Multi-view Bayesian spatio-temporal graph neural networks for reliable traffic flow prediction

Jiangnan Xia[1] · **Senzhang Wang**[1] · **Xiang Wang**[2] · **Min Xia**[3] · **Kun Xie**[4] · **Jiannong Cao**[5]

Abstract

Accurate traffic flow prediction is critically essential to transportation safety and Intelligent Transportation Systems (ITS). Existing approaches generally assume the traffic data are complete and reliable. However, in real scenarios, the traffic data are usually sparse and noisy due to the unreliability of the road sensors. Meanwhile, the global semantic traffic correlations among the road links over the road network are largely ignored by existing works. To address these issues, in this paper we study the novel problem of reliable traffic prediction with noisy and sparse traffic data and propose a Multi-View Bayesian Spatio-Temporal Graph Neural Network (MVB-STNet for short) to effectively address it. Specifically, we first construct the traffic flow graphs from two views, the structural traffic graph based on the topological closeness of the road sensors, and the semantic traffic graph which is constructed based on the traffic flow correlations among all the road sensors. Then the features of the two views are learned simultaneously to more broadly capture the spatial correlations. Inspired by the effectiveness of Bayesian neural networks in handling data uncertainty, we design the Bayesian Spatio-Temporal Long Short-Term Memory Net layer to more effectively learn the spatio-temporal features from the sparse and noisy traffic data. Extensive evaluations are conducted over two real traffic datasets. The results show that our proposal significantly improves current state-of-the-arts in terms of traffic flow prediction with sparse and noisy data.

Keywords Traffic prediction · Data uncertainty · Bayesian graph neural network

1 Introduction

With the fast urbanization of many countries, the number of vehicles increases rapidly in the past several decades, leading to a significant increase in various transportation-related issues such as traffic accidents and congestion. According to the statistics of the World Health Organization, traffic accidents have become one significant public safety issue for many countries [1, 2]. Accurate prediction of urban traffic flows is vitally important to reduce traffic accidents by helping drivers avoid congested roads and better plan their travel routes in advance [3, 4]. Thus accurate traffic prediction has become crucial in reducing the huge harm and economical losses caused by traffic accidents and congestion by supporting the government and policymakers to adopt effective traffic control strategies. However, due to the complex spatio-temporal dependencies of the traffic data and the intrinsic uncertainties of the transportation systems, it is challenging to make an accurate and reliable prediction of urban traffic flow.

As a critical functionality of Intelligent transportation systems (ITS), traffic flow prediction has been extensively studied by the research communities of both computer science and transportation engineering in recent years. Traditionally, statistics-based approaches are widely used for road segment level traffic prediction, such as VAR [5], ARIMA and its variants [6, 7]. Statistics-based approaches generally predict the future traffic trends of a single road link or

✉ Senzhang Wang
szwang@csu.edu.cn

✉ Xiang Wang

[1] School of Computer Science and Engineering, Central South University, Changsha 410083, Hunan, China

[2] College of Meteorology and Oceanology, National University of Defense Technology, Changsha 410073, Hunan, China

[3] Department of Engineering, Lancaster University, Lancaster LA1 4YW, State, UK

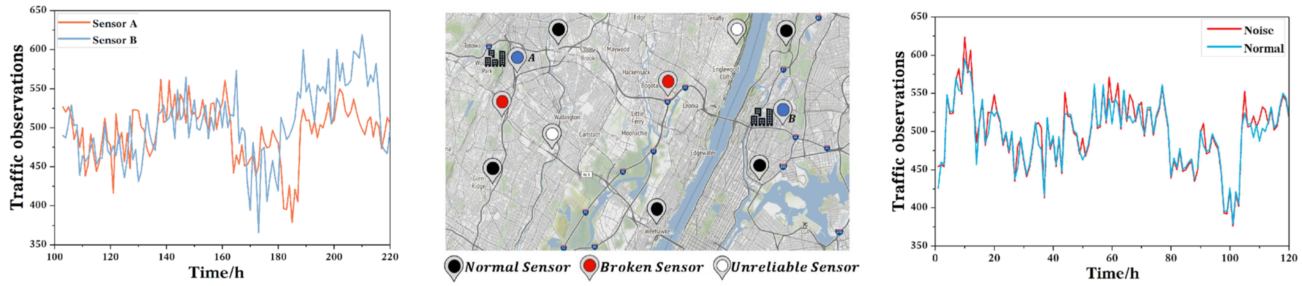
[4] College of Computer Science and Electronic Engineering, Hunan University, Changsha 410012, Hunan, China

[5] Department of Computing, The Hong Kong Polytechnic University, HongKong, China

segment by a linear projection model learned from the historical traffic data. Due to the limited feature learning ability, the performance of statistics-based approaches is usually undesirable due to the randomness and the non-linearity of the traffic flows. Recently, deep learning models such as CNN and RNN have been widely applied to various traffic prediction tasks due to their powerful spatio-temporal feature learning ability [3, 8–10]. Deep learning-based methods such as CNN and GCN are especially effective in predicting the traffic flows over a large road network by capturing the complex spatial correlations among the road links [8, 10, 11]. For example, DCRNN [12] integrated diffusion convolution and sequence-to-sequence structure for traffic flow prediction. STGCN [13] employed ChebNet graph convolution and 1D convolution to predict the traffic flow of all the road segments in the road network. ASTGCN [14] used two attention layers to capture the dynamic correlation of the traffic network in spatial and time dimensions respectively. GMAN [15] utilized an attention mechanism to extract spatial and temporal features more effectively for road network level traffic prediction.

Although considerable research efforts have been made on traffic flow prediction, a major issue of exiting works is that they mostly assume the traffic data are complete and reliable with little noise. However, in real scenarios, the traffic data collected by road sensors are usually sparse and full of noise due to the unreliability of the road sensors. In ITS, the traffic data (e.g. traffic flow or speed) are continuously collected by the road sensors deployed at different locations of the road network. The sensors may fail to work normally and produce wrong or noisy data from time to time due to the long usage time as shown in Fig. 1b, causing unreliable prediction results. As shown in Fig. 1c, unreliable sensors that are used for a long time can produce unreliable data that deviates from the real observations. Another obvious limitation of existing works is that the global semantic correlations of the traffic data over a road network are not fully considered and explored. Existing deep learning approaches such as CNN and GCN mostly capture the local spatial correlation between a road sensor and its neighbor sensors [16], which follows the spatial smoothness ("*near things are more related than distant things*") [17]. However, recent studies have shown that the global semantic correlations are also essential and should not be ignored in human mobility analysis in urban areas [18]. For example, two residential areas (e.g. regions A and B in Fig. 1a) may present very similar traffic patterns as shown in Fig. 1a, although the two areas may be far away from each other. Therefore, how to simultaneously capture the local and global spatial dependencies of traffic data as well as integrate them together to achieve a more accurate and reliable prediction result remains an open and challenging research problem.

To address the above issues, in this paper we propose a Multi-View Bayesian Spatio-Temporal Graph Neural Network model (MVB-STNet for short) to effectively deal with the data uncertainty issue and capture the complex spatio-temporal data dependencies for a more reliable traffic prediction. To more comprehensively capture the spatial correlations of the data, we construct the graphs of two views from the raw traffic sensor data: the local structural view traffic graph and the semantic view traffic graph. The local view graph reflects the spatial connectivity among the sensors, while the semantic view graph is constructed based on the inter-dependencies among the sensor readings (e.g. traffic flow or speed) to reflect their global semantic correlations. Specifically, the semantic view can break through the restriction of geographical distance and learn the potentially similar traffic patterns among multiple roads from a global perspective. We adapt structural view based on topological distance and semantic view based on similar traffic patterns. We use the structural view to learn the spatial correlations between adjacent roads from a local perspective. In addition, we use the semantic view to further learn the dependencies between all roads with similar traffic patterns from a global perspective, overcoming geographical limitations. Such a multi-view learning method can comprehensively capture the complex spatial relationships from the road network.



(a) Traffic patterns of regions A and B **(b)** Traffic sensors deployed in a road network **(c)** Noise data vs normal data

Fig. 1 Illustration of traffic data uncertainty due to unreliable sensors

Bayesian neural networks (BNN) are currently an effective model to handle data uncertainty by setting a probability distribution for the learnable model parameters and regarding the change of model parameters as uncertainty. Inspired by BNN, we propose to integrate the Bayesian model into our model and design a Bayesian Spatio-Temporal Long Short-Term Memory Net (BSTLSN) layer to address the data uncertainty issue for more robustly learning features. Based on the designed BSTLSN layers, an encoder-decoder framework is proposed for traffic sequence data prediction over a road sensor network.

Our major contributions are summarized as follows.

- We for the first time study the uncertainty issue of data acquisition (e.g. noise data or missing data due to sensor failure) in road-network level traffic flow prediction. To effectively address it, a multi-view bayesian spatio-temporal neural network model MVB-STNet is proposed.
- Structural traffic graph and semantic traffic graph are constructed to more broadly capture the spatial correlations of the traffic data under a multi-view feature learning framework. Bayesian model is also integrated into the Spatio-Temporal Long Short-Term Memory Net layers to achieve more reliable prediction results.
- Extensive experiments are conducted on two real traffic flow datasets. The results show that MVB-STNet significantly improves the prediction performance compared with state-of-the-art methods when the traffic data are incomplete and noisy.

The remainder of this paper is organized as follows. Section 2 will review related works. Section 3 will give some important notations and a formal problem definition. Section 4 will show the model framework and introduce the model in detail. Evaluations are given in Sect. 5. Finally, the paper is concluded in Sect. 6.

2 Related work

This work is highly relevant to the topics of traffic prediction and bayesian neural networks. In this section, we will review related works from the two aspects.

2.1 Traffic flow prediction

Generally, traditional traffic flow prediction approaches can be categorized into classical statistics-based methods and machine learning-based methods. Statistics-based traffic prediction models include Autoregressive Integrated Moving Average model (ARIMA) [7], Vector Auto-Regressive (VAR) [19] and their variants. These methods usually require the assumption of data stationarity. However, the traffic data usually present complex spatial-temporal characteristics, and thus the data stationarity assumption may not hold. Statistics-based methods are mostly used for predicting traffic conditions on a single road or over a small road network. They are difficult to capture the highly non-linear spatial-temporal correlations of traffic data over a large traffic network. Machine learning methods such as support vector regression (SVR) [20], random forest regression (RFR) [21] and hidden Markov models [22] are more effective to capture the traffic patterns from a large number of historical traffic data. Although these methods usually perform better than statistics-based methods via capturing non-linear spatio-temporal correlations, their performance is still less promising when working on a large road network with hundreds or even thousands of road links [22].

With the great success of deep learning techniques in the fields of computer vision and natural language processing, considerable attempts have been made to adopt deep learning techniques for traffic flow prediction. A line of studies applied CNN to learn the spatial dependence of road networks by treating the traffic data of a city as two-dimensional images. In this way, spatial correlation among regions can be effectively captured to boost the performance of city-wide traffic prediction. Zhang et al. [23] proposed ST-ResNet, which transformed the traffic flow data of the entire city into images to predict the in and out-flows of each cell region in a city. Yao et al. [24] presented a Spatio-Temporal Dynamic Network (STDN) based on CNN and RNN, which can simultaneously capture temporal and spatial correlation of a road network for traffic prediction. Lin et al. [25] proposed DeepSTN+ model, using point-of-interest (POI) data as external information to consider the effect Of location function on crowd/traffic flow. Yao et al. [26] proposed a Deep Multi-view Spatio-Temporal Network (DMVST-NET) to integrate the temporal, spatial, and semantic views for traffic prediction.

However, CNN is not directly applicable to graphic data as it is designed to process the data in Euclidean space. To process graph data, GCN was invented and attracted rising research interest recently due to its effectiveness in learning features on graphs [27, 28]. GCN models can be also utilized for road network-level traffic prediction as the traffic data of a whole road network can be considered as an attributed graph. Li et al. [12] proposed the Diffusion Convolutional Recurrent Neural Network (DCRNN) to model the traffic flow as a diffusion process on a directed road graph, which significantly improved predictive performance. Wu et al. [29] presented a model named Graph WaveNet, which combined graph convolution and dilated casual convolution to capture spatial-temporal correlations. Yu et al. [13] proposed a model STGCN, which applied ChebNet graph convolution and 1D convolution to extract spatial dependencies and temporal correlations. Guo et al. [14] presented the ASTGCN model which improved STGCN by leveraging two attention layers to capture the dynamic correlations of a road network in both spatial and temporal dimensions. Zheng et al. [15] proposed the GMAN model that integrated the attention mechanism with GCN to more effectively extract the spatio-temporal features for traffic flow prediction.

Although considerable research efforts have been made, there is still a lack of studies on reliable traffic prediction when the traffic sensor data are sparse, noisy, and incomplete. The performance may degrade remarkably when the above-discussed models are directly applied to the studied problem. Thus a more reliable and robust traffic flow prediction model is required.

2.2 Bayes neural network

A traditional deep neural network usually cannot be applied to all data distributions, because the learned parameters are fixed. In traditional deep learning models, the weights are always fixed and randomly initialized at the beginning of model training, which makes the model quite sensitive to input data. In this paper, the raw traffic data are collected by sensors and other devices deployed in the road network, and it is inevitable that such devices will fail in some extreme circumstances, such as sensor failure, noise interference, or poor network signal. In this case, if we train the model with such uncertain data, the performance of the model will degrade. These neural networks are unable to capture the uncertainty in the training data, and thus they will make overconfident predictions and affect the generalization ability of the model [30, 31]. To address this issue, Bayesian neural network (BNN) is proposed [32, 33]. BNN is a kind of random neural network, whose weight parameters are random variables rather than fixed values [30]. BNNs assume that the weights of each layer are not fixed but conform to a distribution, and then sample the weights from this distribution for model training, which will make the model more robust. BNN combines probabilistic modeling with neural networks. In the prediction stage, the probabilistic model generates a complete posterior distribution and a probabilistic guarantee for the prediction results. In the parameter space, it can infer the properties and distribution of learnable parameters in neural networks [34].

[31] proposed a Stochastic Gradient Variational Bayes (SGVB) estimator to approximate the intractable posterior, which can be applied to learn almost any generative model with continuous latent variables. Charles et al. [30] presented a backpropagation-compatible algorithm named Bayes by Backprop to learn a probability distribution on the weights for a neural network, which improved generalization in non-linear regression problems via the learned uncertainty in the weights. Kristiadi et al. [35] theoretically analyzed the approximate Gaussian distributions of the weights for ReLU networks and indicated the uncertainty on a ReLU network can be calibrated by Bayesian. Xiao et al. [36] proposed a variational Bayesian inference-based model by incorporating uncertainty into neural network weights to address the domain shift and uncertainty caused by the inaccessibility of target domain data.

Although BNN has been proven to be effective in addressing the data uncertainty issue and has been applied in the areas of computer vision and image processing, how to incorporate BNN into traffic flow prediction models to achieve a more reliable traffic prediction result is still not fully studied.

3 Problem statement

In this section, we will first define some terminologies to help state the studied problem. Then we will give a formal problem definition.

Definition 1 Road sensor graph A road sensor graph is represented as $G = \{V, E\}$, where V is the node set and E is edge set. Each $v_i \in V$ represents a sensor deployed on a road for traffic observations (e.g. traffic volume or speed) collection. Each edge $e_{i,j} \in E$ represents two direct neighbor sensors v_i, v_j connected by a road link.

Definition 2 Traffic sequence data We use x_i^t to denote the traffic observation of node v_i at time t, and the observations in T time slots form a time series $\boldsymbol{x}_i = \{x_i^1, ..., x_i^t, ..., x_i^T\}$. The traffic observations of all the sensors on G in T time slots form the traffic sequence data, which can be denoted as $\{X^1, ..., X^t, ..., X^T\}$.

Note that some sensors in G may fail to work, and thus the corresponding traffic observations are missing. Some sensors used for a rather long time may output noisy data due to their low reliability. Therefore, the traffic sequence data may be full of uncertainty containing sparse, incomplete, and noisy sensor readings.

Definition 3 Structural traffic graph We denote a structural traffic graph at time slot t as $\mathcal{G}_{str}^t$. Its adjacent matrix $A_{str} \in \mathcal{R}^{N\times N}$ is associated with the road sensor graph G, and the node features $F^t \in \mathcal{R}^{N\times K}$ are associated with X^t, where N denotes the number of sensors and K is the dimension of features.

Definition 4 Semantic traffic graph We denote a semantic traffic graph at time slot t as $\mathcal{G}_{sem}^t$. Different from the structural traffic graph, the adjacent matrix A_{sem} of $\mathcal{G}_{sem}^t$ is constructed based on the semantic correlations among the sensors. Note that the semantic correlations among the sensors are hidden and need to be inferred from the historical traffic data.

Based on the above terminology definitions, we formally define the studied problem as follows.

ProblemDefinition 1 Given a road sensor graph G, the structural traffic graphs $\{\mathcal{G}^t_{str}|t = 1, ..., T\}$, the semantic traffic graphs $\{\mathcal{G}^t_{sem}|t = 1, ..., T\}$ and the traffic sequence data $\{X^1, ..., X^t, ..., X^T\}$, our goal is to give a reliable traffic flow prediction $\{Y^{T+1}\}$ over the road sensor graph G in the next time slot, given that $\{X^1, ..., X^t, ..., X^T\}$ is sparse and full of noise.

4 Methodology

In this section, we will first present an overview of the proposed MVB-STNet model framework and then introduce it in detail in the following subsections.

4.1 Model framework

Figure 2 shows the framework of MVB-STNet, which includes four major steps: data preprocessing, spatio-temporal (ST) encoder, spatio-temporal (ST) decoder, and prediction. In the data preprocessing step, to better capture the spatial correlations, we model the raw traffic data as two views of graphs, structural traffic graphs and semantic traffic graphs as given in the previous definitions. In the ST encoder step, we adapt a semantic encoder and a structural encoder to jointly learn the local and global semantic spatial dependencies of the graphs of the two views. The ST encoder contains several Bayesian Spatio-Temporal Long Short-Term Memory Net (BSTLSN) layers, which will be described in detail in Sect. 4.2. The BSTLSN layers are used to learn the spatio-temporal features on the two views, respectively. To deal with the data uncertainty issue, we incorporate Bayesian neural network into the learning model, and further design the BSTLSN layer whose weight parameters follow a specific distribution (e.g., Gaussian Distribution) to improve the generalization and robustness of the model. The BSTLSN layers will be elaborated in Sect. 4.4.

Next, the learned features of the two views are fused and input into the ST decoder. The ST decoder also consists of several BSTLSN layers. The ST decoder will be introduced in Sect. 4.3. Finally, several LSTM layers are stacked to generate the final prediction on the future traffic sequence data. The overall objective function for the traffic prediction will be described in Sect. 4.4.1. Next, we will introduce the four steps in detail in the following subsections.

4.2 Spatio-temporal encoder

In the data preprocessing step, we convert the raw traffic data collected by the sensors into structural and semantic traffic graphs. The structural traffic graph $\mathcal{G}_{str}$ is built based on the road sensor graph reflecting the geographical connectivity among the road sensors. Next, we introduce how to construct the semantic traffic graph $\mathcal{G}_{sem}$. As in Definition 4, the semantic traffic graph $\mathcal{G}_{sem}$ is constructed based on the semantic correlations among the sensors. For example, if v_i and v_j are two road sensors both near commercial areas, although the road links of v_i and v_j are not

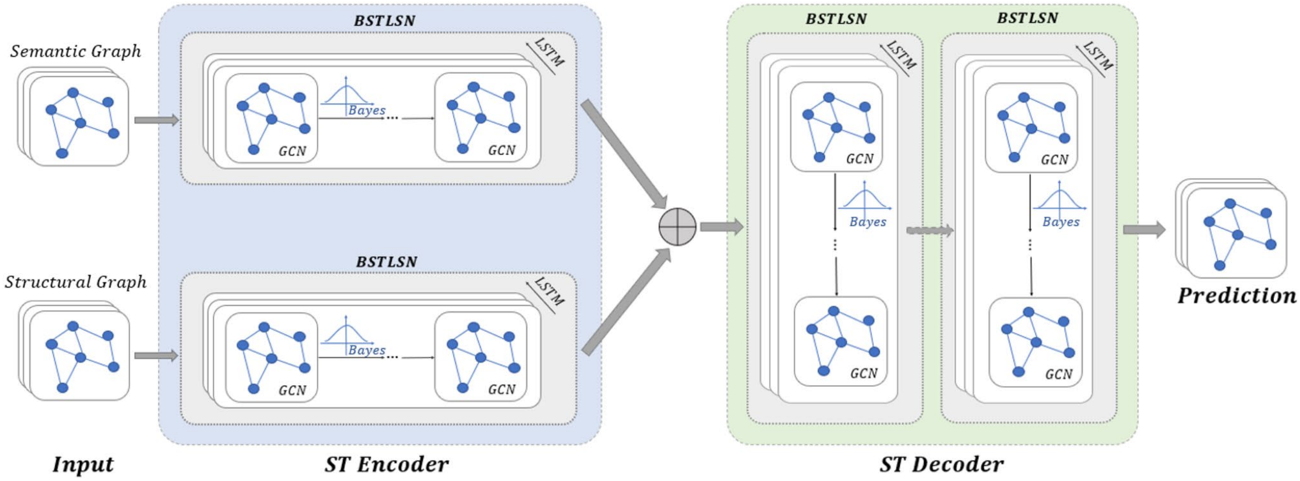


Fig. 2 Framework of proposed MVB-STNet model, which contains four parts: Input, ST Encoder, ST Decoder, and Prediction. First, we model the raw traffic data as two views of graphs. Then, we adopt a semantic encoder and a structural encoder to jointly learn the local and global spatial dependencies of the two views. Next, we fused the learned features of the two views and input them into the ST decoder. Finally, the final prediction on the future traffic sequence data is generated

directly connected and far away from each other, they still may present much similar traffic flow patterns. Therefore, we establish a semantic traffic graph $\mathcal{G}_{sem} = \{V, E_{sem}\}$. The node set V contains road sensors, and the edges E_{sem} represent the semantic similarity between each pair of nodes. We first randomly initialize a learnable node embedding for all the nodes $A_{sem} \in \mathcal{R}^{N \times d_c}$, where d_c is the dimension of node embedding, and each row of A_{sem} denotes the embedding of a node. Then we can multiply A_{sem} and A^T_{sem} to infer the semantic spatial dependencies between each pair of road nodes as follows

$$E_{sem} = Softmax(ReLU(A_{sem}A^T_{sem})) \tag{1}$$

where *Softmax* is used for the normalization of adaptive matrices. Note that this process aims to construct a graph based on node feature similarity.

Spatio-Temporal (ST) Encoder consists of two encoders, one for semantic traffic graph features encoding, and the other for structural traffic graphs encoding. The ST Encoders for the two graphs can be represented as follows.

$$\begin{aligned} h^t_{G_{sem}} &= Encoder(\{\mathcal{G}^t_{sem} | t = 1, \dots, T\}), \\ h^t_{G_{str}} &= Encoder(\{\mathcal{G}^t_{str} | t = 1, \dots, T\}). \end{aligned} \tag{2}$$

Both encoders contain stacked BSTLSN, which will be described in detail in Sect. 4.4. The proposed BSTLSN combines both local spatial and global semantic dependencies of the traffic flow data, and effectively addresses the data uncertainty issue by incorporating the Bayesian neural networks.

4.3 Spatio-temporal decoder

The data representations learned by the ST encoder next need to be decoded for generating the predicted traffic data in the future. As shown in the right part of Fig. 2, the ST decoder will first learn from the structural and semantic traffic graphs separately to obtain both local spatial and global semantic representations. Then the two views of data representations are fused as follows

$$h^t_{fus} = h^t_{G_{sem}} \oplus h^t_{G_{str}}, \tag{3}$$

where $\oplus$ represents a concatenation operation to fuse features of the two views. Then the fused feature representation h^t_{fus} will be input into the stacked BSTLSN for further capturing the complex Spatio-Temporal dependencies of traffic data, and at the same time complete decoding to facilitate the downstream prediction task. The ST decoder can be represented as follows

$$h^t_{decoder} = Decoder(h^t_{fus} | t = 1, \dots, T). \tag{4}$$

4.4 Bayesian spatio-temporal long short-term memory net

In this subsection, we introduce the key module of our model, the BSTLSTM layer in detail. Given a time slot t, we propose to adopt stacked Bayesian Graph Convolutional Network (BGCN) layers whose parameters follow a specific distribution for capturing the spatial correlation and uncertainty of the data. To more broadly capture the spatial and semantic correlations, we construct semantic ST graphs and structural ST graphs simultaneously, and learn the latent representations $h^t_{G_{sem}}$ and $h^t_{G_{str}}$ for graphs of the two views, respectively. The two data representations are calculated as follows

$$\begin{aligned} h^t_{G_{sem}} &= BGCN(\mathcal{G}^t_{sem}, W_{bayes1}), \\ h^t_{G_{str}} &= BGCN(\mathcal{G}^t_{str}, W_{bayes2}), \end{aligned} \tag{5}$$

where $\mathcal{G}^t_{sem}$ is the input of the semantic traffic graph, $\mathcal{G}^t_{str}$ is the input of the structural traffic graph, W_{bayes1} and W_{bayes2} represent the learnable parameters of the two views, respectively. In order to capture the temporal dependency, we next input the learned representations over T time slots to the long short-term memory network layers as follows

$$\begin{aligned} &[h^{t-T+1}_{LSTM_{G_{sem}}}, \cdots, h^t_{LSTM_{G_{sem}}}] = \\ &LSTM([h^{t-T+1}_{G_{sem}}, \cdots, h^t_{G_{sem}}], \theta_{lstm1}), \\ &[h^{t-T+1}_{LSTM_{G_{str}}}, \cdots, h^t_{LSTM_{G_{str}}}] = \\ &LSTM([h^{t-T+1}_{G_{str}}, \cdots, h^t_{G_{str}}], \theta_{lstm2}), \end{aligned} \tag{6}$$

where θ_{lstm1} and θ_{lstm2} represent the parameters of the corresponding LSTM network, $h_{LSTM_{G_{sem}}}$ and $h_{LSTM_{G_{str}}}$ are the outputs of LSTM.

To deal with the data uncertainty issue, we propose to construct a Bayesian Spatio-Temporal Long Short-term Memory Net (BSTLSN). BSTLSN integrates bayesian neural network which considers the parameters of the GCN following a particular distribution. By combining BGCN and LSTM, the process can be calculated as follows

$$\begin{aligned} h^t_{LSTM_{G_{sem}}} &= BSTLSN(\mathcal{G}^t_{sem}, W_{bayes1}, \theta_{lstm1}), \\ h^t_{LSTM_{G_{str}}} &= BSTLSN(\mathcal{G}^t_{str}, W_{bayes2}, \theta_{lstm2}), \end{aligned} \tag{7}$$

where W_{bayes1} and W_{bayes2} represent the learnable parameters of BGCN, θ_{lstm1} and θ_{lstm2} are the learnable parameters of LSTM.

4.4.1 GCN module for spatial correlation learning

To capture the local spatial and global semantic correlations, we propose to use the Graph Convolutional Network (GCN)

[27] to learn features on the graphs of the two views. GCN is used to extract local graphical features for non-Euclidean data. It aggregates the nodal information from neighboring nodes within a graph. This operation inherits the concept of convolution filter from the classical convolutional neural network (CNN). Graph convolution adopts graph connectivity as the filter for neighborhood aggregating to overcome the limitations of non-European input graph data. Such filters define a parametric uniform receptive field. In this way, the neighbors of the raw data are aggregated and result in local information sharing [37]. However, only performing GCN on the structural traffic graph is not enough for fully spatial feature learning. So we conduct GCN on both the structural traffic graph and the semantic traffic graph. Specifically, we conduct spectral graph convolutions [27] on the two graphs as follows.

$$h^t_G = f(H^t, A^t) = \sigma(D^{-\frac{1}{2}}\widetilde{A}^t D^{-\frac{1}{2}} H^t W^t) \tag{8}$$

where $f(\cdot,\cdot)$ represents the GCN operation, H^t and A^t are the node embedding and adjacency matrix of the graph G^t, respectively. $\widetilde{A}^t$ is the A^t with added self-connections. $D_{ii} = \sum_j \widetilde{A}^t_{ij}$ is the degree matrix. W^t means the learnable weight matrix. σ is a nonlinear function. The above formula can be interpreted as the first-order approximation of the local spectral filtering network, which itself is the local approximation of the spectral network convolved in the frequency domain using the graph Fourier transform according to the convolution theorem [38, 39]. Specifically, the spatial hidden features of layer *i-th* can be expressed as follows

$$h^t_{G,n} = \sigma(D^{\frac{1}{2}}\widetilde{A}^t D^{\frac{1}{2}} h^t_{G,n-1} W^t_n), \tag{9}$$

where $h_{G,n}$ represents the feature representation learned at the *i-th* layer, and W^t_n is the trainable matrix of filter parameters in the *n-th* graph convolutional layer.

In general, graph convolution operation reflects the physical relations of the traffic data in the road network. In practice, the traffic characteristics of two adjacent road links often show a strong correlation. For example, if a road link is congested, the traffic flow of its neighbor road link is also likely to be blocked. Such a spatial dependency can be captured by the adjacency matrix in formula (4), so that the traffic features of one node (road link) can be propagated to its neighbor nodes.

4.4.2 LSTM module for temporal dependency learning

Besides the spatial correlations, traffic data also present complex time dependencies [23, 28]. Thus we next input the extracted features from GCN into LSTM layers for temporal dependency feature learning, Compared with RNN, LSTM works better for long sequence modeling due to its long-term memory. Each neuron in the LSTM has three gates: forget gate, input gate, and output gate. By controlling the gate structure, LSTM has the function of long-term memory that captures the time dependency of long sequence data.

First, LSTM determines what information needs to be discarded through the forget gate control as follows.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{10}$$

where h_{t-1} is the output value at previous time t-1, x_t represents the input value at present time t, and $[\cdot,\cdot]$ denotes a vector splicing operation. σ is the sigmoid function that outputs a number between 0 (no pass) and 1 (all pass) to describe how much information can be passed. W_f, b_f and f_t are the weight matrix, bias term, and output of the forget gate, respectively.

Next, we decide what new information to add to the cell state by controlling the input gate. The specific operation for the input gate is as follows.

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i]) \\ \widetilde{C}^t &= tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \\ C_t &= f_t \times C_{t-1} + i_t \times \widetilde{C}_t \end{aligned} \tag{11}$$

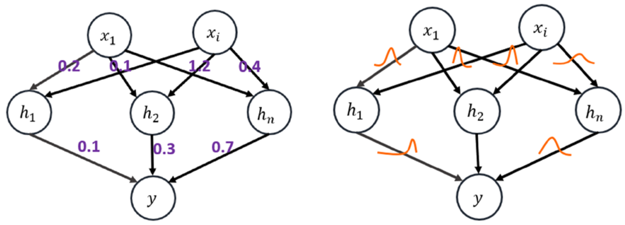
where i_t represents the information to be updated, and $\widetilde{C}^t$ is the new candidate value status created by *tanh* layer. C_t is the state value after updating the memory unit. Finally, we introduce how to determine the output of the memory unit based on the current cell state as follows.

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o]) \\ h_t &= o_t \times tanh(C_t) \end{aligned} \tag{12}$$

where W_o and b_o are the weight matrix and bias term of the output gate. o_t is the output for the output gate which determines which parts of the cell state can be exported. h_t is the final output of the LSTM Memory unit.

4.4.3 Bayesian spatial-temporal network

In order to alleviate the issue of data uncertainty and make the proposed framework more robust, we integrate the idea of Bayesian deep learning [30, 40] into our model. As shown in the left part of Fig. 3, the prediction of a traditional deep neural network model can be considered as a point estimation which means the parameters among the layers are fixed and are randomly initialized at the beginning of the model training. Thus the model is sensitive and easily influenced by the input data. In our task, the collected raw traffic data are sparse, noisy, and incomplete due to the uncertainty of road sensors. Thus the performance of the trained model will be degraded if we directly adopt these uncertain data for training. Following previous works [30, 35, 36], we learned that bayesian methods have the ability to deal with the problem of data uncertainty.



(a) DNN **(b)** BNN

Fig. 3 Illustration of DNN and BNN

Therefore, we propose to incorporate the Bayesian neural network into our model to address the data uncertainty issue.

As shown in the right part of Fig. 3, Bayesian neural networks assume the parameters of each layer follow a distribution rather than are fixed values, and then sample the parameter values from the distribution through the model training. Bayesian networks sample the parameters of each layer from a distribution and use uncertain parameters to deal with the changes of data so that the model has stronger robustness. A significant advantage of BNN is that it can resolve the data uncertainty problem and make the model more robust. From the probability theory point of view, a traditional neural network with one or multiple layers is a probabilistic model $P_r(y|x, \omega)$, where ω represents a collection for all parameters of all layers. In the traditional way of inference, the training of fixed value ω follows the Maximum Likelihood Estimation (MLE) with training set $\mathcal{D} = \{y_i|x_i\}$ as follows

$$\begin{aligned} \omega^{MLE} &= argmax_{\omega} logPr(\mathcal{D}|\omega) \\ &= argmax_{\omega} \sum_i logPr(y_i|x_i, \omega). \end{aligned} \tag{13}$$

MLE does not assume a prior probability of ω, i.e., it assumes that ω has an equal chance of taking any value. If we adopt regularization term as a priori to avoid overfitting, the optimal parameters follow the Maximum a Posterior (MAP) [41] as follows

$$\begin{aligned} \omega^{MAP} &= argmax_{\omega} logPr(\mathcal{D}|\omega) + logPr(\omega) \\ &= argmax_{\omega} logPr(\omega|\mathcal{D}). \end{aligned} \tag{14}$$

If we consider the parameters of the neural network layers following the posterior distributions embedded in the training set, the probability model can exploit data uncertainty and estimate distributions with Bayesian inference [42].

Following the above principle and motivated by previous work [30], we modify the original graph convolution in the GCN module in a Bayesian way. We introduce parameter ω into GCN, and its formula can be expressed as $y = f_{\omega}(x)$, where parameter ω follows the posterior distribution of the training set, so as to deal with the uncertainty of data. We employ zero-mean Gaussian as the prior distribution over the parameter space $Pr(\omega)$. According to Bayes' theorem [41], the posterior distribution can be calculated as follows

$$Pr(\omega|\mathcal{D}) = \frac{Pr(\omega, \mathcal{D})}{Pr(\mathcal{D})} = \frac{Pr(\mathcal{D}|\omega)Pr(\omega)}{Pr(\mathcal{D})}. \tag{15}$$

Nonetheless, $Pr(\omega)$ is hard to get and cannot be analytically estimated. To overcome this problem, we employ variational inference to get a variational distribution $q(\omega|\theta)$ which is parameterized by θ, and use it to approximate the posterior $Pr(\omega|\mathcal{D})$. We can find the optimal variational distribution by minimizing the Kullback-Leibler (KL) divergence between $Pr(\omega|\mathcal{D})$ and $q(\omega|\theta)$:

$$\begin{aligned} \theta^* &= argmin_{\theta} KL(q(\omega|\theta)||Pr(\omega|\mathcal{D})) \\ &= argmin_{\theta} \int q(\omega|\theta) log \frac{q(\omega|\theta)}{Pr(\omega)Pr(\mathcal{D}|\omega)} d\omega \\ &= argmin_{\theta} KL(q(\omega|\theta)||Pr(\omega)) - \mathbb{E}_{q(\omega|\theta)}(logPr(\mathcal{D}|\omega)) \end{aligned} \tag{16}$$

According to formula (16), the *KL* Loss L_{KL} is denoted as follows.

$$L_{KL} = KL(q(\omega|\theta)||Pr(\omega)) - \mathbb{E}_{q(\omega|\theta)}(logPr(\mathcal{D}|\omega)) \tag{17}$$

And we further take the idea of the Monte Carlo method to represent the L_{KL} as

$$L_{KL} = \sum_{i=1}^{n} logq(\omega_i|\theta) - logPr(\omega_i) - logPr(\mathcal{D}|\omega_i), \tag{18}$$

where ω^i is the weight of the *i-th* input data point.

In summary, Bayesian neural networks sample parameter values from a trained distribution to alleviate the problem of data uncertainty. We assume the parameters of GCN follow a specific distribution, and combine Bayesian principles with GCN to invent a Bayesian Graph convolutional Network. The Bayesian Graph Convolutional Network (BGCN) is used to construct BSTLSN layers in ST Encoder. The whole process can be expressed as follows

$$\begin{aligned} & h^t_{G_{sem}} = BGCN(\mathcal{G}^t_{sem}, W_{bayes1}), \\ & h^t_{G_{str}} = BGCN(\mathcal{G}^t_{str}, W_{bayes2}), \\ & h^t_{G_{sem}} : \mathcal{M}^{m\times1\times N\times C} \rightarrow \mathcal{M}^{m\times1\times N\times C'}, \\ & h^t_{G_{str}} : \mathcal{M}^{m\times1\times N\times C} \rightarrow \mathcal{M}^{m\times1\times N\times C'}, \\ & [h^{t-T+1}_{LSTM_{G_{sem}}}, \cdots, h^t_{LSTM_{G_{sem}}}] = \\ & LSTM([h^{t-T+1}_{G_{sem}}, \cdots, h^t_{G_{sem}}], \theta_{lstm1}), \\ & [h^{t-T+1}_{LSTM_{G_{str}}}, \cdots, h^t_{LSTM_{G_{str}}}] = \\ & LSTM([h^{t-T+1}_{G_{str}}, \cdots, h^t_{G_{str}}], \theta_{lstm2}), \end{aligned} \tag{19}$$

where m is the number of batch size, N is the number of nodes, C and C' represent the feature dimensions.

4.5 Overall objective function

The proposed MVB-STNet can be trained in an end-to-end way by minimizing the following training loss function.

$$L_{pre} = \frac{1}{n}\frac{1}{\mathcal{S}}\sum_{i=1}^{n}\sum_{j=1}^{\mathcal{S}}(\hat{Y}_{i,j} - Y_{i,j})^2 \tag{20}$$

where n represents the number of batches and $\mathcal{S}$ is the size of training samples in each batch. $\hat{Y}_{i,j}$ and $Y_{i,j}$ are the predicted traffic flow and the ground truth, respectively.

The final objective function of MVB-STNet contains two parts, the training loss of the prediction task L_{pre} and the KL loss L_{KL}. We integrate them together to achieve the overall loss function as follows.

$$L_{overall} = L_{pre} + \gamma L_{KL} \tag{21}$$

where γ is a hyperparameter to balance the importance of the KL loss. The KL loss is given in formula (16). The pseudocode for the training of MVB-STNet is shown in Algorithm 1.

Algorithm 1 MVB-STNet Algorithm

```
Input: $\{\mathcal{G}_{sem}^t | t = 1, \ldots, T\}$: Historical semantic traffic graphs; $\{\mathcal{G}_{str}^t | t = 1, \ldots, T\}$: Historical structural traffic graphs;
Output: The trained MVB-STNet model.
$\mathcal{D}_{train} \rightarrow \emptyset$
for $t \in \mathcal{T}$ do // $\mathcal{T}$ is available time set
    put an training instance ( $\{\mathcal{G}_{sem}^t, \mathcal{G}_{sem}^t | t \in [t, t+T]\}$, $\{X^t | t \in [t+T+1, t+2T]\}$) into $\mathcal{D}_{train}$ // $T$ is the length of time interval
end for
while not converge do
    Sequentially select a batch of instances $\mathcal{D}_{batch}$ from $\mathcal{D}_{train}$
    $S_{item} \leftarrow 0$
    for $S_{item} < S_{MAX}$ do // $S_{MAX}$ is the number of bayes sampling
        Sample weights $\theta$ from a specific Gaussian distribution
        $h_{G_{str}}^t \leftarrow$ Structural traffic graphs representation learning by $STEncoder$
        $h_{G_{sem}}^t \leftarrow$ Semantic traffic graphs representation learning by $STEncoder$
        $h_{fus}^t \leftarrow$ Integrated structural and semantic representation learning by Eq.17
        $h_{fus}^{'t} \leftarrow$ Decode the learned feature representation by $STDncoder$
        $\mathcal{X}^{t+T+1} \leftarrow LSTM(h_{fus}^{'t})$
        Update $\theta$ based on Eq. 19.
    end for
    Return the learned model parameters
end while
```

Table 1 Dataset description

Dataset	PeMS08	METR-LA
# of data samples	17856	34272
# of sensors (nodes)	170 sensors	207 sensors
Time period	2016/7/1~2016/8/31	2012/3/1 ~2012/6/30
Length of time slot	5 minutes	5 minutes

5 Experiments

5.1 Datasets

We use two publicly available real datasets that are widely adopted in traffic flow prediction for evaluation: *PeMS08* and *METR-LA*. The descriptions of the two datasets are shown in Table 1. The details of the two datasets are introduced as follows.

PeMS08 This traffic dataset is collected from traffic speed sensors in California within 2 months from 2016/7/1 to 2016/8/31 in San Bernardino. There are 170 roads in the dataset, forming a road network with 170 nodes. The traffic observations collected by the sensors include the traffic flow, traffic speed and others. The collected traffic observations on each road are aggregated every 5 minutes.

METR-LA It is a traffic dataset collected from Los Angeles. It contains the traffic observation data including the traffic flow and speed within 4 months from 2012/3/1 to 2012/6/30 of 207 highway sensors, which forms a road network with 207 nodes. The traffic observations on each road are also aggregated every 5 minutes.

5.2 Implementation details and experiment setup

We implement our model with Pytorch framework on NVIDIA Quadro RTX 3090 GPU. The parameters for the model are set as follows.The batch size and learning rate are set to 32 and 0.00001, respectively. We use Adam to optimize our model. We split each dataset into a training set, validation set, and test set with a ratio of 6:2:2. As for each dataset, we use the historical traffic observation value of 12-time slots to predict the traffic observations in the next time slot. We add random noise to the dataset or randomly delete partial data to simulate the uncertainty of the data and verify the reliability of the model.

We adopt Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) as follows as the evaluation metrics.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}||\hat{X}^{t+1} - X^{t+1}||^2}$$
$$MAE = \frac{1}{n}\sum_{t=1}^{n}|\hat{X}^{t+1} - X^{t+1}| \tag{22}$$

where $\hat{X}^{t+1}$ is the prediction and X^{t+1} is the ground truth.

Figure 4 shows the training loss curves of the algorithm on the two datasets. One can see that MVB-STNet converges quickly on both datasets. The loss curves drop smoothly, and there are almost no fluctuations in loss during training. This is mainly due to the data used for training is normalized. As the algorithm converges after around 30 epochs, in the following experiment we will adopt 50 epochs to train MVB-STNet on both datasets.

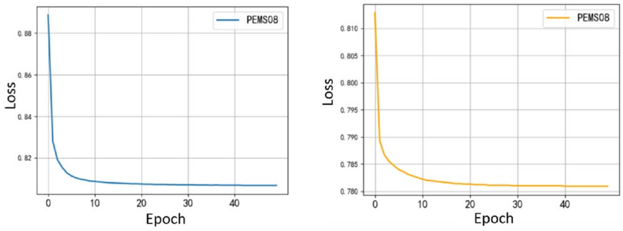
5.3 Baselines

We compare the proposed MVBT-STNet with the following 5 baseline methods, including both traditional statistics-based approaches and state-of-the-art deep learning-based approaches.

- **ARIMA** [7]: Auto-Regressive Integrated Moving Average (ARIMA) is a classical statistics-based method for time series prediction.
- **DCRNN** [12]: DCRNN integrates diffusion graph convolutional networks and recurrent neural network to learn the spatio-temporal correlations for traffic flow prediction.
- **STGCN** [13]: STGCN employs ChebNet graph convolution and 1D convolution to predict the traffic flow of each road segment in the road network.
- **AGCRN** [16]: AGCRN adopts Adaptive Graph Convolutional Recurrent Network to automatically capture the spatial and temporal dependencies in traffic series data for traffic forecasting.
- **ASTGCN** [14]: ASTGCN applies two attention layers extracting the dynamic correlation of traffic network respectively in spatial dimension and time dimension for traffic prediction.

5.4 Parameter analysis

In the overall objective function of formula (21), parameter γ is used to control the importance of the L_{KL} as a regularization term. As this term controls the loss of the Bayesian uncertainty, we first study how and to what extent the parameter γ will affect the model performance. We set γ to the values 0.1, 0.01, 0.001 and 0.0001, respectively, and test the model performance over the two datasets.



(a) PeMS08 **(b)** METR-LA

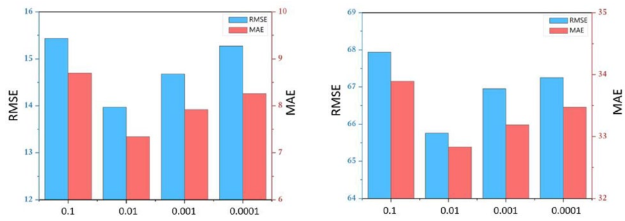
Fig. 4 Loss curves of MVB-STNet on two datasets

The result is shown in Fig. 5. One can see that the best performance is achieved when γ is set to 0.01 for both datasets. A too large (0.1) or too small (0.001) γ will hurt the model performance. This study shows that $\gamma = 0.01$ is a suitable parameter setting for the studied problem over the two datasets. In the following experiment, we set $\gamma = 0.01$.

5.5 Experimental results

The performance comparison result between MVB-STNet and the baselines are shown in Tables 2 and 3. We randomly add some noise or drop some data on the raw data to simulate the uncertainty in the road network. Specifically, for both datasets, we randomly add 10%, 20%, and 30% noise to the raw data to test the model performance, whose results are in Table 2. We next randomly drop 10%, 20%, and 30% data to simulate the incomplete data scenarios, and the corresponding experimental results are shown in Table 3. The best results are highlighted in bold font, and the best results achieved by baselines are underlined.

As shown in Tables 2 and 3, one can see that the proposed MVB-STNet achieves the best results in most cases, which proves that MVB-STNet is much more reliable than baselines in traffic forecasting when the input data are noisy and incomplete. The traditional statistics-based method ARIMA achieves the worse performance among all methods in all cases. This is not surprising, as ARIMA considers the traffic flows of each road link separately as a single time series data by ignoring the spatial correlations among the road links. Thus ARIMA does not work well when some data are missing or noise is added. One can also observe that on both datasets, RMSE and MAE of all the models present an increasing trend with the increase of added noise and the dropped. It implies that more noise and sparser data both make the prediction harder and thus lead to worse prediction performance. As shown in Table 2, MVB-STNet reduces RMSE by 8.1%, 12.6%, and 10.9% on the PeMS08 dataset



(a) METR-LA **(b)** PeMS08

Fig. 5 The effect of different γ values on the model performance

compared with the best results achieved by baseline methods under the three noise data scenarios, respectively. The corresponding MAE drops by 6.3%, 10.6%, and 9.4% for three cases, respectively. Both are significant performance improvements. For the METR-LA dataset, the RMSE and MAE of MVB-STNet drop by 3.6% and 7.9% when the noise ratio is 30%. As shown in Table 3, MVB-STNet reduces RMSE by 3.4%, 11.3%, and 9.5% respectively on the PeMS08 dataset compared with the best results achieved by the baseline methods under the three missing data scenarios. The corresponding MAE drops by 2.9%, 11.2%, and 9.8% for the three cases, respectively. The proposed MVB-STNet also performs the best on the METR-LA dataset.

5.6 Ablation study

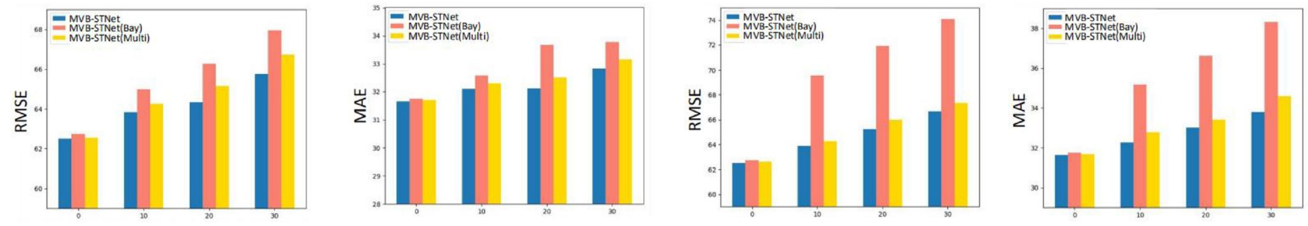
To examine whether the proposed Bayesian module and the multi-view learning module are both helpful to the prediction task, we conduct an ablation study by comparing the performance of the full version MVB-STNet with its variants models MVB-STNet(Bay) and MVB-STNet(Multi). MVB-STNet(Bay) removes the Bayesian uncertainty learning part from the full model. By comparing with it, we test whether BSTLSN layers can effectively deal with the data uncertainty issue and improve the prediction performance. MVB-STNet(Multi) removes the multi-view learning part from the full model. By comparison with this variant, we verify whether the multi-view learning part can effectively capture the local spatial and global semantic features and achieve better performance. The comparison result is shown

Table 2 RMSE and MAE comparison under different ratios of added noise

Model			ARIMA	DCRNN	STGCN	AGCRN	ASTGCN	MVB-STNet
PeMS08	RMSE	10% noise	133.40	85.67	74.62	84.97	69.43	**63.84**
		20% noise	169.33	84.74	73.61	82.95	73.68	**64.34**
		30% noise	201.97	88.69	76.47	86.86	73.77	**65.76**
	MAE	10% noise	88.70	42.32	36.66	41.54	34.27	**32.10**
		20% noise	96.54	42.12	35.94	40.37	36.16	**32.12**
		30% noise	105.42	43.54	37.19	42.14	36.24	**32.83**
METR-LA	RMSE	10% noise	34.15	15.92	**13.05**	16.39	13.54	13.44
		20% noise	56.40	17.22	14.04	17.59	14.25	**13.86**
		30% noise	71.53	18.29	14.49	18.61	15.39	**13.97**
	MAE	10% noise	23.14	9.56	**6.76**	9.46	6.92	6.83
		20% noise	34.70	10.13	7.42	10.15	7.53	**7.22**
		30% noise	40.98	10.84	7.98	10.84	8.63	**7.34**

Table 3 RMSE and MAE comparison under different ratios of missed data

Model			ARIMA	DCRNN	STGCN	AGCRN	ASTGCN	MVB-STNet
PeMS08	RMSE	10% missing	157.10	84.90	73.07	84.93	66.16	**63.90**
		20% missing	174.36	86.01	75.20	85.91	73.55	**65.24**
		30% missing	217.54	86.02	75.52	88.79	73.65	**66.67**
	MAE	10% missing	98.40	42.60	36.26	42.39	33.23	**32.27**
		20% missing	112.65	43.93	37.69	42.29	37.16	**33.01**
		30% missing	124.51	43.97	37.99	44.71	37.49	**33.8**
METR-LA	RMSE	10% missing	42.62	16.86	14.46	16.73	**14.02**	14.18
		20% missing	65.74	19.13	14.83	18.29	15.62	**14.77**
		30% missing	79.20	20.71	16.96	19.46	16.35	**15.27**
	MAE	10% missing	26.37	10.39	7.99	9.99	7.69	**7.38**
		20% missing	37.42	12.40	8.02	11.00	8.77	**7.74**
		30% missing	44.10	13.95	9.82	12.02	9.59	**8.08**



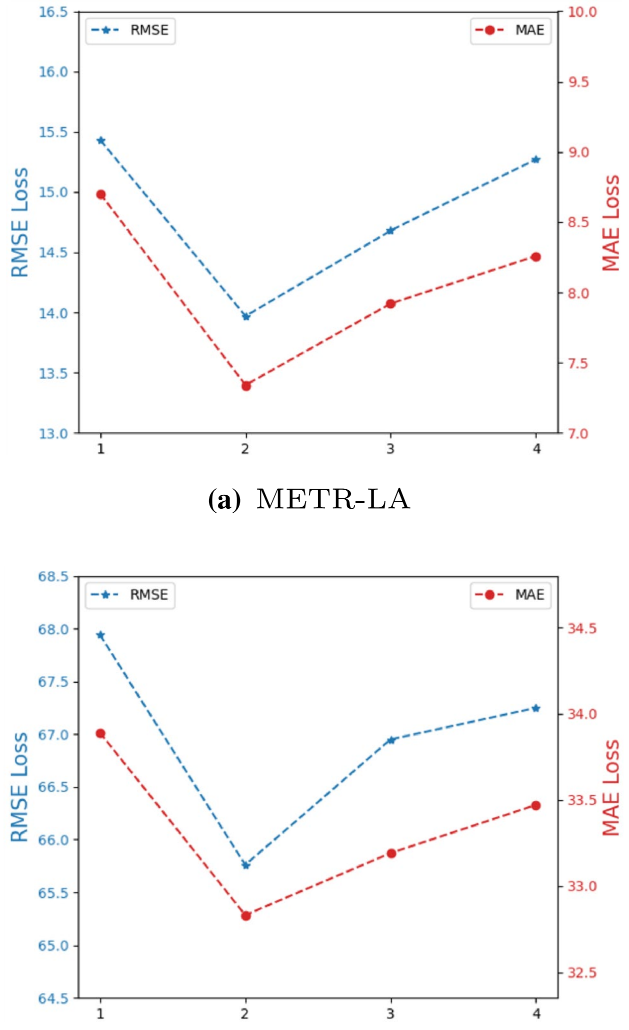
(a) Noise RMSE **(b)** Noise MAE **(c)** Data missing RMSE **(d)** Data missing MAE

Fig. 6 RMSE and MAE comparison with variant methods

in Fig. 6. One can see that the performance of the two variant models is inferior to the full MVB-STNet model, which implies that the proposed two modules are both useful to the studied problem, and removing any one of them will increase the prediction error. One can observe that the model performance generally degrades as the proportion of noise or missing data increases, which is consistent with the previous experiment result. Figure 6 also shows that the Bayesian uncertainty learning module is more important than the multi-view learning module, because the prediction error increases much more significantly when the Bayesian learning module is dropped.

5.7 Ablation study

To examine whether the proposed Bayesian module and the multi-view learning module are both helpful to the prediction task, we conduct an ablation study by comparing the performance of the full version MVB-STNet with its variants models MVB-STNet(Bay) and MVB-STNet(Multi). MVB-STNet(Bay) removes the Bayesian uncertainty learning part from the full model. By comparing with it, we test whether BSTLSN layers can effectively deal with the data uncertainty issue and improve the prediction performance. MVB-STNet(Multi) removes the multi-view learning part from the full model. By comparison with this variant, we verify whether the multi-view learning part can effectively capture the local spatial and global semantic features and achieve better performance. The comparison result is shown in Fig. 6. One can see that the performance of the two variant models is inferior to the full MVB-STNet model, which implies that the proposed two modules are both useful to the studied problem and removing any one of them will increase the prediction error. One can observe that the model performance generally degrades as the proportion of noise or missing data increases, which is consistent with the previous experiment result. Figure 6 also shows that the Bayesian uncertainty learning module is more important than the multi-view learning module, because the prediction error increases much more significantly when the Bayesian learning module is dropped.



(a) METR-LA

(b) PeMS08

Fig. 7 The effect of BSTLSN layer numbers on the model performance

5.8 Model sensitivity analysis

We next study how sensitive the model is to the deep neural structure (e.g., BSTLSN Layers). We show the performance curves of MVB-STNet over the two datasets by setting different BSTLSN layers from 1 to 4.

Figure 7 shows the MAE and RMSE curves under different number of layers of BSTLSN. One can see that the performance on both datasets first drops significantly and then slightly rises up with the increase of BSTLSN layers. It shows that 2 layers of BSTLSN is a reasonable setting in this experiment. This is mainly because we use the GCN to learn the spatial features, and usually a too deep GCN layers will lead to poor performance due to the effect of over smoothness. Only one layer cannot achieve desirable performance either because the complex features cannot be fully captured by only one layer BSTLSN.

6 Conclusion

In this paper, we studied the novel problem of reliable traffic flow with sparse, incomplete and noisy traffic data collected by road sensors with uncertainty. To cope with the uncertainty of data, we proposed a Multi-view Bayesian Spatio-Temporal Network named MVB-STNet. The proposed MVB-STNet first constructed the structural traffic graph and the semantic traffic graph to capture the local spatial and global semantic correlation of traffic data simultaneously. The features of the two views were first learned separately by GCN model and then integrated. The BSTLSN layer was next designed to integrate the Bayesian neural network with the proposed spatio-temporal feature learning network to capture the data uncertainty and made a more reliable prediction result. Extensive evaluation on two real datasets verified the effectiveness of our proposal in traffic flow prediction under various data uncertainty scenarios.

In the future, it would be interesting to further study whether the proposed multi-view Bayesian spatio-temporal prediction model can be used for other spatio-temporal prediction tasks, such as urban crowd flow prediction and demand prediction in on-demand services (e.g. Uber and Didi). We also plan to conduct a deeper study on how to design a more suitable prior distribution of the Bayesian model for a given particular prediction task and dataset.

Acknowledgements This work was supported by the National Science Foundation of China (No. 62172443), the Science and Technology Major Project of Changsha (No. kh2202004), Hunan Provincial Natural Science Foundation of China (No. 2022JJ30053), RGC Theme-based Research Scheme 2020/21 (T41-603/20-R), RGC Collaborative Research Fund (C5026-18G), and the high Performance Computing Center of Central South University.

References

1. Wang S, Cao J, Yu P (2020) Deep learning for spatio-temporal data mining: a survey. IEEE Trans Knowl Data Eng 2:2
2. Pal C, Hirayama S, Narahari S, Jeyabharath M, Prakash G, Kulothungan V (2018) An insight of world health organization (who) accident database by cluster analysis with self-organizing map (som). Traffic Inj Prev 19(sup1):S15–S20
3. Tedjopurnomo DA, Bao Z, Zheng B, Choudhury F, Qin A (2020) A survey on modern deep neural network for traffic prediction: trends, methods and challenges. IEEE Trans Knowl Data Eng 2:2
4. Wang J, Chen Q, Gong H (2020) Stmag: a spatial-temporal mixed attention graph-based convolution model for multi-data flow safety prediction. Inf Sci 525:16–36
5. Chandra SR, Al-Deek H (2009) Predictions of freeway traffic speeds and volumes using vector autoregressive models. J Intell Transp Syst 13(2):53–72
6. Williams BM (2001) Multivariate vehicular traffic flow prediction: evaluation of arimax modeling. Transp Res Rec 1776(1):194–200
7. Williams BM, Hoel LA (2003) Modeling and forecasting vehicular traffic flow as a seasonal arima process: theoretical basis and empirical results. J Transp Eng 129(6):664–672
8. Lee K, Eo M, Jung E, Yoon Y, Rhee W (2021) Short-term traffic prediction with deep neural networks: a survey. IEEE Access 9:739–756
9. Wang Y, Zhang D, Liu Y, Dai B, Lee LH (2019) Enhancing transportation systems via deep learning: a survey. Transp Res Part C Emerg Technol 99:144–163
10. Zhao L, Song Y, Zhang C, Liu Y, Wang P, Lin T, Deng M, Li H (2019) T-gcn: A temporal graph convolutional network for traffic prediction. IEEE Trans Intell Transp Syst 21(9):3848–3858
11. Yuan H, Li G (2021) A survey of traffic prediction: from spatio-temporal data to intelligent transportation. Data Sci Eng 6(1):63–85
12. Li Y, Yu R, Shahabi C, Liu Y (2018) Diffusion convolutional recurrent neural network: data-driven traffic forecasting. In: International Conference on Learning Representations
13. Yu B., Yin H, Zhu Z (2018) Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence, pp. 3634–3640
14. Guo S, Lin Y, Feng N, Song C, Wan H (2019) Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. Proc AAAI Conf Artif Intell 33(01):922–929
15. Zheng C, Fan X, Wang C, Qi J (2020) Gman: a graph multi-attention network for traffic prediction. Proc AAAI Conf Artif Intell 34(01):1234–1241
16. Bai L, Yao L, Li C, Wang X, Wang C (2020) Adaptive graph convolutional recurrent network for traffic forecasting. Adv Neural Inf Process Syst 33:25
17. Tobler WR (1970) A computer movie simulating urban growth in the detroit region. Econ Geogr 46(1):234–240
18. Yao H, Wu F, Ke J, Tang X, Jia Y, Lu S, Gong P, Ye J, Li Z (2018) Deep multi-view spatial-temporal network for taxi demand prediction. In: Proceedings of AAAI
19. Zivot E, Wang J (2006) Vector autoregressive models for multivariate time series. Modeling financial time series with s-plus®, pp. 385–429
20. Castro-Neto M, Jeong Y-S, Jeong M-K, Han LD (2009) Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions. Expert Syst Appl 36(3):6164–6173
21. Johansson U, Boström H, Löfström T, Linusson H (2014) Regression conformal prediction with random forests. Mach Learn 97(1–2):155–176
22. Wang S, Zhang X, Li F, Yu PS, Huang Z (2019) Efficient traffic estimation with multi-sourced data by parallel coupled hidden markov model. IEEE Trans Intell Transp Syst 20(8):3010–3023
23. Zhang J, Zheng Y, Qi D (2017) Deep spatio-temporal residual networks for citywide crowd flows prediction. In: Thirty-first AAAI conference on artificial intelligence

24. Yao H, Tang X, Wei H, Zheng G, Li Z (2019) Revisiting spatial-temporal similarity: a deep learning framework for traffic prediction. Proc AAAI Conf Artif Intell 33(01):5668–5675
25. Lin Z, Feng J, Lu Z, Li Y, Jin D (2019) Deepstn+: context-aware spatial-temporal neural network for crowd flow prediction in metropolis. Proc AAAI Conf Artif Intell 33(01):1020–1027
26. Yao H, Wu F, Ke J, Tang X, Jia Y, Lu S, Gong P, Ye J, Li Z (2018) Deep multi-view spatial-temporal network for taxi demand prediction. Proc AAAI Conf Artif Intell 32(1):2
27. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR)
28. Yang Y, Cao J, Stojmenovic M, Wang S, Cheng Y, Lum C, Li Z (2021) Time-capturing dynamic graph embedding for temporal linkage evolution. IEEE Trans Knowl Data Eng
29. Wu Z, Pan S, Long G, Jiang J, Zhang C (2019) Graph wavenet for deep spatial-temporal graph modeling. In: The 28th International Joint Conference on Artificial Intelligence (IJCAI). International Joint Conferences on Artificial Intelligence Organization
30. Blundell C, Cornebise J, Kavukcuoglu K, Wierstra D (2015) Weight uncertainty in neural network. In: International Conference on Machine Learning. PMLR, pp. 1613–1622
31. Kingma DP, Welling M (2013) Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114
32. Tishby N, Levin E, Solla SA (1989) Consistent inference of probabilities in layered networks: Predictions and generalization. In: International Joint Conference on Neural Networks, vol. 2, pp. 403–409
33. MacKay DJ (1992) A practical bayesian framework for backpropagation networks. Neural Comput 4(3):448–472
34. Mullachery V, Khera A, Husain A (2018) Bayesian neural networks. arXiv preprint arXiv:1801.07710
35. Kristiadi A, Hein M, Hennig P (2020) Being bayesian, even just a bit, fixes overconfidence in relu networks. In: International Conference on Machine Learning. PMLR, pp 5436–5446
36. Xiao Z, Shen J, Zhen X, Shao L, Snoek CG (2021) A bit more bayesian: Domain-invariant learning with uncertainty. arXiv preprint arXiv:2105.04030
37. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 25:1097–1105
38. Defferrard M, Bresson X, Vandergheynst P (2016) Convolutional neural networks on graphs with fast localized spectral filtering. Adv Neural Inf Process Syst 29:3844–3852
39. Bruna J, Zaremba W, Szlam A, LeCun Y (2014) Spectral networks and deep locally connected networks on graphs. In: 2nd International Conference on Learning Representations, ICLR 2014
40. Gal Y, Ghahramani Z (2016) Dropout as a bayesian approximation: representing model uncertainty in deep learning. In: International conference on machine learning. PMLR, pp 1050–1059
41. Bassett R, Deride J (2019) Maximum a posteriori estimators as a limit of bayes estimators. Math Progr 174(1):129–144
42. Wang H, Yeung D-Y (2016) Towards bayesian deep learning: a framework and some existing methods. IEEE Trans Knowl Data Eng 28(12):3395–3408