Latent Gaussian Processes Based Graph Learning for Urban Traffic Prediction

Xu Wang, Pengkun Wang, *Student Member, IEEE*, Binwu Wang, Yudong Zhang, *Student Member, IEEE*, Zhengyang Zhou, *Student Member, IEEE*, Lei Bai, and Yang Wang, *Senior Member, IEEE*

***Abstract*—Traffic prediction facilitates various applications in the fields of smart vehicles and vehicular communications, and the key of successfully and accurately forecasting urban traffic state is to model the complex spatiotemporal correlations within urban traffic networks. However, even though great efforts have been devoted to modeling the spatiotemporal correlations, such issue remains challenging in the following two loci, i) capturing dynamic spatial correlations and ii) efficiently and holistically modeling temporal trends. To tackle these challenges, in this article, we propose a novel Adaptive Graph convolution based Autoencoder with Latent Gaussian processes (AGALG). Specifically, a graph fusion module is proposed to fuse learnable graphs for modeling static spatial correlations and data-driven graphs for dynamic spatial correlations. The fused graphs are fed into an adaptive graph convolutional autoencoder whose encoder represents both historical and future data in a latent space where they are holistically modeled by Gaussian processes generated by a latent Gaussian generator. We apply a Gaussian processes regressor to make estimations of latent representations of future data, and the decoder of autoencoder finally generates predictions. Experimental results on several widely used benchmark datasets validate that the proposed model achieves the state-of-the-art performance.**

***Index Terms*—Gaussian process, traffic pattern, traffic prediction, vehicular data.**

I. Introduction

TRAFFIC prediction, aiming at forecasting future traffic states, serves as the basis of various applications in the area of smart vehicles and intelligent transportation systems [1], [2]. For instance, [3] utilizes traffic trends to perform effective route planning, [4] embeds a short-term traffic prediction method into the vehicle communication system for improving robustness and efficiency. Therefore, developing reliable and accurate traffic prediction approaches has been a research hotspot recently.

Since the traffic state at a given location is strongly correlated to those of its adjacent locations and its historical traffic states, the success of traffic prediction mainly lies in whether the complex spatial and temporal correlations within traffic networks can be extensively extracted. Numerous approaches have been proposed to address the challenge of modeling spatiotemporal correlations in traffic network. Early works apply statistic algorithms including ARIMA [5] and VAR [6] to model temporal trends. As spatial correlations are ignored in these statistic methods, the performances of them are very frustrating. Therefore, following works [7], [8] utilize machine learning algorithms to capture both spatial and temporal correlations. However, considering that the representative abilities of traditional machine learning methods are extremely limited, those works fail to achieve satisfying forecasting accuracy. Therefore, researchers tend to utilize deep neural networks to capture deep spatiotemporal correlations [9] by taking advantage of their superior representative abilities.

Early deep learning based attempts [10], [11] apply Convolutional Neural Networks (CNNs) to abstract spatial correlations in traffic networks. However, urban traffic networks have graph-like spatial structures and it is difficult for CNN based networks to capture such non-Euclidean correlations [12]. Therefore, Graph Convolutional Networks (GCNs) have been introduced to model the non-Euclidean spatial correlations in traffic networks [13]. STGCN and T-GCN [14], [15] apply GCNs on pre-defined graph structures. [16], [17] argue that pre-defined graph structures are not optimal for traffic forecasting and propose self-adaptive adjacency matrices for modeling spatial correlations. However, the learned adjacency matrices remain fixed during predicting and existing works thus fall short of modeling dynamic spatial correlations.[1] *In summary, how to effectively capture such dynamic spatial correlations within traffic networks remains a challenge for existing deep learning based methods.*

Regarding capturing temporal correlations, existing models can be divided into two categories, autoregressive models and multi-step forecasting models. Autoregressive models [18], [19], [20] mainly apply Recurrent Neural Network (RNN) or transformer based models to generate predictions at each time step iteratively. By taking advantage of the recurrent mechanism,

Manuscript received 9 March 2023; revised 12 July 2023; accepted 14 August 2023. Date of publication 23 August 2023; date of current version 17 January 2024. This work was supported in part by the National Natural Science Foundation of China under Grants 62072427 and 12227901 and in part by the Project of Stable Support for Youth Team in Basic Research Field, CAS through Academic Leaders Cultivation Program, USTC under Grant YSBR-005. The review of this article was coordinated by Prof. Ju Ren. *(Corresponding author: Yang Wang.)*

Xu Wang, Pengkun Wang, Binwu Wang, Yudong Zhang, Zhengyang Zhou, and Yang Wang are with the University of Science and Technology of China, Heifei 230026, China (e-mail: wx309@mail.ustc.edu.cn; pengkun@mail.ustc.edu.cn; wbw1995@mail.ustc.edu.cn; zyd2020@mail.ustc.edu.cn; zzy0929@mail.ustc.edu.cn; angyan@ustc.edu.cn).

Lei Bai is with the Shanghai AI Laboratory, Shanghai 200025, China (e-mail: baisanshi@gmail.com).

Digital Object Identifier 10.1109/TVT.2023.3307755

[1] As known, regarding different traffic flow patterns in different time periods, e.g., morning and afternoon, the spatial correlations between two intersections are not fixed, and such time varying correlations are referred to as dynamic spatial correlations.

0018-9545 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on September 18,2024 at 02:53:10 UTC from IEEE Xplore. Restrictions apply.

autoregressive models are capable of using future data as input while training, thus holistically analyzing the temporal trends of traffic state, i.e., capturing long-term correlations by taking both historical and future data into account. However, those autoregressive models, which have to generate prediction iteratively, will definitely lead to unsatisfying inferring speed and some other difficulties in training in such holistic learning scenario. On the contrary, multi-step models [14], [16], [17], [21] treat historical data and future data separately and learn mapping functions from historical data to future data. Given the fact that multi-step models generate predictions on all future time steps at once, they are more time-efficient in some degree. However, those multi-step models, which are unaware of future status of variables during training and cannot model the temporal trends of traffic state from a long-term perspective, will definitely fall short of accuracy in prediction. *In general, how to holistically capture both historical and future long-term temporal correlations in an efficient way remains challenging yet for existing deep learning based methods.*

Regarding the task of traffic forecasting, to tackle the above two challenges, we here propose an Adaptive Graph convolutional Autoencoder with Latent Gaussian process (AGALG). Regarding the first challenge, i.e., how to model dynamic spatial correlations in a more effective manner, we propose a novel Graph Fusion Module (GFM) to extract both static and dynamic correlations in spatial perspective. Specifically, in GFM, we propose two novel graphs, a learnable graph for extracting latent static correlations and a data-driven graph for representing dynamic spatial correlations, and these two graphs are then fused by GFM as the input of subsequent graph convolutional module. Next, based on the fused graph, we construct an Adaptive Graph convolutional Autoencoder (AGA) to further extracting spatial correlations by utilizing its integrated adaptive graph convolution module to capture node-specific patterns. Regarding the second challenge, i.e., how to extract temporal correlations over dynamically changing traffic states, we design a novel Latent Gaussian Generator (LGG) which can simultaneously take the latent representations of both historical and future data into account and model the long-term temporal correlations by constructing the joint distribution of them with Gaussian processes. In particular, during training phase, regarding each individual specific node, LGG uses a joint Gaussian process to constrain the output latent representation of AGA encoder to unify the historical and future long-term correlations of this node; and during predicting phase, a newly proposed Gaussian Processes Regressor (GPR) module can directly construct the estimation of latent code of future traffic state based on the latent code of historical data and the constructed joint Gaussian process, and finally such estimation is then fed to AGA decoder to generate the prediction result. We extract a series of experiments on several widely used benchmark datasets of traffic prediction to evaluate the performance of the proposed model and the results validate that our proposed model outperforms other alternative solutions in terms of traffic prediction. The main contributions of this article are as follows,

- We propose a graph fusion module, GFM, to construct an adjacency matrix for modeling both static and dynamic spatial correlations within traffic network and an adaptive graph convolutional autoencoder, AGA, to capture node-specific patterns.
- We propose a latent Gaussian generator, LGG, for applying Gaussian processes to holistically and efficiently model the long-term temporal correlations in traffic networks and utilize a Gaussian processes regressor, GPR, for inferring.
- By taking advantage of the constraint relationship between latent space of AGA and Gaussian process, for the first time, our model can simultaneously model both historical and future long-term temporal correlation as well as the dynamic spatial correlations in a holistic patterns, hence achieving accurate multi-step prediction.
- We evaluate the proposed model on several widely used benchmark datasets. The experimental results demonstrate the superiority of the proposed model on traffic prediction.

The rest of this article is organized as follows. We review related works in Section II. Section III introduces the preliminaries and Section IV details the proposed model. The experimental settings and results are given in Sections V and VI concludes the article.

II. Related Works

A. Traffic Forecasting

Traffic forecasting facilitates many applications in intelligent transportation systems [22], [23], [24], [25]. Great efforts have been devoted to modeling the spatial and temporal correlations among variables for more accurate traffic prediction. Existing traffic forecasting model can be divided into two categories, autoregressive models and multi-step forecasting models. Autoregressive models use the chain rule to decompose time series, and generate predictions step by step. ARIMA [26] and [5] utilize linear models to capture the interdependencies among time series. Some recent works [27], [28] apply recurrent neural networks to model the correlations among variables in the domain of traffic. DCRNN [18] combine RNN with diffusion convolution to capture both spatial and temporal dependencies among time series. T-GCN [15] capture spatial dependencies with graph convolution and utilize GRU to extract temporal dependencies. With the rise of transformers [29], some researchers introduce transformers into the area of traffic forecasting [19], [20]. Although autoregressive models achieve much better prediction accuracy by taking advantage of novel deep learning models, they fall short of efficiently generating multi-step predictions. Multi-step forecasting models are more deep-learning, which simply learn the projection between historical data and future data and ignore the inherent trends of the multi variables. [14], [16], [21] apply dilated convolutional network to capture the temporal interdependencies, and utilize graph convolution with different kind of adjacency matrices to capture spatial dependencies. SLCNN [30] extends traditional CNN to graph-structured data to build an end-to-end traffic prediction network. [31] proposes a novel Attention Temporal Graph Convolutional Network (A3T-GCN) for the application of traffic flow forecasting, and this proposed network can simultaneously capture both spatial and dynamic temporal correlations. AGCRN [17] utilize RNN to

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on September 18,2024 at 02:53:10 UTC from IEEE Xplore. Restrictions apply.

capture temporal dependencies but does not generate prediction in recurrent manner but directly outputs multi-step predictions based on the latent state at the last historical time step. Similarly, [32], [33], [34] apply transformers to model the temporal dependencies and generate multi-step predictions directly like AGCRN.

B. Gaussian Process and Neural Network

Gaussian processes, which are theoretically able to approximate arbitrary smooth functions satisfying mean square continuity and differentiability [35], have achieved promising performance on univariate time series forecasting problem [36], [37]. The core of utilizing Gaussian processes for univariate regression is to find proper prior forms of kernel functions, which have significant affects on the performance. [38] shows that Bayesian neural networks with infinitely many hidden units converged to Gaussian processes with a particular kernel (covariance) function, which reveals the connections between Gaussian processes and neural networks. Considering the potential of Gaussian processes for modeling functions, Neural Process (NP) [39] and Conditional Neural Process (CNP) [40] combine benefits of neural networks and Gaussian processes, which bypass the kernel function and utilize neural networks to directly generate the conditional distribution of target variable. Following NP, [41] builds a new member of Neural Process Families called GloBal Convolutional Neural Process (GBCoNP), which defines global uncertainty to represent a belief of prior form of function. [42] proposes an additional loss term which is conceptually equivalent to CNP, and the proposed loss can be used as regularization for many kinds of neural networks. In this article, we propose a new member of Neural Process Families to tackles the multi-step multivariate forecasting problem which generates kernel functions for Gaussian processes rather than conditional distributions. The key difference between our model and existing NP models is that existing NP models are trying to implement or approximate Gaussian processes with neural networks while our model is to embed Gaussian processes into neural networks to implement forecasting. Actually, existing NP models can be modified and embedded into our model for replacing Gaussian Process Regressor and Latent Gaussian Generator.

III. Preliminary

In this section, we introduce the basis of this work and formally define the problem of traffic prediction.

A. Traffic State

At time step t, the traffic state collected from the i-th sensor region is denoted as $x_t^i \in \mathbb{R}^d$, where d is the number of features, e.g., traffic speed and traffic flow. Thus, we have the state of the whole traffic network at time step t denoted as $X_t = [X_t^1, X_t^2, \ldots, X_t^N] \in \mathbb{R}^{N\times d}$.

B. Represent Traffic Network With Graph

Recent works utilize graph structure to model the traffic network. A traffic network can be formulated as $G = (X, E)$ where X denotes the set of nodes, i.e., features collected from N sensor regions in the traffic network, and E is the set of edges, which represent the correlations among nodes. We can also formulate a graph with adjacency matrix as $G(X, A)$, where $A \in \mathbb{R}^{N\times N}$ is the adjacency matrix where $A_{ij} = 1$ if $(x_i, x_j) \in E$ and $A_{ij} = 0$ else wise.

C. Gaussian Processes for Time Series Modeling

Gaussian Processes are widely used in modeling time series. A Gaussian process is a collection of random variables, where any finite number of variables have joint Gaussian distributions [35]. A Gaussian process denoted as $\mathcal{GP}(m, k)$ can be specified by its mean function $m(t)$ and covariance function $k(t, t')$, which determine the joint Gaussian distribution given the indexes of variables. For instance, when given indexes of 2 variables $[x_{t_1}, x_{t_2}]$, we have,

$$\begin{bmatrix} x_{t_1} \\ x_{t_2} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_{t_1} \\ \mu_{t_2} \end{bmatrix}, \begin{bmatrix} \Sigma_{t_1 t_1} & \Sigma_{t_1 t_2} \\ \Sigma_{t_2 t_1} & \Sigma_{t_2 t_2} \end{bmatrix} \right) \tag{1}$$

where $\mu_i = m(t_i)$ and $\Sigma_{t_i t_j} = k(t_i, t_j)$, $i, j \in \{1, 2\}$. The key to modeling time series with Gaussian processes is to find suitable mean and covariance (kernel) functions.

D. Traffic Prediction

Give the observed states of all the sensor regions during P historical time steps $[X_{t-P}, X_{t-P+1}, \ldots, X_{t-1}]$, the goal of traffic prediction is to predict the state of traffic network of the following Q time steps $[X_t, X_{t+1}, \ldots, X_{t+Q-1}]$, which can be formulated as estimating the conditional distribution,

$$p(X_t, X_{t+1}, \ldots, X_{t+Q-1} | X_{t-P}, X_{t-P+1}, \ldots, X_{t-1}) \tag{2}$$

E. Optimizing Goal of Applying Latent Gaussian Processes

We derive the optimizing goal of applying latent Gaussian processes for traffic forecasting. For the convenience of derivation and notation, we simplify the goal of traffic forecasting in (2) as modeling $p(X_t|X_{<t})$ which can easily be generalized to multi-step forecasting. To introduce Gaussian processes to traffic forecasting, we apply latent variable to model $p(X_t|X_{<t})$, as,

$$\begin{aligned} p(X_t|X_{<t}) &= \frac{p(X_{\leqslant t})}{p(X_{<t})} \\ &= \mathbb{E}_{Z_{\leqslant t}\sim q(Z_{\leqslant t}|X_{\leqslant t})}\left[\frac{p(X_{\leqslant t}, Z_{\leqslant t})}{p(Z_{\leqslant t}|X_{\leqslant t})}\frac{p(Z_{<t}|X_{<t})}{p(X_{<t}, Z_{<t})}\right] \end{aligned} \tag{3}$$

where $q(Z_{\leqslant t}|X_{\leqslant t})$ is the prior Gaussian process. Given observed traffic state $X_{\leqslant t}$, $q(Z_{\leqslant t}|X_{\leqslant t})$ determine a joint Gaussian distribution for $Z_{\leqslant t}$. For notation brevity, we omit the subscript $Z_{\leqslant t} \sim q(Z_{\leqslant t})$ in the following derivation. Since we have

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on September 18,2024 at 02:53:10 UTC from IEEE Xplore. Restrictions apply.

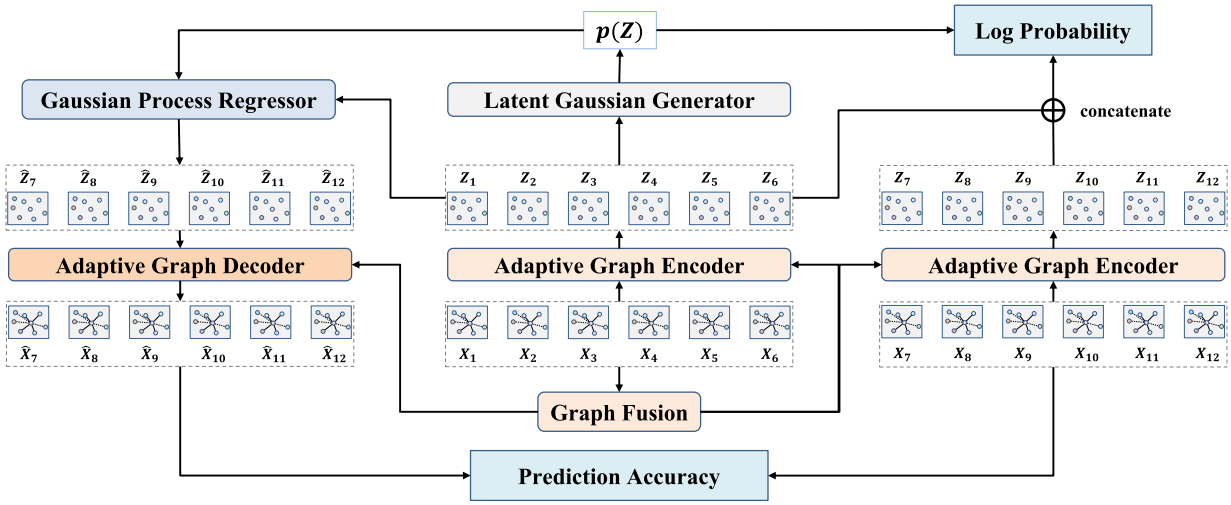

Fig. 1. Solution Overview. Graph Fusion constructs fused graph for capturing both static and dynamic spatial correlations. The fused graph is fed into Adaptive Graph Autoencoder which consists of adaptive graph encoder and decoder. The encoder generates latent representation of traffic state and the decoder recover traffic state based on latent representation. Latent Gaussian Generator constructs Gaussian processes to constrain the latent representation of both historical data and future data, e.g., Z_{1-6} and Z_{7-12} in this figure. Gaussian Process Regressor estimates the latent code of future traffic state according to latent code of historical traffic state and the constructed joint Gaussian process, and finally such estimation is then fed to AGA decoder to generate the prediction result.

$\frac{p(X_{\leqslant t},Z_{\leqslant t})}{p(X_{<t},Z_{<t})} = p(X_t, Z_t|X_{<t}, Z_{<t})$, (3) can be further transformed as,

$$\begin{aligned} p(X_t|X_{<t}) &= \frac{p(X_{\leqslant t})}{p(X_{<t})} \\ &= \mathbb{E}\left[\frac{p(X_t|Z_T)p(Z_t|X_{<t}, Z_{<t})p(Z_{<t}|X_{<t})}{p(Z_{\leqslant t}|X_{\leqslant t})}\right] \end{aligned} \tag{4}$$

Noting that $p(Z_t|X_{<t}, Z_{<t})p(Z_{<t}|X_{<t}) = p(Z_{\leqslant t}|X_{<t})$, we finally get,

$$p(X_t|X_{<t}) = \mathbb{E}\left[\frac{p(X_t|Z_t)p(Z_{\leqslant t}|X_{<t})}{p(Z_{\leqslant t}|X_{\leqslant t})}\right] \tag{5}$$

The goal of our model is to maximize the probability in (5) and the following loss can be obtained by applying ELBO trick [43],

$$\begin{aligned} \mathcal{L} &= -\log p(X_t|X_{<t}) \\ &\leq -\mathbb{E}\left[\log p(X_t|Z_t) + \log \frac{p(Z_{\leqslant t}|X_{<t})}{p(Z_{\leqslant t}|X_{\leqslant t})}\right] \end{aligned} \tag{6}$$

IV. Method

Based on the derived optimizing goal in (6), we propose our model as illustrated in Fig. 1. The first term in (6) means the model needs to minimize prediction error given latent representation Z_t, and the second term minimizes the divergency between prior Gaussian distribution $Z_{\leqslant t} \sim q(Z_{\leqslant t}|X_{\leqslant t})$ and conditional Gaussian distribution $p(Z_{\leqslant t}|X_{<t})$ given historical data $X_{<t}$. According the above optimizing goal, we propose an Adaptive Graph convolutional Autoencoder with Latent Gaussian process (AGALG), which consists of four parts, a graph fusion module (GFM), an adaptive graph autoencoder (AGA), a latent Gaussian generator (LGG) and a Gaussian process regressor (GPR). We detail each component in the following.

A. Graph Fusion Module

For better modeling spatial correlations among nodes in traffic networks, we propose two kind of graphs, i.e., learnable static graph A_S and dynamic data-driven graph A_D, and fuse them in the proposed Graph Fusion Module (GFM).

Learnable static graph: Learnable static graph A_S is defined to model the static spatial correlations among nodes, e.g., intersections connected by roads are certainly correlated. Early works capture such static spatial correlations by constructing distance based or road network based adjacency matrices. However, the matrices generated are intuitive, not specific to the prediction task and are unable to model implicit spatial correlations. To this end, we propose to generate the adjacency matrices for such static spatial correlations in a learning manner. The graph fusion module maintains e-dimension embeddings for each variable, denoted as $E \in R^{N\times e}$, and determines the correlations among nodes by calculating the similarity of their embeddings, which can be formulated as,

$$A_S(i,j) =< E_i, E_j >= \frac{E_i \cdot E_j}{|E_i||E_j|} \tag{7}$$

where $A_S(i,j)$ denotes the adjacency between i-th and j-th nodes, E_i corresponds to the embedding of i-th node, $\cdot$ means dot product and $|E_i|$ means the norm of E_i. Notably, we denote such operation in Fig. 2 with a black cross for brevity.

Dynamic data-driven graph: The learnable static graph A_S keeps fixed after training thus falls short of modeling some dynamic spatial correlations among nodes. For instance, different traffic flow patterns in the morning and noon certainly lead to different spatial correlations. Addressing the issue that static graph can not model the dynamic spatial correlations, we propose a data-driven method to construct dynamic graphs based on the observed traffic state. Specifically, given observed

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on September 18,2024 at 02:53:10 UTC from IEEE Xplore. Restrictions apply.

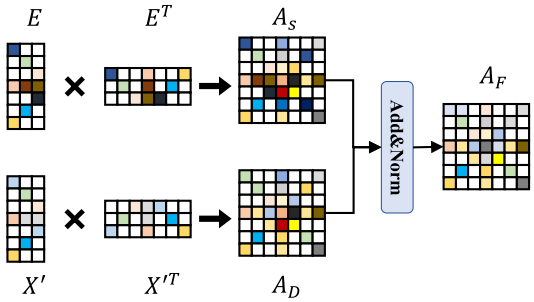

Fig. 2. Graph fusion of learnable static graph A_S and dynamic data-driven graph A_D.

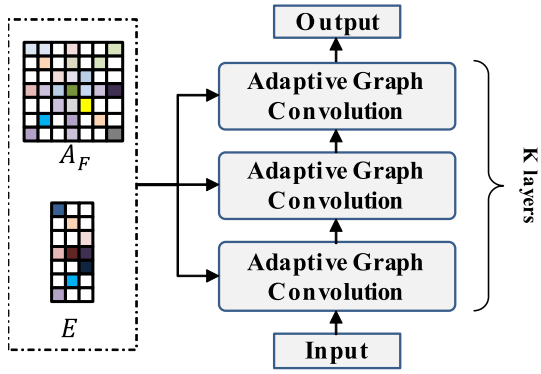

Fig. 3. Architecture of encoder and decoder in AGA.

historical observation X, we apply a 1D convolution on it and construct the dynamic graph A_D in the same way as (7).

$$A_D(i,j) =< X'_i, X'_j >= \frac{X'_i \cdot X'_j}{|X'_i||X'_j|} \tag{8}$$

where X' is the result of applying the 1D convolution on X. Taking advantage of convolution, X' is able to encode local temporal trend [19].

Graph fusion: The obtained learnable static graph A_S and dynamic data-driven graph A_D are further fused as in Fig. 2 before being fed into AGA. The fusion process is simply defined as,

$$A_F = \frac{(A_D + A_S) - \min(A_D + A_S)}{\max(A_D + A_S)} \tag{9}$$

which restricts the values in A_F into [0, 1]. The fused graph A_F are taken as input in both encoder and decoder of AGA.

B. Adaptive Graph Convolutional Autoencoder

The interdependencies among nodes in traffic network result in dependencies of latent representations of nodes, making it intractable to model them by independent Gaussian distributions in latent space. However, constructing dependent distributions of the latent representation of all nodes leads to unacceptable time and memory consumption. Therefore, we propose an autoencoder structure to decouple the dependencies among variables in the encoder and recouple them in the decoder, so that the nodes are decoupled in the latent space. With the rise of graph convolution, many recent works [30], [44], [45] are tend to model the interdependencies among nodes by graph convolution networks. Following those works, we here apply graph convolution to build the autoencoder.

According to [46], we can approximate the graph convolution operation by 1_{st} order Chebyshev polynomial expansion and generalized to high-dimensional GCN as:

$$Y = \left(I + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\right) XW + B \tag{10}$$

where $X \in \mathbb{R}^{N \times d}$ denotes the input, $A, D \in \mathbb{R}^{N \times N}$ are the adjacency matrix and degree matrix, respectively. $W \in \mathbb{R}^{d \times d'}$ and $B \in \mathbb{R}^{d'}$ are learnable parameters. Considering that in the task of traffic forecasting, the adjacency matrix and degree matrix could be implicit and undefined, we modify the adaptive graph convolution proposed in [17] to build an Adaptive Graph convolutional Autoencoder (AGA). The adaptive graph convolution reuses the embeddings proposed in GFM and the computational procedure of the proposed adaptive graph convolution can be formulated as,

$$Y = \text{softmax}(A_F)X \odot (EW_g) + EB_g \tag{11}$$

where $E \in R^{N \times e}, W_g \in \mathbb{R}^{e \times d \times d'}$ and $B_g \in \mathbb{R}^{e \times d'}$ are trainable parameters. Given $X \in \mathbb{R}^{N \times d}$ and $W \in \mathbb{R}^{N \times d \times d'}$, $\odot$ is defined as,

$$X \odot W = [X_1 W_1, X_2 W_2, \ldots, X_N W_N]^T \tag{12}$$

which corresponds to matrix multiplications applied at each row of X and W. And A_F is the adjacency matrix generated by the graph fusion module, which has been introduced as in (9). Both encoder and decoder are stacked adaptive graph convolution layers as in Fig. 3. The encoder and decoder have the same architecture. Given historical data X, the encoder of AGA transforms X into latent code Z_X, and given latent code Z_Y of future data, the decoder of AGA transforms Z_Y back to future data Y. Taking advantage of the stacked graph convolution, the autoencoder is able to decouple the dependencies among variables in encoder and recouple them in the decoder. The decoupling of the correlations among variables is implied in the process of Latent Gaussian Generator (LGG). Since LGG generates independent N Gaussian processes for N nodes, AGA are forced to learn decoupled latent code of different nodes. Otherwise, the model would fail to make accurate prediction or even fail to converge. In the following, we detail the design of LGG.

C. Latent Gaussian Generator

As mentioned, the second term in (6) requires the model to minimize the divergency between prior Gaussian distribution $Z_{\leqslant t} \sim q(Z_{\leqslant t}|X_{\leqslant t})$ and conditional Gaussian distribution $p(Z_{\leqslant t}|X_{<t})$ given historical data $X_{<t}$. To achieve this goal, the proposed Latent Gaussian Generator (LGG) works as the mean and covariance functions of Gaussian processes, and outputs a joint distribution $\hat{p}$ of $Z_{\leqslant t}$ based on $Z_{<t}$. Combined with the encoder of AGA, which outputs $Z_{<t}$ according to $X_{<t}$, we are able to generate $Z_{\leqslant t}$ base on $X_{<t}$.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on September 18,2024 at 02:53:10 UTC from IEEE Xplore. Restrictions apply.

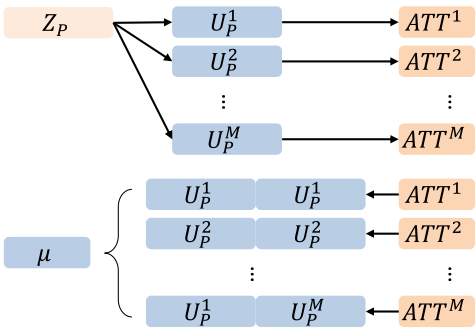

Fig. 4. Generation procedure of mean vector in LGG.

Specifically, in the scenario of traffic forecasting, our goal is to generate Z_{P+Q} base on X_P, where P and Q correspond to the length of historical data and future data. Thus, the dimension of latent space is $\mathbb{R}^{N\times(P+Q)}$, and to construct the joint Gaussian distribution of Z_{P+Q}, a mean vector and covariance matrix are required. Using single learnable mean vector and covariance matrix would severely limit the performance on estimation. Therefore, the proposed LGG maintains a pool of M mean vectors $\boldsymbol{U} \in \mathbb{R}^{M\times N\times(P+Q)}$ and M covariance vectors $\boldsymbol{S} \in \mathbb{R}^{M\times N\times(P+Q)\times 1}$ and generates the conditional distribution by utilizing the pool. Notably, all the mean vectors and covariance vectors are trainable. The procedure of LGG is demonstrated in Fig. 4. Given latent representation $Z_P \in \mathbb{R}^{N\times P}$, LLG first calculates attention values for each mean and covariance vectors by,

$$ATT^m = \frac{\exp(\sum_{i=1}^{N}\sum_{j=1}^{T-1} Z_P^{ij}\boldsymbol{U}^{mij})}{\sum_{k=1}^{M}\exp(\sum_{i=1}^{N}\sum_{j=1}^{T-1} Z_P^{ij}\boldsymbol{U}^{kij})} \tag{13}$$

Then, the mean of conditional distribution can be calculated as,

$$\mu = \sum_{m=1}^{M} ATT^m \boldsymbol{U}^m \tag{14}$$

Considering that the covariance matrix must be positive-semidefinite, we calculate it as,

$$\boldsymbol{s} = \sum_{m=1}^{M} ATT^m \boldsymbol{S}^m$$
$$\Sigma = \boldsymbol{s}\boldsymbol{s}^T + I \tag{15}$$

which makes Σ a positive-definite matrix. So far, we obtain the conditional distribution $p(Z_{P+Q}|X_P)$ as $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

During training, we need to measure the divergency between $q(Z_{P+Q}|X_{P+Q})$ and $p(Z_{P+Q}|X_P)$. Given observed data X_P and corresponding label X_Q, the encoder of AGA represents them into latent space and obtains $Z_{P+Q} = [Z_P, Z_Q]$. Then LGG generates the conditional distribution $p(Z_{P+Q}|X_P)$ and we apply maximum likelihood to train the parameters of LGG by maximizing the posterior probability that Z_{P+Q} is sampled from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$,

$$\max_{\boldsymbol{U},\boldsymbol{S}} \log(p(Z_{P+Q})) \tag{16}$$

After obtaining $p(Z_{P+Q}|X_P)$, the Gaussian process regressor (GPR) generates estimation of Z_Q according to $p(Z_{P+Q}|X_P)$ and Z_P, which is then fed into the decoder of AGG for generating the final prediction.

D. Gaussian Processes Regressor

Given the latent representation Z_P of historical data, LGG generates the conditional distribution of Z_{P+Q}, which is a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ as,

$$\begin{bmatrix} Z_P \\ Z_Q \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mu_P \\ \mu_Q \end{bmatrix}, \begin{bmatrix} \Sigma_{PP} & \Sigma_{PQ} \\ \Sigma_{QP} & \Sigma_{QQ} \end{bmatrix}\right) \tag{17}$$

Based on the above equation, the Gaussian Processes Regressor (GPR) is able to build the conditional distribution of Z_Q according to the joint distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ by,

$$Z_Q|Z_P \sim \mathcal{N}(\mu_Q + \Sigma_{QP}\Sigma_{PP}^{-1}(Z_P - \mu_P),$$
$$\Sigma_{QQ} - \Sigma_{QP}\Sigma_{PP}^{-1}\Sigma_{PQ}) \tag{18}$$

We make estimation $\hat{Z}_Q$ of Z_Q by simply taking the mean of the above distribution,

$$\hat{Z}_Q = \mu_Q + \Sigma_{QP}\Sigma_{PP}^{-1}(Z_P - \mu_P) \tag{19}$$

E. Training and Inferring Procedures

Given input X_P and target X_Q, GFM first computes the fused graph A_F for AGA. The encoder of AGA represents both the input and the target into the latent space as Z_P and Z_Q. Then, LGG constructs the estimated joint distribution p of Z_P and Z_Q according to Z_P, i.e., $p(Z_{P+Q}|X_P)$. According to (18), GPR generates the estimation $\hat{Z}_Q$ of Z_Q. Given $\hat{Z}_Q$, the decoder of AGA makes the prediction and output $\hat{X}_Q$.

With the light of (6), we have the final loss function for training as,

$$\mathcal{L} = \lambda \text{mean}(|\hat{X}_Q - X_Q|) - \log(p(Z_P, Z_Q)) \tag{20}$$

where p denotes the conditional joint distribution $p(Z_P, Z_Q|X_P)$, $\log(p(Z_P, Z_Q))$ is the log probability that (Z_P, Z_Q) is sampled from p, the constructed joint distribution of Z_P and Z_Q. $\text{mean}(|\hat{X}_Q - X_Q|)$ and $\log(p(Z_P, Z_Q))$ correspond to the first and second terms in the (6), respectively, and λ is a hyperparameter to adjust weight of the first term.

V. Experiment

To evaluate the performance of proposed model on traffic forecasting, we employ four datasets and conduct experiments with two kind of settings, i.e., single-step forecasting, using P historical data to predict future status at a given future time step, and multi-step forecasting, using P historical data to predict the status at the following Q time steps.

A. Datasets

As shown in Table I, we employ four widely used multivariate datasets to evaluate the proposed method, including:

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on September 18,2024 at 02:53:10 UTC from IEEE Xplore. Restrictions apply.

TABLE I
DATASETS STATISTICS

Dataset	traffic	TaxiBJ	METR-LA	PEMS-BAY
Timespan	1/1/2008-3/30/2009	11/1/2015-4/10/2016	3/1/2012-6/30/2012	1/1/2017-5/31/2017
#Nodes	963	1024	207	325
Input length	48	48	12	12
Output length	1	1	12	12
Time interval	10 min	30 min	5 min	5 min
Data type	Road occupancy	Taxicab flow	Traffic speed	Traffic speed

Traffic[2] This dataset is collected by 963 sensors and describes the occupancy rate, between 0 and 1, of different car lanes of San Francisco bay area freeways. The measurements cover the period from Jan. 1st 2008 to Mar. 30th 2009 and are sampled every 10 minutes.

TaxiBJ This dataset is introduced by [11], which contains taxicab GPS data in Beijing, China from 1st Nov. 2015 to 10th Apr. 2016. The urban area of Beijing is divided into a 32×32 grids, and each record in TaxiBJ corresponds to the taxicab inflow and outflow of a specific grid within 30 minutes.

METR-LA This dataset is collected by the Los Angeles Metropolitan Transportation Authority [47], and contains average traffic speed measured by 207 loop detectors on the highways of Los Angeles County ranging from Mar 2012 to Jun 2012 with time interval 5 minutes. The total number of time points is 34272.

PEMS-BAY [18] This dataset comes from the California Department of Transportation (Caltrans) Performance Measurement System (PeMS) [48] and is collected by 325 sensors in the Bay Area over a period of 6 months from Jan 1st 2017 to May 31st 2017 with time interval 5 minutes. The total number of time points is 52,116.

B. Baselines and Metrics

The proposed model is compared with different baselines in single-step forecasting, i.e., using P historical data to predict future status at a given future time step, and multi-step forecasting, i.e., using P historical data to predict the status at the following Q time steps. We employ different baselines because single-step forecasting models perform badly on multi-step forecasting task and multi-step forecasting models are unable to achieve competitive accuracy on single-step forecasting. Comparing different SOTA models on different tasks aims at better evaluating the generalization capacity of our model.

Single-step forecasting baselines:

- Vector Auto-Regression (VAR) is a statistic model capturing linear correlations.
- LSTNet [49] discovers the short-term and long-term patterns.
- MTNet [50] maintain a large memory component to capture long-term dependencies.
- DSANet [51] captures both spatial and temporal nonlinear dependencies.
- MTGNN [21] applies graph convolution and dilated inception convolution to capture both spatial and temporal dependencies.
- CATN [52] constructs a tree structure to learn hierarchical dependencies and proposes an end-to-end forecasting model.

Multi-step forecasting baselines:

- DCRNN [18] combines diffusion graph convolutions with recurrent neural networks.
- STGCN [14] combines graph convolution with 1D convolutions.
- Graph WaveNet [16] integrates 1D dilated convolution with diffusion graph convolution.
- GMAN [53] utilizes spatial and temporal attention for forecasting.
- AGCRN [17] proposes an adaptive graph convolution and combine it with RNN.
- GWNET [42] introduces a covariance loss applicable to many kinds of networks.
- FC-GAGA [54] proposes a fully connected spatiotemporal model combined with temporal and graph gating mechanism, which works without prior knowledge of spatial graph.
- USTAN [55] synchronously captures dynamic spatiotemporal correlations with a novel unified spatial-temporal attention network.

Following existing works, Traffic is employed for single-step forecasting and the other three datasets, TaxiBJ, METR-LA and PSME-BAY, are applied for multi-step forecasting. Three typical metrics are employed to comprehensively evaluate the prediction accuracy, i.e., Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). Given prediction $\hat{X} \in \mathbb{R}^{N \times T}$ and corresponding label $X \in \mathbb{R}^{N \times T}$, the metrics can be formulated as follows,

Mean Absolute Error (MAE)

$$\mathrm{MAE}(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{TN} \sum_{i=1}^{T} \sum_{j=1}^{N} |\mathbf{X}_{ij} - \hat{\mathbf{X}}_{ij}| \tag{21}$$

Root Mean Square Error (RMSE)

$$\mathrm{RMSE}(\mathbf{X}, \hat{\mathbf{X}}) = \sqrt{\frac{1}{TN} \sum_{i=1}^{T} \sum_{j=1}^{N} (\mathbf{X}_{ij} - \hat{\mathbf{X}}_{ij})^2} \tag{22}$$

Mean Absolute Percentage Error (MAPE)

$$\mathrm{MAPE}(\mathbf{X}, \hat{\mathbf{X}}) = \frac{1}{TN} \sum_{i=1}^{T} \sum_{j=1}^{N} \left| \frac{\mathbf{X}_{ij} - \hat{\mathbf{X}}_{ij}}{\mathbf{X}_{ij}} \right| \tag{23}$$

[2][Online]. Available: https://archive.ics.uci.edu/ml/datasets/PEMS-SF

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on September 18,2024 at 02:53:10 UTC from IEEE Xplore. Restrictions apply.

TABLE II
FORECASTING PERFORMANCE COMPARISON ON MERE-LA, PEMS-BAY AND TAXIBJ DATASETS ON METRICS MAE, RMSE AND MAPE

Models	Datasets	METR-LA			PEMS-BAY			TaxiBJ		
	Metrics	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
DCRNN		3.60	7.60	10.50	2.07	4.74	4.90	22.51	47.88	26.39
STGCN		4.59	9.40	12.70	2.49	5.69	5.79	23.67	49.21	27.51
Graph WaveNet		3.53	7.37	10.01	1.95	4.52	4.63	22.39	48.21	26.83
GMAN		3.40	7.21	9.72	1.86	4.32	4.31	18.35	43.58	25.86
AGCRN		3.89	9.47	10.39	1.97	4.34	4.58	19.52	44.75	25.85
GWNET		3.53	7.27	9.85	1.91	4.40	4.47	19.64	45.28	24.73
FC-GAGA		3.49	7.33	10.24	1.93	4.40	4.48	18.38	43.61	25.91
USTAN		3.41	7.16	9.81	1.83	4.27	4.18	17.96	43.10	23.76
AGALG		**3.35**	**7.03**	**9.31**	**1.74**	**4.13**	**4.12**	**17.61**	**42.16**	**23.05**

The best results are in bold and the second-best results are underlined.

C. Settings

For single-step forecasting, we split the Traffic datasets into training set, validation set and testing set in chronological order according to the ratio of 6:2:2. Given 48 historical data, all the baselines and the proposed model are required to predict the traffic state at one target future step with horizon as 3, 6, 9 and 12, respectively, i.e., using 48-step input to predict single-step target.

For multi-step forecasting, TaxiBJ, METR-LA and PEMS-BAY are also divided into training set, validation set and testing set in chronological order but with different ratio as 7:1:2 as in [53]. All the baselines and the proposed model are requires to forecast the following 12-step status of all variable, i.e., using 12-step input to predict 12-step target.

All the dataset are normalized with min-max normalization, and the model is trained by Adam optimizer with initial learning rate as 0.001 and batch size as 64 for 100 epochs. λ in (20) is set to 100. Both Encoder and Decoder of AGA contains 4 adaptive graph convolutional layers in the final implementation. Zero initialization is applied on pool of mean vectors U and pool of covariance matrices S is randomly initialized. Zero initialization of covariance matrices is non-sense, which would lead to zero gradients of S. And in our tuning process, we find zero initialization of U results in faster convergence of U. We implement the model with PyTorch1.9 and train it on a Tesla V100 16 GB GPU. All results of baselines are either reproduced according official public code with default settings or cited from existing works.

D. Main Result

Tables II and III provide the prediction performance comparison between baselines and our proposed model AGALG on the four datasets in single-step setting and multi-step setting, respectively. As demonstrated, the AGALG achieves state-of-the-art results on all metrics on all the datasets.

Single-step forecasting: In the single-step experiments, the AGALG outperforms all the baselines on all the three metrics at all the target horizons. Compared to the latest and best performing baseline CATN, the proposed AGALG outperforms it with 6.26%, 10.09% and 6.75% average improvements on MAE, RMSE and MAPE over all horizons on the Traffic dataset. When the target horizon is 9, AGALG outperforms CATN by the most with improvements on MAE, RMSE and MAPE as 10.36%, 12.23% and 9.99%. Such result proves the superiority of AGALG on single-step traffic forecasting. In later section, we will further explore the effect of the size of historical window, i.e., the input length, to evaluate the scalability of the proposed AGALG.

TABLE III
FORECASTING PERFORMANCE COMPARISON ON TRAFFIC DATASET

Models	Metrics	traffic			
		3	6	9	12
VAR	MAE	0.0252	0.0481	0.0642	0.1020
	RMSE	0.0374	0.0679	0.0982	0.1939
	MAPE	0.5658	1.3020	2.2563	2.5818
LSTNet	MAE	0.0193	0.0198	0.0225	0.0229
	RMSE	0.0847	0.0481	0.0509	0.0517
	MAPE	0.1831	0.1897	0.2370	0.2455
TPA-LSTM	MAE	0.0187	0.0192	0.0223	0.0237
	RMSE	0.0310	0.0315	0.0398	0.0498
	MAPE	0.4681	0.7099	0.4813	0.5671
MTNet	MAE	0.3086	0.3108	0.3071	0.3128
	RMSE	0.4120	0.4154	0.4112	0.4225
	MAPE	1.6268	1.6544	1.7265	1.5982
DSANet	MAE	0.0227	0.0203	0.0235	0.0252
	RMSE	0.0322	0.0341	0.0412	0.0475
	MAPE	0.1810	0.1640	0.1910	0.1910
MTGNN	MAE	0.0242	0.0467	0.0289	0.0300
	RMSE	0.0523	0.0789	0.0587	0.0605
	MAPE	0.2307	0.6161	0.2891	0.3231
CATN	MAE	0.0180	0.0183	0.0193	0.0178
	RMSE	0.0309	0.0312	0.0319	**0.0308**
	MAPE	0.1560	0.1589	0.1682	**0.1550**
AGALG	MAE	**0.0168**	**0.0172**	**0.0180**	**0.0183**
	RMSE	**0.0264**	**0.0277**	**0.0280**	0.0309
	MAPE	**0.1471**	**0.1497**	**0.1514**	0.1568

The best results are in bold and the second-best results are underlined.

Multi-step forecasting: Table II shows the average prediction accuracy according to the three metrics cross all 12 future time steps. As demonstrated, AGALG achieves state-of-the-art prediction performance on multi-step forecasting on all the datasets, i.e., TaxiBJ, METR-LA and PEMS-BAY. Compared to the best method of all baselines, AGALG achieves {1.47%, 1.82%, 5.10%}, {4.92%, 3.28%, 1.44%} and {1.95%, 2.18%, 2.99%} improvements of on TaxiBJ, METR-LA and PEMS-BAY with respect to MAE, RMSE and MAPE, respectively.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on September 18,2024 at 02:53:10 UTC from IEEE Xplore. Restrictions apply.

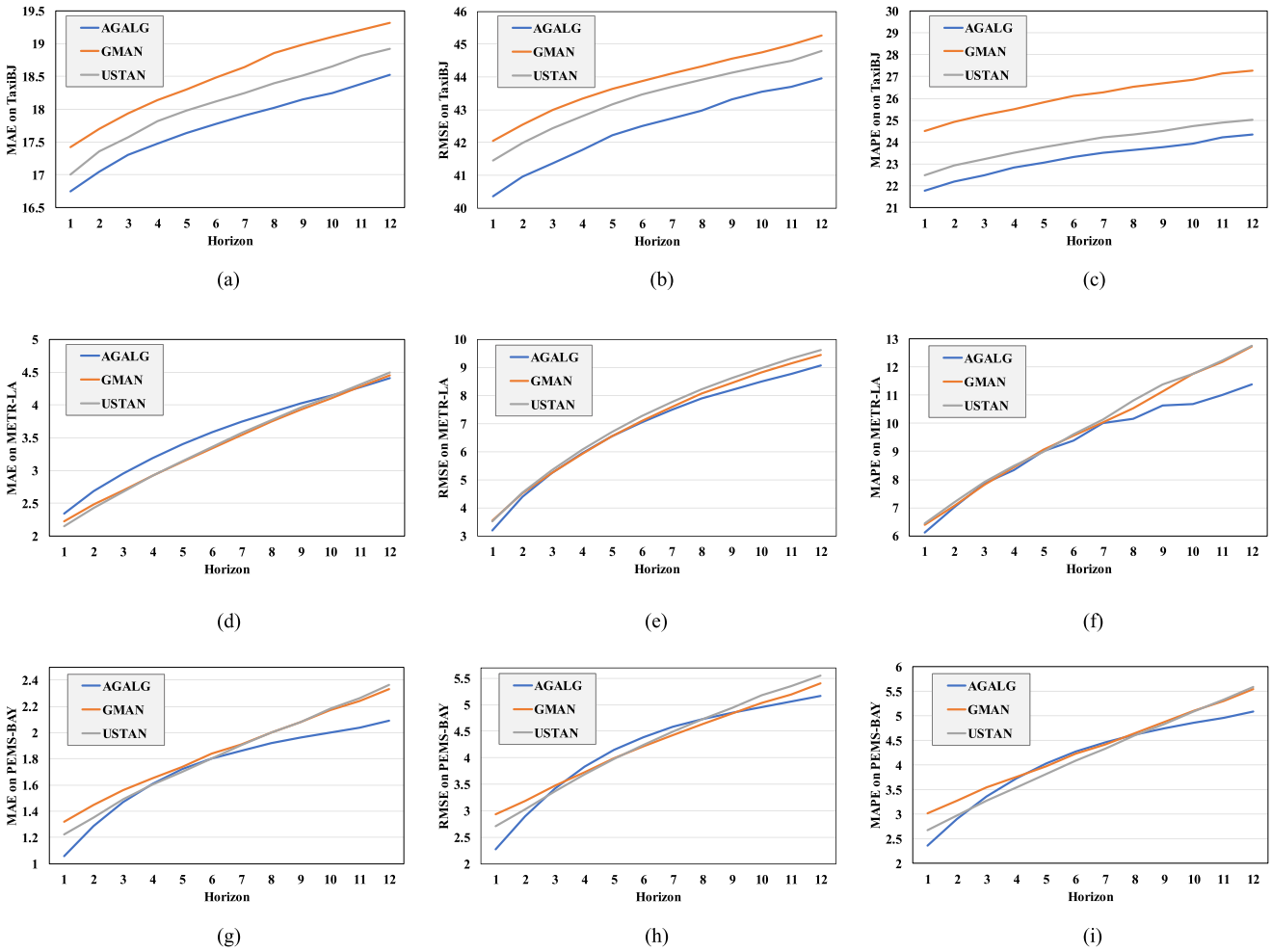

Fig. 5. Prediction performance comparison at each horizon on TaxiBJ, METR-LA and PEMS-BAY.

To further evaluate the finer grained prediction performance of AGALG, we compare the three metrics of AGALG on all the three multi-step datasets at each target time horizon with those of GMAN and USTAN, as they perform the best among all baselines. As shown in Fig. 5, AGALG achieves better performance on at almost all the horizons. Notably, we can find AGALG performs better at larger horizons than the other two baselines. This result further validates the superiority and robustness of the proposed AGALG on multi-step forecasting.

E. Ablation Study

We propose four variants of AGALG to evaluate the effect of different components of AGALG, which are listed as follows,

- *w/o static graph.* GFM module fuses two kind of graphs, static graph and dynamic graph. In this variant, we remove static graph from GFM.
- *w/o dynamic graph.* Similarly, in this variant, we remove dynamic graph from GFM.
- *w/o Gaussian.* In this variant, we replace LGG and GPR in AGALG with LSTM. This variant is designed to evaluate the effect of using Gaussian processes to model the latent representations of nodes.
- *w/o graph.* In this variant, we replace the graph convolutions in AGALG with fully connected layers, which leads to none spatial aggregation among nodes.

We evaluate the multi-step prediction performance of the variants on TaxiBJ, METR-LA and PEMS-BAY. As show in Fig. 6, when graph convolution removed, the performance decreases the most, which proves the importance of spatial correlations on traffic prediction. The huge performance loss of *w/o Gaussian* demonstrates the superiority of the proposed LGG on modeling temporal correlations than LSTM. Also, we can find both static graph and dynamic graph have contribution to the forecasting performance of AGALG.

F. Analysis

Size of Historical Window: To explore the scalability of AGALG on single-step traffic forecasting, we evaluate the effect of the size of historical window by extracting a series of experiments on Traffic with sizes of historical window as 6, 12, 24, 48, respectively, and the target horizon is kept as 3.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on September 18,2024 at 02:53:10 UTC from IEEE Xplore. Restrictions apply.

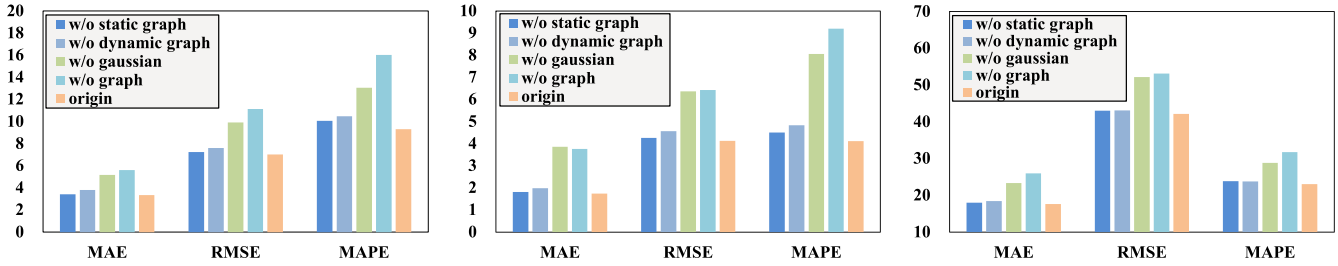

Fig. 6. Prediction performance comparison of different variants and origin AGALG on TaxiBJ, METR-LA and PEMS-BAY. (a) METR-LA. (b) PEMS-BAY. (c) TaxiBJ.

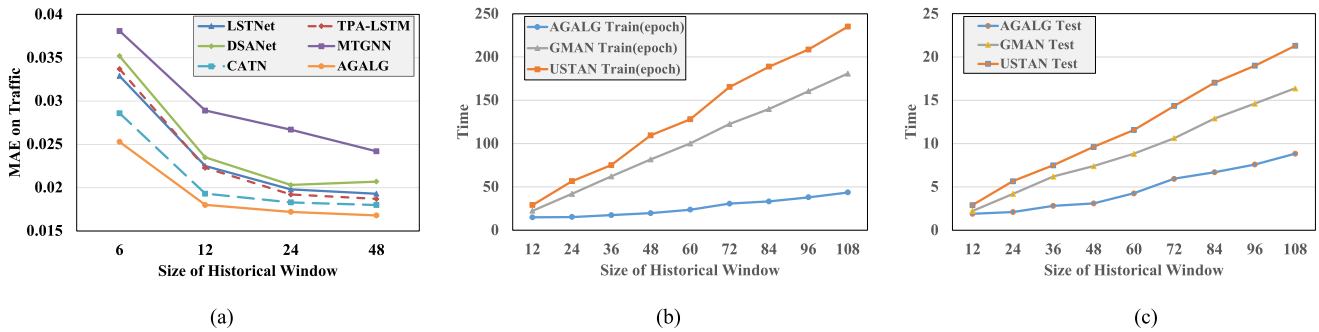

Fig. 7. MAE on Traffic and time consumption with different sizes of the historical window.

MAE is employed to evaluate the prediction accuracy. As shown in Fig. 7(a), the proposed AGALG achieves better MAE than all the baselines with respect to different sizes of historical window. Also, when the size of historical window decreases, the prediction accuracy of AGALG decreases slower than the baselines. Such result validates the effectiveness of using Gaussian processes to model traffic state. Taking advantages of Gaussian processes, AGALG is able to handle the scenario that little historical data is accessible for making prediction, which is a nice property for deployment.

We further compare the time consumption of our model under different sizes of historical window with time consumption of best performed multi-step forecasting models, i.e., GMAN and USTAN. The result is shown in Fig. 7(b) and (c). Compared with GMAN and USTAN, when the size of historical window increases, the time consumption of our model increases much slower than that of baselines. The result indicates the superiority of our model on handling long input sequences.

Size of Pool of LGG: As mentioned in Section IV-C, LLG maintains a pool of mean vectors and covariance vectors, which is used to construct joint Gaussian distributions. The size of pool determines the breadth of searching space of Gaussian distributions that LLG can explore. In this section, we conduct a series of experiments on METR-LA with different sizes of pool as $\{1, 50, 100, 150, 200, 250, 300, 350, 400\}$ and record the average MAE, RMSE and MAPE over 12 horizons. As shown in Fig. 8, although larger pool means larger searching space for Gaussian distributions, the prediction accuracy does not always get better with the size of the pool, which holds for all three

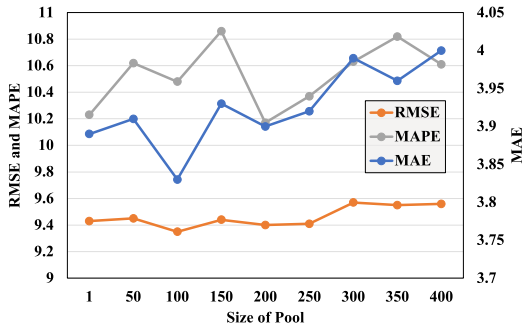

Fig. 8. MAE, RMSE and MAPE on METR-LA with different sizes of pool.

metrics. Based on the metrics on METR-LA and our experience on tuning the parameters, half the number of variables in the datasets is a suitable size of pool.

G. Prediction Visualization

Fig. 9 displays some prediction visualizations on the testing set of PEMS-BAY to more intuitively show the prediction accuracy of the proposed AGALG. We randomly select four nodes from PEMS-BAY and the first 1000 time steps in the testing set of PEMS-BAY are selected for evaluation. As shown, AGALG is able to fit the general trend of traffic state very well, even at peaks and valleys. However, we can observe that in some extremely hard scenarios, where the traffic state changes dramatically, AGALG fails to make that accuracy predictions. In the future work, we will work on handling these extremely hard scenarios.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on September 18,2024 at 02:53:10 UTC from IEEE Xplore. Restrictions apply.

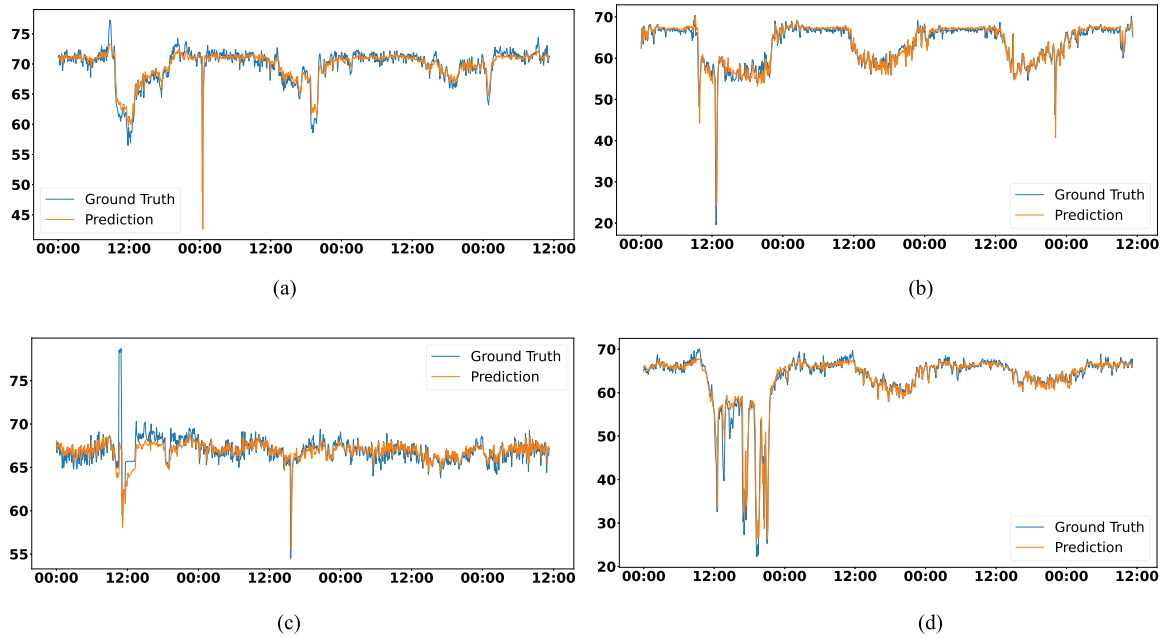

Fig. 9. Prediction visualization on PEMS-BAY.

VI. Conclusion

In this article, we propose an adaptive graph convolutional autoencoder equipped with latent Gaussian process AGALG to tackle the traffic prediction task. AGALG consists of a graph fusion module (GFM), an adaptive graph convolutional autoencoder (AGA), a latent Gaussian generator (LGG) and a Gaussian processes regressor (GPR). GFM fuses two kinds of graph for modeling both static and dynamic spatial correlations in traffic networks. The fused graph is used as the input of AGA, which stacks adaptive graph convolution layers to decouple the dependencies among variables in encoder and to recouple them in decoder. LGG generates joint Gaussian distributions of historical data and future estimation, which constrains the latent space of latent code of AGA. GPR finally generates prediction based on the joint Gaussian distribution produced by LGG. Experimental results on several real-world datasets validates the effectiveness of the proposed AGALG in terms of traffic forecasting.

References

[1] Y. Xun, J. Liu, N. Kato, Y. Fang, and Y. Zhang, "Automobile driver fingerprinting: A new machine learning based authentication scheme," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1417–1426, Feb. 2020.

[2] H. Guo, J. Li, J. Liu, N. Tian, and N. Kato, "A survey on space-air-ground-sea integrated network security in 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 53–87, First Quarter 2022.

[3] J. Li et al., "A traffic prediction enabled double rewarded value iteration network for route planning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4170–4181, May 2019.

[4] P. Chu, J. A. Zhang, X. Wang, G. Fang, and D. Wang, "Semi-persistent resource allocation based on traffic prediction for vehicular communications," *IEEE Trans. Intell. Veh.*, vol. 5, no. 2, pp. 345–355, Jun. 2020.

[5] T. Mai, B. Ghosh, and S. Wilson, "Short-term traffic-flow forecasting with auto-regressive moving average models," in *Proc. Inst. Civil Engineers-Transport*, vol. 167, no. 4, pp. 232–239, 2014.

[6] E. Zivot and J. Wang, "Vector autoregressive models for multivariate time series," in *Modeling Financial Time Series With S-PLUS*, Berlin, Germany: Springer, 2006, pp. 385–429.

[7] R. Chen, C.-Y. Liang, W.-C. Hong, and D.-X. Gu, "Forecasting holiday daily tourist flow based on seasonal support vector regression with adaptive genetic algorithm," *Appl. Soft Comput.*, vol. 26, pp. 435–443, 2015.

[8] U. Johansson, H. Boström, T. Löfström, and H. Linusson, "Regression conformal prediction with random forests," *Mach. Learn.*, vol. 97, pp. 155–176, 2014.

[9] D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. M. Choudhury, and A. K. Qin, "A survey on modern deep neural network for traffic prediction: Trends, methods and challenges," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 4, pp. 1544–1561, Apr. 2022.

[10] X. Shi, 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. Adv. Neural Inf. Process. Syst.*, 2015, pp. 802–810.

[11] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2016, pp. 1655–1661.

[12] J. Ye, J. Zhao, K. Ye, and C. Xu, "How to build a graph-based deep learning architecture in traffic domain: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 3904–3924, May 2022.

[13] Y. Xiao, Z. Xing, A. X. Liu, L. Bai, Q. Pei, and L. Yao, "Cure-GNN: A robust curvature-enhanced graph neural network against adversarial attacks," *IEEE Trans. Dependable Secure Comput.*, no. 1, pp. 1–16, 2022.

[14] 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.

[15] 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. 2020.

[16] 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.

[17] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 17804–17815.

[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. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 11, pp. 5415–5428, Nov. 2022.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on September 18,2024 at 02:53:10 UTC from IEEE Xplore. Restrictions apply.

[20] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 11121–11128.
[21] 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. Discov. Data Mining*, 2020, pp. 753–763.
[22] J. Liu, H. Guo, J. Xiong, N. Kato, J. Zhang, and Y. Zhang, "Smart and resilient EV charging in SDN-enhanced vehicular edge computing networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 1, pp. 217–228, Jan. 2020.
[23] J. Zhang, H. Guo, J. Liu, and Y. Zhang, "Task offloading in vehicular edge computing networks: A load-balancing solution," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 2092–2104, Feb. 2020.
[24] J. Liu, S. Zhang, W. Sun, and Y. Shi, "In-vehicle network attacks and countermeasures: Challenges and future directions," *IEEE Netw.*, vol. 31, no. 5, pp. 50–58, 2017.
[25] Y. Zhang, Y. Yang, W. Zhou, H. Wang, and X. Ouyang, "Multi-city traffic flow forecasting via multi-task learning," *Appl. Intell.*, vol. 51, no. 10, pp. 6895–6913, 2021. [Online]. Available: https://doi.org/10.1007/s10489-020-02074-8
[26] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*. Hoboken, NJ, USA: Wiley, 2015.
[27] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep learning: A generic approach for extreme condition traffic forecasting," in *Proc. SIAM Int. Conf. Data Mining*, SIAM, 2017, pp. 777–785.
[28] N. Laptev, J. Yosinski, L. E. Li, and S. Smyl, "Time-series extreme event forecasting with neural networks at Uber," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1–5.
[29] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[30] 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.*, 2020, pp. 1177–1185.
[31] J. Bai et al., "A3T-GCN: Attention temporal graph convolutional network for traffic forecasting," *ISPRS Int. J. Geo-Inf.*, vol. 10, no. 7, p. 485, 2021.
[32] 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 Conf. Artif. Intell.*, 2019, pp. 922–929.
[33] M. Xu et al., "Spatial-temporal transformer networks for traffic flow forecasting," 2020, *arXiv:2001.02908*.
[34] X. Wang et al., "Traffic flow prediction via spatial temporal graph neural network," in *Proc. Web Conf.*, 2020, pp. 1082–1092.
[35] W. C. KI et al., "Gaussian processes for machine learning," *Int. J. Neural Syst.*, vol. 14, pp. 69–106, 2006.
[36] C. E. Rasmussen, "Gaussian processes in machine learning," in *Summer School on Machine Learning*, Berlin, Germany: Springer, 2003, pp. 63–71.
[37] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain, "Gaussian processes for time-series modelling," *Philos. Trans. Roy. Soc. A: Math. Phys. Eng. Sci.*, vol. 371, no. 1984, 2013, Art. no. 20110550.
[38] R. Neal, "Bayesian learning for neural networks," in *Lecture Notes Statist.*, Berlin, Germany: Springer, 1996.
[39] M. Garnelo et al., "Neural processes," 2018, *arXiv:1807.01622*.
[40] M. Garnelo et al., "Conditional neural processes," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1704–1713.
[41] X. Wang, L. Yao, X. Wang, H.-Y. Paik, and S. Wang, "Global convolutional neural processes," in *Proc. IEEE Int. Conf. Data Mining*, 2021, pp. 699–708.
[42] B. Yoo, J. Lee, J. Ju, S. Chung, S. Kim, and J. Choi, "Conditional temporal neural processes with covariance loss," in *Proc. 38th Int. Conf. Mach. Learn.*,, 2021, pp. 12051–12061.
[43] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.
[44] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 4189–4196.
[45] J. Choi, H. Choi, J. Hwang, and N. Park, "Graph neural controlled differential equations for traffic forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 6367–6374.
[46] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016. [Online]. Available: https://arxiv.org/abs/1609.02907
[47] H. V. Jagadish et al., "Big data and its technical challenges," *Commun. ACM*, vol. 57, no. 7, pp. 86–94, 2014.
[48] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, "Freeway performance measurement system: Mining loop detector data," *Transp. Res. Rec.*, vol. 1748, no. 1, pp. 96–102, 2001.
[49] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long- and short-term temporal patterns with deep neural networks," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, New York, NY, USA, 2018, pp. 95–104. [Online]. Available: https://doi.org/10.1145/3209978.3210006
[50] Y.-Y. Chang, F.-Y. Sun, Y.-H. Wu, and S.-D. Lin, "A memory-network based solution for multivariate time-series forecasting," 2018. [Online]. Available: https://arxiv.org/abs/1809.02105
[51] S. Huang, D. Wang, X. Wu, and A. Tang, "DSANet: Dual self-attention network for multivariate time series forecasting," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, 2019, pp. 2129–2132. [Online]. Available: https://doi.org/10.1145/3357384.3358132
[52] H. He, Q. Zhang, S. Bai, K. Yi, and Z. Niu, "CATN: Cross attentive tree-aware network for multivariate time series forecasting," in *Proc. 36th AAAI Conf. Artif. Intell., 34th Conf. Innov. Appl. Artif. Intell., 12th Symp. Educ. Adv. Artif. Intell.*, 2022, pp. 4030–4038. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/20320
[53] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 1234–1241. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/5477
[54] B. N. Oreshkin, A. Amini, L. Coyle, and M. Coates, "FC-GAGA: Fully connected gated graph architecture for spatio-temporal traffic forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 9233–9241.
[55] W. Long et al., "Unified spatial-temporal neighbor attention network for dynamic traffic prediction," *IEEE Trans. Veh. Technol.*, vol. 72, no. 2, pp. 1515–1529, Feb. 2023.

Xu Wang received the bachelor's degree in automation from North Eastern University, Boston, MA, USA, in 2017. He is currently working toward the Ph.D. degree with the School of Data Science, University of Science and Technology of China, Hefei, China. His research interests mainly include data mining, machine learning and spatial-temporal learning.

Pengkun Wang (Student Member, IEEE) received the bachelor's degree from Jilin University, Changchun, China, in 2017. He is currently working toward the Ph.D. degree with the School of Data Science, University of Science and Technology of China. His research interests mainly include data mining, multi-modal fusion, and computer vision.

Binwu Wang is currently working toward the Ph.D. degree with the School of Data Science, University of Science and Technology of China, Hefei, China. His research interests mainly include data mining and machine learning.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on September 18,2024 at 02:53:10 UTC from IEEE Xplore. Restrictions apply.

Yudong Zhang (Student Member, IEEE) received the bachelor's degree in 2020 from the University of Electronic Science and Technology of China, Hefei, China, where he is currently working toward the Ph.D. degree with the School of Data Science. His research interests include machine learning and data mining, especially their applications in urban computing.

Zhengyang Zhou (Student Member, IEEE) is currently working toward the Doctoral degree with the School of Computer Science and Technology, University of Science and Technology of China. His research interests include machine learning, spatiotemporal data mining and artificial intelligence in traffic applications. He is the Student Member of AAAI.

Lei Bai is currently a PostdoctoralResearch Fellow with the School of Electrical and Information Engineering, University of Sydney, Camperdown NSW, Australia. His research interests include machine learning, spatial-temporal learning, and their applications. He was the recipient of the 2020 Google Ph.D. Fellowship, 2020 UNSW Engineering Excellence Award, and 2021 Dean's Award for Outstanding Ph.D. Theses.

Yang Wang (Senior Member, IEEE) received the Ph.D. degree from the University of Science and Technology of China (USTC), Hefei, China, in 2007, under supervision of Professor Liusheng Huang. He is currently an Associate Professor with USTC. His research interests mainly include wireless networks, distributed systems, data mining, and machine learning.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on September 18,2024 at 02:53:10 UTC from IEEE Xplore. Restrictions apply.