Multi-level Relational Learning with Synergistic Graphs for Multivariate Time Series Forecasting

Yang Xie yxie@std.uestc.edu.cn
Qiao Liu qliu@uestc.edu.cn
Rui Hou hour@std.uestc.edu.cn
Tingting Dai ttdai_18@outlook.com
Tian Lan* lantian1029@uestc.edu.cn
University of Electronic Science and Technology of China, Chengdu, 611731, China

Editors: Vu Nguyen and Hsuan-Tien Lin

Abstract

Multivariate Time Series (MTS) forecasting involves analyzing the evolution and interrelationships of multiple variables over time. Effectively mining relationships between MTS variables remains challenging as variables may imply multiple relational patterns. Recently, graph-based approaches have exhibited substantial effectiveness in capturing relationships between MTS variables. However, these methods often adhere to the paradigm of capturing low-level pairwise relationships, which limits their ability to capture other high-level beyond pairwise relational patterns. To address this issue, we present a synergistic graph learning framework that combines the modeling advantages of graphs and hypergraphs to uncover more comprehensive relational patterns. This framework mainly consists of two parts. Firstly, we introduced a Synergistic Relation Construction module, which incorporates dynamic graph and hypergraph structures to model low-level pairwise and high-level beyond pairwise relationships among variables, representing multi-level relational patterns through obtained adjacency matrices and incidence matrices. Additionally, we developed a Synergistic Relation Learning mechanism, that leverages novel synergistic graph and hypergraph convolutional networks to facilitate spatial dependency interactions across multi-levels, along with temporal convolutional networks to capture more comprehensive spatial-temporal dependencies. We conducted comprehensive experiments on four benchmark datasets, and experimental results demonstrate that our model outperforms the state-of-the-art performance.

Keywords: Multivariate time series forecasting, Multi-level relational learning, Graph neural network, Hypergraph neural network

1. Introduction

Time series forecasting involves inferring future trends and patterns by analyzing past observational data. Accurate predictions of time series play a crucial role in decision-making and planning across various societal domains. Multivariate Time Series (MTS) forecasting is a significant subtask in time series forecasting Bai et al. (2020); Frigola-Alcalde (2016). MTS focuses on the evolution and interrelationships of multiple variables over time, rendering it more challenging to forecast future trends.

In the early stage, traditional statistical methods Frigola (2015); Frigola-Alcalde (2016); Roberts et al. (2013), such as Vector AutoRegressive and Gaussian Processes, were widely

* Corresponding author.

employed for MTS tasks. These methods were used to learn linear dependencies at different time points to forecast future trends. However, they heavily depend on predefined assumptions and may fail to capture nonlinear patterns. With the advent of deep neural networks, there has been a shift towards leveraging their powerful modeling capabilities in handling non-stationary and nonlinear dependencies for MTS forecasting. Various approaches have emerged, including Convolutional Neural Networks (CNNs) that utilize local receptive fields Li et al. (2021), Recurrent Neural Networks (RNNs) for sequence modeling Lai et al. (2018); Shi et al. (2015), and Transformer models employing global attention mechanisms Huang et al. (2019); Shih et al. (2019). While these mechanisms excel in in identifying dependencies among features over lengthy durations, they predominantly disregard the essential influence of spatial information in MTS forecasting endeavors.

Recently, Graph Neural Networks (GNNs) have gained widespread attention for MTS tasks, due to their ability to model spatial relationships among multivariate variables. In structured MTS data, predefined graph structures have been utilized to simulate the spatial relationships between variables Li et al. (2018); Yu et al. (2018). The local spatial dependencies among variables were then learned through the neighborhood information aggregation mechanism of graph convolution networks. Furthermore, to capture the spatial dependencies among variables that require global dependency, the global attention mechanism is integrated with graph convolution Guo et al. (2019); Song et al. (2020). For unstructured MTS data, constructing predefined graphs is challenging. Therefore, models typically employ adaptive learning strategies to capture the spatial dependencies among variables Chen et al. (2023); Cai et al. (2024); Wu et al. (2020); Ye et al. (2022, 2021). Although these methods can model spatial relationships through predefined or adaptively learned adjacency matrices, their ability to capture multiple relational patterns and complex spatial dependencies is often limited Zhang et al. (2019).

In real-world MTS data, variables exhibit various associative patterns and display high-level interactions Shang and Chen (2024). Figure. 1 shows the normalized stock price trends of four NASDAQ 100 index component companies from November 2016 to December 2016. It's worth noting that technology companies such as Apple, Amazon, and Facebook consistently share a similar trend, contrasting sharply with the performance of the catering company Starbucks. This difference illustrates that there are not only directly pairwise low-level relationships between variables but also may indicate the presence of other abstract higher-level relationship patterns (e.g. relationships among technology companies). However, existing graph-based methods fail to fully capture these relationship patterns as conventional graph structure adheres to a message-passing paradigm between pairwise, ignoring other abstract and semantic higher-level relationship patterns, leading to incomplete modeling of relationships between variables (*Limitation 1*). Meanwhile, when learning spatial dependencies in MTS data, existing graph convolutional networks either only propagate low-level correlations between nodes, or only use mined high-level information through hypergraphs which ignores the information sharing and interaction between multiple levels. (*Limitation 2*).

In this paper, we propose a **M**ulti-level Relational Learning with **S**ynergistic **G**raphs (**MSG**) framework to address the aforementioned limitations. Specifically, for *Limitation 1*, we introduce a synergistic graph learning framework that combines the modeling capabilities of both dynamic graphs and hypergraphs. This framework adaptively models low-level

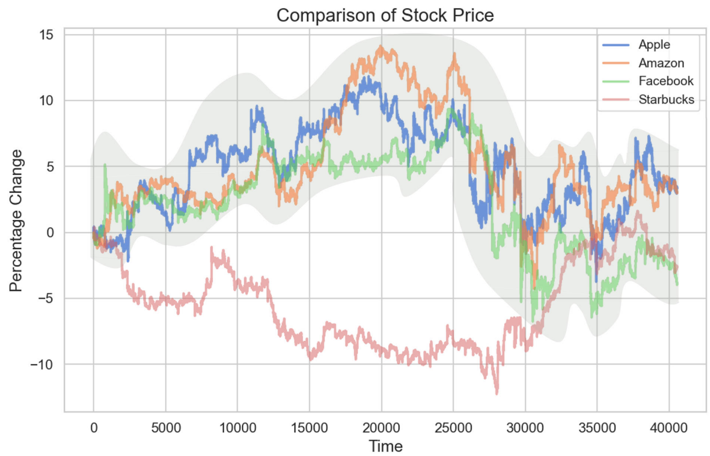

Figure 1: An illustration of non-pairwise relationship in stock price. The stock price trends of the three technology companies, Apple, Amazon, and Facebook exhibit a somewhat similar trend, distinctly differing from the trend of the catering company Starbucks. These differences may indicate the presence of other high-level beyond pairwise relationship patterns among the multivariate variables.

pairwise relationships through graphs and facilitates shared high-level connections among non-pairwise variables through hypergraphs to model multi-level relationship patterns. Furthermore, to address *Limitation 2*, we present a novel dual-layer attribute-enhanced hypergraph convolution network. This approach leverages hyperedges from the incidence matrices to represent high-order attribute information of variables, enabling attribute-layer hypergraph convolution to capture high-level spatial dependencies. By employing a dual-layer network, we enhance the interaction between high-level attribute features and low-level original features through forward and backward propagation of spatial information utilizing incidence matrices. In summary, our primary contributions are outlined as follows:

- We propose a synergistic graph learning framework, encompassing both low-level pairwise relationships and high-level beyond pairwise relationships through dynamic graphs and hypergraphs to unravel multi-level relationship patterns among variables in MTS forecasting.

- We design a novel dual-layer attribute-enhanced hypergraph convolutional network that effectively captures high-level spatial dependencies among variables at the attribute layer and facilitates multi-level information sharing through a dual-layer information propagation mechanism.

- We conducted extensive experiments on four real-world MTS datasets with no prior graph structure. The results illustrate that our MSG outperforms state-of-the-art methods.

2. Related Work

2.1. MTS Forecasting.

Multivariate time series forecasting predicts future points using multiple variables. Traditional methods like ARIMA struggle with complexity, while Bayesian methods like Gaussian Processes Frigola (2015) face nonlinearities. Deep learning methods, HyDCNN Li et al. (2021) employ CNNs to capture locally correlated feature dependencies. LSTNet and FC-LSTM Lai et al. (2018); Shi et al. (2015) utilize RNNs to capture temporal dependencies. TPA-LSTM Shih et al. (2019) and DSANet Huang et al. (2019) enhance long-term dependencies with attention mechanisms, exploring correlations across different time periods. However, dismissing the internal relationships among variables poses challenges when validating real MTS data.

2.2. Graphs for MTS Forecasting.

Graph Neural Networks (GNNs) Kipf and Welling (2017) improve spatial relationships between variables using weighted edges. In MTS tasks, pioneering works like STGNN and DCRNN Yu et al. (2018); Li et al. (2018) utilize graph structures to simulate non-Euclidean distances between variables. ASTGCN Guo et al. (2019) combines attention mechanisms and spatial-temporal Graph Convolutional Networks (GCNs) to capture spatial-temporal features. STSGCN Song et al. (2020) addresses spatial-temporal data heterogeneity by aggregating long-range dependencies. GCRNN and FourierGNN Ye et al. (2021); Yi et al. (2024) optimize static graph structures for learning variable relationships. TPGNN Liu et al. (2022) establishes dynamic graph structures among variables using static matrix basis and time-varying coefficients. MAGNN and ESG Chen et al. (2023); Ye et al. (2022) propose dynamic graph learning networks for MTS data. However, these methods primarily model low-level pairwise relationships, ignoring high-level abstract relationships.

2.3. Hypergraphs for MTS Forecasting.

A hypergraph extends the concept of a graph by allowing edges to connect multiple nodes, providing a more comprehensive representation of relationships. It's effective for illustrating complex associations and higher-level relationships, promising improved modeling in MTS forecasting. The Hypergraph Neural Networks (HGNNs) Feng et al. (2019) pioneered the use of hypergraph neural networks to capture high-level relationships in MTS forecasting. Another approach involves adjusting feature embeddings Jiang et al. (2019). Techniques enhancing hypergraph features include HCRNNs Yi and Park (2020) for structured sensor networks, spatial-temporal synchronization Wang and Zhu (2022) for traffic flow, Li et al. (2022) merging hypergraph and graph neural structures for crime rate prediction, Ma et al. (2022) using fuzzy clustering for stock price trends, and Xu et al. (2022) introducing trainable multi-scale hypergraphs for collective behavior trajectories.

While HGCNs have found applications in scenarios like traffic flow and crime rate prediction, their predictive performance on MTS data remains suboptimal due to the challenge of capturing complex multi-level relationship interactions among variables. Models like ReMo Wu et al. (2023) utilize hypergraphs to handle unstructured MTS data, representing variables and relationships as nodes and hyperedges. However, relying solely on hypergraphs may not suffice for some datasets, as they struggle to capture multi-level relationship patterns effectively. In contrast to prior approaches, we propose a novel MSG designed to enhance the model's ability to capture diverse correlation patterns in unstructured MTS data. MSG captures low-level pairwise and high-level beyond pairwise relationships through adaptively synergistic graph learning framework and introduces a dual-layer propagation learning mechanism for multi-level interactive sharing.

3. Preliminary

3.1. Problem Formulation.

In this paper, we focus on multivariate time series forecasting. Formally, given a sequence of observed time series signals $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_t, \cdots, \boldsymbol{x}_T\}$, where $\boldsymbol{x}_t \in \mathbb{R}^{N \times 1}$ denotes the values at time step t, N is the number of variable, $\boldsymbol{x}_{i,t}$ denotes the ith variable at time step t, our aim are to forecasting the future value $\widehat{\boldsymbol{x}}_{t+H} \in \mathbb{R}^{N \times 1}$ at single time step $t + H$ based on the historical values of the previous L time steps, where H denotes look-ahead horizon. The problem can be formulated as:

$$\boldsymbol{x}_{t-L+1:t} \xrightarrow{\mathcal{F}} \widehat{\boldsymbol{x}}_{t+H} \tag{1}$$

where $\mathcal{F}$ denotes the mapping function that the model intends to parameterize for single-step forecasting.

3.2. Graph & Hypergraph.

A graph is represented by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of nodes, $\mathcal{E}$ is the set of edges. The adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ describes the relationship between pairs of nodes. We use N to denote the number of nodes in a graph. A hypergraph can be denoted as $\mathcal{HG} = (\mathcal{HV}, \mathcal{HE})$ where $\mathcal{HV}$ denotes the node set and $\mathcal{HE} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_k\}$ denotes the hyperedge set. Hypergraphs allow a hyperedge to connect multiple nodes, which differs from a graph structure. We utilize an incidence matrix $\boldsymbol{I} \in \mathbb{R}^{N \times K}$ to explain the structure of a hypergraph. Hypergraph adjacency matrix $\boldsymbol{A}_H \in \mathbb{R}^{K \times K}$ aims to characterize the relationship between hyperedges. N and K represent the number of nodes and hyperedges, respectively.

4. Methodology

In this section, we present our model MSG and its main part. The overall framework is shown in Fig. 2. MSG consists of three parts: the multi-scale feature extraction module to capture different temporal and spatial characteristics across scale in Fig. 2(a), the **S**ynergistic **R**elationship **C**onstruction (SRC) module to model multiple relationship patterns in Fig. 2(b), and the **S**ynergistic **R**elationship **L**earning (SRL) module to learn intricate spatial-temporal dependencies in Fig. 2(c). A more detailed description is as follows:

4.1. Multi-Scale Feature Extraction

We constructed a multi-scale feature extraction network Chen et al. (2023) to transform the original time series into feature representations of different scales. It extracts details

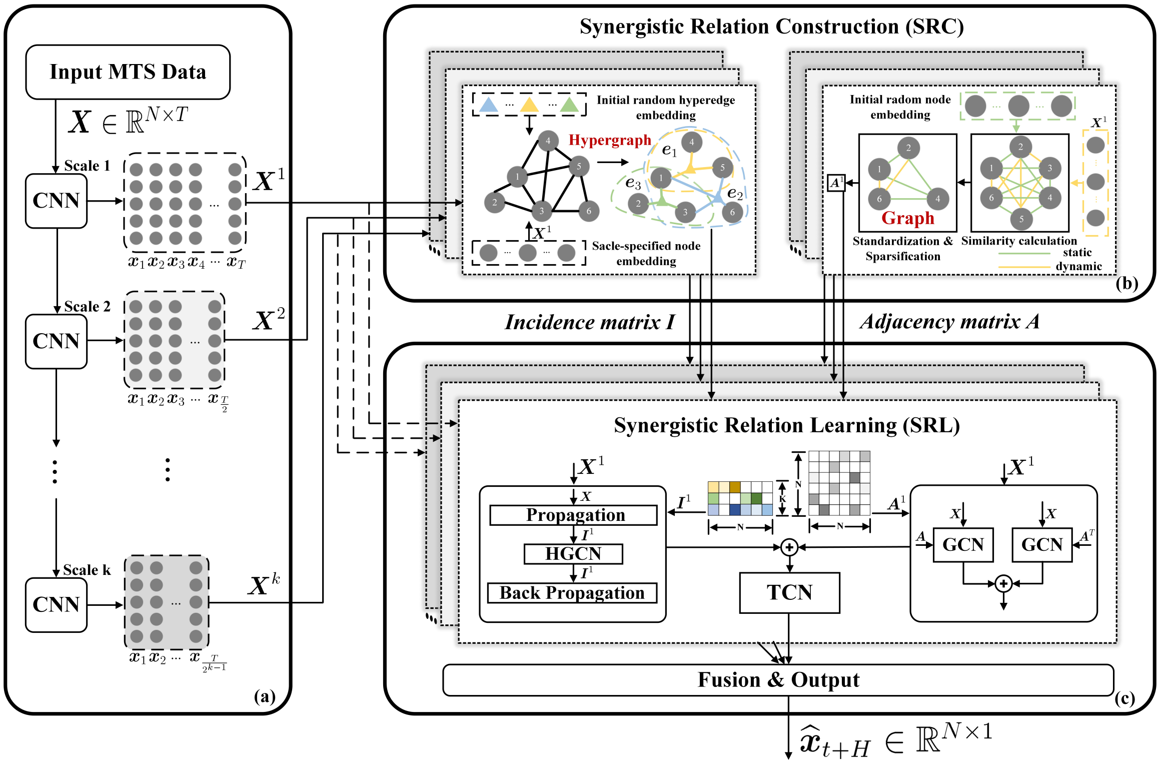

Figure 2: The overall framework of MSG.

from small-scale features and captures slow-changing trends from large-scale features. As shown in Fig. 2(a), firstly, the original MTS input undergoes sequential CNN operations in the temporal dimension. Each layer generates feature representations at multiple scales. The feature representation at the kth scale, is defined as $\boldsymbol{X}^k \in \mathbb{R}^{N\times \frac{T}{2^{k-1}}\times F^k}$, where N represent the number of variables, $\frac{T}{2^{k-1}}$ is the sequence length in kth dimension and F^k is the feature dimension of the kth scale. Then, We use two parallel convolutional networks to add features of a certain scale. One use kernel sizes 1×7, 1×6, and 1×3 at each layer, the stride is set to 2; the other with kernel size 1×1 and a 1×2 pooling layer, formally:

$$\boldsymbol{X}_1^k = \mathrm{ReLU}\left(\theta_{\mathrm{con1}}^k * \boldsymbol{X}^{k-1} + \boldsymbol{b}_1^k\right) \tag{2}$$

$$\boldsymbol{X}_2^k = \mathrm{Pooling}\left(\mathrm{ReLU}\left(\theta_{\mathrm{con2}}^k * \boldsymbol{X}^{k-1} + \boldsymbol{b}_2^k\right)\right) \tag{3}$$

where θ_{con1}^k and θ_{con2}^k are different convolution kernels, $\boldsymbol{b}_2^k$, $\boldsymbol{b}_2^k$ are biases. Finally, we compute the feature representations for each scale $\boldsymbol{X}^k$:

$$\boldsymbol{X}^k = \boldsymbol{X}_1^k + \boldsymbol{X}_2^k \tag{4}$$

4.2. Synergistic Relationship Construction

In this part, we propose a Synergistic Relationship Construction module utilizing graph and hypergraph structures for capturing multi-level relationship patterns between variables, and this structure eliminates the need for pre-defined graphs. The details is shown in Fig. 2(b).

Graph Structure Construction. For the graph structures, we design a parameter overlay involving the combination of data-driven and parameter-driven strategies to adaptively model dynamic relationships shown in Fig. 3. Firstly, The static graphs learned in a parameter-driven manner are adopted in Wu et al. (2020) to acquire the inter-node distance relationships. The formulas are as follows:

$$\boldsymbol{M}_{s1}^{k} = \tanh\left(\boldsymbol{E}_{\text{rand}}^{k}\theta_{s1}^{k}\right), \ \ \boldsymbol{M}_{s2}^{k} = \tanh\left(\boldsymbol{E}_{\text{rand}}^{k}\theta_{s2}^{k}\right) \tag{5}$$

$$\boldsymbol{A}_{s}^{k} = \text{ReLU}\left(\tanh\left(\boldsymbol{M}_{s1}^{k}\left(\boldsymbol{M}_{s2}^{k}\right)^{T} - \boldsymbol{M}_{s2}^{k}\left(\boldsymbol{M}_{s1}^{k}\right)^{T}\right)\right) \tag{6}$$

where $\boldsymbol{E}_{\text{rand}}^{k}$ represents randomly initialized node embeddings at kth scale, θ_{s1}^{k} and θ_{s2}^{k} are learnable parameter during training, tanh and ReLU are the activate function. Static representation $\boldsymbol{M}_{s1}^{k}$ and $\boldsymbol{M}_{s2}^{k}$ were used to get the adjacency matrix, which represents the static low-level pairwise relationships at kth scale.

Subsequently, we design a data-driven strategy to learn dynamic low-level pairwise patterns as the limited parameters $\boldsymbol{M}_{s}^{k}$ can't grow infinitely to model dynamic features that change constantly. Therefore, We treat the data itself as dynamic features at each timestep and use an MLP to map the original MTS data $\boldsymbol{X}^{k} \in \mathbb{R}^{N\times T}$ to the dynamic information $\boldsymbol{U}^{k} \in \mathbb{R}^{F\times F}$. And $\boldsymbol{U}^{k}$ is then merged with the dynamic representation $\boldsymbol{M}_{d1}^{k}$ and $\boldsymbol{M}_{d2}^{k}$, thereby learning the dynamic adjacency matrix $\boldsymbol{A}_{d}^{k} \in \mathbb{R}^{N\times N}$ at kth scale.

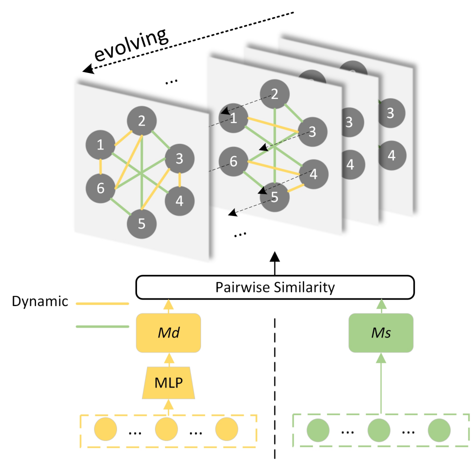

Figure 3: The details parameter overlay involving the combination of data-driven and parameter-driven strategy. We add static and dynamic pairwise relationships to represent low-level relational patterns.

$$\boldsymbol{U}^{k} = \text{ReLU}\left(\theta_{2}^{k}\left(\text{ReLU}\left(\theta_{1}^{k}\boldsymbol{X}^{k}\right)\right)\right) \tag{7}$$

$$\boldsymbol{M}_{d1}^{k} = \tanh\left(\boldsymbol{E}_{\text{rand}}^{k}\theta_{d1}^{k}\boldsymbol{U}^{k}\right), \ \ \boldsymbol{M}_{d2}^{k} = \tanh\left(\boldsymbol{E}_{\text{rand}}^{k}\theta_{d2}^{k}\boldsymbol{U}^{k}\right) \tag{8}$$

$$\boldsymbol{A}_{d}^{k} = \text{ReLU}\left(\tanh\left(\boldsymbol{M}_{d1}^{k}\left(\boldsymbol{M}_{d2}^{k}\right)^{T} - \boldsymbol{M}_{d2}^{k}\left(\boldsymbol{M}_{d1}^{k}\right)^{T}\right)\right) \tag{9}$$

By combining the static and dynamic adjacency matrices, we derive the final adjacency matrix $\boldsymbol{A}^{k}$ at the kth scale, effectively capturing static and dynamic pairwise relationships patterns $\mathcal{A} = \left\{\boldsymbol{A}^{1}, \boldsymbol{A}^{2}, \ldots, \boldsymbol{A}^{k}\right\}$ across multiple scales.

$$\boldsymbol{A}^{k} = \boldsymbol{A}_{s}^{k} + \boldsymbol{A}_{d}^{k} \tag{10}$$

Hypergraph Structure Construction. While the adjacency matrix can only capture pairwise relationships between variables, we introduce hypergraph structures to establish high-level beyond pairwise relational patterns. Specifically, we randomly initialize hyperedge (which can also be cluster) $\boldsymbol{E}_{\text{cluster}}^{k}$ at kth scale multiple with φ_{2}^{k} to get hyperedge embedding

$\boldsymbol{P}_2^k \in \mathbb{R}^{K \times F}$. Then we multiply $\boldsymbol{P}_1^k$ and the transpose of $\boldsymbol{P}_2^k$ to get the incidence matrix $\boldsymbol{I}^k \in \mathbb{R}^{N \times K}$, where N is the number of variables, K is the number of hyperedges. The incidence matrix at kth scale can be expressed as:

$$\boldsymbol{P}_1^k = \tanh\left(\boldsymbol{X}^k \varphi_1^k\right), \;\; \boldsymbol{P}_2^k = \tanh\left(\boldsymbol{E}_{\text{cluster}}^k \varphi_2^k\right) \tag{11}$$

$$\boldsymbol{I}^k = \text{ReLU}\left(\tanh\left(\boldsymbol{P}_1^k \left(\boldsymbol{P}_2^k\right)^T\right)\right) \tag{12}$$

φ_1^k and φ_2^k are trainable parameters at kth scale. Similarly, $\mathcal{I} = \left\{\boldsymbol{I}^1, \boldsymbol{I}^2, \ldots, \boldsymbol{I}^k\right\}$ contains the set of all incidence matrices, which can express high-level beyond pairwise relationships at multiple scales among variables.

The multi-scale adjacency matrix set $\mathcal{A}$ and incidence matrix set $\mathcal{I}$ were sent to subsequent SRL modules to learn spatial-temporal dependencies.

4.3. Synergistic Relationship Learning

The SRL mechanism, shown in Fig. 2(c), leverages novel synergistic graph and hypergraph convolutional networks to facilitate the interaction of spatial dependencies across multi-levels, along with temporal convolutional networks to capture more comprehensive spatial-temporal dependencies.

Graph Convolutional Networks. In the spatial dependencies learning phase, firstly, we use bidirectional graph convolution networks to learn the incoming and outgoing information of a node.

$$\boldsymbol{H}_G^k = GNN_{\text{in}}^k\left(\boldsymbol{X}^k, \boldsymbol{A}^k, \theta_{\text{in}}^k\right) + GNN_{\text{out}}^k\left(\boldsymbol{X}^k, \left(\boldsymbol{A}^k\right)^T, \theta_{\text{out}}^k\right) \tag{13}$$

where θ_{in}^k and θ_{out}^k are parameters for incoming and outgoing, $\boldsymbol{H}_G^k \in \mathbb{R}^{N \times T \times F}$ is the spatial feature coming from bidirectional graph convolution at kth scale.

Dual-layer Attribute-enhanced Hypergraph Convolutional Networks. Unlike conventional hypergraph convolutional networks, this network has a dual-layer structure, allowing for cross-layer interaction and learning between node layer features and attribute layer features, thereby enhancing the spatial dependency feature association of multi-levels. As shown in Fig. 4, we first propagate the original nodes to hyperedges (represented by node e), which are also considered as cluster centers with common attributes of multiple nodes. Secondly, we learn the high-level relationships of nodes in the attribute layer by updating the $\boldsymbol{A}_H$ matrix. Finally, we construct the incidence matrix to associate hyperedges with original nodes, establishing non-linear relationships between the attribute layer and nodes. For example, the red arrow in Fig. 4 indicates the spatial dependence from node 5

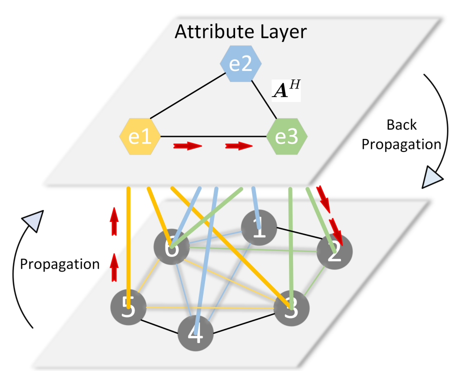

Figure 4: The details of the dual-layer attribute-enhanced propagation.

to node 2. In traditional GCNs, node 5 and node 2 might not have a direct correlation. However, by exploring the spatial dependencies between the attribute layer and nodes, node 5 and node 2 can be connected through path node 5 to attribute 1 to attribute 3 to node 2.

To get the hyperedge embedding matrix $\boldsymbol{S}^k$, we multiply the incidence matrix and embeddings of input at the kth scale. And add an additional adjacency learnable matrix $\boldsymbol{A}_H^k \in \mathbb{R}^{K \times K}$ to extract the high-level spatial dependencies at attribute layer between hyperedges. The formula can be formulated as:

$$\boldsymbol{S}^k = \phi\left(\boldsymbol{A}_H^k \boldsymbol{I}^k \boldsymbol{X}^k\right) + \boldsymbol{I}^k \boldsymbol{X}^k \tag{14}$$

where $\boldsymbol{S}^k \in \mathbb{R}^{K \times F}$ represents the embedding representation of K hyperedges, each hyperedge dimension is F, $\boldsymbol{I}^k$ is the incidence matrix at kth scale, $\boldsymbol{X}^k$ represents embedding scales at kth layer. Further, we aggregated these hyperedges to generate original node embedding $\boldsymbol{H}_{HG}^k \in \mathbb{R}^{N \times T \times F}$ at kth scale. The formula is shown in Eq. (15).

$$\boldsymbol{H}_{HG}^k = \boldsymbol{I}^k \boldsymbol{S}_l^k = \boldsymbol{I}^k \left(\phi\left(\boldsymbol{A}_H^k \boldsymbol{I}^k \boldsymbol{X}^k\right) + \boldsymbol{I}^k \boldsymbol{X}^k\right) \tag{15}$$

Fusion GCNs & HGCNs. We add the two types of spatial features together, enabling the model to learn a more comprehensive representation of spatial dependencies. β is the hyper-parameter.

$$\boldsymbol{H}^k = \beta \boldsymbol{H}_G^k + (1-\beta) \boldsymbol{H}_{HG}^k \tag{16}$$

Temporal Convolutional Networks. After learning the spatial dependencies, we offer temporal convolutional networks (TCNs) to capture the temporal dependencies. We feed the $\boldsymbol{H}^k$ obtained in the previous layer into a temporal convolution layer to obtain spatial-temporal representation $\boldsymbol{H}_{full}^k$ at each scale.

$$\boldsymbol{H}_{full}^k = \theta_{tcn}^k * \boldsymbol{H}^k \tag{17}$$

We perform a convolution operation on $\boldsymbol{H}_{full}^k \in \mathbb{R}^{N \times F}$ in the time dimension. θ_{tcn}^k denotes the different convolution kernels in different scales.

4.4. Fusion & Output

In this section, we first concatenate feature representations from multiple scales to obtain $\boldsymbol{H} \in \mathbb{R}^{k \times N \times F}$. Subsequently, we perform average pooling across the scale dimension to achieve $\text{Pooling}(\boldsymbol{H}) \in \mathbb{R}^{N \times F}$.And we utilize an MLP layer with a kernel size of $1 \times F$, and a CNN with a 1×1 kernel size to transform $\text{Pooling}(\boldsymbol{H})$ into the required dimension, resulting in the final prediction $\widehat{\boldsymbol{x}}_{\boldsymbol{t+H}} \in \mathbb{R}^{N \times 1}$. Finally, the output $\widehat{\boldsymbol{x}}_{\boldsymbol{t+H}}$ is compared with the ground truth labels, and parameters are iteratively learned through a back propagation gradient update mechanism. The formulas are as follows:

$$\begin{gathered} \boldsymbol{H} = \text{Concat}\left(\boldsymbol{H}_{full}^1, \boldsymbol{H}_{full}^2, \ldots, \boldsymbol{H}_{full}^k\right) \\ \widehat{\boldsymbol{x}}_{\boldsymbol{t+H}} = \text{MLP}\left(\text{Pooling}\left(\boldsymbol{H}\right)\right) \end{gathered} \tag{18}$$

where Concat denotes the concatenation operation. We use the Mean Square Error (MSE) loss function to achieve convergence of the model.

5. Experiments

5.1. Experimental Setup

Datasets. We evaluate the model performance on four benchmark datasets. Details of the datasets can be found in the footnotes.

- Exchange-Rate[1]: The dataset comprises the daily sampled exchange rates of eight countries spanning from 1990 to 2016.
- Electricity[2]: The hourly electricity consumption in kWh recorded from 2012 to 2014, involving 321 clients.
- Traffic[3]: A dataset comprising 48 months of hourly data from 2015 to 2016, obtained from the California Department of Transportation. The dataset delineates road occupancy rates.
- Nasdaq[4]: A subset of the full NASDAQ 100 stock dataset. It includes 105 days of stock data starting from July 26, 2016 to December 22, 2016.

Following existing works Chen et al. (2023); Guo et al. (2019); Wu et al. (2019, 2020), we split the four datasets into the training set (60%), test set (20%) and validation set (20%) in chronological order. The source code is available in our repository[5].

Experimental Settings. The model is implemented in Python with PyTorch 1.9.0 and trained on an NVIDIA Tesla A100 GPU. Following the existing work Wu et al. (2020, 2019), the input window size T is set to 168, The learning rate is set to 0.0001, and we use Adam optimizer to propagate all the trainable parameters. We aim to forecast at single-step intervals for the next 3, 6, 12, and 24 steps, denoted as the horizon $h = (3, 6, 12, 24)$.

Evaluation Matrix. We evaluate the model's performance using the Root Relative Squared Error (RSE) and the Empirical Correlation Coefficient (CORR). The formulas are defined as follows:

$$\begin{aligned} \mathrm{RSE} &= \frac{\sqrt{\sum_{i=1}^{n} \left(\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i\right)^2}}{\sqrt{\sum_{i=1}^{n} \left(\boldsymbol{x}_i - \bar{\boldsymbol{x}}\right)^2}} \\ \mathrm{CORR} &= \frac{\sum_{i=1}^{n} \left(\boldsymbol{x}_i - \bar{\boldsymbol{x}}\right)\left(\hat{\boldsymbol{x}}_i - \bar{\hat{\boldsymbol{x}}}\right)}{\sqrt{\sum_{i=1}^{n} \left(\boldsymbol{x}_i - \bar{\boldsymbol{x}}\right)^2}\sqrt{\sum_{i=1}^{n} \left(\hat{\boldsymbol{x}}_i - \bar{\hat{\boldsymbol{x}}}\right)^2}} \end{aligned} \tag{19}$$

Baselines. We have chosen eight competitive spatial-temporal methods, namely TPA-LSTM Shih et al. (2019), AGCRN Bai et al. (2020), Graph WaveNet Wu et al. (2019), MTGNN Wu et al. (2020), MTHetGNN Wang et al. (2022), ESG Ye et al. (2022), MAGNN Chen et al. (2023), and ReMo Wu et al. (2023). TPA-LSTM uses an attention-based RNN.

1. https://github.com/laiguokun/multivariate-time-series-data
2. https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014
3. http://pems.dot.ca.gov
4. https://cseweb.ucsd.edu/yaq007/NASDAQ100_stock_data
5. https://github.com/YoungXiee/MSG

Table 1: Baseline comparison for multivariate time series methods.

Dataset		Exchange-Rate				Traffic				Electricity				Nasdaq			
		Horizon				Horizon				Horizon				Horizon			
Methods	Metrics	3	6	12	24	3	6	12	24	3	6	12	24	3	6	12	24
TPA-LSTM	RSE	0.0174	0.0241	0.0341	0.0444	0.4487	0.4658	0.4641	0.4765	0.0823	0.0916	0.0964	0.1006	0.0012	0.0013	0.0018	0.0024
	CORR	0.9790	0.9709	0.9564	0.9381	0.8812	0.8717	0.8717	0.8629	0.9439	0.9337	0.9250	0.9133	0.9745	0.9707	0.9705	0.9627
AGCRN	RSE	0.0269	0.0331	0.0374	0.0476	0.4379	0.4635	0.4694	0.4707	0.0766	0.0894	0.0921	0.0967	0.0012	0.0018	0.0022	0.0023
	CORR	0.9717	0.9615	0.9531	0.9334	0.8850	0.8670	0.8679	0.8664	0.9408	0.9309	0.9222	0.9183	0.9878	0.9877	0.9816	0.9701
Graph WaveNet	RSE	0.0251	0.0300	0.0381	0.0486	0.4484	0.4689	0.4725	0.4741	0.0746	0.0922	0.0909	0.0962	0.0018	0.0022	0.0023	0.0026
	CORR	0.9740	0.9640	0.9510	0.9294	0.8801	0.8674	0.8646	.8646	0.9459	0.9310	0.9267	0.9226	0.9953	0.9943	0.9840	0.9834
MTGNN	RSE	0.0194	0.0259	0.0349	0.0456	0.4162	0.4754	0.4461	0.4535	0.0745	0.0878	0.0916	0.0953	0.0015	0.0018	0.0020	0.0026
	CORR	0.9786	0.9708	0.9551	0.9372	0.8963	0.8667	0.8794	0.8810	0.9474	0.9316	0.9278	0.9234	0.9912	0.9876	0.9834	0.9754
MTHetGNN	RSE	0.0198	0.0259	0.0345	0.0451	0.4826	0.5198	0.5147	0.5250	0.0749	0.0892	0.0959	0.0969	0.0016	0.0026	0.0020	0.0030
	CORR	0.9769	0.9701	0.9539	0.9360	0.8643	0.8452	0.8744	0.8418	0.9456	0.9307	0.8783	0.8782	0.9919	0.9897	0.9849	0.9799
ESG	RSE	0.0181	0.0246	0.0345	0.0468	0.4235	0.4685	0.4508	0.4746	**0.0718**	0.0844	0.0898	0.0962	0.0015	0.0028	0.0024	0.0031
	CORR	0.9792	0.9717	0.9564	0.9392	0.8943	0.8692	0.8741	0.8656	0.9494	0.9372	**0.9321**	**0.9279**	0.9901	0.9887	0.9830	0.9705
MAGNN	RSE	0.0183	0.0246	0.0343	0.0474	**0.4097**	0.4555	0.4423	**0.4434**	0.0745	0.0876	0.0908	0.0963	0.0010	0.0011	0.0018	0.0020
	CORR	0.9778	0.9712	0.9557	0.9399	**0.8992**	0.8753	0.8815	0.8813	0.9476	0.9323	0.9282	0.9217	0.9975	0.9951	0.9864	0.9846
ReMo	RSE	**0.0173**	**0.0240**	0.0340	0.0443	0.4414	0.4380	0.4634	0.4645	0.0779	0.0879	0.0949	0.1011	0.0015	0.0016	0.0017	0.0025
	CORR	0.9753	0.9680	0.9565	0.9373	0.8750	0.8839	0.8713	0.8703	0.9381	0.9227	0.9167	0.9058	0.9977	0.9953	0.9914	0.9838
MSG	RSE	0.0178	0.0251	**0.0335**	**0.0442**	0.4118	**0.4289**	**0.4403**	0.4463	0.0735	**0.0837**	**0.0887**	**0.0934**	**0.0008**	**0.0011**	**0.0015**	**0.0019**
	CORR	**0.9883**	**0.9828**	**0.9737**	**0.9609**	0.8978	**0.8890**	**0.8826**	**0.8815**	**0.9494**	**0.9373**	0.9263	0.9245	**0.9979**	**0.9958**	**0.9921**	**0.9850**

AGCRN introduces an adaptive GCRN. Graph WaveNet combines GCNs and CNNs. MTGNN includes a graph learning layer. MTHetGNN utilizes relation and temporal embeddings with a heterogeneous graph. ESG employs evolutionary graphs. MAGNN features a multi-scale pyramid structure. ReMo uses a multi-view hypergraph approach.

5.2. Experimental Results

Table 1 shows the experimental results of the proposed MSG. On the NASDAQ dataset, MSG achieved the best results across all horizons, with an average reduction in the RSE metric by 10.43%. On the exchange rate dataset, based on the thousandth percentile on the CORR metric, MSG outperformed the best baseline model by an average of 1.5 percentage points across all horizons, a performance improvement five times that of other models. On the traffic and electricity datasets, MSG also achieved the best results in 5 out of 8 horizons. Achieving good performance on datasets across various task contexts demonstrates the strong robustness of our model.

Through analysis, we found that MSG shows a notable improvement in performance on the NASDAQ and Exchange-Rate datasets for the following primary reasons: Compared to other datasets, the shared connections and high-level features in stock-type data are more significant and prevalent. Thus, the ability of MSG to model relationships at different levels is key to its performance improvement. Moreover, the enhancement in performance on other datasets also indicates that our model can adapt to the needs of multi-domain MTS prediction tasks, rather than being designed for a specific type of data.

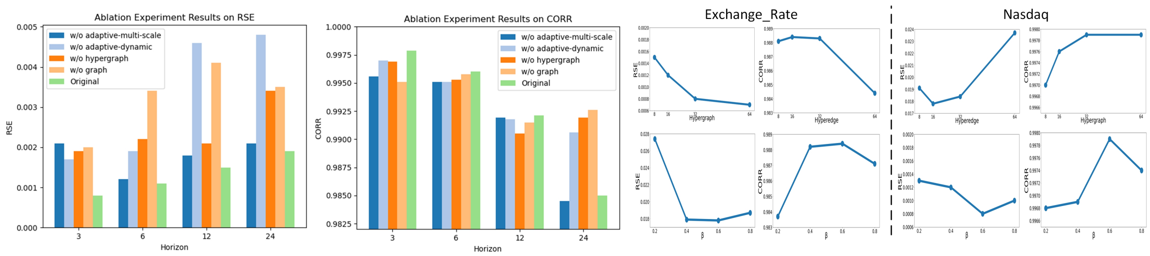

Figure 5: Ablation study on the left and Hyperparameter Study on the right.

5.3. Ablation Studies

We conducted extensive ablation experiments to evaluate each module's effectiveness in MSG. We removed the multi-scale and dynamic learning strategies to showcase our model's efficacy in capturing dynamic patterns. We also removed graphs and hypergraphs to validate the necessity of the proposed approach in capturing multi-level relational patterns. Ablation types are defined as follows:

- **w/o adaptive-multi-scale**: MSG without the multi-scale learning strategy.
- **w/o adaptive-dynamic**: MSG without the dynamic learning strategy.
- **w/o hypergraph**: synergistic graph without the driven of the hypergraph.
- **w/o graph**: synergistic graph without the driven of the graph.

Ablation experimental results on the Nasdaq dataset shown in in Fig. 5 left illustrate that the multi-scale learning strategy is critical to prediction accuracy, and its exclusion (dark blue bars) results in a 47% average accuracy drop across four levels. At the same time, the omission of the dynamic learning strategy (light blue bars) caused the model's average accuracy in RSE to drop significantly by 136.5%. This confirms the apparent complex relationships between variables at different time scales in MTS data and emphasizes the importance of multi-scale and dynamic strategies in capturing variable relationships. Furthermore, when we remove hypergraphs or graphs individually (indicated in orange and yellow bars), the model still retains the integrity of the remaining modules, but they only perform at the baseline level. Compared with the original MSG, the RSE accuracy dropped significantly by 154.1% and 88.9% respectively. This subtle finding emphasizes the importance of a balanced integration of hypergraphs and graphs, highlighting their complementary roles in capturing relationships at different levels.

We also study the hyperparameters of MSG focusing on the number of hyperedges and the weights combination of graphs and hypergraphs. The experimental results on Exchange-Rate and Nasdaq datasets are shown in Fig. 5 right. We adjusted the number of hyperedges within the range of 8, 16, 32, and 64 from the first row and observed their significant impact on performance, particularly when datasets have varying numbers of variables. For example, the exchange-rate dataset with 8 variables and the Nasdaq dataset with 82 variables showed optimal hyperedge sizes of 16 and 64, respectively. Setting an appropriate number of hyperedges is crucial for better results. The combination of graph and

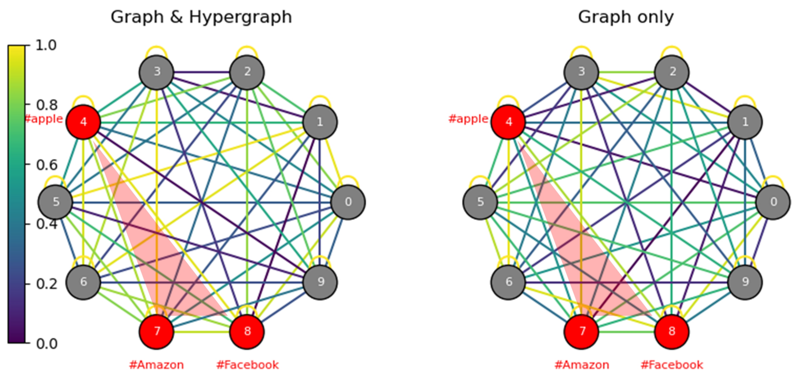

Figure 6: Case Study. We visualized the correlation graph in NASDAQ, stocks with IDs 4, 7, and 8 represent Apple, Amazon, and Facebook, respectively. The left panel utilizes synergistic graphs and hypergraphs, while the right panel only uses graph structures.

hypergraph weights, using β values from 0.2 to 0.8 in row two, revealed the equal importance of both components in synergistic graph learning. However, excessive dominance of either graphs or hypergraphs led to performance degradation. This highlights the significance of simultaneously modeling low-level and high-level relationships among variables for effective relation pattern mining.

5.4. Case Study

To provide additional support for the justification of our proposed synergistic graph in learning multiple patterns, we computed the average embeddings of ten stocks in the Nasdaq dataset at different scales and visualized the correlation graph among these stocks. Stock IDs 4, 7, and 8 represent Apple, Amazon, and Facebook, respectively. In Fig. 1, we have already demonstrated a common trend among these three stocks. However, as shown in the right panel of Fig. 6, the traditional graph structure clearly fails to capture this shared connection, where the relationship between Apple and Facebook is significantly stronger than that between Apple and Amazon. In contrast, as illustrated in the left panel, our proposed synergistic graph learning mechanism makes the relationships among Apple, Amazon, and Facebook closer and more consistent. This indicates that our proposed mechanism, which models and learns beyond pairwise relationships through the propagation of information via hyperedges, effectively captures and models this sharing connection among multiple variables.

6. Conclusion

In this paper, we present a novel MSG framework for multivariate time series forecasting. We proposed a synergistic graph mechanism that seamlessly integrates dynamic graphs and hypergraphs to adaptively capture multi-level relationships among variables. Additionally, we put forth a novel spatial learning strategy that combines synergistic graph and hypergraph convolutional and networks to effectively facilitate the interaction of spatial dependencies across multi-levels, along with temporal convolutional networks, which learns more comprehensive spatial-temporal dependencies. We conducted an intriguing experiment by integrating the capabilities of hypergraphs and graphs, and the model exhibited superiority on baseline datasets. Our study may offer valuable insights for modeling intricate relationships among variables in future MTS tasks.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful discussion and feedback. This work is supported by the National Natural Science Foundation of China (U2336204), The key development projects of the Sichuan Provincial Science and Technology Plan (2024YFG0005), and the National Natural Science Foundation of China (U22B2061).

References

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815, 2020.

Wanlin Cai, Yuxuan Liang, Xianggen Liu, Jianshuai Feng, and Yuankai Wu. Msgnet: Learning multi-scale inter-series correlations for multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11141–11149, 2024.

Ling Chen, Donghui Chen, Zongjiang Shang, Binqing Wu, Cen Zheng, Bo Wen, and Wei Zhang. Multi-scale adaptive graph neural network for multivariate time series forecasting. IEEE, 2023.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3558–3565, 2019.

Roger Frigola. *Bayesian time series learning with Gaussian processes*. PhD thesis, University of Cambridge, 2015.

Roger Frigola-Alcalde. *Bayesian time series learning with Gaussian processes*. PhD thesis, University of Cambridge, 2016.

Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2019.

Siteng Huang, Donglin Wang, Xuehan Wu, and Ao Tang. Dsanet: Dual self-attention network for multivariate time series forecasting. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 2129–2132, 2019.

Jianwen Jiang, Yuxuan Wei, Yifan Feng, Jingxuan Cao, and Yue Gao. Dynamic hypergraph neural networks. In *IJCAI*, pages 2635–2641, 2019.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 95–104, 2018.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018.

Yangfan Li, Kenli Li, Cen Chen, Xu Zhou, Zeng Zeng, and Keqin Li. Modeling temporal patterns with dilated convolutions for time-series forecasting. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(1):1–22, 2021.

Zhonghang Li, Chao Huang, Lianghao Xia, Yong Xu, and Jian Pei. Spatial-temporal hypergraph self-supervised learning for crime prediction. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 2984–2996. IEEE, 2022.

Yijing Liu, Qinxian Liu, Jian-Wei Zhang, Haozhe Feng, Zhongwei Wang, Zihan Zhou, and Wei Chen. Multivariate time-series forecasting with temporal polynomial graph neural networks. volume 35, pages 19414–19426, 2022.

Xiang Ma, Tianlong Zhao, Qiang Guo, Xuemei Li, and Caiming Zhang. Fuzzy hypergraph network for recommending top-k profitable stocks. *Information Sciences*, 613:239–255, 2022.

Stephen Roberts, Michael Osborne, Mark Ebden, Steven Reece, Neale Gibson, and Suzanne Aigrain. Gaussian processes for time-series modeling. *Philos. Trans. R. Soc. A*, 371(1984): 20110550, 2013.

Zongjiang Shang and Ling Chen. Mshyper: Multi-scale hypergraph transformer for long-range time series forecasting. *arXiv preprint arXiv:2401.09261*, 2024.

Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28, 2015.

Shun-Yao Shih, Fan-Keng Sun, and Hung-yi Lee. Temporal pattern attention for multivariate time series forecasting. *Machine Learning*, 108(8-9):1421–1441, 2019.

Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu 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, pages 914–921, 2020.

Yi Wang and Di Zhu. Shgcn: a hypergraph-based deep learning model for spatiotemporal traffic flow prediction. In *Proceedings of the 5th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, pages 30–39, 2022.

Yueyang Wang, Ziheng Duan, Yida Huang, Haoyan Xu, Jie Feng, and Anni Ren. Mthetgnn: A heterogeneous graph embedding framework for multivariate time series forecasting. *Pattern Recognition Letters*, 153:151–158, 2022.

Jinming Wu, Qi Qi, Jingyu Wang, Haifeng Sun, Zhikang Wu, Zirui Zhuang, and Jianxin Liao. Not only pairwise relationships: Fine-grained relational modeling for multivariate time series forecasting. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 4416–4423, 2023.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 2019.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi 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*, page 753–763, 2020.

Chenxin Xu, Maosen Li, Zhenyang Ni, Ya Zhang, and Siheng Chen. Groupnet: Multiscale hypergraph neural networks for trajectory prediction with relational reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, volume 2022, pages 6498–6507, 2022.

Junchen Ye, Leilei Sun, Bowen Du, Yanjie Fu, and Hui Xiong. Coupled layer-wise graph convolution for transportation demand prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4617–4625, 2021.

Junchen Ye, Zihan Liu, Bowen Du, Leilei Sun, Weimiao Li, Yanjie Fu, and Hui Xiong. Learning the evolutionary and multi-scale graph structure for multivariate time series forecasting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2296–2306, 2022.

Jaehyuk Yi and Jinkyoo Park. Hypergraph convolutional recurrent neural network. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3366–3376, 2020.

Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhendong Niu. Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective. *Advances in Neural Information Processing Systems*, 36, 2024.

Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 2018.

Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. Graph convolutional networks: a comprehensive review. *Computational Social Networks*, 6(1):1–23, 2019.