Granger Causal Inference for Interpretable Traffic Prediction

Lei Zhang[1], Kaiqun Fu[2], Taoran Ji[1], and Chang-Tien Lu[1]

2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC) | 978-1-6654-6880-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/ITSC55140.2022.9922211

Abstract— **Modeling spatial dependency is crucial to solving traffic prediction tasks; thus, spatial-temporal graph-based models have been widely used in this area in recent years. Existing approaches either rely on a fixed pre-defined graph (e.g., a road network) or learn the correlations between locations. However, most methods suffer from spurious correlation and do not sufficiently consider the traffic's causal relationships. This study proposes a Spatiotemporal Causal Graph Inference (ST-CGI) framework for traffic prediction tasks that learn both the causal graph and autoregressive processes. We decouple the spatiotemporal traffic prediction process into two steps; the causal graph inference step and the autoregressive step, where the latter relies on the former. Optimizing the entire framework on the autoregressive task approximates the Granger causality test and thus enables excellent interpretability of the prediction. Extensive experimentation using two real-world datasets demonstrates the outstanding performance of the proposed models.**

I. INTRODUCTION

Highway traffic prediction has long been of interest to both industry and academia as a typical spatial-temporal data mining problem. Reliable, accurate, and consistent real-time traffic information is the key to success in developing and implementing an Intelligent Transportation System (ITS). Subsystems of an ITS, such as the Advanced Traveler Information System (ATIS) and the Advanced Traffic Management System (ATMS), rely heavily on high-quality real-time traffic data to provide road users up-to-date advisories and to implement traffic control schemes. In the past, the collection of real-time data was the foremost goal, but recently many agencies have begun to consider taking advantage of the vast archived datasets for "real-time forward looking analysis." With predictive data, proactive transportation management is a feasible option. For example, adaptive traffic signal control is more effective if it is based on predicted traffic volume.

A general guideline for most existing traffic prediction models can be summarized by Tobler's First Law (TFL), which says that "Everything is related to everything else, but things that are nearby are more related than distant things." TFL indicates the importance of bridging the spatial dependency and the importance of the statistical spatial correlations between different locations. Most existing traffic prediction methods follow TFL by representing the spatial dependency with a road network or the Euclidean distance. However, road networks and Euclidean distances are not accurate for many real-world circumstances. For example, two nearby traffic sensors may be positioned in two lanes of traffic moving in opposite directions on the highway, and thus have quite a very low spatial dependency. Some other new methods learn the correlations between locations with history data and achieve better results. However, these methods suffer from low interpretability. A moderate correlation between two locations could be the result of confounding variables or biased data. Inspired by the work in causal inference in recent years[1], [2], we propose solving the traffic prediction task by considering the Granger causal relationship among traffic sensors. Granger causality is a classical statistical concept of causality that is based on prediction [3]. By explicitly modeling the Granger causal relationship between the traffic sensors, the model becomes more stable and interpretable, which is highly desirable in academia and industry. Modeling the causal relationship in traffic prediction tasks is beyond the scope of TFL and has not been studied yet.

Despite its importance, however, learning causal relationship in traffic data involves significant technical challenges: 1) **Learning causal relationships on dynamic data.** The dynamics of traffic data can be considered as the result of the ground truth causal relationships. Tracing the causes of continuous dynamic data is difficult. 2) **The difficulty in learning causal relationships and the autoregressive model simultaneously.** Inferring interpretable causal relationships while trying to improve the accuracy of the autoregressive model is nontrivial. 3) **Model scalability on large spatiotemporal data**. The number of entity-to-entity relationships to infer is $O(n^2)$ of the number of locations.

We present a novel Graph Neural Network (GNN)-based Spatiotemporal Causal Graph Inference (ST-CGI) framework to address the above challenges. This framework infers the Granger causality graph for all locations while optimizing the autoregressive model, which depends on the causality graph. This study's significant contributions are as follows: 1) the design of a novel framework for both traffic prediction and traffic causal inference; 2) proposal for a more efficient approximation of Granger causality; 3) Extensive experiments for both performance and interpretability.

The rest of this paper is organized as follows: Section 2 reviews the background and related work. Section 3 introduces the preliminaries. Section 4 presents all the components of our ST-CGI framework. The experiments on real-world data are presented in Section 5, and the paper concludes with a summary of the research in Section 6.

*This work was not supported by any organization

[1]Lei Zhang, Taoran Ji, and Chang-Tien Lu are with the Department of Computer Science, Virginia Tech, 22043, Falls Church, USA zhanglei@vt.edu, jtr@vt.edu, clu@vt.edu

[2]Kaiqun Fu is with the Department of Electrical Engineering and Computer Science, South Dakota State University, Brookings, SD 57007, USA Kaiqun.Fu@sdstate.edu

II. Related Work

In this section, we provide a detailed review of the current state of research for traffic forecasting tasks.

A. Traffic Forecasting

Modeling spatial dependency is critical to the success of the traffic prediction task. In recent years, GNNs have been widely used to model spatial dependency. As GNNs require a pre-defined graph structure, researchers have proposed different ways to construct the graph in traffic prediction tasks. GEML defines the adjacent square areas in a grid and uses pre-weighted aggregators as the GNN kernel [4]. TGC-LSTM defines the traffic graph convolution based on the physical network topology (i.e., the road network) and applies K-order Graph Convolution Network (GCN)[5] in the model. Other studies have defined semantic similarities in a graph according to their respective data and application [6], [7], [4]. ST-MGCN utilized all three of the graphs mentioned above in its model and defined the GNN kernel as the aggregation of all graphs [6]. DCRNN defines the graph by thresholding the Euclidean distance and applies diffusion convolution as the GNN kernel, while STGCN defines the graph in a similar way and applies ChebyNet on it. In the last two years, researchers have noticed the importance of the graph structure in a model and have proposed calculating the attention scores as the correlation strength among different geo-locations. ASTGCN calculates the graph with trainable matrices and input data from two nodes [8]. AGCRN [9], GMAN [10], and Graph WaveNet [11] all calculate correlations with trainable node embeddings. SLCNN also uses trainable matrices but splits the graphs into four categories: dynamic global graph, dynamic local graph, static global graph, and static local graph [12].

B. Causal Inference

Traditional constraint-based causal inference methods construct DAGs using conditional independence tests and Markov equivalence classes. Some examples include the kernel method [13] and the PC algorithm [14]. These methods involve a multiple testing problem where the tests are usually conducted independently. The testing results may conflict, and they require sophisticated post-processing. Recently, Zhu et al. proposed an End-to-End Reinforcement Learning model to infer the graph-structured causal relationship, but the source data they used were static [15]. For dynamic data (e.g., time series data), Granger causality based methods also involve multiple tests and are not scalable on large datasets. Moreover, traditional Granger causality-based methods (e.g., VAR or spectral based) are inaccurate when the sparsity of the ground truth causal graph is unknown [16]. Some researchers have tried to extend Granger causality by replacing the VAR with recurrent neural networks [17], but the scalability issues are still unsolved.

III. Problem Formulation

In this section, we introduce the basics of causal inference on time series data. Then we formulate the traffic prediction and causal inference task in an optimization problem in a single framework.

A. Preliminaries

We target on the spatial temporal traffic forecasting problem by following the same formulation as previous work [11], [12], [18]. Consider multitudinous traffic series that contains N correlated univariate time series represented as $X = \{X_{0,-}, X_{1,-}, ..., X_{t,-}, ...\}$, where $X_{t,-} = \{X_{t,1}, X_{t,2}, ..., X_{t,N}\}^T \in \mathbb{R}^{N\times 1}$ is the traffic data of N traffic sensors at time step t, our target is to predict the future values of the traffic time series based on the observed historical values. We formulate the problem as finding a function g to forecast the future step traffic flow based on the past S steps historical data:

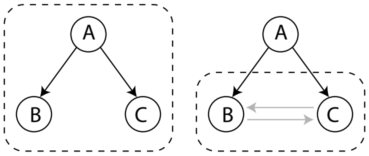
$$X_{t+1,-} = g(X_{[t-S:t],-}) \tag{1}$$

Definition 1: Granger Causality: A time series $X_{t,i}$ is Granger causal of another time series $X_{t,j}$ if including the history of $X_{-,i}$ improves prediction of $X_{-,j}$ over knowing of the history of $X_{-,j}$ alone. Specifically, this is quantified by comparing the prediction error variances of the one-step linear predictor, $\widehat{X}_{t,j}$, under two different models, the **restricted model** and the **unrestricted model**. The unrestricted model $g_u(X_{[t-S:t],-})_j = \widehat{X}_{t+1,j}$ uses the full histories of all the time series for prediction. The restricted model $g_r(X_{[t-S:t],-i})_j = \widehat{X}_{t+1,j}$ omits the putatively causal time series from the set of predictive time series. $X_{-,i}$ Granger causes $X_{-,j}$ if

$$\begin{aligned} &var(|X_{t+1,j} - g_u(X_{[t-S:t],-})_j| \mid X_{[t-S:t],-}) > \\ &var(|X_{t+1,j} - g_r(X_{[t-S:t],-i})_j| \mid X_{[t-S:t],-i}) \end{aligned} \tag{2}$$

here $var(|X_{t+1,j} - g_u(X_{[t-S:t],-i})_j| \mid X_{[t-S:t],-i}$ denotes the prediction performance of model g_u given histories data on all the nodes except for node i, i.e., $X_{[t-S:t],-i}$.

Granger formulated a statistical definition of causality based on the premise that (i) a cause occurs before its effect, and (ii) knowledge of a cause improves prediction of its effect. However, a determination of Granger causality does not guarantee true causality. The Granger causality tests fulfill only the Humean definition of causality, which identifies cause and effect relations as those having constant conjunctions. If a common third process drives both X and Y with different lags, one may still fail to reject the alternative hypothesis of Granger causality. Figure (1) shows two cases of causal inference. In Case 1, node A is the confounder of both nodes B and node C. If all three nodes are all included in the data, even though node B and node C are correlated, Granger causality can identify that node B and C do not have a causal relationship. However, in Case 2, when the confounder is not included in the observational data, Granger causality cannot discover the hidden confounding node, A. As a result, Granger causality may infer an inaccurate causal relationship between nodes B and C. Each of the components will be introduced in the following sections.



(a) Case 1 (b) Case 2

Fig. 1. Granger causality can identify confounders (Case 1) but cannot identify hidden confounders (Case 2).

While Granger causality is a potential method for causal discovery, it depends highly on the selection of a multivariate autoregressive function. In traditional Granger causality methods, the multivariate autoregressive function is typically a linear model that cannot capture nonlinear interactions between entities. Furthermore, Granger causality requires running the autoregressive function multiple times to determine the marginal predictability when including and excluding different variables, which is complicated and time-consuming. To resolve all of these issues, we propose to build an end-to-end neural network model.

The Granger causality inference problem can be formulated as the following optimization problem:

$$\begin{aligned} G_C &= \arg\min_{G}\Big(\sum_{t=S}^{\mathscr{T}} |X_{t+1} - g_{W^*}(G, X_{[t-S:t]})| + \lambda\mathscr{R}(G)\Big) \\ \textit{where } W^*(G) &= \arg\min_{W} \sum_{t=S}^{\mathscr{T}} |X_{t+1} - g_W(G, X_{[t-S:t]})| \end{aligned} \tag{3}$$

where g is the neural network model parameterized with W. $\mathscr{R}$ is the regularization term on the causal graph, which enforces fewer edges in the graph. A concrete solution for g is introduced in the following sections.

The autoregressive function g takes the time series data and causal graph as input. When g only considers one-hop neighbors in G, the conditional log-likelihood given the embedding of nodes for predicting the graph signal $LL(X_{-,j}|e_{i,j})$ is equal to $LL(X_{-,-}|e_{i,j})$ because edge $e_{i,j}$ can only affect the prediction for node j. As a result, optimizing (3) guarantees that each edge $e_{i,j}$ in G indicates the Granger causality from node i to node j.

There are two terms in the outer optimization. The first term optimizes for the best autoregressive function G, which depends on the causal graph G. The second term $\lambda\mathscr{R}(G)$ penalizes the number of edges in the graph. As a result, if the edge does not contribute to the first term's prediction, the edge will be removed in the outer optimization problem. According to the analysis above, the problem in 3 is a bi-level optimization problem.

B. ST-CGI Framework

The bi-level optimization problem in equation (3) is expensive to solve. For every causal relationship $i \rightarrow j, i \neq j$, we need to build a restricted model and an unrestricted model. Therefore, the computational complexity is at least $O(n^2)$. The causal graph we are trying to learn is a relatively stable relationship $\left[X_{(t-S):t}\right] \xrightarrow{G} X_{t+1}$. According to Norbert Wiener, whose work Granger cauaslity work built upon, "if a time series X causes a time series Y, then past values of X should contain information that help predict Y above and beyond the information contained in past values of Y alone". That inspired us to decouple the bi-level optimization problem as a multi-goal optimization problem that can be solved iteratively. The causal graph could be approximated with the relationship $\left[X_{(t-S):t}\right] \xrightarrow{G'} X_t$. Instead of optimizing on the causal graph directly, we propose to optimize the weight of a GNN that generates the causal graph. This approximation can be represented as a single-layer optimization problem in the following equation (4):

$$\begin{aligned} &\arg\min_{\Theta, W}\Big(\sum_{t=S}^{\mathscr{T}} |X_{t+1} - g(W, G^*, X_{[t-S:t]})| + \lambda\mathscr{R}(G_t^*)\Big) \\ &G_t^* = CGI(\Theta, X_{[t-S:t]}) \end{aligned} \tag{4}$$

where CGI is the causal graph estimator parameterized with Θ, and $\mathscr{R}$ is the L_1 regularization of the causal graph.

We propose to solve the causal inference and the autoregressive tasks simultaneously in a single end-to-end framework. Both the causal graph and the prediction for the next timestamp are inferred from the input data. The prediction for the next timestamp is dependent on the inferred causal graph. According to this requirement, we propose a framework, shown in Figure (2), with three modules trained jointly, namely a time series encoder, e, a causal graph estimator, and a causal graph-based autoregressive module.

The time series encoder e learns the high-dimensional representations of the multivariate input time series data. As shown in equation (5), the input data from P nodes $X \in \mathbb{R}^{P\times D\times S}$ are transformed into an embedding representation $h \in \mathbb{R}^{P\times R}$, where D is the input dimension and R is the hidden dimension.

$$h_{t-S:t} = e(X_{t-S:t}) \tag{5}$$

As shown in equation (6), the causal graph estimator f learns the affinity matrix of the causal graph $A \in \mathbb{R}^{P\times P}$ from the embedding.

$$A_{t-S,t} = f(h_{t-S:t}) \tag{6}$$

Given the causal graph, the autoregressive model g tries to approximate the underlying linear/nonlinear mapping from historical traffic to future traffic.

$$\widehat{X}_{t+1} = g(A_{t-S,t}, X_{t-S:t}) \tag{7}$$

The model's overall formulation can be written as equation (8).

$$\begin{aligned} \widehat{X}_{t+1} &= \mathscr{F}_\theta(X_{t-S:t}) \\ &= g(f(e(X_{t-S:t})), X_{t-S:t}) \end{aligned} \tag{8}$$

Given a long period of historical time series data and the problem formulation, we optimize the function in equation (9)

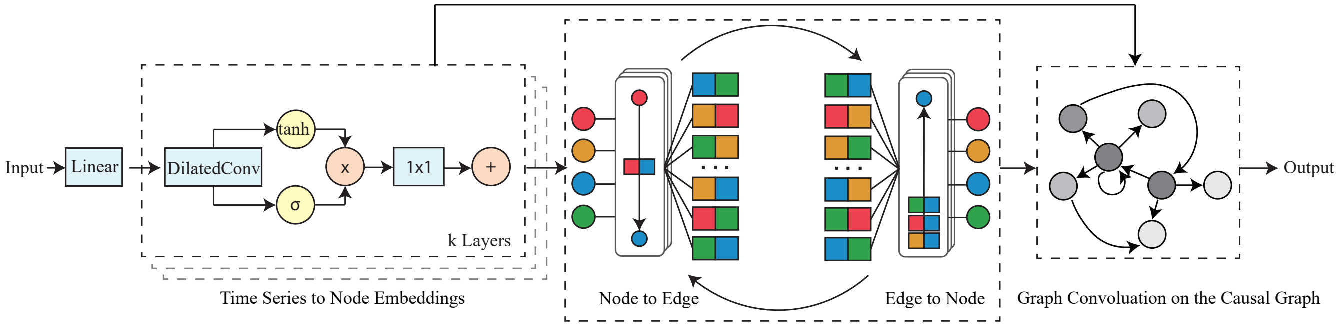


Fig. 2. ST-CGI framework: The left module is the TCN-based time series encoder. The middle module is the deep graph neural network-based causal graph estimator. The right module is the single layer GCN for traffic forecasting. The traffic forecasting is based on both the traffic data embedding from the left module and the inferred causal graph from the middle module.

$$
\begin{aligned}
&f,g,e = \arg\min_{f,g,e} \mathscr{L}(f,g,e,X_{train}) \\
&\text{where } \mathscr{L}(f,g,e,X_{train}) = \\
&\frac{1}{S\mathscr{T}} \sum_{s=1}^{S} \sum_{t=S}^{\mathscr{T}} (X_{t+1,s} - g(f(e(X_{t-S:t})),X_{t-S:t}))
\end{aligned} \tag{9}
$$

IV. Methodology

In this section, we demonstrate the detailed implementations of the ST-CGI framework's three components mentioned in Section III-B.

A. Causal Convolution for Time Series Encoding

The left module in Figure 2 demonstrates the structure of the time series encoder. Given raw multivariate time series data with lag S, the first step is to encode the time series into unified embeddings. The embedding should be a condensed vector that contains information about the temporal trends, temporal dependency, and temporal delays of the time series. The embedding must retain as much information as possible about the causal relationships between nodes.

Instead of using RNN-based units (e.g., LSTM, GRU), we opt to use dilated causal convolution network as our temporal convolution network (TCN) for encoding the time series data[19]. Dilated causal convolution networks allow an exponentially larger receptive field when the layer depth increases.

As shown in equation 10, gated dilated causal convolution [20] is utilized on every layer of the module. Because the model can become very deep when the input time series data is long, skip connections [21] are also used for each layer of convolutions. This way, the residual functions can be more easily learned during the optimization.

$$
h = tanh(W_{f,k} * x) \odot \sigma(W_{g,k} * x) \tag{10}
$$

where W is the learnable parameter, k is the layer index, f and g denote filter and gate, $*$ is the convolution operator, $\odot$ denotes an element-wise multiplication operator, and σ is a sigmoid function.

B. Causal Graph Estimator

After the time series on each location/node has been encoded as embeddings, we assume the embeddings contain essential information about the dynamics of the time series. The next step is to infer the causal relationship between the embeddings. As the causal relationship lies in a discrete space, GNN models are suitable for inferring the causal graph. GNNs have been shown to capture the relational connectivity between nodes [22].

It is natural to assume that the causal graph is related to a transportation road network in the spatial environment. For example, locations in closer proximity to one another have a higher chance of impacting one another. Different from the NRI model [22], we applied GNN models on the existing network structure rather than the complete graph. There are several benefits of starting from an existing graph. First, the computational costs can be greatly reduced. Second, the GNN model converges faster. By stacking k GNN layers, the GNN model's receptive field is k-hops over neighbors in the existing graph, which is sufficient for inferring most of the causal relationships. In other words, the formula in (6) can be changed to $A_{t-S,t} = f(A', h_{t-S:t})$ where A' is the affinity matrix of the existing graph and f is a GNN model. Given the traffic flow embeddings h from the time series encoder in Eq. (10), the GNN model computes the following node to edge ($v \rightarrow e$) and edge to node ($e \rightarrow v$) message passing operations:

$$
\begin{aligned}
&input : h_v^{(0)} = h \\
&v \rightarrow e : h_e^{(m)'} = \sigma(W_e^{(m)} \begin{bmatrix} h_v^{(m)} R \\ h_v^{(m)} S. \\ h_e^{(m-1)} \end{bmatrix} + b_e^{(m)}) \\
&e \rightarrow v : h_v^{(m)'} = \sigma(W_v^{(m)} \begin{bmatrix} h_e^{(m)} R^T \\ h_v^{(m-1)} \end{bmatrix} + b_v^{(m)}) \\
&output : A = f_{out}(h_e^{(n)})
\end{aligned} \tag{11}
$$

where $h_v^{(m)}$ is the m-th layer node embedding, $h_e^{(m)}$ is the m-th layer edge embedding, f_{out} is the read-out function that transforms the last layer of the edge embeddings into the affinity matrix of the causal graph. R and S are the receiver and sender matrices of edges (detailed in the Appendix)

The middle module in Figure 2 demonstrates the structure of the causal graph estimator. The node to edge message passing and the edge to node message passing can be repeated multiple times to increase the receptive field of the graph neural network.

C. Causal Graph Based Autoregressive Model

After the causal graph is inferred, we apply only one layer of GCN on the temporal embeddings of the input. The reason for this is that we focus on only inferring direct Granger causal relations rather than indirect causal relations. Using a single-layer GCN in the autoregressive stage enforces the causal graph estimator to learn a Granger causal graph. Given the traffic embeddings from the time series encoder and the affinity matrix inferred from the causal graph estimator, the final forecasting step

$$X_{t+1} = \sigma(AhW) \tag{12}$$

where h is the embedding representation of nodes learned from Eq. (5), and W is the parameter to optimize in this step.

D. Relationship with Other Models

Our proposed method is totally different from those who only used pre-defined graph structure (e.g., road networks, distance-based graphs, similarity graphs) for modeling the spatial dependency. Such methods include DCRNN[18], GEML [4], TGC-LSTM [5], ST-MGCN[6], STG2Seq[7]. The existing or pre-defined graph structures are helpful for learning the spatial dependency. However, it is inevitable that such graphs are noisy or miss certain information for different applications.

The most similar methods to our work are those that try to learn/optimize a graph structure for modeling the spatial dependency. Such methods include ASTGCN [8], AGCRN [9], GMAN [10], Graph WaveNet [11], and SLCNN [12]. Compared with these studies, our work has several unique features. Firstly, our framework is based on the Granger causality theory, which formulates the causality relationship, while the others depend on correlation relationships that are difficult to interpret. Secondly, the causal relationship we learn in the ST-CGI framework is directed, while most of the other methods regard the spatial dependency from a to b and from b to a are the same. Thirdly, ST-CGI decouples the spatial dependency modeling and the traffic data autoregressive task, which makes the model easy to interpret.

V. Experiment

In this section, we introduce the experiment settings, performance comparison, and interpretability illustration.

A. Datasets

We verify ST-CGI on two public traffic network datasets, METR-LA and PEMS-BAY. METR-LA contains four months of statistics on traffic speed from 207 sensors on Los Angeles County's highways. PEMS-BAY contains six months of traffic speed information from 325 sensors in the Bay area. We adopt the same data pre-processing procedures as in [11]. The raw time series data are aggregated into five-minute windows. Z-score normalization is applied to the aggregated data. The datasets are split in chronological order, with 70% used for training, 10% for validation, and 20% for testing.

B. Baseline Methods

SLCNN is compared with a traditional method, Auto-Regressive Integrated Moving Average (ARIMA), and the state-of-the-art methods including DCRNN[18], STGCN[23], SLCNN[12], and Graph Wavenet (GWN)[11].

C. Experiment Settings

For the time series encoder, similar to [11], we use sequence of dilation factors 1, 2, 1, 2, 1, 2, 1, 2. Dropout with p=0.3 is applied to the outputs of the time series encoder. For the causal graph estimator, We apply a sequence of $v \to e, e \to v, v \to e, e \to v, v \to e$ message passing GNN operations. The read-out function f_{out} in equation (11) is set to be two dense layers. The initial graph is the same distance-based graph used in [18], [11]. We train our model using Adam optimization algorithm with an initial learning rate of 0.001.

D. Performance Comparison

Following the same settings used by DCRNN, GWN, and SLCNN, the forecasting tasks are set in three levels, including 15 minutes, 30 minutes, and 1 hour ahead of forecasting. Table I and Table II show the comparison of different approaches for short term (15 minutes) traffic forecasting on both datasets. Table III and Table IV show the comparison of different approaches for long term (30/60 minutes) traffic forecasting on both datasets. All of the methods are evaluated based on three commonly used metrics in traffic forecasting tasks, including Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). We have the following observations.

(1) For short term (15 minutes) traffic prediction, our proposed ST-CGI method achieves overall the best performance among all the methods.

(2) For long term (30/60 minutes) traffic prediction, GWN and SLCNN have very competitive performance. On METR-LA data, ST-CGI performs better than GWN but fails to beat SLCNN. On PEMS-BAY data, ST-CGI performs better than SLCNN but fails to beat GWN. The performance difference between ST-CGI and the best performed models is minor.

It is noteworthy that ARIMA is the only interpretable method among the baseline methods, as ARIMA can be considered as a linear method. All of the rest baseline methods lack a theoretical foundation for interpretation. Our

TABLE I
PERFORMANCE COMPARISON FOR 15 MINUTES TRAFFIC PREDICTION ON THE METR-LA DATASET

Model	METR-LA (15 min)		
	MAE	RMSE	MAPE
ARIMA	3.99	8.21	9.6%
DCRNN	2.77	5.38	7.3%
STGCN	2.87	5.54	7.4%
SLCNN	**2.53**	5.18	6.7%
GWN	2.69	5.15	6.9%
ST-CGI	2.60	**5.14**	**6.7%**

TABLE II
PERFORMANCE COMPARISON FOR 15 MINUTES TRAFFIC PREDICTION ON THE PEMS-BAY DATASET

Model	PeMS-BAY (15 min)		
	MAE	RMSE	MAPE
ARIMA	1.62	3.30	3.5%
DCRNN	1.38	2.95	2.9%
STGCN	1.46	3.01	2.9%
SLCNN	1.44	2.90	3.0%
GWN	1.30	2.74	2.7%
SG-CGI	**1.29**	**2.69**	**2.7%**

proposed ST-CGI method not only outperforms the only interpretable method but also performs better or on par with the non-interpretable deep neural network methods.

E. Interpretability Investigation

The theoretical foundations in causal inference enable ST-CGI great power in interpretability. After the model is trained, the first two modules in the framework can estimate the causal graph of given data. Causal graphs are directed graphs where an edge from node i to node j indicates the Granger causal relationship from i to j. In the traffic forecasting task, an edge with a high weight from traffic sensor i to traffic sensor j means that the traffic flow at location j is heavily affected by the traffic flow at location i.

TABLE III
PERFORMANCE COMPARISON FOR 30/60 MINUTES TRAFFIC PREDICTION ON THE METR-LA DATASET

Model	METR-LA (30/60 min)		
	MAE	RMSE	MAPE
ARIMA	5.15/6.90	10.45/13.23	12.7%/17.4%
DCRNN	3.15/3.60	6.45/7.60	8.8%/10.5%
STGCN	3.48/4.45	6.84/8.41	9.4%/11.8%
SLCNN	**2.88/3.30**	**6.15/7.20**	**8.0%/9.7%**
GWN	3.07/3.53	6.22/7.37	8.4%/10.0%
ST-CGI	3.01/3.44	6.19/7.28	8.3%/9.8%

TABLE IV
PERFORMANCE COMPARISON FOR 30/60 MINUTES TRAFFIC PREDICTION ON THE PEMS-BAY DATASET

Model	PeMS-BAY (30/60 min)		
	MAE	RMSE	MAPE
ARIMA	2.33/3.38	4.76/6.50	5.4%/8.3%
DCRNN	1.74/2.07	3.97/4.74	3.9%/4.9%
STGCN	2.00/2.67	4.31/5.73	4.1%/5.4%
SLCNN	1.72/2.03	3.81/4.53	3.9%/4.8%
GWN	**1.63/1.95**	**3.70/4.52**	**3.7%/4.6%**
SG-CGI	1.68/2.01	3.77/4.65	3.8%/4.8%

Figure 3 shows a causal graph inferred by ST-CGI on the METR-LA dataset. Despite that the directed edges are difficult to spot due to a large number of nodes, we can still get some information from the Figure. The first observation is that there are some "hubs" with large numbers of outgoing edges. This means that the traffic flow in this area is mainly influenced by a small number of locations. To better demonstrate whether the hubs make sense or not, we picked two nodes with denser outgoing edges in the graph and showed them in Figure 4 in the Appendix. The node 202 (coordinate: 34.14604, -118.2243) in Figure 4 (a) is the freeway exit/entrance of Glendale Freeway according to Google Map. The node 177 (coordinate: 34.11966, -118.23143) in Figure 4 is a very crowded freeway interchange. We also checked some less important hubs with less outgoing edges, and they turn out to be places like Home Depot and residence communities.

One important observation is that the Granger causal relationship between two adjacent locations is usually not strong, which appears counterintuitive. According to the inferred causal graphs, a distant hub may have a greater impact on traffic flow in one location than the closest traffic sensor on the same road. It makes sense because the nature of Granger causality is to eliminate the indirect causes which have no incremental benefit on the predictions. In a real-world scenario, the traffic flow 15 minutes later near one traffic sensor should come from locations that are further than its closest traffic sensors.

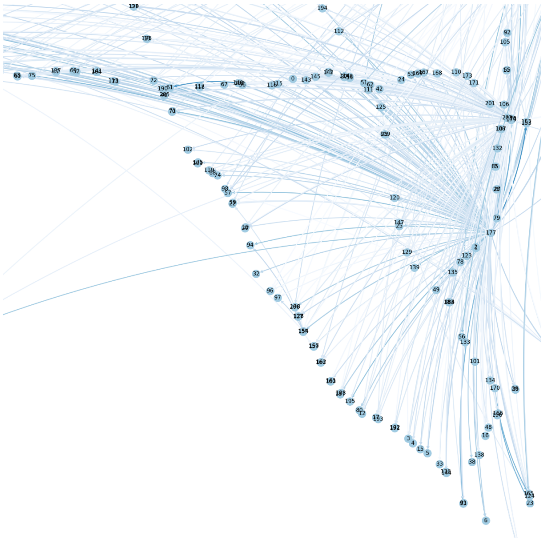


Fig. 3. Visualization of an inferred causal graph on the METR-LA dataset.

VI. CONCLUSION

In this study, we propose a novel interpretable model (ST-CGI) for traffic prediction tasks based on Granger causality.

Specially, we formulate the traffic prediction task a bi-level optimization task with a causal inference module and an autoregressive module, where the latter module depends on the former module. The model does not only make predictions of the traffic flow but also provides evidence of the predictions which are based on. Experiments on two real-world datasets show that ST-CGI achieves state-of-the-art results and has advantages in making shorter-term predictions. The interpretability and visualization function of ST-CGI is highly desired by domain experts and decision-makers.

Appendix

Matrices in the Causal Graph Estimator

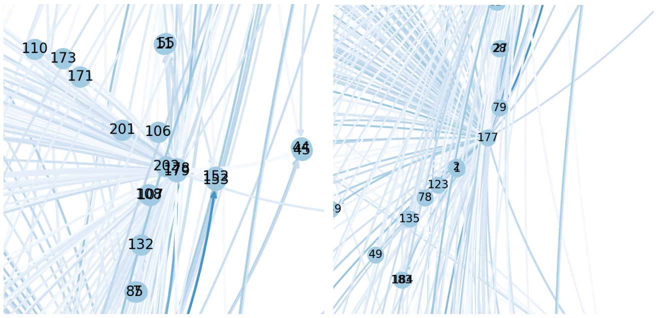
A is the adjacency matrix of the road network. We adopt the same implementation of A in DCRNN[18] and Graph Wavenet (GWN) [11] which is based on pre-calculated road network distances between sensors.

R is the receiver-edge matrix. It can be calculated with numpy-like pseudocode as $R = onehot(where(A)[0])$.

S is the sender-edge matrix. It can be calculated with numpy-like pseudocode as $S = onehot(where(A)[1])$.

Case Study

The example in the case study is illustrated in Fig. 4.



(a) Node 202 (ID: 7175920)   (b) Node 177 (ID: 767554)

Fig. 4. Example of “hubs” in the causal graph

References

[1] Y. Bengio, T. Deleu, N. Rahaman, N. R. Ke, S. Lachapelle, O. Bilaniuk, A. Goyal, and C. Pal, “A meta-transfer objective for learning to disentangle causal mechanisms,” in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=ryxWIgBFPS

[2] K. Kuang, P. Cui, B. Li, M. Jiang, S. Yang, and F. Wang, “Treatment effect estimation with data-driven variable decomposition,” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[3] C. Granger, “Testing for causality: a personal viewpoint,” in *Essays in econometrics: collected papers of Clive WJ Granger*, 2001, pp. 48–70.

[4] Y. Wang, H. Yin, H. Chen, T. Wo, J. Xu, and K. Zheng, “Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modeling,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1227–1235.

[5] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, “Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting,” *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[6] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu, “Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3656–3663.

[7] L. Bai, L. Yao, S. S. Kanhere, X. Wang, and Q. Z. Sheng, “Stg2seq: Spatial-temporal graph to sequence model for multi-step passenger demand forecasting,” in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, 7 2019, pp. 1981–1987. [Online]. Available: https://doi.org/10.24963/ijcai.2019/274

[8] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, “Attention based spatial-temporal graph convolutional networks for traffic flow forecasting,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 922–929.

[9] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, “Adaptive graph convolutional recurrent network for traffic forecasting,” in *34th Conference on Neural Information Processing Systems*, 2020.

[10] C. Zheng, X. Fan, C. Wang, and J. Qi, “Gman: A graph multi-attention network for traffic prediction,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 1234–1241.

[11] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, “Graph wavenet for deep spatial-temporal graph modeling,” in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, S. Kraus, Ed. United States of America: Association for the Advancement of Artificial Intelligence (AAAI), 2019, pp. 1907–1913, international Joint Conference on Artificial Intelligence 2019, IJCAI-19 ; Conference date: 10-08-2019 Through 16-08-2019. [Online]. Available: https://ijcai19.org/

[12] Q. Zhang, J. Chang, G. Meng, S. Xiang, and C. Pan, “Spatio-temporal graph structure learning for traffic forecasting,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 1177–1185.

[13] X. Sun, D. Janzing, B. Schölkopf, and K. Fukumizu, “A kernel-based causal learning algorithm,” in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 855–862.

[14] P. Spirtes, C. N. Glymour, R. Scheines, and D. Heckerman, *Causation, prediction, and search*. MIT press, 2000.

[15] S. Zhu, I. Ng, and Z. Chen, “Causal discovery with reinforcement learning,” in *International Conference on Learning Representations*, 2019.

[16] B. Lusch, P. D. Maia, and J. N. Kutz, “Inferring connectivity in networked dynamical systems: Challenges using granger causality,” *Physical Review E*, vol. 94, no. 3, p. 032220, 2016.

[17] H. Huang, C. Xu, and S. Yoo, “Bi-directional causal graph learning through weight-sharing and low-rank neural network,” in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 319–328.

[18] Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion convolutional recurrent neural network: Data-driven traffic forecasting,” in *International Conference on Learning Representations*, 2018.

[19] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” *CoRR*, vol. abs/1609.03499, 2016. [Online]. Available: http://arxiv.org/abs/1609.03499

[20] W. Hua, Y. Zhou, C. M. De Sa, Z. Zhang, and G. E. Suh, “Channel gating neural networks,” in *Advances in Neural Information Processing Systems*, 2019, pp. 1886–1896.

[21] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[22] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel, “Neural relational inference for interacting systems,” *arXiv preprint arXiv:1802.04687*, 2018.

[23] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting,” in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3634–3640.