Designing Specialized Two-Dimensional Graph Spectral Filters for Spatial-Temporal Graph Modeling

Yuxin Chen[1], **Fangru Lin**[1], **Jingyi Huo**[1], **Hui Yan**[1*]

[1]School of Computer Science and Engineering, Nanjing University of Science and Technology, China
{cyx0520,linfangru,918101170104,yanhui}@njust.edu.cn

Abstract

Spatial-temporal graph modeling is challenging due to the diverse node interactions across spatial and temporal dimensions. Recent studies typically adopt Graph Neural Networks (GNNs) to perform node-level aggregation at different time steps, acting as a series of low-pass graph spectral filters, for node interaction modeling. However, these filters, confined to the spatial dimension, are ill-suited for processing signals of nodes with inherent spatial-temporal interdependencies. Moreover, oversimplified low-pass filtering fails to fully exploit information from diverse node interactions. To address these issues, we propose a Spatial-Temporal Spectral Graph Neural Network (STSGNN), which designs specialized two-dimensional (2-D) graph spectral filters for comprehensive spatial-temporal graph modeling. First, based on the normalized Laplacian spectrum of spatial and temporal graphs, we extend existing graph spectral theory from a univariate spatial dimension to a bivariate spatial-temporal dimension through a 2-D Discrete Graph Fourier Transform (2-D DGFT). Then, we leverage the bivariate Bernstein polynomial approximation, with learned basis coefficients, to design 2-D filters with specialized spectral properties for unified spatial-temporal signal filtering. Finally, the filtered signals, with refined spatial-temporal representations, are fed into well-designed pyramidal gated convolution modules to acquire multiple ranges of spatial-temporal dependencies. Experiments on traffic and meteorological prediction tasks demonstrate that STSGNN achieves state-of-the-art performance. Additionally, we visualize the 2-D filters learned from different inputs to enhance the model's interpretability.

Introduction

Spatial-temporal graph modeling is a crucial research subject that focuses on representing and analyzing systems with inherent spatial-temporal interdependencies using graph structures. It has been widely applied to numerous real-world scenarios, such as traffic flow forecasting (Jiang et al. 2023) and air quality inference (Zhang et al. 2020). Effective spatial-temporal graph modeling provides essential insights and accurate predictions that are vital for decision-making and policy formulation in various dynamic systems, thereby improving the service quality of related applications.

Prevailing methods in spatial-temporal graph modeling commonly integrate Graph Neural Networks (GNNs) (Kipf and Welling 2016) with temporal encoders (Tang et al. 2022) to capture the interdependent nature of data across both spatial and temporal dimensions. These methods typically begin by constructing graphs to represent spatial relations among different entities. GNNs then adopt the adjacency matrices of these graphs to perform node-level aggregations for spatial correlation modeling, while temporal encoders focus on capturing dependencies for each entity across different timestamps. By combining GNNs and temporal encoders into a unified framework, many representative GNN-based methods (Zhou et al. 2023) have been developed, achieving advanced performance in spatial-temporal graph modeling.

In general, the node-level aggregation operations in these GNN-based methods originate from a localized first-order approximation of spectral graph convolutions (Du et al. 2022), which can be seen as a special form of low-pass filters. Existing studies (Bo et al. 2021) show that low-pass filtering, which mainly retains the commonality of node features, is the key to developing stable and accurate GNN-based models for spatial-temporal learning. Early attempts (Wu et al. 2019; Cao et al. 2020) adopt a fixed low-pass filter for node representation learning, that is, updating node representation by aggregating neighbors' information over a fixed graph structure, which cannot model the evolving node-wise relations in dynamic systems. Later advanced approaches (Li et al. 2023) adaptively build a series of low-pass filters, each tailored for different time steps, for better spatial-temporal graph modeling.

Nevertheless, most of these works design filters exclusively for refining spatial information and require additional temporal encoders to complete the spatial-temporal modeling. Such a separation of spatial and temporal modeling naturally violates the real-world unified spatial-temporal interdependencies (Song et al. 2020; Wang et al. 2024), which largely hinders the forecasting performance of these models. Moreover, these low-pass filters, focusing on retaining the commonality of related nodes by forcing similar representations, fail to fully exploit systems with diverse node interactions, i.e., interactions between related nodes can result in either similar or distinct observations. For example, in road networks, intersections sharing an upstream-downstream relationship tend to exhibit similar traffic flow, while traffic

*Corresponding Author.

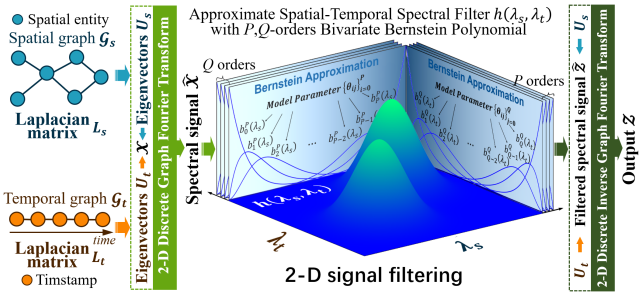

Figure 1: The 2-D graph spectral filtering in STSGNN.

flow between intersections located parallelly on the same origin-destination route often shows an increase in one accompanied by a decrease in the other. These issues prompt us to consider whether we can design filters with appropriate spectral properties to process the observed spatial-temporal signals in a unified manner.

To this end, we propose a Spatial-Temporal Spectral Graph Neural Network (STSGNN), characterized by two-dimensional (2-D) graph spectral filtering for spatial-temporal graph modeling. First, as shown in the left part of Figure 1, akin to building a graph $\mathcal{G}_s$ in the spatial dimension, we treat different timestamps as nodes and establish edge connections between proximate time steps to build a temporal graph $\mathcal{G}_t$. Subsequently, based on the symmetric normalized Laplacian matrices of $\mathcal{G}_s$ and $\mathcal{G}_t$, denoted as L_s and L_t, we introduce a 2-D Discrete Graph Fourier Transform (2-D DGFT) to extend the existing graph spectral theory from the spatial dimension to the spatial-temporal dimension, which allows us to transform the input spatial-temporal signal $\mathcal{X}$ into a 2-D spectral graph domain. As presented in the right part of Figure 1, for an arbitrary 2-D spectral graph filter $h : [0, 2] \times [0, 2] \longmapsto [0, 1] \times [0, 1]$ over the spectrum of L_s and L_t, we employ a P, Q-order bivariate Bernstein polynomial approximation, formulated as $h(\lambda_s, \lambda_t) = \sum_{p=0}^{P} \sum_{q=0}^{Q} \theta_{pq} b_p^P(\lambda_s) b_q^Q(\lambda_t)$, to approximate h. The non-negative coefficients θ_{pq} of the two groups of Bernstein bases $b_p^P(\lambda_s)_{p=0}^{P}$ and $b_q^Q(\lambda_t)_{q=0}^{Q}$ represent the filter values uniformly sampled from the range $[0, 2] \times [0, 2]$. Here, we leverage the spatial-temporal information of the input signals to adaptively generate these coefficients, thereby enabling the construction of filters with specialized spectral properties. After passing through the constructed 2-D filter, we adopt a 2-D Inverse Graph Fourier Transform (2-D IDGFT) to transform the filtered spectral signal back to the nodal domain. Thanks to the decoupling characteristics of the Bernstein bases for each variable, this filtering process can be represented as an integrated tensor-matrix multiplication formula involving L_s, L_t and $\mathcal{X}$. Lastly, we further design a pyramidal gated convolution module to capture multiple ranges of spatial-temporal dependencies, which are crucial for acquiring reliable spatial-temporal predictions. The main contributions of this paper are summarized as follows:

- We extend graph spectral theory to the spatial-temporal dimension using 2-D DGFT, thereby defining 2-D filters over the normalized Laplacian spectrum of the constructed spatial and temporal graphs. Furthermore, a P, Q-order bivariate Bernstein polynomial approximation, with adaptive Bernstein basis coefficients, is utilized to build the 2-D filter with specialized spectral property for comprehensive spatial-temporal signal processing.
- We design a novel pyramidal gated convolution to capture multiple ranges of spatial-temporal dependencies by fusing multi-scale features with gated convolutions.
- We evaluate STSGNN on traffic and meteorological prediction tasks, showing it outperforms state-of-the-art baselines. Visualization experiments also demonstrate its ability to learn interpretable 2-D spectral filters.

Related Work

GNN-based models are pivotal in spatial-temporal graph modeling due to their ability to capture hidden patterns in irregular spatial signals over time. Typically, these models use graph convolutions as low-pass filters (Kipf and Welling 2016; Bo et al. 2021), smoothing node features by propagating low-frequency information, which is often key to their success, as connected nodes are generally assumed to share similar characteristics. Early approaches, such as DCRNN (Li et al. 2018), and STGCN (Yu, Yin, and Zhu 2018), which propagate low-frequency information over graphs based on geographical or physical relations, usually incur misleading message passing (Zhou et al. 2023), as these related nodes sometimes exhibit distinct patterns. To tackle this issue, STFGNN (Li and Zhu 2021) and STGODE (Fang et al. 2021) use the Dynamic Time Warping (DTW) algorithm to construct semantic spatial graphs based on node-wise feature similarity. StemGNN (Cao et al. 2020) and AGCRN (Bai et al. 2020) infer node embeddings from training data and utilize similarities in these embeddings to generate adaptive spatial graphs. Furthermore, innovations like ASTGCN (Guo et al. 2019) and GMAN (Zheng et al. 2020) incorporate attention mechanisms into GNNs to model dynamic systems. These models with reconstructed graphs based on feature similarity can benefit more from the low-pass filtering, which mainly retain the commonality of node features. However, only passing low-frequency information inevitably incur the loss of information arising from diverse node interactions, and continuous low-pass filtering makes node representations indistinguishable, causing over-smoothing. Moreover, these methods, requiring additional temporal encoders to capture spatial and temporal dependencies separately, violate the nature of unified spatial-temporal information propagation in real-world scenarios.

Preliminaries

Notations and Problem Formulation

We use $\mathcal{G} = (V, E, A)$ to represent a graph structure, where $V = \{v_1, v_2, ..., v_N\}$ is a set of N nodes ($|V| = N$), and E is a set of edges connecting these nodes. For spatial-temporal signal, we use $X_t = (x_t^{(1)}, x_t^{(2)}, ..., x_t^{(N)}) \in \mathbb{R}^{N \times D}$ to represent the observation from all entities at the t-th time step, where $x_t^{(i)} \in \mathbb{R}^D$ is the observation vector from

the i-th entity at the t-th time step. Given the T time steps historical spatial-temporal signal $\mathcal{X}_h = (X_1, X_2, \ldots, X_T)$ over a graph $\mathcal{G}$, the goal of spatial-temporal graph modeling is to build a mapping function f that can predict the spatial-temporal signal for the following $T^{'}$ time steps:

$$[X_1, X_2, \ldots, X_T; \mathcal{G}] \xrightarrow{f} [X_{T+1}, X_{T+2}, \ldots, X_{T+T'}] \quad (1)$$

Tensor Calculation

Given two multidimensional tensor $\mathcal{H}$ and $\mathcal{T}$, a tensor multiplication operation can be defined as:

$$\left(\mathcal{H} \times_{12}^{23} \mathcal{T}\right)_{il} = \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \mathcal{H}_{ijk} \cdot \mathcal{T}_{jkl} \quad (2)$$

where $\mathcal{H} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{T} \in \mathbb{R}^{n_2 \times n_3 \times n_4}$ are the input tensors, $\left(\mathcal{H} \times_{12}^{23} \mathcal{T}\right) \in \mathbb{R}^{n_1 \times n_4}$ is the output tensor. The notation $\times_{12}^{23}$ denotes the tensor multiplication operation with certain labeled dimensions of the two inputs that are not included in the output. Specifically, the superscript and subscript in $\times_{12}^{23}$ correspond to the labeled dimensions of the left and right multiplier tensors, $\mathcal{H}$ and $\mathcal{T}$. In this work, we also explore tensor-matrix multiplication, a special case of tensor multiplication, and provide an example as follows:

$$\left(M \times_1^2 \mathcal{H}\right)_{ljk} = \sum_{i=1}^{n_1} M_{li} \cdot \mathcal{H}_{ijk} \quad (3)$$

where $M \in \mathbb{R}^{n_1^{'} \times n_1}$ denotes the multiplier matrix, and $\left(M \times_1^2 \mathcal{H}\right) \in \mathbb{R}^{n_1^{'} \times n_2 \times n_3}$ denotes the output tensor.

Bivariate Bernstein Polynomial Approximation

The bivariate Bernstein polynomial approximation (Phillips and Phillips 2003) is a mathematical method developed for approximating a function of two variables, which is an extension of the Bernstein polynomial used for single-variable functions. Given an arbitrary continuous bivariate function $f(t,k)$ defined on the unit square $t,k \in [0,1] \times [0,1]$, the bivariate Bernstein polynomial approximation of orders P and Q in variables t and k for function f is defined as:

$$\begin{aligned} B_{P,Q}(t,k) &:= \sum_{p=0}^{P} \sum_{q=0}^{Q} \theta_{pq} b_p^P(t) b_q^Q(k) \\ &= \sum_{p=0}^{P} \sum_{q=0}^{Q} f\left(\frac{p}{P}, \frac{q}{Q}\right) \binom{P}{p} t^p (1-t)^{p^{'}} \binom{Q}{q} k^q (1-k)^{q^{'}} \end{aligned} \quad (4)$$

where $b_p^P(t) = \binom{P}{p} t^p (1-t)^{p^{'}}$, $p = 0, 1, ..., P$ and $b_q^Q(k) = \binom{Q}{q} k^q (1-k)^{q^{'}}$, $q = 0, 1, ..., Q$, denote the Bernstein basis polynomials, i.e., two sets of Bernstein bases, for variables t and k, respectively. Especially, we denote $P-p$ and $Q-q$ as $p^{'}$ and $q^{'}$ for the sake of simplicity. $\theta_{pq} = f\left(\frac{p}{P}, \frac{q}{Q}\right)$ represents the value of the function $f(t,k)$ at the points $\left(\frac{p}{P}, \frac{q}{Q}\right)$,

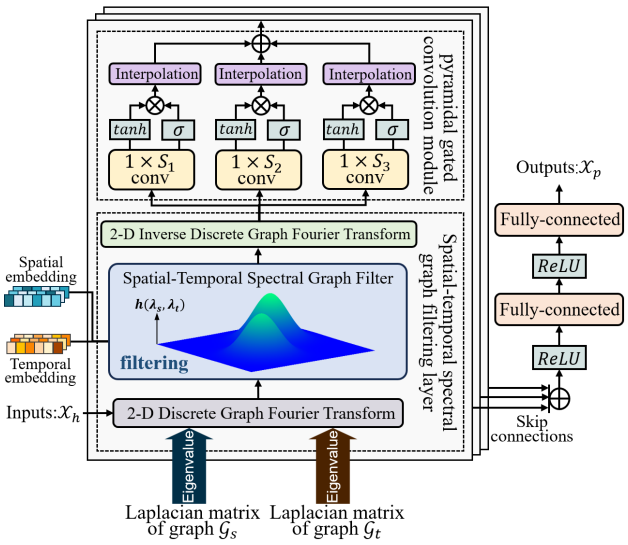

Figure 2: Framework of the proposed STSGNN.

serving as the coefficient for $b_p^P(t)$ and $b_q^Q(t)$. This formulation uses a sum of products of Bernstein basis polynomials in two variables to construct a polynomial, which closely approximates the behavior of $f(t,k)$ over the unit square.

Model

Figure 2 shows the framework of the proposed STSGNN, consisting of stacked Spatial-Temporal (ST) blocks and an output layer. Each ST block includes a spatial-temporal spectral graph filtering layer and a pyramidal gated convolution module, detailed below.

Adjacency Matrix Construction

Two kinds of graphs are utilized in our model. The first graph is the spatial graph $\mathcal{G}_s = (V_s, E_s, A_s)$, where V_s is a finite set of N spatial nodes, which corresponds to N entities in the systems, E_s is as set of edges among these nodes, and $A_s \in \mathbb{R}^{N \times N}$ is the spatial adjacency matrix. Especially, in this work, we focus on traffic and meteorological forecasting tasks. Following previous studies (Yu, Yin, and Zhu 2018; Zhou et al. 2023), we build node connections based on geographical information, which reveals the intuitive node-wise interactions within the traffic and meteorological systems. The mathematical formulation of A_s is presented as:

$$(A_s)_{ij} = \begin{cases} 1, & \text{if } \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) \geq \epsilon, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

where d_{ij} is the distance between node i and j. σ^2 and ϵ are thresholds to control sparsity of matrix A_s

The second graph is the temporal graph $\mathcal{G}_t = (V_t, E_t, A_t)$, where V_t is a finite set of T nodes, corresponding to T consecutive time steps, E_t is the set of edges connecting proximate time steps, and $A_t \in \mathbb{R}^{T \times T}$ is the temporal adjacency matrix. The mathematical formulation of A_t is presented as:

$$(A_t)_{ij} = \begin{cases} 1, & \text{if } 0 < |i-j| \leq \tau, \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

where $(A_t)_{ij} = 1$ denotes that the i-th time step is related to the j-th time step. This graph is constructed under the assumption that interactions are most pronounced between proximate time steps, a principle that aligns with real-world temporal correlations. Accordingly, we adopt the 2τ nearest time steps as contextual neighbors for each temporal node.

Unlike most existing works that build 1-D low-pass filters based on various constructed spatial graphs, our STSGNN can develop arbitrary 2-D filters, encompassing both spatial and temporal dimensions, specialized for input signals over the normalized Laplacian spectrum of two intuitive graphs.

Spatial-Temporal Graph Spectral Filtering

Given a graph signal $X \in \mathbb{R}^{N \times F}$ and the eigendecomposition of the symmetric normalized Laplacian matrix $L = U\Lambda U^T$, where U is the matrix of eigenvectors, and $\Lambda = diag[\lambda_1, \lambda_2, \ldots, \lambda_N]$ is the diagonal matrix of eigenvalues. We formulate the classic form of graph spectral filtering as:

$$Ug(\Lambda)U^T X \tag{7}$$

Specifically, we first use Graph Fourier Transform (GFT) to convert the signal X into the spectral domain by multiplying it with U^T. After that, we apply a filter $g(\Lambda)$, where g represents a 1-D filter function, to the spectral signal and then transform it back to the nodal domain via Inverse Graph Fourier Transform (IGFT), i.e., multiplying the filtered signal with U. A well-suited filter can effectively refine signal representations and enhances model performance. However, most existing GNN-based prediction models, which primarily approximate g as low-pass filters specified for the spatial dimension, inherently struggle with signals exhibiting complex spatial-temporal interdependencies.

For the aforementioned reason, our specialized spectral spatial-temporal filtering is developed. Firstly, inspired by the 2-D discrete Fourier transform, we extend the original GFT to the 2-D Discrete Graph Fourier Transform (2-D DGFT) to analyze the spectral content of spatial-temporal signals in a discrete manner. This is achieved by treating the input spatial-temporal signal as a multidimensional tensor and applying discrete GFT to it. Specifically, let a multidimensional tensor $\mathcal{X} \in \mathbb{R}^{T \times N \times F}$ represent the input spatial-temporal signal, $L_s = U_s \Lambda_s U_s^T$ and $L_t = U_t \Lambda_t U_t^T$ denote the eigendecomposition of the symmetric normalized Laplacian matrices of the spatial and temporal graphs, where U_s and U_t are the matrices of the corresponding eigenvectors, $\Lambda_s = diag[\lambda_{s,1}, \lambda_{s,2}, \ldots, \lambda_{s,N}]$ and $\Lambda_t = diag[\lambda_{t,1}, \lambda_{t,2}, \ldots, \lambda_{t,T}]$ are the diagonal matrices of corresponding eigenvalues. We define the 2-D DGFT as:

$$\mathcal{GF}_{st}\left(\mathcal{X}\right) = U_t^T \times_1^2 \left(U_s^T \times_2^2 \mathcal{X}\right) = \widehat{\mathcal{X}} \tag{8}$$

where $\widehat{\mathcal{X}}$ denotes the signal in the spectral domain.Then, we can further define the 2-D Inverse Discrete Graph Fourier Transform (2-D IDGFT) as:

$$\mathcal{GF}_{st}^{-1}(\widehat{\mathcal{X}}) = U_s \times_2^2 \left(U_t \times_1^2 \widehat{\mathcal{X}}\right) = \mathcal{X} \tag{9}$$

Based on the definitions of 2-D DGFT and 2-D IDGFT, we employ $h : [0, 2] \times [0, 2] \longmapsto [0, 1] \times [0, 1]$ to denote an arbitrary 2-D filter function and thus obtaining the formulation of the 2-D filtering operation on signal $\mathcal{X}$ as:

$$\mathcal{GF}_{st}^{-1}\left(h(\lambda_s, \lambda_t) \odot \mathcal{GF}_{st}(\mathcal{X})\right) \tag{10}$$

with $h(\lambda_s, \lambda_t)$ equals to:

$$\begin{bmatrix} h(\lambda_{s,1}, \lambda_{t,1}) & h(\lambda_{s,2}, \lambda_{t,1}) & \ldots & h(\lambda_{s,N}, \lambda_{t,1}) \\ h(\lambda_{s,1}, \lambda_{t,2}) & h(\lambda_{s,2}, \lambda_{t,2}) & \ldots & h(\lambda_{s,N}, \lambda_{t,2}) \\ \vdots & \vdots & & \vdots \\ h(\lambda_{s,1}, \lambda_{t,T}) & h(\lambda_{s,2}, \lambda_{t,T}) & \ldots & h(\lambda_{s,N}, \lambda_{t,T}) \end{bmatrix} \tag{11}$$

where $\odot$ is the Hadamard product. From the formulation above we can clearly see the whole process of 2-D graph spectral filtering operations. Specifically, we first employ the 2D-DGFT to transform the input signal into the 2-D spectral domain, allowing the application of a 2-D filter $h(\lambda_s, \lambda_t)$ to the spectral signal $\widehat{\mathcal{X}}$ for unified exploitation of spatial-temporal interdependencies. Then we utilize 2D-IDGFT to transform the processed signal back to the nodal domain.

Given the formula of 2-D filtering operation in Eq 10, the key objective of our work is to design a 2-D filter well-suited for the input signal. Motivated by the work in (He et al. 2021), we leverage the bivariate Bernstein polynomial approximation to achieve this goal. Based on the illustration of the Bivariate Bernstein Polynomial Approximation in Preliminaries, we have Lemma 1 as follows:

Lemma 1. Given an arbitrary bivariate continuous function $f(t, k)$ defined on $[0, 1] \times [0, 1]$, let $B_{P,Q}(t, k)$ denote its bivariate Bernstein polynomial approximation as defined in Eq 4. We have $B_{P,Q}(t, k) \to f(t, k)$ as $P, Q \to \infty$.

For the filter function $h : [0, 2] \times [0, 2] \longmapsto [0, 1] \times [0, 1]$, we let $t = \frac{\lambda_s}{2}$, $k = \frac{\lambda_t}{2}$, and $f(t, k) = h(2t, 2k)$, so that the bivariate Bernstein polynomial approximation becomes applicable, where $\theta_{pq} = f(\frac{p}{P}, \frac{q}{Q}) = h(\frac{2p}{P}, \frac{2q}{Q})$, $b_p^P(\frac{\lambda_s}{2}) = \binom{P}{p}\left(\frac{\lambda_s}{2}\right)^p \left(1 - \frac{\lambda_s}{2}\right)^{p'}$ and $b_q^Q(\frac{\lambda_t}{2}) = \binom{Q}{q}\left(\frac{\lambda_t}{2}\right)^q \left(1 - \frac{\lambda_t}{2}\right)^{q'}$ for $p = 1, 2, ..., P$ and $q = 1, 2, ..., Q$. Consequently, we can approximate $h(\lambda_s, \lambda_t)$ by:

$$\begin{aligned} & B_{P,Q}\left(\frac{\lambda_s}{2}, \frac{\lambda_t}{2}\right) \\ &= \sum_{p=0}^{P}\sum_{q=0}^{Q} \theta_{pq} b_p^P(\lambda_s) b_q^Q(\lambda_t) \\ &= \sum_{p=0}^{P}\sum_{q=0}^{Q} \theta_{pq} \frac{1}{2^P}\binom{P}{p}\lambda_s^p \left(2 - \lambda_s\right)^{p'} \frac{1}{2^Q}\binom{Q}{q}\lambda^q \left(2 - \lambda_t\right)^{q'} \\ &= \sum_{p=0}^{P} \frac{1}{2^P}\binom{P}{p}\lambda_s^p \left(2 - \lambda_s\right)^{p'} \sum_{q=0}^{Q} \theta_{pq} \frac{1}{2^Q}\binom{Q}{q}\lambda^q \left(2 - \lambda_t\right)^{q'} \end{aligned} \tag{12}$$

and Lemma 1 ensures that $B_{P,Q}\left(\frac{\lambda_s}{2}, \frac{\lambda_t}{2}\right) \to h\left(\lambda_s, \lambda_t\right)$ as $P, Q \to \infty$.

After that, we aim to approximate arbitrary 2-D spectral filtering operations with $B_{P,Q}\left(\frac{\lambda_s}{2}, \frac{\lambda_t}{2}\right)$. In practice, we first

reformulate the 2-D spectral filtering in Eq 10 into a pure tensor-matrix multiplication form as:

$$\begin{aligned}
&\mathcal{GF}_{st}^{-1}\left(h(\lambda_s,\lambda_t)\odot\mathcal{GF}_{st}(\mathcal{X})\right)\\
&=U_s\times_2^2\left(U_t\times_1^2\left(h(\lambda_s,\lambda_t)\odot\left(U_t\times_1^2\left(U_s\times_2^2\mathcal{X}\right)\right)\right)\right)\\
&=U_s\times_2^2\left(U_t\times_1^2\left(\boldsymbol{H}(\lambda_s,\lambda_t)\times_{12}^{24}\left(U_t\times_1^2\left(U_s\times_2^2\mathcal{X}\right)\right)\right)\right)\\
&=\left(U_s\times_3^2\left(U_t\times_1^2\boldsymbol{H}(\lambda_s,\lambda_t)\times_1^2U_t^T\right)\times_1^4U_s^T\right)\times_{12}^{24}\mathcal{X}
\end{aligned}\tag{13}$$

with $\boldsymbol{H}\in\mathbb{R}^{T\times T\times N\times N}$ equals to:

$$diag\begin{bmatrix}diag\left[h(\lambda_{s,1},\lambda_{t,1}),\ldots,h(\lambda_{s,N},\lambda_{t,1})\right]\\ \vdots\\ diag\left[h(\lambda_{s,1},\lambda_{t,T}),\ldots,h(\lambda_{s,N},\lambda_{t,T})\right]\end{bmatrix}\tag{14}$$

Then, we replace $h(\lambda_{s,i},\lambda_{t,j})$ with $B_{P,Q}\left(\frac{\lambda_{s,i}}{2},\frac{\lambda_{t,j}}{2}\right)$ for $i=1,2,\ldots,N$ and $j=1,2,\ldots,T$, to approximate the 2-D filtering operation in Eq 10. Particularly, given a spatial-temporal signal $\mathcal{X}$, we leverage the decoupling characteristics of the bivariate Bernstein polynomial approximation to achieve the 2-D filtering in spatial-temporal spectral graph filtering layer via tensor-matrix multiplication:

$$\begin{aligned}
&\left(U_s\times_3^2\left(U_t\times_1^2\boldsymbol{B}_{PQ}(\frac{\lambda_s}{2},\frac{\lambda_t}{2})\times_1^2U_t^T\right)\times_1^4U_s^T\right)\times_{12}^{24}\mathcal{X}\\
&=\sum_{p=0}^{P}\frac{1}{2^P}\binom{P}{p}L_s^p\left(2I_s-L_s\right)^{p'}\\
&\quad\times_2^2\sum_{q=0}^{Q}\theta_{pq}\frac{1}{2^Q}\binom{Q}{q}L_t^q\left(2I_t-L_t\right)^{q'}\times_1^2\mathcal{X}
\end{aligned}\tag{15}$$

with $\boldsymbol{B}_{PQ}\in\mathbb{R}^{T\times T\times N\times N}$ equals to:

$$diag\begin{bmatrix}diag\left[B_{P,Q}\left(\frac{\lambda_{s,1}}{2},\frac{\lambda_{t,1}}{2}\right),\ldots,B_{P,Q}\left(\frac{\lambda_{s,N}}{2},\frac{\lambda_{t,1}}{2}\right)\right]\\ \vdots\\ diag\left[B_{P,Q}\left(\frac{\lambda_{s,1}}{2},\frac{\lambda_{t,T}}{2}\right),\ldots,B_{P,Q}\left(\frac{\lambda_{s,N}}{2},\frac{\lambda_{t,T}}{2}\right)\right]\end{bmatrix}\tag{16}$$

where $I_s\in\mathbb{R}^{N\times N}$ and $I_t\in\mathbb{R}^{T\times T}$ are identity matrices, each coefficient θ_{pq} can be either set to $h(\frac{2p}{P},\frac{2q}{Q})$ to approximate a predetermined filter h, or learnt from the graph structure and signal in an end-to-end manner. Especially, for an arbitrary continuous 2-D filter function $h:[0,2]\times[0,2]\longmapsto[0,1]\times[0,1]$, by setting $\theta_{pq}=h(\frac{2p}{P},\frac{2q}{Q})$, Lemma 1 ensures that $B_{P,Q}\left(\frac{\lambda_s}{2},\frac{\lambda_t}{2}\right)\to h(\lambda_s,\lambda_t)$ as $P,Q\to\infty$. Consequently, we have:

$$\begin{aligned}
&\left(U_s\times_3^2\left(U_t\times_1^2\boldsymbol{B}_{PQ}(\frac{\lambda_s}{2},\frac{\lambda_t}{2})\times_1^2U_t^T\right)\times_1^4U_s^T\right)\times_{12}^{24}\mathcal{X}\\
&\to\left(U_s\times_3^2\left(U_t\times_1^2\boldsymbol{H}(\lambda_s,\lambda_t)\times_1^2U_t^T\right)\times_1^4U_s^T\right)\times_{12}^{24}\mathcal{X}\\
&\quad=\mathcal{GF}_{st}^{-1}\left(h(\lambda_s,\lambda_t)\odot\mathcal{GF}_{st}(\mathcal{X})\right)
\end{aligned}\tag{17}$$

as $\theta_{pq}=h(\frac{2p}{P},\frac{2q}{Q})$ and $P,Q\to\infty$.

We integrate spatial and temporal information of the input signal to adaptively generate Bernstein basis coefficients for specialized 2-D spectral filtering. Spatial information is preserved using position embedding, while temporal information is transformed into a temporal embedding. These embeddings are combined to generate coefficients, enabling the learning of a specialized 2-D filter. The construction of the coefficient sets is detailed in the supplementary material.

Pyramidal Gated Convolution

Recent studies (Lan et al. 2022) show that diverse spatial-temporal information enhances insight into dynamic systems and improves models' inference capability. To achieve this, we deploy a pyramidal gated convolution module after each spatial-temporal spectral graph filtering layer to effectively capture multi-range spatial-temporal information.

The specific structure of the pyramidal gated convolution module is depicted in Figure 2, which employs a L-Level pyramid gated convolution layer to extract multi-scale spatial-temporal information for a filtered feature $\mathcal{Z}\in\mathbb{R}^{T\times N\times F}$. Specifically, at the l-th level ($l=1,...,L$), we apply a convolution layer, where its kernel size and stride equal to $1\times2^{l-1}$, to generate a feature $\mathcal{Z}^{'}\in\mathbb{R}^{\frac{T}{2^{l-1}}\times N\times2F}$. This feature is then fed into the gated tanh unit to enhance its representation ability as follows:

$$\Gamma*_\tau\mathcal{Z}=\phi(\mathcal{Z}_{fh}^{'})\odot\sigma(\mathcal{Z}_{sh}^{'})\in\mathbb{R}^{\frac{T}{2^{l-1}}\times N\times F}\tag{18}$$

where Γ denotes the convolution kernel, $*_\tau$ is gated convolution operation, $\phi(\cdot)$ is tanh function, $\sigma(\cdot)$ is the sigmoid function ,and $\mathcal{Z}_{fh}^{'}\in\mathbb{R}^{\frac{T}{2^{l-1}}\times N\times F}$ and $\mathcal{Z}_{sh}^{'}\in\mathbb{R}^{\frac{T}{2^{l-1}}\times N\times F}$ are the first and second half of $\mathcal{Z}^{'}$ relative to the channel dimension, respectively. After that, we upsample the output feature of gated convolution unit back to $\mathbb{R}^{T\times N\times F}$ using bilinear neighbor interpolation. For convenience, the upsampled feature is denoted as $\mathcal{Z}^l$. Finally, the output of pyramidal gated convolution module can be computed as:

$$\mathcal{Z}_{out}=Linear(\mathcal{Z}^{l_1}\oplus\mathcal{Z}^{l_2}\oplus\mathcal{Z}^{l_3})\tag{19}$$

where $\oplus$ denotes feature concatenation, and the concatenated feature representation is fed into a fully connected layer to get the output $\mathcal{Z}_{out}\in\mathbb{R}^{T\times N\times F}$ of the pyramidal gated convolution module. This module has an advantage in extracting both short-term and long-term features of the spatial-temporal data by using multi-scale gated convolution, and alleviating the noise within the dynamic systems through the stride convolution and interpolation operations.

Other Components

Output layer We apply a skip connection after each ST block to convert the outputs $\mathcal{Z}_{out}\in\mathbb{R}^{T\times N\times F}$ into $\mathcal{Z}_{sk}\in\mathbb{R}^{T\times N\times F_{sk}}$, where F_{sk} represents the skip dimension. The outputs from all skip connection layers are then summed and passed through two fully connected layers with ReLU activation, directly generating predictions for the next $T^{'}$ steps.

Complexity Analysis The computational cost of STS-GNN mainly comes from the spatial-temporal spectral

Models	PeMS03			PeMS04			PeMS07			PEMS08			KA-UW		KA-TEMP		KA-PM$_{2.5}$	
	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
VAR	23.65	24.51%	38.26	24.54	17.24%	38.61	50.22	32.22%	75.63	19.19	13.10%	29.81	2.228	2.865	8.729	10.278	27.562	34.796
FC-LSTM	21.33	23.33%	35.11	27.14	18.20%	41.59	29.89	13.39%	45.74	23.08	14.91%	35.06	2.172	2.755	3.410	3.981	20.200	28.698
STGCN	17.02	16.98%	29.36	21.16	13.83%	34.89	24.94	10.99%	39.17	17.50	11.29%	27.09	2.005	2.648	2.471	3.184	17.401	27.033
STSGCN	17.49	16.81%	29.22	21.20	13.91%	33.59	24.30	10.22%	39.08	17.13	10.98%	26.77	1.706	2.288	1.674	2.377	15.247	23.433
SteamGNN	16.42	17.07%	26.90	21.36	14.82%	33.37	27.45	14.32%	41.35	16.39	10.75%	25.75	1.729	2.325	2.087	2.687	16.751	25.169
MTGNN	16.87	17.37%	26.89	19.98	14.13%	31.92	23.92	12.43%	35.86	15.03	10.23%	24.86	1.867	2.513	2.035	2.743	15.861	24.643
AGCRN	15.70	14.86%	27.52	19.83	13.10%	32.56	21.17	8.90%	35.09	16.19	10.38%	25.56	1.689	2.286	1.685	2.383	15.128	23.601
STGODE	16.50	16.69%	27.84	20.84	13.77%	32.82	22.59	10.14%	37.55	16.81	10.62%	25.98	1.699	2.296	1.759	2.401	14.864	22.926
STGNCDE	15.57	15.06%	27.09	19.21	12.77%	31.09	20.62	8.85%	34.04	15.45	9.92%	24.81	1.747	2.335	2.069	2.821	14.602	23.067
GRETO	15.62	15.82%	27.27	19.89	13.38%	32.03	21.27	9.13%	34.39	15.94	10.26%	25.28	1.713	2.315	1.801	2.498	14.351	22.963
ASTGCN	17.29	17.18%	29.37	22.84	16.48%	35.01	23.87	10.61%	37.55	18.41	12.31%	27.68	1.851	2.393	2.295	3.051	15.378	23.486
GMAN	15.84	15.52%	26.71	19.28	13.14%	31.58	20.86	9.06%	34.06	14.98	9.82%	24.43	1.755	2.347	1.989	2.787	14.726	22.988
DSTAGNN	15.52	15.28%	27.11	19.37	12.78%	31.56	21.48	9.04%	34.49	15.76	9.98%	24.72	1.742	2.349	1.716	2.450	14.922	23.501
STSGNN	**14.61**	**14.43%**	**25.24**	**18.69**	**12.24%**	**30.84**	**19.87**	**8.29%**	**32.97**	**14.38**	**9.25%**	**23.81**	**1.671**	**2.258**	**1.635**	**2.252**	**13.890**	**22.591**

Table 1: Comparison on traffic and meteorological datasets.

graph filtering in Equation 15. The time complexity is $O(PQF(NE_t + TE_s))$, where E_s and E_t represent the number of edges in $\mathcal{G}_s$ and $\mathcal{G}_t$, respectively. While this formula implies high computational complexity, the sparsity of the constructed graphs significantly reduces the actual cost, as the time complexity mainly depends on E_s and E_t.

Experiments

Datasets

To evaluate the performance of STSGNN, we conduct extensive experiments on six cross-domain spatial-temporal graph datasets, each with diverse and dynamic node-wise correlations. **Traffic**: We utilize four widely-studied traffic forecasting datasets from the Caltrans Performance Measurement System (PeMS) (Chen et al. 2001), including **PeMS03**, **PeMS04**, **PeMS07**, and **PeMS08**. **Climate**: We use three meteorological characteristics, including U wind components, urban temperature recordings, and PM2.5 concentrations, collected from 184 main cities across China (KnowAir) (Wang et al. 2020), which we denote as **KA-UW**, **KA-TEMP**, and **KA-PM**$_{2.5}$, respectively.

Baselines

We compare our method with respective three types of baselines. (1)Time Series Prediction Models: We choose VAR (Stock and Watson 2001) and FC-LSTM (Sutskever, Vinyals, and Le 2014). (2)GNN-based Models: We choose STGCN (Yu, Yin, and Zhu 2018), STSGCN (Song et al. 2020), SteamGNN (Cao et al. 2020), MTGNN (Wu et al. 2020), AGCRN (Bai et al. 2020), STGODE (Fang et al. 2021), STGNCDE (Choi et al. 2022), and GRETO (Zhou et al. 2023). (3) Self-attention-based Models: We choose ASTGCN (Guo et al. 2019), GMAN (Zheng et al. 2020) and DSTAGNN (Lan et al. 2022). All methods are re-implemented based on their original configurations.

Experimental Settings

All experiments are conducted on a Linux server (CPU: Intel(R) Xeon(R) Gold 5218R CPU @ 2.10GHz, GPU: NVIDIA GeForce GTX 3090). The following hyperparameters were established for STSGNN: the orders of the bivariate Bernstein polynomial approximation are set to $P = 10$, $Q = 5$. The sizes of L-level pyramid gated convolution are set to $\{l_1, l_2, l_3\} = \{1, 3, 6\}$. The dimension of all hidden representations in the ST blocks is set to $F = 32$, with a total of 8 ST blocks are used. We train our model using the Adam optimizer with a learning rate of 0.002, 100 epochs, early stopping with a patience of 20, and a batch size of 32. The L1 loss function is used to evaluate model performance. We split all datasets in chronological order into 60% for training, 20% for validation, and 20% for testing. The 12 steps of historical data are used to predict the observations in the following 12 time steps. For the traffic dataset, accuracy is measured using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). For the weather datasets, following (Wang et al. 2020), MAPE is excluded due to the presence of negative values. Each experiment is repeated five times, with the average performance reported, and **bold** and underline indicate the 1st and 2nd best results, respectively. The source code is included in the supplementary material and can also be accessed at https://github.com/Eighree/STSGNN.

Performance Comparison

Table 1 compares the performance of our STSGNN with various baselines. From the table, we can make the following observations: (1) GNN-based models and self-attention-based models perform better than traditional time series prediction models, which fail to capture spatial dependencies. (2) Our STSGNN outperforms all the baselines across all metrics. Specifically, compared to the second-best method, STSGNN achieves an average improvement of 4.1%/5.5%/3.0% for MAE/MAPE/RMSE in traffic prediction tasks, and improvements of 3.2%/2.7% for MAE/RMSE in meteorological forecasting tasks. (3) Self-attention-based models generally outperform GNN-based models, as they tend to perform time-specific low-pass filtering on spatial-temporal signals, thereby reducing the risk of misleading message passing. (4) GRETO seeks to address the limita-

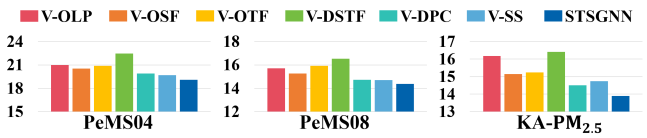

Figure 3: Ablation experiments of STSGNN

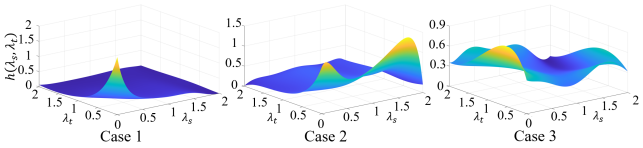

Figure 4: 2-D Filters learnt from different inputs

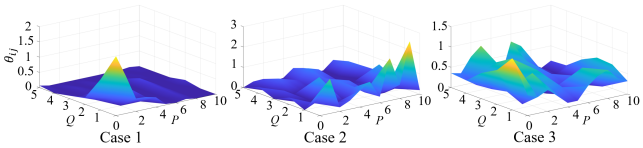

Figure 5: Coefficients θ_{ij} learnt from different inputs.

tions of low-pass filtering through signed message passing and achieves competitive performance. However, it struggles to clearly represent its learned filters and fails to model spatial-temporal correlations in a unified way, limiting its effectiveness in spatial-temporal graph modeling.

Model Analysis

Ablation Study To verify the effectiveness of different modules in STSGNN, we conducted an ablation study on three datasets, including PeMS04, PeMS08, and KA-PM$_{2.5}$, by comparing STSGNN with the following variants [1]:

- V-OLP: only apply low-pass filtering on the input signal.
- V-OSF: only utilize spatial graph spectral filtering.
- V-OTF: only utilize temporal graph spectral filtering.
- V-DSTF: discard the 2-D graph filtering operation.
- V-DPC: discard the pyramidal gated convolution module.
- V-SS: utilize spatial graph based on pattern similarity.

Figure 3 reports the comparison results, from which we can draw the following conclusions: (1) The comparison between STSGNN and V-OLP shows that processing signals with specialized filters is clearly superior to using low-pass filters. (2) STSGNN outperforms V-OSF and V-OST, highlighting the importance of integrating both spatial and temporal information for spatial-temporal graph modeling. (3) V-DSTF performs significantly worse than STSGNN, indicating that processed signals with clear spatial-temporal patterns can enhance model performance. (4) V-DPC's performance is worse than STSGNN's because this variant fails to exploit multiple ranges of spatial-temporal information within dynamic systems. (5) STSGNN performs better than V-SS, illustrating that performing signal filtering over the graph based on diverse node interactions provides richer information for spatial-temporal graph modeling compared to filtering based solely on pattern similarity.

Visualization of 2-D filters In this part, we construct three sub-graphs of equal size, each associated with signals exhibiting distinct spatial-temporal characteristics, based on PeMS04 dataset. We then investigate the spectral properties

[1] V-OLP and V-SS are detailed in the supplementary material.

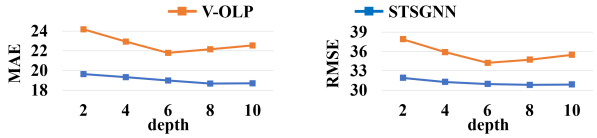

Figure 6: Comparing STGNN and V-OLP at different depths

of the constructed 2-D filters tailored to these different inputs through visualization experiments. Specifically, for the first two sub-graphs, we connect nodes with similar traffic flow patterns. In the first, traffic flow at each node fluctuates smoothly over time, while in the second, the fluctuations are larger. The third sub-graph randomly connects nodes, ignoring spatial-temporal patterns. We have detailed the construction of the three sub-graphs in the supplementary material.

We feed these sub-graphs and their corresponding signals into STSGNN and visualize the learned 2-D filters in Figure 4. In the first case, the model learns a filter that primarily passes low-frequency information in both the spatial and temporal dimensions, reflecting high assortativity in these two dimensions. In the second case, STSGNN uses a low-pass filter for the temporal dimension and a complex comb filter for the spatial dimension to manage erratic traffic transmissions. In the last case, the model learns more complex 2-D filters to handle diverse node interactions across both spatial and temporal dimensions. By observing the characteristics of these inputs and their associated filters, we find that STSGNN can learn 2-D filters tailored to the characteristics of the inputs. In Figure 5, we further visualize the coefficients θ_{pq} learned in this experiment. By comparing Figure 4 and Figure 5, we can observe that the curves of the filters and the curves of the coefficients are almost the same. This is because the basis coefficients are highly correlated with the spectral properties of the target filter, which indicates that the filters learned by STSGNN have strong interpretability.

Network Depth Analysis Unlike existing graph-based methods prone to over-smoothing from simple low-pass filtering, our STSGNN uses 2-D filters with complex spectral properties, making it robust and suitable for deeper networks. Figure 6 compares STSGNN with V-OLP, which relies only on low-pass filtering, at different network depths, i.e., the number of ST blocks. Experiments on the PeMS04 dataset show that as depth increases, V-OLP's performance declines, while STSGNN remains stable, demonstrating its robustness in spatial-temporal graph modeling.

Conclusion

We propose a novel STSGNN model for spatial-temporal graph modeling in the spectral domain, using a 2-D Discrete Graph Fourier Transform and bivariate Bernstein polynomial approximation for unified spatial-temporal spectral filtering. To the best of our knowledge, this is the first attempt to extend existing graph spectral theory into the bivariate spatial-temporal dimension. Additionally, pyramidal gated convolution modules capture multi-range spatial-temporal information, enhancing model performance. Extensive experiments demonstrate the superiority of our model.

References

Bai, L.; Yao, L.; Li, C.; Wang, X.; and Wang, C. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33: 17804–17815.

Bo, D.; Wang, X.; Shi, C.; and Shen, H. 2021. Beyond low-frequency information in graph convolutional networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 3950–3957.

Bui, K.-H. N.; Cho, J.; and Yi, H. 2022. Spatial-temporal graph neural network for traffic forecasting: An overview and open research issues. *Applied Intelligence*, 52(3): 2763–2774.

Cao, D.; Wang, Y.; Duan, J.; Zhang, C.; Zhu, X.; Huang, C.; Tong, Y.; Xu, B.; Bai, J.; Tong, J.; et al. 2020. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in neural information processing systems*, 33: 17766–17778.

Chen, C.; Petty, K.; Skabardonis, A.; Varaiya, P.; and Jia, Z. 2001. Freeway performance measurement system: mining loop detector data. *Transportation research record*, 1748(1): 96–102.

Choi, J.; Choi, H.; Hwang, J.; and Park, N. 2022. Graph neural controlled differential equations for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 6367–6374.

Du, L.; Shi, X.; Fu, Q.; Ma, X.; Liu, H.; Han, S.; and Zhang, D. 2022. Gbk-gnn: Gated bi-kernel graph neural networks for modeling both homophily and heterophily. In *Proceedings of the ACM Web Conference 2022*, 1550–1558.

Fang, Z.; Long, Q.; Song, G.; and Xie, K. 2021. Spatial-temporal graph ode networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 364–373.

Guo, S.; Lin, Y.; Feng, N.; Song, C.; and Wan, H. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 922–929.

He, M.; Wei, Z.; Xu, H.; et al. 2021. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. *Advances in Neural Information Processing Systems*, 34: 14239–14251.

Jiang, R.; Wang, Z.; Yong, J.; Jeph, P.; Chen, Q.; Kobayashi, Y.; Song, X.; Fukushima, S.; and Suzumura, T. 2023. Spatio-temporal meta-graph learning for traffic forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 8078–8086.

Kipf, T. N.; and Welling, M. 2016. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Lan, S.; Ma, Y.; Huang, W.; Wang, W.; Yang, H.; and Li, P. 2022. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting. In *International conference on machine learning*, 11906–11917. PMLR.

Li, M.; and Zhu, Z. 2021. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 4189–4196.

Li, Y.; Yu, R.; Shahabi, C.; and Liu, Y. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations*.

Li, Z.; Yu, J.; Zhang, G.; and Xu, L. 2023. Dynamic spatio-temporal graph network with adaptive propagation mechanism for multivariate time series forecasting. *Expert Systems with Applications*, 216: 119374.

Phillips, G. M.; and Phillips, G. M. 2003. Bernstein polynomials. *Interpolation and Approximation by Polynomials*, 247–290.

Song, C.; Lin, Y.; Guo, S.; and Wan, H. 2020. 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, 914–921.

Stock, J. H.; and Watson, M. W. 2001. Vector autoregressions. *Journal of Economic perspectives*, 15(4): 101–115.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Tang, X.; Chen, H.; Xiang, W.; Yang, J.; and Zou, M. 2022. Short-term load forecasting using channel and temporal attention based temporal convolutional network. *Electric Power Systems Research*, 205: 107761.

Wang, S.; Li, Y.; Zhang, J.; Meng, Q.; Meng, L.; and Gao, F. 2020. Pm2. 5-gnn: A domain knowledge enhanced graph neural network for pm2. 5 forecasting. In *Proceedings of the 28th international conference on advances in geographic information systems*, 163–166.

Wang, Y.; Xu, Y.; Yang, J.; Wu, M.; Li, X.; Xie, L.; and Chen, Z. 2024. Fully-Connected Spatial-Temporal Graph for Multivariate Time-Series Data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 15715–15724.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; and Zhang, C. 2020. 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*, 753–763.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; and Zhang, C. 2019. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 1907–1913.

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

Zhang, K.; Thé, J.; Xie, G.; and Yu, H. 2020. Multi-step ahead forecasting of regional air quality using spatial-temporal deep neural networks: a case study of Huaihai Economic Zone. *Journal of Cleaner Production*, 277: 123231.

Zheng, C.; Fan, X.; Wang, C.; and Qi, J. 2020. Gman: A graph multi-attention network for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 1234–1241.

Zhou, Z.; Huang, Q.; Lin, G.; Yang, K.; Bai, L.; and Wang, Y. 2023. Greto: remedying dynamic graph topology-task discordance via target homophily. In *The eleventh international conference on learning representations*.