Spatio-Temporal AutoEncoder for Traffic Flow Prediction

Mingzhe Liu, *Graduate Student Member, IEEE*, Tongyu Zhu, Junchen Ye, Qingxin Meng, Leilei Sun, *Member, IEEE*, and Bowen Du, *Member, IEEE*

***Abstract*— Forecasting traffic flow is an important task in urban areas, and a large number of methods have been proposed for traffic flow prediction. However, most of the existing methods follow a general technical route to aggregate historical information spatially and temporally. In this paper, we propose a different approach for traffic flow prediction. Our major motivation is to more effectively incorporate various intrinsic patterns in real-world traffic flows, such as fixed spatial distributions, topological correlations, and temporal periodicity. Along this line, we propose a novel autoencoder-based traffic flow prediction method, named Spatio-Temporal AutoEncoder (ST-AE). The core of our method is an autoencoder specially designed to learn the intrinsic patterns from traffic flow data, and encode the current traffic flow information into a low-dimensional representation. The prediction is made by simply projecting the current hidden states to the future hidden states, and then reconstructing the future traffic flows with the trained autoencoder. We have conducted extensive experiments on four real-world data sets. Our method outperforms existing methods in several settings, particularly for long-term traffic flow prediction.**

***Index Terms*— Traffic flow prediction, spatio-temporal autoencoder, hidden state extraction.**

I. Introduction

TRAFFIC flow prediction plays an important role in intelligent transportation systems, which has been widely studied for many years. The most recent efforts utilized spatio-temporal neural networks to forecast the traffic flows of road network, and have achieved reasonable prediction results [1], [2], [3]. These methods usually consist of two fundamental modules: a spatial learning module like convolution neural networks (CNN) or graph neural networks (GNN) to model the spatial interactions of different regions or road segments, and a temporal learning module such as recurrent neural network (RNN) or temporal convolution network (TCN) to capture the evolutionary dynamics of traffic flows.

However, most of the existing methods follow a same technical route, that is, achieving the prediction of traffic flows by aggregating historical traffic flow data spatially and temporally [4], [5], [6]. Recently, it has been realized that the predefined spatial relationship such as geographic distance or topological distance is not sufficient enough to model the spatial dependence of traffic flows of different road segments. Mining the spatial impacts of traffic flows from the traffic flow time series themselves and utilizing the data-driven adjacency matrix to model the spatial interactions of traffic flows could result in better prediction result [3], [7], [8]. Some other research assumes that the spatial dependence of traffic flows varies with time, and therefore uses dynamic adjacency matrix in the spatio-temporal convolution computing to refine the prediction results. Despite the new progresses of spatio-temporal methods in traffic flow forecasting, rare efforts have been made to improve the prediction performance by exploring the intrinsic patterns underlying the large amount of historical traffic flow time series, and then incorporating these intrinsic patterns into the prediction methods.

As shown in Figure 1(a), we respectively calculate the Pearson correlation coefficients (PCC) of various traffic flow sequence of different sensors and different time periods in dataset PEMS03 collected from the Caltrans Performance Measurement System, to demonstrate the topological correlation and temporal periodicity of traffic flow. The PCC between different sensors and between different time periods is not lower than 0.9 and 0.7 respectively, which indicates that we are able to incorporate the intrinsic patterns of available traffic flows in a simpler way, formulated as a lower-dimensional hidden state. Through the projection of hidden state, the future traffic flow time series can be reconstructed, as shown in Figure 1(c).

However, predicting traffic flow through pattern features extraction still faces the following challenges: (1) Different from the previous spatio-temporal deep neural network method that only aggregates historical information, a new spatio-temporal data prediction framework needs to be designed, which can predict future traffic flow from the intrinsic patterns rather than just aggregating the available data. (2) The learned low-dimensional hidden state should contain enough information of intrinsic patterns to support data reconstruction at multiple time steps while avoiding information loss to the greatest extent. (3) The hidden states of traffic flow patterns corresponding to different time periods are time-varying, and the designed framework should have the ability to transform hidden state from current to future.

Manuscript received 21 June 2022; revised 24 November 2022; accepted 13 January 2023. Date of publication 16 February 2023; date of current version 8 May 2023. This work was supported by the National Natural Science Foundation of China under Grant 62272023. The Associate Editor for this article was X. Ma. *(Corresponding author: Tongyu Zhu.)*

Mingzhe Liu, Tongyu Zhu, Junchen Ye, Leilei Sun, and Bowen Du are with the State Key Laboratory of Software Development Environment (SKLSDE), School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: mzliu1997@buaa.edu.cn; zhutongyu@buaa.edu.cn; yjchen@buaa.edu.cn; leileisun@buaa.edu.cn; dubowen@buaa.edu.cn).

Qingxin Meng is with the Department of Entrepreneurship, Marketing and Management Systems, University of Nottingham Ningbo China, Ningbo, Zhejiang 315104, China (e-mail: qingxin.meng@nottingham.edu.cn).

Digital Object Identifier 10.1109/TITS.2023.3243913

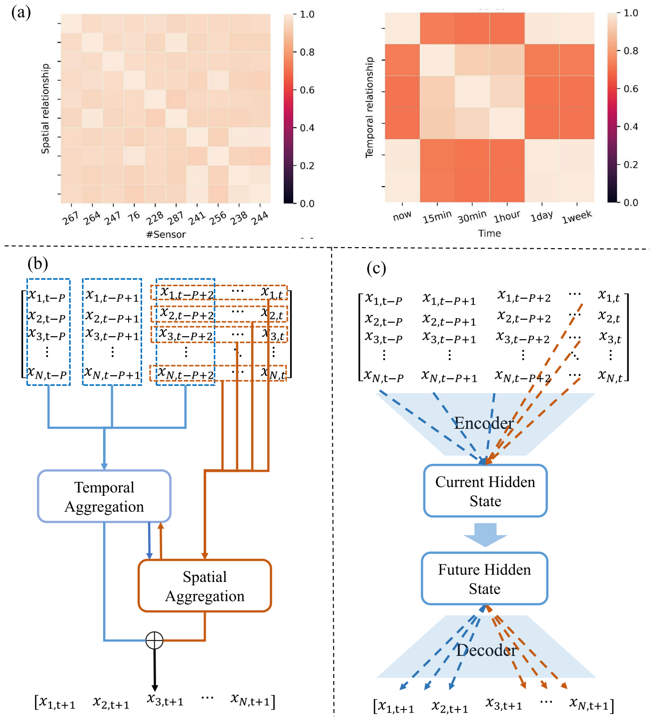

(a)
Spatial relationship
1.0
0.8
0.6
0.4
0.2
0.0
267
264
247
76
228
287
241
256
238
244
#Sensor
Temporal relationship
now
15min
30min
1hour
1day
1week
Time
(b)
$x_{1,t-P}$ $x_{1,t-P+1}$ $x_{1,t-P+2}$ ⋯ $x_{1,t}$
$x_{2,t-P}$ $x_{2,t-P+1}$ $x_{2,t-P+2}$ ⋯ $x_{2,t}$
$x_{3,t-P}$ $x_{3,t-P+1}$ $x_{3,t-P+2}$ ⋯ $x_{3,t}$
$x_{N,t-P}$ $x_{N,t-P+1}$ $x_{N,t-P+2}$ ⋯ $x_{N,t}$
Temporal Aggregation
Spatial Aggregation
$[x_{1,t+1}$ $x_{2,t+1}$ $x_{3,t+1}$ ⋯ $x_{N,t+1}]$
(c)
$x_{1,t-P}$ $x_{1,t-P+1}$ $x_{1,t-P+2}$ ⋯ $x_{1,t}$
$x_{2,t-P}$ $x_{2,t-P+1}$ $x_{2,t-P+2}$ ⋯ $x_{2,t}$
$x_{3,t-P}$ $x_{3,t-P+1}$ $x_{3,t-P+2}$ ⋯ $x_{3,t}$
$x_{N,t-P}$ $x_{N,t-P+1}$ $x_{N,t-P+2}$ ⋯ $x_{N,t}$
Encoder
Current Hidden State
Future Hidden State
Decoder
$[x_{1,t+1}$ $x_{2,t+1}$ $x_{3,t+1}$ ⋯ $x_{N,t+1}]$

Fig. 1. The motivation of our work. (a)The topological correlation and temporal periodicity of traffic flow data is strong enough, indicating the traffic flow has intrinsic patterns. (b)The existing traffic prediction methods usually follows the aggregation of historical spatio-temporal information, while (c)we extract the pattern features and reconstruct the future traffic flow.

To address the above challenges, this paper proposes a novel autoencoder-based framework for traffic flow prediction, named Spatio-Temporal AutoEncoder (ST-AE). Theoretically, the framework can be extended to various spatio-temporal data or multivariate time series data. First, we particularly design an autoencoder for the proposed framework to obtain the low-dimensional hidden states of traffic flow data within a fixed time window. We use temporal convolution and graph convolution to learn historical temporal dependencies and topological correlations respectively in the encoder, and design transposed temporal convolution combined with graph convolution to achieve traffic flow data reconstruction. Second, a method for projecting low-dimensional hidden states of intrinsic patterns from history to future is included in the framework. Finally, we combine the pretrained autoencoder with the projection of hidden state to predict traffic flow for any number of multiple time steps in the future.

In summary, this paper has the following contributions:

- *A novel framework for traffic flow prediction* is proposed, learning the hidden states of intrinsic patterns from available traffic flow data and reconstructing the future traffic flow by projected hidden states.
- *A temporal and graph convolution based autoencoder* for the framework is specially designed. The autoencoder extracts current traffic flow information into a low-dimensional representation based on the correlations and temporal dependencies by dilated convolution and transposed dilated convolution.
- The experiments on real-world datasets indicate that utilizing hidden states with the trained autoencoder allows for more accurate predictions at further time steps.

II. Preliminaries

A. Problem Formalization

Assuming that there are a total of N sensors or detectors included in the studied area, the traffic flow in a certain time window $[t+1, t+T]$ with length T is denoted as $\boldsymbol{X}^{t+1:t+T} = [\boldsymbol{X}^{t+1}, \cdots, \boldsymbol{X}^{t+T}] \in \mathbb{R}^{N \times T \times d}$, where d is the dimension of traffic states. In this paper, $d = 1$ since we only focus on a single traffic state, i.e., the traffic flow. Given N sensors in the studied area, and the traffic flow $x_i^t \in \mathbb{R}$ is the number of passing vehicles counted by sensor i in a fixed time interval denoted by t. Therefore, the traffic flow of the past P time steps can be represented as a matrix $\boldsymbol{X}^{t-P+1:t} = [\boldsymbol{X}^{t-P+1}, \boldsymbol{X}^{t-P+2}, \cdots, \boldsymbol{X}^{t}] \in \mathbb{R}^{N \times P}$.

The aim of traffic flow prediction is to learn a function $\mathcal{F}$, which maps the past traffic flow $\boldsymbol{X}^{t-P+1:t}$ to the future traffic flow $\hat{\boldsymbol{Y}}^{t+T+1:t+T+Q}$ of Q time steps. Formally, the traffic flow prediction problem is defined as:

$$\hat{\boldsymbol{Y}}^{t+T+1:t+T+Q} = \mathcal{F}(\boldsymbol{X}^{t-P+1:t}). \tag{1}$$

The previous research related to traffic prediction usually predicts the traffic state of the most recent multiple time steps directly, i.e., $T = 0$. In our work, in addition to predicting the traffic flow of the recent multiple time step, we also predict the traffic flow of multiple time steps in the further future, i.e. $T \geq Q$.

B. Solutions of Traffic Prediction

Most of the existing traffic prediction methods can be regarded as the aggregation of historical information based on spatio-temporal dependencies. These methods usually include a module $\mathcal{F}_s$ that learns spatial dependencies, implemented by techniques such as CNN or GNN; and a module $\mathcal{F}_t$ that learns temporal dependencies, implemented by techniques such as RNN, TCN or attention-based methods. Let $\boldsymbol{H} = \phi(\boldsymbol{X}) \in \mathbb{R}^{N \times P \times d}$ represent the embeddings or features of N nodes in P time steps, where $\phi(\cdot)$ is the embedding function and d is dimension of features. We formalize the framework adopted by these existing works as follows:

$$\mathcal{F}_s(\boldsymbol{H}) = \sigma\left(\sum\nolimits_{i \in \mathcal{N}(i)} \boldsymbol{W} \cdot \boldsymbol{H}_{i,:,:}\right), \tag{2}$$

$$\mathcal{F}_t(\boldsymbol{H}) = \sigma\left(AGG_{i=t-P+1}^{i=t}(\boldsymbol{H}_{:,i,:})\right), \tag{3}$$

$$\hat{\boldsymbol{Y}} = \mathcal{F}_o\left(\mathcal{F}_s \circ \mathcal{F}_t(\boldsymbol{H})\right), \tag{4}$$

where $\mathcal{N}_{(i)}$ is the neighborhood of node i based on topological relationships, $\boldsymbol{W}$ is the weight parameters, $AGG(\cdot)$ represents the aggregation of temporal information, $\sigma(\cdot)$ is the activation function, $\circ$ represents function composition, and $\mathcal{F}_o$ represents the output function.

However, as mentioned in Section I, we consider the traffic flow data inherently contains features of various patterns. Therefore, the new framework we designed is different from the existing methods, which using pretrained autoencoder to

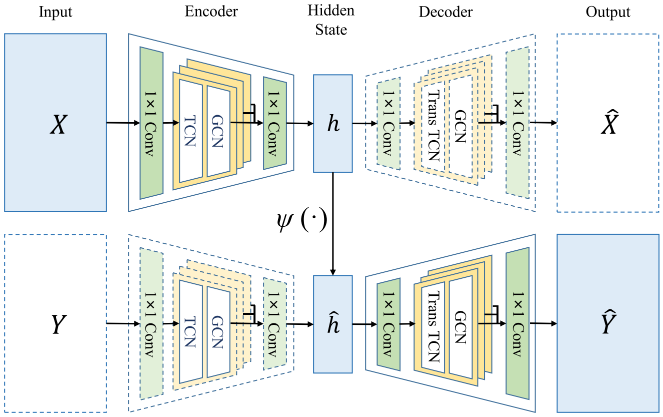


Fig. 2. The framework of the proposed method.

extract the hidden states of intrinsic patterns, projects the hidden states to the future and reconstructs the predicted traffic flow data. Our proposed framework is formalized as follows:

$$\boldsymbol{h} = \delta_{enc}(\boldsymbol{X}), \tag{5}$$

$$\hat{\boldsymbol{X}} = \delta_{dec}(\boldsymbol{h}), \tag{6}$$

$$\hat{\boldsymbol{Y}} = \delta_{dec}(\psi(\boldsymbol{h})), \tag{7}$$

where $\delta_{enc}(\cdot)$ and $\delta_{dec}(\cdot)$ are encoder and decoder in autoencoder, h is the hidden states of intrinsic patterns, and $\hat{\boldsymbol{X}}$ is the reconstructed historical data.

III. Methodologies

A. Framework Overview

Our framework aims to learn a low-dimensional hidden states for traffic flow that contains the intrinsic pattern features of original history input, which can be projected to the future hidden states and used to reconstruct future output for traffic flow prediction. The framework overview is shown as Figure 2. The framework mainly consists of three parts. First, we design an encoder $\delta_{enc}(\cdot)$ based on temporal convolutional network and graph convolutional network to extract the hidden states of traffic flow data. Second, we design a decoder $\delta_{dec}(\cdot)$ with a symmetric structure to the encoder, which utilizes transposed dilated convolution to achieve the reconstruction of the traffic flow time series. The proposed encoder and decoder are combined into an autoencoder, named Spatio-Temporal AutoEncoder (ST-AE). Finally, we design an attention-based hidden states projection function $\psi(\cdot)$ to adaptively project hidden states from history to future, i.e., $\hat{\boldsymbol{h}} = \psi(\boldsymbol{h})$, where $\hat{\boldsymbol{h}}$ is the hidden states corresponding to the predicted traffic flow, and $\boldsymbol{h}$ is the hidden states corresponding to the history traffic flow.

B. Spatio-Temporal Autoencoder

Since traffic flow data has special temporal dependencies and topological correlations, in order to ensure that the extracted hidden states can contain enough pattern features to support traffic flow reconstruction, we particularly design a Spatio-Temporal AutoEncoder for our proposed framework. ST-AE can autonomously learn to extract representative intrinsic pattern features and support the reconstruction of traffic flow data only through low-dimensional hidden states. The structure of the proposed Spatio-Temporal AutoEncoder is shown as Figure 3. The details are introduced as follows.

1) Temporal Dependencies Learning: Since the traffic flow data has strong time dependencies, the autoencoder we design should have the ability to process time series data. In the existing research on time series, the methods used for temporal dependencies learning can be roughly divided into three categories [9]: RNN-based methods (such as LSTM and GRU), CNN-based methods (such as TCN), and attention-based methods (such as Transformer). Among them, RNN-based methods have been utilized in autoencoders by some studies, but their computation that requires sequential processing for each time step limites ability to deal with long sequence data. Attention-based methods fully consider the relationship between time steps of different scales, but this also leads to high computational cost. Therefore, we believe that the dilated causal convolution mechanism in TCN [10] not only has a flexible receptive field, but also occupies lower memories, and is more suitable for building an autoencoder that can process time series. Considering the fitting ability and computational cost, we design the autoencoder based on the thought of TCN [10].

In the encoder, the information at each time step is first embedded by a 1 × 1 convolution. For the entire traffic flow sequence, we use multi-layer 1D dilated convolutions to progressively learn the time dependencies and compress the length of time steps. By flexibly setting the dilation factor in each layer and number of layers, the model can learn the information of all historical time steps while using as little memory as possible. For a time series $\boldsymbol{x}$, the dilated convolution operation using a filter $\boldsymbol{f}$ with the kernel size of $1 \times k$ and the dilated factor of d is formally defined as:

$$z(s) = \boldsymbol{x} \star_d \boldsymbol{f}(s) = \sum_{i=0}^{K-1} \boldsymbol{f}(i)\boldsymbol{x}(s - d \cdot i), \tag{8}$$

where $z(s)$ represents the time step s of the compressed sequence $\boldsymbol{z}$. The encoder can compress the original traffic flow sequence of length T into a hidden state of length 1 by using 1D dilated convolution multiple times, and output the intrinsic pattern's hidden state $\boldsymbol{h}$ through another 1×1 convolution.

In the decoder, we innovatively design the structure using 1D transposed dilated convolution. The structure of the decoder is symmetrical with that of the encoder, which takes the hidden state output by the encoder as input, and utilizes multiple layers of transposed dilated convolution to gradually expand the information of a single time step to the same time step length as the original input. When the settings of the dilation factor and the number of layers of the decoder is consistent with those of the encoder, as long as the encoder samples the information of all time steps in the original data, the decoder can reconstruct all time steps without omission. Although from the point of view of information theory, the operation of convolution is irreversible, the transposed convolution operation can restore the original information that

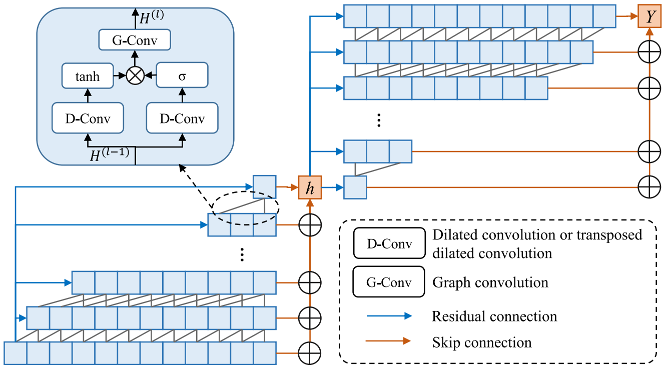


Fig. 3. Structure of the spatio-temporal autoencoder.

preserves the positional relationship through the training of parameters, which is necessary for sequential data. For hidden state $\boldsymbol{z}$, the transposed dilated convolution operation using a filter $\boldsymbol{f}$ with kernel size of $1 \times k$ and dilated factor of d is formally defined as:

$$\boldsymbol{x}(s + d \cdot i) = \boldsymbol{z} \star_d^{-1} \boldsymbol{f}(s) = \sum_{s=0}^{T-1} \sum_{i=0}^{K-1} \boldsymbol{f}(i)\boldsymbol{z}(s). \tag{9}$$

On the basis of dilated convolution and transposed dilated convolution, we also add gated mechanisms in both the encoder and decoder to control the temporal convolution to transmit effective information to their next layer. We use 2 different filters to compress or expand the sequential data of the current layer, utilize the tangent hyperbolic function and the sigmoid function as the activation function respectively, and perform element-wise product of the activated results. The setting of the gated mechanism in the encoder and decoder is exactly the same, which is formally defined as:

$$\boldsymbol{Z}_{enc}^{(l)} = tanh(\boldsymbol{H}_{enc}^{(l-1)} \star_d \boldsymbol{\Theta}_{enc,1}) \odot \sigma(\boldsymbol{H}_{enc}^{(l-1)} \star_d \boldsymbol{\Theta}_{enc,2}), \tag{10}$$

$$\boldsymbol{Z}_{dec}^{(l)} = tanh(\boldsymbol{H}_{dec}^{(l-1)} \star_d^{-1} \boldsymbol{\Theta}_{dec,1}) \odot \sigma(\boldsymbol{H}_{dec}^{(l-1)} \star_d^{-1} \boldsymbol{\Theta}_{dec,2}), \tag{11}$$

where $\boldsymbol{\Theta}_{enc}$ and $\boldsymbol{\Theta}_{dec}$ are the parameters of dilated convolution and transposed dilated convolution respectively. $tanh(\cdot)$ and $\sigma(\cdot)$ are tangent hyperbolic function and sigmoid function, and $\odot$ is element-wise product. $\boldsymbol{H}_{enc}^{(l-1)}$ and $\boldsymbol{H}_{dec}^{(l-1)}$ are the hidden state output by last layer. $\boldsymbol{Z}_{enc}^{(l)}$ and $\boldsymbol{Z}_{dec}^{(l)}$ are the output of temporal dependencies learning in current layer.

2) Correlations Learning: The proposed autoencoder based on dilated convolution and transposed dilated convolution can already learn temporal dependencies, but spatial dependencies learning also have always been the focus of research in traffic prediction. Although in Section I, we believe that the time dimension in the traffic flow data already contains sufficient information, the correlation between different traffic flow series also exists objectively, and capturing these correlations may help improve the accuracy of traffic flow prediction.

As mentioned in Section I, traffic flow data has sufficient topological correlation only by relying on sequence data itself without introducing external knowledge. The topological correlation (refer to correlation for short) includes self-correlation and cross-correlation. Therefore, inspired by the work of [3], we employ a data-driven approach to adaptively learn correlations of different traffic flow sequences.

We add correlation learning module to both the encoder and decoder. The added modules have exactly the same structure but do not share parameters in encoder and decoder because the compression of past time steps and the expansion of future time steps have different correlation when prediction. We model the correlation of traffic flow as a graph structure, each sensor corresponds to a node in the graph, then the adjacency matrix of the graph is represented as $\boldsymbol{A} \in \mathbb{R}^{N \times N}$. For the self-correlation, we learn by adding an identity matrix. For the cross-correlation, we build two learnable embeddings $\boldsymbol{E}_1, \boldsymbol{E}_2 \in \mathbb{R}^{N \times c}$ for each node, where N is the number of sensors and c is the dimension of embeddings. The embeddings are multiplied to form the cross-correlation between different traffic flow sequences, which are adaptively updated during training. Since the parameters of the correlation learning module in the encoder and decoder are the same, we omit the subscripts of *enc* and *dec* for simplicity. The adaptive correlation adjacency matrix is formally defined as:

$$\boldsymbol{A} = \boldsymbol{I}_N + softmax(ReLU(\boldsymbol{E}_1 \boldsymbol{E}_2^T)), \tag{12}$$

where $\boldsymbol{I}_N \in \mathbb{R}^{N \times N}$ is the identity matrix. We utilize graph convolution to aggregate information of correlated traffic flows. The used graph convolution can aggregate multi-hop information, taking the adaptive adjacency matrix $\boldsymbol{A}$ and the compressed or expanded temporal hidden state of each layer $\boldsymbol{Z}^{(l)}$ as input, and the graph convolution employed is formally defined as:

$$\widetilde{\boldsymbol{H}}^{(l)} = \sum_{k=0}^{K} (\boldsymbol{A})^k \boldsymbol{Z}^{(l)} \boldsymbol{W}_k, \tag{13}$$

where $\boldsymbol{W}_k$ is the weighted parameters, $\boldsymbol{Z}^{(l)}$ is the output of temporal dependencies learning in current layer, and $\widetilde{\boldsymbol{H}}^{(l)}$ is the output of correlations learning in current layer. By adding correlation learning modules in both encoder and decoder, the proposed autoencoder is able to transfer the information between correlated sequence data while capturing temporal dependencies, which improves both feature extraction and data reconstruction.

3) Residual and Skip Connection: In order to ensure the stability of model training and sufficient information transfer in each layer, we add residual connections and skip connections to the encoder and decoder on the basis of temporal dependencies learning and correlations learning. Their structures in the encoder and decoder are completely symmetrical and have the same parameter size.

Since the time step lengths corresponding to the hidden states of each layer are different, based on the thought of residual connection, after the dilated convolution and graph convolution in current layer, the current input $\boldsymbol{H}^{(l-1)}$ is also passed to the next layer after simple transformation $f_{res}(\cdot)$, so that the model can get more information from the original data. Since the residual connections in the encoder and decoder are the same, we omit the subscripts of *enc* and *dec* for simplicity, which can be formally defined as:

$$\boldsymbol{H}^{(l)} = \widetilde{\boldsymbol{H}}^{(l)} + f_{res}^{(l)}(\boldsymbol{H}^{(l-1)}). \tag{14}$$

In order to make the proposed autoencoder better deal with the tasks of feature extraction and data reconstruction to achieve more accurate traffic flow prediction, we use skip connections to aggregate the hidden states of all layers in the encoder after a simple transformation $f_{skip}(\cdot)$ to form the hidden states of patterns $\boldsymbol{h}$. Similarly, in the decoder, the hidden states of all layers are simply transformed and aggregated to form reconstructed data $\hat{\boldsymbol{X}}$ or $\hat{\boldsymbol{Y}}$. The skip connection is formally defined as:

$$\boldsymbol{h} = \sum_{l=0}^{L} f_{skip}^{(l)}(\boldsymbol{H}_{enc}^{(l)}), \tag{15}$$

$$\hat{\boldsymbol{Y}} = \sum_{l=0}^{L} f_{skip}^{(l)}(\boldsymbol{H}_{dec}^{(l)}), \tag{16}$$

where L is the number of convolution layers. We find through experiments that the model has the best predictive performance when f_{res} and f_{skip} are linear transformations. Through all the above components, we build a complete spatio-temporal autoencoder that fully considers the characteristics of traffic flow data, and the learned hidden state of intrinsic pattern features can be applied to a variety of downstream tasks.

C. Projection of Hidden States

To make predictions about future traffic flow, we need to establish a time-varying relationship between past and future traffic flow. Different from previous studies, since we can already extract lower-dimensional traffic flow pattern features through the ST-AE, we only need to project the intrinsic pattern's hidden state from current to future. As shown in Figure 2, $\boldsymbol{h} \in \mathbb{R}^{N' \times T'}$ is the hidden state of intrinsic pattern extracted from the current traffic flow $\boldsymbol{X}$, where N' and T' are the spatial and temporal dimensions of hidden state, respectively. Intuitively, the future traffic flow also has a corresponding hidden state $\hat{\boldsymbol{h}}$, from which we can reconstruct the future traffic flow. Therefore, the projection layer of the hidden state from the current to the future can be regarded as a time-aware nonlinear mapping function.

We believe that the intrinsic pattern's hidden state contains key implicit features of the original data, while still having some spatio-temporal properties. For example, the hidden state of time series data is still periodic, which is verified in Section IV-E4. Since the learned hidden state still contains the corresponding non-explicit features in the temporal dimension T' and the spatial dimension N', we adapt the attention mechanism to transform the hidden state. We utilize the multi-head self-attention mechanism in Transformer [11] to learn the time-varying relationship from two directions, that is, using a temporal attention function $f_{att}^{tem}(\cdot)$ and a spatial attention function $f_{att}^{spa}(\cdot)$ to map the hidden state from current to future, respectively. In order to fuse the learning results of the two multi-head attentions, we use the gated mechanism to control the transfer of information between the current and future hidden state. The projection function of the hidden state is formally defined as:

$$\phi = \sigma\left(\boldsymbol{W}_1 f_{att}^{tem}(\boldsymbol{h}) + \boldsymbol{b}_1 + \boldsymbol{W}_2 f_{att}^{spa}(\boldsymbol{h}) + \boldsymbol{b}_2\right), \tag{17}$$

$$\hat{\boldsymbol{h}} = \phi \odot f_{att}^{tem}(\boldsymbol{h}) + (1 - \phi) \odot f_{att}^{spa}(\boldsymbol{h}) + \boldsymbol{h}, \tag{18}$$

where $\boldsymbol{W}_1$, $\boldsymbol{W}_2$, $\boldsymbol{b}_1$, $\boldsymbol{b}_2$ are parameters of weight and bias, $f_{att}^{tem}(\cdot)$ and $f_{att}^{spa}(\cdot)$ are the multi-head attention operation functions, $\sigma(\cdot)$ represents the sigmoid activation function, and ϕ is the gate. The projection function maps the current hidden states to the future hidden states by studying their intrinsic correlated patterns. Through the above approach, our model has the ability to predict the traffic flow at further time steps just by learning the projected hidden state.

D. Training Strategy

In order to maximize the performance of the model and improve the prediction accuracy, we first use the Spatio-Temporal AutoEncoder to pretrain for data reconstruction on the training set, that is, the input is the original traffic flow data $\boldsymbol{X}$, and the output is the reconstructed traffic flow data $\hat{X}$, as shown in Equations 5 and 6. We then utilize the pretrained autoencoder for traffic flow prediction. We load the parameters of the encoder and decoder of the pretrained model, and add the projection function $\psi(\cdot)$ between the encoder and decoder, using the combined model for supervised learning. The input is the original traffic flow data $\boldsymbol{X}$, and the output is the traffic flow to be predicted $\hat{\boldsymbol{Y}}$, i.e., $\hat{\boldsymbol{Y}} = \delta_{dec}(\psi(\delta_{enc}(\boldsymbol{X})))$. Since the encoder and decoder themselves also have the ability to learn temporal dependencies, we do not fix the parameters of the encoder and decoder, allowing them to be fine-tuned during training. And since the spatio-temporal autoencoder and the projection function have different fitting abilities, we set different learning rates for them. The loss functions for pre-training and prediction are formally defined as:

$$L_{pretrain}(\boldsymbol{X}, \hat{\boldsymbol{X}}) = \sum_{i=t-P+1}^{t} |\boldsymbol{X}_{:,i} - \hat{\boldsymbol{X}}_{:,i}|, \tag{19}$$

$$L_{predict}(\boldsymbol{Y}, \hat{\boldsymbol{Y}}) = \sum_{i=t+T+1}^{t+T+Q} |\boldsymbol{Y}_{:,i} - \hat{\boldsymbol{Y}}_{:,i}|, \tag{20}$$

where $|\cdot|$ is the L$_1$-loss of the vectors, and $\boldsymbol{X}_{:,i}$ represents the traffic flow of all sensors at time step i.

IV. Experiments

A. Datasets

Our experiments use four datasets from the Caltrans Performance Measurement System [12]: PEMS03, PEMS04, PEMS07, and PEMS08. The relevant information of these data sets is shown in Table I. We only use the traffic flow data in data sets for model training and prediction. All data sets are sampled every 5 minutes, and the data have been z-score normalized during training and testing. The source codes of ST-AE are publicly available from https://github.com/LMZZML/ST-AE.

TABLE I
DESCRIPTION OF DATASETS

Dataset	#Sensors	Start time	Granularity	#Time step
PEMS03	358	2012/5/1	5min	26208
PEMS04	307	2017/7/1	5min	16992
PEMS07	883	2017/5/1	5min	28224
PEMS08	170	2012/3/1	5min	17856

B. Baselines

To validate the effectiveness of our method, we compare our method with the following baseline methods.

- XGBoost: A classic gradient boosting tree-based method widely used in regression tasks.
- FC-LSTM: Long Short-Term Memory Network with fully connected layers to expand information of original data.
- DCRNN [1]: Diffusion Convolution Recurrent Neural Network combined bi-directional random walk on distance-based graph with GRU in an encoder-decoder manner.
- STGCN [2]: Spatio-temporal Graph Convolutional Networks utilize graph convolution and casual convolution to learn spatial and temporal dependencies.
- Graph WaveNet [3]: A framework combines adaptive adjacency matrix into graph convolution with 1D dilated convolution.
- STSGCN [5]: Spatial-Temporal Synchronous Graph Convolutional Network utilizes localized spatio-temporal subgraph module to model localized correlations independently.
- AGCRN [4]: Adaptive Graph Convolutional Recurrent Network decompose the adjacency matrix and parameters of the graph convolution layer.
- STFGNN [6]: Spatial-Temporal Fusion Graph Neural Network constructs the temporal graph learned based on similarities between time series.

C. Evaluation Metrics

We utilize three evaluation metrics, MAE, MAPE and RMSE, to evaluate the effectiveness of each method. They measure the difference between the ground truth and the predicted value. The three evaluation metrics are formally expressed as follows.

$$MAE = \frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\left|\hat{y}_t^i - y_t^i\right|, \tag{21}$$

$$MAPE = \frac{100\%}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\left|\frac{\hat{y}_t^i - y_t^i}{y_t^i}\right|, \tag{22}$$

$$RMSE = \sqrt{\frac{1}{NT}\sum_{i=1}^{N}\sum_{t=1}^{T}\left(\hat{y}_t^i - y_t^i\right)^2}, \tag{23}$$

where $\hat{y}_i^t$ is the element in the predicted result $\hat{\boldsymbol{Y}}$, and y_i^t is the element in the ground truth $\boldsymbol{Y}$.

D. Experiment Setting

In the short-term prediction, we use the past 12 time steps to predict the future 12 time steps, i.e., use the past one hour to predict the future one hour; in the long-term prediction, we use the past 12 time steps to predict the future 13 to 24 and 25 to 36 time steps respectively, i.e. use the past hour to predict the second and third hour in the future. For the four data sets, we divide the first 60% samples as the training set, the next 20% as the validation set, and the rest 20% as the test set.

For our proposed method, we set the dimension of the hidden state to $N \times 2$, which is one-sixth the size of the original sampled data, where N is the number of sensors. Both dilated convolution and transposed dilated convolution are 8 layers, and the dilation factor of each layer is alternated between 1 and 2. The channels of the 1×1 convolution are set to 32. The number of hops of graph convolution is set to 2, and the dimension of node embeddings in the adjacency matrix is set to 10. The number of heads of the multi-head attention mechanism is 8. When training the prediction model, we set different learning rates for the projection function and the pretrained autoencoder. The initial learning rate of the autoencoder is 0.0001, and the initial learning rate of the projection function is 0.001. Both learning rates can be decayed according to the performance of the validation set.

E. Results

1) Short-Term Prediction: We first predict traffic flow on the conventional setting employed in recent traffic flow prediction research, using the traffic flow of the past hour to predict the traffic flow of the next hour. The predicted results are shown in Table II. From the results, we can see that our method outperforms the baseline methods on most of the metrics on the four data sets, and the improvement is most obvious on PEMS03 and PEMS08. This indicates that our proposed novel framework is effective when dealing with the traffic flow prediction problem and outperforms most existing methods on the conventional setting.

Among all the baseline methods, XGBoost, as a machine learning regression method, is seriously affected by the length of the time step, and the error is extremely large at larger time steps. Among all deep learning methods, the performance of DCRNN and STGCN are not well due to only capturing the correlation based on geographic information and simple temporal aggregation. Graph WaveNet and AGCRN, as adaptive graph learning methods, perform well on various data sets, but because they only aggregate historical spatio-temporal dependencies without further analysis of the aggregated information, there is a little gap of accuracy between them and our proposed method. STSGCN and STFGNN, as the latest methods to add temporal connectivity in spatial graph modeling, also perform poorly because their essence is still the spatio-temporal aggregation of observed information.

2) Long-Term Prediction: In order to demonstrate that the hidden state we extract contains implicit features of traffic flow patterns, and that our proposed framework has the ability to reconstruct future traffic flow sequences from the hidden state, we predict traffic flows over longer periods of

TABLE II

THE RESULT OF TRAFFIC FLOW PREDICTION FOR NEXT 1 HOUR

Dataset	Method	15min			30min			60min		
		MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
PEMS03	XGBoost	17.6686	0.1781	25.7280	20.9996	0.2147	30.0847	28.4236	0.3135	39.4211
	FC-LSTM	18.6827	0.1955	32.9931	18.8611	0.1964	33.2397	19.5583	0.2029	34.1788
	DCRNN	15.2103	0.1743	25.1654	16.8606	0.1901	27.9368	19.9616	0.2199	32.6642
	STGCN	14.7753	0.1560	24.1301	16.1636	0.1641	26.6496	18.7114	0.1790	30.8042
	Graph WaveNet	13.9818	0.1539	23.4727	15.2864	0.1637	25.7226	17.7525	0.1783	30.3526
	STSGCN	15.5227	0.1522	25.2545	17.2771	0.1666	28.4852	20.4098	0.1904	33.7455
	AGCRN	14.4129	0.1475	25.9915	15.6644	0.1638	27.9519	18.1047	0.1886	31.5794
	STFGNN	15.1854	0.1501	25.3023	16.6894	0.1612	27.7500	19.3751	0.1829	32.0534
	ST-AE	**13.9531**	**0.1399**	**23.5400**	**15.0188**	**0.1479**	**25.3253**	**17.2946**	**0.1692**	**28.5030**
PEMS04	XGBoost	22.8156	0.1571	33.7150	26.7803	0.1895	39.0121	35.6786	0.2738	50.3715
	FC-LSTM	22.7928	0.1855	37.6994	22.8703	0.1855	37.8127	23.1842	0.1852	38.1755
	DCRNN	19.6533	0.1517	31.2972	21.8075	0.1683	34.1114	26.2000	0.1843	39.9192
	STGCN	19.7036	0.1483	31.1503	20.7066	0.1528	32.8623	22.1402	0.1692	34.9950
	Graph WaveNet	18.7530	0.1414	29.8036	20.4062	0.1585	31.9182	23.2181	0.1943	35.4140
	STSGCN	19.6867	0.1307	31.0517	21.1803	0.1391	33.3163	24.6133	0.1615	38.1348
	AGCRN	19.1374	0.1279	30.5962	20.3953	0.1372	32.4284	23.1057	0.1582	36.2185
	STFGNN	18.7226	**0.1232**	29.9832	19.7557	**0.1285**	31.6970	21.6484	**0.1401**	34.3420
	ST-AE	**18.2617**	0.1337	**29.1936**	**19.5230**	0.1453	**31.1133**	**21.2463**	0.1753	**33.4214**
PEMS07	XGBoost	23.8386	0.1069	36.6069	28.2303	0.1307	42.4260	38.1904	0.1931	55.0515
	FC-LSTM	31.2224	0.1420	57.6790	31.3721	0.1431	57.7680	31.6011	0.1452	57.6559
	DCRNN	21.1936	0.1032	33.1574	23.3022	0.1186	36.5063	27.3041	0.1329	42.2762
	STGCN	20.8498	0.0968	32.4956	22.5148	0.1022	35.1632	25.2718	0.1110	39.2796
	Graph WaveNet	19.7367	0.0922	31.6912	21.7290	0.0987	34.7506	25.2234	0.1192	39.3465
	STSGCN	21.3875	0.0896	33.9975	24.0095	0.1004	38.5270	28.9879	0.1227	46.3207
	AGCRN	20.2253	0.0856	32.4852	21.9126	0.0929	35.1626	25.0545	0.1085	39.5443
	STFGNN	20.4564	0.0858	32.8885	22.2117	0.0926	36.0882	25.2017	**0.1054**	40.7608
	ST-AE	**19.5953**	**0.0840**	**31.4706**	**21.4089**	**0.0923**	**34.3089**	**24.6022**	0.1099	**38.5588**
PEMS08	XGBoost	16.9765	0.1102	26.1136	19.9544	0.1319	30.3752	26.6602	0.1882	39.2848
	FC-LSTM	21.6879	0.1387	35.2213	21.9588	0.1399	35.3809	23.0241	0.1452	37.2381
	DCRNN	15.0498	0.1015	23.3138	16.4737	0.1090	25.8150	18.8483	0.1234	29.6323
	STGCN	15.1619	0.1056	23.2692	16.4038	0.1110	25.3736	18.6704	0.1208	28.6799
	Graph WaveNet	14.9218	0.1021	23.3421	16.6878	0.1074	26.1012	19.2889	0.1283	29.9003
	STSGCN	15.7269	0.1011	24.2341	17.0029	0.1082	26.3702	19.6990	0.1239	30.3372
	AGCRN	15.1075	0.0964	23.6536	16.0935	0.1026	25.4187	18.1646	0.1174	28.6984
	STFGNN	15.4096	0.0992	23.9279	16.7667	0.1067	26.3075	19.4293	0.1223	30.2502
	ST-AE	**14.1711**	**0.0939**	**22.3985**	**15.2500**	**0.0989**	**24.3530**	**17.2425**	**0.1147**	**27.3488**

time. We perform the long-term prediction on PEMS03 and PEMS08. We predict the traffic flow for the next 3 hours separately, and the average error of each hour is shown in Table III. Since machine learning methods perform extremely poorly on long-term prediction, XGBoost is no longer listed in the table.

From the results in Table III, it can be found that our proposed framework achieves complete advantages in long-term prediction. The improvement of ST-AE is more obvious on longer time step. The main reason is that the information aggregated by the existing methods is temporal locality, and their predictions for longer time steps are only based on local information, but they do not model future temporal dependencies in method design. Our proposed method models the positional information of future time steps based on the hidden state, so the improvement is more obvious.

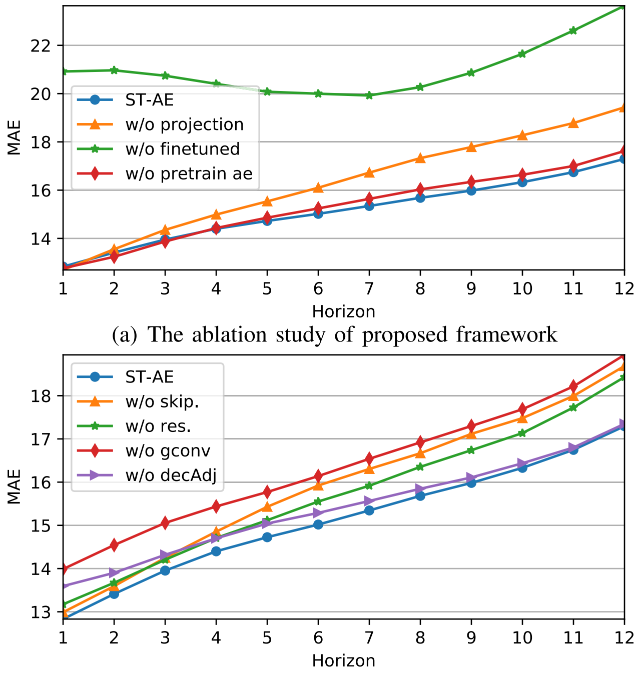
3) Ablation Study: To demonstrate the effectiveness of various components and steps in our proposed method, we perform ablation study on the PEMS03 dataset. The prediction is still using the past 12 time steps to predict the future 12 time steps. The ablation study mainly focuses on two parts, the overall framework and the designed autoencoder. For the overall framework, we name the models without different components as follows.

- w/o projection: Remove the projection function $\psi(\cdot)$ from the framework, i.e. the hidden state output by the encoder is sent directly to the decoder.
- w/o finetuned: Fix the parameters of the pretrained autoencoder so that it is not updated during training.
- w/o pretrain ae: The autoencoder no longer performs pretraining, and directly uses all random parameters to train the prediction model.

The result of the overall framework's ablation study is shown in Fig. 4(a). From the result, it can be found that: first, the projection function is crucial for the improvement of prediction accuracy, especially for long-term prediction.

TABLE III
THE RESULT OF LONG-TERM TRAFFIC FLOW PREDICTION FOR NEXT 3 HOUR

Dataset	Method	0-1h			1-2h			2-3h		
		MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE
PEMS03	FC-LSTM	18.9417	0.1974	33.3371	21.8149	37.6048	0.2120	23.2701	0.2416	39.7176
	DCRNN	16.9845	0.1913	28.0533	22.8956	0.2446	36.1356	27.2772	0.3163	44.7585
	STGCN	16.7501	0.1753	27.6864	21.3696	0.2146	35.0350	28.5032	0.2664	45.1864
	Graph WaveNet	15.3456	0.1608	25.7397	19.4040	0.2102	31.7850	21.7462	0.2156	35.9741
	STSGCN	17.3840	0.1665	28.5655	23.8939	0.2299	39.3531	28.7939	0.2715	48.0824
	AGCRN	15.7947	0.1674	28.2003	19.8531	0.2076	33.1676	22.8556	0.2275	37.2437
	STFGNN	16.7568	0.1616	27.8334	21.9309	0.2017	37.2473	25.5852	0.2358	44.2922
	ST-AE	**15.1436**	**0.1496**	**25.3601**	**19.1615**	**0.1834**	**31.5115**	**21.4224**	**0.2071**	**34.4938**
PEMS08	FC-LSTM	22.0783	0.1406	35.7969	23.5853	0.1456	39.7280	25.0876	0.1579	41.6078
	DCRNN	16.5109	0.1096	25.8156	21.1396	0.1509	32.8317	24.6788	0.1786	37.8446
	STGCN	16.7453	0.1124	25.7742	21.9117	0.1406	32.8059	23.8873	0.1691	35.6258
	Graph WaveNet	16.6479	0.1099	25.9212	19.4647	0.1333	30.5715	22.1563	0.1679	34.1785
	STSGCN	17.1846	0.1095	26.5375	22.0468	0.1392	34.2782	25.7880	0.1649	40.2783
	AGCRN	16.2103	0.1036	25.6008	21.6530	0.1369	33.5086	24.9457	0.1607	38.3659
	STFGNN	16.8973	0.1076	26.3789	22.0504	0.1380	34.5546	24.5423	0.1594	38.1390
	ST-AE	**15.3250**	**0.1008**	**24.3246**	**18.9814**	**0.1282**	**29.6424**	**20.4273**	**0.1429**	**32.3049**



(a) The ablation study of proposed framework

(b) The ablation study of the autoencoder

Fig. 4. Ablation study on PEMS03.

After removing the projection function, the prediction error grows more and more with time steps. This indicates that the projection function does indeed play the role of mapping the past hidden state to the future hidden state as much as possible. Second, the prediction error is significantly increased after fixing the parameters of the ST-AE, which indicates that the ST-AE has the ability to model temporal dependencies and is affected by temporal changes. Finally, the model without pretraining has slightly higher prediction errors than the model with pretraining, indicating the effectiveness of the training strategy used.

For the designed autoencoder, we name the models without different components as follows.

- w/o skip: ST-AE without skip connections. The hidden state is the output of the encoder's last layer, and the predicted traffic flow is the output of the decoder's last layer.
- w/o res: ST-AE without residual connections and only backpropagate convolutional information.
- w/o gconv: ST-AE without the graph convolution, i.e. removing correlation learning module from encoder and decoder.
- w/o: decAdj: The correlation learning modules of the encoder and decoder use the same adjacency matrix parameters.

The result of the designed autoencoder's ablation study is shown in Fig. 4(b). It can be found from the results that: first, removing the correlation learning has the largest impact on the prediction accuracy, which indicates that there are indeed correlations between the traffic flow sequences of different sensors. Second, both residual connections and skip connections are effective for improving model accuracy, which ensure sufficient and stable transfer of information, especially at distant time steps. Finally, the encoder and decoder perform better when using different correlation adjacency matrices, indicating that the past and future correlations are time-varying.

4) Case Study: To better demonstrate the difference between our framework and existing traffic flow prediction methods, we conduct case study on PEMS03, as shown in Figure 5. Due to space limitations, we select one sensor in PEMS03, and plot the ground truth (Figure 5(a)) and predicted value (Figure 5(d)) of its next time step, as well as the value of each dimension of the historical hidden state h (Figure 5(b)) and the future hidden state $\hat{h}$ (Figure 5(c)) at the corresponding time step. There are two dimensions of the hidden state are

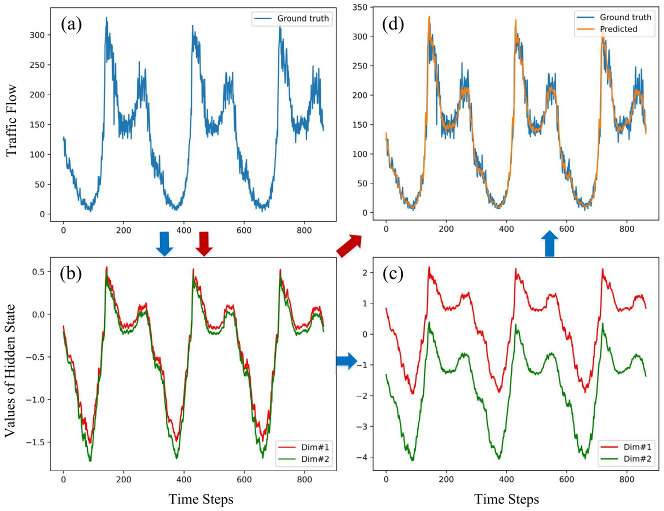


Fig. 5. The case study on PEMS03. (a) and (d) are the ground truth and predicted value of traffic flow, and (b) and (c) are the values of hidden state before and after projection function.

plotted, and their values are extracted from the full information of the history window as described in Section III.

As can be seen from Figure 5, the values of the hidden state we extract varies with time and have a certain periodicity, which indicates that the hidden state captures the pattern features in the traffic flow well. Before the projection function $\psi(\cdot)$, the values of each dimension of the hidden state are similar, which are different after projection function. This indicates that the information of the hidden state is expanded when it is mapped to the future. The framework adopted by the existing traffic prediction methods is shown by the red arrow in Figure 5, i.e., after aggregating the past temporal and spatial information, the prediction result is obtained directly through an output layer. The idea of our framework is shown by the blue arrows in Figure 5, there are also projection function and decoder to further analyze future correlations and temporal dependencies, so there is an improvement in accuracy than existing methods.

V. Related Work

The traffic prediction problem has been widely studied in the fields of spatio-temporal data mining and multivariate time series forecasting [13], including but not limited to traffic flow prediction [5], [6], [14], traffic speed prediction [1], [15], [16], [17], traffic demand prediction [18], [19], [20], [21], etc. The spatio-temporal dependencies captured from traffic data can support applications such as infrastructure layout and traffic resource dispatching [22], [23]. With the rapid development of deep learning methods, the model design for traffic prediction are becoming more and more complex. These models capture the features and relationships of historical traffic states as much as possible, and introduce external knowledge to aggregate historical information for predicting the future traffic states.

In the field of spatio-temporal data mining, the original deep learning methods divide the studied area into grids to model spatial relationships. Liu et al. [24] combined convolution with LSTM to extract spatio-temporal information of traffic flow, and learned periodic features through bi-directional LSTM. Yu et al. [25] converted traffic speeds into static images and utilized CNN and LSTM to learn spatial dependencies and temporal dynamics respectively. With the rise of Graph Neural Network (GNN) [26], [27], more and more researches modeled the transportation network as a graph structure [1], [2], [14]. Originally, these graph-based traffic prediction methods modeled the transportation networks through geographic distance, learning spatial dependencies by pre-defined and explicit characteristics. Li et al. [1] proposed diffusion convolution to capture spatial dependencies on graphs and utilize a GRU-based encoder-decoder architecture to capture temporal dependencies. Yu et al. [2] captured spatio-temporal dependencies by sandwiching two layers of causal convolution with one layer of graph convolution.

On the basis of the above works, more researches of spatio-temporal prediction have begun to follow the framework of modeling spatial and temporal dependencies separately and aggregating all dependency information to predict future states, and the definitions of spatial correlations become diverse. [3], [8], [28]. Wu et al. [3] proposed an adaptive graph adjacency matrix to discover implicit spatial relationships other than the explicit spatial relationship such as geographical distance. Zhang et al. [8] proposed structure learning convolution, arguing that the existing spatial dependencies learning for traffic is actually the information aggregation of different topological structures in non-Euclidean space. Ma et al. [29] treated the traffic network as an image, proposed CapsNet to extract the spatial features between the roadway links from the traffic state images and utilized a nested LSTM structure to capture the hierarchical temporal dependencies in traffic sequence data. Some studies have begun to consider adding temporal information to the graph to achieve dynamic spatial dependency representation or spatio-temporal synchronous learning [5], [6], but they have not yet departed from the essence of aggregating historical spatio-temporal information.

In recent years, some works in the field of multivariate time series forecasting have begun to involve traffic and other spatio-temporal data, and have also achieved good performance [7], [30], [31]. Wu et al. [7] learned temporal dependencies by dilated inception layer, modeled the relationship between different variables as an uni-directed graph, and used mix-hop to learn the correlation. Cao et al. [31] utilized discrete Fourier transform to model temporal dependencies and graph Fourier transform to model inter-series correlations. These works have more complex temporal dependencies modeling approach, but the formalization of the problem and method is consistent with the above mentioned methods in the field of spatio-temporal data mining.

Although there are also small amounts of works on feature extraction of sequence data [32], [33], [34], the problem they solve or the method adopted is completely different from ours. Kieu et al. [32] applied RNNs to autoencoders to implement outlier detection by reconstructing time series data. Wei et al. [33] used the encoder part in AE to extract the upstream and downstream traffic flow features and then sent them to LSTM to learn the time dependencies, which is still the aggregation of historical information. Nguyen et al. [34] inserted the LSTM between the encoder and decoder of the autoencoder for the purpose of nonlinear decomposition of multivariate time series. More related to our work, Ye et al. [35] proposed CoST-Net, which decomposed the urban traffic demand into a combination of hidden demand bases through a deep convolution autoencoder, and mixed the hidden states of multiple traffic modes through heterogeneous LSTMs. Ye et al. [35] and our method both leverage the autoencoder mechanism to extract the lower-dimension hidden states from traffic data. However, the spatial dependence of CoST-Net is captured by CNN on the grid map, and its traffic demand prediction results also depend on the correlation between different traffic modes. Different from their method, our work learns the correlations in non-European space, and does not require the auxiliary information from other traffic modes. In summary, there is still no existing works to predict traffic flow by extracting hidden states of traffic flow's patterns and reconstructing sequence data.

VI. Conclusion

We propose a novel traffic flow prediction framework named ST-AE. Specifically, different from the previous traffic flow prediction work that only aggregates observed spatio-temporal data, our framework extracts the hidden states of traffic flow dynamics by modeling intrinsic patterns through a specially designed autoencoder. The hidden states are projected from history to future to enable traffic flow prediction. Finally, the future traffic flow status is reconstructed by the trained decoder. We conduct experiments on four real-world data sets, and the experimental results show that our framework achieves the best performance in several settings. The accuracy improvement on long-term prediction is particularly significant. In summary, our method can predict future traffic flow just through the hidden state of intrinsic and information-rich patterns. For future work, we will explore how to extract more informative hidden states from longer historical time series.

References

[1] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–16.

[2] 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.*, Jul. 2018, pp. 3634–3640.

[3] 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.*, Aug. 2019, pp. 1–7.

[4] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1–12.

[5] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial-temporal synchronous graph convolutional networks: A new framework for spatial–temporal network data forecasting," in *Proc. AAAI*, vol. 34, no. 1, 2020, pp. 914–921.

[6] S. Li, L. Ge, Y. Lin, and B. Zeng, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," in *Proc. AAAI Artif. Intell.*, Jul. 2021, pp. 1–8.

[7] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2020, pp. 753–763.

[8] Q. Zhang, J. Chang, G. Meng, S. Xiang, and C. Pan, "Spatio-temporal graph structure learning for traffic forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 1, 2020, pp. 1177–1185.

[9] B. Lim and S. Zohren, "Time-series forecasting with deep learning: A survey," *Phil. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 379, no. 2194, Apr. 2021, Art. no. 20200209.

[10] S. Bai, J. Zico Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*.

[11] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[12] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, "Freeway performance measurement system: Mining loop detector data," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1748, no. 1, pp. 96–102, Jan. 2001.

[13] F. Li et al., "Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution," *ACM Trans. Knowl. Discovery Data*, May 2021.

[14] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial–temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI*, vol. 33, no. 1, 2019, pp. 922–929.

[15] Z. Lv, J. Xu, K. Zheng, H. Yin, P. Zhao, and X. Zhou, "LC-RNN: A deep learning model for traffic speed prediction," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3470–3476.

[16] L. Zhao et al., "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2019.

[17] L. Han, B. Du, L. Sun, Y. Fu, Y. Lv, and H. Xiong, "Dynamic and multi-faceted spatio-temporal deep learning for traffic speed forecasting," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discovery Data Mining*, Aug. 2021, pp. 547–555.

[18] H. Yao et al., "Deep multi-view spatial–temporal network for taxi demand prediction," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018, pp. 1–8.

[19] 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*, Feb. 2018, pp. 736–744.

[20] X. Geng et al., "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, 2019, pp. 3656–3663.

[21] B. Du, X. Hu, L. Sun, J. Liu, Y. Qiao, and W. Lv, "Traffic demand prediction based on dynamic transition convolutional neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 1237–1247, Feb. 2020.

[22] X. Liu, X. Qu, and X. Ma, "Improving flex-route transit services with modular autonomous vehicles," *Transp. Res. E, Logistics Transp. Rev.*, vol. 149, May 2021, Art. no. 102331.

[23] X. Liu, X. Qu, and X. Ma, "Optimizing electric bus charging infrastructure considering power matching and seasonality," *Transp. Res. D, Transp. Environ.*, vol. 100, Nov. 2021, Art. no. 103057.

[24] Y. Liu, H. Zheng, X. Feng, and Z. Chen, "Short-term traffic flow prediction with conv-LSTM," in *Proc. 9th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2017, pp. 1–6.

[25] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," *Sensors*, vol. 17, no. 7, p. 1501, Jun. 2017.

[26] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, Feb. 2017, pp. 1–14.

[27] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–12.

[28] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proc. AAAI*, 2020, vol. 34, no. 1, pp. 1234–1241.

[29] X. Ma, H. Zhong, Y. Li, J. Ma, and Y. Wang, "Forecasting transportation network speed using deep capsule networks with nested LSTM models," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 4813–4824, Aug. 2021.

[30] B. N. Oreshkin, A. Amini, L. Coyle, and M. J. Coates, "FC-GAGA: Fully connected gated graph architecture for spatio-temporal traffic forecasting," 2020, *arXiv:2007.15531*.

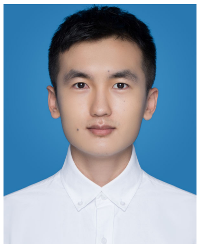
[31] D. Cao et al., "Spectral temporal graph neural network for multivariate time-series forecasting," 2021, *arXiv:2103.07719*.

[32] T. Kieu, B. Yang, C. Guo, and C. S. Jensen, "Outlier detection for time series with recurrent autoencoder ensembles," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 2725–2732.

[33] W. Wei, H. Wu, and H. Ma, "An autoencoder and LSTM-based traffic flow prediction method," *Sensors*, vol. 19, no. 13, p. 2946, Jul. 2019.

[34] N. Nguyen and B. Quanz, "Temporal latent auto-encoder: A method for probabilistic multivariate time series forecasting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 10, 2021, pp. 9117–9125.

[35] J. Ye, L. Sun, B. Du, Y. Fu, X. Tong, and H. Xiong, "Co-prediction of multiple transportation demands based on deep spatio-temporal neural network," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2019, pp. 305–313.



Mingzhe Liu (Graduate Student Member, IEEE) received the B.S. degree from the College of Computer Science and Technology, Shandong University, China, in 2020. He is currently pursuing the M.S. degree in computer science and engineering with Beihang University, Beijing, China. His research interests include intelligent transportation, deep learning, and spatio-temporal data mining.

Tongyu Zhu received the B.S. degree from Tsinghua University in 1992 and the M.S. degree from Beihang University, Beijing, China, in 1999. He is currently an Associate Professor with the State Key Laboratory of Software Development Environment, School of Computer Science, Beihang University. His research interests include intelligent traffic information processing and network application.

Junchen Ye received the B.S. degree in computer science from Beihang University, China, in 2018, where he is currently pursuing the Ph.D. degree in software engineering. His research interests include intelligent transportation, deep learning, and graph neural networks.

Qingxin Meng received the bachelor's degree from the Department of Precision Machinery and Precision Instruments, University of Science and Technology of China, and the Ph.D. degree in management science and information systems from Rutgers University. She is currently an Assistant Professor with the Nottingham Business School, Ningbo, China. She has devoted herself to the research of applying big data techniques in talent management. Her research has been published in ACM Knowledge Discovery and Data Mining (KDD) and *Informs Journal on Computing* (INFORMS JOC).

Leilei Sun (Member, IEEE) received the Ph.D. degree from the Institute of Systems Engineering, Dalian University of Technology, in 2017. He is currently an Assistant Professor with the School of Computer Science, Beihang University, Beijing, China. From 2017 to 2019, he was a Post-Doctoral Research Fellow with the School of Economics and Management, Tsinghua University. His research interests include machine learning and data mining.

Bowen Du (Member, IEEE) received the Ph.D. degree in computer science and engineering from Beihang University, Beijing, China, in 2013. He is currently a Professor with the State Key Laboratory of Software Development Environment, Beihang University. His research interests include smart city technology, multi-source data fusion, and traffic data mining.