Spatio-Temporal Graph Neural Network with Hidden Confounders for Causal Forecast

Xinxin Luo[1], Wei Yin[2,*] and Zhuang Li[3]

[1]School of Cyber Science and Engineering, Southeast University, No.2, Southeast University Road, Jiangning District, Nanjing, Jiangsu Province, 21189, China

[3]School of Economics and Management, School of Cyber Science and Engineering, Southeast University, No.2, Southeast University Road, Jiangning District, Nanjing, Jiangsu Province, 21189, China

[3]School of Foreign Languages, Southeast University, No.2, Southeast University Road, Jiangning District, Nanjing, Jiangsu Province, 21189, China

Abstract

There are various unknown association patterns are involved in the dynamic spatiotemporal dependencies of multivariate time series forecasting, making it a challenging task. However, current mainstream time series prediction models often neglect potential causal relationships. Due to the presence of hidden confounders, these models inadvertently learn spurious relationships. Overlooking potential causal relationships and spurious relationships may lead to limited generalization capabilities when handling out-of-distribution data. To address this challenge, we draw inspiration from the field of causal inference and incorporate a random intervention mechanism. We propose the Causal Intervention Spatiotemporal Graph Neural Network(CISTGNN), which offers a causal perspective on time series forecasting. Our approach involves randomly sampling and recombining variant patterns across different periods to create an intervention distribution, thereby eliminating the misleading effects of hidden confounders. A series of experiments test on a real traffic flow dataset to validate the effectiveness of the proposed method. Compared to the baseline model, our model improves by 1.37 percent on average.

Keywords

Hidden confounders, Causal inference, Graph neural networks, Spatio-temporal dependencies, Causal forecasting

1. Introduction

Multivariate time series data, found in various domains like cloud computing, traffic, energy, finance, and social networks [1], is essential for understanding historical trends and making predictions based on past observations. However, effectively capturing the complex interdependencies between variables and dynamic patterns in such data presents a significant challenge. Traditional analytical tools, including Support Vector Regression (SVR) [2], Gradient Boosting Decision Trees (GBDT) [3], Vector Auto-Regression (VAR) [4], and Auto-Regressive Integrated Moving Average (ARIMA) [5], often struggle to handle the intricacies of these time series relationships and lead to less accurate predictions [6, 7]. With the advent of deep learning technologies, various neural networks such as Convolutional Neural Networks (CNN) [8], Recurrent Neural Networks (RNN) [9, 10], and Transformers [11, 12] have shown significant promise in modeling real-world time series data. However, a critical limitation of these methods is their inability to explicitly account for spatial relationships among time series in a non-Euclidean space [13], restricting their applicability [14]. Recently, there has been a growing interest in Spatial-Temporal Graph Neural Networks (STGNN) [15, 16, 17, 18, 19] for modeling multivariate time series data. These STGNNs demonstrate a robust capability to handle graph-structured data by treating variables as nodes within multivariate time series. This development opens up exciting new possibilities in the field [18].

However, most time series forecasting methods tend to prioritize spatiotemporal correlations within sequences while neglecting the underlying physical principles and causal relationships between these

ICCBR AI Track'24: Special Track on AI for Socio-Ecological Welfare at ICCBR2024, July 1, 2024, Mérida, Mexico

*Corresponding author.

230219085@seu.edu.cn (X. Luo); yinwei_seu@seu.edu.cn (W. Yin); 230218888@seu.edu.cn (Z. Li)

https://github.com/xinxinluo123/ (X. Luo); https://em.seu.edu.cn/yw_36723/list.htm (W. Yin)

0009-0001-2345-3841 (X. Luo); 0000-0003-0432-4355 (W. Yin); 0009-0005-6714-5356 (Z. Li)

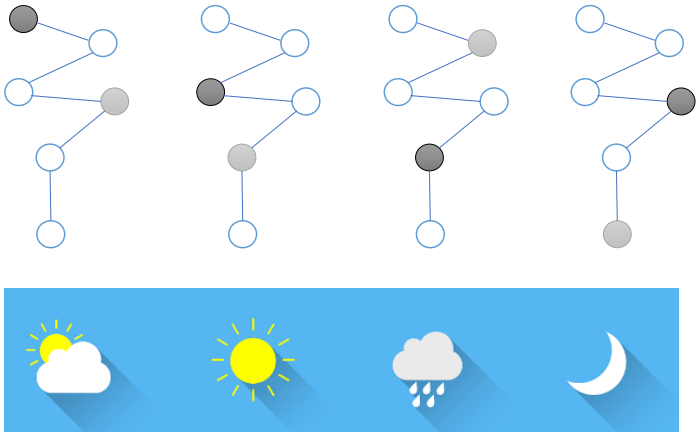

Figure 1: Hidden confounders are unobserved factors that can affect traffic flow, such as weather. In different weather conditions, traffic flow may vary significantly in certain locations. In the graph, darker nodes correspond to higher traffic flow. In this example, weather factors serve as hidden confounders affecting the model's generalization.

models [19, 20, 21]. When external conditions come into play, spatiotemporal correlations can become unstable, potentially leading to spurious correlations in observed outcomes. As we delve into the mechanisms behind the generation of observational data, causality becomes critical. Table 1 compares the researchers' different solutions for hidden confounders. For instance, some researchers [22, 23] have suggested that there exists a certain level of correlation between taxi and bicycle flows, which can be mutually beneficial for multitask learning. Under normal weather conditions, a correlation is observed between taxi and bicycle flows since during peak commuting hours, people's travel patterns align, resulting in similar trends for both modes of transportation. However, in adverse weather conditions, bicycle demand decreases due to unfavorable weather conditions, while taxi demand increases, leading to opposite trends for both during the same period. This demonstrates that weather conditions act as hidden confounders influencing the observed correlation between taxi and bicycle flows. In this context, weather factors serve as a prime example of how hidden confounding variables can significantly impact the model's predictive performance and robustness. Similarly, in various domains, there may exist hidden confounders that exert an influence on the predictive performance and robustness of models.

Table 1
Comparision of various approach for handling hidden confounders.

Approach	Assumption	Domian	Key Functionality
CAPE [24]	Hidden confounders	Event Forecasting	Individual Treatment Effect (ITE) Estimation
CCHMM [23]	Hidden confounders	Multimodal Traffic Prediction	Causal Conditional Hidden Markov Model
CaST [25]	Hidden confounders	Traffic Flow Prediction	Back-door adjustment & Front-door adjustment
STNSCM [26]	Hidden confounders	Bike Flow Prediction	Frontdoor Criterion & Counterfactual
CausalGNN [27]	Hidden confounders	Epidemic Forecasting	Causal module & Attention-based dynamic GNN
DCCF [28]	Hidden confounders	Recommender Systems	Front-door adjustment

To address this challenge, we conduct a comprehensive causal analysis of the prediction process of the spatiotemporal graph neural network. This analysis gives us a deep understanding of the relationships among various elements, including inputs $\mathbf{X}$, hidden confounders $\mathbf{HC}$, causal features $\mathbf{C}$, shortcut features $\mathbf{S}$, hidden layers $\mathbf{H}$, and predictions $\mathbf{Y}$, as shown in Fig 2. Firstly, hidden confounders are unobserved factors that can affect traffic flow, such as weather, holidays, and unexpected events, as shown in Fig 1. Secondly, shortcut features are features that are correlated with the prediction target but not causally related. These could include weather, holidays, and unforeseen events, which may form hidden confounders and give rise to shortcut features. Lastly, causal features are features that reflect causal relationships in traffic flow, such as traffic signals, traffic rules, traffic events, and traffic

demand. These features help machine learning models to understand and predict changes in traffic flow more accurately, thereby enhancing the interpretability and reliability of predictions. However, these shortcut features can unintentionally create a backdoor path, leading to spurious correlations between causal features and predictions. Therefore, our strategy to address this issue focuses on mitigating the effects of hidden confounders. We aim to enhance the model's generalization capability by effectively leveraging the potential of causal features while filtering out shortcut features.

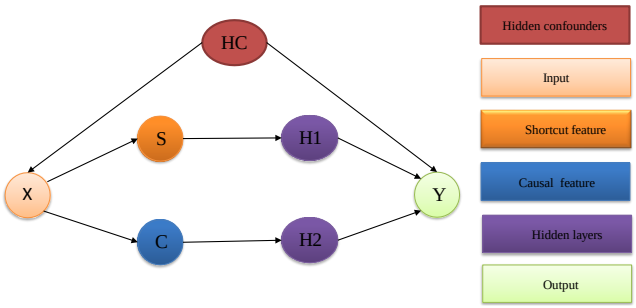

Figure 2: Structural causal graph for spatiotemporal graph neural networks. Causal features are features that reflect causal relationships in traffic flow, while shortcut features are features that are correlated with the prediction target but not causally related.

In response to these challenges, we propose a novel CISTGNN known as the Dynamic Graph Attention Network Based on Causal Intervention. Our approach effectively addresses out-of-distribution by identifying and harnessing stable spatiotemporal patterns with reliable predictive capabilities. To achieve this, we introduce a disentangled spatial attention network that captures both variant and invariant patterns within dynamic graphs. This network empowers each node to focus on its historical neighbors through a disentangled attention information propagation mechanism. Drawing inspiration from the field of causal inference, we incorporate a random intervention mechanism. This innovative approach involves sampling and recombining variant patterns across different periods to create an intervention distribution, thereby eliminating the misleading effects of variable patterns. Consequently, our model becomes capable of capturing and leveraging stable spatiotemporal patterns that offer dependable predictive performance, even when dealing with out-of-distribution. Our contribution can be summarized as follows:

- Firstly, we analyze the physical mechanisms behind data generation. This analysis forms the foundation for constructing a causal graph, which explicitly describes the causal relationships among various factors in the data and identifies the impact of hidden confounders.
- Secondly, we propose a spatiotemporal graph neural network prediction model based on the principle of causal intervention. In our model, we use the backdoor criterion to effectively mitigate the influence of hidden confounders.
- Finally, the reliability and validity of our model are validated by extensive experiments on real-world datasets. Compared with the baseline model, our model improves by 1.37% on average.

2. Related work

Spatiotemporal Graph Neural Networks are a type of graph neural network used for processing spatiotemporal data, capturing dependencies in both spatial and temporal dimensions [29]. They find applications in multivariate time series forecasting, such as traffic flow prediction and energy consumption forecasting. The approach involves representing multivariate time series data as a graph, where nodes represent variables and edges denote relationships. Spatiotemporal features are then extracted using graph convolution and temporal convolution operations for prediction [13]. Examples include STGCN [17] and MTGNN [18] for handling non-Euclidean spaces and automatically learning spatial dependencies. DCRNN [16] models traffic flow as a diffusion process on a directed graph, while

ASTGNN [19] utilizes attention mechanisms to extract features. Recently, DSTGN [6] was introduced, which can extract static and dynamic graph matrices to model long-term and short-term patterns separately.

Disentangled Representation Learning is a machine learning approach aimed at acquiring representations from data that can separate different factors or features [30]. Disentangled Representation Learning is applied to attention mechanisms, mainly involving encoding vectors for queries, keys, and values separately as content and positional vectors, thereby achieving disentangled representations of content and position [31]. There are various methods for applying disentangled representation learning to attention mechanisms, such as DeBERTa [30], Disenhan [32], and DisenKGAT [33]. They each use different approaches to construct and compute disentangled matrices to achieve disentangled attention for content and position.

Causal Inference plays a crucial role in shaping the design of machine learning algorithms, providing essential guidance for their development. As artificial intelligence continues to advance, an increasing number of researchers are recognizing the pivotal role of causal inference in addressing the limitations of existing AI methods, particularly in areas like abstraction, reasoning, and interpretability. In his book *The Book of Why*, Turing Award winner Judea Pearl categorizes causal inference into three levels: the first level is "association"; the second level is "intervention"; and the third level is "counterfactual inference" [34]. Hidden confounders represent potential influences on causal inference [35]. These factors are variables that exhibit correlations with both the independent and dependent variables but do not lie on the causal path. The presence of hidden confounders can result in spurious correlations or biases between independent and dependent variables, consequently interfering with or obscuring causal effects. To mitigate the impact of hidden confounders, it becomes imperative to employ methods aimed at identifying and controlling these hidden confounders.

3. Preliminaries

This paper focuses on the prediction of multivariate time series and leverages a graph-based structure to capture the relationships among various variables. In this approach, individual variables in the data are treated as nodes in a graph, and the observations associated with each node are interpreted as either the node's features or graph signals, which is an intuitive and natural methodology. To represent the connections between these nodes, a graph adjacency matrix is employed.

Definition 1 (Graph G): We represent the relationships among multivariate variables using the notation $G = (V, E, A)$. In this context, the graph consists of a set of vertices (or nodes) $V \in \mathbb{R}^N$ and a set of edges E, where each edge connects two vertices, indicating the presence of a specific relationship between them. The matrix $A \in \mathbb{R}^{N \times N}$ describes the connections between these nodes. The adjacency matrix has dimensions $N \times N$, where N represents the number of vertices in the graph.

The objective of predicting time-varying graph signals is to learn a mapping function that can estimate the future features or properties of nodes within the graph based on their historical attributes. This process entails leveraging the historical attributes of the nodes over the past T_p time steps to make predictions about the features or attributes of the nodes for the future T_f time steps.

$$\left(\mathbf{X}_{t-T_p+1}, \mathbf{X}_{t-T_p+2}, \ldots, \mathbf{X}_t\right) \xrightarrow{f} \left(\mathbf{X}_{t+1}, \mathbf{X}_{t+2}, \ldots, \mathbf{X}_{t+T_f}\right)$$

4. Method

In this section, we will begin by presenting the comprehensive flow of our model, as shown in Fig 3. We will commence with an analysis of causal inference as applied to the learning process of graph neural networks. We will involve introducing the relevant Structural Causal Models (SCM) and discussing the concept of backdoor adjustment. Given our assumptions, we think that hidden confounders play a pivotal role in influencing the model's generalization. Consequently, we propose a novel strategy to mitigate the adverse effects of these hidden confounders. The overarching model comprises four

key components: a causal feature learning module, a K-layer spatiotemporal convolution module, a causal intervention module, and a predictive output module. The interaction between each module involves several steps. First, we disentangle the features into causal and non-causal features using the causal feature module. Next, the model learns causal features and non-causal features separately in K-layer spatiotemporal features. Then, we perform random interventions on non-causal features and fuse causal features. Finally, we make the final prediction using the prediction module.

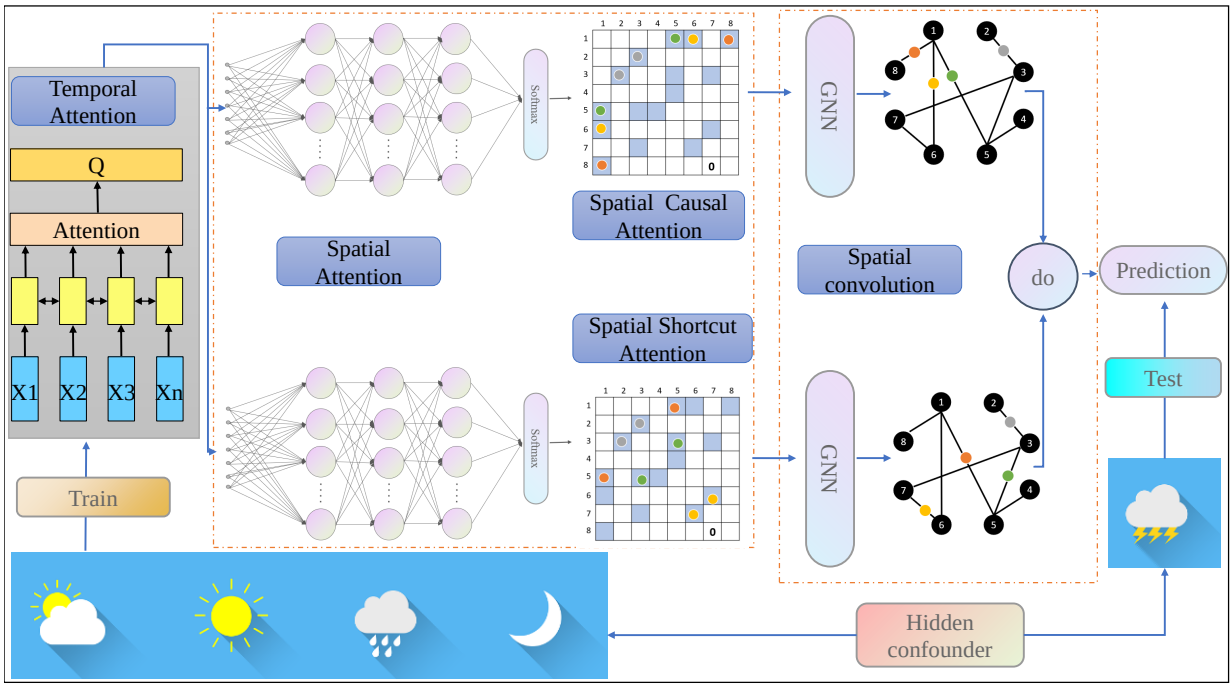

Figure 3: The overall model framework of CISTGNN. Hidden confounders can precipitate out-of-distribution generalization issues. Our model is composed of a temporal attention module, a spatial attention module, a causal intervention module, and a forecasting module. We disentangled the spatial attention module into two distinct components: Spatial Causal Attention and Spatial Shortcut Attention. In the end, we alleviate the bias induced by hidden confounders through the causal intervention.

4.1. Strutural causal model

We first analyze the underlying mechanisms of data generation from the perspective of structural causal models, which can provide better interpretability for the model. We propose a novel framework, CISTGNN, which incorporates causal reasoning into the prediction of future time series in a spatio-temporal environment. We use a structural causal model (SCM) to describe the causal relationship between input $\mathbf{X}$, hidden confounders $\mathbf{HC}$, causal feature $\mathbf{C}$, shortcut features $\mathbf{S}$, spatiotemporal state $\mathbf{H}$, and prediction target $\mathbf{Y}$, as shown in Fig 2, where directed edges denote the causal relationships among nodes. We assume that causal feature $\mathbf{C}$ and shortcut features $\mathbf{S}$ can be disentangled from the spatio-temporal data $\mathbf{X}$ and integrate them to form the spatio-temporal state $\mathbf{H}$, thus describing the dynamic spatio-temporal patterns in the data.

4.2. Causal intervention via backdoor criterion

The contextual condition $\mathbf{HC}$ is defined as the common cause of both $\mathbf{X}$ and $\mathbf{H}$. This situation can lead to $\mathbf{H}$ being biased towards the general state while potentially disregarding specific environmental factors due to dataset limitations, resulting in an unfair bias of $\mathbf{H}$. It is evident from Fig 2 that certain backdoor paths exist, including $\mathbf{HC} \rightarrow \mathbf{X} \rightarrow \mathbf{C}$ and $\mathbf{HC} \rightarrow \mathbf{X} \rightarrow \mathbf{S}$. Breaking the link $\mathbf{HC} \rightarrow \mathbf{X} \rightarrow \mathbf{C}$

enables $\mathbf{X}$ to fairly incorporate each contextual condition $\mathbf{C}$ into the spatiotemporal state $\mathbf{H}$. To achieve graph representation learning, we ought to eliminate these backdoor paths. Fortunately, causal theory [35, 34] offers a practical solution: we can apply do-calculus to variable $\mathbf{C}$ to eliminate these backdoor paths by estimating $\mathbf{Pm}(\mathbf{H}|\mathbf{C}) = \mathbf{P}(\mathbf{H}|\text{do}(\mathbf{C}))$.

$$\begin{aligned} P(H \mid do(C)) &= P_m(H \mid C) \\ &= \sum_{s \in \mathcal{HC}} P_m(H \mid C, s) P_m(s \mid C) \\ &= \sum_{s \in \mathcal{HC}} P_m(H \mid C, s) P_m(s) \\ &= \sum_{s \in \mathcal{HC}} P(H \mid C, s) P(s), \end{aligned} \tag{1}$$

From the analysis of Formula (1), it is known that we can mitigate the bias brought by hidden confounders through the backdoor criterion. In our paper, our approach is to perform random interventions on non-causal features and fuse causal features.

4.3. Disentangled dynamic Spatial attention

To better learn causal features, we disentangle spatial features into invariant patterns and variant patterns. In the spatial dimension, different nodes interact with each other, and this interaction is highly dynamic. To capture this dynamism, this paper employs an attention mechanism that can adaptively capture spatial causal relationships. We use the following attention mechanism[19]:

$$\mathbf{S} = \mathbf{B}_s \cdot \sigma \left(\left(\mathcal{X}_h^{(l-1)} \mathbf{A}_1 \right) \mathbf{A}_2 \left(\mathbf{A}_3 \mathcal{X}_h^{(l-1)} \right)^T + \mathbf{b}_s \right) \tag{2}$$

In this context, $\mathcal{X}_h^{(l-1)}$ represents the input data of the l-th layer, which is a three-dimensional tensor characterized by dimensions $N \times C_{l-1} \times T_{l-1}$. Here, N signifies the number of nodes, C_{l-1} represents the number of channels, and T_{l-1} denotes the time length. B_s stands for the parameter matrix for spatial attention, which is a two-dimensional matrix with dimensions $N \times N$. The bias for the spatial attention mechanism denoted as b_s, is a one-dimensional vector with dimensions N. Parameter A_1 is a one-dimensional vector with dimensions T_{l-1}. Additionally, A_2 represents the parameter matrix for spatial attention with dimensions $C_{l-1} \times T_{l-1}$, and A_3 is a one-dimensional vector with dimensions C_{l-1}. To enhance the expressive power of the model, we apply the activation function σ, which is a non-linear function. The attention matrix S, is a two-dimensional matrix sized $N \times N$. Each element $\mathbf{S}_{\text{i,j}}$ of this matrix represents the relevance strength between node i and node j.

$$\mathbf{P_I} = softmax\left(\mathbf{S}_{\text{i,j}}\right), \mathbf{P_v} = softmax\left(-\mathbf{S}_{\text{i,j}}\right) \tag{3}$$

In the context of our model, we denote the masks for invariant and variant patterns as $\mathbf{P_I}$ and $\mathbf{P_v}$ respectively [36]. Notably, there exists a negative correlation between these two patterns. This negative correlation arises from the observation that dynamic neighbors with higher attention scores in one pattern tend to have lower attention scores in the other pattern. This intriguing relationship suggests that, in different patterns, the model's focus on dynamic neighbors can vary. This variation, in turn, enhances the model's ability to effectively capture specific structures and changes in the data.

4.4. Temporal attention

In the temporal dimension, traffic conditions exhibit correlations between different time steps, and the nature of this correlation can change in response to varying situations. To effectively address this dynamic nature of temporal data, we employ an attention mechanism that dynamically assigns varying

levels of importance to different time steps. This approach enables the model to adapt its focus based on specific situations, thereby enhancing its ability to capture the evolving correlations and variations in the data.

$$\mathbf{Q} = \mathbf{B}_q \cdot \sigma \left(\left(\left(\mathcal{X}_h^{(l-1)} \right)^T \mathbf{M}_1 \right) \mathbf{M}_2 \left(\mathbf{M}_3 \mathcal{X}_h^{(r-1)} \right) + \mathbf{b}_q \right) \tag{4}$$

$$\mathbf{Q}'_{i,j} = softmax\left(\mathbf{Q}_{\mathrm{i,j}}\right) \tag{5}$$

within our model, we employ learnable parameters denoted as $\mathbf{B}_q, \mathbf{b}_q \in \mathbb{R}^{T_{l-1} \times T_{l-1}}, \mathbf{M}_1 \in \mathbb{R}^N, \mathbf{M}_2 \in \mathbb{R}^{C_{l-1} \times N}$, and $\mathbf{M}_3 \in \mathbb{R}^{C_{l-1}}$. These parameters are essential for computing the elements of the spatio-temporal attention matrix, denoted as Q. This matrix, with dimensions $T_{l-1} \times T_{l-1}$, serves as a representation of the correlations between different time steps in the input data. The value of each element $\mathbf{Q}_{i,j}$ in this matrix signifies the strength of dependency between the i-th and j-th time steps.

4.5. Spatail temporal convolution

The spatiotemporal convolution module utilizes graph convolution to model spatial structures and standard convolution to simulate temporal dynamics. The synergy of these modules equips the neural network to handle spatiotemporal data with precision, significantly improving its modeling capabilities for complex tasks.

Graph convolution extends convolution operations to graph structures, capturing spatial features in graph data, useful for tasks like traffic flow prediction. It's implemented through spectral and spatial methods. Spectral methods use Laplacian matrices but have high computational complexity. Spatial methods aggregate features with weighted sums, proving more efficient and adaptable to dynamic graphs. The Laplacian matrix, derived from adjacency and degree matrices, is crucial for structural characterization. Eigenvalue decomposition of it yields matrices $\mathbf{\Lambda}$ and U, providing spectral properties insights. Graph Fourier transform, facilitated by U, shifts signals from spatial to frequency domains, enabling filtering. Graph convolution employs a kernel function g_θ and U, involving a sequence of graph Fourier transform, filtering, and inverse transform, forming a comprehensive framework for graph signal processing.

Firstly, we can represent the Laplacian matrix and its eigenvalue decomposition as an equation:

$$\mathbf{L} = \mathbf{D} - \mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \tag{6}$$

where $\mathbf{A}$ is the adjacent matrix, $\mathbf{I}_N$ is a unit matrix, and the degree matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$ is a diagonal matrix, consisting of node degrees, $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$.

Secondly, we can represent the graph Fourier transform and inverse graph Fourier transform as an equation:

$$\hat{x} = \mathbf{U}^T x, x = \mathbf{U}\hat{x} \tag{7}$$

Graph convolution is a method used to extract spatial features and dependencies in signals on graphs. The core idea is to define a kernel function g_θ using the Laplacian matrix and its eigenvalues, and then apply this kernel function to filter the signal x on the graph. The graph convolution operation can be represented by the following formula:

$$g_\theta *_G x = g_\theta(\mathbf{L})x = g_\theta\left(\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T\right)x = \mathbf{U}g_\theta(\mathbf{\Lambda})\mathbf{U}^T x \tag{8}$$

in graph convolution, $*G$ signifies the operation transforming signals on a graph. Initially, $g\theta$ and x undergo graph Fourier transforms to the frequency domain. These are then multiplied, and the convolution result is obtained via inverse graph Fourier transform. The matrix U facilitates the forward transform, while $\mathbf{U}^T$ enables the inverse, translating signals between spatial and frequency domains.

Combined, these steps form the graph convolution process, elucidating its foundational concepts and procedural flow.

Nevertheless, in the context of large-scale graphs, a direct eigenvalue decomposition of the Laplacian matrix can be exceedingly time-consuming. To mitigate this challenge, our paper adopts a more efficient approach based on the utilization of Chebyshev polynomials to approximate the solution[37].

$$g_\theta *_G x = g_\theta(\mathbf{L})x = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\mathbf{L}})x \tag{9}$$

The recursive definition of Chebyshev polynomials is integral to our approach. $\tilde{\mathbf{L}} = \frac{2}{\lambda_{\max}}\mathbf{L} - \mathbf{I}_N, \lambda_{\max}$ is the maximum eigenvalue of the Laplacian matrix.This definition is as follows: $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$, with initial values $T_0(x) = 1, T_1(x) = x$. This recursive definition is pivotal in our method as it effectively extracts information from up to $\mathrm{K}-1$ neighboring nodes centered around each focal node and applies the convolution kernel g_θ. To finalize the graph convolution process, we employ the Rectified Linear Unit (ReLU) as the activation function, denoted as $\mathrm{ReLU}\left(g_\theta *_G x\right)$. This combination of steps allows efficient and scalable processing of large-scale graphs, effectively approximating the convolution operation of the Laplacian matrix using Chebyshev polynomials.

To dynamically adapt and fine-tune the correlations between nodes, our approach introduces a novel element. For each term of the Chebyshev polynomial, we perform a multiplication operation by taking $T_k(\widetilde{\mathbf{L}})$ and multiplying it by spatial attention matrix $\mathbf{P_I} \in \mathbb{R}^{N\times N}$ and $\mathbf{P_V} \in \mathbb{R}^{N\times N}$. This operation is symbolized as $T_k(\tilde{\mathbf{L}}) \odot \mathbf{P_I}$ and $T_k(\tilde{\mathbf{L}}) \odot \mathbf{P_V}$, where the symbol $\odot$ represents the Hadamard product. Consequently, the graph convolution formula mentioned earlier can be succinctly represented as follows:

$$g_\theta *_G x = g_\theta(\mathbf{L})x = \sum_{k=0}^{K-1} \theta_k \left(T_k(\tilde{\mathbf{L}}) \odot \mathbf{P_I}\right) x \tag{10}$$

$$g_\theta *_G x = g_\theta(\mathbf{L})x = \sum_{k=0}^{K-1} \theta_k \left(T_k(\tilde{\mathbf{L}}) \odot \mathbf{P_v}\right) x \tag{11}$$

This definition can be extended to accommodate graph signals with multiple channels. For instance, in recent developments, the input is represented as $\mathcal{X}_h^{(l-1)}$, where each node's features encompass C_{l-1} channels. At each time step t, C_l filters are applied to the graph X^t, yielding $g_\theta *_G x$, wher $\Theta = (\Theta_1, \Theta_2, \ldots, \Theta_{C_l}) \in \mathbb{R}^{K\times C_{l-1}\times C_l}$ represents the convolutional kernel parameters[38]. Consequently, each node is updated by leveraging information from its 0 to $\mathrm{K}-1$ order neighbors.

After conducting a graph convolution operation to capture the neighborhood information of each node in the spatial dimension, then we introduce a standard convolutional layer in the temporal dimension. The primary purpose of this convolutional layer is to update the node signals by effectively integrating information from adjacent time slices.

$$\mathcal{X}_h^{(l)} = \mathrm{Re}\,LU\left(\Phi * \left(\mathrm{Re}\,LU\left(g_\theta *_G \hat{\mathcal{X}}_h^{(l-1)}\right)\right)\right) \tag{12}$$

In this formula, the parameter $\mathcal{X}_h^{(l-1)}$ denotes the output of the l-th layer within the recent component. θ represents the parameters of the time-dimension convolution kernel, g_θ signifies the graph convolution kernel function, $*_G$ denotes the graph convolution operation, and $\mathrm{Re}\,LU$ stands for the rectified linear unit activation function. This formula can be broken down into the following sequential steps:

We introduce a spatial-temporal convolution module that relies on spatial-temporal convolution operations and spatial-temporal attention mechanisms. The purpose is to extract both spatial and temporal features from traffic data. To achieve this, we stack multiple spatial-temporal convolution modules and spatial-temporal attention modules within a spatial-temporal block. This strategy allows us to comprehensively capture a broad spectrum of dynamic spatial-temporal correlations. To ensure that the output of each component aligns with the prediction target, we have incorporated a fully connected layer after each component. Additionally, we utilize the rectified linear unit (ReLU) as the activation function.

5. Experiment

5.1. Dateset

We propose a novel spatiotemporal graph neural network called CISTGNN, which excels in handling spatiotemporal data. We evaluate its performance through a multi-step prediction task, aiming to predict data for multiple future time steps. We conduct Experiments on four publicly available traffic datasets that already include graph structures. On datasets with graph structures, we compare CISTGNN with other spatiotemporal graph neural networks that use predefined graphs.

5.2. Experimental settings

We evaluate the performance of our proposed CISTGNN model using four traffic datasets, PeMSD3, PeMSD4, PeMSD7, and PeMSD8, both of which come equipped with graph structures. We divide the dataset into training, validation, and test sets in a 6:2:2 ratio, maintaining the chronological order. In the multi-step prediction task, we leverage data from the preceding 12 time steps to forecast data for the subsequent 12-time steps, effectively predicting traffic flow for the next hour based on the previous hour's data. Our CISTGNN model is implemented using the PyTorch framework, and the experiments are conducted on a machine featuring an Intel(R) Xeon(R) Silver 4310 2.10GHz 12-core CPU and an NVIDIA GeForce GTX 4090 with 24 GB of GPU memory. The source code for our CISTGNN can be found at https://github.com/xinxinluo123/CISTGNN. During model training, we employ the Adam optimizer with a learning rate of 0.001. The model's multi-step prediction task consists of Stacking three spatio-temporal layers. The first layer of the output module consists of 512 output channels, while the second layer has 12 output channels. Our training process set 80 epochs, with a node embedding dimension set at 10. The batch size remains configured at 32. Additionally, we leverage Chebyshev polynomials of order 2, utilizing a total of 64 Chebyshev filters and 64 temporal filters.

5.3. Baseline methods and metrics

To evaluate the performance of the multi-step prediction task, we employ three evaluation metrics: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE). We denote the real signal as Y and the predicted signal as $\hat{Y}$. Both are represented as matrices with dimensions $N \times T$, where N signifies the number of nodes, T signifies the number of time steps, and ρ indicates the test set. These metrics provide valuable insights into the accuracy and robustness of our predictions.

We assess the performance of our model through a multi-step prediction task and conduct a comparative analysis against nine baseline models: FC-LSTM [39], TCN [40], VAR [4], SVR [41], DCRNN [16], STGCN [17], ASTGCN [19], Graph WaveNet [42], STSGCN [43].

Table 2
Baseline comparison on traffic datasets for spatial–temporal GNN.

Dataset	Metrics	FC-LSTM	TCN	VAR	SVR	DCRNN	STGCN	ASTGCN	Graph WaveNet	STSGCN	CISTGNN
PeMSD3	MAE	21.33	19.33	19.72	19.77	19.56 ± 0.32	17.62 ± 0.13	18.67 ± 0.42	19.85 ± 0.03	17.48 ± 0.15	$\mathbf{17.03 \pm 0.22}$
	RMSE	35.11	33.24	32.38	32.78	29.86 ± 0.47	33.87 ± 1.18	30.71 ± 1.02	32.94 ± 0.18	29.21 ± 0.56	$\mathbf{28.99 \pm 0.14}$
	MAPE (%)	22.33	19.86	20.50	23.04	16.83 ± 0.13	17.33 ± 0.94	19.85 ± 1.06	19.31 ± 0.49	16.78 ± 0.20	$\mathbf{16.49 \pm 0.60}$
PeMSD4	MAE	26.2	23.11	24.44	26.18	24.42 ± 0.06	23.90 ± 0.17	22.90 ± 0.20	25.54 ± 0.03	21.19 ± 0.10	$\mathbf{20.88 \pm 0.10}$
	RMSE	40.49	37.25	37.76	38.91	37.48 ± 0.10	36.43 ± 0.22	35.59 ± 0.35	39.70 ± 0.04	$\mathbf{33.65 \pm 0.20}$	33.77 ± 0.07
	MAPE (%)	19.30	15.48	17.27	22.84	16.86 ± 0.09	13.67 ± 0.14	16.75 ± 0.59	17.29 ± 0.24	13.90 ± 0.05	$\mathbf{13.63 \pm 0.02}$
PeMSD7	MAE	29.96	32.68	27.96	28.45	24.45 ± 0.85	26.22 ± 0.37	28.13 ± 0.70	26.85 ± 0.05	24.26 ± 0.14	$\mathbf{24.03 \pm 0.04}$
	RMSE	43.94	42.23	41.31	42.67	37.61 ± 1.18	39.18 ± 0.42	43.67 ± 1.33	42.78 ± 0.07	39.23 ± 0.27	$\mathbf{38.41 \pm 0.74}$
	MAPE (%)	14.34	14.22	12.11	14.00	10.67 ± 0.53	10.74 ± 0.16	13.31 ± 0.55	12.12 ± 0.41	10.21 ± 0.05	$\mathbf{10.05 \pm 0.07}$
PeMSD8	MAE	22.20	22.69	19.83	20.92	18.49 ± 0.16	18.79 ± 0.49	18.72 ± 0.16	19.13 ± 0.08	17.13 ± 0.09	$\mathbf{17.07 \pm 0.01}$
	RMSE	33.06	35.79	29.24	31.23	27.30 ± 0.22	28.23 ± 0.36	28.99 ± 0.11	31.05 ± 0.07	$\mathbf{26.80 \pm 0.18}$	27.24 ± 0.03
	MAPE (%)	15.02	14.04	13.08	14.24	11.69 ± 0.06	10.55 ± 0.30	12.53 ± 0.48	12.68 ± 0.57	10.96 ± 0.07	$\mathbf{10.86 \pm 0.02}$

5.4. Comparison and analysis of prediction results

We utilize the CISTGNN model for the multi-step prediction task. In the context of multi-step prediction, Table 2 provides an overall performance comparison between our CISTGNN model and 9 representative comparison methods. This comparison is based on average MAE, RMSE, and MAPE across 12 prediction time steps, leading to the following observations:

Table 2 shows the comparison of the prediction performance of CISTGNN with the 9 benchmark methods. We observe that (1) time series prediction models, including traditional methods (i.e., VAR), and machine learning-based methods (i.e., SVR) because they only consider temporal features but not spatial correlation, which is equally important for spatio-temporal traffic prediction. Therefore, they have the worst prediction performance (2) Spatiotemporal graph neural networks generally perform better because they use graph neural networks to further model spatial correlation. DCRNN is a typical RNN-based method for spatiotemporal graph data prediction, and STGCN, ASTGCN, Graph WaveNet, and STSGCN are four typical CNN-based methods that only focus on the correlation of spatio-temporal data and ignore the causality of spatio-temporal data.

We conduct experiments on the PEMS03 dataset. The results show that our method improves the MAE, RMSE, and MAPE(%) by 0.82, 0.89, and 1.16 respectively, compared to the baseline method [19]. Similarly, on the PEMS07 dataset, our method improves the MAE, RMSE, and MAPE(%) by 1.55, 1.24, and 1.27 respectively. Also, on the PEMS04 dataset, our method improves MAE, RMSE, and MAPE(%) by 1.54, 1.58, and 2.24 respectively, and on the PEMS08 dataset, our method improves MAE, RMSE, and MAPE(%) by 1.26, 1.31 and 1.64 respectively. Our model outperforms ASTGCN with a 2% improvement in MAE on both the PeMSD4 and PeMSD8 datasets.

5.5. Ablation study

Our CISTGNN model comprises several key components, all of which contribute to the model's performance. To validate their effectiveness, we conducted ablation experiments on the PeMSD4 and PeMSD8 datasets by removing the following components:

- CISTGNN w/o causal intervention: Causal intervention has been removed, and spatiotemporal attention is now directly transferred to the graph convolutional neural network, focusing solely on the dynamic attention mechanism without causal intervention.
- CISTGNN w/o causal attention: Eliminate the causal attention module and replace it with direct causal intervention on non-causal data.
- CISTGNN w/o causal intervention + causal attention: The causal intervention and causal attention models in CISTGNN have been removed, allowing the model to concentrate solely on relevance learning.

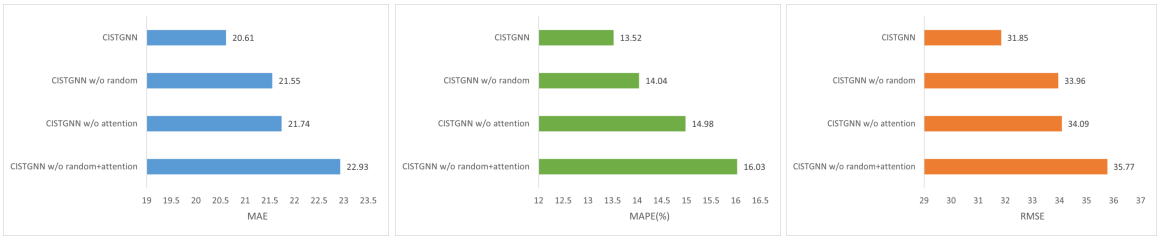

Figure 4: Results of CISTGNN in the ablation tests on the PeMSD4.

We evaluate the test results using three metrics: RMSE, MAPE, and MAE. The specific results are shown in Fig 4 and Fig 5. From these results, we can make the following observations:

- The CISTGNN model outperforms all datasets, confirming the effectiveness of each component.

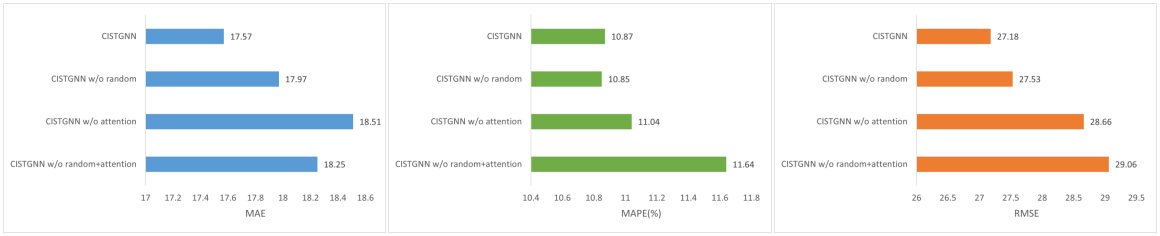

Figure 5: Results of CISTGNN in the ablation tests on the PeMSD8.

- Removing causal intervention leads to a performance decrease, particularly in terms of RMSE on the PeMSD4 dataset and MAE on the PeMSD8 dataset.
- Removing causal attention leads to performance degradation, especially in PeMSD8, which emphasizes the importance of establishing causal attention when establishing dynamic dependencies.

6. Conclusion

In our study, we rethink the utilization of spatio-temporal graph neural networks for multivariate time-series prediction, with a specific emphasis on the causal perspective. We've discovered that existing spatio-temporal graph neural network learning approaches frequently depend on shortcut features to support their predictions. However, these shortcut features might inadvertently introduce confounders and create backdoor paths, leading to erroneous correlations in spatiotemporal graph neural network learning. To address this confounding effect, we introduce a causality-based attention-learning mechanism and a causal intervention mechanism guided by causality theory. We propose the Causal Intervention Spatiotemporal Graph Neural Network (CISTGNN), which offers a causal perspective on time series forecasting. CISTGNN is composed of two crucial components: the Causal Spatiotemporal Attention Module and the Causal Intervention Module. By distinguishing between causal relationships and spurious ones, we reduce the model's reliance on shortcut features and effectively leverage causal features. Experimental results validate the efficacy of this approach.

References

[1] B. Lim, S. Zohren, Time-series forecasting with deep learning: a survey, Philosophical Transactions of the Royal Society A 379 (2021) 20200209.

[2] C.-J. Lu, T.-S. Lee, C.-C. Chiu, Financial time series forecasting using independent component analysis and support vector regression, Decision support systems 47 (2009) 115–125.

[3] E. Rady, H. Fawzy, A. M. A. Fattah, Time series forecasting using tree based methods, J. Stat. Appl. Probab 10 (2021) 229–244.

[4] E. Zivot, J. Wang, Vector autoregressive models for multivariate time series, Modeling financial time series with S-PLUS® (2006) 385–429.

[5] S. Siami-Namini, N. Tavakoli, A. S. Namin, A comparison of arima and lstm in forecasting time series, in: 2018 17th IEEE international conference on machine learning and applications (ICMLA), IEEE, 2018, pp. 1394–1401.

[6] Z. Li, J. Yu, G. Zhang, L. Xu, Dynamic spatio-temporal graph network with adaptive propagation mechanism for multivariate time series forecasting, Expert Systems with Applications 216 (2023) 119374.

[7] M. Jin, Y. Zheng, Y.-F. Li, S. Chen, B. Yang, S. Pan, Multivariate time series forecasting with dynamic graph neural odes, IEEE Transactions on Knowledge and Data Engineering (2022).

[8] R. Wan, S. Mei, J. Wang, M. Liu, F. Yang, Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting, Electronics 8 (2019) 876.

[9] H. Hewamalage, C. Bergmeir, K. Bandara, Recurrent neural networks for time series forecasting: Current status and future directions, International Journal of Forecasting 37 (2021) 388–427.
[10] S. Siami-Namini, N. Tavakoli, A. S. Namin, The performance of lstm and bilstm in forecasting time series, in: 2019 IEEE International conference on big data (Big Data), IEEE, 2019, pp. 3285–3292.
[11] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, L. Sun, Transformers in time series: A survey, arXiv preprint arXiv:2202.07125 (2022).
[12] Z. Geng, J. Xu, R. Wu, C. Zhao, J. Wang, Y. Li, C. Zhang, Stgaformer: Spatial–temporal gated attention transformer based graph neural network for traffic flow forecasting, Information Fusion 105 (2024) 102228.
[13] G. Jin, Y. Liang, Y. Fang, J. Huang, J. Zhang, Y. Zheng, Spatio-temporal graph neural networks for predictive learning in urban computing: A survey, arXiv preprint arXiv:2303.14483 (2023).
[14] M. Jin, G. Shi, Y.-F. Li, Q. Wen, B. Xiong, T. Zhou, S. Pan, How expressive are spectral-temporal graph neural networks for time series forecasting?, arXiv preprint arXiv:2305.06587 (2023).
[15] M. Jin, H. Y. Koh, Q. Wen, D. Zambon, C. Alippi, G. I. Webb, I. King, S. Pan, A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection, arXiv preprint arXiv:2307.03759 (2023).
[16] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, arXiv preprint arXiv:1707.01926 (2017).
[17] B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting, arXiv preprint arXiv:1709.04875 (2017).
[18] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, C. Zhang, Connecting the dots: Multivariate time series forecasting with graph neural networks, in: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, 2020, pp. 753–763.
[19] S. Guo, Y. Lin, N. Feng, C. Song, H. Wan, Attention based spatial-temporal graph convolutional networks for traffic flow forecasting, in: Proceedings of the AAAI conference on artificial intelligence, volume 33, 2019, pp. 922–929.
[20] C. Zheng, X. Fan, C. Wang, J. Qi, Gman: A graph multi-attention network for traffic prediction, in: Proceedings of the AAAI conference on artificial intelligence, volume 34, 2020, pp. 1234–1241.
[21] R. Jiang, Z. Wang, J. Yong, P. Jeph, Q. Chen, Y. Kobayashi, X. Song, S. Fukushima, T. Suzumura, Spatio-temporal meta-graph learning for traffic forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, 2023, pp. 8078–8086.
[22] X. Deng, Z. Zhang, Comprehensive knowledge distillation with causal intervention, Advances in Neural Information Processing Systems 34 (2021) 22158–22170.
[23] Y. Zhao, P. Deng, J. Liu, X. Jia, M. Wang, Causal conditional hidden markov model for multimodal traffic prediction, arXiv preprint arXiv:2301.08249 (2023).
[24] S. Deng, H. Rangwala, Y. Ning, Robust event forecasting with spatiotemporal confounder learning, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 294–304.
[25] Y. Xia, Y. Liang, H. Wen, X. Liu, K. Wang, Z. Zhou, R. Zimmermann, Deciphering spatio-temporal graph forecasting: A causal lens and treatment, Advances in Neural Information Processing Systems 36 (2024).
[26] P. Deng, Y. Zhao, J. Liu, X. Jia, M. Wang, Spatio-temporal neural structural causal models for bike flow prediction, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, 2023, pp. 4242–4249.
[27] L. Wang, A. Adiga, J. Chen, A. Sadilek, S. Venkatramanan, M. Marathe, Causalgnn: Causal-based graph neural networks for spatio-temporal epidemic forecasting, in: Proceedings of the AAAI conference on artificial intelligence, volume 36, 2022, pp. 12191–12199.
[28] S. Xu, J. Tan, S. Heinecke, V. J. Li, Y. Zhang, Deconfounded causal collaborative filtering, ACM Transactions on Recommender Systems 1 (2023) 1–25.
[29] S. Wang, J. Cao, S. Y. Philip, Deep learning for spatio-temporal data mining: A survey, IEEE transactions on knowledge and data engineering 34 (2020) 3681–3700.
[30] P. He, X. Liu, J. Gao, W. Chen, Deberta: Decoding-enhanced bert with disentangled attention,

arXiv preprint arXiv:2006.03654 (2020).
[31] M. Yin, Z. Yao, Y. Cao, X. Li, Z. Zhang, S. Lin, H. Hu, Disentangled non-local neural networks, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16, Springer, 2020, pp. 191–207.
[32] Y. Wang, S. Tang, Y. Lei, W. Song, S. Wang, M. Zhang, Disenhan: Disentangled heterogeneous graph attention network for recommendation, in: Proceedings of the 29th ACM international conference on information & knowledge management, 2020, pp. 1605–1614.
[33] J. Wu, W. Shi, X. Cao, J. Chen, W. Lei, F. Zhang, W. Wu, X. He, Disenkgat: knowledge graph embedding with disentangled graph attention network, in: Proceedings of the 30th ACM international conference on information & knowledge management, 2021, pp. 2140–2149.
[34] J. Pearl, D. Mackenzie, The book of why: the new science of cause and effect, Basic books, 2018.
[35] J. Pearl, Causality, Cambridge university press, 2009.
[36] Z. Zhang, X. Wang, Z. Zhang, H. Li, Z. Qin, W. Zhu, Dynamic graph neural networks under spatio-temporal distribution shift, Advances in neural information processing systems 35 (2022) 6074–6089.
[37] M. Simonovsky, N. Komodakis, Dynamic edge-conditioned filters in convolutional neural networks on graphs, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 3693–3702.
[38] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907 (2016).
[39] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, Advances in neural information processing systems 27 (2014).
[40] S. Bai, J. Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, arXiv preprint arXiv:1803.01271 (2018).
[41] M. Awad, R. Khanna, M. Awad, R. Khanna, Support vector regression, Efficient learning machines: Theories, concepts, and applications for engineers and system designers (2015) 67–80.
[42] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, Graph wavenet for deep spatial-temporal graph modeling, arXiv preprint arXiv:1906.00121 (2019).
[43] C. Song, Y. Lin, S. Guo, H. Wan, Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting, in: Proceedings of the AAAI conference on artificial intelligence, volume 34, 2020, pp. 914–921.