STP-TrellisNets+: Spatial-Temporal Parallel TrellisNets for Multi-Step Metro Station Passenger Flow Prediction

Junjie Ou, Jiahui Sun, Yichen Zhu, Haiming Jin, *Member, IEEE*, Yijuan Liu, Fan Zhang, *Member, IEEE*, Jianqiang Huang, *Member, IEEE*, and Xinbing Wang, *Senior Member, IEEE*

Abstract—The drastic increase of metro passengers in recent years inevitably causes the overcrowdedness in the metro systems. Accurately predicting passenger flows at metro stations is critical for efficient metro system management, which helps alleviate such overcrowdedness. Compared to the prevalent next-step prediction, multi-step passenger flow prediction could prominently increase the prediction duration and reveal finer-grained passenger flow variations, which better helps metro system management. Thus, in this paper, we address the problem of *multi-step metro station passenger (MSP) flow prediction*. In light of MSP flows' unique spatial-temporal characteristics, we propose *STP-TrellisNets+*, which for the first time augments the newly-emerged temporal convolutional framework *TrellisNet* for multi-step MSP flow prediction. The temporal module of STP-TrellisNets+ (named *CP-TrellisNetsED*) employs a Closeness TrellisNet followed by a *Periodicity TrellisNets-based Encoder-Decoder (P-TrellisNetsED)* to jointly capture the short- and long-term temporal correlation of MSP flows. In parallel to CP-TrellisNetsED, its spatial module (named *GC-TrellisNetsED*) adopts a novel transfer flow-based metric to characterize the spatial correlation among MSP flows, and implements another TrellisNetsED on multiple *diffusion graph convolutional networks (DGCNs)* in time-series order to capture the dynamics of such spatial correlation. Extensive experiments with two large-scale real-world automated fare collection datasets demonstrate that STP-TrellisNets+ outperforms the state-of-the-art baselines.

Index Terms—Metro station passenger flow, multi-step prediction, TrellisNet, encoder-decoder, diffusion graph convolution

1 Introduction

Nowadays, with the rapid development of metro systems, a vastly increasing number of urban residents choose to take the metro for their daily transportation (e.g., commuting, shopping). However, such a drastic increase in the number of metro passengers has apparently overcrowded the metro systems in many cities, which threatens the public safety and exacerbates the difficulty of metro system management. Clearly, knowing beforehand the number of passengers entering and exiting metro stations is critical for proper metro line scheduling and timely metro staff preallocation, which could help handle the aforementioned overcrowdedness in metro systems.

Existing methods for predicting metro station passenger flow [2], [3], [4], [5], [6] focus on next-step prediction, which aim to forecast the passenger flow in the next time step. Nevertheless, *multi-step prediction* is more beneficial due to the following two reasons. On one hand, given a fixed length for a time step, multi-step prediction can extend the prediction timespan, which offers a farther-sighted view for future passenger flows. On the other hand, predicting the passenger flows over multiple shorter time steps within the original one can help reveal finer-grained passenger flow variations. Therefore, in this paper, we study the problem of *multi-step metro station passenger (MSP) flow prediction* with the data collected from the automated fare collection (AFC) system.

In practice, the MSP flows in a metro system usually show strong spatial-temporal correlations. That is, the MSP flow of one station is both temporally correlated with those in the past, and spatially correlated with those of other stations. As a result, capturing such spatial-temporal correlations is essential for accurately predicting MSP flows. Although a series of studies have effectively accomplished other spatial-temporal prediction tasks [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], such as traffic volume and highway speed prediction, they invariably become suboptimal for MSP flow prediction due to the unique spatial-temporal characteristics of MSP flows. In what follows, we will elaborate upon the design of our novel deep learning framework optimized to better capture MSP flows' spatial-temporal correlations.

From the *temporal* perspective, RNN-based models [14], [15], [16], [17], [18] have been regarded as effective to

- *Junjie Ou, Jiahui Sun, Yichen Zhu, Haiming Jin, Yijuan Liu, and Xinbing Wang are with Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: {j_michael, jhsun1997, zyc_ieee, jinhaiming, yooeachen, xwang8}@sjtu.edu.cn.*
- *Fan Zhang is with SIAT, Chinese Academy of Sciences, Beijing 100045, China. E-mail: zhangfan@siat.ac.cn.*
- *Jianqiang Huang is with Alibaba Damo Academy, Hangzhou, Zhejiang 311121, China. E-mail: jianqiang.jqh@gmail.com.*

Manuscript received 7 February 2021; revised 16 May 2022; accepted 18 June 2022. Date of publication 1 July 2022; date of current version 5 June 2023.
This work was supported in part by NSF China under Grants 42050105, U21A20519, U20A20181, 61902244, and 62061146002, and in part by the Major Scientific Research Project of Zhejiang Lab under Grant 2019DB0ZX01.
(Corresponding authors: Haiming Jin and Xinbing Wang.)
Recommended for acceptance by L. Xiong.
Digital Object Identifier no. 10.1109/TKDE.2022.3187690

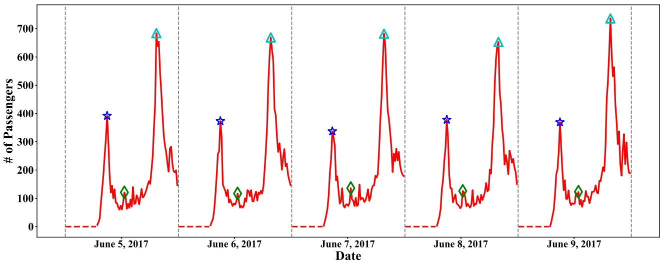
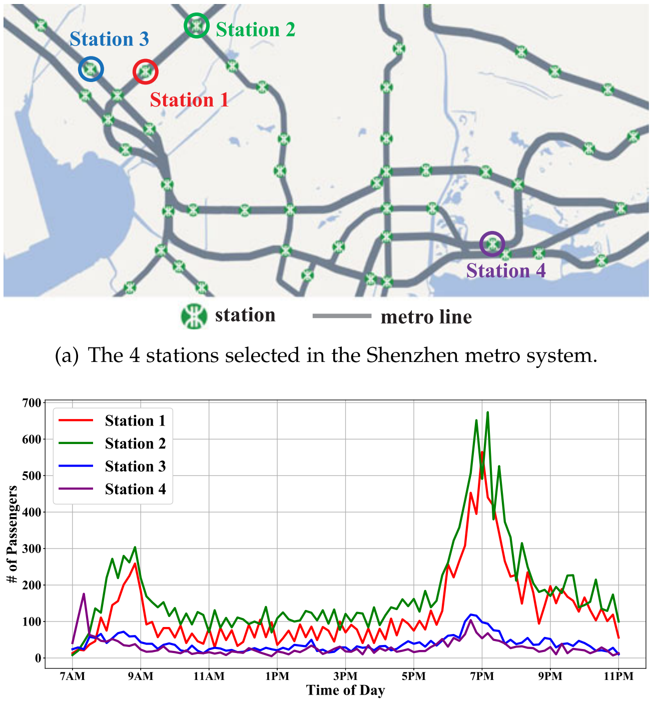


Fig. 1. An example that shows the approximate periodicity and discontinuity of MSP flows, where the curve represents the MSP flows in 5 consecutive days of metro station Fanshen in Shenzhen, the same marker corresponds to the same time of day, and the dashed lines indicate that the metro station was closed in those intervals.



(a) The 4 stations selected in the Shenzhen metro system.

(b) The MSP flows of the 4 stations marked in Figure 2(a) in one day.

Fig. 2. The MSP flows of 4 metro stations in Shenzhen in one day, where close stations show similar (e.g., Stations 1 and 2) or distinctive (e.g., Stations 1 and 3) flow patterns, and distant stations show similar (e.g., Stations 3 and 4) or distinctive (e.g., Stations 1 and 4) flow patterns as well.

capture the non-linear temporal correlation of short time series. However, the MSP flows at least in the past few hours have to be considered to predict those in the future. Clearly, the excessive number of data samples contained in such long historical MSP flows will inevitably make the training of RNN rather inefficient, because of its inherent recurrent architecture and serial operations. Thus, instead of RNN, we adopt the *Temporal Convolutional Network (TCN)* [19], [20], [21], whose parallel framework combined with its exponentially enlarging receptive fields across the convolution layers enables much faster training than RNN. More specifically, we employ a novel TCN variant called *Trellis Network (TrellisNet)* [22], which inherits the above merits of the traditional TCN and additionally enhances its ability to capture the non-linearity of time series by combining input injection with a non-linear transformation.

Though promising, TrellisNets could not unleash its full power unless the temporal characteristics of MSP flows are appropriately taken into consideration. As shown in Fig. 1, on one hand, the MSP flow of a station typically shows a similar pattern within each day. If handled appropriately, such approximate periodicity could offer useful side information for predicting future flows. On the other hand, the MSP flows between two adjacent days are naturally discontinuous, because the metro system typically closes between mid-night and early morning. Properly dealing with such flow discontinuity then becomes another critical factor that affects the prediction performance. To address the above problems, we propose inputing the MSP flows in the past few hours into a TrellisNet, whose output is then *skip-concatenated* with those at the same time instance in the past few days which integrates the short-term *closeness* and long-term *periodicity* data with a skip-concatenation effectively addressing the flow discontinuity issue.

With the above skip-concatenated data, the next coming issue is how to carry out accurate prediction over multiple future time steps. A recent significant breakthrough for multi-step prediction is enabled by the RNN-based Encoder-Decoder [18], whose encoder employs chain-structured RNNs encoding the input sequence to an intermediate state and decoder employs several other chain-structured RNNs taking the encoded state as the initial state and producing the multi-step results one by one. Nevertheless, this approach intrinsically causes the issue of error accumulation in each future step, because the subsequent RNN's prediction takes as input the result of the former one. To address this issue, we design a novel *TrellisNets-based Encoder-Decoder (TrellisNetsED)*. First, TrellisNetsED employs an Encoder TrellisNet encoding the above skip-concatenated data to a shorter encoded sequence. Then, another Decoder TrellisNet follows and applies parallel temporal convolution on the encoded sequence to yield an equal-length sequence containing the multi-step prediction results. In this way, the proposed TrellisNetsED outputs the temporal-correlated multi-step results yet with each step's final output independent with each other, which can effectively ease the error accumulation issue inherent in RNN-based Encoder-Decoder.

Taking all the above issue into consideration, we propose a novel framework named *Closeness-Periodicity TrellisNets-based Encoder-Decoder (CP-TrellisNetsED)* with a *Closeness TrellisNet (C-TrellisNet)* taking as input the short-term closeness data, whose output is skip-concatenated with long-term periodicity data and is further fed into a *Periodicity TrellisNets-based Encoder-Decoder (named P-TrellisNetsED)*. In this way, the CP-TrellisNetsED outputs the multi-step results which integrate the short- and long-term temporal characteristics of MSP flows.

From the *spatial* perspective, using CP-TrellisNetsED alone is far from enough to capture the spatial correlations of MSP flows. Inspired by the recent success of graph convolutional networks (GCNs) [15], [16], [17], [18], [19], [20], [21] on learning graph-structured correlations, we propose to empower TrellisNets with the ability to learn spatial correlations by integrating it with GCNs.

TABLE 1
Notation Table

Notation	Description
n, q	The number of metro stations and the maximum index for a time interval.
$f_{i,t}^{\mathrm{in}}, f_{i,t}^{\mathrm{out}}$	The MSP inflow and outflow of station i in time interval t.
ξ_{ij}^{t}	The transfer flow-based metric representing the spatial correlation among stations.
$\mathbf{X}_{1:T}$	The input closeness data in past T time intervals.
$\mathbf{P}_{1:DS}$	The input S steps' day-periodicity data in past D days.
$\mathbf{M}_t$	The adjacency matrix of directed metro graph in time interval t.
$\mathbf{Y}_{1:n,t}$	The input node features of DGCN in time interval t.
$\Gamma^{(c)}(\cdot)$	The entire operation of C-TrellisNet with c layers.
$\Upsilon^{(S)}(\cdot)$	The entire operation of TrellisNetsED with S steps' output.
$\Psi(\cdot)$	The entire operation of diffusion graph convolution.
$\widehat{\mathbf{X}}_{1:S}$	The eventual output of CP-TrellisNetsED module.
$\widehat{\mathbf{Y}}_{1:S}$	The eventual output of GC-TrellisNetsED module.
$\widehat{\mathbf{Z}}_{1:S}$	The final output of STP-TrellisNets+.

The first step towards such integration is to construct an appropriate graph representation for the metro system. As shown in Fig. 2, geographical attributes, such as distance and connectivity, which oftentimes well characterize the spatial correlation in other scenarios, do not necessarily determine that among MSP flows. In practice, the MSP flow of one station depends on the passenger flows transferred between it and other stations, and thus the transfer flows could naturally serve as the metric to represent the spatial correlation in our metro scenario. As a result, we propose to represent the metro system as a *metro direct graph* with vertices corresponding to metro stations, and edge weights proportional to the *transfer flows* between pairs of stations.

Clearly, the edge weights of the metro direct graph are dynamic due to the time-varying nature of transfer flows. In order to capture such dynamic spatial correlation, we propose a novel framework named *Graph Convolutional TrellisNets-based Encoder-Decoder (GC-TrellisNetsED)*, which integrates TrellisNets-based Encoder-Decoder with graph convolutions. More specifically, GC-TrellisNetsED contains multiple *Diffusion Graph Convolutional Networks (DGCNs)* [23] placed along the time steps with each of them taking the metro directed graph at the corresponding time step as input. Then, the outputs of these DGCNs are combined and fed to a TrellisNets-based Encoder-Decoder in time-series order. In this way, the GC-TrellisNetsED outputs the multi-step results which captures the aforementioned dynamic spatial correlation.

Collectively from the *spatial-temporal* perspective, we aggregate the outputs of the parallel CP-TrellisNetsED and GC-TrellisNetsED to obtain the multi-step predicted MSP flows. As a result, the overall architecture named *Spatial-Temporal Parallel TrellisNets (STP-TrellisNets+)* could well capture the short- and long-term temporal correlation, as well as the dynamic spatial correlations of MSP flows.

To summarize, this paper makes the following contributions.

- In this paper, we address the multi-step MSP flow prediction problem by proposing STP-TrellisNets+, a novel deep learning framework which learns the spatial-temporal correlations of MSP flows. We adopt TrellisNet for MSP flow's temporal correlation learning and further design a novel TrellisNets-based Encoder-Decoder structure to fulfill the task of multi-step forecasting. To the best of our knowledge, this paper is the first one that adopts and augments TrellisNets for multi-step spatial-temporal prediction tasks.
- We propose a novel framework CP-TrellisNetsED which implements a TrellisNet followed by a TrellisNets-based Encoder-Decoder in serial with a skip-concatenation to jointly capture the short- and long-term temporal correlation of MSP flows and address the flow discontinuity issue.
- We design a novel transfer flow-based metric to represent the spatial correlations among MSP flows, and further propose a novel framework GC-TrellisNetsED consisting of multiple DGCNs in time-series order with their outputs connected to a TrellisNets-based Encoder-Decoder, which captures the dynamics of such correlation.
- We conduct extensive experiments on two real-world large-scale AFC datasets with about 1.5 billion records within 30 days in Shenzhen, China and about 70 million records within 25 days in Hangzhou, China. The experimental results show that our proposed STP-TrellisNets+ outperforms the existing baseline methods.

2 Methodology

In this section, we give a formal description of the multi-step MSP flow prediction problem, as well as an overview and the design details of the proposed STP-TrellisNets+. All frequently used notation is shown in Table 1.

2.1 Overview

We consider an urban metro system with a set of n stations denoted as $\{1, 2, \ldots, n\}$ and split the overall timeline (e.g., 1 mo) into a set of time intervals, denoted as $\{0, 1, \ldots, q\}$, with equal length (e.g., 10 minutes). Next, we introduce in Definition 1 the concepts of MSP inflow and outflow and in Definition 2 the multi-step MSP flow prediction problem addressed in this paper.

Definition 1 (MSP Inflow and Outflow). *For a metro station i, the MSP inflow $f_{i,t}^{\mathrm{in}}$ and outflow $f_{i,t}^{\mathrm{out}}$ in time interval t are defined respectively as the number of passengers entering and exiting station i during time interval t.*

Definition 2 (Multi-Step MSP Flow Prediction Problem). *Given the historical flow data of the metro system until time interval t, the MSP flow prediction problem aims to predict the MSP outflow of each metro station*[1] *in the next S time intervals $t+1, t+2, \ldots, t+S$.*

[1] In this paper we focus on predicting MSP outflow, and leave MSP inflow prediction in our future work.

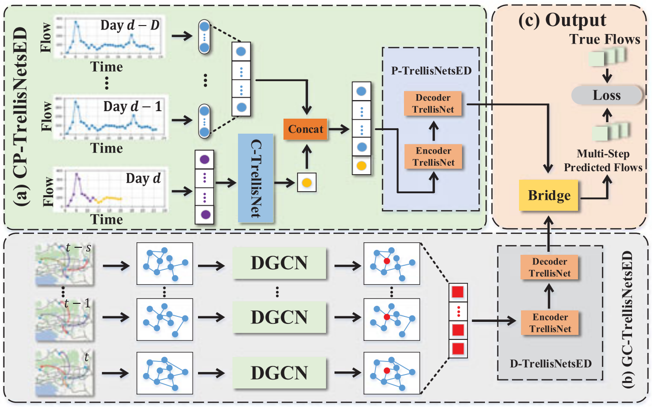


Fig. 3. The overall architecture of STP-TrellisNets+, where d represents the current day and FC denotes the Fully Connected layer. The MSP inflows and MSP outflows history are taken as input for MSP flow prediction.

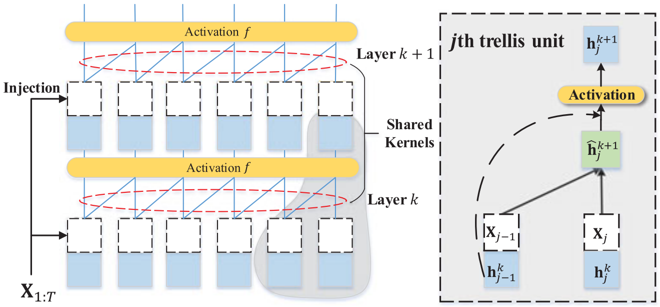


Fig. 4. An example of the TrellisNet architecture for MSP flow prediction.

To address the above multi-step MSP flow prediction problem, we propose a novel deep learning framework *Spatial-Temporal Parallel TrellisNets+ (STP-TrellisNets+)* shown in Fig. 3, which consists of a temporal module *Closeness-Periodicity TrellisNets-based Encoder-Decoder (CP-TrellisNetsED)*, a spatial module *Graph Convolutional TrellisNets-based Encoder-Decoder (GC-TrellisNetsED)*, and an aggregation module Output. The CP-TrellisNetsED takes the historical MSP outflows as input, and outputs a result that captures both the short-term and long-term temporal correlation of MSP outflows. The GC-TrellisNetsED takes the historical MSP inflows and multiple metro directed graphs along the time steps as input, and outputs a result that captures the dynamic spatial correlation of MSP outflows. Then, the Output module employs a Bridge operation to aggregate the outputs from the CP-TrellisNetsED and the GC-TrellisNetsED into the final multi-step prediction result. The details of each part will be described in the following Sections 2.2, 2.3, and 2.4.

2.2 Long Short-Term Temporal Correlation: CP-TrellisNetsED

In this work, we adopt a special temporal convolutional network characterized by weight sharing and input injection with a non-linear transformation called *TrellisNet* [22] to capture the non-linear temporal correlation of MSP flows. The reasons for which we adopt TrellisNet are two-fold. On one hand, its temporal convolutional architecture with a weight sharing mechanism enables stable gradient, fast training speed, and exponentially large receptive field size to better handle the long historical MSP flow sequence than recurrent architectures. On the other hand, its input injection, which integrates the deep features with the original input sequence, followed by a non-linear transformation improves the performance of the temporal convolutional architecture for capturing the non-linearity of MSP flows.

Considering the discontinuity between the long-term and short-term historical MSP outflows, we propose the CP-TrellisNetsED which first uses a Closeness TrellisNet (C-TrellisNet) to capture the short-term temporal correlation, followed by another *Periodicity TrellisNets-based Encoder-Decoder (P-TrellisNetsED)*, a structure we design for predicting the multi-step MSP flows, to further capture the long-term temporal correlation.

2.2.1 Short-Term Temporal Correlation: C-TrellisNet

For the whole metro system, in time interval t, we use an n-dimensional vector $\mathbf{x}_t = (f_{1,t}^{\text{out}}, f_{2,t}^{\text{out}}, \ldots, f_{n,t}^{\text{out}}) \in \mathbb{R}^n$ to represent the MSP outflows of the n stations. To predict the MSP outflows in next S time intervals, we use the MSP outflows in the past T time intervals as the historical input data. As a result, we have a length-T MSP outflow data sequence $\mathbf{X}_{1:T}$ as the input of C-TrellisNet given as

$$\mathbf{X}_{1:T} = \left(\mathbf{x}_{t-(T-1)}, \mathbf{x}_{t-(T-2)}, \ldots, \mathbf{x}_t\right).$$

As illustrated in Fig. 4, a full TrellisNet is built by stacking its basic units across the time steps and layers to constitute a trellis-like network. Specifically, the jth unit in layer k of a TrellisNet consists of the hidden output $\mathbf{h}_{j-1}^k \in \mathbb{R}^n$ and $\mathbf{h}_j^k \in \mathbb{R}^n$ from the previous layer $k-1$ and an injection of vectors $\mathbf{X}_{j-1}$ and $\mathbf{X}_j$ from the input sequence. The transformation in the jth unit is represented as the following Equation (1),

$$\begin{cases} \widehat{\mathbf{h}}_j^{k+1} = W_1\left[\mathbf{X}_{j-1} \parallel \mathbf{h}_{j-1}^k\right] + W_2\left[\mathbf{X}_j \parallel \mathbf{h}_j^k\right] \\ \mathbf{h}_j^{k+1} = f\left(\widehat{\mathbf{h}}_j^{k+1}, \mathbf{h}_{j-1}^k\right) \end{cases}, \tag{1}$$

where $\widehat{\mathbf{h}}_j^{k+1} \in \mathbb{R}^n$ is the pre-activation output, $\mathbf{X}_{j-1}$ and $\mathbf{X}_j$ correspond to the $(j-1)$th and jth vector in the input sequence, $\parallel$ is the concatenation operator, W_1 and W_2 are kernel weights, $\mathbf{h}_j^{k+1} \in \mathbb{R}^n$ represents the output of the jth unit in layer k, and $f : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ is a nonlinear activation function applied to $\widehat{\mathbf{h}}_j^{k+1}$ and $\mathbf{h}_{j-1}^k$.

We apply the above transformation procedure across all time steps and all layers as shown in Fig. 4, using the same kernel weight matrix. Mathematically, given the input MSP outflow data sequence $\mathbf{X}_{1:T}$, the computation at each layer k of C-TrellisNet can be summarized as

$$\mathbf{h}_{1:T}^{k+1} = f\left(\left(\mathbf{h}_{1:T}^k \parallel \mathbf{X}_{1:T}\right) * \mathbf{W},\ \mathbf{h}_{1:T-1}^k\right), \tag{2}$$

where $*$ denotes the 1D causal convolution operation with zero padding that convolves the output of the previous layer with only data from the past time intervals, and $\mathbf{W}$ denotes the kernel weight matrix parameters which is shared across all layers. Note that, similar to the generic temporal convolution network, we add a dilation factor in the 1D convolution operation to enlarge our C-TrellisNet's receptive fields.

At level 0, we initialize $\mathbf{h}^0_{1:T} = \mathbf{0}$. After stacking c layers, we take the last output of the cth layer (i.e., $\mathbf{h}^{c+1}_T$) which is an n-dimensional vector as the output of C-TrellisNet, i.e.,

$$\Gamma^{(c)}(\mathbf{X}_{1:T}) = \mathbf{h}^{c+1}_T, \tag{3}$$

where $\Gamma^{(c)}(\cdot)$ denotes the entire operation of C-TrellisNet with c layers.

2.2.2 Long-Term Temporal Correlation: P-TrellisNetsED

The C-TrellisNet introduced above takes only the outflows within the past few *close* time intervals (i.e., several hours) as input. However, *long-term temporal correlation* also should be considered in our MSP flow prediction problem.

Apparently, the values of the MSP flows among day-periodic time intervals (i.e., the same time intervals in the past days) are very close to each other. Such phenomenon indicates that MSP flows have an approximate long-term periodicity. To capture this long-term periodicity in MSP flows, we define the *day-periodicity data* as the MSP outflows in the same time intervals with the predicting target intervals $t+1, \ldots, t+S$ in the past D days, given as

$$\mathbf{P}_{1:DS} = \left(\underbrace{\overbrace{\mathbf{x}_{t+1-Dm}, \ldots, \mathbf{x}_{t+S-Dm}}^{S}, \ldots, \overbrace{\mathbf{x}_{t+1-m} \cdots, \mathbf{x}_{t+S-m}}^{S}}_{DS} \right),$$

where m denotes the number of time intervals in a day.

Furthermore, we observe that MSP flows between two adjacent days are discontinuous in time. In light of this, to better capture the long- and short-term correlation simultaneously, we concatenate the day-periodicity data with the output of C-TrellisNet (i.e., $\mathbf{h}^{c+1}_T$) as $\mathbf{I}_{1:DS+1} = \mathbf{P}_{1:DS} \mid\mid \mathbf{h}^{c+1}_T$ and feed it into a Periodicity TrellisNets-based Encoder-Decoder (P-TrellisNetsED). In what follows, we will detail the structure of P-TrellisNetsED.

As shown in Fig. 5, the P-TrellisNetsED contains an Encoder and a Decoder which are two TrellisNets respectively. The Encoder TrellisNet encodes the original input sequence (i.e., $\mathbf{I}_{1:DS+1}$) and yields a length-S encoded sequence (i.e., $\mathbf{L}_{encoded}$) by extracting the last hidden state of each layer (referred to Copy operation) and concatenating them in order. With the temporal convolution operation, the information of the entire input sequence is aggregated to the encoded sequence.

More specifically, according to Equation (2), the computation at each layer of Encoder TrellisNet are

$$\mathbf{he}^S_{1:DS+1} = f\big(\big(\mathbf{he}^{S-1}_{1:DS+1} \mid\mid \mathbf{I}_{1:DS+1}\big) * \mathbf{W}_1,\ \mathbf{he}^{S-1}_{1:DS}\big),$$

where $\mathbf{he}^S_{1:DS+1}$ denotes the hidden states of layer S of Encoder TrellisNet and $\mathbf{W}_1$ denotes the kernel weight matrix parameters of Encoder TrellisNet. With the above transformation, we get the encoded sequence as

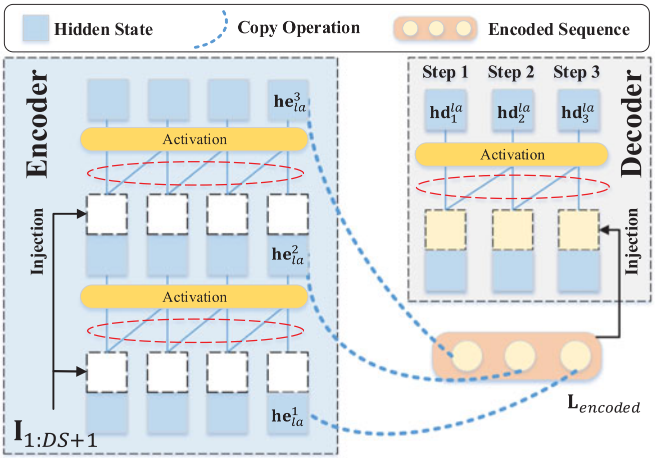


Fig. 5. An example of the TrellisNetsED architecture for multi-step MSP flow prediction.

$$\mathbf{L}_{encoded} = \text{Encoder}(\mathbf{I}_{1:DS+1}) = \{\underbrace{\mathbf{he}^S_{la}, \mathbf{he}^{S-1}_{la}, \ldots, \mathbf{he}^1_{la}}_{S}\},$$

where $\mathbf{he}^S_{la}$ represents the last hidden state of layer S of Encoder TrellisNet. Note that the hidden state at last layer $\mathbf{he}^S_{la}$ appears at the first position in the encoded sequence.

Then, another TrellisNet in Decoder takes the encoded sequence as input and produces an equal-length sequence composed of all hidden outputs of last layer. According to Equation (2), we have

$$\begin{aligned}\text{Decoder}(\mathbf{L}_{encoded}) &= \mathbf{hd}^{la}_{1:S} \\ &= f\big(\big(\mathbf{hd}^{la-1}_{1:S} \mid\mid \mathbf{L}_{encoded}\big) * \mathbf{W}_2,\ \mathbf{hd}^{la-1}_{1:S}\big),\end{aligned}$$

where $\mathbf{W}_2$ denotes the kernel weight matrix parameters of Decoder TrellisNet and $\mathbf{hd}^{la}_{1:S}$ denotes the hidden state of the last layer of Decoder TrellisNet storing the corresponding Sth step's prediction result of P-TrellisNetsED.

We express the entire operation of P-TrellisNetsED with S steps' outputs with $\Upsilon^{(S)}(\cdot)$ and thus our entire proposed CP-TrellisNetsED performs the procedure,

$$\widehat{\mathbf{X}}_{1:S} = \Upsilon^{(S)}\Big(\mathbf{P}_{1:DS} \mid\mid \Gamma^{(c)}(\mathbf{X}_{1:T})\Big), \tag{4}$$

where $\widehat{\mathbf{X}}_{1:S} \in \mathbb{R}^{n \times S}$ is the eventual output of the CP-TrellisNetsED consisting of each station i's result $\widehat{X}_{i,1:S}$.

2.3 Dynamic Spatial Correlation: GC-TrellisNetsED

Clearly, the CP-TrellisNetsED defined above well captures MSP outflows' temporal correlation. However, considering temporal correlation alone is typically not enough to predict MSP outflows accurately. For example, when a special activity is going to be held somewhere in a city, the passenger outflows of nearby stations may increase dramatically in the future, which is hard to be learned from temporal features only.

To address this issue, we design a novel structure called GC-TrellisNetsED which captures not only the *spatial correlation* among metro stations, but also the *dynamics* of such

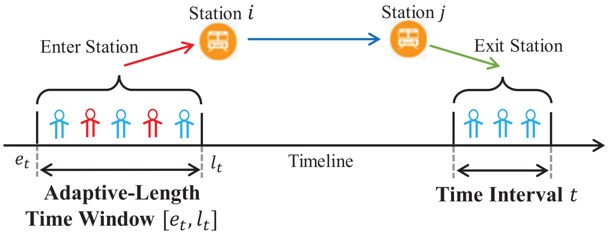


Fig. 6. Illustration example of transfer flow-based metric, where a blue person represents a passenger exiting station j during time interval t who takes the metro from station i, and a red person represents a passenger who enters station i during time window $[e_t, l_t]$, but does not exit at station j.

correlation. Specifically, we place multiple DGCNs along the time steps and connect the output of each DGCN to a Dynamic TrellisNets-based Encoder-Decoder (D-TrellisNetsED) in time-series order, as shown in Fig. 3b.

2.3.1 Diffusion Convolution on Metro Directed Graph

Clearly, the outflow of a station is influenced by the inflows of both the nearby and remote stations in the past few time intervals, and such influence is positively correlated with the transfer flow between two stations. Hence, we propose a novel *transfer flow-based metric* to represent the spatial correlation among metro stations given in Definition 3.

Definition 3. *(Transfer Flow-Based Metric) Given the time interval t and the origin-destination station pair (i, j), we denote $\mathcal{A}_{ij}^t$ as the set of passengers who enter station i at some time instances, and exit station j in time interval t. The time window during which the passengers in $\mathcal{A}_{ij}^t$ enter station i is denoted as $[e_t, l_t]$. Based on the above notations, we define the transfer flow-based metric ξ_{ij}^t as*

$$\xi_{ij}^t = \frac{A_{ij}^t}{O_i^t}, \tag{5}$$

where A_{ij}^t denotes the cardinality of $\mathcal{A}_{ij}^t$ and O_i^t denotes the number of passengers entering station i in the time window $[e_t, l_t]$.

Fig. 6 illustrates a toy example to facilitate the understanding of the proposed metric. In this example, $A_{ij}^t = 3$, $O_i^t = 5$, and thus $\xi_{ij}^t = 3/5$. Note that the length of the time interval $[e_t, l_t]$ in Definition 3 is adaptive w.r.t. the time interval t. With such adaptive-length time window, the metric can represent the spatial correlation between any pair of origin-destination stations (i, j), regardless of their distance. Then, with the metric defined in Definition 3, we model the whole metro system as a *metro directed graph* given in Definition 4.

Definition 4. *(Metro Directed Graph) At each time interval t, we model the whole metro system as a weighted directed graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t, \mathbf{M}_t)$, where $\mathcal{V} = \{1, 2, \ldots, n\}$ denotes the vertex set consisting of all metro stations, $\mathcal{E}_t$ denotes the edge set, and $\mathbf{M}_t = \{a_{ij}^t\} \in [0, 1]^{n\times n}$ denotes the adjacency matrix with ξ_{ij}^t's serving as the edge weights, i.e.,*

$$a_{ij}^t = \begin{cases} \xi_{ij}^t, & \text{if } \ i \neq j \\ 0, & \text{if } \ i = j \end{cases}. \tag{6}$$

By Definition 4, the metro directed graph models the metro system into a graph structure with adaptive edge weights, which we use as input to the *graph convolution network (GCN)* to capture the spatial correlation among MSP flows. According to [24], GCNs could be either spectral-based or spatial-based. The spectral-based models only apply to undirected graph due to the symmetry requirement of Laplacian matrix factorization, and are thus not suitable for our metro directed graph. Inspired by [18], [23], we integrate the diffusion process into spatial-based graph convolution and employ the *diffusion graph convolution network (DGCN)* on the metro directed graph to capture the spatial correlation of MSP flows.

More specifically, to predict the MSP outflows in next S time intervals $t+1, t+2, \ldots, t+S$, each DGCN in GC-TrellisNet takes an n-dimensional vector $\mathbf{Y}_{1:n,t}$ consisting of the n stations' MSP inflows at time interval t as the input node features, given as

$$\mathbf{Y}_{1:n,t} = \left(f_{1,t}^{\text{in}}, f_{2,t}^{\text{in}}, \ldots, f_{n,t}^{\text{in}}\right)^{\text{tr}},$$

where $(\cdot)^{\text{tr}}$ represents the transpose operation. Then, we define the diffusion graph convolution in DGCN as

$$\Psi(\mathbf{M}_{t+1}, \mathbf{Y}_{1:n,t}) = \mathbf{M}_{t+1} \cdot \mathbf{Y}_{1:n,t} \circ \theta_w, \tag{7}$$

where $\Psi(\cdot)$ denotes the defined diffusion graph convolution, $\circ$ denotes the element-wise product, and θ_w represents the model parameters of DGCN. In fact, DGCN models the information propagation on the metro directed graph by a diffusion process that transfers the input node feature to one of its neighboring nodes with a certain transition probability characterized by the transfer flow-based metric.

2.3.2 Temporal Convolution on DGCNs

Note that the calculation defined by Equation (7) is carried out at time interval t, but requires the adjacency matrix $\mathbf{M}_{t+1}$ which is not available at time interval t. To address this issue, GC-TrellisNetsED uses the D-TrellisNetsED which employs the same structure as P-TrellisNetsED on top of multiple DGCNs placed in time-series order, which captures the dynamics of the adjacency matrix $\mathbf{M}_t$.

Specifically, GC-TrellisNetsED uses $s+1$ DGCNs in parallel, and to predict the MSP outflows in next S time intervals, we feed into each DGCN respectively the adjacency matrices $\mathbf{M}_{t-s}, \mathbf{M}_{t-s+1}, \ldots, \mathbf{M}_t$ but the same node feature sequence $\mathbf{Y}_{1:n,t}$. Then, the D-TrellisNetsED takes the output of each DGCN along the time steps as its input sequence, and thus our proposed GC-TrellisNetsED in fact carries out the operation,

$$\widehat{\mathbf{Y}}_{1:S} = \Upsilon^{(S)}\left(\Psi(\mathbf{M}_{t-s}, \mathbf{Y}_{1:n,t}), \ldots, \Psi(\mathbf{M}_t, \mathbf{Y}_{1:n,t})\right), \tag{8}$$

where $\widehat{\mathbf{Y}}_{1:S}$ denotes the output of the D-TrellisNetsED.

2.4 Bridge and Prediction

As aforementioned, STP-TrellisNets+ integrates two different parts, i.e., the temporal module CP-TrellisNetsED and the spatial module GC-TrellisNetsED. Specifically, as

TABLE 2
Detailed Information of the Evaluated Datasets

Properties	Datasets	
	Shenzhen	Hangzhou
# of Stations	165	81
# of Records	1.5 billion	70 million
Time span	6/1/2017 - 6/30/2017	1/1/2019 - 1/25/2019

TABLE 3
Example of One Piece of AFC Record in the Datasets

Field	Value
User Card ID	685374531
Timestamp (ts)	20170604201430
In or Out (io)	In
Line Number	11
Station Name (sn)	Nanshan

shown in Fig. 3c, it merges the output of the CP-TrellisNetsED with that of GC-TrellisNetsED by a fusion (referred to as Bridge), and uses the merged value as predictive MSP outflow vector $\widehat{\mathbf{Z}}_{1:S}$ which contains next S steps' prediction result (i.e., $t+1, t+2, \ldots, t+S$). That is, the final output of the STP-TrellisNets+ satisfies that

$$\widehat{\mathbf{Z}}_{1:S} = \tanh\Big(\mathbf{W}_x \circ \widehat{\mathbf{X}}_{1:S} + \mathbf{W}_y \circ \widehat{\mathbf{Y}}_{1:S}\Big),$$

where $\mathbf{W}_x$ and $\mathbf{W}_y$ are the learnable parameters that measure how the results of temporal correlation learning and dynamic spatial correlation learning affect multi-step MSP flow prediction, and $\circ$ denotes again the element-wise multiplication.

In the training process, we aim to minimize the mean square error (MSE) represented by the loss function

$$Loss(\mathbf{\Theta}) = \frac{1}{bn}\sum_{a=1}^{b}\sum_{i=1}^{n}\sum_{\tau=1}^{S}\Big(Z^a_{i,t+\tau} - \widehat{Z}^a_{i,t+\tau}\Big)^2,$$

where b denotes the number of samples in a training batch and $\mathbf{\Theta}$ denotes all learnable parameters of the STP-TrellisNets+.

3 Experiments

In this section, we introduce the datasets, data preprocessing, the baseline methods with which we compare STP-TrellisNets+, the setup of our experiments and the corresponding experimental results.

3.1 Datasets

In this paper, we evaluate the performance of STP-TrellisNets+ on two large-scale real-world *automated fare collection (AFC)* datasets, namely Shenzhen and Hangzhou. The Shenzhen dataset is collected from the metro AFC system in the city of Shenzhen, China, consisting of about 1.5 billion AFC records from June 1st to June 30th, 2017. The Hangzhou dataset, made available for the Global Urban Computing AI Challenge 2019@TIANCHI,[2] contains about 70 million AFC records in the city of Hangzhou, China from January 1st to January 25th, 2019. Detailed information of the two datasets are given in Table 2 below.

3.2 Data Preprocessing

3.2.1 MSP Inflow and Outflow

Each piece of the AFC record in our datasets contains features including *user card ID*, *metro station name*, *inbound or outbound label*, and *recording timestamp*. The following Table 3 shows an example of one piece of record in the datasets.

We use $\mathcal{D}$ to denote the set of AFC records in one dataset (i.e., Shenzhen or Hangzhou). Then, we count the number of inbounds and outbounds of each metro station i in each time interval t to obtain the MSP inflow $f^{\text{in}}_{i,t}$, such that

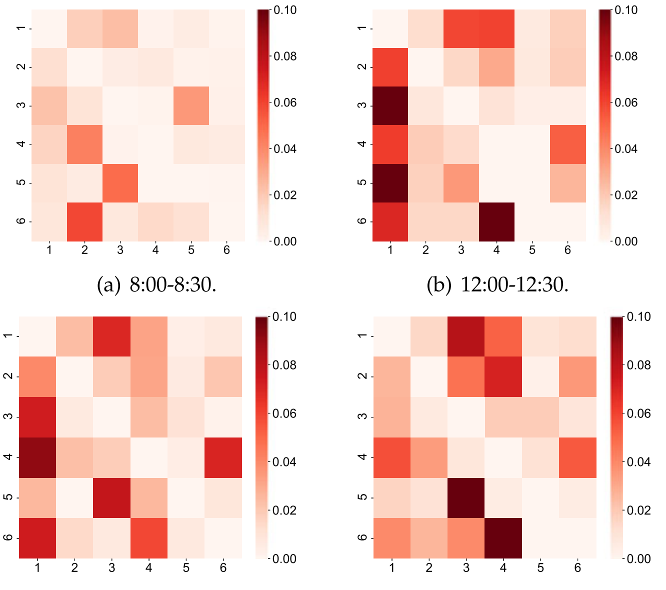
$$f^{\text{in}}_{i,t} = \sum_{r\in\mathcal{D}} \mathbb{1}\{r.sn = i \ \& \ r.io = \text{In} \ \& \ r.ts \in \text{interval} \quad t\},$$

and the MSP outflow $f^{\text{out}}_{i,t}$, such that

$$f^{\text{out}}_{i,t} = \sum_{r\in\mathcal{D}} \mathbb{1}\{r.sn = i \ \& \ r.io = \text{Out} \ \& \ r.ts \in \text{interval} \quad t\}.$$

3.2.2 Metro Directed Graph

According to Definition 4, we obtain ξ^t_{ij} from our AFC datasets to construct the adjacency matrix $\mathbf{M}_t$ that corresponds to each time interval t. Note that the starting and ending time instances of the adaptive-length time window $[e_t, l_t]$ is obtained by tracking the user card ID in the records. Furthermore, we use the inflow of each metro station obtained above to construct the node feature vector $\mathbf{Y}_{1:n,t}$. Fig. 7 shows the adjacency matrices in 4 different time intervals within one day with 6 stations indexed as Station $1, 2, \ldots, 6$ which correspond respectively to Station Shenzhenbei,



(a) 8:00-8:30.

(b) 12:00-12:30.

(c) 16:00-16:30.

(d) 22:00-22:30.

Fig. 7. Adjacency matrices of 6 stations in 4 different time intervals within a day.

[2] http://bit.ly/3bWORwC

Shangmeilin, Wuhe, Longhua, Tanglang, and Shangtang in the city of Shenzhen, China. From this figure, we can observe that such adjacency matrices change over time, which shows the dynamic nature of the spatial correlation among stations.

3.3 Baseline Methods

In this paper, we compare our STP-TrellisNets+ with the following baseline methods. For all the baseline methods, we use the outflows in the last few time intervals (i.e., $t-T,\dots,t$) as the historical data to predict the outflows in the following 3 time intervals (i.e., $t+1, t+2, t+3$).

- *Last-Value:* Last-Value returns the MSP outflow value of previous time interval (i.e., t) as the predicted outflow value of the future steps.
- *Historical Average (HA):* HA uses the average MSP outflow values in past one day as the predicted outflow value of the future steps.
- *Vector Autoregression (VAR):* VAR [25] is a multivariate generalization of autoregression, which captures the correlation among multiple time series. Here, we take the outflows of all stations in each time interval as a vector. The order of lags is set as 2.
- *Multi-Layer Perceptron (MLP):* MLP consists of multiple layers of neurons in a feed-forward way. We build an MLP model with a single hidden layer of 256 units.
- *Long Short-Term Memory (LSTM):* LSTM [26] is a class of RNN that is capable of capturing the long-term temporal correlation of time-series data. The hidden size of LSTM is set as 256.
- *Convolutional LSTM (ConvLSTM):* ConvLSTM [27] is a variant of LSTM, in which convolutional structure[3] is used to capture the spatial correlation of data. The convolution kernel size is set as 3.
- *Diffusion Convolutional Recurrent Neural Network (DCRNN):* DCRNN [18] is a spatial-temporal deep learning model, which combines diffusion convolutional network and gated recurrent unit to capture the spatial and temporal correlations of traffic data. The maximum step K for random walk is 3.
- *Spatio-Temporal Graph Convolitional Network (STGCN):* STGCN [13] formulates the traffic prediction problem on graphs and builds the model with complete convolutional structures. The channels in its three ST-Conv blocks are set as 64, 16, and 64 respectively, and the kernel sizes of both graph convolution and temporal convolution are 3.
- *Graph Wavenet:* Graph Wavenet [20] is a graph neural network model, which captures the spatial correlation with a diffusion convolution layer and learns the temporal correlation using generic TCN. The channels of all hidden layers are set as 32.

Besides, for all the GCN-based baselines, i.e., DCRNN, STGCN, and Graph WaveNet, we follow their default settings and construct the input metro graph utilizing the adjacency matrix among stations in metro lines.

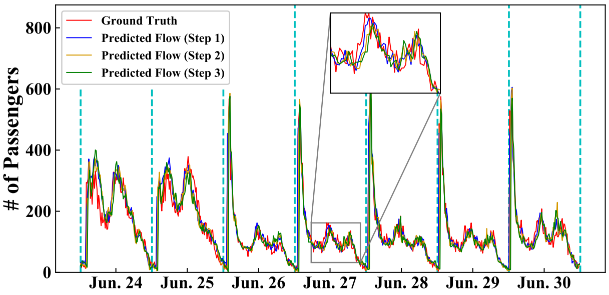


Fig. 8. Comparison of the multi-step MSP outflows predicted by STP-TrellisNets+ and the ground truth values of station Shaoniangong in the Shenzhen dataset.

3.4 Experimental Setup

In our experiments, we employ three widely-adopted metrics, namely Mean Absolute Error (MAE) such that

$$\mathrm{MAE} = \frac{1}{bn}\sum_{j=1}^{b}\sum_{i=1}^{n}\left|Z_{i,t+\tau}^{j} - \widehat{Z}_{i,t+\tau}^{j}\right|,$$

Root Mean Square Error (RMSE) such that

$$\mathrm{RMSE} = \sqrt{\frac{1}{bn}\sum_{j=1}^{b}\sum_{i=1}^{n}\left(Z_{i,t+\tau}^{j} - \widehat{Z}_{i,t+\tau}^{j}\right)^{2}},$$

and Mean Absolute Percentage Error (MAPE), such that

$$\mathrm{MAPE} = \frac{1}{bn}\sum_{j=1}^{b}\sum_{i=1}^{n}\left|\frac{Z_{i,t+\tau}^{j} - \widehat{Z}_{i,t+\tau}^{j}}{Z_{i,t+1}^{j}}\right|$$

to evaluate the performances of STP-TrellisNets+ and the baseline methods on τth predictive step.

The evaluation datasets are split into training, validation, and testing datasets with the ratio of $7:1:2$. For CP-TrellisNetsED, the input sequence lengths of C-TrellisNet and P-TrellisNetsED are adjusted to cover the fixed historical 10 time intervals and time span 3 days respectively, which correspond to the short- and long-term historical data respectively. For GC-TrellisNetED, we implement 3 DGCNs in parallel and use the 3 adjacency matrices of the past 3 time intervals as inputs.

3.5 Performance Evaluation

3.5.1 Comparison With Ground Truths

In Figs. 8 and 9, we visualize the comparison between the step-wise MSP outflows predicted by STP-TrellisNets+ and the ground truth values of two representative metro stations from two datasets respectively in one week. From the figures, we could observe that the MSP outflows predicted by STP-TrellisNets+ at all steps are fairly close to the ground truth values.

3.5.2 Comparison With Baseline Methods

We compare our STP-TrellisNets+ with all baseline methods under 2 different lengths w.r.t each step's time interval. Tables 4 and 5 summarize the 3 steps' prediction performances

[3] Because the metro system cannot be treated as an image-like structure, we treat the metro line as a 1D grid and use 1D CNN to capture the spatial correlation in our experiment.

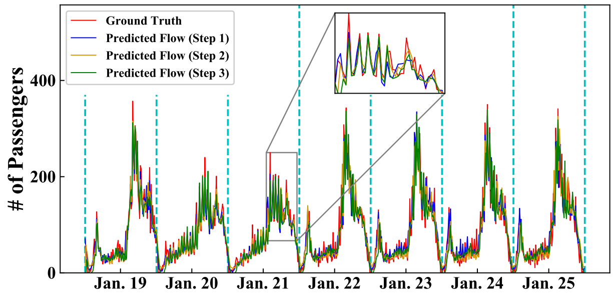


Fig. 9. Comparison of the multi-step MSP outflows predicted by STP-TrellisNets+ and the ground truth values of station Xiasha Jiangbin in the Hangzhou dataset.

of STP-TrellisNets+ and all baseline methods on two datasets under 10 and 30 minutes' time interval lengths respectively. Next, we will discuss some observations of the results and the rationale behind them.

First, it is easily observable that STP-TrellisNets+ achieves the best multi-step prediction performance in terms of all performance metrics both on short (10-minute) and long (30-minute) time interval. The 10-minute setting reports smaller MAE and RMSE than the 30-minute setting at the same step, because the latter has a higher fluctuation of MSP flow within one time interval, therefore leading to larger errors. Moreover, all the prediction methods' performance on Shenzhen dataset in terms of MAE and RMSE are smaller than on Hangzhou dataset. That is because Shenzhen dataset has much more AFC records than Hangzhou dataset hence offering the adequate training data for obtaining the smaller prediction errors. In contrast, we can observe the MAPE value show slightly higher on Shenzhen than on Hangzhou dataset. We analyze the reason is there are larger variations of MSP flow on Shenzhen dataset.

The two most naive methods Last-Value and HA, taking the last and average value of historical MSP flows as predictive flows respectively, both have undesirable prediction performance, which reflects the massive dynamic changes of MSP flow values and proves the necessity of proposing effective multi-step MSP flow prediction method. As VAR assumes the static linear temporal correlation among time intervals, it fails to capture the nonlinear correlation, and thus have the much worser performances on the prediction of MSP flows than other more complex models. Although nonlinear activation functions empower MLP and LSTM the ability to learn the nonlinear temporal correlation, MLP and LSTM do not capture spatial-temporal correlation in an explicit manner. ConvLSTM employs conventional convolutional structure to model the spatial correlation, which is not completely applicable to non-euclidean graph-structured data in our problem. STGCN employs graph convolution network to capture the spatial correlation of MSP outflows but fail to consider the dynamics of such correlation.

Among DCRNN, Graph Wavenet and our STP-TrellisNets+, all of which exploit the diffusion convolution to capture the spatial correlation while exploring different methods to extract the temporal correlation, the STP-TrellisNets+ achieves the best performance. The reasons behind are two-fold. On one hand, compared to DCRNN and Graph Wavenet, our STP-TrellisNets+ are more effective in capturing the temporal correlation with its TrellisNet structure and a proper consideration of the long-term periodicity of MSP outflows. On the other hand, our STP-TrellisNets+ are better than the other two models on capturing the dynamic spatial correlation of MSP outflows, because it uses a novel transfer flow-based metric to represent the spatial correlation and further integrates the diffusion convolution with TrellisNets to capture the dynamics of such correlation.

TABLE 4
Multi-Step Prediction Performance Under 10-Minute Time Interval Setting

Datasets	Models	MAE			RMSE			MAPE (%)		
		Step 1	Step 2	Step 3	Step 1	Step 2	Step 3	Step 1	Step 2	Step 3
Shenzhen	Last-Value	22.62	36.78	47.63	42.80	50.08	61.23	30.79	31.62	32.33
	HA	35.66	35.89	36.07	76.82	76.91	76.99	52.28	52.23	52.21
	VAR	22.57	28.76	35.60	43.67	48.90	57.37	30.05	31.89	32.14
	MLP	19.70	25.32	32.43	31.42	39.43	46.76	26.57	27.80	28.36
	LSTM	17.86	25.44	31.28	30.17	38.41	45.88	25.77	26.43	27.81
	ConvLSTM	17.73	25.23	32.89	29.43	37.61	44.45	23.67	24.28	25.90
	DCRNN	16.75	24.87	31.08	27.03	36.86	43.91	21.57	22.49	23.67
	STGCN	15.57	16.14	17.15	26.79	27.46	28.57	20.71	21.75	23.94
	Graph WaveNet	15.57	15.77	16.39	26.04	26.35	27.29	20.89	21.33	22.53
	STP-TrellisNets+	**14.60**	**14.86**	**14.95**	**24.54**	**25.10**	**25.31**	**14.72**	**15.15**	**15.24**
Hangzhou	Last-Value	29.78	35.34	43.22	51.27	58.35	64.88	24.59	25.66	26.69
	HA	32.06	32.15	32.21	70.04	70.05	70.06	26.38	26.44	26.50
	VAR	34.63	40.57	46.28	55.77	60.39	69.70	24.56	25.77	26.29
	MLP	32.64	38.63	44.87	48.29	57.86	63.90	25.73	26.80	27.35
	LSTM	31.65	37.89	44.10	46.38	55.32	60.78	24.76	25.66	26.30
	ConvLSTM	30.62	37.88	43.75	40.15	48.28	55.73	22.33	23.67	25.19
	DCRNN	28.72	36.59	42.93	36.88	43.99	52.00	21.34	22.89	23.52
	STGCN	22.40	23.22	24.13	36.25	37.72	39.01	18.25	19.16	20.65
	Graph WaveNet	22.04	22.44	22.93	35.35	36.29	37.14	17.94	18.21	19.19
	STP-TrellisNets+	**19.40**	**20.09**	**21.53**	**30.72**	**32.02**	**34.03**	**14.55**	**15.36**	**16.04**

TABLE 5
Multi-Step Prediction Performance Under 30-Minute Time Interval Setting

Datasets	Models	MAE			RMSE			MAPE (%)		
		Step 1	Step 2	Step 3	Step 1	Step 2	Step 3	Step 1	Step 2	Step 3
Shenzhen	Last-Value	64.55	72.32	79.03	162.89	170.14	179.22	32.84	33.90	34.28
	HA	101.34	102.99	102.75	222.46	223.29	223.05	74.88	75.35	75.54
	VAR	60.43	66.34	72.24	150.02	158.43	163.04	26.33	27.96	28.61
	MLP	46.38	53.98	61.23	80.32	89.07	96.13	19.25	20.38	21.86
	LSTM	44.35	52.80	59.51	75.90	82.13	90.46	18.36	19.24	20.75
	ConvLSTM	40.92	47.20	56.42	66.38	74.56	81.81	15.78	16.31	17.64
	DCRNN	34.64	44.17	52.78	58.03	63.07	72.71	14.38	15.39	16.78
	STGCN	27.51	32.26	38.33	44.26	53.52	65.43	14.50	16.11	19.06
	Graph WaveNet	27.80	30.88	34.00	42.71	47.76	52.90	14.41	15.83	17.32
	STP-TrellisNets+	**26.64**	**29.03**	**31.23**	**40.44**	**42.68**	**48.96**	**11.03**	**11.80**	**12.01**
Hangzhou	Last-Value	84.61	90.34	97.30	176.33	181.18	188.64	24.56	25.07	26.45
	HA	85.60	86.34	85.99	197.10	197.25	197.07	21.69	21.51	21.15
	VAR	83.89	89.07	96.32	200.87	208.23	216.78	26.43	27.46	28.98
	MLP	68.37	78.30	88.48	120.20	127.28	138.35	24.12	25.69	26.05
	LSTM	64.26	71.08	77.09	110.30	115.62	121.47	22.75	23.90	24.93
	ConvLSTM	62.19	69.33	75.24	105.56	112.73	119.39	22.67	23.04	24.98
	DCRNN	58.28	67.90	74.51	67.32	78.93	84.13	17.13	18.86	20.09
	STGCN	36.94	42.19	50.52	56.92	67.55	81.94	9.90	11.30	17.10
	Graph WaveNet	36.18	39.65	42.40	54.83	61.21	66.33	9.82	10.77	11.83
	STP-TrellisNets+	**34.45**	**37.79**	**40.34**	**50.89**	**55.17**	**59.64**	**9.71**	**10.34**	**11.76**

In further analysis on step-wise prediction performance, most methods (i.e., HA, VAR, MLP, LSTM and ConvLSTM) do not explicitly make design for multi-step prediction hence accordingly showing inferior performance under such scenario. DCRNN employs the RNN-based encoder-decoder to generate the multi-step prediction which is accompanied by the severe error accumulation in each future steps thus deteriorates rapidly with the time. Such phenomenon also occurs in STGCN which integrates the previous-step prediction result for future step prediction. In contrast, our STP-TrellisNets+ achieves better prediction results at all predictive steps and degrades slower when steps increasing.

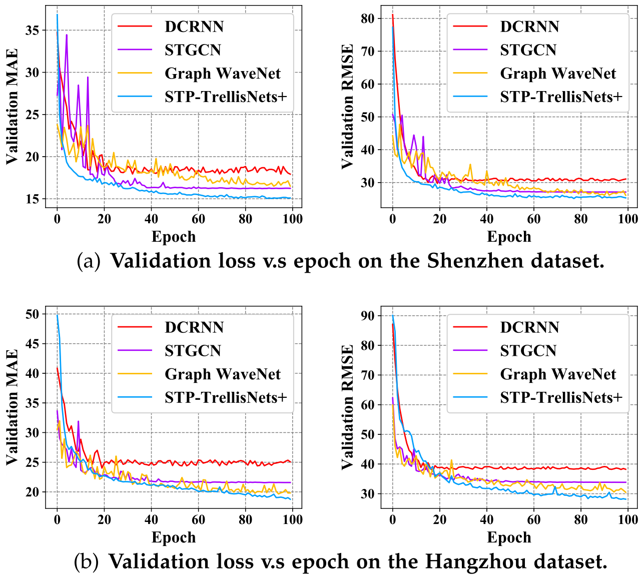
3.5.3 Loss Convergence Comparison Among Spatial-Temporal Approaches

In this section, we discuss the framework convergence during the training procedure of our proposed STP-TrellisNets +, compared to another three state-of-the-art spatial-temporal traffic prediction models (i.e., DCRNN, STGCN and Graph Wavenet). As shown in Fig. 10, there are four curves in each chart representing the validation losses of DCRNN, STGCN, Graph Wavenet and STP-TrellisNets+ in the first 100 training epochs respectively where the curve values denote their overall validation loss over 3 predictive steps under the 30-minute' interval setting.

At the beginning of training, our proposed STP-TrellisNets+ shows a relative higher validation loss than the other three models. This is because STP-TrellisNets+ considers more complex temporal features and spatial correlation dynamics which thus possesses more parameters to learn. With the epoch increasing, the loss curve of STP-TrellisNets + drops with a faster speed and smoother fluctuation and finally converges to much lower MAE and RMSE on two datasets, comparing with the other three models. Such results prove that our STP-TrellisNets+ owns the stable gradient and fast training speed with its TrellisNet architecture and show the superiority of STP-TrellisNets+ on MSP flows prediction with jointly capturing the long- and short-term temporal correlation and dynamic spatial correlation.

3.5.4 Comparison of Models for Capturing Temporal Correlation

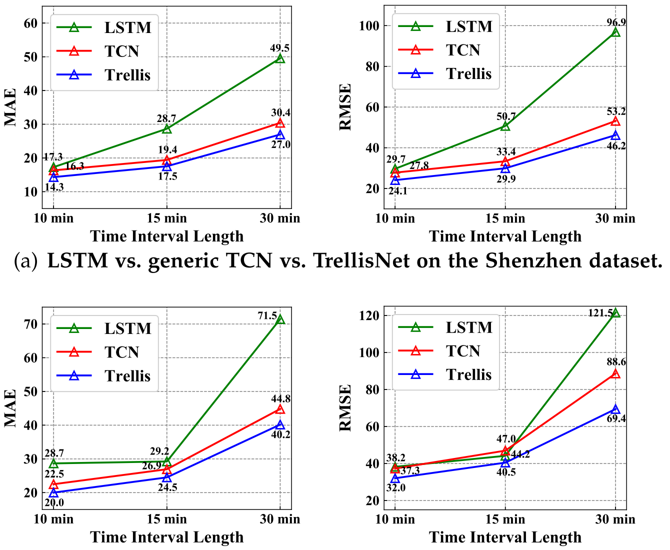
In this part, we evaluate the performance of TrellisNet on capturing the temporal correlation of MSP outflows to empirically verify the motivation of using TrellisNets. We compare three different models' performance for MSP outflow prediction namely 1) LSTM, 2) generic TCN with 4



(a) **Validation loss v.s epoch on the Shenzhen dataset.**

(b) **Validation loss v.s epoch on the Hangzhou dataset.**

Fig. 10. The validation loss versus training epochs comparison among DCRNN, STGCN, Graph WaveNet and STP-TrellisNet+.



(a) **LSTM vs. generic TCN vs. TrellisNet on the Shenzhen dataset.**

(b) **LSTM vs. generic TCN vs. TrellisNet on the Hangzhou dataset.**

Fig. 11. The MSP flow prediction performance comparison among LSTM, generic TCN, and C-TrellisNet.

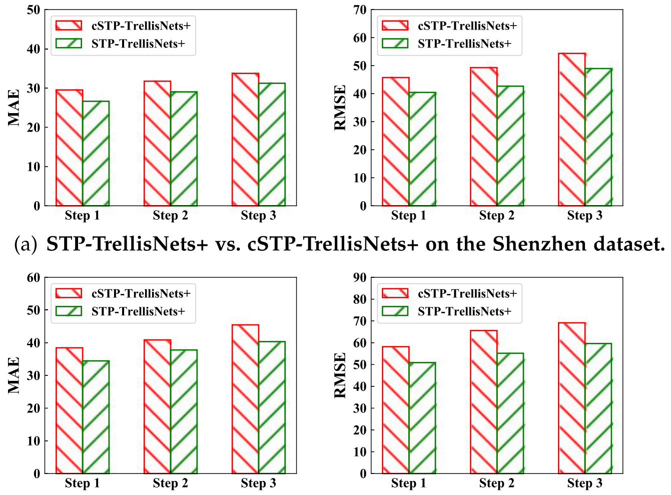
convolution layers and dilation factors $(1,1,2,2)$ in each layer, and 3) the C-TrellisNet introduced in Section 2.2.1 with 4 layers. Because all these three models do not intrinsically possess the multi-step prediction ability, we compare their prediction errors of the MSP outflow in the next time interval $t+1$ using the historical data in the past 24 hours.

Fig. 11 summarizes the comparison results of the above three models (i.e., LSTM, TCN and Trellis) on two datasets, where the lines in each chart denote their prediction errors under different length of next time interval (i.e., 10. 15 and 30 minutes) respectively. From this figure, we can see that C-TrellisNet, which inherits the advantages of TCN and RNN on capturing the temporal correlation, indeed outperforms the other two models.

3.5.5 Necessity of Handling the Discontinuity of MSP Flows

As we have introduced in Section 2.2, we propose the CP-TrellisNetsED architecture to handle the discontinuity between long-term and short-term MSP flows, because the MSP flows between two adjacent days are discontinuous in time. To justify the necessity of such design choice, we compare the performance between STP-TrellisNets+ and a variant of STP-TrellisNets+, namely *cSTP-TrellisNets+*, which omits the discontinuity of the MSP flows between two adjacent days and employs only the P-TrellisNetsED to learn both the short- and long-term temporal correlation of MSP flows and produce the multi-step prediction results. Fig. 12 provides the results of 3 steps' prediction errors of STP-TrellisNets+ and cSTP-TrellisNets+ under 30 minutes' time interval length on two datasets.

We see that compared to the cSTP-TrellisNets+, our STP-TrellisNets+ yields significantly better performance of MSP flow prediction on both the Shenzhen and Hangzhou datasets. Such results indicate the necessity of handling the discontinuity of MSP flows, and verify the effectiveness of our CP-TrellisNetsED on this task.



(a) **STP-TrellisNets+ vs. cSTP-TrellisNets+ on the Shenzhen dataset.**

(b) **STP-TrellisNets+ vs. cSTP-TrellisNets+ on the Hangzhou dataset.**

Fig. 12. The multi-step prediction performance comparison between the STP-TrellisNets+ and cSTP-TrellisNets+.

3.5.6 Comparison With Architectural Variants

In our experiments, to further verify the effectiveness of the proposed architecture of STP-TrellisNets+, we also vary its structure, and evaluate the performances of the following architectural variants.

- *P-TrellisNetsED:* In this variant, we employ only the P-TrellisNetED structure introduced in Section 2.2.2 which yields the multi-step prediction results merely considering the short-term temporal correlation of MSP outflows.
- *CP-TrellisNetsED:* In this variant, we retain the CP-TrellisNetsED structure in STP-TrellisNets+. Clearly, the long- and short-term temporal correlation is captured by this variant without considering the spatial correlation.
- *Single DGCN:*In this variant, apart from CP-TrellisNetsED, we use a single DGCN without a TrellisNets-based Encoder-Decoder connected to its output, which apparently cannot capture the dynamics of spatial correlation. Furthermore, the input adjacency matrix of the DGCN at time interval t is set to be $\mathbf{M}_t$.

Table 6 shows their performances of multi-step MSP flow prediction on the Shenzhen dataset. We can easily observe that STP-TrellisNets+ achieves a much better performance than P-TrellisNetsED and CP-TrellisNetsED, because they cannot capture spatial correlation. Besides, Single DGCN's

TABLE 6
Comparison With Architectural Variants Under 10-Minute and 30-Minute Time Interval Settings

Models	10-minute	30-minute
	Step 1/2/3	Step 1/2/3
P-TrellisNetED	16.22/17.31/18.03	31.26/34.16/36.18
CP-TrellisNetsED	16.01/16.98/17.40	30.69/ 33.72/ 35.44
Single DGCN	15.08/15.90/16.12	28.34/ 31.59/ 32.92
STP-TrellisNets+	**14.60/14.86/14.95**	**26.64/29.03/31.23**

TABLE 7
Comparison With Variants of Multi-Step Prediction Strategies Under 10-Minute and 30-Minute Time Interval Settings

	10-minute	30-minute
Models	Step 1/2/3	Step 1/2/3
Step-by-Step	14.75/18.36/21.02	31.05/36.36/40.64
Plus FC Layer	15.20/16.22/17.07	51.92/38.18/53.61
Multi-TrellisNets	15.16/15.43/15.78	29.60/34.90/40.34
Multi-STP-TrellisNets	15.01/15.39/15.53	28.45/33.11/40.23
STP-TrellisNets+	**14.60/14.86/14.95**	**26.64/29.03/31.23**

mediocre performance demonstrates that the effectiveness and necessity of capturing the dynamics of spatial correlation in the spatial module of STP-TrellisNets+.

3.5.7 Analysis of Multi-Step Prediction Strategy

In our proposed approach, we design a novel TrellisNets-based Encoder-Decoder (TrellisNetsED) structure to generate the multi-step MSP flow prediction results. To further validate the effectiveness of such structure, in this section, we conduct the empirical experiments on different multi-step prediction strategy. We compare four variants obtained by 1) replacing the TrellisNetsED module by a TrellisNet and taking the next-step prediction output of TrellisNet as input for future steps' prediction, named *Step-by-Step*, 2) replacing the TrellisNetsED module by a TrellisNet and adding a fully connected layer (FC layer) following the TrellisNet to transform the single-step output to multiple outputs, named *Plus FC Layer*, 3) replacing the TrellisNetsED module by implementing multiple individual TrellisNets for each step's prediction, named *Multi-TrellisNets*, and 4) revising the model by replacing the TrellisNetsED module with TrellisNet and training multiple models for each step's prediction, named *Multi-STP-TrellisNets* respectively.

Table 7 presents the MAE results over 3 steps of each variant on the Shenzhen dataset. From this table, we can see that the Step-by-Step strategy deteriorates rapidly over multiple steps because it incurs the error from previous predictive step while the Plus FC Layer strategy without considering the additional temporal correlation among each future step achieves the worser performance at all steps than our STP-TrellisNets+. Moreover, the Multi-TrellisNets and Multi-STP-TrellisNets strategies both show inferior performance, because they also overlook the temporal dependency learning among each prediction step. Such results prove the effectiveness of our proposed TrellisNets-based Encoder-Decoder structure for multi-step MSP flows prediction.

3.5.8 Joint Influence of CP-TrellisNetsED' Input Length and the Number of DGCNs in GC-TrellisNetsED

In this set of experiments, we evaluate the MSP flow prediction performance under 30-minute time interval setting on the Hangzhou dataset, with various number of days of long-term periodic data and number of DGCNs in GC-TrellisNetED in order to explore the best parameter setting of STP-TrellisNets+. As shown in Fig. 13, our choice of using 3 DGCNs outperforms the structures with 2 and 4 DGCNs in terms of RMSE. Furthermore, our STP-TrellisNets+ structure with 3 DGCNs achieves the best performance by using the past 3 days' periodic MSP flows as inputs to the P-TrellisNetsED.

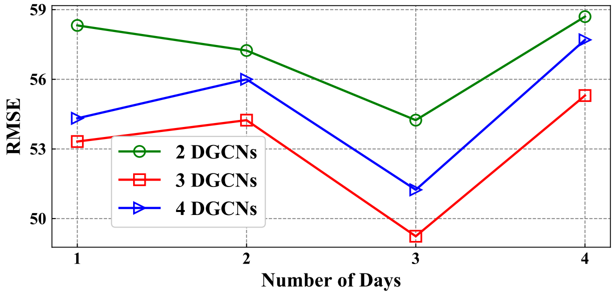


Fig. 13. Joint influence of CP-TrellisNetsED' input length and the number of DGCNs.

4 Related Work

Traffic prediction plays a key role in traffic management, and has received wide attention over the years. Essentially, traffic prediction aims at predicting future values of traffic conditions, such as taxi demand [28], traffic speed [17] and crowd flows [29].

Inspired by the ability of RNN to model the correlation of sequential data, RNNs [14], [15] were adopted for traffic prediction problems. However, since RNN suffers from gradient vanishing, it is usually hard for it to handle long-range input sequences. While several variants of RNN have been proposed to address this problem (e.g., LSTM [16], GRU [17]), they still suffer from high memory requirement and time-consuming training due to its recurrent architecture. Instead of employing RNN, our proposed STP-TrellisNets+ effectively exploits the strengths of a novel TCN variant TrellisNet on handling the long-range input sequence and capturing the non-linearity of MSP flows.

Moreover, several prior works [18], [30], [31], [32] focused on the multi-step prediction task. [18], [30] employed the RNN-based Encoder-Decoder structure to fulfill the multi-step prediction. [31] introduced a GCN-based sequence-to-sequence model for citywide multi-step passenger demand forecasting. [32] proposed the masked temporal attention for multi-step traffic prediction. However, these methods integrated the already predicted value for subsequent time steps' prediction and thus inevitably encountered the error accumulation problem. Compared to the above approaches, we design a novel TrellisNets-based Encoder-Decoder to predict the multi-step MSP flow, unleashing the power of TrellisNet's parallel convolutional framework to ease the error accumulation issue. Specifically, our STP-TrellisNets+ has an architecture with a TrellisNet followed by a TrellisNets-based Encoder-Decoder to jointly consider the long- and short-term temporal correlation of MSP flows.

Recently, several studies further considered spatial correlation by treating the traffic data as a regular image [28], [29], [33], [34], [35] or a graph [13], [15], [16], [17], [18], [19], [20], [31], [32], [36], [37] with geographical attributes (e.g., distance, connectivity) among city regions, road segments, etc., to represent the spatial correlation. Among them, [28] proposed a local CNN method on the image to consider the correlation of taxi demand among spatially nearby city regions. [29] employed a ConvPlus model on the grid structure of city to model the long-range spatial dependency among the crowd flows in different regions. [13] employed a spectral graph convolution on the graph to extract the meaningful spatial features of road traffic speed. In addition, with self-attention mechanism, [32], [36] could capture more global spatial correlations from the graph-structured data. These works either modeled spatial correlation in terms of geographical attributes or assumed static spatial correlation. However, in our metro scenario, spatial correlation is not consistent with geographical attributes and changes in a dynamic way, which make them not fully applicable. Although [38], [39] introduced some dynamics into the graph representation of GCN's input, such graph structure is essentially still based on geographical attributes, which is far from enough to capture the dynamic spatial correlation among MSP flows. Different from them, in this paper, we propose to represent the metro system as a metro directed graph based on a novel transfer flow-based metric, because the transfer flows among metro stations determine their MSP flows. Furthermore, we integrate the diffusion graph convolution with a TrellisNets-based Encoder-Decoder to capture the dynamics of MSP flows' spatial correlation.

Several recent works [2], [3], [4], [5], [6], [40], [41], [42] specifically paid attention to the metro scenario. [40] addressed the crowd flow distribution prediction problem across the entire train network. [41], [42] estimated the route choices of passengers in complex metro networks. Clearly, the problems addressed by these three works are different from the station-level passenger flow prediction problem studied in this paper. [2], [3] employed 1D CNN to capture the spatial correlation of stations in one single metro line, which is not suitable for a practical metro system with multiple metro lines. [4] gave an empirical study to predict MSP flows, but the proposed model can hardly handle the nonlinearity and complexity of metro traffic data. [5] employed a sequence learning model to predict the passenger flows but neglected the long-term periodicity and spatial correlation of MSP flows. Furthermore, [6] modeled the MSP flows as the weakly-dependent tensor data and designed a tensor completion algorithm to address the prediction problem. However, such dependency was characterized by static geographical and contextual attributes, which is suboptimal for representing MSP flows' dynamic spatial correlation. Different from them, our proposed STP-TrellisNets+ focuses on addressing the multi-step MSP flow prediction problem in complex real-world metro systems with dedicatedly designed modules to capture both the long- and short-term temporal correlation and dynamic spatial correlation of MSP flows.

5 Conclusion and Future Work

In this paper, we address the multi-step MSP flow prediction problem from both the spatial and temporal perspectives by proposing a novel deep learning framework STP-TrellisNets +. Specifically, from the *temporal* perspective, we adopt TrellisNet to handle the long-range input historical data of MSP flows, and propose CP-TrellisNetsED consisting of a TrellisNets followed by a TrellisNets-based Encoder-Decoder in serial for capturing the long- and short-term temporal correlation of MSP flows. The necessity and effectiveness of such design choices are validated by our various experimental results. Especially, our comparison, given in Section 3.5.5, between the STP-TrellisNets+ with cSTP-TrellisNets+ which has only one TrellisNets-based Encoder-Decoder as its temporal module indicates that our CP-TrellisNetsED structure is effective in jointly capturing the long-term and short-term temporal correlation. From the *spatial* perspective, we propose the GC-TrellisNetsED framework, which uses a transfer flow-based metric to characterize the spatial correlation among MSP flows, and employs multiple DGCNs along the time steps with their outputs fed to a TrellisNets-based Encoder-Decoder to capture the dynamics of such spatial correlation. Such design choices that essentially integrate TrellisNets with graph convolution are also shown to be effective by our experimental results that compare STP-TrellisNets+ with various architectural variants in Section 3.5.6. Jointly from the *spatial-temporal* perspective, our STP-TrellisNets+ that fuses the outputs of the CP-TrellisNetsED and GC-TrellisNetsED as the final multi-step predicted MSP flows outperforms all the baseline approaches in our extensive experiments with two large-scale real-world AFC datasets.

Although the STP-TrellisNets+ is tailored in various aspects to deal with the special features of MSP flows, we envision that it is, or at least some of its components are, promising to be generalized to other prediction tasks with only minor adaptations. For future work, we would investigate the generalization ability of the STP-TrellisNets+ from the following two aspects: (1) exploiting the CP-TrellisNetsED for multi-step prediction scenarios which also exhibit long- and short-term temporal correlation; (2) augmenting the feature engineering process and graph convolution operations of the GC-TrellisNetED to capture the dynamic spatial correlation of other graph-structured data which also possess such dynamics.

Acknowledgments

The authors like to thank Alibaba Damo Academy for the long-term support and cooperation. This paper is an extended version of [1] which has been published in 29th ACM International Conference on Information and Knowledge Management.

References

[1] J. Ou *et al.*, "STP-TrellisNets: Spatial-temporal parallel TrellisNets for metro station passenger flow prediction," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 1185–1194.

[2] Y. Liu, Z. Liu, and R. Jia, "DeepPF: A deep learning based architecture for metro passenger flow prediction," *Transp. Res. C: Emerg. Technol.*, vol. 101, pp. 18–34, 2019.

[3] Y. Ning *et al.*, "ST-DRN: Deep residual networks for spatio-temporal metro stations crowd flows forecast," in *Proc. Int. Joint Conf. Neural Netw.*, 2018, pp. 1–8.

[4] L. Tang, Y. Zhao, J. Cabrera, J. Ma, and K. L. Tsui, "Forecasting short-term passenger flow: An empirical study on Shenzhen metro," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3613–3622, Oct. 2019.

[5] S. Hao, D.-H. Lee, and D. Zhao, "Sequence to sequence learning with attention mechanism for short-term passenger flow prediction in large-scale metro system," *Transp. Res. C: Emerg. Technol.*, vol. 107, pp. 287–300, 2019.

[6] Z. Li, N. D. Sergin, H. Yan, C. Zhang, and F. Tsung, "Tensor completion for weakly-dependent data on graph for metro passenger flow prediction," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 4804–4810.

[7] S. Shekhar and B. M. Williams, "Adaptive seasonal time series models for forecasting short-term traffic flow," *Transp. Res. Rec.*, vol. 2024, pp. 116–125, 2008.

[8] X. Zhou, Y. Shen, Y. Zhu, and L. Huang, "Predicting multi-step citywide passenger demands using attention-based neural networks," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, 2018, pp. 736–744.

[9] B. Shen, X. Liang, Y. Ouyang, M. Liu, W. Zheng, and K. M. Carley, "StepDeep: A novel spatial-temporal mobility event prediction framework based on deep neural network," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 724–733.

[10] T.-Y. Fu and W.-C. Lee, "DeepIST: Deep image-based spatio-temporal network for travel time estimation," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, 2019, pp. 69–78.

[11] L. Bai, L. Yao, S. S. Kanhere, X. Wang, W. Liu, and Z. Yang, "Spatio-temporal graph convolutional and recurrent networks for citywide passenger demand prediction," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, 2019, pp. 2293–2296.

[12] C. Meng, X. Yi, L. Su, J. Gao, and Y. Zheng, "City-wide traffic volume inference with loop detector data and taxi trajectories," in *Proc. 25th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2017, Art. no. 1.

[13] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 3634–3640.

[14] A. Zonoozi, J.-J. Kim, X.-L. Li, and G. Cong, "Periodic-CRN: A convolutional recurrent model for crowd density prediction with recurring periodic patterns," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 3732–3738.

[15] X. Geng *et al.*, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, Art. no. 449.

[16] Y. Wang, H. Yin, H. Chen, T. Wo, J. Xu, and K. Zheng, "Origin-destination matrix prediction via graph convolution: A new perspective of passenger demand modeling," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1227–1235.

[17] C. Chen *et al.*, "Gated residual recurrent graph neural networks for traffic prediction," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, Art. no. 60.

[18] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learn. Representations*, 2018.

[19] S. Fang, Q. Zhang, G. Meng, S. Xiang, and C. Pan, "GSTNet: Global spatial-temporal network for traffic flow prediction," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 2286–2293.

[20] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 1907–1913.

[21] B. Lu, X. Gan, H. Jin, L. Fu, and H. Zhang, "Spatiotemporal adaptive gated graph convolution network for urban traffic flow forecasting," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 1025–1034.

[22] S. Bai, J. Z. Kolter, and V. Koltun, "Trellis networks for sequence modeling," in *Proc. Int. Conf. Learn. Representations*, 2019.

[23] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 2001–2009.

[24] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[25] E. Zivot and J. Wang, "Vector autoregressive models for multivariate time series," in *Modeling Financial Time Series With S-Plus*, Berlin, Germany: Springer, 2006.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, 1997.

[27] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 802–810.

[28] H. Yao *et al.*, "Deep multi-view spatial-temporal network for taxi demand prediction," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, Art. no. 316.

[29] Z. Lin, J. Feng, Z. Lu, Y. Li, and D. Jin, "Deepstn: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 1020–1027.

[30] X. Zhou, Y. Shen, L. Huang, T. Zang, and Y. Zhu, "Multi-level attention networks for multi-step citywide passenger demands prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 5, pp. 2096–2108, May 2021.

[31] L. Bai *et al.*, "STG2seq: Spatial-temporal graph to sequence model for multi-step passenger demand forecasting," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 1981–1987.

[32] C. Park *et al.*, "ST-GRAT: A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed," in *Proc. 29th ACM Int. Conf. Inform. Knowl. Manag.*, 2020, pp. 1215–1224.

[33] B. Du *et al.*, "Deep irregular convolutional residual LSTM for urban traffic passenger flows prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 972–985, Mar. 2020.

[34] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.

[35] J. Zhang, Y. Zheng, J. Sun, and D. Qi, "Flow prediction in spatio-temporal networks based on multitask deep learning," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 3, pp. 468–478, Mar. 2020.

[36] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, Art. no. 114.

[37] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1720–1730.

[38] Z. Diao, X. Wang, D. Zhang, Y. Liu, K. Xie, and S. He, "Dynamic spatial-temporal graph convolutional neural networks for traffic forecasting," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, Art. no. 110.

[39] M. Wang *et al.*, "Dynamic spatio-temporal graph-based CNNs for traffic prediction," 2018, *arXiv:1812.02019*.

[40] Y. Gong, Z. Li, J. Zhang, W. Liu, Y. Zheng, and C. Kirsch, "Network-wide crowd flow prediction of sydney trains via customized online non-negative matrix factorization," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 1243–1252.

[41] J. Zhao, Q. Qu, F. Zhang, C. Xu, and S. Liu, "Spatio-temporal analysis of passenger travel patterns in massive smart card data," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 3135–3146, Nov. 2017.

[42] G. Sun, Y. Xiong, and Y. Zhu, "How the passengers flow in complex metro networks?," in *Proc. 29th Int. Conf. Sci. Statist. Database Manage.*, 2017, Art. no. 23.

Junjie Ou received the BS degree in information and communication engineering from Beijing Jiaotong University, Beijing, China, in 2018. He is currently working toward the PhD degree in information and communication engineering with Shanghai Jiao Tong University. His research interests include spatial-temporal data mining and time-series forecasting.

Jiahui Sun received the BS degree in electrical and information engineering from Tianjin University, Tianjin, China, in 2019. He is currently working toward the PhD degree in information and communication engineering with Shanghai Jiao Tong University, Shanghai, China. His current research interests include optimization, deep reinforcement learning, and graph representation learning.

Yichen Zhu received the BE degree in computer science and technology from Shanghai Jiao Tong University, Shanghai, China, in 2019. He is currently working toward the PhD degree in computer science and technology with Shanghai Jiao Tong University, Shanghai, China. His current research interests include spatial-temporal traffic prediction.

Haiming Jin (Member, IEEE) received the BS degree from Shanghai Jiao Tong University, Shanghai, China, in 2012, and the PhD degree from the University of Illinois at Urbana–Champaign (UIUC), Urbana, Illinois, in 2017. He is currently a tenure-track assistant professor with the John Hopcroft Center for Computer Science and the Department of Electronic Engineering, Shanghai Jiao Tong University. Before this, he was a post-doctoral research associate with the Coordinated Science Laboratory, UIUC. He is broadly interested in addressing unfolding research challenges in the general areas of urban computing, cyber-physical systems, crowd and social sensing systems, network economics and game theory, reinforcement learning, and mobile pervasive and ubiquitous computing.

Yijuan Liu received the BS degree in electronic engineering from Wuhan University, Wuhan, China, in 2019. She is currently working toward the MS degree in electronic engineering with Shanghai Jiao Tong University, Shanghai, China. Her current research interest is urban computing.

Fan Zhang (Member, IEEE) received the PhD degree in communication and information system from the Huazhong University of Science and Technology, in 2007. He was a post-doctoral fellow with the University of New Mexico and with the University of Nebraska-Lincoln from 2009 to 2011. He is currently a professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. He is also the director of the Shenzhen Institute of Beidou Applied Technology. His research topics include intelligent transportation systems, urban computing, and AI technology.

Jianqiang Huang (Member, IEEE) is a director of Alibaba DAMO Acamedy. He received the second prize of National Science and Technology Progress Award, in 2010. His research interests focus on the artificial intelligence in the city brain project of Alibaba.

Xinbing Wang (Senior Member, IEEE) received the BS degree (with hons.) from the Department of Automation, Shanghai Jiaotong University, Shanghai, China, in 1998, the MS degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2001, and the PhD degree, major in the Department of Electrical and Computer Engineering, minor in the Department of Mathematics, North Carolina State University, Raleigh, in 2006. Currently, he is a professor with the Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai, China. He has been an associate editor for *IEEE/ACM Transactions on Networking* and *IEEE Transactions on Mobile Computing*, and the member of the Technical Program Committees of several conferences including ACM MobiCom 2012, 2018-2019, ACM MobiHoc 2012-2019, IEEE INFOCOM 2009-2020.