix.

#### 3.1 Recent OD matrices completion

The recent OD matrices (as defined in Sect. 2) collected before the predicted time interval are likely to be incomplete due to the time gap between the entrance and exit of the subway stations. The incompleteness degree of OD values varies with the time difference between input time interval and predicted time interval, passenger arrival time distribution among OD pairs. The closer the input time interval is to the predicted time interval, the more incomplete its OD matrix is likely to be. In addition, the longer passenger travel time of an OD pair, the more incomplete its OD value is likely to be. To obtain more valuable realtime mobility information for prediction, this section proposes to reconstruct recent OD matrices by a gate mechanism from two perspectives, i.e. historical stable passenger mobility and latest dynamic passenger mobility, with the available historical destination distribution, inflow data and passenger arrival time distribution data. A toy example of recent OD matrix estimation is shown in Fig. 2.

As mentioned in Sect. 2, each recent OD matrix $M_t$ is composed of the available finished OD matrix $M_{t,P}^f$ and unobserved delayed OD matrix $M_{t,P}^d$. Further, the delayed OD matrix is the product of the available delayed inflow $IN_{t-P}^d$ and its corresponding destination distribution probability $DP_{t,P}^d \in \mathbb{R}^{N \times N}$, namely Delayed Destination Distribution Probability. Its $i_{th}$ row is $DP_{t,i,j}^d = (e_{t,i,j})_{1 \times N}$, referring to the destination distribution probability of delayed passengers entering station $i$ at input time interval $t$ and $\sum_{j=1}^{N} e_{t,i,j} = 1$. The complete Recent OD Matrix can be represented as follows:

\[
M_t = M_{t,P}^f + M_{t,P}^d = M_{t,P}^f + IN_{t,P}^d \ast DP_{t,P}^d
\]

Based on Eq. 2, if we can estimate $DP_{t,P}^d$, we can then estimate $M_t$. Further, $DP_{t,P}^d$ can be estimated by the estimation of $e_{t,i,j}^d$, which refers to the probability of delayed passengers entering station $i$ destining to station $j$. To estimate $e_{t,i,j}^d$, we rewrite it based on Bayes theorem as follows:

\[
e_{t,i,j}^d = p(D_j|O_i^d) = \frac{p(O_i^d|D_j, O_i) \ast p(D_j|O_i)}{\sum_{j=1}^{N} p(D_j|O_i) \ast p(O_i^d|D_j, O_i)}
\]
where $O_i$ refers to the passengers entering origin station $i$ at $t$ and $O^d_i$ refers to delayed passengers who cannot finish their journeys before $t'$. $D_j$ refers to the passengers whose destinations are station $j$. $p(D_j | O_i)$ is the probability of passengers entering station $i$ whose destinations are station $j$, defined as Destination Ratio. $p(O^d_i | D_j, O_i)$ is the delayed ratio of passengers entering $i$ and destining $j$, defined as Delayed Passenger Ratio. Based on Eq. 3, to obtain the estimation of $e^d_{ij,t}$, we need to estimate $p(D_j | O_i)$ and $p(O^d_i | D_j, O_i)$.

### 3.1.1 Delayed passenger ratio estimation

Intuitively, the delayed passenger ratio $p(O^d_i | D_j, O_i)$ is related to passengers’ arrival time and the predicted time interval. Based on this observation, we define a related concept as follows:

**Definition 8** (*Arrival time distribution probability*). Due to the influence of passenger walking speed, waiting time and other factors, the travel time of the trips belonging to same OD pair may be different. We do statistic on all the historical trips starting at time interval $t$ of OD pair $ij$ to obtain their arrival time distribution on each time interval during one day, denoted as follows: $T^d_{ij} = \{tr^d_{ij,1}, \ldots, tr^d_{ij,K}\}$ where $K$ is the number of time intervals during one day and $tr^d_{ij,k}$ is the number of trips starting during $t$ and finishing during $k$. Note that if $k < t$, $tr^d_{ij,k} = 0$ because all the trips haven’t started yet. We normalize $T^d_{ij}$ to define ArrivalTimeDistributionProbability of OD pair $ij$ as follows: $C^d_{ij} = \{c^d_{ij,1}, \ldots, c^d_{ij,K}\}$ where $\sum_{k=1}^{K} c^d_{ij,k} = 1$. $c^d_{ij,k}$ represents the proportion of the trips traveling between station $i$ and $j$ with arrival time in the range $[\delta(k-1), \delta k]$. 

![Fig. 1 The architecture of model CA-SATCN](image1)

![Fig. 2 A toy example of recent OD matrix estimation](image2)
The δ is the time granularity and consistent with predicted time granularity. Further, the ratio of trips with arrival time larger than \((t'-1)\delta\) is denoted as \(C_{i,d}^{ij} = \sum_{t} e_{i,d}^{ij}\). For those passengers entering origin station \(i\) at time interval \(t\), if their arrival time is larger than \((t'-1)\delta\), they cannot finish their trip before \(t'\), i.e., they are delayed passengers. Therefore, we utilize the statistical \(C_{i,d}^{ij}\) as an approximation of delayed passenger ratio, i.e., \(\hat{p}(O_{t'}|D_{t}, O_{t}) = C_{i,d}^{ij}\) and in this way, the stable statistical prior passenger pattern information can be incorporated into our model.

3.1.2 Destination ratio estimation

Similar to the prediction of the next OD matrix, we want to estimate destination ratio \(p(D_{t+1}|O_{t})\) to contain two temporal patterns, i.e., long-term periodicity and real-time dynamicity.

1) Long-term periodicity. We utilize the passenger mobility information at the same time slots of the previous weeks to extract the long-term periodicity for destination ratio estimation. Specifically, the average historical destination distribution of passengers entering subway at time interval \(t\) at previous weeks is denoted as \(\mathbf{W}_{t} = (a_{i,j}^{t})_{N \times N}\). The \(a_{i,j}^{t}\) denotes the probability of passengers entering station \(i\) at \(t\) who destined to station \(j\) and satisfies \(\sum_{j=1}^{N} a_{i,j}^{t} = 1\). Further, we estimate destination ratio by containing the long-term periodicity information as \(\hat{p}(D_{t+1}|O_{t}) = a_{i,j}^{t}\).

2) Real-time dynamicity. The destination distribution probability of passengers entering origin station \(i\) before input time interval \(t\) and arriving at their destinations during \(t\) is denoted as \(R_{t} = [r_{i,1}^{t}, \ldots, r_{i,N}^{t}] \in \mathbb{R}^{1 \times N}\) where \(\sum_{j=1}^{N} r_{i,j}^{t} = 1\). \(R_{t}\) is totally observable before predicted time interval \(t'\) and it contains the recent passenger flow mobility information. To extract the real-time passenger mobility information, we approximate destination ratio as \(\hat{p}(D_{t}|O_{t}) = r_{i,j}^{t}\).

3.1.3 Estimation by a fusion gate

We have obtained one estimation of \(\hat{p}(O_{t'}|D_{t}, O_{t}) = C_{i,d}^{ij}\) and two estimations of \(\hat{p}(D_{t+1}|O_{t}) = a_{i,j}^{t}\) and \(\hat{p}(D_{t}|O_{t}) = r_{i,j}^{t}\), we integrate them to obtain two estimations of \(e_{i,d}^{t+1,j}\) as follows:

\[
\begin{align*}
\hat{e}_{i,d}^{t+1,j} &= e_{i,d}^{t+1,j} = \frac{a_{i,j}^{t} \times C_{i,d}^{ij}}{\sum_{k=1}^{K} a_{i,k}^{t} \times C_{i,d}^{ij}} \\
\hat{e}_{i,d}^{t+1,j} &= e_{i,d}^{t+1,j} = \frac{r_{i,j}^{t} \times C_{i,d}^{ij}}{\sum_{k=1}^{K} r_{i,k}^{t} \times C_{i,d}^{ij}}.
\end{align*}
\]  

The corresponding estimations of delayed destination distribution probability \(DP_{t,f}^{d} = (e_{i,d}^{t+1,j})_{N \times N}\) and \(DPR_{t,f}^{d} = (e_{i,d}^{t,j})_{N \times N}\). The former contains the long-term periodicity of passenger mobility, and the latter contains the real-time dynamicity of passenger mobility.

Inspired by the gate mechanism in LSTM, we design a fusion gate to identify the importance of two estimations (i.e., \(DP_{t,f}^{d}\), \(DPR_{t,f}^{d}\)) to get the final estimation as follows:

\[
\hat{e}_{i,d}^{t+1,j} = \sigma(\mathbf{DP}_{t,f}^{d} \cdot \mathbf{W}_{1} + b_{1}) \cdot \mathbf{W}_{2} + b_{2}
\]

where \(\sigma\) is the sigmoid activation function. \(\cdot\) is matrix multiplication and \(\odot\) is Hadamard product. \(W_{1} \in \mathbb{R}^{2N \times F}, W_{2} \in \mathbb{R}^{F \times N}\) are trainable weights and \(b_{1}, b_{2}\) are corresponding biases. Further, we estimate the complete recent OD matrix \(\hat{M}_{t}\) by Eq. 2.

3.2 Flow distribution augmentation

As mentioned in 1, the destination distribution of passenger flow has data sparsity problem. Some OD values are very small or even zeros. Therefore, they are insufficient for extracting passenger moving pattern information, degrading model performance. However, except OD flow, other characteristics of stations also contain valuable spatiotemporal information for prediction, e.g. the spatial context and time context. The spatial context refers to a station’s physical features. One is about station’s geographical and connectivity position in the metro network. According to Tobler’s first law of geography, near things are more related to each other. A station in a metro system might have more influence on its adjacent neighbors than its distant neighbors. In addition, the connectivity between two stations also influences their passenger patterns.

Another is about station’s functional characteristics (e.g. educational / business / residential area). Stations with similar functionality are likely to share similar mobility patterns. The time context refers to the time attributes of passenger flow of a station, e.g., day of week, time of day and so on. The time context can reflect the passenger mobility patterns in some extent for that passengers have different travel behaviors at different time contexts. In this section, we aim to augment the flow distribution of each station at each time interval by learning and fusing the spatial context and time context.
3.2.1 Spatial context learning

As mentioned above, we extract two spatial properties (i.e., its travel time-based connectivity and region function) of each station by learning the corresponding latent embeddings.

**Travel time-based connectivity representation.** We measure the travel time between station $i$ and station $j$ as the average travel time of all the historical trips between these stations, defined as $t_{ij}$. However, only when these stations are adjacent, their edge weight is assigned as $t_{ij}$ and if they are not adjacent, the edge weight is assigned zero. Specifically, the travel-time-based connectivity matrix is denoted as $A = (a_{ij})_{N \times N}$, and $a_{ij}$ is defined as follows:

$$a_{ij} = \begin{cases} t_{ij}, & \text{if } i \text{ is connected directly to } j \\ 0, & \text{otherwise} \end{cases}$$

Note that the edge weight measured by travel time contains the geographical distance information between two directly connected stations. Therefore, the travel-time-based connectivity matrix not only contains the connectivity between stations in the metro network but also contains the relative geographical position of metro stations. We use graph embedding model LINE [12] to extract a station’s connectivity-based position representation on a metro graph. Compared with other graph embedding methods, e.g. DeepWalk and Node2Vec, LINE can not only model global connectivity of the metro network by the first-order proximity but also capture the local structure of a target station by second-order proximity, denoted as $L = \text{LINE}(A)$. $L = [L^1, \cdots, L^N] \in \mathbb{R}^{N \times L}$ is the spatial embedding matrix for the whole network. $L^i \in \mathbb{R}^L$ is the embedding vector for station $i$ to preserve its position information.

**Region function representation.** Intuitively, the historical destination distribution of a station can reflect its functional characteristic. We follow the idea [13, 14] to utilize latent Dirichlet allocation (LDA for short) to learn the function representation of each station. LDA is widely used in capturing the hidden topics in large-scale documents. LDA assumes that each document has multiple topics and each word in a document belongs to a certain topic. Therefore, LDA can infer the hidden topics of a document based on all the observed words. In our scenario, an origin station is analogous to a document, and its region function is similar to the topic of a document. The destination distribution of an origin station is treated as words in a document. The historical average destination distributions of the whole network is similar to the corpus of all documents, and we feed them into LDA to obtain functional representations of all the stations in the metro network. Note that LDA does not participate in the end-to-end training process of our model, and it is just utilized to produce input feature, namely the region function embedding for every station.

We use $D^i$ to represent the functional representation of station $i$. We concatenate the representations of $L^i$ and $D^i$ of the station $i$ to get the spatial representation $S^i$. Such spatial representation is static, and it is the same for each time interval.

3.2.2 Time context learning

For a station $i$ at time interval $t$, its time attributes can help the model better learn the correlation of passenger movement patterns at different time intervals. We utilize a temporal embedding method to encode its temporal attributes into a temporal vector. Specifically, we adopt one-hot coding to encode the time-of-day attribute into a $K$ dimension space, denoted as $D_t \in \mathbb{R}^K$, where $K$ is the total number of the time intervals in one day defined in Sect. 2. The day-of-week attribute is encoded into a 7 dimension space, denoted as $H_t \in \mathbb{R}^7$. To align the two temporal vectors, all temporal embeddings are fed into two fully connected layers with $T$ output units and added together to get the final temporal representation as $T^i_t = W_{h_t} \text{ReLU}(W_{h_t} H_t + b_1) + W_{d_t} \text{ReLU}(W_{d_t} D_t^i + b_2) \in \mathbb{R}^T$ where $W_{h_t} \in \mathbb{R}^{T \times 7}$, $W_{h_t} \in \mathbb{R}^T$, $W_{d_t} \in \mathbb{R}^{T \times K}$, $W_{d_t} \in \mathbb{R}^T$ are trainable parameters.

3.2.3 Compact destination distribution

As mentioned in 1, the original distribution representation of each station at each time interval (i.e. Destination Distribution Vector) is high dimension. In order to distill valuable mobility information and align with the spatiotemporal embeddings, we get a dense and compact destination distribution for each station at each time interval, by feeding destination distribution vector $E_t^i$ into two fully connected layers as $C^i_t = \text{ReLU}(W_{f_t} \text{ReLU}(W_{f_t} E_t^i) + b_1) + b_2) \in \mathbb{R}^C$ where the number of output units $C$ is much smaller than the number of stations $N$. $W_{f_t} \in \mathbb{R}^{C \times N}$ and $W_{f_t} \in \mathbb{R}^{C \times C}$ are trainable parameters. Note that there are three kinds of destination distribution vectors, i.e., $E_t^i$ can represent the destination distribution vector from estimated recent OD data, daily OD data and weekly OD data.

To fuse the above features to get a final high-level flow representation $F_t^i$ of station $i$ at time interval $t$, we first concatenate the spatial representation and time representation and then merge the compact destination distribution with its spatiotemporal representation by adding them together in different weights as follows:
\[ F_i^t = \text{ReLU}(W_1 C_i^t + W_2 (W_7 T_i^t || W_5 S^t)) \in \mathbb{R}^F \]  
\[ e_{ij}^t = \left( K_i^t \right)^T Q_j^t \in \mathbb{R} \]  
\[ z_{ij}^t = \frac{\exp(e_{ij}^t)}{\sum_{j=1}^{N} \exp(e_{ij}^t)} \in \mathbb{R} \]  
where \( e_{ij}^t \) is the attention and \( z_{ij}^t \) is the normalized attention through Softmax function. The scaling factor is \( \frac{1}{\sqrt{d}} \).

### 3.3 Global mobility trend-based spatiotemporal dependency modeling

In the section above, we have already obtained a rich and dense representation for each station at each time interval. Based on these augmented features, this section aims to model the global and dynamic spatial correlations among stations and dig out multiple temporal moving patterns in the passenger mobility data by a self-attention mechanism and temporal convolutional network.

#### 3.3.1 Self-attention for global dynamic spatial dependency extraction

We have extracted the spatiotemporal representation of each station at each time interval. In this section, we capture the global and high-level passenger mobility pattern for each station by learning from other stations in the metro network. We define these learned stations as the context stations of the target station. In recent years, transformer made huge success initially in natural language processing [15] and is extended to other domains, e.g. traffic domain. Some works utilized transformer-based methods to extract the spatiotemporal dependencies in traffic tasks, such as trajectory prediction [16], traffic flow forecasting [17] and achieved state-of-the-art results. The power of transformer network is due to its self-attention mechanism, which can model the complex correlations among objects globally. In this paper, we design a variant of self-attention mechanism to extract the global spatial correlation in the network and generate high-level feature for each station at each time interval (as shown in Fig. 3).

For a target station \( i \) and its context station \( j \) at a given time interval \( t \), to measure the amount of passenger mobility knowledge that \( i \) can learn from \( j \), we project the high-level representation of \( i \) into a latent key space and that of \( j \) into a query space, resulting in the corresponding vectors named Key, Query for these two stations as follows:

\[ K_i^t = f_{\text{Key}}(F_i^t) \in \mathbb{R}^F \]  
\[ Q_i^t = f_{\text{Query}}(F_i^t) \in \mathbb{R}^F \]  

where \( F_i^t \) is the high-level embedding of station \( i \) at time interval \( t \). \( f_{\text{Key}}, f_{\text{Query}} \) are three fully connected layers with activation function ReLu. Then, the inner product of key vector of station \( i \) and query vector of station \( j \) is calculated to represent the dynamic spatial attention that \( i \) should pay on \( j \) as follows:

\[ U_i^t = \text{ReLU} \left( W_2 \text{ReLU} \left( W_1 \left( \sum_{j=1}^{N} z_{ij}^t C_j^t \right) \right) \right) \in \mathbb{R}^U \]

where \( W_1, W_2 \) are trainable parameters.

The high-level representation that contains the global spatial dependency information has been constructed for each station at time interval \( t \) and we stack such representations of all stations together, obtaining the high-level representation for the whole network as \( U = [U_1^t, U_2^t, \cdots, U_N^t] \in \mathbb{R}^{N \times U} \).

#### 3.3.2 TCN for multiple temporal dependency extraction

We have extracted the global spatial correlations among stations in the metro network and generated high-level representation for the network at each time interval. In this section, we identify three temporal trends of passengers' mobility, i.e., recent trend, daily trend, and weekly trend. The weekly trend and daily trend are about long-term periodic dependencies. The recent trend is about short-term dynamic dependency. We construct three sequences and feed them into three TCN-based modules to model the temporal trends sufficiently.

1. **Temporal sequences construction.** We construct the corresponding sequences for three trends based on the historical high-level representations generated by self-attention mechanism, i.e., \([U_1, U_2, \ldots, U_{Ww}]\) where \([t_1, t_2, \ldots, t_{Ww}]\) are historical time intervals.

   - **Weekly sequence.** Weekly trend reflects periodicity within a week in trips. We utilize the high-level representations at the same time intervals of previous days with the same week attribute (e.g. Monday) from previous \( W \) weeks for prediction, denoted as \( U_{W} = [U_1, \ldots, U_{Ww}] \in \mathbb{R}^{W \times N \times U} \) where \([t_1, \ldots, t_{Ww}] = [t' - 7K, \ldots, t' - 7K] \) and \( K \) is the number of total time intervals in one day. \( t' \) is the predicted time interval.

   - **Daily sequence.** The weekly trend cannot capture the movement pattern of those less regular trips...
generated by some activities in a few days (e.g. a sport game lasting for some consecutive days), namely daily trend. To extract the daily trend in OD flow, we utilize representations at the same time intervals of previous D days for prediction, denoted as $U_D = [U_{t_1}, \ldots, U_{t_D}] \in \mathbb{R}^{D \times N \times U}$ where $[t_1, \ldots, t_D] = [t' - K, \ldots, t' - KD]$.

**Recent sequence.** Both weekly and daily trends can only capture temporal dependency in OD flow from the same time intervals at historical days but ignore dependency from last R intervals at the same day, namely recent trend, which also has an obvious influence on OD flow at the predicted time interval, denoted as $U_R = [U_{t_1}, \ldots, U_{t_R}] \in \mathbb{R}^{R \times N \times U}$ where $[t_1, \ldots, t_R] = [t' - 1, \ldots, t' - R]$.

(2) **Temporal convolution network.** Previous OD Matrix prediction works mainly leveraged long short-term memory network (LSTM) to capture the temporal dependency in OD flows [6]. However, the complex structure of LSTM leads to more memory requirement and longer training time [18]. Recently, some traffic works focusing on prediction of traffic speed [19], traffic flow [20] and traffic state [18, 21] utilized temporal convolution network (TCN) [22] to extract temporal dependency because TCN can tackle sequence modeling problem more efficiently in a non-recursive manner and achieve a better performance. In this paper, we leverage TCN (as shown in Fig. 4) to extract each temporal trend from its corresponding OD sequence, simply denoted as $[U_{t_1}, U_{t_2}, \ldots, U_{t_R}]$. In each TCN layer, the $j_{th}$ output feature of station $i$ at time $t$ is generated by the dilated causal convolution as follows:

$$Y_{i,j}^{l+1} = \rho \left( \sum_{m=1}^{U^i} \sum_{k=0}^{K-1} W_{j,m,k} U_{i-dk,m}^l \right)$$

where $U^i$ is the number of input feature at layer $l_{th}$, $j \in \{1, \ldots, U^{l+1}\}$ and $U^{l+1}$ is the number of output features. $U_{i-dk,m}^l \in \mathbb{R}$ is the $m_{th}$ input feature of station $i$ at time $t - dk$. $d$ is the dilation rate controlling the skipping distance. $\rho$ is the activation function. $K$ is the kernel size and kernel $\Theta = (w_{j,m,k})_{K \times U^i}$ is trainable. The same convolution kernel is applied to all the metro stations. The whole TCN layer can be formulated as follows:

$$Y_{i}^{l+1} = \rho \left( \Theta *_{D} U_{i}^{l} \right)$$

where $l$ is the $l_{th}$ layer. $U_{i}^{l} \in \mathbb{R}^{N \times U}$ is the input of $l_{th}$ layer. $Y_{i}^{l+1} \in \mathbb{R}^{N \times U^{l+1}}$ is the output of $l+1_{th}$ layer. Following previous works [19], the dilation rate $d^l = 2^l$. TCN layer leverages zero padding strategy to keep the output length the same as the input length. Since we only predict the OD matrix of one future step, we take the last step of TCN output as its final output.

(3) **Weighted fusion for multiple temporal trends.** We feed the high-level sequence of each trend into the corresponding TCN to extract the long-term dependency in historical OD flow. We denote the output of each trend as $H_R, H_D, H_W$. These outputs are merged according to their importance in OD prediction task, and we attach different weights to them on a data-driven basis as follows:

$$\hat{M}_d = ReLU(H_R * W_r + H_D * W_d + H_W * W_w)$$

where $\hat{M}_d \in \mathbb{R}^{N \times N}$ is the prediction result. $[W_r, W_d, W_w]$ are trainable parameters. We choose ReLU as the activation function because the number of passengers cannot be negative.
4 Experiments

In this section, extensive experiments are conducted to answer the following research questions.

RQ1: How does our model perform at prediction accuracy compared with other benchmarks on different datasets?
RQ2: Is the proposed data completion method effective to improve the model performance?
RQ3: How does each temporal trend contribute to improve prediction accuracy?
RQ4: Is our model sensitive to the hyper-parameters?

4.1 Experiment settings

4.1.1 Datasets

We conduct our experiments in two real-world large-scale subway datasets collected by the automatic fare collection systems (AFCs) in big cities, i.e., Shenzhen, Hangzhou, on the premise of complying with the security and privacy policies. The AFC in metro system has generated a very large amount of records of all passengers, including both entry station records and exit station records. An entry record and its corresponding exit record are preprocessed into a complete trip, which contains five kinds of information, i.e., the smart card ID, the entry station ID, the entry time, the exit station ID and the exit time. Specifically, Shenzhen covers nearly 100 million records of 118 stations for twelve months in 2014. Hangzhou has about 5 million records of 80 stations in January 2019. We do statistics on the travel time distribution of the whole datasets. As shown in Fig. 5, in both datasets, more than 80% of the trips are within 60 min. We focus on 7:00-23:00 of each day and aggregate the trips on a 30-minute time granularity to construct OD matrices. As to the data split, we divide 70% of each dataset into train data, 10% into validation data, 20% into test data chronologically. The data normalization method is Z-Score. The statistic details of datasets are elaborated in Table 1.

4.1.2 Baselines

Baselines in this paper can be divided into two categories. The first is the generalized methods utilized as baselines in previous OD metro works [4, 10], including HA, Ridge, ANN, FC-LSTM, ConvLSTM and GCN. The second is the recent state-of-the-art OD prediction models, including CASCNN in metro OD prediction, and GEML in rail-hailing OD prediction. OD prediction in different scenarios shares the complex spatiotemporal challenge in some extent.

HA (Historical average) is the simplest and the most commonly used baseline in time-series forecasting tasks. HA in this paper averages the previous OD values of an OD pair to predict its future value in the next step. Its performance is evaluated on all the OD pairs.

Ridge is a linear regression method with L2 regularizer. It tends to treat all the input features equally. We utilize the RidgeCV class from Sklearn and set its two parameters as alpha = [1, 0.1, 0.01, 0.001] and cv = 5.

ANN (Artificial neural network) is the simplest neural network, and it can capture the linearity and nonlinearity of traffic data. Our ANN has two hidden layers and an output layer. Each hidden layer has 128 units and the output layer has N (i.e. the number of stations) units. The activation function of all layers is rectified linear units (ReLU).

FC-LSTM (Fully connected long short-term memory network) is typically used for capturing long-term temporal

Fig. 5 Travel time distributions on SZ and HZ datasets
ConvLSTM has three layers with 8, 8, and 1 filters, respectively, and the kernel size of all the filters is set to be $3 \times 3$ in the model.

$GCN$ (Graph convolution network) [24] can extract the graph-based spatial dependency through aggregating the features from neighbors in a traffic graph. Afterward, a fully connected layer is utilized to transform the outputs of $GCN$ at previous time intervals to obtain the final result.

$CASCNN$ (Channel-wise attentive split convolutional neural network) [10] is a deep learning framework for metro OD prediction. It only utilizes the historical OD matrix at previous days and recent inflow/outflow information for prediction by channel-wise attention in time-series data. There are also two hidden layers with 128 units and an output layer with $N$ units in our LSTM. The activation function of the output layer is ReLU function. We utilize LSTM to predict the OD vector of an original station $i$ with its previous OD vectors as input. All the stations share the same LSTM parameters.

*ConvLSTM* (Convolutional LSTM) [23] replaces the fully connection operation in FC-LSTM with convolution operation. It can extract both the temporal and grid-based spatial dependency, while FC-LSTM is only able to extract temporal dependency. Following the previous works [10], ConvLSTM has three layers with 8, 8, and 1 filters, respectively, and the kernel size of all the filters is set to be $3 \times 3$ in the model.
In this paper, our main task is to predict OD matrix at the target time interval \( t' \). Following previous works [4, 10], we adopt mean square error (MSE) as the loss function. The predicting errors of large OD values are enlarged through MSE. Thus, MSE pays more attention on OD pairs with higher volumes, which are also considered to be more important in metro scenario. The loss function is as follows:

\[
\mathcal{L} = \text{MSE}(M_r - \hat{M}_r) = \frac{1}{N \times N} \sum_{i=1}^{N} \sum_{j=1}^{N} (m^{ij}_r - \hat{m}^{ij}_r)^2
\]  

4.1.3 Loss function

4.1.4 Evaluation metrics

Following previous works [4, 10, 25, 26], we evaluate model performance with four widely applied metrics, i.e., mean absolute error (MAE), root mean square error (RMSE) and weighted mean absolute percentage error (WMAPE), symmetric mean absolute percentage error (SMAPE) as follows:

\[
\begin{align*}
\text{MAE} &= \frac{1}{N \times N} \sum_{i=1}^{N} \sum_{j=1}^{N} |m^{ij}_r - \hat{m}^{ij}_r| \\
\text{RMSE} &= \sqrt{\frac{1}{N \times N} \sum_{i=1}^{N} \sum_{j=1}^{N} (m^{ij}_r - \hat{m}^{ij}_r)^2} \\
\text{WMAPE} &= \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} |m^{ij}_r - \hat{m}^{ij}_r|}{\sum_{i=1}^{N} \sum_{j=1}^{N} m^{ij}_r} \\
\text{SMAPE} &= \frac{1}{N \times N} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{|m^{ij}_r - \hat{m}^{ij}_r|}{(m^{ij}_r + \hat{m}^{ij}_r)/2 + c}
\end{align*}
\]

MAE and RMSE are sensitive to data scale, while WMAPE and SMAPE are data-scale independent. RMSE is also sensitive to the outliers. WMAPE can avoid zero-division and over-skewing confronting by MAPE. While WMAPE is inclined to underestimate, SMAPE is symmetric and can avoid such problem. \( c \) is a small positive constant to avoid zero-division, set as 1 [25].

4.1.5 Experiment settings

GEML [25] and CASCNN [10] follow the parameter settings in original papers, and the hyper-parameters of other baselines are tuned on the validation set. In terms of training, we utilize Adam optimizer [27] to optimize all the deep learning models and the initial learning rate is 0.01. We take the learning rate decay strategy to get the best results of each model. The maximum number of epochs is set as 1000, and an early stop strategy is taken if the validation loss stops decreasing for 20 epochs. The batch size for Shenzhen datasets is set as 32, for Hangzhou dataset as 16. We utilize Python and TensorFlow for the code. The experiments are run on a GPU (32GB) machine with TESLA V100.

4.2 Experiment results

As shown in Table 2, there are three kinds of input patterns corresponding to three kinds of temporal trends, i.e. R\( ^t \) for incomplete recent OD input, D\( ^t \) for daily OD input and W\( ^t \) for weekly OD input. We test the generalized methods (e.g., HA, Ridge, ANN, FC-LSTM, ConvLSTM, GCN) on all the input patterns to observe their best results, which are highlighted in bold. As to the state-of-the-art OD prediction approaches (e.g. GEML, CASCNN), their problem setting is to take the specific pattern as input. The best experiment results under each dataset are highlighted in red.

We visualize the prediction and ground truth on randomly chosen OD pairs on each dataset at a randomly chosen day of weekday or weekend. As shown in Fig. 6, the prediction result of our model (i.e., the red curve) gives the best fit to the ground truth (i.e., the blue curve) in all datasets.

4.2.1 Performance comparison

Table 2 shows performance of all models under different input patterns on two metro datasets. Our model achieves better performance than baselines. Compared with GEML, i.e., the state-of-the-art baseline, CA-SATCN improves the performance by [5%, 4%] in WMAPE and [3%, 4%] in SMAPE on two datasets, respectively.

The traditional statistical methods, i.e., HA and Ridge, achieve the best performance at weekly trend pattern. These two methods are good at extracting linearity in traffic data, indicating that linear correlation in weekly OD data is stronger than that in daily OD data or recent OD data. It is in consistence with the common knowledge that the weekly passenger mobility pattern is more regular, while the recent passenger movement is more stochastic. The simple deep learning methods (i.e., ANN, FC-LSTM, ...
ConvLSTM and GCN) perform best when fed with the recent OD data, probably due to their superior capacity to capture the complex nonlinearity in OD data. Among these approaches, FC-LSTM has the best performance, indicating that the extracting temporal dependency is vital for prediction. ConvLSTM can also capture the temporal pattern in OD data, while its performance is worse than LSTM. This is probably because it treats OD matrix as grids and intends to extract spatial dependency from it, which is obviously not appropriate for that OD matrix reflects the OD flow pattern instead of metro network spatial topology. GCN is gifted at extracting the static graph-based spatial dependency while it cannot deal with temporal pattern. Another observation is that all these deep learning approaches have the worst performance at the daily input pattern, referring that the daily trend is less valuable compared with the recent trend and weekly trend. Such conclusion can be augmented by the observation that although CASCNN has the best performance over the methods with the daily input pattern, its performance is worse compared with some baselines fed by recent input, e.g. FC-LSTM. GEML can capture both temporal pattern-and graph-based spatial dependency in a unified way; thus, it has the best performance over other benchmarks. However, GEML defines the spatial correlation manually and statistically and only captures the recent trend. Compared to GEML, our model reconstructs the recent OD matrices to get more useful real-time mobility information for prediction. It also utilizes the self-attention mechanism to dynamically capture the complex spatial correlations among stations globally and extract three kinds of temporal trends sufficiently, resulting in better prediction.

### 4.2.2 Efficiency comparison

The Table 3 shows the efficiency of all the deep learning models on Shenzhen dataset. ANN takes 3.3 s per epoch in the training stage, and it is the fastest, perhaps due to its simplest structure. GCN also runs fast because it does not capture the temporal dependency. FC-LSTM is slower than GCN for it has more trainable parameters. Our model runs a little slower than FC-LSTM for that it captures three temporal trends with TCN, while FC-LSTM just captures one trend. GEML is based on LSTM and attention mechanism with three kinds of graph convolutions to capture spatial properties. It is slower than our model. CNN-based models, including CASCNN and ConvLSTM, are slower than other models, and ConvLSTM is the slowest model.

| Models    | ANN (s) | GCN (s) | FC-LSTM (s) | CA-SATCN (s) | GEML (s) | CASCNN (s) | ConvLSTM (s) |
|-----------|---------|---------|-------------|--------------|----------|------------|--------------|
| Train     | 3.2     | 4.1     | 8.9         | 10.1         | 13.9     | 14.6       | 15.5         |
| Test      | 0.1     | 0.3     | 1.5         | 1.7          | 1.8      | 2.1        | 2.3          |
| Total     | 3.3     | 4.4     | 10.4        | 11.8         | 15.7     | 16.7       | 17.8         |

Fig. 6 Visualization of the prediction and ground truth on randomly chosen OD pairs on different datasets.
4.2.3 Model ablation analysis

We perform ablation tests to demonstrate the effectiveness of each component in our model. Due to the space limitation, Table 4 only shows the ablation results on Hangzhou dataset but those on Shenzhen datasets have similar conclusions.

It can be observed that the four metrics of all variants deteriorate in some extent, proving that all components are necessary to improve prediction accuracy. However, their contributions are different. When we remove the recent matrices, the variant obtains the worst performance with 10% increased in WMAPE and 7% increased in SMAPE, which indicates that recent trend contributes the most for future OD flow prediction in our model. Without recent OD matrices completion, the model performance also deteriorates to a large extent, i.e., 4% increased in both WMAPE and SMAPE, which proves the effectiveness of data completion. The spatiotemporal contexts learning is also important for prediction, without which WMAPE increases 3% and SMAPE increases 2%. Besides, the experiments show that weekly matrices contribute more to improve prediction accuracy than daily matrices does, indicating that the next step OD flow depends more on weekly trend than daily trend. The spatial context learning is more important than the temporal context learning for prediction. Overall, we can clearly observe that our full model achieves the best result, illustrating that the three temporal trends, data completion and spatiotemporal contexts are beneficial to the improvement of model performance.

![Fig. 7 Data completeness of different OD pairs at different input steps on SZ and HZ datasets](image-url)

Table 4 Model degradation in HZ dataset

| Modules                                | MAE  | RMSE | WMAPE | SMAPE  |
|----------------------------------------|------|------|-------|--------|
| No recent matrices                     | 2.37 | 4.54 | 0.47  | 0.43   |
| No daily matrices                      | 2.06 | 4.15 | 0.40  | 0.39   |
| No weekly matrices                     | 2.13 | 4.19 | 0.41  | 0.40   |
| No spatiotemporal contexts learning   | 2.05 | 4.13 | 0.40  | 0.38   |
| No spatial context learning            | 1.94 | 3.90 | 0.39  | 0.38   |
| No time context learning               | 1.89 | 3.75 | 0.38  | 0.37   |
| No recent OD matrices completion       | 2.14 | 4.33 | 0.41  | 0.40   |
| All                                    | 1.79 | 3.61 | 0.37  | 0.36   |

Each time we remove one component and keep the remainder unchanged to test the effectiveness of each component. Note that when we remove the recent matrices, we also remove the recent OD matrices completion. Each experiment is repeated five times to calculate the means and standard errors to get more convincing results. The best results are highlighted in bold.
4.2.4 Effectiveness of data completion

Data completeness of different randomly chosen OD pairs with different passenger flow scale on two datasets. The number of the input steps is set at 5.

Figure 7 visualizes the performance of data completion on randomly chosen OD pairs with different passenger flow scales on two datasets. We show the experiment results at the best length of recent trend on both datasets, i.e., 5 input steps. The green curve represents the collected partially finished trips before the target time interval. The blue curve represents all the finished trips in a whole day but unobservable before the target time interval. The red curve represents the estimated trips by our proposed data completion method. It can be observed that the estimated OD values are closer to the ground truths on OD pairs with different scales than the incomplete OD values. The data completion method plays an important role for the input steps close to the output step by alleviating the sparsity problem. For example, as we can see in SZ OD1 pair, the first step closest to the predicted step has nearly zero value, which refers to the serve loss of passenger mobility information. After the data completion method, we can obtain an estimated value 602 close to the true value 611 at input step one. The estimated values can provide more valuable recent distribution information. Also, as shown in Table 4, the data completion approach contributes to improve the model performance by decreasing 4% in both WMAPE and SMAPE in Hangzhou dataset, demonstrating its effectiveness.

4.2.5 Hyper-parameter analysis

In this section, we aim to test the sensitiveness of our model on hyper-parameters. Due to the space limitation, we choose some important parameters and test them on Shenzhen dataset, i.e., the sequence lengths of three temporal trends \( [R, D, W] \). We keep other parameters as default values when varying the target parameter. The default values of sequence length in sensitive test are \( R = 3, D = 3, W = 3 \).

As shown in Fig. 8, when we change the length of recent input, the performance of the model improves as the length increases and it reaches the best at 5 steps. However, as the steps increases, the model performance degrades, which might due to the disturbance of irrelevant information at longer steps. The best value of daily sequence is also step 5 and that of weekly sequence is at 4. The variation of model performance at recent test is larger than that of daily or weekly test, suggesting that parameter of recent trend seems more important in our model.

5 Related works

5.1 OD matrix prediction in traffic domain

A considerable number of works have been conducted on traffic flow prediction [28–32], while Origin-Destination (OD) matrix prediction receives less attention for its greater complexity [33, 34]. To date, related works on metro OD prediction are critically few [10, 35]. Although metro OD demand has different contextual setting from other OD demands, e.g. road network [36–38], taxi OD demand [39–41] and so on [42]. They share some common challenges (e.g. data sparsity, complex spatiotemporal dependency) in some extent. Therefore, we extend the introduction to similar OD matrix prediction in traffic domain.

The methods proposed to solve OD demand problem can be roughly divided into three categories. The first class is conventional methods like least-square approach [1], Kalman filtering modeling [43, 44] and probability-based analysis [5]. They are hard to deal with a large amount of data, and their performances need to be improved. The second category is machine learning methods [45]. For example, the matrix factorization-based methods [2, 4, 3] can solve the dimension curse in OD prediction. However, they are hard to extract complex spatiotemporal correlations in OD flows. The last group is deep learning methods [10, 37, 38, 46]. Some works develop LSTM variants to learn temporal property in OD data [6, 26]. However, it can’t capture the spatial correlation in OD prediction. Recently, graph neural networks are applied in capturing the spatial correlations in OD prediction [25, 36, 37, 40, 41]. However, most of them predefined the spatial correlations, resulting in learning only static spatial dependency. In this paper, we focus on learning the dynamic spatial dependency in metro OD scenario.

5.2 Transformer

Transformer was first proposed in natural language processing [15] and gained extensive popularity quickly. It has also been successfully applied in other domains, such as computer vision [47], recommendation [48, 49] due to its powerful self-attention mechanism. In traffic domain, some pioneering works have applied transformer in several research directions and achieved state-of-the-art performance [50]. [51] utilized self-attention mechanism in transformer to develop a spatial attention mechanism to capture the dynamic correlations.
between sensors in the road network. In order to encode traffic evolving patterns, [30] designed a self-attention layer to automatically perform the temporal aggregation in bike sharing scenario. [52] integrated a spatial transformer network into CNN to perform a geometric transformation in traffic sign recognition. [53] proposed a novel spatial-temporal transformer network capturing spatial-temporal dependencies in long-term traffic flow prediction. However, to the best of our knowledge, the existing transformer methods are not yet applied in OD prediction problem. In this paper, we adopt the core mechanism of transformer, i.e., the self-attention mechanism to capture the dynamic spatial correlation in metro OD prediction.

6 Conclusion

This paper proposes a model CA-SATCN to conduct short-term OD prediction in rail transit network. In our model, a newly adaptive fusion method capturing long-term stable periodicity and short-term dynamic correlation is proposed to solve the incomplete recent matrices problem in metro scenario. Embedding methods are utilized to enrich flow distribution to alleviate the data sparsity problem. A variant of self-attention mechanism is designed to capture the dynamic global spatial dependency in OD flows. Afterward, three temporal trends, i.e., recent trend, daily trend and weekly trend, are captured by three temporal convolutional networks, respectively. Extensive experiments are carried out on two real-world metro datasets, showing the superiority of our model over other benchmarks.

Acknowledgment This work is supported by the National Key R&D Program of China (No.2019YFB2102100); Shenzhen Industrial Application Projects of undertaking the National Key R & D Program of China (No.CJGJD20210408091600002); Shenzhen Basic Research Program (No.JCYJ20190812153212464); The Fundamental Research Foundation of Shenzhen Technology and Innovation Council (No. KCXFGZZ2021221173613035).

Data availability The data that support the findings of this study are not openly available due to passengers privacy and are available from the corresponding author upon reasonable request by email.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Bierlaire M, Crittin F (2004) An efficient algorithm for real-time estimation and prediction of dynamic OD tables. Oper Res 52(1):116–127
2. Gong Y, Li Z, Zhang J, Liu W, Zheng Y, Kirsch C (2018) Network-wide crowd flow prediction of Sydney trains via customized online non-negative matrix factorization. In: Proceedings of the 27th ACM international conference on information and knowledge management, pp. 1243–1252
3. Gong Y, Li Z, Zhang J, Liu W, Zheng Y (2020) Online spatio-temporal crowd flow distribution prediction for complex metro system. IEEE Transactions on knowledge and data engineering
4. Cheng Z, Trepanier M, Sun L (2022) Real-time forecasting of metro origin-destination matrices with high-order weighted dynamic mode decomposition. Transportation Science
5. Dai X, Sun L, Xu Y (2018) Short-term origin-destination based metro flow prediction with probabilistic model selection approach. J Adv Transp. https://doi.org/10.1155/2018/5942763
6. Li D, Cao J, Li R, Wu L (2020) A spatio-temporal structured LSTM model for short-term prediction of origin-destination matrix in rail transit with multisource data. IEEE Access 8:84000–84019
7. Jiang W, Ma Z, Koutsopoulos HN (2022) Deep learning for short-term origin-destination passenger flow prediction under partial observability in urban railway systems. Neural Comput Appl 34:1–18
8. Noursalehi P, Koutsopoulos HN, Zhao J (2021) Dynamic origin-destination prediction in urban rail systems: a multi-resolution spatio-temporal deep learning approach. IEEE Trans Intell Transp Syst 23:5106–5115
9. Toqué F, Côme E, El Mahrsi MK, Oukhellou L (2016) Forecasting dynamic public transport origin-destination matrices with long-short term memory recurrent neural networks. In: 2016 IEEE 19th International conference on intelligent transportation systems (ITSC), pp. 1071–1076
10. Zhang J, Che H, Chen F, Ma W, He Z (2021) Short-term origin-destination demand prediction in urban rail transit systems: a channel-wise attentive split-convolutional neural network method. Transp Res Part C: Emerg Technol 124:102928
11. Zhang J, Chen F, Wang Z, Liu H (2019) Short-term origin-destination forecasting in urban rail transit based on attraction degree. IEEE Access 7:133452–133462
12. Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q (2015) LINE: large-scale information network embedding. In: Proceedings of the 24th international conference on world wide web, WWW 2015, Florence, Italy, May 18-22, 2015, pp. 1067–1077
13. Zhao J, Zhang L, Ye J, Xu C (2021) MDLF: a multi-view-based deep learning framework for individual trip destination prediction in public transportation systems. IEEE Trans Intell Transp Syst 23:1–14
14. Yuan J, Zheng Y, Xie X (2012) Discovering regions of different functions in a city using human mobility and pois. In: The 18th ACM SIGKDD International conference on knowledge discovery and data mining. KDD ’12, Beijing, China, August 12-16, 2012, pp. 186–194
15. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. arXiv preprint arXiv:1706.03762
16. Yu C, Ma X, Ren J, Zhao H, Yi S (2020) Spatio-temporal graph transformer networks for pedestrian trajectory prediction. Eur Conf Comput Vis 12357:507–523
17. Xu M, Dai W, Liu C, Gao X, Lin W, Qi G-J, Xiong H (2020) Spatial-temporal transformer networks for traffic flow forecasting. arXiv preprint arXiv:2001.02908
18. Yu B, Yin H, Zhu Z (2018) Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In: IJCAI, pp. 3634–3640
19. Ge L, Li H, Liu J, Zhou A (2019) Temporal graph convolutional networks for traffic speed prediction considering external factors. In: MDM, pp. 234–242
20. Fang S, Zhang Q, Meng G, Xiang S, Pan C (2019) GSTNET: Global spatial-temporal network for traffic flow prediction. In: IJCAI, pp. 2286–2293

21. Wu Z, Pan S, Long G, Jiang J, Zhang C (2019) Graph wavenet for deep spatial-temporal graph modeling. In: IJCAI

22. Bai S, Kolter JZ, Koltun V (2018) An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv:1803.01271

23. Shi X, Chen Z, Wang H, Yeung D-Y, Wong W-K, Woo W-c (2015) Convolutional lstm network: a machine learning approach for precipitation nowcasting. arXiv preprint arXiv:1506.04214

24. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907

25. Wang Y, Yin H, Chen H, Wo T, Xu J, Zheng K (2019) Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modeling. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery and data mining, pp. 1227–1235

26. Chu K-F, Lam AYS, Li VOK (2020) Deep multi-scale convolutional LSTM network for travel demand and origin-destination predictions. IEEE Trans Intell Transp Syst 21(8):3219–3232

27. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980

28. Park C, Lee C, Bahng H, Tae Y, Jìn S, Kim K, Ko S, Choo J (2020) St-grat: A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed. In: Proceedings of the 29th ACM international conference on information and knowledge management, pp. 1215–1224

29. Xue Q, Guo T, Chen Y, Xiao Y, Wang X, Zhao BY (2020) Deep graph convolutional networks for incident-driven traffic speed prediction. In: Proceedings of the 29th ACM International conference on information and knowledge management, pp. 1665–1674

30. Zhang X, Huang C, Xu Y, Xia L (2020) Spatial-temporal convolutional graph attention networks for citywide traffic flow forecasting. In: Proceedings of the 29th ACM international conference on information and knowledge management, pp. 1853–1862

31. Yang X, Xue Q, Ding M, Wu J, Gao Z (2021) Short-term prediction of passenger volume for urban rail systems: a deep learning approach based on smart-card data. Int J Prod Econ 231:107920

32. Yang X, Xue Q, Yang X, Yin H, Qu Y, Li X, Wu J (2021) A novel prediction model for the inbound passenger flow of urban rail transit. Inf Sci 566:347–363

33. Zhang D, Xiao F, Shen M, Zhong S (2021) Dnca: a novel dynamic node-edge attention network for origin-destination demand prediction. Transp Res Part C Emerg Technol 122:102851

34. Tang K, Cao Y, Chen C, Yao J, Tan C, Sun J (2021) Dynamic origin-destination flow estimation using automatic vehicle identification data: a 3D convolutional neural network approach. Comput-Aided Civil Infrastructure Eng 36(1):30–46

35. Zhao J, Rahbee A, Wilson NH (2007) Estimating a rail passenger trip origin-destination matrix using automatic data collection systems. Comput-Aided Civil Infrastructure Eng 22(5):376–387

36. Hu J, Yang B, Guo C, Jensen CS, Xiong H (2020) Stochastic origin-destination matrix forecasting using dual-stage graph convolutional, recurrent neural networks. In: 2020 IEEE 36th International conference on data engineering (ICDE), pp. 1417–1428

37. Xiong X, Ozbay K, Jin L, Feng C (2019) Dynamic prediction of origin-destination flows using fusion line graph convolutional networks. CoRR arxiv:1905.00406

38. Duan Z, Zhang K, Chen Z, Liu Z, Tang L, Yang Y, Ni Y (2019) Prediction of city-scale dynamic taxi origin-destination flows using a hybrid deep neural network combined with travel time. IEEE Access 7:127816–127832

39. Liu L, Qiu Z, Li G, Wang Q, Ouyang W, Lin L (2019) Contextualized spatial-temporal network for taxi origin-destination demand prediction. IEEE Trans Intell Transp Syst 20(10):3875–3887

40. Shi H, Yao Q, Guo Q, Li Y, Zhang L, Ye J, Li Y, Liu Y (2020) Predicting origin-destination flow via multi-perspective graph convolutional network. In: 2020 IEEE 36th International conference on data engineering (ICDE), pp. 1818–1821

41. Ke J, Qin X, Yang H, Zheng Z, Zhu Z, Ye J (2021) Predicting origin-destination ride-sourcing demand with a spatio-temporal encoder-decoder residual multi-graph convolutional network. Transp Res Part C: Emerg Technol 122:102858

42. Ma J, Chan J, Rajasegarar S, Ristanoski G, Leckie C (2020) Multi-attention 3d residual neural network for origin-destination crowd flow prediction. In: 2020 IEEE International conference on data mining (ICDM), pp. 1160–1165

43. Barceló Bugeda J, Montero Mercadé L, Bullejos M, Serch O, Carmona Bautista C (2012) A kalman filter approach for the estimation of time dependent od matrices exploiting bluetooth traffic data collection. In: TRB 91st Annual meeting compendium of papers DVD, pp. 1–16

44. Zheng F, Liu J, Zuylen Hv, Wang K, Liu X, Li J (2019) Dynamic OD prediction for urban networks based on automatic number plate recognition data: Parametric vs. non-parametric approaches. In: 2019 IEEE intelligent transportation systems conference (ITSC), pp. 4037–4042

45. Ma Y, Kuik R, van Zuylen HJ (2013) Day-to-day origin-destination tuple estimation and prediction with hierarchical Bayesian networks using multiple data sources. Transp Res Rec 2343(1):51–61

46. Zhang J, Zheng Y, Qi D (2017) Deep spatio-temporal residual networks for citywide crowd flows prediction. In: Thirty-first AAAI conference on artificial intelligence

47. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S (2020) End-to-end object detection with transformers. In: European conference on computer vision, pp. 213–229

48. Li, J., Wang, Y., McAuley, J (2020) Time interval aware self-attention for sequential recommendation. In: Proceedings of the 13th International conference on web search and data mining, pp. 322–330

49. Xia L, Huang C, Xu Y, Dai P, Zhang B, Bo L (2020) Multiplex behavioral relation learning for recommendation via memory augmented transformer network. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, pp. 2397–2406

50. Cai L, Janowicz K, Mai G, Yan B, Zhu K (2020) Traffic transformer: capturing the continuity and periodicity of time series for traffic forecasting. Trans GIS 24(3):736–755

51. Zheng C, Fan X, Wang C, Qi J (2020) Gman: a graph multi-attention network for traffic prediction. Proc AAAI Conf Artif Intell 34:1234–1241

52. Arcos-García Á, Alvarez-Garcia JA, Soria-Morillo LM (2018) Deep neural network for traffic sign recognition systems: an analysis of spatial transformers and stochastic optimisation methods. Neural Netw 99:158–165

53. Li S, Jin X, Xuan Y, Zhou X, Chen W, Wang Y-X, Yan X (2019): Predicting origin-destination flow via multi-perspective graph convolutional network. In: Proceedings of the 29th ACM international conference on information and knowledge management, pp. 1215–1224

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.