IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 12 December 2023, accepted 10 January 2024, date of publication 23 January 2024, date of current version 13 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3357783

RESEARCH ARTICLE

Spatio-Temporal Contextual Conditions Causality and Spread Delay-Aware Modeling for Traffic Flow Prediction

YIJUN XIONG[1,2,3] AND HUAJUN WANG[3]
[1]Business School, Chengdu University, Chengdu 610106, China
[2]Research Center of Chengdu–Chongqing Economic Circle and Chengdu Metropolitan Area Construction, Chengdu University, Chengdu 610106, China
[3]College of Geophysics, Chengdu University of Technology, Chengdu 610059, China

Corresponding author: Huajun Wang (wanghuajun03@163.com)

ABSTRACT Mobility is essential for all of us, and the daily routine of the majority is impacted by vehicular transportation. Thus, the ability to predict traffic flow is a challenging task in the field of intelligent transportation systems. However, achieving precise predictions of the state of traffic is a complex undertaking, there are two challenges: 1) Existing studies do not explicitly account for the causal influence of the ‘‘trigger effect’’ from contextual conditions on spatial dependencies. 2) Prior methods ignore the fact that there is a time delay in the spread of information in large-scale regions. To address these limitations, we present a novel Graph Structural Causality Spread Delay-aware Model (i.e., GSCSDM) for accurate traffic flow prediction. First, we develop a contextual causality graph that learns the spatial graph structure under the ‘‘triggering effect’’. Second, we present a spread time-delay module that captures the information spread delaying triggered by contextual conditions in global regions. Furthermore, we construct a multi-graph fusion matrix to extract spatial correlation from diverse perspectives, which enhances the understanding of regions’ state interaction. Experiments on two real datasets demonstrate that GSCSDM significantly outperforms the state-of-the-art methods. Since the ‘‘trigger effect’’ widely exists in practical datasets, the proposed framework may also cast light on other spatial-temporal applications.

INDEX TERMS Causality graph, contextual conditions, multi-graph convolution, spread time-delay.

I. INTRODUCTION

Traffic flow prediction is vital for smart city initiatives and spatio-temporal data mining. The field of traffic flow prediction has been extensively studied for many years, dating back to the 1930s, and has been extensively studied and practiced with successful results. This involves predicting future traffic volume based on past observations [1], [2], [3]. This functionality powers a variety of services related to road management, city planning, public safety, vehicle dispatching, and congestion relief [4]. Accurate traffic forecasting, as a fundamental element of intelligent transportation systems, has garnered growing attention for its timely delivery. Besides, it is also crucial for dynamic traffic management, as it enables recommending time-saving routes for drivers. However, traffic prediction is challenging due to the presence of complex spatio-temporal relationships influenced by contextual events (e.g., traffic accidents and weather conditions). Indeed, prediction complexity arises from the varying traffic patterns that stem from responses to real-time contextual conditions. As a result, analyzing external events can assist cities in handling traffic more effectively.

Traffic prediction methods relied on statistics [5] and machine learning [6] in the past. The former commonly adopt statistical theory for user behavior simulation in traffic, such as Auto-regressive Integrated Moving Average (ARIMA) and Kalman filtering. ARIMA [7] depends on assumptions related to weak stationarity and then fits historical data to forecast future points. The Kalman filter method [8] adopts the previously estimated state and the current state to obtain

The associate editor coordinating the review of this manuscript and approving it for publication was Yanli Xu.

the optimal prediction. However, these approaches are ineffective in capturing the non-linearity feature in each traffic flow, as they often depend on assumptions related to weak stationarity. Regarding the latter, many early studies rely heavily on feature engineering, such as K Nearest Neighbors (KNN) and Support Vector Regression (SVR). KNN [9] matches the current value with the K-nearest neighbors to predict the next state. SVR [10] minimizes the prediction error by bringing sample sets close to the hyperplane. However, they do not model the complex spatio-temporal correlations in traffic networks, significantly reducing the accuracy of prediction.

Recently, deep learning has demonstrated powerful higher-order feature learning capabilities in areas such as computer vision and natural language, so it is also widely used in traffic flow prediction. These methods are mainly relied on Convolutional Neural Networks (CNN), they capture static spatial relationships based on Euclidean distances by dividing the road network into regular grids. It can encode graph structure into a lower-dimensional space to capture latent features. For example, Traffic Flow Forecasting Network (TFFNet) [11] adopts deep convolutional networks for capturing hierarchical spatial structure across different depths from local to global regions. Reference [12] enhances the associative memory of features in recurrent neural networks, thereby improving the accuracy of predictions. Recurrent Neural Network Long-term Forecasting (RNN-LF) [13] utilizes various data sources to forecast long-term traffic flow data. This framework is built upon a recurrent neural network with input incorporating external factors such as weather and accidents. Nevertheless, CNN cannot accurately reflect spatial correlations with complex non-Euclidean graph structures based on regular grids.

In the field of spatial-temporal relationship modeling, there has been increased attention on Graph Neural Networks (GNNs) [14], [15], [16], [17], especially Graph Convolution Network (GCN) [18], [19], [20], [21]. Prior Knowledge Enhanced Time-varying Graph Convolution Network (PKET-GCN) [22] extracts dynamic and static spatial features between nodes by characterizing external and internal factors affecting them. Attention based Spatial-temporal Graph Convolutional Networks (ASTGCN) [23] proposes a novel spatio-temporal graph convolutional network to capture complex dynamic spatio-temporal relationships and patterns to represent the temporal features. Attention-based Spatial–temporal Adaptive Dual-graph Convolutional Network (ASTA-DGCN) [24] constructs a dual-graph convolution and a sequential convolution to capture the dynamic spatio-temporal patterns between different locations based on latent temporal correlation and hidden weights. Deep Hybrid Spatio-temporal Dynamic Neural Network (DHSTNet) [25] applies different weights to each branch and combines the results of the four features for predicting traffic crowd flows. In other words, GNNs have the ability to not just represent graph structures but also retain the features of individual nodes.

Despite most of these GNNs methods focusing on dynamic graphs with structures evolved in time, these methods have at least two disadvantages in predicting traffic flow: 1) The ability to model complex nonlinear causality in spatial correlation is inadequate. As shown in Figure 1(a), when influenced by the same external factors, people tend to choose the same traffic patterns. This results in a strong spatial-temporal dependence of traffic flows in different locations. However, traditional traffic flow prediction methods usually simply integrate data and contextual conditions into the model as inter-factors but ignore causality between them [26], [27]. More information entanglement may result in capturing spurious spatial correlations. 2) The graph structure that is influenced by contextual conditions exhibits the effect of information-passing time-delay. As shown in Figure 1(b), nodes A and B are not necessarily adjacent nodes and may experience a delay effect on traffic patterns of several minutes when an external factor happens in one place. This delay effect is ignored by the existing methods based on the instant information-passing mechanism [28], [29]. More precisely, a contextual condition intervenes in the transmission of spatial information between nodes at t_0, causing the one node's tendency to link with other nodes to evolve to become higher or lower from t_1 to t_2. In other words, we need to further analyze the "triggering effect" from contextual conditions on dynamic graph structure.

To tackle the aforementioned gaps, we present an innovative approach known as the Graph Structural Causality Spread Delay-aware Model (GSCSDM) for traffic flow prediction. To be specific, we develop a contextual causality graph to capture the underlying causality from contextual conditions. We further integrate multi-hop nodes' historical traffic patterns into memory items to explicitly model the time-delay of spatial information passing in global regions. In addition, we adopt the multi-graph convolution to reflect the spatial relationship from diverse perspectives.

We summarize our contributions as follows:

- We propose a new network model, called GSCSDM, to efficiently detect causality and information-passing delaying from contextual conditions on the spatial correlation for traffic flow prediction. This enables more precise modeling of the context conditions of traffic data to improve prediction accuracy.
- We develop a novel contextual causality graph (CCG) to capture latent spatial relationships by encoding the "trigger effect". To the best of our knowledge, our work is the first one that successfully applies the contextual causality-based module to traffic flow prediction.
- We present a spread time-delay module to extract the information-passing delaying triggered from contextual conditions by traffic pattern matching. This module enhances the feature's understanding in global regions, thereby improving the prediction performance.
- We construct a multi-graph fusion matrix to extract spatial correlation from diverse perspectives, which

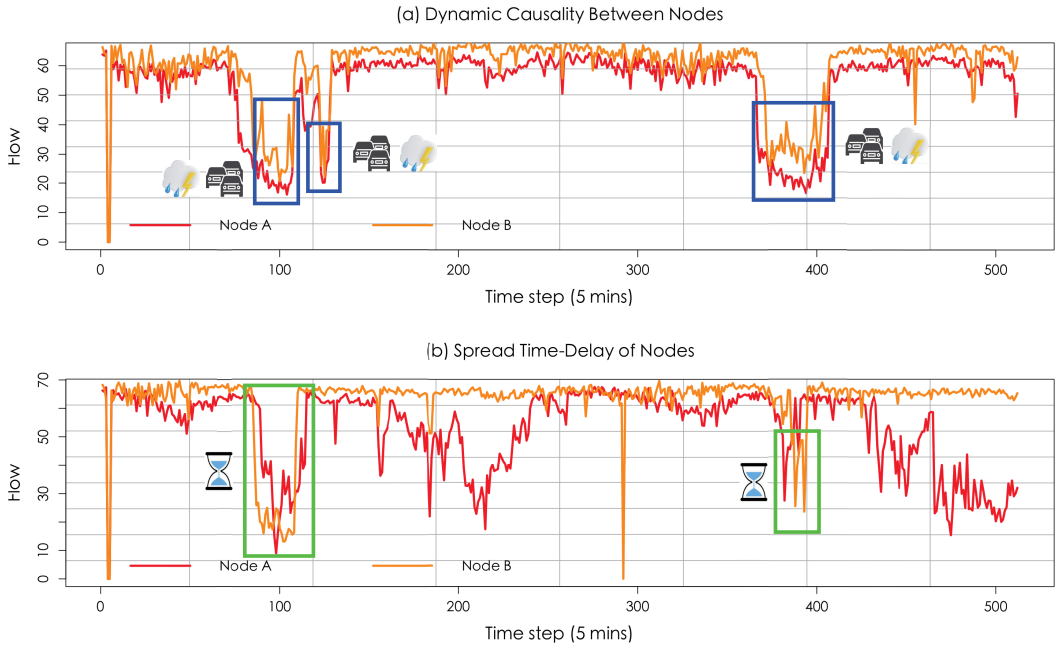


FIGURE 1. **The findings about traffic prediction.**

improves the comprehension of regions' state interaction from static and dynamic graph structure.

- Our proposed algorithm has been extensively tested on real road traffic data sets. The results show that it outperforms several baseline methods, including state-of-the-art algorithms.

II. RELATED WORK

A. TRAFFIC FLOW PREDICTION

Recent studies have utilized a GCN to train node representations that extract spatial relations for traffic prediction. Diffusion Convolutional Recurrent Neural Network (DCRNN) [30] is a deep learning model that integrates spatio-temporal dependencies in traffic flow prediction. Graph Sample and Aggregate (GraphSAGE) [31] uses neighbor sampling and aggregation to process nodes and enable graph convolution networks to operate on large-scale graphs. Attention based Spatial-temporal Graph Convolutional Network (ASTGCN) [32] tackles traffic prediction issues through modeling dynamic spatio-temporal correlation. Adaptive Graph Convolutional Recurrent Network (AGCRN) [33] employs node-level convolution filters according to the node embedding. Coupled Layer-wise Graph Convolution for Transportation Demand Prediction (CCRNN) [34] learns dynamic neighbor correlation by a hierarchical coupling mechanism. Dynamic Graph Convolutional Recurrent Network (DGCRN) [35] handles dynamic graph structure by learning matrices at each recursive step. However, these works only focus on spatial graph structures in temporal continuous scenarios but overlook the contextual conditions, resulting in very limited effects of graph structure. Attention-Based Spatio-Temporal Graph Convolutional Network Considering External Factors (ABSTGCN-EF) [36] models the traffic flow as diffusion on a digraph and extracts the spatial characteristics of traffic flow through GCN. Spatio-temporal Attention Point Processes (APP) [37] proposes a novel model that utilizes attention and point processes to capture exogenous and endogenous factors for traffic congestion prediction. Integrated Spatio-temporal Graph Convolutional Network (ISTGCN) [38] incorporates elements such as weather conditions, traffic accidents, and special events to capture traffic dynamics. These studies only integrated contextual conditions with spatial-temporal graph structure but ignore causality between them.

B. GRAPH STRUCTURE LEARNING

Although, GCN has been widely used for graph structure learning in traffic flow prediction, there are still two problems. Graph Attention Temporal Convolutional Network (GATCN) [39] has captured only the temporal dependency within the road network through variants of Graph Attention Networks (GAT) and Long-Short Term Memory (LSTM). Dynamic Graph Convolutional Recurrent Network (DGCRN) only utilizes a pre-defined adjacency matrix to represent the information transfer process for dynamic node states. In addition, methods such as Spatio-temporal Graph Convolutional Networks (STGCN) [40] and Multi-graph Convolutional Neural Network (MGCNN) [41] that also represent the spatial relationship between roads using a static adjacency matrix fail to capture the dynamic changes in spatial dependency. Although these models have proven effective in certain traffic prediction scenarios, they ignore the dynamic spatial graph structure. Dynamic Spatial-Temporal Aware Graph Neural Network (DSTAGNN) [42] captures the dynamic spatial correlation between nodes based on their past traffic flow data directly. Transformer-enhanced Spatial Temporal Graph Neural Network (DetectorNet) [43] focuses on learning dynamic spatial structure to improve the process based on the diffusion GCN for traffic prediction. In addition, Spatio-Temporal Graph Neural Network (STGNN) [44] proposes a position attention mechanism to gather information from nearby nodes and use it to create a dynamic graph structure. However, the spatial correlation may be affected at some specific time by diverse perspectives, so the effect of time-delay might occur in the spatial information-passing.

III. PRELIMINARIES

A. TASK DEFINITION

This paper is to forecast future traffic flow by utilizing past samples. We denote the road network as a graph $\mathcal{G} = (V, E, A)$, where V represents a set of N nodes, E denotes the set of edges, and $A \in R^{N\times N}$ is a weighted adjacency matrix. GSCSDM learns the temporal correlation of each node individually while preserving the data continuity. The final predict result $\widetilde{\mathcal{X}}$ as follows:

$$\widetilde{\mathcal{X}} = \begin{bmatrix} \widetilde{x}^1_{t+1}, \widetilde{x}^2_{t+1}, \ldots, \widetilde{x}^N_{t+1} \\ \cdots \\ \widetilde{x}^1_{t+\tau}, \widetilde{x}^2_{t+\tau}, \ldots, \widetilde{x}^N_{t+\tau} \end{bmatrix} \tag{1}$$

where $\widetilde{x}$ represents the output of N nodes at the time step $[t+1, \ldots, t+\tau]$, where τ is the forecast horizon. In particular, we uses T as a time window to forecast the observations of horizons τ minutes ahead. It means that the input of GSCSDM is the time window $[t - T + 1, t - T, \ldots, t]$, and the output is the series $[t + 1, \ldots, t + \tau]$. The time series of N nodes make up each time step. GSCSDM learns the spatial correlation on different time windows of each node during the prediction process, then combines the outputs of all nodes to produce the final results. The time series of N nodes make up each time step.

IV. METHODOLOGY

Our proposed GSCSDM can model graph structure subject to "trigger effect" from contextual conditions causality with time-delay effects. Contextual causality graph promote useful latent representations and suppress useless representations under the contextual conditions. Spread time-delay module captures the effect of information-passing with time-delay on spatial correlation. We provide a detailed introduction to the causality and time-delay mechanisms in this section. The overall framework of model is depicted in Figure 2.

A. CONTEXTUAL CAUSALITY GRAPH

In addition to interacting with each other, features of locations with polymorphic state are affected by time-varying contextual conditions in traffic flow network, so spatial dependencies must take contextual conditions and causality into account. Our purpose is to focus on causality induced from time-varying contextual conditions. Thus, as shown in Figure 2, we build a contextual causality graph that closely tightly links contextual conditions with spatial correlation. We generate a corresponding contextual heterogeneous sequence from input data $\mathcal{X}^{T\times N\times D}$ based on various contextual conditions. $\mathcal{X}$ is generated from the historical traffic flow states. T is the length of time window, N is the number of nodes and D is the hidden dimension. From these series, we can easily calculate the corresponding hidden representations, named contextual causality representations(CCR), which is defined as:

$$\begin{cases} t_s = [t_i, t_{i+1}, \ldots, t_{i+s}], ||^{L_e} t_s < T \\ I_s = FCL(||^N PAD(\mathcal{X}(||^{L_e} t_s, n, :))) \end{cases} \tag{2}$$

where $I_s \in R^{T\times N\times d}$, $||^{L_e}$ denotes the concentrate operation for the heterogeneous sequences based on contextual data (e.g., meteorological conditions), which influences traffic flow regularities. t_s is the number of time slots occupied by one contextual condition. Note that, for different nodes n, the lengths of heterogeneous sequences t_s are different, thus we need to adopt the zero-padding operation to ensure that the sequences are equal time window length T. $||^N$ denotes the concentration for the latent representations on different nodes. I_s is used to learn the contextual influence-aware vectored representation of each node. According to the influence of contextual conditions on different nodes, it has the ability to enhance important features and reduce the influence of features that are not relevant to the current task. This allows each node to have unique expression, which is helpful for subsequent graph structure calculation.

We apply Gating Mechanisms (GM) to capture the important-aware temporal trend of each process. This allows each process to make distinctive representations, which is beneficial for subsequent similarity calculations. At each time step, the $\mathcal{X}_t$ and previous time step H_{t-1} are concatenated as the input of gated linear unit:

$$\begin{aligned} q_e = tanh(\psi_1(\mathcal{X}_t||H_{t-1})^T + a)\odot \\ \sigma(\psi_2(\mathcal{X}_t||H_{t-1})^T + b) \in R^{N\times d} \end{aligned} \tag{3}$$

$\odot$ denotes the element-wise product, a and b denote parameters of convolution, ψ_1 and ψ_2 denote the dilated causal convolution (DCC). I_s is used to generate the dynamic latent representation of each node as causal weighted.

$$E_t = I_s(t, :, :) \odot q_e \in R^{N\times d} \tag{4}$$

where E_t is used as input feature for next equation.

B. SPREAD TIME-DELAY

The use of traffic flow forecasting is often time-sensitive and can be affected by contextual conditions like weather and accidents [45]. Traffic patterns in the real-life transportation network typically spread under "trigger effect", and time of propagation to reach the adjacent locations is modeled as time-varying delays. Furthermore, traffic patterns (such as congestion) at a certain location spread along the road (line) on the network, ultimately evolving into regional time-delay propagation [46]. In other words, time-varying delay refers to the inclusion of neighbors' previous state

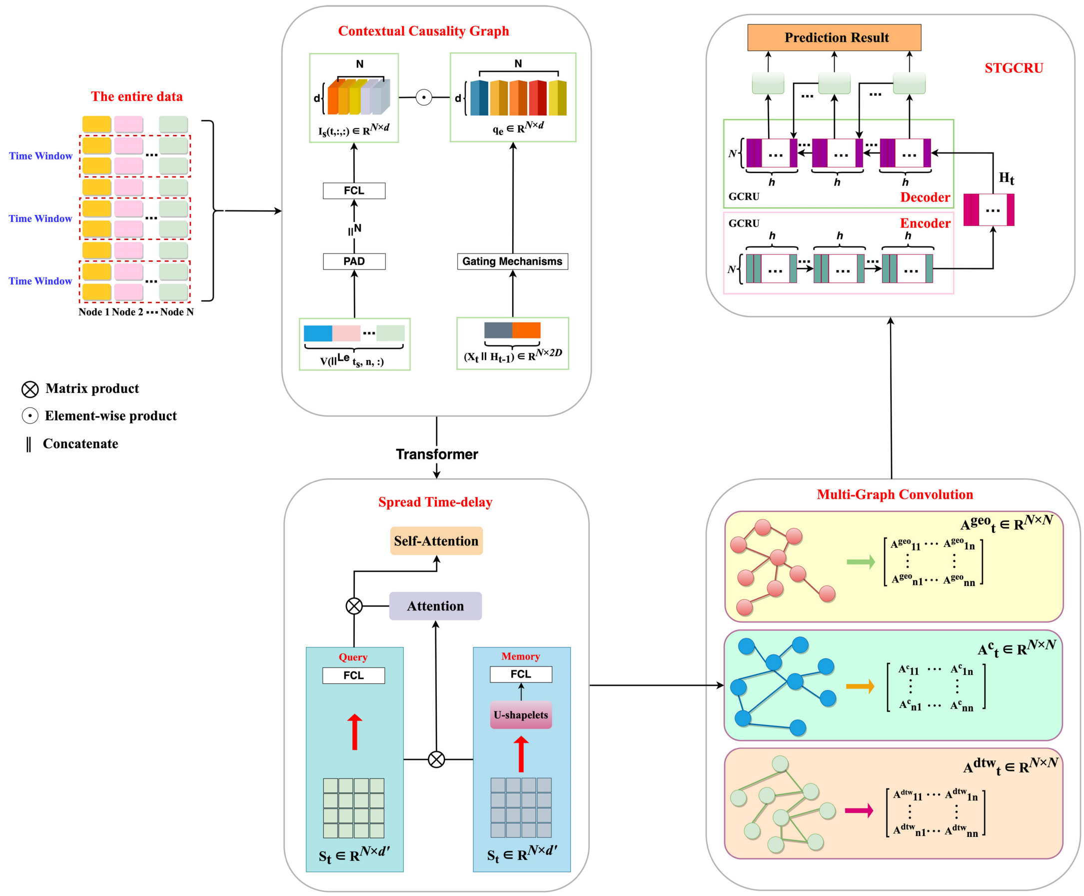


FIGURE 2. **Framework of graph structural causality spread delay-aware model (GSCSDM).**

information and also tends to diffuse to non-adjacent locations. Hence, we capture the global information-passing delaying triggered by contextual conditions based on pattern matching mechanism.

First, the global dependencies representation $S_t \in R^{N\times d'}$ is generated by Transformer layer as:

$$Q_t^S = E_{t::}W_Q^S, K_t^S = E_{t::}W_K^S, V_t^S = E_{t::}W_V^S \tag{5}$$

$$S_t = softmax(\frac{(Q_t^S)(K_t^S)^\top}{\sqrt{D}})V_t^S \tag{6}$$

where W_Q^S,W_K^S and $W_V^S \in R^{d\times d'}$ are learnable parameters and d' is the dimension of the query, key, and value matrix in this work. Next, latent traffic patterns representation from the historical information of all nodes embedding S_t by u-shapelets [47]. This method utilizes the differences of local features among different time series for clustering, and its results are more accurate. Each cluster is represented by its centroid c_m, which is a slice data of length L. Then, $\mathbb{C} = \{c_m|m \in [1, \ldots, K]\} \in R^{K\times L}$ represents the centroid set as representation of different traffic patterns, where K is the total number of centroid. Here, we are inspired by memory networks that memorizes common features in similar traffic pattern [48]. Notably, we generate memory items with global context from S_t, which allows the attention score to effectively reflect traffic pattern similarity between nodes within long-range regions directly, rather than only in neighboring regions. We thereby leverage the idea of memory networks to match the feature with patterns and build a memory pattern matching (MPM), where we set $p^T \in R^{K\times \ddot{d}}$ as memory item, K and $\ddot{d}$ denote the number of items and dimension of each item in the memory, respectively. Finally, we additionally specify the following as the primary purposes of MPM:

$$\begin{cases} q_t^{(i)} = FCL(h_t^{(i)}) \\ p_m^{\mathrm{T}} = FCL(c_m) \\ \alpha_m^{(i)} = softmax(q_t^{(i)} \times p_m) \\ M_t^{(i)} = \sum_{m=1}^{K} \alpha_m^{(i)} \times p_m \end{cases} \tag{7}$$

where we use superscript (i) as row index. For instance, $h_t^{(i)} \in R^{1\times d'}$ represents i-th node vector in $S_t \in R^{N\times d'}$. We directly use the fully connect layer (FCL) to transform the final hidden state $h_t^{(i)}$ to the query $q_t^{(i)} \in R^{1\times \ddot{d}}$ and traffic pattern c_m to the memory $p_m^{\mathrm{T}} \in R^{1\times \ddot{d}}$. $\alpha_m^{(i)}$ is a global attention score that is obtained by matching the $q_t^{(i)}$ with the multi-hop p_m^{T}. Afterwards, augmenting representation is performed followed by concatenation of h_t and M_t to obtain the representation $Z_t = [h_t, M_t] \in R^{N\times(\dot{d}+\ddot{d})}$ that is used as input feature for next equation. Finally, we follow the idea of self-interaction to make the inter-similarity learn randomness and injected dynamic causality into the graph structure:

$$A_t^c = ReLU(\phi(\frac{Z_t Z_t^{\mathrm{T}}}{\sqrt{\dot{d}+\ddot{d}}})) \in R^{N\times N} \tag{8}$$

where ϕ denotes the learnable weighted parameters.

C. MULTI-GRAPH CONVOLUTION

Graph structure is influenced by node distance and feature similarity in addition to contextual conditions and causality [49]. Therefore, we propose a multi-graph convolution network (MGCN) to combine a nodes connection graph, a similarity graph, and an dynamic graph. This network takes into account both static and dynamic relationships between nodes.

The predefined graph structure also shows the hidden potential inter-node dependence from other perspectives. For predefined graph structure, we construct the geographical similarity graph A^{geo} based on Euclidean distance and the temporal feature similarity graph A^{dtw} based on time-series shape. Their respective weighted adjacency matrices as follow:

$$A_{i,j}^{geo} = \begin{cases} exp(-\frac{d_{i,j}^2}{\sigma^2}), & i \neq j \, and \, exp(-\frac{d_{i,j}^2}{\sigma^2})) \leqslant \epsilon \\ 0, & otherwise \end{cases} \tag{9}$$

where ϵ is a threshold. A^{geo} is base on Tobler's first law of geography, roads in close to each other, are likely to share similar usage patterns.

$$A_{i,j}^{dtw} = \begin{cases} 1, & exp(-DTW(i,j)) > \rho \\ 0, & otherwise \end{cases} \tag{10}$$

where ρ is a threshold. A^{dtw} is based on the Dynamic Time Warping (DTW) [51] algorithm can effectively measure similarity of traffic flow.

The MGCN can be defined as follows:

$$\begin{cases} A_t' = \frac{1}{2}(ReLu(A_t) + ReLu(A_t^T)) \in R^{N\times N} \\ L_t = \frac{1}{2} D_t^{-\frac{1}{2}}(D_t - A_t')D_t^{-\frac{1}{2}} \in R^{N\times N} \\ \mathcal{X}_t^n = \psi_1 L_t^{geo}\mathcal{X}_t^{n-1} + \psi_2 L_t^{dtw}\mathcal{X}_t^{n-1} + \\ \psi_3 L_t^c \mathcal{X}_t^{n-1} \in R^{N\times \dot{D}} \\ \mathcal{X}_t^{out} = \sigma(\mathcal{X}_{\star\mathcal{G}}\Theta) = \sigma(Linear(\mathcal{X}_t^n)) \in R^{N\times C} \end{cases} \tag{11}$$

where A_t' ensures symmetry of the original A_t^T, D_t is degree matrix of A_t', L^{geo}, L^{dtw} and L^c denotes symmetric normalized Laplacian, ψ_1, ψ_2, and ψ_3 are trainable parameters that measure the contribution levels of different graphs, n represents the convolutional depth.

We input the results of multi-graph convolution into the spatio-temporal graph convolutional operation and recurrent unit (RU) to denote STGCRU as a basic encoder-decoder unit for prediction:

$$\begin{cases} u_t = \sigma([\hat{X}_t, \hat{H}_{t-1}]W_u + b_u) \\ r_t = \sigma([\hat{X}_t, \hat{H}_{t-1}]W_r + b_r) \\ C_t = tanh([r_t \odot \hat{H}_{t-1}, \hat{X}_t]W_c + b_c) \\ \hat{H}_t = u_t \odot C_t + (1-u_t)\odot \hat{H}_{t-1} \end{cases} \tag{12}$$

$\hat{X}_t$ is the input vector time t, and $\hat{H}_{t-1}$ is the hidden state at time $t-1$. W denotes graph convolution operation, $\odot$ is the element-wise multiplication and σ denotes the nonlinear activation function.

D. TRAINING LOSS

As a supervisory model, to improve its ability to distinguish between different traffic patterns on various nodes, we adjust the memory parameters using two constraints. We integrate the three loss functions: feature loss, contrast loss and consistency loss to train our model [50]:

$$argmin(\mathcal{L}) = argmin(\mathcal{L}_{fea} + k_1\mathcal{L}_{cont} + k_2\mathcal{L}_{cons}) \tag{13}$$

E. FEATURE LOSS

By penalizing the variations in intensity, the feature loss helps the decoder to recreate the traffic flow in a way that is more similar to its original state. Here, RMSE can reflect the error between the decoder output $\widetilde{\mathcal{X}}$ and the ground truth $\mathcal{X}$:

$$\mathcal{L}_{fea} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\widetilde{\mathcal{X}}^i - \mathcal{X}^i)^2} \tag{14}$$

where n is the number of samples. For all metrics, a lower value denotes better performance.

F. CONTRAST LOSS

By promoting queries to be closer to the nearest memory item, the feature contrast loss reduces differences within the same class. It calculates the differences between them using the $\mathcal{L}_2$ norm and applies penalties:

$$\mathcal{L}_{cont} = \sum_t^T \sum_i^N \|q_t^i - p_j\| \tag{15}$$

where j represents the index of the closest item to the query q_t^i. As demonstrated in our experimental results, the act of solely training our model via the compactness loss feature typically results in homogenized items and condensed queries within the embedding space. This, however, results in a loss of capacity to capture diverse normal patterns.

G. CONSISTENCY LOSS

All items exhibit a high degree of similarity. It should be noted, however, that the items within the memory require sufficient separation from each other to effectively account for multiple normal data patterns. Thus, feature separateness loss is denoted as:

$$\mathcal{L}_{cons} = \sum_{t}^{T}\sum_{i}^{N}[\|q_t^i - p_j\| - \|q_t^i - p_e\| + \alpha] \tag{16}$$

where the query q_t^i is established as the query, alongside its closest item p_j and secondary nearest item p_e, which serve as anchoring points for the positive and hard negative samples, respectively.

V. EXPERIMENTS

In this section, we perform extensive experiments on two public datasets and assess our framework in relation to downstream prediction tasks. The results indicate that GSCSDM surpasses state-of-the-art models by a significant margin.

A. DATASETS

We used two datasets: METR-LA and PEMS-BAY, which contain the traffic speed data from 207 sensors in Los Angeles ranging from 1-3-2012 to 30-6-2012 and 325 sensors in Bay Area ranging from 1-1-2017 to 31-5-2017 respectively. Meteorological data comes from the NCEI website (National Centers for Environmental Information). Table 1 shows the details of the two datasets. Nodes data was collected and combined in 5 minutes intervals. We use a sliding window strategy to generate samples, and then split each data set into the training, validation, and test sets with a ratio of 7:1:2 follow the tradition.

B. PARAMETER SETTINGS

For the two datasets, we used the first 12 time steps to predict the next 12 time steps. The Adam optimizer was employed with a learning rate of 0.01 and a batch size of 64 in the present study. The optimizer was subject to early stopping if the validation error converged terminated after 100 epochs. The evaluation of the model performance was based on the three metrics: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). All experiments are implemented in Python 3.10 software with two NVIDIA GeForce RTX 3080 graphics cards.

C. BASELINES

In the experiments, we compare our method with ten state of-the-art methods:

- **HA** [7]: HA considers traffic flow as a seasonal pattern and forecasts future flow based on the average of past seasons.
- **FC-LSTM** [52]: Sequence To Sequence Learning With Neural Networks (FC-LSTM) is a large deep LSTM, which belong to a special kind of RNN models.
- **STGCN** [40]: STGCN employs graph convolution and gated causal convolution instead of LSTM or GRU.
- **AGCRN** [33]: AGCRN uses adaptive modules to augment graph convolution and integrate them into RNN.
- **DCRNN** [30]: DCRNN captures spatio-temporal dependence through bi-directional random walks and encoder-decoder.
- **ST-LGSL** [53]: Spatio-Temporal Latent Graph Structure Learning (ST-LGSL) is trained on the entire dataset, using graph nodes with time-aware features.
- **Graph WaveNet** [54]: Graph WaveNet proposed wave network, which incorporated an dynamic adjacency matrix into convolution layers.
- **MTGNN** [55]: Multivariate Time Series Forecasting with Graph Neural Networks (MTGNN) introduces a unique combination of mix-hop spread and dilated inception layers to capture self-adaptive spatio-temporal dependence.
- **GMAN** [56]: Graph Multi-Attention Network (GMAN) uses an Encoder-decoder architecture, which consists of multiple attention mechanisms.
- **SLC** [57]: Spatio-Temporal Graph Structure Learning (SLC) expands the capabilities of a traditional CNN to learn structure of network with graphs.

D. COMPARISON WITH BASELINES

Table 2 displays the prediction accuracy of various models on two datasets at different time intervals (15 minutes, 30 minutes and 60 minutes) and in general. GSCSDM exhibits superior performance in long-term prediction.

The conventional machine learning approach, HA, is prone to inadequate prediction performance owing to its deficient nonlinear representation capability. In contrast, deep learning strategies exhibit impressive nonlinear representation potential. Within the purview of deep learning methodologies, FC-LSTM, despite its temporal correlation handling ability, exhibits deficient predictive efficiency due to its exclusion of spatial correlations. STGCN, AGCRN, DCRNN, ST-LGSL, DetectorNet, Graph WaveNet, MTGNN, and GMAN enhanced prediction accuracy by incorporating spatio-temporal dependency. The more recent model SLCNN has a better performance in a baseline approach. At training time, our model continuously optimizes and tends to converge during training.

Based on our observations, our proposed framework GSCSDM has generally outperformed the other models on two datasets in 12 horizons. This highlights the superiority of our approach. The reason is that STGCN, AGCRN, DCRNN, and MTGNN only used a static adjacency matrix. DetectorNet, GMAN, Graph WaveNet, and SLCNN utilized both dynamic and static adjacency matrix, but they are all based on simple tensor multiplication or attention mechanism. In addition, all of these methods ignored the impact of contextual conditions. At training time, our model continuously optimizes and tends to converge during training. The possible reason is that METR-LA and PEMS-BAY are relatively small-scale datasets, and the strong feature similarity is captured by GSCSDM. The only shortcoming is that our model is only slightly weaker in horizon 12 than SLCNN on METR-LA. The possible reason is that our model

TABLE 1. Details of two datasets.

Datasets	METR-LA	PEMS-BAY	External Factors	Los Angeles	Bay Area
Data type	Speed	Speed	Weather	sunny, rainy et al.	sunny, rainy et al.
Time interval	5 minutes	5 minutes	Holidays	rush hours, holidays	rush hours, holidays
Records	32,000+	52,000+	Wind speed (km/h)	[8.2, 80.50]	[8.1, 73.75]

TABLE 2. The prediction performance of different model on the METR-LA and PEMS-BAY dataset.

Datasets	Models	Horizon 3			Horizon 6			Horizon 12		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
METR-LA	HA	4.1623	7.8051	13.00%	4.1678	7.8072	13.00%	4.1689	7.8096	13.00%
	FC-LSTM	3.4421	6.3021	9.60%	3.7743	7.2312	10.90%	4.3753	8.6923	13.20%
	STGCN	2.8805	5.7398	7.62%	3.4743	7.2456	9.57%	4.5976	9.4077	12.70%
	AGCRN	2.8601	5.5501	7.55%	3.2513	6.5785	8.99%	3.6822	7.5645	10.46%
	DCRNN	2.7721	5.3864	7.31%	3.1578	6.4517	8.80%	3.6087	7.6066	10.50%
	ST-LGSL	2.6751	5.0859	6.86%	3.0537	6.1256	8.40%	3.5036	7.2195	10.10%
	Graph WaveNet	2.6927	5.1537	6.93%	3.0751	6.2289	8.37%	3.5364	7.3732	10.01%
	MTGNN	2.6943	5.1825	6.86%	3.0556	6.1776	8.19%	3.4921	7.2367	9.87%
	GMAN	2.7716	5.4834	7.25%	3.0776	6.3421	8.35%	3.4068	7.2201	9.72%
	SLCNN	2.5342	5.1865	6.73%	2.8807	6.1543	8.01%	3.3025	7.2033	9.72%
	GSCSDM(Ours)	**2.5323**	**5.0565**	**6.68%**	**2.9402**	**6.1277**	**7.98%**	**3.3413**	**7.2548**	**9.72%**
PEMS-BAY	HA	2.8811	5.5923	6.80%	2.8836	5.5939	6.80%	2.8842	5.5954	6.80%
	FC-LSTM	2.0525	4.1932	4.80%	2.2013	4.5504	5.20%	2.3732	4.6933	5.70%
	STGCN	1.3617	2.9621	2.92%	1.8123	4.2743	4.17%	2.4908	5.6932	5.79%
	AGCRN	1.3633	2.8821	2.93%	1.6921	3.8722	3.86%	1.9807	4.5933	4.63%
	DCRNN	1.3832	2.9533	2.90%	1.7436	3.9722	3.90%	2.0701	4.7412	4.90%
	ST-LGSL	1.3153	2.7638	2.39%	1.6833	3.8662	3.81%	1.8642	4.5043	4.54%
	Graph WaveNet	1.3184	2.7531	2.53%	1.6315	3.7121	3.67%	1.9511	4.5212	4.62%
	MTGNN	1.3209	2.7914	2.77%	1.6507	3.7412	3.69%	1.9465	4.4922	4.53%
	GMAN	1.3516	2.9122	2.87%	1.6515	3.8232	3.74%	1.9234	4.4945	4.52%
	SLCNN	1.4443	2.9564	3.03%	1.7232	3.8111	3.94%	2.0354	4.5321	4.84%
	GSCSDM(Ours)	**1.3125**	**2.7345**	**2.70%**	**1.6305**	**3.7033**	**3.65%**	**1.9046**	**4.4816**	**4.51%**

separately learns spatio-temporal features, while SLCNN can learn both simultaneously. In addition, Figure 3 shows the predicted results for the 15 minutes more detail.

Our proposed memory-augment mechanism can better detect conditions causality with time-delay with significantly improved prediction performance. From the predicted results, it can be seen that GSCSDM responds more quickly and accurately to dynamic changes in peak traffic.

E. ABLATION STUDY

In this section, we will demonstrate the effectiveness of the fusion by presenting three variant models. We conducted ablation experiments with these models on METR-LA.

- CCG w/o CCR: It removes the contextual causality representations from contextual causality graph.
- CCG w/o GM: It removes the gated linear unit from contextual causality graph.
- CCG w/ DCC: It replaces the element-wise product with concatenation in (4).
- PTR w/o MPM: It removes the memory pattern matching from spread time-delay.
- MGC w/o L^c: It removes the dynamic causality graph L^c.
- MGC w/o L^{geo}: It removes the geographical similarity graph L^{geo}.
- MGC w/o L^{dtw}: It removes the feature similarity graph L^{dtw}.

TABLE 3. Comparison with variants of GSCSDM on METR-LA.

Category	Models	Average		
		MAE	RMSE	MAPE
Multi-Graph	w/o L^c	3.1056	6.4432	8.33%
	w/o L^{geo}	3.0366	6.3843	8.19%
	w/o L^{dtw}	3.0466	6.4243	8.29%
Spread Time-delay	w/o **MPM**	3.1563	6.6122	8.67%
Contextual Conditions Graph	w/o **CCR**	3.2311	6.8455	8.85%
	w/o **GM**	3.0866	6.4335	8.22%
	w/ **DCC**	3.1168	6.5021	8.52%
	GSCSDM	**2.9833**	**6.2076**	**8.02%**

As shown in Table 3, it shows that each module contributes to the model. Blocking each module will reduce the performance. The performance of CCG w/o **CCR** is the lowest, which indicates that the causality of contextual conditions cannot be ignored. Dynamic causal graphs A^c are more important than the other two types of graphs in terms of their contribution. The introduction of causal graphs can greatly enhance performance as they capture implicit causality that static feature graphs cannot. In addition, geographical similarity graph and temporal similarity graph are also essen-

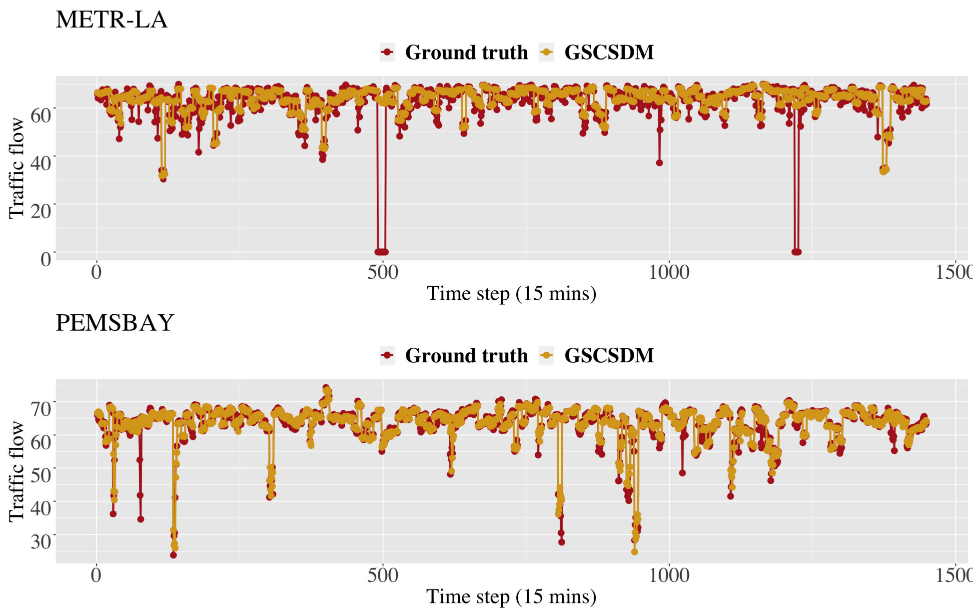


FIGURE 3. **Prediction result on two datasets.**

tial. Dynamic causal graphs can work together with static feature graphs to effectively model complex traffic networks. As shown in Figure 4, we represented the change process of the contribution coefficient ψ in the multi-graph convolution during the training period. The dynamic graph L^c has relatively large weights, whereas the two static graphs have relatively small weights. For the information-passing delay, L^c is crucial for modeling the similarity of traffic pattern into the dynamic graph. As the model is trained, it captures the dense local correlation and sparse long-range dependency, which aligns with the similarity of global traffic patterns. This illustrates that delivery time-delay and causality have a significant effect on graph structure. In addition, the model cannot ignore static graph structure and needs to extract potential spatial dependencies from multi-view graphs.

The performance of CCG w/ **DCC** is relatively low. This indicates that the integration methods are less effective in capturing the effect of contextual conditions compared to the causal method. The low performance of CCG w/o **DCC** also suggests that contextual conditions have a significant causal effect on predictive performance, and the model captures this potential relationship well. If these conditions are not taken into account, the model cannot effectively handle the disturbances caused by them. Removing MPM would significantly increase prediction loss. MPM effectively matches traffic patterns with data features to extract delay effects triggered by contextual conditions.

F. COMPUTATIONAL COMPLEXITY

We further evaluate the computational costs for DCRNN, STGCN, Graph WaveNet, and GMAN. All the experiments are conducted on the same GPU. Table 4 reports the average training speed for one epoch. STGCN is efficient with fully convolutional structures. DCRNN is highly time-consuming due to the recurrent structures for training with joint loss for multiple time steps. Graph WaveNet is low time-consuming due to the stacked dilated 1D convolution component. GMAN's training time is proportional to the number of multiple attention mechanisms. GSCSDM runs 3.1 times faster than Graph WaveNet but 2.5 times slower than GMAN in training. GSCSDM yields an approximate computational cost in comparison to DCRNN. The 3D convolution operation of SLCNN is time consuming. In addition, GSCSDM demonstrates the highest efficiency at the inference stage, except slightly weaker than STGCN. Note that GSCSDM can be scales to achieve higher predictive performance under contextual conditions without excessive computational complexity, whereas the other models ignore the effect of contextual conditions.

TABLE 4. **The computation cost on the PEMS-BAY dataset.**

	Computation Time	
Model	Training (s/epoch)	Inference (s)
DCRNN	118.85s	5.44s
GMAN	49.52s	57.15s
Graph WaveNet	393.24s	66.79s
STGCN	131.15s	5.64s
SLCNN	381.25s	21.53s
GSCSDM	124.94s	9.23s

G. OPTIMIZER

We tested the performance of four common optimizers (such as SGD, Adagrad, RMSProp, and Adam) during the training process. Here, we set the maximum number of epochs to 100, and when 35% epochs do not exceed the previous best performance, we will stop the training process early. SGD is hard to train and performs the validation loss

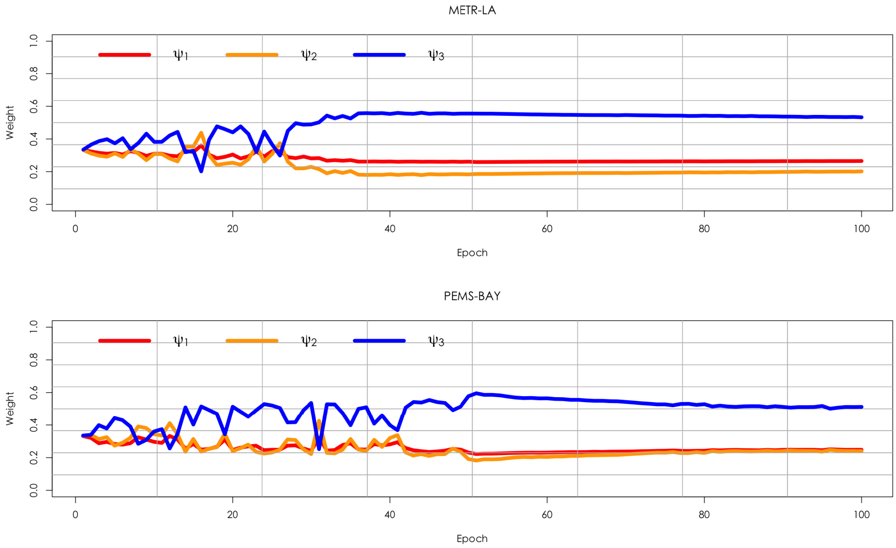
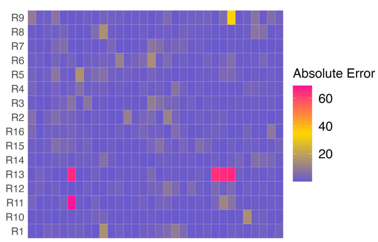


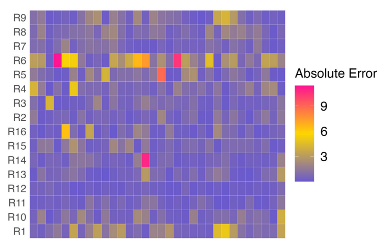
FIGURE 4. **The change process of the contribution coefficient ψ in the multi-graph convolution during the training period of two datasets.**



(a) 5 minutes

(b) 60 minutes

FIGURE 5. **The absolute error of GSCSDM on METR-LA.**



(a) 5 minutes

(b) 60 minutes

FIGURE 6. **The absolute error of GSCSDM on PEMS-BAY.**

20.6166 due to its weight update direction not being correct. In addition, SGD also fails to independently overcome the problem of local optimum, resulting in worst performance. The convergence performance of the other three optimizers is similar on two datasets respectively. Specifically, the overall validation loss based on Adam is 2.8225, Adagrad is 2.8838 and RMSprop is 2.8656 on METR-LA. Thus, the loss functions obtained based on Adam, Adagrad, and RMSprop have the optimal solution, while SGD carries the risk of local optimal solution. Furthermore, in the case of large datasets, the Adam optimizer has faster convergence speed and better performance.

H. VISUALIZATION

Figure 5 and 6 depict the absolute errors of the GSCSDM model on various prediction tasks for METR-LA and PEMS-BAY. The model exhibits favorable performance for short-term prediction, effectively capturing the temporal trend. However, due to the complexity and variability of real-world traffic conditions, its effectiveness decreases as the prediction range increases. Furthermore, we find that our model predict the diffusion of traffic patterns more accurately. Our model also performs feature extraction in both the temporal and spatial dimensions, and incorporates a multi-graph mechanism to respond more quickly to dynamic changes in traffic trends.

VI. CONCLUSION

We considered the "triggering effect" of contextual conditions on traffic flow as a challenge to describe the prediction problem. It shows that due to the integration approach, there are spurious graph structure in spatial correlation extraction, failing in coping with "triggering effect" caused by the contextual conditions. Thus, we proposed a method called GSCSDM that decomposes "triggering effect" into causality and time-delay. Among them, the contextual causality graph is a basis for GSCSDM to enhance important features caused by contextual conditions. It embeds the contextual influence-aware vectored representation into the causal weighted, which enables the model to capture the causality. Then, we propose a memory pattern matching, which captures the information-passing delaying based on pattern similarity. The experimental results demonstrated that our GSCSDM over various state-of-the-art prediction methods.

Discussion: In future works, there are some more interesting issues can be further discussed,

- The effectiveness of GSCSDM further demonstrates that it is not rational to learn spatial relationship directly ignoring contextual conditions, which are key factors for the graph structure evolving. In addition, the trend of road state evolving is always continuous and instantaneous under contextual conditions. Therefore, how to effectively model these two dynamics of the road simultaneously in order to extract the graph structure features requires further exploration.
- GSCSDM effectively captures the spread time-delay in the regional network by global attention mechanism. However, due to the over-smoothing phenomenon between multi-hop nodes, how to mitigate over-smoothing deserves further investigation in traffic network.

DATA AVAILABILITY

The METR-LA and PEMS-BAY data used to support the findings of this study have been deposited at https://github.com/liyaguang/DCRNN.

CONFLICTS OF INTEREST

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

CONFLICTS OF INTEREST

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

REFERENCES

[1] Y. Wang, D. Zhang, Y. Liu, B. Dai, and L. H. Lee, "Enhancing transportation systems via deep learning: A survey," *Transp. Res. C, Emerg. Technol.*, vol. 99, pp. 144–163, Feb. 2019.

[2] B. Pu, Y. Liu, N. Zhu, K. Li, and K. Li, "ED-ACNN: Novel attention convolutional neural network based on encoder–decoder framework for human traffic prediction," *Appl. Soft Comput.*, vol. 97, Dec. 2020, Art. no. 106688.

[3] X. Kong, J. Zhang, X. Wei, W. Xing, and W. Lu, "Adaptive spatial–temporal graph attention networks for traffic flow forecasting," *Appl. Intell.*, vol. 52, pp. 4300–4316, 2021.

[4] J. Wang, J. Jiang, W. Jiang, C. Li, and W. X. Zhao, "LibCity: An open library for traffic prediction," in *Proc. 29th Int. Conf. Adv. Geographic Inf. Syst.*, 2021, pp. 145–148.

[5] B. M. Williams, "Multivariate vehicular traffic flow prediction: Evaluation of ARIMAX modeling," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1776, no. 1, pp. 194–200, Jan. 2001. [Online]. Available: https://api.semanticscholar.org/CorpusID:110427797

[6] C.-H. Wu, J.-M. Ho, and D. T. Lee, "Travel-time prediction with support vector regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 276–281, Dec. 2004. [Online]. Available: https://api.semanticscholar.org/CorpusID:11214220

[7] S. V. Kumar and L. Vanajakshi, "Short-term traffic flow prediction using seasonal ARIMA model with limited input data," *Eur. Transp. Res. Rev.*, vol. 7, no. 3, pp. 21–29, Sep. 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:45118161

[8] S. V. Kumar, "Traffic flow prediction using Kalman filtering technique," *Proc. Eng.*, vol. 187, pp. 582–587, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:67689935

[9] G. Lin, A. Lin, and D. Gu, "Using support vector regression and K-nearest neighbors for short-term traffic flow prediction based on maximal information coefficient," *Inf. Sci.*, vol. 608, pp. 517–531, Aug. 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:250106701

[10] H. Xu and C. Jiang, "Deep belief network-based support vector regression method for traffic flow forecasting," *Neural Comput. Appl.*, vol. 32, no. 7, pp. 2027–2036, Apr. 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:198190863

[11] S. Sun, H. Wu, and L. Xiang, "City-wide traffic flow forecasting using a deep convolutional neural network," *Sensors*, vol. 20, no. 2, p. 421, Jan. 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:210332193

[12] V. Osipov, V. Nikiforov, N. Zhukova, and D. Miloserdov, "Urban traffic flows forecasting by recurrent neural networks with spiral structures of layers," *Neural Comput. Appl.*, vol. 32, no. 18, pp. 14885–14897, Sep. 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:212691976

[13] A. Belhadi, Y. Djenouri, D. Djenouri, and C.-W. Lin, "A recurrent neural network for urban long-term traffic flow forecasting," *Appl. Intell.*, vol. 50, pp. 3252–3265, May 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:218653682

[14] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," 2013, *arXiv:1312.6203*.

[15] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 1–9. [Online]. Available: https://api.semanticscholar.org/CorpusID:3016223

[16] T. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[17] F. Monti, M. M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 1–12. [Online]. Available: https://api.semanticscholar.org/CorpusID:18052422

[18] K. Lee and W. Rhee, "DDP-GCN: Multi-graph convolutional network for spatiotemporal traffic forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 134, Jan. 2022, Art. no. 103466. [Online]. Available: https://api.semanticscholar.org/CorpusID:211066627

[19] Y. Djenouri, A. Belhadi, G. Srivastava, and J. C.-W. Lin, "Hybrid graph convolution neural network and branch-and-bound optimization for traffic flow forecasting," *Future Gener. Comput. Syst.*, vol. 139, pp. 100–108, Feb. 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:252470978

[20] A. Khaled, A. M. T. Elsir, and Y. Shen, "TFGAN: Traffic forecasting using generative adversarial network with multi-graph convolutional network," *Knowledge-Based Syst.*, vol. 249, Aug. 2022, Art. no. 108990. [Online]. Available: https://api.semanticscholar.org/CorpusID:248735833

[21] Z. Shi, Y. Zhang, J. Wang, J. Qin, X. Liu, H. Yin, and H. Huang, "DAGCRN: Graph convolutional recurrent network for traffic forecasting with dynamic adjacency matrix," *Exp. Syst. Appl.*, vol. 227, Oct. 2023, Art. no. 120259. [Online]. Available: https://api.semanticscholar.org/CorpusID:258424399

[22] Y. Bao, J. Liu, Q. Shen, Y. Cao, W. Ding, and Q. Shi, "PKET-GCN: Prior knowledge enhanced time-varying graph convolution network for traffic flow prediction," *Inf. Sci.*, vol. 634, pp. 359–381, Jul. 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:256484167

[23] R. Liu and L. Chen, "Attention based spatial–temporal graph convolutional networks for short-term load forecasting," *J. Phys., Conf. Ser.*, vol. 2078, no. 1, Nov. 2021, Art. no. 012051.

[24] D. Xia, B. Shen, J. Geng, Y. Hu, Y. Li, and H. Li, "Attention-based spatial–temporal adaptive dual-graph convolutional network for traffic flow forecasting," *Neural Comput. Appl.*, vol. 35, pp. 17217–17231, Apr. 2023.

[25] A. Ali, Y. Zhu, and M. Zakarya, "Exploiting dynamic spatio-temporal correlations for citywide traffic flow prediction using attention based neural networks," *Inf. Sci.*, vol. 577, pp. 852–870, Oct. 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:238479286

[26] X. Zhang, C. Huang, Y. Xu, L. Xia, P. Dai, L. Bo, J. Zhang, and Y. Zheng, "Traffic flow forecasting with spatial–temporal graph diffusion network," *Proc. AAAI Conf. Artif. Intell.*, May 2021, vol. 35, no. 17, pp. 15008–15015. [Online]. Available: https://api.semanticscholar.org/CorpusID:235363413

[27] T. Zhang, W. Ding, T. Chen, Z. Wang, and J. Chen, "A graph convolutional method for traffic flow prediction in highway network," *Wireless Commun. Mobile Comput.*, vol. 2021, pp. 1–8, Jul. 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:235826186

[28] L. Zhao et al., "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:57189442

[29] J. Wang, W. Wang, W. Yu, X. Liu, K. Jia, X. Li, M. Zhong, Y. Sun, and Y. Xu, "STHGCN: A spatiotemporal prediction framework based on higher-order graph convolution networks," *Knowl.-Based Syst.*, vol. 258, Dec. 2022, Art. no. 109985. [Online]. Available: https://api.semanticscholar.org/CorpusID:252974737

[30] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2017, *arXiv:1707.01926*.

[31] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. NIPS*, 2017, pp. 1–11.

[32] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial–temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 1–11.

[33] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," 2020, *arXiv:2007.02842*.

[34] J. Ye, L. Sun, B. Du, Y. Fu, and H. Xiong, "Coupled layer-wise graph convolution for transportation demand prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 1–12.

[35] F. Li, J. Feng, H. Yan, G. Jin, F. Yang, F. Sun, D. Jin, and Y. Li, "Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution," *ACM Trans. Knowl. Discovery Data*, vol. 17, no. 1, pp. 1–21, Feb. 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:233476480

[36] J. Ye, S. Xue, and A. Jiang, "Attention-based spatio-temporal graph convolutional network considering external factors for multi-step traffic flow prediction," *Digit. Commun. Netw.*, vol. 8, no. 3, pp. 343–350, Jun. 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:244233777

[37] S. Zhu, R. Ding, M. Zhang, P. V. Hentenryck, and Y. Xie, "Spatio-temporal point processes with attention for traffic congestion event modeling," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7298–7309, Jul. 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:218674543

[38] A. Gupta, M. K. Maurya, N. Goyal, and V. K. Chaurasiya, "ISTGCN: Integrated spatio-temporal modeling for traffic prediction using traffic graph convolution network," *Appl. Intell.*, vol. 53, pp. 29153–29168, Jun. 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:264443800

[39] K. Zhang, F. He, Z. Zhang, X. Lin, and M. Li, "Graph attention temporal convolutional network for traffic speed forecasting on road networks," *Transportmetrica B, Transp. Dyn.*, vol. 9, no. 1, pp. 153–171, Jan. 2021.

[40] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 1–7.

[41] D. Chai, L. Wang, and Q. Yang, "Bike flow prediction with multi-graph convolutional networks," in *Proc. 26th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2018, pp. 1–12.

[42] S. Lan, Y. Ma, W. Huang, W. Wang, H. Yang, and P. Li, "DSTAGNN: Dynamic spatial–temporal aware graph neural network for traffic flow forecasting," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 11906–11917.

[43] H. Li, S. Zhang, X. Li, L. Su, H. Huang, D. Jin, L. Chen, J. Huang, and J. Yoo, "DetectorNet: Transformer-enhanced spatial temporal graph neural network for traffic prediction," in *Proc. 29th Int. Conf. Adv. Geograph. Inf. Syst.*, Nov. 2021, pp. 133–136. [Online]. Available: https://api.semanticscholar.org/CorpusID:240354523

[44] X. Wang, Y. Ma, Y. Wang, W. Jin, X. Wang, J. Tang, C. Jia, and J. Yu, "Traffic flow prediction via spatial temporal graph neural network," in *Proc. Web Conf.*, Apr. 2020, pp. 1082–1092. [Online]. Available: https://api.semanticscholar.org/CorpusID:215838304

[45] T. Wang, S. Ni, T. Qin, and D. Cao, "TransGAT: A dynamic graph attention residual networks for traffic flow forecasting," *Sustain. Comput., Informat. Syst.*, vol. 36, Dec. 2022, Art. no. 100779. [Online]. Available: https://api.semanticscholar.org/CorpusID:251336334

[46] W. Wei, B. H. Mao, S. K. Chen, S. Liu, and D. J. Xu, "Spatial propagating characteristic study of urban traffic congestion considering spatial nonstationarity," *Trans. Beijing. Inst. Technol.*, vol. 36, no. S2, pp. 103–106, Dec. 2016.

[47] J. Zakaria, A. Mueen, and E. Keogh, "Clustering time series using unsupervised-shapelets," in *Proc. IEEE 12th Int. Conf. Data Min.*, 2012, pp. 785–794. [Online]. Available: https://api.semanticscholar.org/CorpusID:3329986

[48] J. Jiang, C. Han, W. X. Zhao, and J. Wang, "PDFormer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2023, vol. 37, no. 4, pp. 4365–4373. [Online]. Available: https://api.semanticscholar.org/CorpusID:255999991

[49] J. Liu, Y. Kang, H. Li, H. Wang, and X. Yang, "STGHTN: Spatial–temporal gated hybrid transformer network for traffic flow forecasting," *Appl. Intell.*, vol. 53, pp. 12472–12488, May 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:252611155

[50] H. Park, J. Noh, and B. Ham, "Learning memory-guided normality for anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14360–14369.

[51] K. Bringmann, N. Fischer, I. van der Hoog, E. Kipouridis, T. Kociumaka, and E. Rotenberg, "Dynamic dynamic time warping," 2023, *arXiv:2310.18128*.

[52] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," 2014, *arXiv:1409.3215*.

[53] J. Tang, T. Qian, S. Liu, S. Du, J. Hu, and T. Li, "Spatio-temporal latent graph structure learning for traffic forecasting," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2022, pp. 1–8.

[54] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial–temporal graph modeling," 2019, *arXiv:1906.00121*.

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

[56] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," 2019, *arXiv:1911.08415*.

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

YIJUN XIONG received the B.S. degree from the Chengdu University of Information Technology, in 2007, and the M.S. degree in engineering from the Chengdu University of Technology, in 2010. He has been a Teacher with Chengdu University, since 2010. His research interests include deep learning and statistical analysis.

HUAJUN WANG received the B.S. degree from Beijing Normal University, in 1986, and the M.S. and Ph.D. degrees in engineering from the University of Electronic Science and Technology of China, in 1997 and 2000, respectively. He has been a Professor with the Chengdu University of Technology, since 2003. His research interest includes space physics.