# GT-CausIn: a novel causal-based insight for traffic prediction

**Anonymous authors**
Paper under double-blind review

## Abstract

Traffic forecasting is an important issue of spatiotemporal series prediction. Among different methods, graph neural networks have achieved so far the most promising results, learning relations between graph nodes then becomes a crucial task. However, improvement space is very limited when these relations are learned in a node-to-node manner. The challenge stems from (1) obscure temporal dependencies between different stations, (2) difficulties in defining variables beyond the node level, and (3) no ready-made method to validate the learned relations. To confront these challenges, we define legitimate traffic variables to discover the causal structure of the traffic network. The causal relation is carefully checked with statistic tools and case analysis. We then present a novel model named *Graph Spatial-Temporal Network Based on Causal Insight* (GT-CausIn), where graph diffusion layers and temporal convolutional network (TCN) layers are integrated with causal knowledge to capture dependencies in spatiotemporal space. Experiments are carried out on two real-world traffic datasets: PEMS-BAY and METR-LA, which show that GT-CausIn significantly outperforms the state-of-the-art models.

## 1 Introduction

As an important aspect of smart cities (Nagy & Simon, 2018), traffic forecasting is gaining more and more popularity in this data era. Traffic forecasting aims at predicting future traffic volumes with historical records. Different models have been proposed and evolved. Knowledge-driven models focus on defining functional characteristics of traffic elements and simulating various traffic service demands (Cascetta, 2013), where analysis is generally theory-based and requires constrained mathematical assumptions. Data-driven models contain three major categories: time-series analysis methods (e.g., ARIMA (Hamed et al., 1995), SARIMA (Williams & Hoel, 2003)), traditional machine learning methods (e.g., SVM (Smola & Schölkopf, 2004), SVR (Evgeniou et al., 2000)) and deep learning based methods (e.g., CNN (Ma et al., 2017; Guo et al., 2019)).

Deep learning methods outperform the others by a large margin and have become the mainstream solution in the industry. Guo et al. (2019) partition a city into a grid-based map and use 3D convolution to capture the spatiotemporal correlation of traffic data. Geng et al. (2019) build road graphs from multiple views based on pixel images and design a Contextual Gated Recurrent Neural Network (CGRNN) for information integration. Li et al. (2018) model traffic flow as a diffusion process for directed graph, encoder-decoder structure and scheduled sampling are further proposed to reduce error accumulation and improve model performance. Zheng et al. (2020) skip convolution layers and benefit from multi-head attention layers to embed spatial and temporal information.

The traffic road network provides a natural premium connection map for traffic flow analysis, and we argue that its inherent regime is not limited to node-to-node impacts. For example, the inflow of a highway service station is equal to its outflow within a certain period of time. Therefore, it is interesting to analyze the relations between different traffic variables. In recent years, causal discovery (Pearl, 2009) opens up the door to the analysis of causal relations behind statistical correlations and is widely explored in the computational biology domain (Pranay & Nagaraj, 2021; Cundy et al., 2021). However, to the best of our knowledge, few works consider causal observation for spatiotemporal forecasting.

In this work, we focus on improving traffic speed prediction with insight revealed by a causal discovery program. Speed change is thoroughly used for causal discovery, as it is a good indication of traffic conditions and can lead to changes in neighboring roads. In summary, our main contributions are:

- We define legitimate traffic variables for causal structure discovery and conduct dense experiments on the PEMS-BAY dataset to learn their causal relations. We also find out that this knowledge can be generalized to the METR-LA dataset.
- We present an effective and efficient framework to capture spatial-temporal dependencies, which is named *Graph Spatial-Temporal Network Based on Causal Insight* (GT-CausIn). The core idea is to assemble causal insight, spatial dependency modeling, and temporal dependency modeling in a way that information can flexibly flow between different perspectives at different scales.
- We validate our model on two real-world public datasets and achieve state-of-the-art performance.

## 2 RELATED WORK

**Causal structural discovery** The objective of causal structural discovery is to combine statistical and logical reasoning to obtain causal relations between variables, which are represented by a graph. Fast Causal Inference (FCI) was proposed by Spirtes et al. (2000), which estimates the Markov equivalence class of causal Maximal Ancestral Graph (MAG). Recently, many papers came up with ideas to improve the high computational cost of causal inference. Zečević et al. (2021) step towards this problem by learning interventional distributions with an over-parameterized sum-product network. Akbari et al. (2021) propose a recursive constraint-based method that removes a specific type of variables at each iteration. Rohekar et al. (2021) further introduce Iterative Causal Discovery (ICD) based on FCI. Each iteration of ICD refines a graph recovered from previous iterations with a smaller set of conditions and higher statistical reliability. As a result, ICD requires fewer conditional independence tests and is more computational resources friendly. In this work, we benefit from ICD to discover the causal structure of defined variables.

**Spatial dependency** Because of its intrinsic advantages of exploring topological spaces, graph-based neural networks stand out from diverse deep learning models and achieve so far the best results. Cao et al. (2020) develop the StemGNN layer to extract in-series and intra-series correlations with the help of Graph Fourier Transformer (GFT) and Inverse Graph Fourier Transformer (IGFT). Derrow-Pinion et al. (2021) rethink the relations between road segments and collects information from node level, segment level, and super-segment level. Dynamic transition matrices are adopted in many works (Du et al., 2020; Han et al., 2021b; Shin & Yoon, 2022), where nodes with similar embeddings are assumed to have higher connection weights. However, there may be some problems with this intuitive hypothesis. Matrices are fixed after training, and embedding similarity may yaw because of sensor dysfunction and different external situations (e.g., hot/cold weather), as a result, the adaptive matrix can get biased. In this work, we employ graph diffusion layers designed by Li et al. (2018) with predefined roads as a transition matrix, which provides reliable physical connections in the real world.

**Temporal dependency** The most popular way to discover temporal dependency is through RNN-based networks. Long Short-Term Memory neural network (LSTM) is adopted in the work of Lin et al. (2018); Yao et al. (2018), while Li et al. (2018) and Jin et al. (2022) arrange RNN block in their core model structure. Geng et al. (2019) and Lv et al. (2020) apply Gated Recurrent Unit (GRU). Due to the time-consuming problem and complex gate mechanism of RNN-based networks, we assign TCN to model temporal dependency, which captures long sequences in a non-recursive manner (Han et al., 2021b; Shin & Yoon, 2022).

## 3 METHODOLOGY

In this section, we start by formulating the studied problem, then present an overview of our model structure, and end up with a detailed introduction of its crucial component layers.

### 3.1 Problem formulation

In a traffic forecasting scenario, sensors are scattered on the road to record traffic speed at a given time, and thus can be represented as a directed graph. In this work, a graph is denoted by $\mathcal{G}(\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ where $\mathcal{V}$ is the node set ($|\mathcal{V}| = N$), $\mathcal{E}$ is the edge set and $\boldsymbol{W}$ is the adjacency matrix ($|\boldsymbol{W}| = N \times N$). Main notations used in the paper are summarised in Appendix A. In the real world, the adjacency matrix is a representation of proximity and connectivity between node sensors.

Suppose each node's feature is of dimension $P$ (e.g., velocity, volume, occupancy), at timestamps $t$, a graph signal can be defined as $\boldsymbol{X}^{(t)}(\boldsymbol{X}^{t} \in \mathbb{R}^{N \times P})$. The predicted signal at time $t$ is noted as $\hat{\boldsymbol{X}}^{(t)}(\hat{\boldsymbol{X}}^{(t)} \in \mathbb{R}^{N \times Q})$, where $Q$ is the number of the desired features. We consider the prediction task as using $T'$ prior timestamps to predict $T$ following, in other words, given graph $\mathcal{G}$ and complementary information $\mathcal{I}$ (e.g., day of the week), we aim to find the best mapping function $f$ from space $\mathbb{R}^{N \times P \times T'}$ to $\mathbb{R}^{N \times Q \times T}$:

$$(\mathcal{G}; \mathcal{I}; \boldsymbol{X}^{(t-T'+1)}, \boldsymbol{X}^{(t-T')}, \cdots, \boldsymbol{X}^{(t)}) \xrightarrow{f} (\hat{\boldsymbol{X}}^{(t+1)}, \cdots, \hat{\boldsymbol{X}}^{(t+T-1)}, \cdots, \hat{\boldsymbol{X}}^{(t+T)})$$

In this work, we focus on using historical one-hour speed data for 15min, 30min, and 60min ahead speed forecasting $(P = Q = 1)$. More specifically, when the speed readings are aggregated into 5 minutes windows, it turns out to be:

$$(\mathcal{G}; \mathcal{I}; \boldsymbol{X}^{(t-11)}, \boldsymbol{X}^{(t-10)}, \cdots, \boldsymbol{X}^{(t)}) \xrightarrow{f} (\hat{\boldsymbol{X}}^{(t+3)}, \hat{\boldsymbol{X}}^{(t+6)}, \hat{\boldsymbol{X}}^{(t+12)})$$

### 3.2 Model structure overview

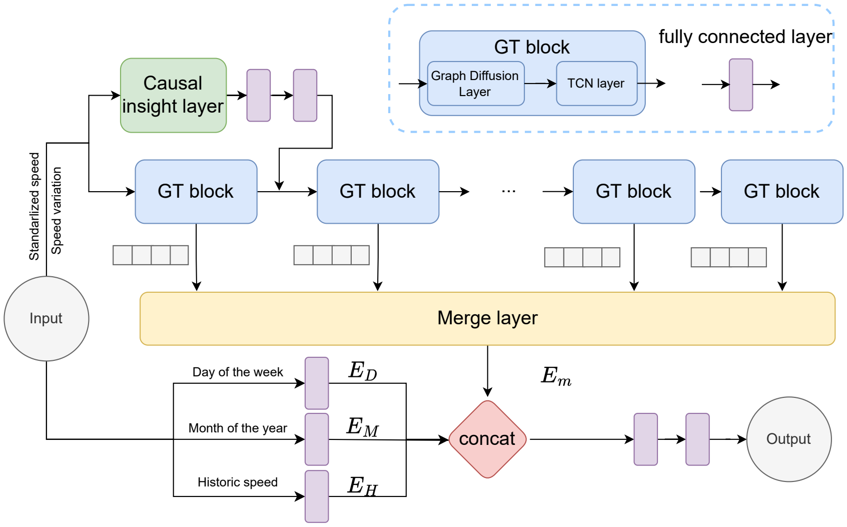


Figure 1: System architecture for the *Graph Spatial-Temporal Network Based on Causal Insight*. Time series are fed into the causal insight layer and a series of GT blocks. Skip connections are used at the end of each GT block to guard useful information. Periodic features are finally merged to make predictions.

Figure 1 is a brief illustration of GT-CausIn structure. The main components of GT-CausIn model consist of the causal insight layer, graph diffusion layer, TCN layer, merge layer, and inherent feature layer. A detailed explanation of each layer can be found in Section 3.3.

Graph diffusion layers facilitate information propagation through spatial nodes, but not inter-time series communication. TCN layer, which focuses on convolution along the temporal dimension, becomes complementary to graph diffusion layers. For the sake of expression simplicity, we name GT block as the combination of one graph diffusion layer and one TCN layer. Serializing multiple GT blocks can thus disclose inter-spatiotemporal correlations.

The input speed, together with its differentiated variant, is fed into a causal insight layer, where the neighbor-level information is carefully investigated. Two fully connected layers are stacked to better fit vector space after the first GT block.

The number of GT block $L$ is not limited to a certain value, bigger $L$ means a bigger receptive field, but model complexity is increased at the same time and important long-sequence information may be dispersed, the effects of $L$ will be further studied in Section 4.4. Skip connections are added to stack hidden states learned at different scales, which eases feature propagation for short-term and long-term prediction.

### 3.3 Component Layers

**Causal insight layer** We conduct causal structure discovery experiments on the PEMS-BAY dataset to identify correlations between stations. We hold the view that, in a speed prediction scenario, changes in the external environment and events such as accidents can be reflected in speed changes. A detailed experiment description is offered in Appendix B.

We start by introducing important notations. Given node $v_i$, we note $\mathbb{I}_1(i), \mathbb{O}_1(i)$ the set of its first-order in-neighbors/out-neighbors and $\mathbb{I}_2(i), \mathbb{O}_2(i)$ its second-order in-neighbors/out-neighbors, which can be mathematically formulated as equations below. At the graph level, we remind that the input is denoted as $\boldsymbol{X}$. The integrated features with adjacency matrix are considered as neighbor-level embedding: for the first-order in-neighbors/out-neighbors, it is denoted as $\boldsymbol{I}_1, \boldsymbol{O}_1$; for the second-order, it is denoted as $\boldsymbol{I}_2, \boldsymbol{O}_2$.

$$\mathbb{I}_1(i) = \{j | (j,i) \in \mathcal{E}, i \neq j\} \tag{1}$$

$$\mathbb{O}_1(i) = \{j | (i,j) \in \mathcal{E}, i \neq j\} \tag{2}$$

$$\mathbb{I}_2(i) = \{j | \exists k, k \notin \{i,j\}, (j,k) \in \mathcal{E}, (k,i) \in \mathcal{E}, i \neq j\} \tag{3}$$

$$\mathbb{O}_2(i) = \{j | \exists k, k \notin \{i,j\}, (i,k) \in \mathcal{E}, (k,j) \in \mathcal{E}, i \neq j\} \tag{4}$$

Defining variables is a crucial step for causal discovery. Given time $t$ and node $v_i$, for each timestamp $t_0$, we define variables to be the **speed variation** feature of $X_i^{(t_0)}, I_{1i}^{(t_0)}, O_{1i}^{(t_0)}, I_{2i}^{(t_0)}, O_{2i}^{(t_0)}$. A strong link between $v_i, \mathbb{I}_1(i)$ and $\mathbb{O}_1(i)$ is revealed from the causal discovery program, which is double-checked with statistical correlation and case analysis in Appendix C. Consequently, we design a causal insight layer to capture the correlation between $v_i, \mathbb{I}_1(i)$ and $\mathbb{O}_1(i)$.

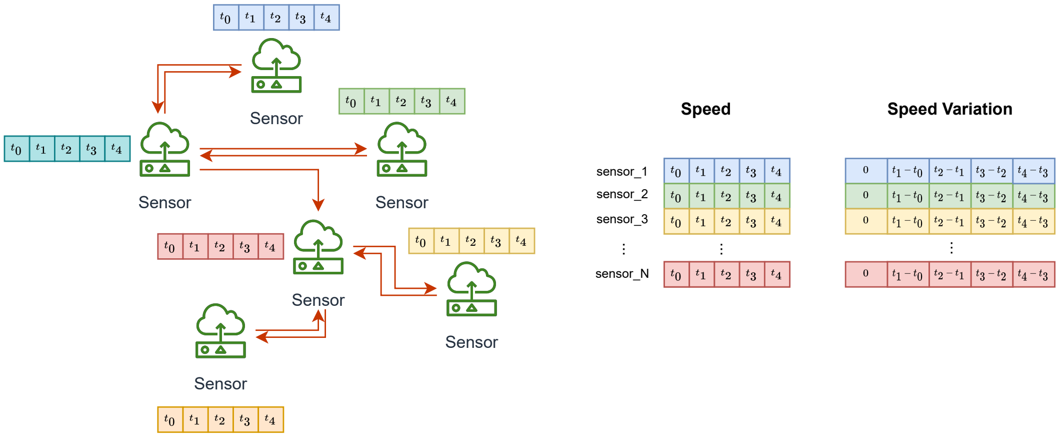


Figure 2: Example of deriving speed variation of each node with 5 timestamps as input. The speed variation is concatenated with speed to form the input of the causal insight layer.

Figure 2 represents road sensors as directed graph nodes. The speed detected by each sensor is a time series, and speed changes are obtained by differentiating adjacent time slices. With the speed variation of each node and road connection matrix, we can deduce the embedding of neighbor levels. To be cautious with the possible mutual influence between $\mathbb{I}_1(i)$, $\mathbb{O}_1(i)$, and different nodes $v_i$, we adopt a self-attention mechanism to learn inherent relations between them. Neighbor-level and

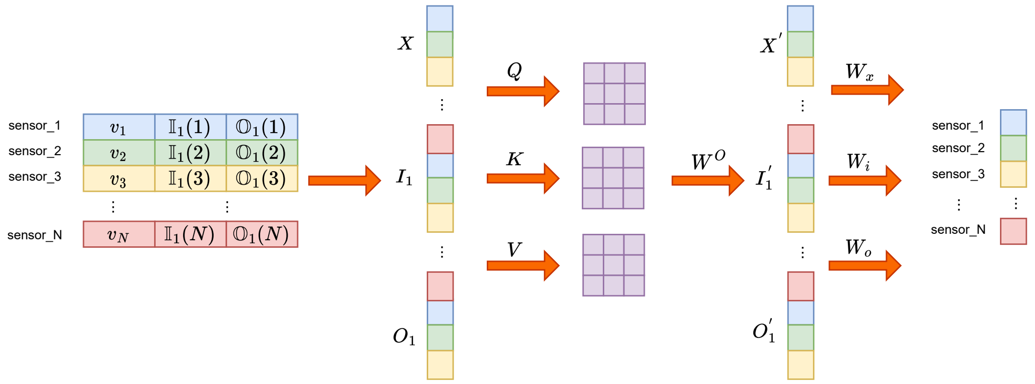


Figure 3: A brief illustration of the causal insight layer. Node features are first concatenated with neighbor embedding features, then pass to an attention layer to reveal interaction influence. Three matrices are followed to learn individual influence on each node.

node-level embedding spaces are then converted to individual node spaces. The whole process is formulated as below and is depicted in Figure 3.

$$\boldsymbol{I}_1 = \boldsymbol{D}_{I1}^{-1}(\boldsymbol{W}^T - \boldsymbol{I})\boldsymbol{X}, \boldsymbol{O}_1 = \boldsymbol{D}_{O1}^{-1}(\boldsymbol{W} - \boldsymbol{I})\boldsymbol{X} \tag{5}$$

$$\boldsymbol{S} = \boldsymbol{X} \bigoplus \boldsymbol{I}_1 \bigoplus \boldsymbol{O}_1 \tag{6}$$

$$\boldsymbol{S}^{'} = \boldsymbol{X}^{'} \bigoplus \boldsymbol{I}_1^{'} \bigoplus \boldsymbol{O}_1^{'} = \text{attention}(\mathcal{S}\boldsymbol{W}_q, \mathcal{S}\boldsymbol{W}_k, \mathcal{S}\boldsymbol{W}_v)\boldsymbol{W}^O \tag{7}$$

$$\text{output} = \boldsymbol{W}_x\boldsymbol{X}^{'} + \boldsymbol{W}_i\boldsymbol{I}_1^{'} + \boldsymbol{W}_o\boldsymbol{O}_1^{'} \tag{8}$$

where $\boldsymbol{D}_{I1} = \text{diag}((\boldsymbol{W}^T - \boldsymbol{I})\mathbf{1}), \boldsymbol{D}_{O1} = \text{diag}((\boldsymbol{W} - \boldsymbol{I})\mathbf{1})$, $\mathbf{1}$ is the all one vector.

**Graph diffusion layer** Spatial dependency is modeled in this part. Consider a directed graph $\mathcal{G}$, the diffusion process is modeled by a Markov process with the transition matrix $\boldsymbol{D}_O^{-1}\boldsymbol{W}$ and a starting probability $\alpha \in [0, 1]$ ($\boldsymbol{D}_O = \text{diag}(\boldsymbol{W}\mathbf{1})$). Teng et al. (2016) show that the ultimate distribution is converged to a stationary contribution:

$$\mathcal{P} = \sum_{k=0}^{\infty} \alpha(1-\alpha)^k(\boldsymbol{D}_O^{-1}\boldsymbol{W})^k \tag{9}$$

where $k$ is the diffusion step. Based on this theory, Li et al. (2018) present a formal expression of the graph signal diffusion process, which truncates the diffusion process to be $K$ steps and assigns trainable weights for each step.

For each input feature $p \in \{1, \cdots, P\}$ and output feature $q \in \{1, \cdots, Q\}$,

$$\boldsymbol{H}_{:,q} = \boldsymbol{X}_{:,p\star\mathcal{G}} f_{\boldsymbol{\theta}_{q,p,:,:}} = \sum_{k=0}^{K-1}(\theta_{q,p,k,1}(\boldsymbol{D}_O^{-1}\boldsymbol{W})^k + \theta_{q,p,k,2}(\boldsymbol{D}_I^{-1}\boldsymbol{W}^T)^k)\boldsymbol{X}_{:,p} \tag{10}$$

where $\boldsymbol{X}$ is the layer input, $\boldsymbol{H}$ the layer output, $\boldsymbol{\theta} \in \mathbb{R}^{Q\times P\times K\times 2}$ the filter parameter, $\boldsymbol{D}_I = \text{diag}(\mathbf{1}\boldsymbol{W})$ and $\boldsymbol{W}^T$ the transpose of graph adjacency matrix $\boldsymbol{W}$.

Different from the popular spectral GCN method proposed by Kipf & Welling (2017), which is only applicable for undirected graphs, graph diffusion layers simulate well the diffusion process of directed graphs, and become a natural choice for traffic scenarios.

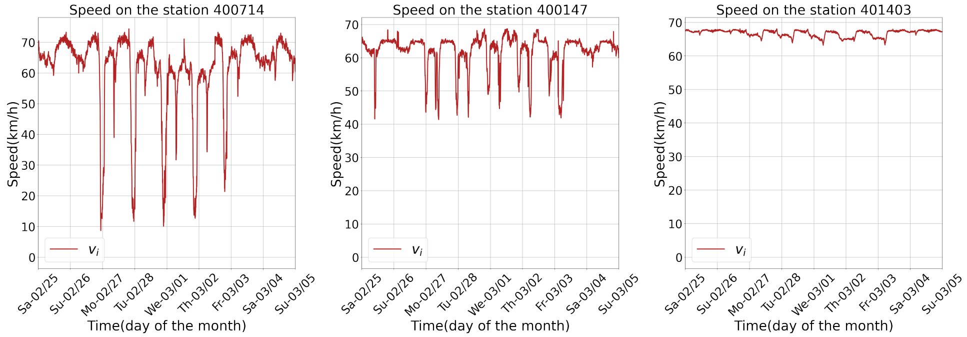
**TCN layer** We adopt the dilated causal convolution described in Yu & Koltun (2016) as our TCN layer. In contrast to RNN-based networks, dilated causal convolution can capture information over long-range sequences while getting rid of the gradient explosion problem. Concerning dilated causal convolution, stacking dilated layers ensures the exponential receptive field, and zero padding guarantees the temporal causal order since only historical information is involved to predict the current time step. Considering input feature dimension as $P$, output dimension as $Q$, given a time-series input $\boldsymbol{X} \in \mathbb{R}^{N \times P \times T}$ and a filter with trainable parameter $\boldsymbol{\theta} \in \mathbb{R}^{Q \times P \times K}$, the dilated causal convolution at time $t$ can be formulated as:

$$\boldsymbol{H}_{:,q,t} = \boldsymbol{X}_{:,p,:} \star f_{\theta_{q,p,k}}(t) = \sum_{k=0}^{K-1} \theta_{q,p,k} x_{:,p,t-d*k} \tag{11}$$

where $K$ is the kernel size, $d$ is the dilation factor, which decides the skipping distance.

**Merge layer** Although GT block extracts dependencies in the temporal and spatial dimension, information may still be lost for such a long series forecasting task. As a result, we keep the embedding after each GT block and concatenate them to enhance information flow over great lengths.

**Inherent feature layer** We notice that speed changes periodically for stations as shown in Figure 4. Hence, we extract the day of the week, the month of the year, and historic speed as inherent features. Day of the week and month of the year features are represented as one-hot vectors while historic speed is the average speed of past days at the same moment. Since people's travel routines are very different on weekdays and weekends, we only use the past five weekday records or the past two weekend records to calculate historic speed.



(a) Station 400714

(b) Station 400147

(c) Station 401403

Figure 4: Periodic speed changes of stations of PEMS-BAY within one week.

The inherent features are integrated as follows: each feature goes through a dense layer to get embedding $\boldsymbol{E}_D$ (day of the week), $\boldsymbol{E}_M$ (month of the year), and $\boldsymbol{E}_H$ (historic speed). Together with embedding $\boldsymbol{E}_m$ obtained from merge layer, all features are fused into a long feature matrix by concatenation, e.g., $\boldsymbol{F} = \boldsymbol{E}_m \bigoplus \boldsymbol{E}_D \bigoplus \boldsymbol{E}_M \bigoplus \boldsymbol{E}_H$, then we stack two fully connected layers upon $\boldsymbol{F}$ to ensure that the output is of the desired shape.

# 4 Experiments

## 4.1 Dataset description

We conduct dense experiments on two public datasets: PEMS-BAY (Li et al., 2018) and METR-LA (Jagadish et al., 2014). PEMS-BAY is collected by California Transportation Agencies (CalTrans) Performance Measurement System (PeMS). It contains 325 highway sensors of the Bay Area from Jan 1st 2017 to Jun 30th 2017. For METR-LA, 207 loop detectors on the highway of Los Angeles County are selected from Mar 1st 2012 to Jun 30th 2012. For both datasets, traffic speed readings are aggregated into 5 minutes windows. Based on the road connection map, we adopt the same way as Li et al. (2018) to get the graph adjacency matrix. With pairwise road network distances between sensors, the adjacency matrix representing connection from any sensor $i$ to any sensor $j$ is expressed as:

$$W_{i,j} = \exp\left(\frac{-\text{dist}(v_i, v_j)^2}{\sigma}\right) \quad \text{if dist}(v_i, v_j) \leq \kappa, \text{otherwise } 0 \tag{12}$$

where $\sigma$ is the standard deviation, $\kappa$ the threshold, and dist$(i, j)$ the distance between station $v_i$ and station $v_j$. For both datasets, 80% of the data serves as the training set, 10% as the test set, and the remaining 10% as the validation set.

Table 1: Performance comparison with baseline models for traffic speed forecasting. The improvement between GT-CausIn and the state-of-the-art model is significant on most metrics, and the margin becomes more evident with the increase of the forecasting horizon. GT-CausIn also shows a global advantage over GT-NoCausIn, where the causal insight layer is taken off from GT-CausIn.

| **Data** | **Method** | **15min** | | | **30min** | | | **60min** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | MAE | RMSE | MAPE | MAE | RMSE | MAPE | MAE | RMSE | MAPE |
| METR-LA | DCRNN | 2.77 | 5.38 | 7.30% | 3.15 | 6.45 | 8.8% | 3.60 | 7.59 | 10.5% |
| | Graph Wavenet | 2.69 | 5.15 | 6.90% | 3.07 | 6.22 | 8.37% | 3.53 | 7.37 | 10.01% |
| | GMAN | 4.04 | 8.53 | 10.2% | 4.59 | 9.85 | 11.69% | 5.33 | 11.21 | 13.60% |
| | ST-GRAT | 2.60 | 5.07 | 6.61% | 3.01 | 6.21 | 8.15% | 3.49 | 7.42 | 10.01% |
| | SLCNN | **2.53** | 5.18 | 6.70% | 2.88 | 6.15 | 8.00% | 3.30 | 7.20 | 9.70% |
| | DGCRN | 2.62 | **5.01** | 6.63% | 2.99 | 6.05 | 8.02% | 3.44 | 7.19 | 9.73% |
| | DMSTGCN | 2.85 | 5.54 | 7.54% | 3.26 | 6.56 | 9.19% | 3.72 | 7.55 | 10.96% |
| | PGCN | 2.70 | 5.16 | 6.98% | 3.08 | 6.22 | 8.38% | 3.54 | 7.36 | 9.94% |
| | GT-NoCausIn | 2.82 | 5.52 | 7.18% | 3.27 | 6.71 | 8.90% | 3.95 | 8.24 | 11.67% |
| | GT-CausIn | 2.61 | 5.07 | **6.60%** | **2.73** | **5.34** | **7.09%** | **3.06** | **6.11** | **8.30%** |
| | Improvement | -3.2% | -1.2% | 0.2% | 5.2% | 11.7% | 11.4% | 7.3% | 15.0% | 14.4% |
| PEMS-BAY | DCRNN | 1.38 | 2.95 | 2.90% | 1.74 | 3.97 | 3.90% | 2.07 | 4.74 | 4.90% |
| | Graph Wavenet | 1.30 | 2.74 | 2.73% | 1.63 | 3.70 | 3.67% | 1.95 | 4.52 | 4.63% |
| | GMAN | 1.34 | 2.82 | 2.81% | 1.62 | 3.72 | 3.63% | 1.86 | 4.32 | 4.31% |
| | ST-GRAT | 1.29 | 2.71 | 2.67% | 1.61 | 3.69 | 3.63% | 1.95 | 4.54 | 4.64% |
| | SLCNN | 1.44 | 2.90 | 3.00% | 1.72 | 3.81 | 3.90% | 2.03 | 4.53 | 4.80% |
| | DGCRN | **1.28** | 2.69 | **2.66%** | 1.59 | 3.63 | 3.55% | 1.89 | 4.42 | 4.43% |
| | DMSTGCN | 1.33 | 2.83 | 2.80% | 1.67 | 3.79 | 3.81% | 1.99 | 4.54 | 4.78% |
| | PGCN | 1.30 | 2.73 | 2.72% | 1.62 | 3.67 | 3.63% | 1.92 | 4.45 | 4.55% |
| | GT-NoCausIn | 1.43 | 2.85 | 3.00% | 1.79 | 3.84 | 4.01% | 2.25 | 4.90 | 5.44% |
| | GT-CausIn | 1.30 | **2.54** | 2.68% | **1.47** | **2.92** | **3.10%** | **1.70** | **3.37** | **3.68%** |
| | Improvement | -1.6% | 5.6% | -0.8% | 7.5% | 19.6% | 12.7% | 8.6% | 22.0% | 14.6% |

### 4.2 Experiment setting

All models are implemented with PyTorch (Paszke et al., 2019) and trained with an Adam optimizer with an annealing learning rate. Experiments on dataset METR-LA are carried out on two Nvidia Tesla V100 servers and PEMS-BAY on two Nvidia Tesla A100 servers. The detailed parameter setting for all experiments executed is available in Appendix F.

All methods are evaluated based on three commonly used metrics in traffic forecasting, which are Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE). The formula of all metrics can be found in Appendix E. MAE is used as the loss function. Missing values are linearly filled for training, and are excluded for results evaluation.

### 4.3 Comparison with baselines

We compare GT-CausIn with various popular traffic forecasting models in recent years, including (1) DCRNN: Diffusion Convolutional Recurrent Neural Network (Li et al., 2018); (2) Graph Wavenet (Wu et al., 2019), which develops an adaptive dependency matrix based on node embedding; (3)GMAN: Graph Multi-Attention Network for Traffic Prediction (Zheng et al., 2020), of which the main structure is an encoder and a decoder both with several attention blocks; (4) ST-GRAT: Spatio-temporal Graph Attention Networks for Accurately Forecasting Dynamically Changing Road Speed (Park et al., 2020), which includes spatial attention, temporal attention, and spatial sentinel vectors; (5) SLCNN: Structure Learning Convolution Neural Network (Zhang et al., 2020), which learns the graph structure information with convolutional methods; (6) DGCRN: Dynamic Graph Convolutional Recurrent Network (DGCRN) (Li et al., 2021), which filters node embedding to generate dynamic graph at each time step; (7) DMSTGCN: Dynamic and Multi-faceted Spatio-temporal Deep Learning for Traffic Speed Forecasting (Han et al., 2021a), which provides a multi-faceted fusion module to incorporate the hidden states learned at different stages; (8) PGCN: Progressive Graph Convolutional Networks for Spatial-temporal Traffic Forecasting (Shin & Yoon, 2022), which constructs progressive adjacency matrices by learning in training and test phases.

Our approach is different from all the approaches above, we first use a causal discovery program to discover a general rule between node neighbors of different orders, then build a model named

Table 2: Ablation study on causal effects, GT-BadCausIn integrates no neighbor embedding in causal insight layer. GT-CausIn achieves consistent better results than GT-BadCausIn.

| **Data** | **Method** | **15min** | | | **30min** | | | **60min** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | MAE | RMSE | MAPE | MAE | RMSE | MAPE | MAE | RMSE | MAPE |
| METR-LA | GT-BadCausIn | 2.65 | 5.17 | 6.72% | 2.86 | 5.71 | 7.58% | 3.18 | 6.50 | 8.85% |
| | GT-CausIn | 2.61 | 5.07 | 6.60% | 2.73 | 5.34 | 7.09% | 3.06 | 6.11 | 8.30% |
| | Improvement | 1.5% | 1.9% | 1.8% | 4.5% | 6.5% | 6.5% | 3.8% | 6.0% | 6.2% |
| PEMS-BAY | GT-BadCausIn | 1.35 | 2.63 | 2.80% | 1.56 | 3.13 | 3.34% | 1.81 | 3.64 | 4.01% |
| | GT-CausIn | 1.30 | 2.54 | 2.68% | 1.47 | 2.92 | 3.10% | 1.70 | 3.37 | 3.68% |
| | Improvement | 3.7% | 3.4% | 4.3% | 5.8% | 6.7% | 7.2% | 5.0% | 7.4% | 8.2% |

GT-CausIn to enhance prediction performance. Although the causality relation is discovered with dataset PEMS-BAY, it generalizes as well on the other dataset METR-LA. Besides, the proposed causal insight layer is integrated with directed graph diffusion layers to model the spatial dependency. TCN layers and skip connections are further used to discover the temporal dependency.

A comparison of GT-CausIn and other baseline models is shown in Table 1. The causal insight layer is taken out from GT-CausIn and the rest model is named as GT-NoCausIn. All the baseline results are taken from its original paper if it is available and from Shin & Yoon (2022) if not[1]. We notice that models with dynamic training graph generally outperforms other models, however, as Table 1 indicates, GT-CausIn achieves the best overall result, especially for mid-term (30min) and long-term (60min) prediction. We need to remark that DCRNN (Li et al., 2018) shares similar graph diffusion layers and GMAN (Zheng et al., 2020) widely adopt the attention mechanism, yet our model exceeds them on all metrics, which justifies our model structure. The large margin between GT-NoCausIn and GT-CausIn shows that causal insights can significantly improve prediction accuracy.

### 4.4 Ablation study

**Causal effects** In GT-CausIn, we put special attention to relations between node $v_i$, $\mathbb{I}_1(i)$ and $\mathbb{O}_1(i)$, nonetheless, this conclusion drawn from the causal discovery program may be questioned. Therefore, we replace Equation (6, 7, 8) as below to check its effectiveness, naming this model GT-BadCausIn.

$$\boldsymbol{S} = \boldsymbol{X} \bigoplus \boldsymbol{X} \bigoplus \boldsymbol{X} \tag{13}$$

$$\boldsymbol{S}^{'} = \boldsymbol{X}_0^{'} \bigoplus \boldsymbol{X}_1^{'} \bigoplus \boldsymbol{X}_2^{'} = \text{attention}(\mathcal{S}\boldsymbol{W}_q, \mathcal{S}\boldsymbol{W}_k, \mathcal{S}\boldsymbol{W}_v)\boldsymbol{W}^O \tag{14}$$

$$\text{output} = \boldsymbol{W}_{x_0}\boldsymbol{X}_0^{'} + \boldsymbol{W}_{x_1}\boldsymbol{X}_1^{'} + \boldsymbol{W}_{x_2}\boldsymbol{X}_2^{'} \tag{15}$$

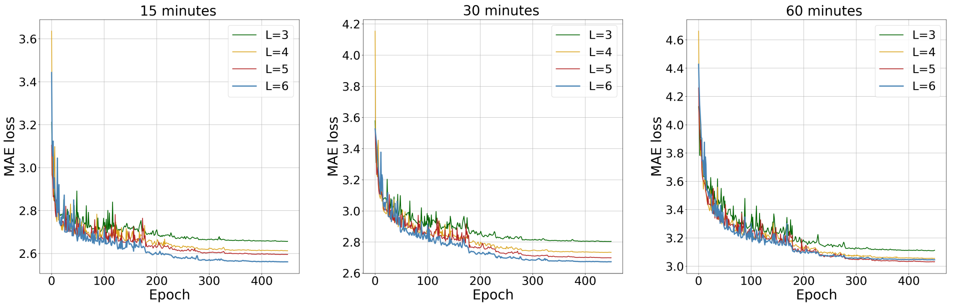
Experiments results are summarised in Table 2. The impact brought by neighbor embedding in the causal insight layer is positive for all metrics on both datasets, especially for mid-term (30min) and long-term (60min) predictions. The reason behind may be the propagation time of causal effects. Besides, we can observe that the improvement is smaller on METR-LA than on PEMS-BAY, which may result from: (1) a larger percentage of missing values in METR-LA (8.11%) than in PEMS-BAY (0.003%); (2) causal discovery program is only executed on the PEMS-BAY dataset.

**Number of stacking GT blocks** In Section 3.2, we introduced that the number of GT blocks $L$ is not limited to a certain value. Larger $L$ roughly corresponds to a larger spatiotemporal receptive field and enables the model to capture broader spatiotemporal dependency. We compare model performance with different $L \in \{3, 4, 5, 6\}$ on MAE of different time horizons in Figure 5. We can observe a steady rise when $L$ goes up. Since the gap between $L = 3$ and $L = 4$ is more important than between $L = 4$ and $L = 5$, taking the increased model complexity into account, we adopt $L = 4$ as our best parameter.

### 4.5 Model interpretation

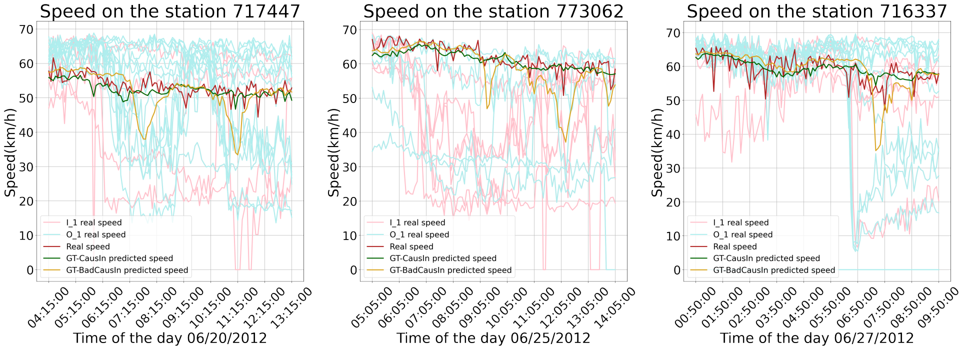
To better understand our model, we compare it with GT-BadCausIn on 60min ahead prediction task. We shift the input window to get continuous predictions in Figure 6, observing a notable per-

[1] Only GMAN and DMSTGCN authors implement experiments on different datasets.



(a) MAE/15min (b) MAE/30min (c) MAE/60min

Figure 5: Parameter analysis of $L$ (number of stacking GT layers) on METR-LA dataset



(a) Station 71447 (b) Station 773062 (c) Station 716337

Figure 6: Figures (a), (b), and (c) offer a comparison between GT-CausIn and GT-BadCausin for prediction task on METR-LA, differing on the target station and target time. Each figure contains the ground truth of the considered station and its neighbors, as well as its 60min ahead prediction. It is notable that the predictions of GT-BadCausIn are heavily dependent on some neighbors, and are therefore swayed off the ground truth. The same cannot be said for GT-causIn, instead of being influenced by some stations, it has a broader outlook of its neighbors, resulting in a much more faithful and smooth prediction.

formance margin between GT-CausIn and GT-BadCausIn. As described in Section 4.4, the only structural difference between these two models is the dissimilar perspectives integrated. For GT-BadCausIn, only node-level information $\boldsymbol{X}$ is investigated. For GT-CausIn, neighbor-level information $\boldsymbol{I}$ and $\boldsymbol{O}$ are also included. This structural change results in different token attention scores of the causal insight layer and is responsible for this disparity, this is further investigated in Appendix D.

## 5 Conclusion

In this paper, we implemented knowledge learned from a causal discovery program with deep learning models and proposed *Graph Spatial-Temporal Network Based on Causal Insight* to capture spatiotemporal dependencies. Specifically, we serialize graph diffusion layers and TCN layers to capture dependencies between spatial flow and temporal flow, we also use skip connections to guarantee information propagation for long sequences. We creatively design a causal insight layer that focuses on stations, their first-order in-neighbors, and their first-order out-neighbors. When evaluating our model on two real-world traffic datasets, we achieved significantly better overall prediction than baselines. We also conducted ablation studies to show the effectiveness of causal knowledge and the influence of the stacking layer number. In the future, we plan to (1) apply the proposed model to other spatial-temporal forecasting tasks; (2) investigate scalable methods to apply to large-scale datasets.

## References

Sina Akbari, Ehsan Mokhtarian, AmirEmad Ghassami, and Negar Kiyavash. Recursive causal structure learning in the presence of latent variables and selection bias. *Advances in Neural Information Processing Systems*, 34:10119–10130, 2021.

Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Congrui Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, et al. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in neural information processing systems*, 33:17766–17778, 2020.

Ennio Cascetta. *Transportation systems engineering: theory and methods*, volume 49. Springer Science & Business Media, 2013.

Chris Cundy, Aditya Grover, and Stefano Ermon. Bcd nets: Scalable variational approaches for bayesian causal discovery. *Advances in Neural Information Processing Systems*, 34:7095–7110, 2021.

Austin Derrow-Pinion, Jennifer She, David Wong, Oliver Lange, Todd Hester, Luis Perez, Marc Nunkesser, Seongjae Lee, Xueying Guo, Brett Wiltshire, et al. Eta prediction with graph neural networks in google maps. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 3767–3776, 2021.

Bowen Du, Xiao Hu, Leilei Sun, Junming Liu, Yanan Qiao, and Weifeng Lv. Traffic demand prediction based on dynamic transition convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems*, 22(2):1237–1247, 2020.

Theodoros Evgeniou, Massimiliano Pontil, and Tomaso Poggio. Regularization networks and support vector machines. *Advances in computational mathematics*, 13(1):1–50, 2000.

D Freedman and R Pisani. R. purves: Statistics, 2007.

Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 3656–3663, 2019.

Shengnan Guo, Youfang Lin, Shijie Li, Zhaoming Chen, and Huaiyu Wan. Deep spatial–temporal 3d convolutional neural networks for traffic data forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3913–3926, 2019.

Mohammad M Hamed, Hashem R Al-Masaeid, and Zahi M Bani Said. Short-term prediction of traffic volume in urban arterials. *Journal of Transportation Engineering*, 121(3):249–254, 1995.

Liangzhe Han, Bowen Du, Leilei Sun, Yanjie Fu, Yisheng Lv, and Hui Xiong. Dynamic and multi-faceted spatio-temporal deep learning for traffic speed forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery &amp; Data Mining*, KDD '21, pp. 547–555, New York, NY, USA, 2021a. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467275. URL https://doi.org/10.1145/3447548.3467275.

Liangzhe Han, Bowen Du, Leilei Sun, Yanjie Fu, Yisheng Lv, and Hui Xiong. Dynamic and multi-faceted spatio-temporal deep learning for traffic speed forecasting. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 547–555, 2021b.

H. V. Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M. Patel, Raghu Ramakrishnan, and Cyrus Shahabi. Big data and its technical challenges. *Commun. ACM*, 57(7):86–94, jul 2014. ISSN 0001-0782. doi: 10.1145/2611567. URL https://doi.org/10.1145/2611567.

Junchen Jin, Dingding Rong, Tong Zhang, Qingyuan Ji, Haifeng Guo, Yisheng Lv, Xiaoliang Ma, and Fei-Yue Wang. A gan-based short-term link traffic prediction approach for urban road networks under a parallel learning framework. *IEEE Transactions on Intelligent Transportation Systems*, 2022.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=SJU4ayYgl.

Fuxian Li, Jie Feng, Huan Yan, Guangyin Jin, Fan Yang, Funing Sun, Depeng Jin, and Yong Li. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2021.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=SJiHXGWAZ.

Lei Lin, Zhengbing He, and Srinivas Peeta. Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach. *Transportation Research Part C: Emerging Technologies*, 97:258–276, 2018.

Mingqi Lv, Zhaoxiong Hong, Ling Chen, Tieming Chen, Tiantian Zhu, and Shouling Ji. Temporal multi-graph convolutional network for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 22(6):3337–3348, 2020.

Xiaolei Ma, Zhuang Dai, Zhengbing He, Jihui Ma, Yong Wang, and Yunpeng Wang. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, 17(4):818, 2017.

Attila M. Nagy and Vilmos Simon. Survey on traffic prediction in smart cities. *Pervasive and Mobile Computing*, 50:148–163, 2018. ISSN 1574-1192. doi: https://doi.org/10.1016/j.pmcj.2018.07.004. URL https://www.sciencedirect.com/science/article/pii/S1574119217306521.

Cheonbok Park, Chunggi Lee, Hyojin Bahng, Yunwon Tae, Seungmin Jin, Kihwan Kim, Sungahn Ko, and Jaegul Choo. St-grat: A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed. CIKM '20, pp. 1215–1224, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368599. doi: 10.1145/3340531.3411940. URL https://doi.org/10.1145/3340531.3411940.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 8024–8035, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

Judea Pearl. *Causality*. Cambridge university press, 2009.

SY Pranay and Nithin Nagaraj. Causal discovery using compression-complexity measures. *Journal of Biomedical Informatics*, 117:103724, 2021.

Raanan Y Rohekar, Shami Nisimov, Yaniv Gurwicz, and Gal Novik. Iterative causal discovery in the possible presence of latent confounders and selection bias. *Advances in Neural Information Processing Systems*, 34:2454–2465, 2021.

Yuyol Shin and Yoonjin Yoon. Pgcn: Progressive graph convolutional networks for spatial-temporal traffic forecasting. *arXiv preprint arXiv:2202.08982*, 2022.

Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

Pater Spirtes, Clark Glymour, Richard Scheines, Stuart Kauffman, Valerio Aimale, and Frank Wimberly. Constructing bayesian network models of gene expression networks from microarray data. 2000.

Shang-Hua Teng et al. Scalable algorithms for data and network analysis. *Foundations and Trends® in Theoretical Computer Science*, 12(1–2):1–274, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Billy M Williams and Lester A Hoel. Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. *Journal of transportation engineering*, 129(6):664–672, 2003.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 1907–1913. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/264. URL `https://doi.org/10.24963/ijcai.2019/264`.

Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In Yoshua Bengio and Yann LeCun (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL `http://arxiv.org/abs/1511.07122`.

Matej Zečević, Devendra Dhami, Athresh Karanam, Sriraam Natarajan, and Kristian Kersting. Interventional sum-product networks: Causal inference with tractable probabilistic models. *Advances in Neural Information Processing Systems*, 34:15019–15031, 2021.

Qi Zhang, Jianlong Chang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Spatio-temporal graph structure learning for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 1177–1185, 2020.

Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 1234–1241, 2020.

# A Notation

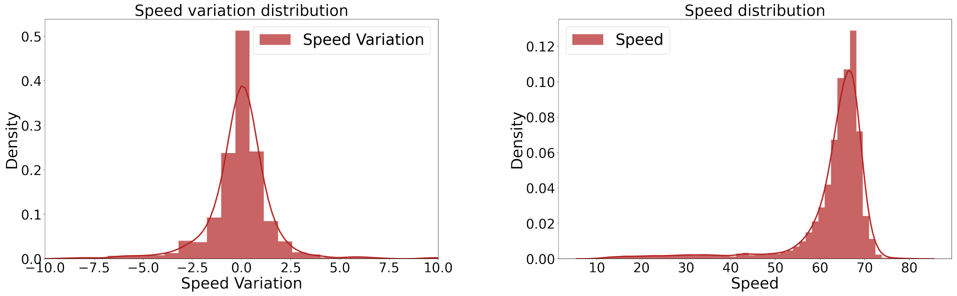
Table 3: Core concept definitions

| Concept | Description |
|---|---|
| $\mathcal{G}(\mathcal{V}, \mathcal{E})$ | Graph with node set $\mathcal{V}$ and edge set $\mathcal{E}$, $\lvert\mathcal{V}\rvert = N$ |
| $v_i$ | The $i$-th node |
| $\boldsymbol{W}$ | Adjacency matrix of graph, $\boldsymbol{W} \in \mathbb{R}^{N\times N}$ |
| $\boldsymbol{D}_I, \boldsymbol{D}_O$ | In-degree/out-degree matrix |
| $\mathbf{1}$ | All one vector |
| $\boldsymbol{X}, \hat{\boldsymbol{X}}$ | Graph signal and the predicted graph signal |
| $\boldsymbol{I}$ | Identity matrix |
| $\mathbb{I}_1(i), \mathbb{O}_1(i)$ | Set of first-order in-neighbors/out-neighbors of $i$-th node |
| $\mathbb{I}_2(i), \mathbb{O}_2(i)$ | Set of second-order in-neighbors/out-neighbors of $i$-th node |
| $\boldsymbol{I}_1, \boldsymbol{O}_1$ | Integrated embedding of first-order in-neighbors/out-neighbors |
| $\boldsymbol{I}_2, \boldsymbol{O}_2$ | Integrated embedding of second-order in-neighbors/out-neighbors |

# B Causal discovery experiment setting

The causal structure discovery analyzes the causal relation between different variables. In this work, we adopt ICD (Rohekar et al., 2021) as our causal discovery program. The input of ICD is a set of variable values, and the output is the graph adjacency matrix, in which each element represents the importance of the causal relation between variables. The algorithm is only implemented on the PEMS-BAY dataset since there are too many missing values in METR-LA (8.11%) than in PEMS-BAY (0.003%).

Defining variables is a crucial step for causal discovery. Compared to the speed itself, the speed variation overall represents better the changes in traffic volume as well as the occurrence of external events, as it more clearly expresses the changes in traffic. What's more, Figure 7 shows that speed variation is a natural Gaussian distribution, which is necessarily required for ICD.



(a) Speed variation distribution

(b) Speed distribution

Figure 7: PEMS-BAY data distribution.

Given node $v_i$, we note $\mathbb{I}_1(i), \mathbb{O}_1(i)$ the set of its first-order in-neighbors/out-neighbors and $\mathbb{I}_2(i), \mathbb{O}_2(i)$ its second-order in-neighbors/out-neighbors. At the graph level, the input is denoted as $\boldsymbol{X}$, the integrated features along the adjacency matrix of first-order in-neighbors/out-neighbors are denoted as $\boldsymbol{I}_1, \boldsymbol{O}_1$ and second-order as $\boldsymbol{I}_2, \boldsymbol{O}_2$, which can be mathematically expressed as below.

$$\boldsymbol{W}_{I1} = \boldsymbol{W}^T - \boldsymbol{I}, \boldsymbol{W}_{O1} = \boldsymbol{W} - \boldsymbol{I} \tag{16}$$

$$\boldsymbol{I}_1 = (\text{diag}(\boldsymbol{W}_{I1}\mathbf{1})^{-1}\boldsymbol{W}_{I1})\boldsymbol{X}, \boldsymbol{O}_1 = (\text{diag}(\boldsymbol{W}_{O1}\mathbf{1})^{-1}\boldsymbol{W}_{O1})\boldsymbol{X} \tag{17}$$

$$\boldsymbol{W}_{I2} = \boldsymbol{W}_{I1}\boldsymbol{W}_{I1}, \text{diag}(\boldsymbol{W}_{I2}) = 0, \boldsymbol{W}_{O2} = \boldsymbol{W}_{O1}\boldsymbol{W}_{O1}, \text{diag}(\boldsymbol{W}_{O2}) = 0 \tag{18}$$

$$\boldsymbol{I}_2 = (\text{diag}(\boldsymbol{W}_{I2}\mathbf{1})^{-1}\boldsymbol{W}_{I2})\boldsymbol{X}, \boldsymbol{O}_2 = (\text{diag}(\boldsymbol{W}_{O2}\mathbf{1})^{-1}\boldsymbol{W}_{O2})\boldsymbol{X} \tag{19}$$

Given time $t$ and node $i$, for each timestamp $t_0$ between $t$ and $t + 30\,\mathrm{min}$, we define variables to be the speed variation of $X_i^{(t_0)}, I_{1i}^{(t_0)}, O_{1i}^{(t_0)}, I_{2i}^{(t_0)}, O_{2i}^{(t_0)}$. Since the PEMS-BAY dataset aggregates traffic speed readings into 5 minutes windows, causal variables will be stored in a set of 6-time slices, resulting in 30 causal inferred variables below. The output of ICD is thus a $30 \times 30$ matrix $\boldsymbol{C}$, where $C_{i,j}$ represents the causal relation between variable $i$ and variable $j$.

$$\begin{aligned}\mathbb{V} &= \{V_0, V_1, \ldots, V_{30}\} \\ &= \{X_i^{(t)}, I_{1i}^{(t)}, O_{1i}^{(t)}, I_{2i}^{(t)}, O_{2i}^{(t)}, \cdots, X_i^{(t+5)}, I_{1i}^{(t+5)}, O_{1i}^{(t+5)}, I_{2i}^{(t+5)}, O_{2i}^{(t+5)}\}\end{aligned} \tag{20}$$

We randomly sample data in the dataset and then process it to obtain the above 30 causal variables. To reduce accidental error, we sample 2000 samples each time as input of ICD, and repeat 100 times to sum the output matrix $\boldsymbol{C}$.

## C Causal result validation

In this section, we carefully validate causal results obtained in Appendix B in two ways: (1) coherence with the correlation between variables; (2) case analysis.

**Correlation between variables** We apply the sampling algorithm mentioned in Appendix B to the random data and the accident data, and adopt Pearson correlation coefficient (Freedman & Pisani, 2007) to calculate the correlation between the different variables. The result can be plotted in Figure 8, in which the size of the circle indicates the strength of its relations.

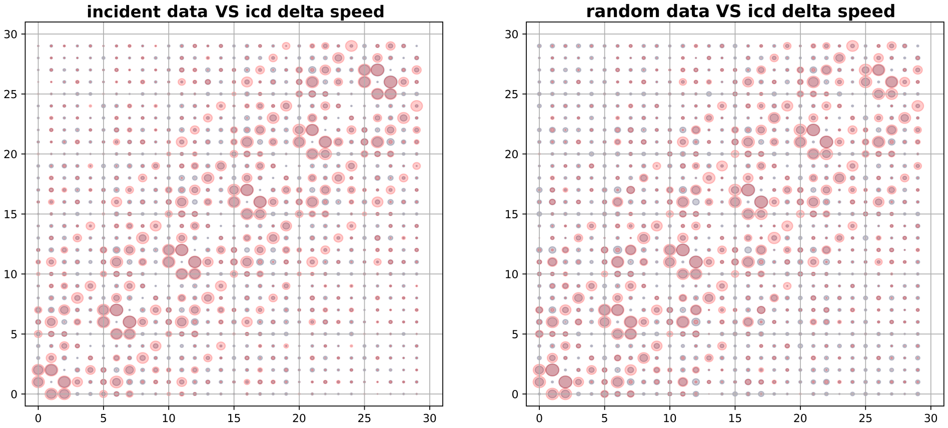


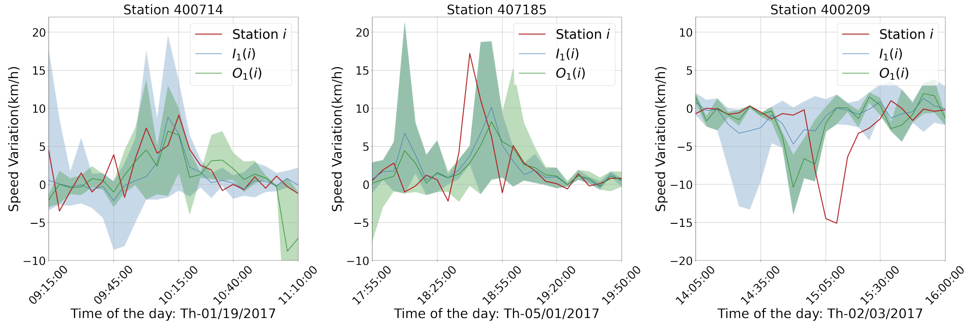
Figure 8: ICD result vs data correlation, on the left: accident data, on the right: random data. Labels represent the variables defined in Equation (20). Red circles indicate results deduced from ICD. Gray circles indicate the Pearson correlation coefficients between variables mined from incident data (left) and random data (right).

Four phenomena can be observed in Figure 8: (1) The relation obtained by causal discovery holds for both random data and accident data, indicating that the speed changes at one station do have an effect on neighboring stations and that this law is universal, without the need to include an additional judgment about whether an accident has occurred; (2) The causality is mainly reflected in spatial and temporal space, there is no clear spatiotemporal intertwined causality; (3) There is another interesting effect that the station $v_i$, its out-neighbor $\mathbb{O}_1(i)$ and in-neighbor $\mathbb{I}_1(i)$ are causally related to each other; (4) For relations in the temporal dimension, variables are more connected to themselves from $t$ to $t+1$ and $t+2$.

**Case analysis** In the real world there may be many reasons behind speed changes, such as sudden traffic jams, weather phenomena, and car accidents. In light of these unpredictable external factors, and to ensure that the results and conclusions on causality seen before are founded in reality, it is essential to plot the speed of a node and its neighbors over time and analyze its behaviors.

To do so, a dataset of notable events is created. These events are characterized by a sudden and lasting spike in speed variation, positive or negative. Due to the large scale of the PEMS-BAY dataset, it is necessary to sample random mini-batches and calculate the weighted speed change in the station and its neighbors. Sorting through these average velocity fluctuations, the most influential ones are selected, i.e. highest and lowest peaks.

According to these notable events seen in Figure 9, we can confirm the mutual impacts of speed changes between a station, its in-neighbors, and its out-neighbors. In most cases, these three entities share a similar pattern for speed variation, as shown in Figures 9a and 9b. We also observe a delayed impact in some cases, as shown in Figure 9c, where the speed variation is spread from its out-neighbors.



(a) Event at station 400714 (b) Event at station 407185 (c) Event at station 400209

Figure 9: Notable events displaying the correlation between the station and its neighbors. Curves $I_1(i)$ and $O_1(i)$ denote the weighted sum formulated in Equation (17) and shadow gives the speed variation interval of the first-order in-neighbors/out-neighbors.

## D Attention score analysis

In this section we focus on the 60min ahead prediction task on station 717477 of METR-LA at 11:15:00. As seen in Figure 10, there is a performance gain for GT-CausIn in comparison to GT-BadCausIn model. This is due to a structural difference seen in 4.4 which includes neighbor-level information in the causal insight layer.

To look into what is happening inside the causal insight layer, we take out the score matrices. We revise that the attention scores are calculated by processing the score matrix on a softmax layer (Vaswani et al., 2017), which guarantees that the sum of each row equals one. Given input $\boldsymbol{X}$, the attention layer is formulated as below.

$$\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_q, \boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_k, \boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_v \tag{21}$$

$$\text{attention} = \text{softmax}\frac{(\boldsymbol{Q}\boldsymbol{K}^T)}{\sqrt{d_k}}\boldsymbol{V} \tag{22}$$

where $\boldsymbol{W}_q, \boldsymbol{W}_k$ and $\boldsymbol{W}_v$ are trainable weights, $\frac{1}{\sqrt{d_k}}$ is the scalar factor, $\boldsymbol{Q}$ is query matrix, $\boldsymbol{K}$ is the key matrix and $\boldsymbol{V}$ is the value matrix. The attention score is $\boldsymbol{S} = \frac{(\boldsymbol{Q}\boldsymbol{K}^T)}{\sqrt{d_k}}$, and its element $S_{i,j}$ weights the attention $\boldsymbol{X}_i$ pays to $\boldsymbol{X}_j$.

In our case, the attention input is $\boldsymbol{S} = \boldsymbol{X} \bigoplus \boldsymbol{I}_1 \bigoplus \boldsymbol{O}_1$ for GT-CausIn and $\boldsymbol{S} = \boldsymbol{X} \bigoplus \boldsymbol{X} \bigoplus \boldsymbol{X}$ for GT-BadCausIn, as indicated in Equation (7) for GT-CausIn and Equation (14) for GT-BadCausIn. We take rows corresponding to the station $v_i$ to focus on how it is impacted. Afterward, we subdivide the columns into three parts, each part contains tokens of all graph nodes from different perspectives, as noted by the labels in the bottom.

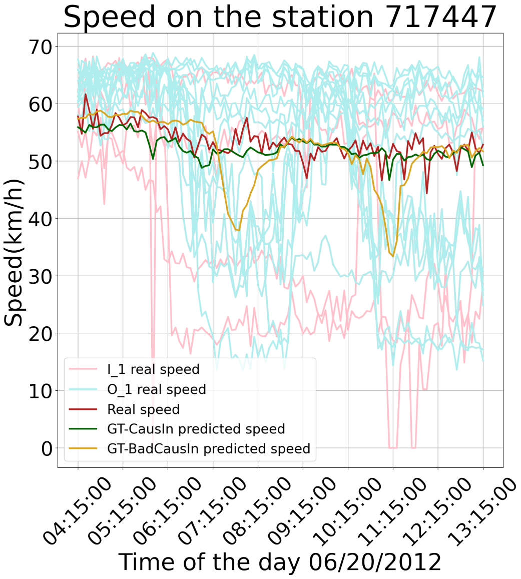


Figure 10: Ground truth of station 71447/neighbors, 60min ahead prediction speed of station 71447.

With these tokens in hand, the attention scores are visualized with heat maps. Figure 11 shows attention scores inside the GT-BadCausIn model, that includes solely $\boldsymbol{X}$ tokens. In contrast, in Figure 12, the attention scores inside GT-CausIn are not constrained to $\boldsymbol{X}$, but also including $\boldsymbol{I}_1$ and $\boldsymbol{O}_1$ tokens. Finally the performance difference seen in Figure 10 can be explained by different attention scores in the Figures 11 and 12. In the first case, the attention is mainly distributed on some nodes. In the second case, the attention has a global view on all the nodes thanks to neighbor-level tokens, e.g., $\boldsymbol{I}_1, \boldsymbol{O}_1$.

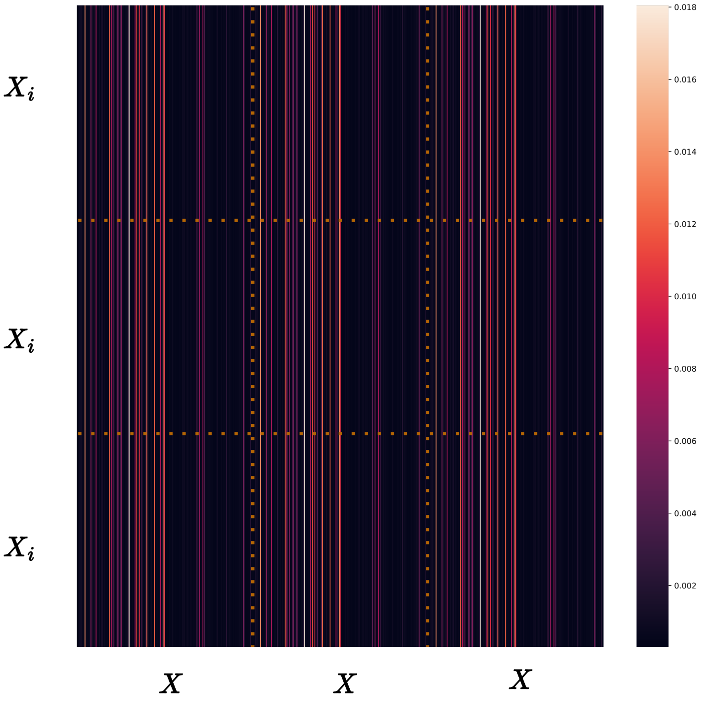


Figure 11: Heat map of attention score in causal insight layer for GT-BadCausIn

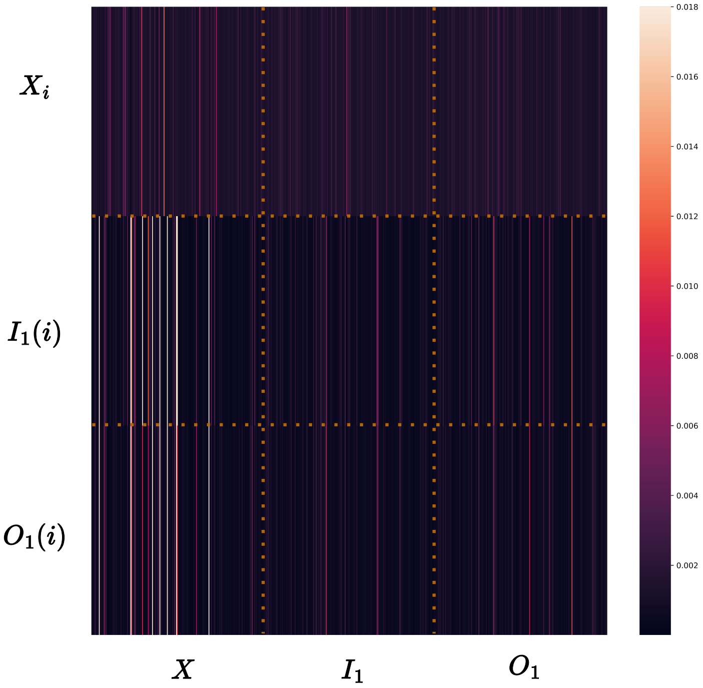


Figure 12: Heat map of attention score in causal insight layer for GT-CausIn

# E Metrics

Note $\boldsymbol{x} = \{x_1, \cdots, x_n\}$ the ground truth, $\hat{\boldsymbol{x}} = \{\hat{x}_1, \cdots, \hat{x}_n\}$ predicted value, and $\Omega$ indices of observed samples, the metrics are defined as follows.

$$\text{MAE}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{|\boldsymbol{\Omega}|} \sum_{i \in \boldsymbol{\Omega}} |x_i - \hat{x}_i| \tag{23}$$

$$\text{RMSE}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \sqrt{\frac{1}{|\boldsymbol{\Omega}|} \sum_{i \in \boldsymbol{\Omega}} |(x_i - \hat{x}_i)^2} \tag{24}$$

$$\text{MAPE}(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{1}{|\boldsymbol{\Omega}|} \sum_{i \in \boldsymbol{\Omega}} |\frac{x_i - \hat{x}_i}{x_i}| \tag{25}$$

# F Detailed experimental settings

For all models, we stack $L = 4$ GT blocks (except ablation study on $L$) and the output dimension of each block is 8. In each graph diffusion layer, the maximum number of steps of random walks is 3. The kernel size is 3 for all TCN layers. We use a step learning rate with $\gamma = 0.5$ for all experiments and adopt a grid search to find the best initial learning rate for each experiment, as shown in Table 4.

Table 4: Learning rate setting

| **Data** | **Model** | **Initial lr\start step\step size** |
|---|---|---|
| PEMS-BAY | GT-NoCausIn | 0.002\50\20 |
| | GT-CausIn | 0.006\50\20 |
| | GT-BadCausIn | 0.007\50\20 |
| METR-LA | GT-NoCausIn | 0.008\180\50 |
| | GT-CausIn | 0.004\180\50 |
| | GT-BadCausIn | 0.004\180\50 |
| | GT-CausIn($L = 3$) | 0.005\180\50 |
| | GT-CausIn($L = 5$) | 0.006\180\50 |
| | GT-CausIn($L = 6$) | 0.004\180\50 |