



Traffic congestion propagation inference using dynamic Bayesian graph convolution network

Sen Luan ^a, Ruimin Ke ^b, Zhou Huang ^c, Xiaolei Ma ^{a,d,*}

^a School of Transportation Science and Engineering, Beijing Key Laboratory for Cooperative Vehicle Infrastructure System and Safety Control, Beihang University, Beijing 100191, China

^b Department of Civil Engineering, The University of Texas at El Paso, El Paso, TX 79968, USA

^c Institute of Remote Sensing and Geographic Information System, Peking University, Beijing 100871, China

^d Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing 100191, China



ARTICLE INFO

Keywords:

Traffic congestion propagation
Bayesian
Graph Network
Dynamic

ABSTRACT

Congestion, whether recurrent or non-recurrent, propagates through the road network. The process of congestion propagation from a particular road to its neighbors can be regarded as a kind of message passing with a directed relationship. Existing methods have created a solid foundation for characterizing congestion propagation; however, they are either built upon simplified assumptions in traffic flow theory or predefined relationships among road sections, which would lead to downgraded accuracy in practice. This paper proposes a dynamic Bayesian graph convolutional network (DBGCN), which integrates Bayesian inference into a deep learning framework. Therefore, the rules of congestion propagation in the network can be actively learned from the observed data instead of predefining them based on prior knowledge. Experimental results on 971 testbeds in a regional road network in Beijing demonstrate that DBGCN outperforms the state-of-the-art models in inferring the congestion propagation spatiotemporal coverage and reveals variations in congestion propagation patterns according to the road network structure. Furthermore, the proposed model can simulate the congestion propagation process in customized scenarios by learning the latent congestion propagation rules. The results in different scenarios show that the change of congestion source location leads to distinct congestion magnitude, and the propagation of congestion will eventually stop at the road sections with strong shunting effect.

1. Introduction

Traffic congestion has two types: recurrent and non-recurrent congestion. Recurrent congestion is often predictable, appearing at the fixed bottlenecks of a road network during similar times of the day. Non-recurrent congestion is usually caused by abnormal incidents (e.g., traffic crashes and adverse weather) and is also called incident congestion. Incident congestion may randomly occur at any time and place (Anbaroglu et al., 2014), which substantially increases the difficulty in preventing and controlling its negative effects. If transportation agencies cannot formulate effective congestion mitigation strategies timely, then local congestion may affect neighboring road sections and even lead to network-scale traffic paralysis. Traffic congestion propagates in the road network from the

* Corresponding author at: School of Transportation Science and Engineering, Beijing Key Laboratory for Cooperative Vehicle Infrastructure System and Safety Control, Beihang University, Beijing 100191, China.

E-mail address: xiaolei@buaa.edu.cn (X. Ma).

congestion source regardless of its type. Therefore, the key to formulating effective congestion mitigation strategies lies in modeling the propagation rules of congestion in the road network.

For decades, many efforts have been devoted to the research of congestion propagation inference. Mathematical analysis models have been used earlier to simulate the spatiotemporal phenomena of congestion propagation. The model with the single-car as the research object is called the microscopic model, of which car following (Treiber et al., 2000) and queuing theory (Laih, 1994) are two representative ones. The results of the microscopic model may have distortion problems because obtaining the data of all driving vehicles is difficult. In contrast, the macroscopic model mainly uses traffic flow, density, and speed to describe traffic phenomena. Daganzo (1994) proposed the cellular transmission model (CTM) based on traffic flow theory to describe the propagation process of congestion in the road network. Since then, a series of different CTM-based methods have appeared (Long et al., 2008; Zhang et al., 2012; Ran et al., 2016) to describe the propagation characteristics of recurrent and incident congestion. They have created a solid foundation in the research area of congestion propagation, but this type of microscopic model lacks applicability due to an excessive number of underlying mathematical assumptions and oversimplified modeling of the road network into regular grids (Qi et al., 2013; Zhao et al., 2017).

Meanwhile, some studies regard the propagation process of congestion in the road network as a state transition between adjacent road sections (Lei et al., 2015; Nguyen et al., 2017; Fan et al., 2019). These studies use the Bayesian network (BN) with the help of data-driven ideas to learn the state transition probability between road sections from historical data and then infer congestion propagation. Although BN is good at inferring directed relationships, most current BN-based research discretizes continuous speed data into several traffic states. This kind of data discretization makes it difficult to fully capture and get insight into the spatiotemporal dependence in the process of congestion propagation.

In recent years, an emerging group of methods for this problem has been based on graph convolutional networks (GCNs), which not only retain road network topology information but also learn the spatiotemporal characteristics of traffic state evolution. Therefore, the traffic road network is represented as a graph $G = (v, A)$, where node v represents a road section, and A denotes an adjacent matrix, indicating the connectivity between nodes. The adjacent matrix enables the passage of traffic state information through neighboring nodes. The GCN-based models achieve superior performance in traffic prediction at the road network level due to this information-passing mechanism (Wu et al., 2019; Song et al., 2020; Yu et al., 2020; Cui et al., 2020). Regarding congestion evolution as a kind of traffic state information passing through the road network, the GCN-based model can provide a feasible solution for congestion propagation analysis.

Although GCN has achieved remarkable success in traffic prediction, its application to congestion propagation inference remains challenging. GCN relies on the information-passing structure defined by the adjacent matrix. However, the adjacency matrix is mostly predefined in the existing research according to the link connectivity or further weighted based on the link capacity and inter-link distance (Jiang and Luo, 2021). In practice, the predefined adjacency matrix cannot accurately describe the process of congestion propagation due to two main reasons. First, road sections are interconnected in a traffic network, but congestion does not spread endlessly in the road network. Second, the predefined adjacency matrix is fixed over time, which does not conform to the time-varying characteristics of traffic state evolution. Therefore, learning the adjacency matrix from continuously evolving data rather than relying only on the road network topology is necessary.

The above concerns prompted the proposal of a dynamic Bayesian GCN (DBGCN), which manages to integrate Bayesian inference with deep learning networks. DBGCN comprises two data-driven modules. The first module aims to use Bayesian inference to learn the dynamic propagation rules of congestion in the road network from historical data. The second module comprises multiple GCNs. Each GCN accepts the adjacency matrix transformed from the congestion propagation rules, enabling the nodes in the graph to aggregate the information from different adjacency nodes during the congestion propagation process. Experimental results based on actual congestion data show that DBGCN achieves superior performance to the state-of-the-art models. The contribution of the paper is three-fold.

- A dynamic adjacent weighting mechanism among road network sections is designed to mimic the time-varying feature of traffic flow. Such dynamic adjacent matrices are fed into a GCN for traffic congestion evolution modeling, where traditional GCNs consider fixed adjacent matrices.
- The proposed model can effectively learn the rules of congestion propagation and realize accurate congestion propagation inference by integrating Bayesian inference with deep learning.
- Different congestion scenarios are further constructed through the learned congestion propagation rules to explore the impact of road network topology and congestion duration on congestion propagation.

The rest of this paper is organized as follows. The following section summarizes related research on traffic congestion propagation and discusses the application of GCNs in traffic prediction. The third section describes the spatiotemporal characteristics of congestion propagation and the construction of DBGCN. The fourth section introduces testbed selection and data collection. The fifth section compares the results from the proposed model with those from the baseline models and then analyzes the congestion evolution in different scenarios. The last section presents the concluding remarks and future work.

2. Literature review

The key to congestion propagation analysis is to obtain the propagation rules of congestion in the road network. Many methods have been developed in the past few decades for this purpose. This paper divides these methods into two categories: simulation-driven

and artificial intelligence- driven (AI-driven).

2.1. Simulation -driven methods

Simulation -driven methods have been earlier applied to congestion propagation analysis. These methods are usually implemented via simulation due to the complex and dynamic traffic flow characteristics in networks. These **simulation methods** can be divided into **microscopic** and **macroscopic** ones according to the different objects.

The **microscopic** models take each vehicle as the research object and transform the traffic congestion propagation process into a queuing problem. Moreles et al. (1986) used the curves of arrival and departure rates to estimate the vehicle queue length induced by traffic incidents. Subsequently, the cumulative arrival and departure curves based on kinematic wave theory are adopted to estimate the queue length, realizing model consistency with the realistic traffic condition (Newell, 1993). Helbing (2003) proposed an improved dynamic queuing model based on the traffic flow characteristics of the road cross-section to describe the hysteresis of the traffic flow and congestion propagation. However, the congestion propagation law obtained by the micro model may be distorted due to the use of a simplified traffic network. Thus, microscopic models have not been widely used in large-scale networks.

The **macroscopic** model can be traced back to the continuum model of traffic flow proposed by Lighthill and Whitham (1955) and Richards (1956), which is also called the LWR model. Daganzo (1994) proposed the cell transmission model (CTM) based on the discrete form of the LWR model. The CTM model divides the road network into sections or grids, and each unit is treated as a cell. CTM can effectively describe traffic congestion's spatial and temporal evolution by analyzing the traffic parameters (volume, density, and speed) of each cell at each time step and is thus widely used in the propagation modeling of recurrent and non-recurrent congestions. Chris (2007) proposed the EKF-CTM model by introducing the extended Kalman filter into the CTM to estimate the macroscopic traffic state. Bourrel (2003) established a micro- and micro-hybrid model based on the LWR model to study the formation and dissipation mechanisms of congestion caused by traffic incidents. Long et al. (2011) studied the influential factors, such as the distribution of stop line width, the road channelization area, the road length, and the accident location into the propagation formation, based on the improved CTM and modeled the dissipation of traffic congestion through computer simulation. However, these CTM-based models excessively rely on the underlying theory. The standardized and simplified simulation network is usually designed as the testbed, resulting in the lack of sufficient applicability in modeling realistic transportation networks (Qi et al., 2013; Zhao et al., 2017). Interestingly, Saberi et al. (2020) regarded congestion propagation as the spread of epidemic and then introduced the susceptible-infected-recovered (SIR) model to explore the mechanism of the propagation and dissipation of congestion dependent on time-varying travel demand. And the author validated their model using empirical and simulated-based experiments at the city-wide level.

Collectively, the above-mentioned simulation-driven methods can describe the propagation of congestion in a vivid way, but they all need traffic flow theory as the model foundation. For this reason, the collection and calibration of multiple traffic parameters becomes challenging when these models are applied to large-scale road networks.

2.2. AI-driven methods

AI-driven models often learn spatiotemporal dependencies from the representation of traffic data. An easy-to-implement AI-driven method is BN, where the propagation of congestion on adjacent road sections is regarded as a state transition between nodes in the network. Sun et al. (2006) defined road sections as cause-and-effect nodes in BN and used the Gaussian mixture model to calculate the joint probability distribution between the two types of nodes. Nguyen et al. (2017) combined multiple congested road sections into a congestion tree structure according to temporal and spatial dependence and then used BN to calculate the joint probability of the congestion tree structure. They suggested that increased attention should be provided to the congestion trees with a high probability because these trees may be the network bottlenecks due to the frequent occurrence. Fan et al. (2019) learned the congestion propagation structure with the objective function of maximizing the network state transition probability by discretizing the speed value into four traffic states (severely congested, congested, low speed, and smooth). Although BN is good at inferring directed relationships, most current BN-based research discretizes continuous speed data into several discrete traffic states. Data discretization can reduce the model complexity, but it is not conducive to measuring the evolution trend of congestion in a microscopic and fine manner.

Traffic data with spatiotemporal attributes can be converted into two-dimensional matrices or images in Euclidean space. Therefore, deep learning models, such as long short-term memory (LSTM) neural networks in Ma et al. (2015), bi-directional LSTM proposed by Cui et al., 2018, multi-channel convolution neural network (Ke et al., 2020), generative adversarial networks (Liang et al., 2018), or their combinations (Yang et al., 2019), are widely introduced into the field of traffic forecasting. However, traffic congestion propagation cannot be equated with traffic prediction, which also relies on the road network topology. The non-Euclidean characteristic of road network topology contributes to the unsatisfactory performance of these traditional deep learning models (Wu et al., 2020). The improvement of this problem is attributed to the GCN proposed by Kipf and Welling (2016). GCN extends the convolution operation from traditional data (e.g., image or grid-like data) to graph data. The traffic network can naturally be regarded as a graph, in which each node refers to a road section. Therefore, the GCN-based model is suitable for traffic state evolution inference in city-wide road networks (Zhang et al., 2020).

In only a few years, a myriad of GCN variants has emerged for specific purposes. Yu et al. (2017) proposed STGCN, which comprises multiple spatiotemporal convolution modules. Each module comprises a graph convolution layer (Defferrard et al., 2016) and a convolution sequence learning layer to capture the spatiotemporal dependence of traffic flow. Wu et al. (2019) proposed the Graph WaveNet, in which the number of one-dimensional convolution components exponentially expands as the number of layers increases to aggregate information from long-term sequences. Song et al., (2020) proposed STSGCN, which uses a spatiotemporal

synchronization modeling mechanism to capture complex local spatiotemporal dependencies and heterogeneity. Considering the dynamic characteristics of the traffic state, Yu et al. (2020) proposed an extended GCN, comprises multiple graph convolution streams containing different numbers of convolutions. The GCN uses dynamic adjacency matrices in each convolution stream, describing distinct traffic state propagation modes under different periods. In addition, Cui et al. (2020) introduced the Markov process into the GCN framework to solve missing data encountered in traffic prediction. The above-mentioned GCN-based models outperform previous deep learning models due to their super-capturing capability of space-time dependence (Jiang and Luo, 2021).

However, the existing GCNs still face critical challenges in congestion propagation inference. The majority of previous studies directly predefine the adjacency matrix based on prior fixed knowledge (essentially the road connectivity). Thus, additional efforts, including Gaussian mixture CNNs (Monti et al., 2017), attention mechanism (Velickovic et al., 2017), and adaptive path layer (Liu et al., 2019), are devoted to learning weights between nodes dynamically. Although these studies believe that the weights between pairs of nodes are different and must be updated, they still rely on a predefined adjacency matrix. Undoubtedly, the propagation of congestion is subject to the physical constraint of the road network topology, whereas congestion propagation rules are more complicated than the road network topology. For example, a node may be affected by non-first-order neighbor nodes, and no relationship may exist between neighboring nodes due to the shunting effect in the road network. Therefore, learning dynamic congestion propagation patterns from continuously evolving data is attempted using a hybrid method based on prior and data-driven knowledge.

3. Methodology

3.1. Problem statement and preliminary

3.1.1. Congestion propagation inference

Congestion propagation can be modeled as a process in which the speed drop of a particular road section causes the speed reduction of a series of upstream road sections. This process is essentially a kind of information-passing mechanism. Assume that x_i^t is the observed feature of the congestion source, and the state of its upstream road x_i as the congestion propagates can be inferred by:

$$x_i^{t+1} = \gamma \left(x_i^t, F_{i,j \in Pa(i)} (x_i^t, x_j^t) \right), \quad (1)$$

where γ is a differentiable function (such as multilayer perceptron), $j \in Pa(i)$ indicates that the change in traffic state of road j has an impact on the state of road i , and $F(\bullet)$ denotes an aggregate function (e.g., sum, mean, or max).

3.1.2. Bayesian property in congestion propagation

Congestion propagation relies on the conditional dependence of the traffic state between upstream and downstream road sections. Thus, congestion propagation is believed to satisfy the Bayesian property. More importantly, the dependence between road sections may be time-varying due to the dynamics in traffic evolution. Fig. 1 plots a three-time slice directed acyclic graph (DAG) to demonstrate this dynamic Bayesian property in the congestion propagation process.

This DAG has the following characteristics. (1) The road network topology remains constant over time (solid arrow). (2) The state of the road network at the next time slice only depends on the current state of the road network (dashed arrow). (3) The state of a single road is affected by the roads with connections, and the connections may be time-varying. Therefore, the Bayesian property in congestion propagation can be written as Eq. (2).

$$Pr(X^{t+1}|X^t) = \prod_{i=1}^N Pr(x_i^{t+1}|Pa(x_i^{t+1})), \quad (2)$$

where $Pa(x_i^{t+1})$ represents a set of road sections that impact road section i at time slice $t + 1$. The dynamics of the congestion propagation rules are manifested in this paper as the possible variation of $Pa(x_i)$ over time, that is, the presence or absence of the dotted

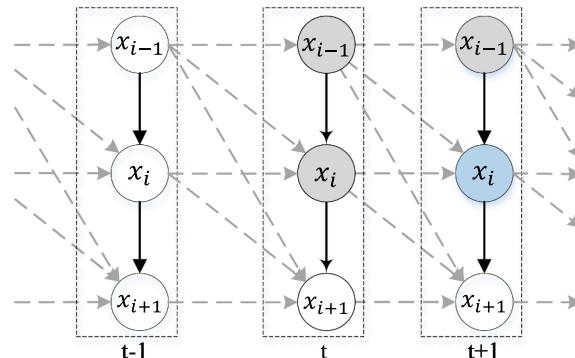


Fig. 1. Three time-slice DAG for congestion propagation.

arrow.

3.1.3. Graph representation of congestion propagation

A DAG formed by congestion propagation can be expressed as $\mathcal{G} = (\mathcal{V}, \mathcal{A}^t)$ with a set of nodes $\mathcal{V} = \{v_1, v_2, \dots, v_S\}$ and an adjacent matrix \mathcal{A}^t . Each node v represents a road section, and \mathcal{A}^t shows the possible influence between nodes in two adjoining time slices. For example, $\mathcal{A}_{ij}^t = 1$ indicates that node v_i in time slice t generates an impact on v_j in time slice $t + 1$. Let W represent the intensity of influence (or weight) between nodes, and X denotes the features corresponding to node \mathcal{V} . Eq. (1) can then be rewritten in a general GCN format as:

$$X^{t+1} = \gamma(\mathcal{A}^t \odot W)X^t. \quad (3)$$

The adjacency matrix and the weight determine the congestion propagation rules. To this end, the following section will introduce learning of the two aspects from historical congestion data in the leverage of DBGCN.

3.2. Framework of DBGCN

As shown in Fig. 2, the overall framework of DBGCN comprises three modules: (1) data input layer, (2) Bayesian inference module, and (3) a stacked GCN module. The data input layer extracts data using a time lag mechanism, and the value of time lags determines the number of time slices in the DAG. The data sets with time lags are then inputted to the Bayesian inference module to learn the congestion propagation network structure. Finally, the adjacent matrixes converted from the congestion propagation network and the data sets are sent to the stacked GCN module for model training and validation. In addition, the related data samples and codes are available on GitHub: https://github.com/luansenda/congestion_propagation_inference.

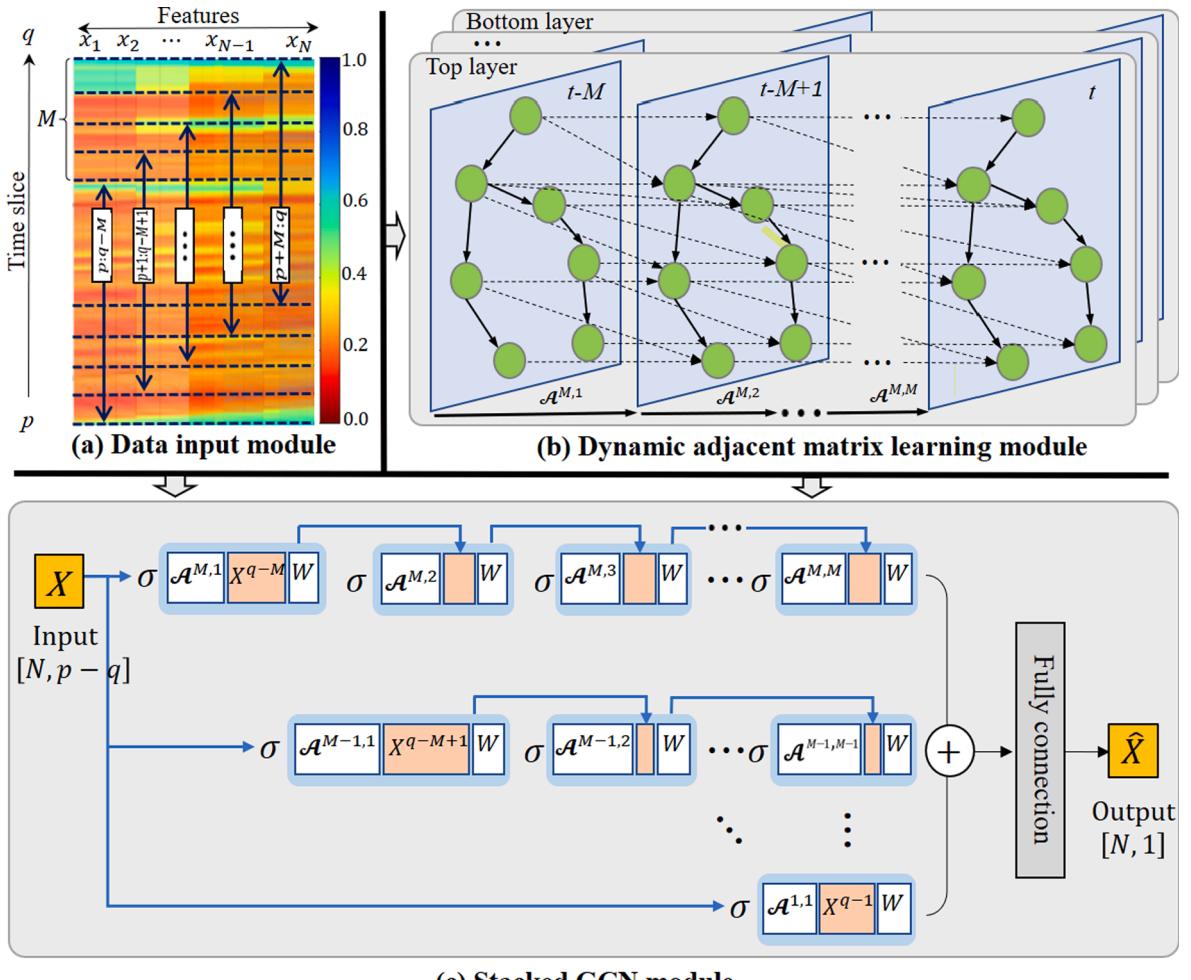


Fig. 2. The overall framework of DBGCN.

3.2.1. Input layer processing

A time lag mechanism is added in input data extraction to explore the effects of previous traffic status on the current traffic status. The time window of the data set slides one step to the past when the time lag increases by one. Fig. 2a depicts a complete congestion propagation process, which starts at time slice p and ends at time slice q . If the value of time lag is M , then the data set will be slidably divided into $M + 1$ data subsets, $[X^{p:q-M}, X^{p+1:q-M+1}, \dots, X^{p+M-1:q-1}, X^{p+M:q}]$. Data format conversion is then performed according to Eq. (4) to prepare for learning the congestion propagation rules between adjacent time slices.

$$\mathbf{D} = \begin{bmatrix} [X^{p:q-M}, X^{p+1:q-M+1}] \\ [X^{p+1:q-M+1}, X^{p+2:q-M+2}] \\ \vdots \\ [X^{p+M-1:q-1}, X^{p+M:q}] \end{bmatrix} \quad (4)$$

Notably, an excessively large M is not allowed because the time interval of the traffic state is 5 min, and $M < (q - p)$ must also be met. For example, if congestion lasts for 20 min, then $q - p$ is equal to 4, indicating that the data set only has four-time slices. Therefore, the value of M should be less than 4. In addition, the congestion state in this paper is a linear mapping of travel speed. Section 4.1 will comprehensively introduce the congestion representation and the filtration of congestion data.

In fact, the data processing procedure is also common in deep learning networks in order to expand training samples. For example, the [training, validation] sets are usually characterized as data from $[t_1 \sim t_5, t_6 \sim t_8], [t_2 \sim t_6, t_7 \sim t_9], \dots, [t_{n-4} \sim t_n, t_{n+1} \sim t_{n+3}]$. Another functionality of the data input layer in this paper is to convert the data into a format that is convenient for dynamic Bayesian learning. Let's take $M = 3$ as an example to illustrate. Fig. 3 presents the new data set composed of 4 subsets. The data size has changed from $(q - p) \times N$ to $(q - p - 3) \times (4^*N)$. By doing so, the data in each row can reflect the evolution of all road sections (x_1, \dots, x_N) at four consecutive moments, which facilitates learning the dynamic dependencies among road segments in the second module.

3.2.2. Learning mechanism of congestion propagation

The adjacency matrix is a digital representation of the congestion propagation network structure. The learning methods of network structure are usually classified into constraint-based (Spirtes and Meek, 1995) and scoring search-based (Koller and Friedman, 2009). The constraint-based method uses prior knowledge to determine the directed relationship between nodes. The scoring search-based method uses the defined scoring function to find the network structure with the highest matching degree with the sample data. Nevertheless, it is worth noting that structure learning is always an NP-hard problem (Luo et al., 2020).

This paper introduces a hybrid algorithm combining constraints and the score function to obtain a reasonable network structure. As mentioned in section 3.1.2, the current state of the DAG is only affected by its previous state. Therefore, the dynamic propagation structure learning procedure can be decomposed into the learning of every two-time slice DAG. Each two-time slice DAG can be converted into an adjacent matrix. As shown in Fig. 2(b), if $M = 3$, there will be a three-layer stacked propagation network. Below we take the first two-time slice DAG ($\mathcal{G}^{3,1}$) at the top layer as an example to illustrate the network structure learning procedure.

Step 1: Constrain the directed relationship between nodes based on empirical knowledge. The process of congestion propagation has two natural constraints. First, congestion will only propagate from the source node to its upstream neighbor nodes during its occurrence. Second, the current state of the node is affected by its previous state, indicating that the node is self-connected. Therefore, for all adjacent matrices, the relationship between nodes satisfies the following constraints:

$$\left\{ \begin{array}{l} A_{i,i} = 1 \\ A_{i,k} = 0, \text{ if } \text{node } k \text{ is downstream of node } i \end{array} \right. \quad \begin{array}{l} \text{all these are upstreams by} \\ \text{declaration of road section} \\ \text{traffic} \\ \text{then, opposite to these traffic are said to be downstream} \end{array} \quad (5)$$

Step 2: Construct a scoring function to find the optimal structure. The optimal structure is essentially the result of combinatorial

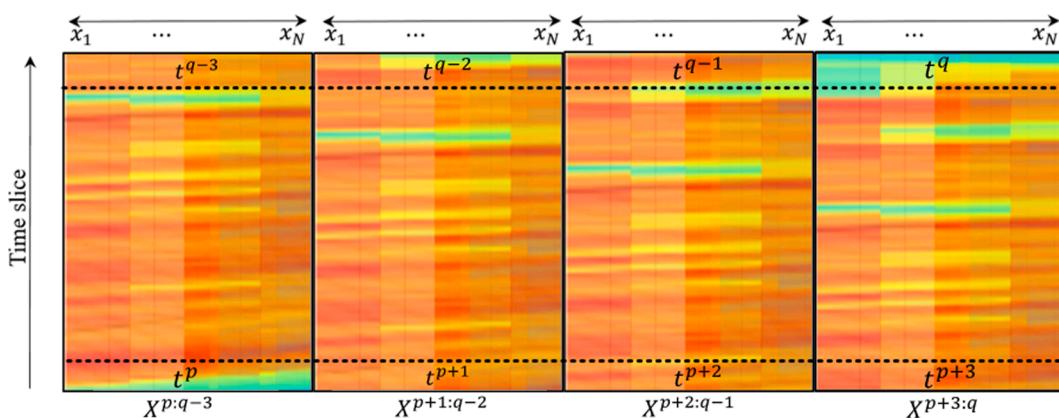


Fig. 3. The new data set with $M = 3$.

optimization of multiple parent–child node pairs $\langle j, i \rangle, j \neq k$. For each $\langle j, i \rangle$, the stochastic process in which x_j exerts an effect on x_i , is most suitable to be described by a multivariate Gaussian distribution (Chen et al., 2017) because the parent node j is not necessarily unique. Therefore, the conditional probability distribution of node pair $\langle j, i \rangle$ considering the observation data $D = [X^{p:q-3}, X^{p+1:q-2}]$ can be written as:

$$P(x_i, x_j | D) = (2\pi)^{-N/2} |C|^{-1/2} \exp \left\{ \frac{D_i C^{-1} (D_i)^T}{-2} \right\} \quad (6)$$

where $C = \text{Cov}(D_{ij})$ denotes the covariance, and N is equal to $q - M - p + 1$.

The score function corresponding to the entire \mathcal{G}^m should be the joint probability of all node pairs. Given the data D^m , this function can be written as:

$$\text{Score}(\mathcal{G}^{3,1} | D) = \frac{P(\mathcal{G}^{3,1}) P(D | \mathcal{G}^{3,1})}{P(D)} = \frac{P(\mathcal{G}^{3,1})}{P(D)} \prod_i P(x_i, x_j | D_{ij}). \quad (7)$$

In the above formula, $\frac{P(\mathcal{G}^m)}{P(D^m)}$ is a constant because the formation of each structure \mathcal{G}^m is equally possible. Therefore, Eq. (7) is equivalent to:

$$\text{Score}(\mathcal{G}^{3,1} | D) = \prod_i P(x^i, x^j | D_{ij}). \quad (8)$$

Step 3: Structure search process. After the scoring function is defined, structure learning can be transformed into a heuristic search problem in the structure space. The max-min hill-climbing method (Tsamardinos et al., 2006) is applied to traverse the possible network structure while satisfying the constraints in *step 1*. The one with the largest score is determined as the optimal structure \mathcal{G}_o^m , which is then converted to the adjacency matrix required by the GCN module as follows.

$$\mathbf{A}^{3,1} = \mathcal{G}_o^{3,1} = \underset{\mathcal{G}^{3,1}}{\text{argmaxScore}}(\mathcal{G}^{3,1} | D). \quad (9)$$

For any two-time slice DAG network, repeat the above steps 2 and 3 to obtain the corresponding adjacency matrix $\mathbf{A}^{M,m}, m \leq M$.

3.2.3. Stacked GCN module

The current congestion state will be affected by the previous state of the upstream section. The state of the road section is likely to be affected by that of the remote road section as the time lag grows. To this end, a stacked GCN is built as shown in Fig. 2c, where each layer uses a different time lag. The top layer of graph convolutions will aggregate the earlier traffic state when inferring the current traffic state. Similarly, the bottom layer of graph convolutions will only aggregate the traffic state of the previous one-time slice. The feature output for a layer with a time lag of M is:

$$\mathbf{H}_M^{t+1} = \prod_{m=1}^M \gamma \mathbf{A}^{M,m} \mathbf{X}^{t-M} \mathbf{W}^{M,m} \quad (10)$$

Finally, a fully connection layer is used to map the feature output of each layer to the sample space in the form of a weighted sum as follows:

$$\widehat{\mathbf{X}}^{t+1} = \sum_M \omega_M \mathbf{H}_M^{t+1} \quad (11)$$

where ω_M denotes the learnable parameters in the fully connected layer.

4. Data source

The study area is located in a regional road network in Beijing. This road network is divided into 2525 sections according to the topology. The Alibaba AutoNavi map, as the largest map navigation provider in China, provides the time-dependent raw traffic data for each road section. For research purposes, the following tasks are still needed: (1) to characterize the congestion state, (2) to determine the testbed, and (3) to collect data for model training and testing.

4.1. Congestion characterization

This paper uses travel speed to characterize the traffic state. Speed data normalization is performed because of the influence of the road network heterogeneity, as shown in Eq. (12).

$$C_i(t) = \frac{\widehat{v}_i(t)}{v_i^{\max}}, \quad (12)$$

where $\widehat{v}_i(t)$ represents the average speed of road i in the t -th time slice, and v_i^{\max} is a fixed value indicating the upper speed limit of the road section. In addition, the length of the time slice is 5 min to weaken the interference of random factors.

Thus, $C_i(t)$ is still a continuous value between [0,1], which can linearly characterize crowdedness. A slow $C_i(t)$ leads to low speed

and severe congestion. Saberi et al. (2019) indicated that the congestion state can be determined by:

$$C_i(t). < \rho, \rho \in (0, 1). \quad (13)$$

In addition to speed data, traffic log data are also necessary to determine a reasonable threshold ρ . The traffic log, which is also provided by Alibaba AutoNavi, is a type of user-generated content, which includes recurrent and incident congestion records caused by traffic accidents and road constructions. Travelers using AutoNavi Apps voluntarily report congestion and traffic incidents encountered to the platform for information sharing. Combined with the official incidents provided by the local traffic authorities, these logs are subsequently supervised and verified by other users until they are cleared. Therefore, the information of such log data is detailed and credible. As shown in Table 1, each log contains the ID, congestion type, start and end times, location (road ID), and related textual description of the log. Among this information, the location in the log is selected as the source of congestion.

A total of 907 recurrent congestion and 161 incident congestion reports were collected from the traffic log in December 2019. The $C_i(t)$ value for each congestion log is calculated during the congestion duration. Fig. 4 shows the histogram of the obtained $C_i(t)$ sequence, in which the maximum value of $C_i(t)$ is 0.46. This paper sets the threshold ρ as 0.5 to identify the entire process of congestion from occurrence to dissipation comprehensively. That is, less than $0C_i(t).5$ is used as the criterion to determine the congestion state.

4.2. Testbeds and datasets

Congestion does not spread endlessly due to the shunting effect in the road network. Therefore, using the entire road network as the testbed when analyzing congestion propagation patterns is unnecessary. Reducing the scale of studied road networks introduces two benefits: (1) it effectively avoids the problem of node combination explosion in the congestion propagation structure learning, and (2) substantially shortens the training time of the GCN model. Therefore, a plausible way is to select K upstream road sections adjacent to the congestion source section as a testbed.

First, the location information (road ID) in the traffic log is used to locate the source of congestion. Next, the congestion source is connected with other congested road sections detected by Eq. (13) into a congestion tree according to time validity and spatial connectivity to determine the appropriate K value. Algorithm 1 provides the detailed process of constructing a congestion tree. The data input contains two parts: a congestion source at t period, R_s^t , and the location information of all road sections $R(O, D)$, where (O, D) represents the coordinates of the two endpoints of a road section. In line 1, the congestion tree with a parent node and a flag is defined. The core of the algorithm is the do-while loop structure (line 2 ~ 11), which is used to continuously find the upstream node of the parent node until the flag equals 0. In each sub-loop, we first find the road r upstream of the congestion source R_s . If the traffic state of r at $t+1$ period, r^{t+1} , is congested, then: (1) add r^{t+1} to $CTrees$, (2) let r^{t+1} replace R_s^t as the new parent node, (3) keep the flag equals 1. Otherwise, make flag equal to 0 (line 3 ~ 10).

Algorithm 1. Congestion tree building process.

Input: congestion source R_s^t ; set of road location information $R(O, D)$

Output: congestion trees, $CTrees$

1. Let $CTrees = [R_s^t]$, $flag = 1$
 2. Do{
 3. for each road section r in R :
 4. find the r that meets: $R_s.O == r.D$
 5. if r^{t+1} is in congestion:
 6. add r^{t+1} to $CTrees$
 7. $R_s^t = r^{t+1}$
 8. $flag = 1$
 9. else:
 10. $flag = 0$
 11. }while ($flag == 0$)
 12. Return $CTrees$
-

In the study area, the spatial locations of the 1068 congestion events (including 907 recurrent congestion 161 incident congestion) are

Table 1
Description of traffic log data.

Fields	Examples
Log ID	353,028,590
Congestion type	1: recurrent congestion 2: incident congestion
Start time	2020-12-01 11:47
End time	2020-12-01 12:20
location (road ID)	5,121,408,072,784,281,765
Description	An accident on the G6 Jingzang Expressway (south to north).

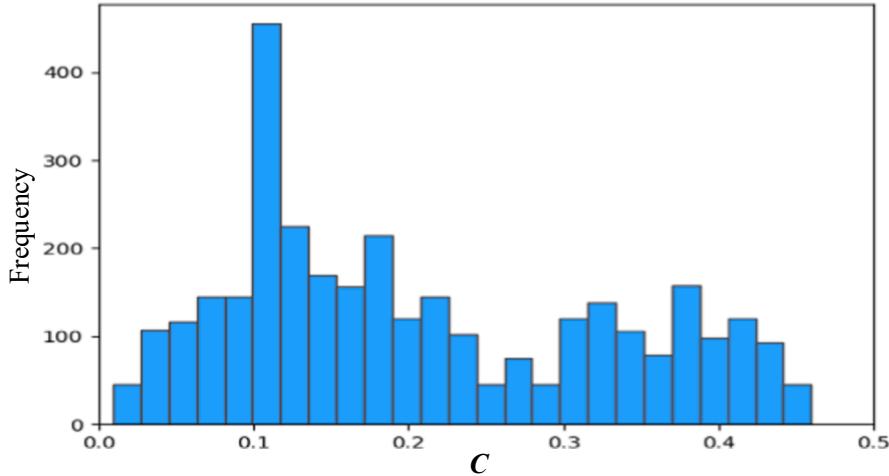


Fig. 4. Histogram of traffic state sequence during the reported congestion.

shown in Fig. 5a. Based on the occurrence time of congestion sources, it was found that almost all 907 recurrent congestions occurred during peak hours. Among the 161 incident congestions, 64 incidents occurred during off-peak hours, and the remaining 97 incidents were triggered by recurrent congestion. Taking the event location as the congestion source, with the help of Algorithm 1, we can generate a total of 971 congestion trees, including 810 trees with a single recurrent congestion source, 64 trees with a single incident congestion source, and 97 trees with double-congestion sources (i.e., recurrent congestion with an incident). The statistical results show that the tree with a single congestion source contains up to 13 road sections, and the tree with a double-congestion source contains up to 28 road sections. In order to maintain the uniformity of the data input when preparing the testbed, we set the K value under the two conditions to 15 and 30, respectively. Fig. 5b, 5c and 5d demonstrate three exemplary testbeds with different congestion sources, where testbed 1 refers to a single recurrent congestion source at link 0, testbed 2 refers to a single incident congestion source at link 0, and testbed 3 corresponds to a scenario with a double-congestion source at link 0 and 30 respectively. This paper aims to use the observed traffic state at the congestion source as the evidence variable (embedded node) to infer congestion propagation in the testbed based on the learned rule.

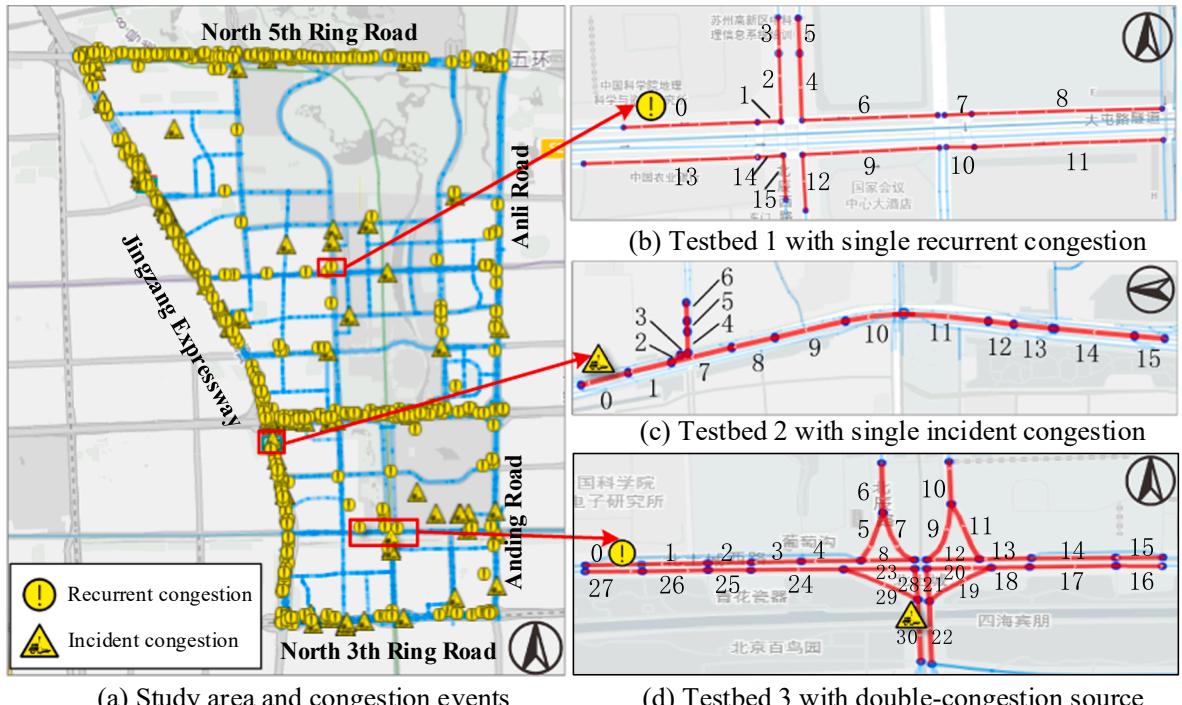


Fig. 5. Study area and three testbeds.

The traffic state values of all the 971 testbeds during the congestion duration are used as the test set. The training set corresponding to each testbed can then be collected from the historical data (2019/https://doi.org/10.2019/11) according to Eq. (13). In the training set, 20% of the data is randomly selected for cross-validation. Additionally, interpolation is performed with the mean value of its previous and next speeds values to handle the missing data.

5. Experimental results

This section compares the proposed model with baseline models to verify whether the learning of congestion propagation rules improves the model performance on traffic state inference. The influence of the road network topology and duration of the congestion source on the congestion propagation is also explored by simulating different congestion scenarios based on the learned congestion propagation rules.

5.1. Baseline models

Since the congestion indicator is derived from the traffic speed, the congestion propagation inference is realized inevitably through traffic speed forecasting under special circumstances (e.g., travel peaks, traffic incidents). Many studies have involved congestion propagation inference or short-term traffic forecasting. The proposed DBGCN model cannot be compared with all the models designed thus far. Our primary purpose is to confirm whether the dynamic adjacent matrix learned from DAG could enhance the inference performance. The second purpose is to explore whether the early traffic state brings benefits to the inference of the current traffic state, which is also the reason for constructing a stacked model framework. For the above two purposes, this paper selects three representative models as benchmarks:

- **DBN** (Fan et al., 2019). DBN is a directed probability graph model that uses probability to describe the transition of different traffic states between adjacent road sections. For continuous feature inputs (such as speed), this transfer process is modeled by a Gaussian mixture model, in which the state of the child node can be inferred by aggregating the information of its parent node.
- **Extended GCN** (Yu et al., 2020). Extended GCN (E-GCN) is one of the state-of-the-art GCN models with a stacked framework, and its basic module follows the STGCN structure (Yu et al., 2017). This model leveraged multiple weight-adjacency matrixes to describe the impact of the oldest state on the current traffic state. However, the connectivity between its nodes is directly predefined following the road network topology.
- **Graph Attention Network (GAN)**. GAN (Velickovic et al., 2017) introduces an attention mechanism based on GCN to obtain an adaptive adjacency matrix, making up for the lack of feature extraction from the predefined topology. The attention weight matrix can be simplified as $e_{ij} = \text{LeakyReLU}(\vec{a}[E_i E_j^T])$, where E_i and E_j represent the cause and effect node embedding with learnable parameters, respectively. $\vec{a}[\bullet]$ is the realization of the edge attention mechanism based on the feedforward network, $\text{LeakyReLU}(\bullet)$ denotes the non-linear activation function. Finally, $\text{softmax}(e_{ij})$ is used to normalize the attention weight.

5.2. Parameter setting

In the data input layer, the state information at the earlier time slice will be aggregated into the current state as time lag (M) increases. The value of M should not be excessively large considering the limited duration of traffic congestion. M is set to 1, 2, and 3 in this paper.

The mean square error (MSE) between the ground truth and the inferred state is used as the loss function in the model training process. The Adam optimization method (Kingma and Ba, 2014) is used to update the parameters of the GCN model, and the learning rate is set to 10^{-3} to accelerate the convergence of the training process. Moreover, the early stopping strategy is adopted to prevent overfitting. The model training process will be aborted if the MSE does not improve significantly (not more than 10^{-6}) during ten consecutive iterations.

5.3. Result analysis

The mean absolute error (MAE) and root-mean-square error (RMSE) are adopted to evaluate the model performance on congestion propagation inference in this study. The performance of the model with three different time lag input data for all 971 testbeds is shown in Fig. 6. It can be clearly seen that the MAE and RMSE values from the DBN model have a high degree of dispersion. This indicates that the DBN model based on the linear relationship between nodes is difficult to describe the complex spatiotemporal process of congestion evolution. In contrast, the MAE and RMSE values from the GCN-based model are relatively concentrated, implying that the depiction of the nonlinear relationship between nodes greatly improves the model performance. In addition, as the time lag increases, more outliers of MAE and RMSE values can be observed. In addition, as the time lag increases, the number of MAE and RMSE outliers (large error values) of each model has increased.

In order to compare model performance in different scenarios more intuitively, we calculated the average and standard deviation (SD) of the MAE and RMSE values from all the testbeds under different time lags, as shown in Table 2. Based on the average values of MAE and RMSE, the proposed DBGCN model outperforms the other three models, suggesting that the learning of the congestion propagation network effectively improves the accuracy of congestion propagation inference. The performance of the DBN model is far

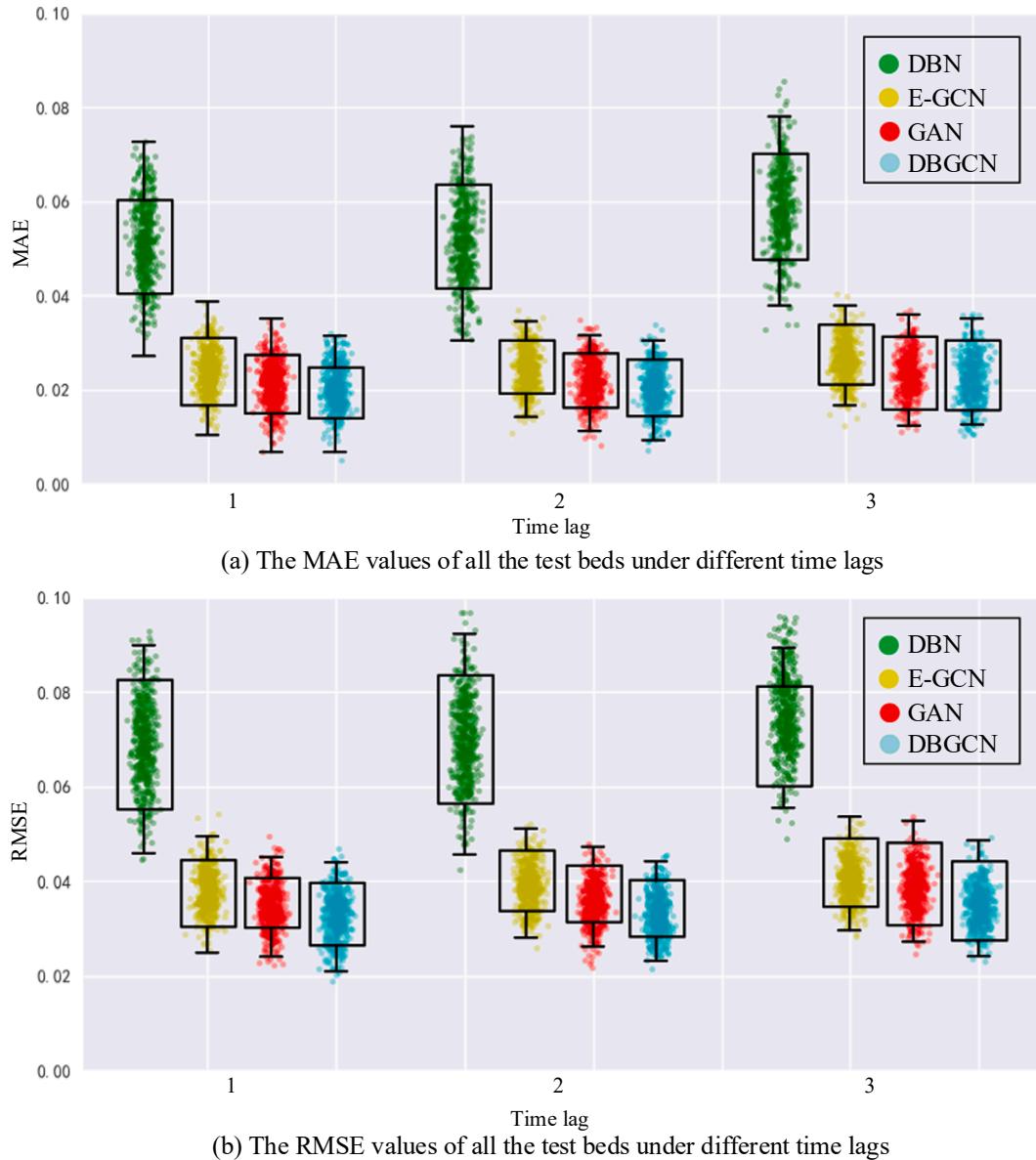


Fig 6. Model performance comparison.

Table 2
Model performance comparison.

Indicator	Time lag	DBN	E-GCN	GAN	DBGCN
Average MAE (SD)	1	0.0507 (0.0109)	0.0239 (0.0068)	0.0212 (0.0061)	0.0203 (0.0057)
	2	0.0518 (0.0142)	0.0246 (0.0072)	0.0219 (0.0065)	0.0206 (0.0062)
	3	0.0585 (0.0165)	0.0266 (0.0079)	0.0234 (0.0071)	0.0227 (0.0068)
Average RMSE (SD)	1	0.0685 (0.0128)	0.0380 (0.0073)	0.0348 (0.0066)	0.0325 (0.0059)
	2	0.0696 (0.0139)	0.0391 (0.0084)	0.0357 (0.0082)	0.0329 (0.0079)
	3	0.0766 (0.0145)	0.0403 (0.0088)	0.0384 (0.0083)	0.0348 (0.0082)

Note: the SD refers to standard deviation.

inferior to the GCN-based models because DBN is essentially a linear expression between nodes. Meanwhile, the deep learning models capture the non-linear relationship between nodes. More specifically, compared with E-GCN with predefined adjacent matrix, GAN with attention mechanism improves model performance, and DGGCN with Bayesian causal inference makes the accuracy of the model further improved. On the other hand, the errors of all four models show a gradually increasing trend as the time lag grows. This trend indicates that the traffic state of one particular node is comprehensively influenced by its one-step parent node (in the time dimension) during congestion propagation. This phenomenon is consistent with the first-order Markov property in Bayesian networks. That is, the current traffic state of the studied object depends only on the last previous state. In addition, the SD changes in different scenarios are consistent with the dispersion degree of the corresponding box plot in Fig. 6.

The performance of the four models is further evaluated on the three testbeds, as shown in Fig. 5. Testbed 1 represents one recurring congestion near an urban road intersection with a duration of 1 h, testbed 2 illustrates non-recurring congestion on the expressway with a time duration of 1.5 h, and testbed 3 refers to a double-congestion source scenario with a time duration of 2 h. Fig. 7 shows the

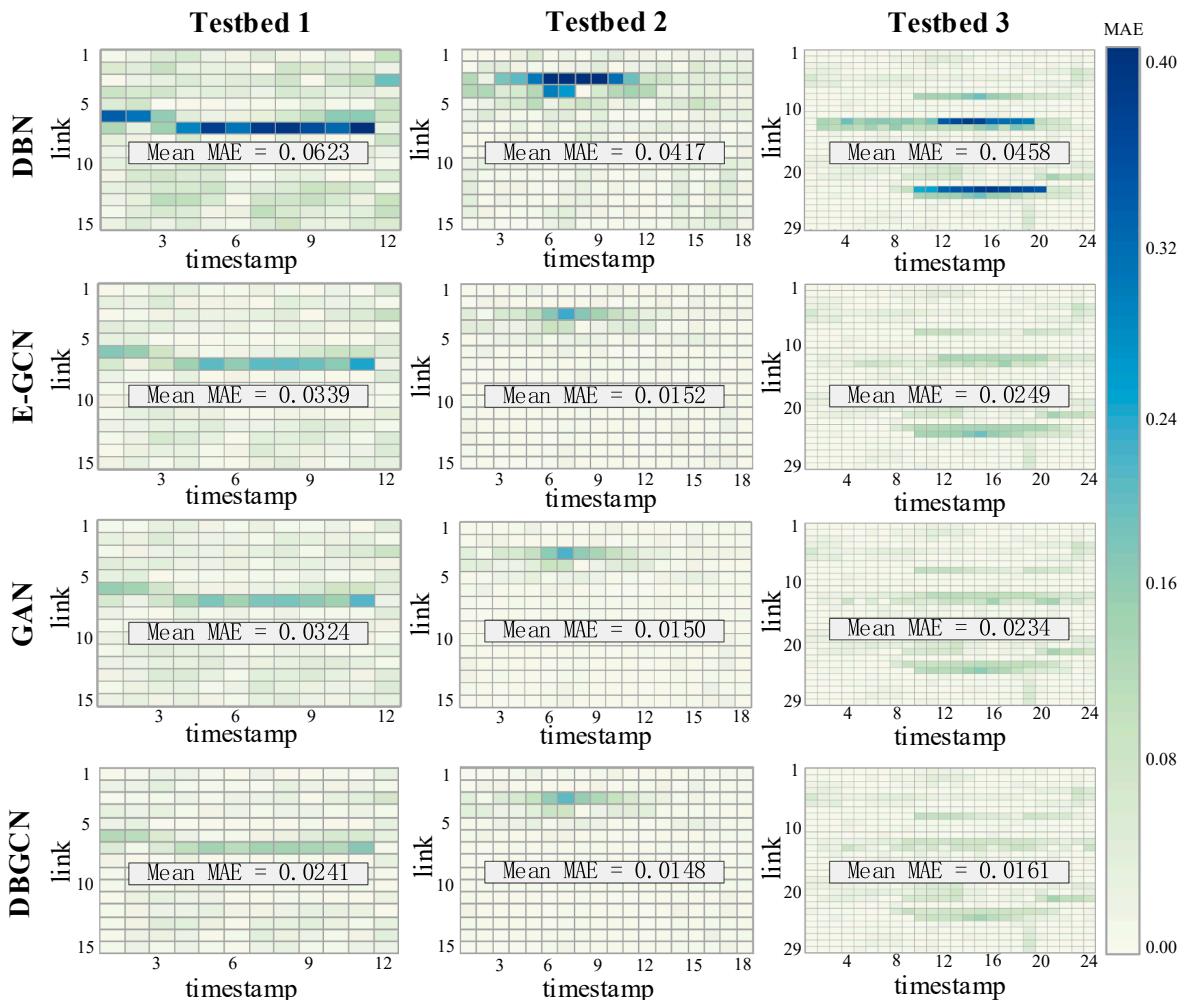


Fig. 7. Performance of the three models on testbeds with different attributes.

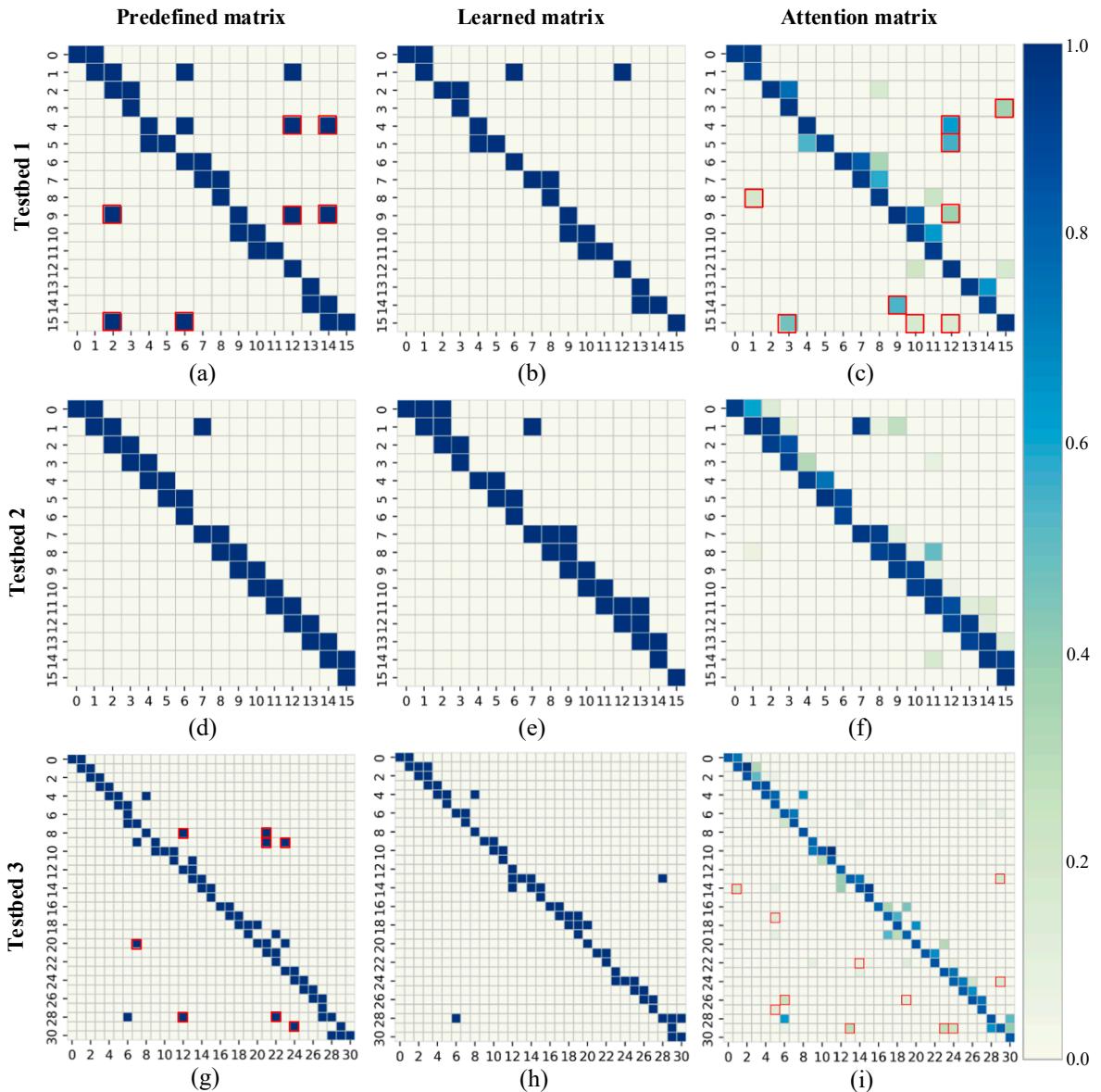


Fig. 8. Predefined and learned adjacent matrixes.

MAE heat maps of the four models in the case of $M = 1$. On testbed 1, the MAE heat maps of E-GCN and GAN are almost the same. Compared with them, the MAE value of DBGCN is significantly reduced. However, the superiority of DBGCN no longer exists on the expressway because its MAE value is only reduced by 0.0004 and 0.0002 compared with E-GCN and GAN. On testbed 3, the performance of each model is very similar to that on testbed 1. The performance differences of the three GCN-based models on different testbeds will be subsequently analyzed through the adjacency matrix.

In general, from the perspective of road network properties, the performance of the four models on the expressway is better than that on the urban roads, mainly attributed to the simple road network structure of the expressway. On the other hand, with reference to the road network topology, the road sections close to the intersection tend to have relatively larger MAE values. More importantly, it is known that the misjudgment of the traffic state of a single link will affect the inference of the entire congestion propagation. According to the link-level MAE value sequence, the proposed DBGCN model can more accurately evaluate the traffic state of each link, which is the key to ensuring the accuracy of congestion propagation inference.

The generated adjacency matrices from the three graph-based models are plotted as shown in Fig. 8 to gain additional insight into their performance differences on different road networks. The subplots on the left side are the adjacency matrices used by E-GCN, which are predefined by the road network topology, stipulating that one node only affects itself and the upstream first-order neighbor node. The subplots in the middle column are the adjacent matrices for DBGCN, which is learned by a hybrid method based on empirical constraints and Bayesian causal inference. The subplots at the right side are the attention matrix derived from GAN. It is worth noting that for the three types of matrices, the first two represent whether there is a causal relationship between nodes (0 or 1), and the last one represents the weight between nodes (0 ~ 1).

On testbed 1, compared with the learned adjacent matrix (Fig. 8b), the predefined adjacency matrix (Fig. 8a) has more connected node pairs (marked by red box). In reality, the extra node pairs in the predefined matrix may not exhibit a causal relationship. For example, at an intersection, the shunting effect will weaken or even eliminate the causal effect between the traffic states of adjacent road sections. Similarly, the attention matrix (Fig. 8c) also shows more weighted node pairs than that in the learned matrix. The extra node pairs in the attention matrix often generate lower weights, and the two nodes inside are not adjacent according to the road topology in Fig. 5a. Therefore, in cases of using the predefined matrix or attention matrix, the nodes in the DAG would gain information from non-influential or weakly influential nodes, thereby resulting in high MAE values. On the contrary, the adjacency matrices of the three models on testbed 2 are highly similar (Fig. 8d, 8e, and 8f), with the exception of a few nodes that influence their second-order neighbor. This is the reason why the learned adjacency matrix on testbed 2 does not produce a significant improvement in model performance. Although testbed 3 corresponds to a larger-scale local network, the difference among the three matrices is very similar to their difference on tested 1, implying that the performance of the proposed DBGCN model is not sensitive to the network complexity.

5.4. Simulation analysis

The DBGCN model can simulate the process of traffic congestion propagation in customized scenarios based on the learned congestion propagation rules in the road network. Therefore, this section will use this simulation mechanism to explore the impact of road topology at the congestion source and duration on traffic congestion propagation. To this end, two scenarios are designed on the

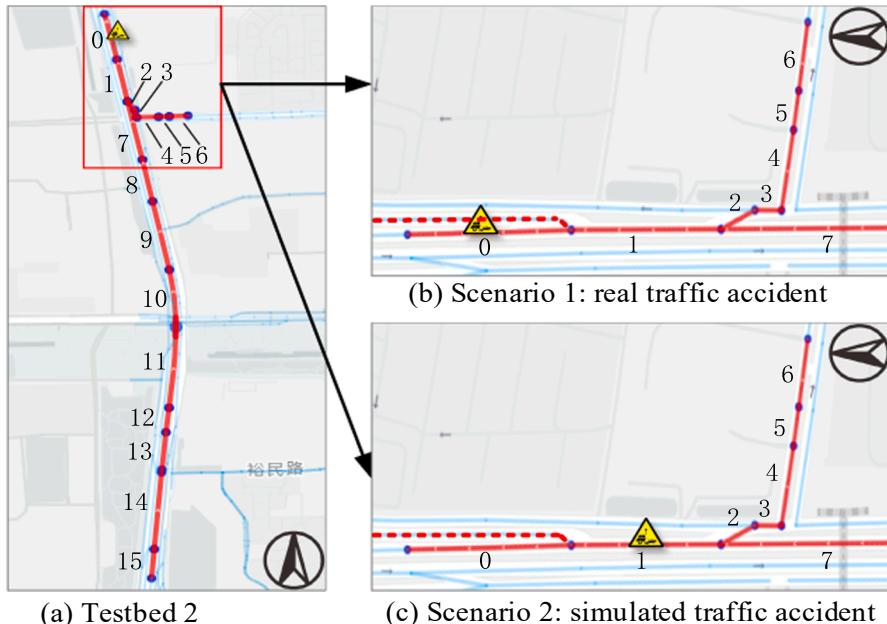


Fig. 9. Two scenarios based on testbed 2.

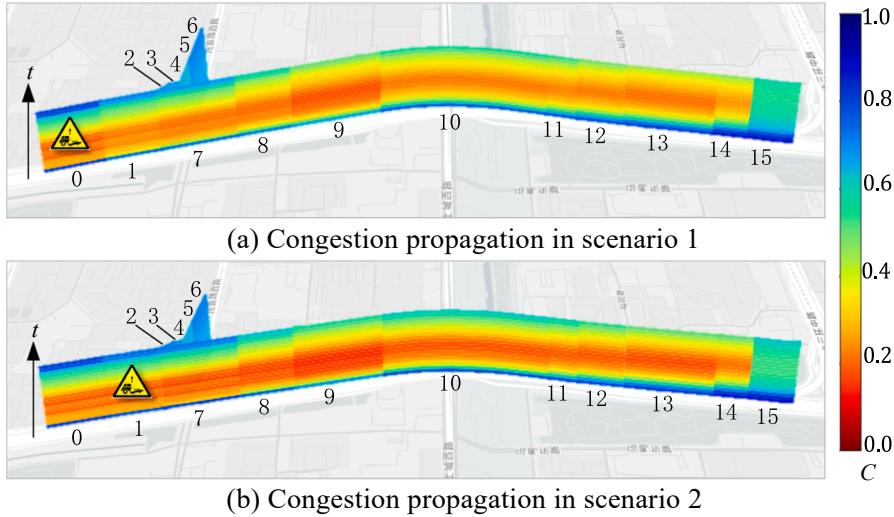


Fig. 10. Inference of congestion propagation in a macroscopic view.

basis of testbed 2 for comparison and analysis, as shown in Fig. 9. Scenario 1 (Fig. 9b) is congestion caused by an actual traffic accident at node 0, while scenario 2 (Fig. 9c) is a simulation scenario, in which the accident in scenario 1 is assumed to occur on node 1. That is, the observation state of node 0 is completely assigned to node 1. Therefore, the simulation process in this paper aims to use nodes 0 and 1 as evidence nodes to infer the propagation process of congestion in the testbed.

First, the congestion propagation inference results of the above two scenarios are drawn macroscopically, as shown in Fig. 10. Congestion gradually propagates from the road section where the accident occurred (node 0, 1) to the upstream road sections over time and eventually stopped at node 14. Moreover, the ordinary urban roads (nodes 4, 5, and 6) connected to node 1 via the ramp (node 2) are unaffected by the congestion. The adjacent matrix (Fig. 8d) shows the absence of connection between nodes 3 and 4. Such an absence means that the information of speed reduction cannot pass from node 3 to the upstream nodes 4, 5, and 6. Similarly, node 15 will not aggregate the congestion information from node 14. This phenomenon shows that traffic congestion will be gradually consumed by the shunting effects in the road network during the propagation process. Therefore, congestion does not propagate endlessly. However, the two scenarios still have certain discrepancies. The congestion severity reflected in scenario 2 is higher than that in scenario 1.

The traffic state evolution on each road section is microscopically plotted in Fig. 11 to gain a clear insight into the heterogeneity between the two scenes. The significant difference can be observed in the first time slice after the accident, as shown by the blue dotted circle in Fig. 11b, g, and h. The normalized speed values (C) of nodes 2, 7, and 8 in scenario 2 more rapidly decrease than those in scenario 1. The road network topology shows the presence of a ramp on the right side of node 0 for leaving the highway, which may be the cause of such a difference. In scenario 1, the exit ramp can rapidly release partial traffic pressure on the upstream section during traffic accidents. However, in scenario 2, the traffic accident is assumed to occur at node 1, preventing the vehicles traveling on the upstream sections from leaving the accident section due to the absence of alternative diversion routes. Thus, the corresponding traffic states immediately deteriorate. By contrast, from the perspective of the time dimension, the spatial range affected by congestion gradually expands over time, but this expansion trend is non-linear. The congestion gradually propagates to long-distance road sections (Fig. 11l, m, and n) during its occurrence. No new road sections are congested as the congestion continues; thus, the normalized speed value (crowdedness) of nodes 3, 4, 5, 6, 14, and 15 is always higher than the threshold 0.5 (Fig. 11c, d, e, f, and o).

The comparison and analysis results of the propagation of congestion in the two scenarios are presented as follows. (1) The change of congestion source location leads to distinct congestion magnitude. (2) After the congestion source appeared, the congestion gradually spread to the upstream section as the congestion source continued. (3) More interestingly, the congestion propagation finally stops at a certain road section despite the congestion source lasts long enough, which is related to the shunting effect in the road network. Therefore, in a local road network, there always be a spatial boundary where congestion propagation stops.

6. Conclusions

This paper proposes a new interoperable deep learning framework, named DBGCN, to infer the propagation of congestion (including recurrent and non-recurrent ones) in the road network. By considering the congestion propagation network as a DAG, Bayesian theory is first used to learn the dynamic propagation structure of congestion from historical congestion data and then obtain the time-varying adjacency matrix. **The learned propagation structure is then converted into an adjacency matrix to construct a stacked GCN framework.** To our knowledge, this study is among the first efforts to integrate Bayesian inference with graph deep learning to infer congestion propagation.

Such an effort has introduced considerable improvements in the task of congestion propagation inference, especially in complex

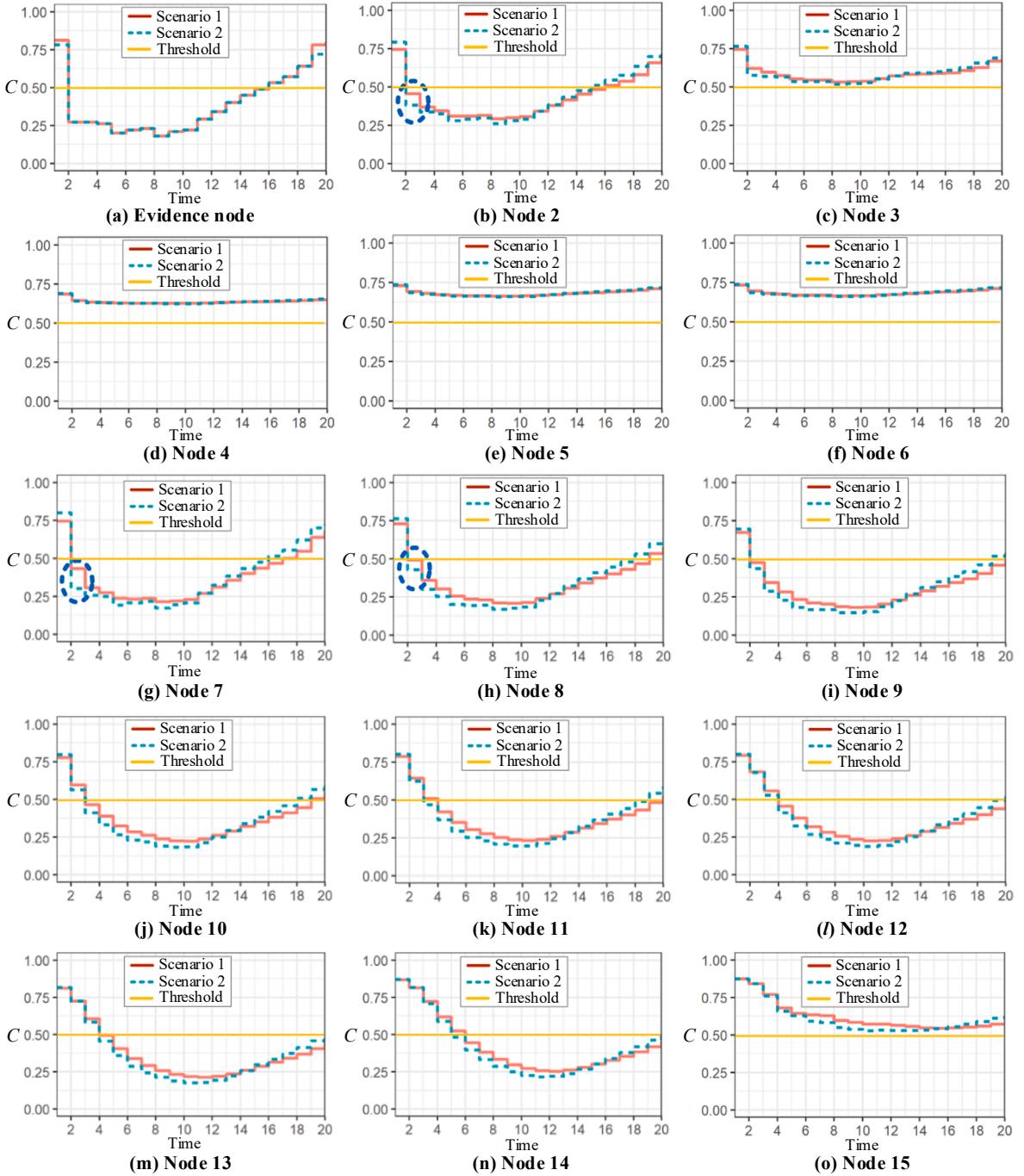


Fig. 11. Congestion evolution process in the two scenarios.

urban road networks, compared with the existing graph-based probability and deep learning models. The difference of congestion propagation rules from prior knowledge can be learned from historical data. Comparison of the experimental results under different time lags reveals that using the traffic state at the last previous time slice (time lag = 1) to infer the current traffic state has the highest accuracy, which is consistent with the first-order Markov property in Bayesian networks. The results from simulation experiments show that changes in the location of congestion source bring about differences in the congestion magnitude. On the other side, the spread of congestion depends on the continuation of the source of congestion, but it will eventually stop at the road section with strong shunting effect. These findings are consistent with the congestion propagation mechanism, which also shows the effectiveness of DBGCN in congestion propagation network learning.

The future work will attempt to incorporate additional road attributes, such as length and capacity, into the model to learn realistic

congestion propagation patterns. In addition, adjusting the model structure to maximize the use of historical data may be an attempt to improve the performance of the model.

CRediT authorship contribution statement

Sen Luan: Methodology, Writing – original draft, Software. **Ruimin Ke:** Formal analysis, Visualization. **Zhou Huang:** Software, Validation. **Xiaolei Ma:** Conceptualization, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This paper is supported by National Key R&D Program of China (2018YFB1601600).

References

- Anbaroglu, B., Heydecker, B., Tao, C., 2014. Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks. *Transportation Research Part C* 48, 47–65.
- Bourrel, E., Henn, V., 2002. Mixing micro and macro representations of traffic flow: a first theoretical step. In: Proceedings of the 9th meeting of the Euro Working Group on Transportation, pp. 610–616.
- Chen, M., Yu, G., Chen, P., Wang, Y., 2017. A copula-based approach for estimating the travel time reliability of urban arterial. *Transportation Research Part C: Emerging Technologies* 82, 1–23.
- Cui, Z., Ke, R., Pu, Z., Wang, Y., 2018. Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. arXiv preprint arXiv:1801.02143.
- Cui, Z., Lin, L., Pu, Z., Wang, Y., 2020. Graph Markov network for traffic forecasting with missing data. *Transportation Research Part C: Emerging Technologies* 117, 102671.
- Daganzo, C.F., 1994. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transportation Research Part B: Methodological* 28 (4), 269–287.
- Defferrard, M., Bresson, X., Vandergheynst, P., 2016. Convolutional neural networks on graphs with fast localized spectral filtering. arXiv preprint arXiv:1606.09375.
- Fan, X., Zhang, J., Shen, Q., 2019. Prediction of Road Congestion Diffusion based on Dynamic Bayesian Networks. *Journal of Physics: Conference Series* 1176, 22046.
- Helbing, D., 2003. A section-based queueing-theoretical traffic model for congestion and travel time analysis in networks. *Journal of Physics A: Mathematical and General* 36 (46), L593.
- Jiang, W., Luo, J., 2021. Graph Neural Network for Traffic Forecasting: A Survey. arXiv preprint arXiv:2101.11174.
- Ke, R., Li, W., Cui, Z., Wang, Y., 2020. Two-stream multi-channel convolutional neural network for multi-lane traffic speed prediction considering traffic volume impact. *Transportation Research Record* 2674 (4), 459–470.
- Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
- Kipf, T.N., Welling, M., 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.
- Koller, D., Friedman, N., 2009. Probabilistic graphical models: principles and techniques. MIT press.
- Laih, C.-H., 1994. Queueing at a bottleneck with single- and multi-step tolls. *Transportation Research Part A Policy & Practice* 28 (3), 197–208.
- Lei, X., Wang, W., Xue, G., Hao, Y., Dai, W., 2015. Discovering Traffic Outlier Causal Relationship Based on Anomalous DAG. Springer, Cham.
- Liang, Y., Cui, Z., Tian, Y., Chen, H., Wang, Y., 2018. A deep generative adversarial architecture for network-wide spatial-temporal traffic-state estimation. *Transport. Res. Rec.* 2672 (45), 87–105.
- Lighthill, M.J., Whitham, G.B., 1955. On kinematic waves II. A theory of traffic flow on long crowded roads. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* 229 (1178), 317–345.
- Liu, Z., Chen, C., Li, L., Zhou, J., Li, X., Song, L., Qi, Y., 2019. Genepath: Graph neural networks with adaptive receptive paths. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 4424–4431.
- Long, J., Gao, Z., Zhao, X., Lian, A., Orenstein, P., 2011. Urban Traffic Jam Simulation Based on the Cell Transmission Model. *Netw. Spat. Econ.* 11 (1), 43–64.
- Luo, Y., Peng, J., Ma, J., 2020. When causal inference meets deep learning. *Nature Machine Intelligence* 2 (8), 426–427.
- Ma, X., Tao, Z., Wang, Y., Yu, H., Wang, Y., 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies* 54, 187–197.
- Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.M., 2017. Geometric deep learning on graphs and manifolds using mixture model cnns. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 5115–5124.
- Newell, G.F., 1993. A simplified theory of kinematic waves in highway traffic, part I: General theory. *Transportation Research Part B: Methodological* 27 (4), 281–287.
- Nguyen, H., Liu, W., Chen, F., 2017. Discovering Congestion Propagation Patterns in Spatio-Temporal Traffic Data. *IEEE Transactions on Big Data* 1.
- Qi, H.S., Wang, D.H., Peng, C., 2013. Formation and Propagation of Local Traffic Jam. *Discrete Dynamics in Nature and Society*, 2013(3-23) 2013 (2 b).
- Ran, T., Xi, Y., Li, D., 2016. Simulation analysis on urban traffic congestion propagation based on complex network. *2016 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*.
- Richards, P.I., 1956. Shock waves on the highway. *Oper. Res.* 4 (1), 42–51.
- Saberi, M., Hamedmoghadam, H., Ashfaq, M., et al., 2020. A simple contagion process describes spreading of traffic jams in urban networks. *Nat. Commun.* 11 (1), 1–9.
- Song, C., Lin, Y., Guo, S., Wan, H., 2020. Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (1), 914–921.
- Spirites, P., Meek, C., 1995. Learning Bayesian networks with discrete variables from data., KDD, pp. 294–299.

- Sun, S., Zhang, C., Yu, G., 2006. A Bayesian network approach to traffic flow forecasting. *Ieee T. Intell. Transp.* 7 (1), 124–132.
- Treiber, M., Hennecke, A., Helbing, D., 2000. Congested Traffic States in Empirical Observations and Microscopic Simulations. *Physical Review E* 62, 1805–1824.
- Tsamardinos, I., Brown, L.E., Aliferis, C.F., 2006. The max-min hill-climbing Bayesian network structure learning algorithm. *Mach. Learn.* 65 (1), 31–78.
- Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.
- Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y., 2020. A comprehensive survey on graph neural networks. *Ieee T. Neur. Net. Lear.*
- Wu, Z., Pan, S., Long, G., Jiang, J., Zhang, C., 2019. Graph wavenet for deep spatial-temporal graph modeling. *arXiv preprint arXiv:1906.00121*.
- Yang, H., Liu, C., Gottsacker, C., Ban, X., Zhang, C., Wang, Y., 2019. Cell-speed prediction neural network (cpnn): A deep learning approach for trip-based speed prediction.
- Yu, B., Lee, Y., Sohn, K., 2020. Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (GCN). *Transportation research part C: emerging technologies* 114, 189–204.
- Yu, B., Yin, H., Zhu, Z., 2017. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting.
- Zhang, A., Gao, Z., 2012. CTM-based Propagation of Non-recurrent Congestion and Location of Variable Message Sign. *Computational Sciences and Optimization (CSO), 2012 Fifth International Joint Conference on*.
- Zhang, Q., Chang, J., Meng, G., Xiang, S., Pan, C., 2020. Spatio-temporal graph structure learning for traffic forecasting. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1177–1185.
- Zhao, B., Xu, C., Liu, S., 2017. A data-driven congestion diffusion model for characterizing traffic in metrocity scales. *2017 IEEE International Conference on Big Data (Big Data)*.