GinAR+: A Robust End-To-End Framework for Multivariate Time Series Forecasting with Missing Values

Chengqing Yu, Fei Wang, Zezhi Shao, Tangwen Qian, Zhao Zhang, Wei Wei, Zhulin An, Qi Wang and Yongjun Xu

***Abstract*—Spatial-Temporal Graph Neural Networks (STGNNs) have been widely utilized in multivariate time series forecasting (MTSF), but they rely on the assumption of data completeness. In practice, due to factors such as natural disaster, STGNNs frequently encounter the challenge of missing data resulting from numerous malfunctioning data collectors. In this case, on the one hand, due to the presence of missing values, STGNNs easily generate incorrect spatial correlations, leading to the performance degradation. On the other hand, STGNNs require separate training of models for different missing rates, limiting their robustness. To address these challenges, we first propose two important components (interpolation attention and adaptive graph convolution), which utilize normal values to recover missing values into reliable representations and reconstruct spatial correlations. Then, we replace the fully connected layers in simple recursive units with these two components and propose Graph Interpolation Attention Recursive Network (GinAR), aiming to recursively correct spatial correlations and achieve end-to-end MTSF with missing values. Finally, we use data with different missing rates as positive and negative data pairs. By employing contrastive learning to train GinAR, we propose GinAR+ and enhance its robustness to data with different missing rates. Experiments validate the superiority of GinAR+ and our motivation.**

***Index Terms*—Contrastive learning, Graph interpolation attention recursive network, Multivariate Time Series Forecasting with Missing Values.**

I. Introduction

Multivariate time series forecasting (MTSF) is extensively utilized across various domains such as transportation, environmental, energy, and others [1]–[5]. It predicts future values of intricately linked time series based on their historical data, significantly aiding decision-making processes [6]–[8]. Multivariate time series (MTS) can be conceptualized as a classical form of spatial-temporal graph data, exemplified by traffic flow, where each variable is chronologically collected through sensors positioned independently [9]. MTS exhibit two key patterns: temporal dependency [10] and spatial correlation [11]. Temporal dependency delineates complex chronological patterns, such as causal relationships and periodicity, while spatial correlation reveals how time series interact across spatial dimensions [12]. Consequently, effectively mining these spatial-temporal dependencies is important for MTSF to not only precisely predict future values but also to enhance our understanding of their interplay [13]–[16].

Spatial-Temporal Graph Neural Networks (STGNNs) combine the sequence model and graph convolution (GCN) to capture spatial-temporal dependencies of MTS and achieve significant progress in MTSF [17], but their superior performances heavily rely on the data quantity [18]. In other words, they need to make full use of historical observations of all variables to achieve accurate forecasting [19]. In reality, due to factors such as natural disasters and component failures, data collectors can easily malfunction and fail to output data normally [20]. In this case, existing models need to use historical observations with missing values to predict future values of MTS, which limits their forecasting performances [21]. As shown in Fig. 1(a), with the missing rate increases, the forecasting error (mean absolute error) of existing STGNNs [22]–[24] significantly increases. This phenomenon makes us consider an important question: how can we enable STGNNs to achieve satisfactory results in MTSF with missing values? Considering the characteristics of STGNNs and the impact of missing values on multivariate time series, we believe that STGNNs need to address two core challenges: the tendency to generate incorrect spatial correlations and poor robustness. Next, we will explain these two challenges in detail.

Existing STGNNs [25], [26] rely on features from all variables to compute similarity and generate spatial correlations, making them highly susceptible to missing values. For example, as shown in Fig. 1(b), when the data is normal, there is a similar pattern between time series 1 and 2, and therefore they are modeled by STGNNs as having a spatial correlation. However, when both time series 2 and 3 have missing values (usually zero values [24]) simultaneously, STGNNs incorrectly generate a spatial correlation between them, while determining that there is no spatial correlation between time series 1 and 2. This phenomenon causes STGNNs to easily overlook and make mistakes when generating spatial correlations for MTS with missing values, thereby affecting the forecasting performance. In addition, since the status (normal or abnormal) of different data collectors can change over time, the missing

Chengqing Yu, Fei Wang, Zezhi Shao, Tangwen Qian, Zhao Zhang, Zhulin An, Qi Wang and Yongjun Xu are with the Institute of Computing Technology, CAS, Beijing 100190, China. Chengqing Yu, Fei Wang and Yongjun Xu are also with the University of Chinese Academy of Sciences, Beijing 100049, China. (e-mail: yuchengqing22b@ict.ac.cn; wangfei@ict.ac.cn; shaozezhi@ict.ac.cn; qiantangwen@ict.ac.cn; zhangzhao2021@ict.ac.cn; anzhulin@ict.ac.cn; wangqi08@ict.ac.cn; xjy@ict.ac.cn)

Wei Wei are with the School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China. (e-mail: weiw@hust.edu.cn)

Corresponding Authour: Fei Wang.

Manuscript received April 19, 2021; revised August 16, 2021.

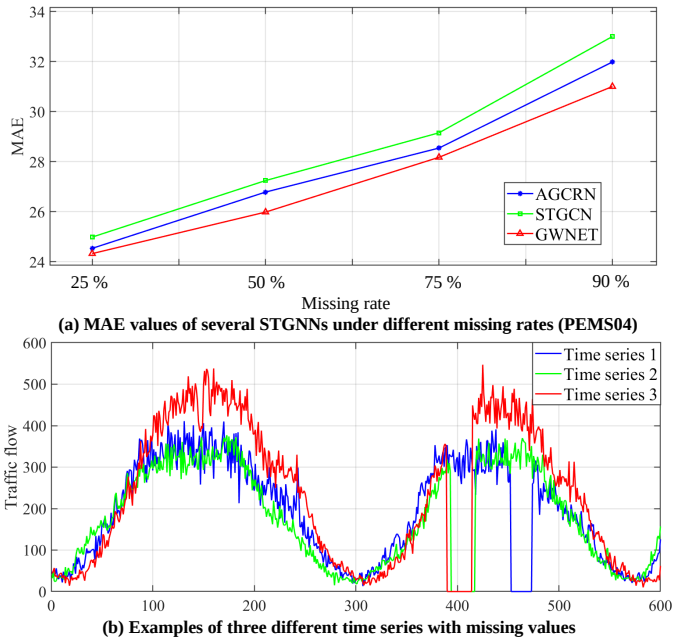


Fig. 1. Examples of MTSF with missing values on the PEMS04 dataset. (a) As the missing rate increases, the MAE values of multiple STGNNs significantly increase. (b) Three time series with missing values. Due to the existing of missing values, STGNNs easily generate incorrect spatial correlations for these three time series.

rate changes across different time steps. Existing STGNNs [27], [28] usually require training a separate model for each missing rate to ensure forecasting accuracy, which further limits their robustness and practicability.

At present, an intuitive approach to addressing the above two issues is to use an imputation model to restore missing values, and then to use the forecasting model to mine the restored data [29], [30]. Although two-stage models have achieved certain results, they still have the following problems: (1) Existing imputation models [31], [32] rely on prior knowledge to generate the spatial correlation between missing and normal values, thereby achieving the recovery of missing values. However, prior knowledge cannot fully screen all normal values, which limits the effectiveness of data recovery and thus lead to error accumulation. (2) Existing imputation models [33], [34] also require training a separate model for different missing rates. In practice, the missing rate is often not constant, which brings about problems of poor robustness. In summary, due to the inability to fully utilize normal values and issues of poor robustness, existing two-stage models still struggle to achieve satisfactory results.

To address these challenges, we believe that the forecasting model needs to have two key capabilities: (1) It needs to effectively use normal values to correct spatial correlations during the modeling process. (2) It needs to enhance its ability to adapt to data with different missing rates. To obtain correct spatial correlations, we propose an end-to-end framework called Graph Interpolation Attention Recursive Network (GinAR). We utilize Simple Recursive Units (SRU), based on the RNN framework, as the foundational architecture and introduce two key components: Interpolation Attention (IA) and Adaptive Graph Convolution (AGCN), to replace the fully connected (FC) layers in SRU. This modification aims to achieve end-to-end forecasting while accurately correcting spatial correlations. On one hand, during the recursive modeling process, IA initially establishes correspondences between normal and missing values for each time step, subsequently employing attention mechanisms to convert all missing values into plausible representations. On the other hand, for the representations refined by IA, we employ AGCN to reconstruct spatial correlations. With all missing values restored, AGCN can more effectively utilize these representations to develop a more reliable graph structure, thereby enhancing the accuracy of spatial correlation predictions. To enhance GinAR's ability to adapt to data with different missing rates, we first use data with different missing rates at the same time steps as positive data pairs and data from different time steps as negative data pairs. Then, based on the positive and negative data pairs, we train GinAR using contrastive learning and propose GinAR+. Since contrastive learning can help the model enhance the difference of negative data pairs and the similarity of positive data pairs [35], GinAR+ is able to distinguish data with different missing rates and improve its robustness. Through the above methods, GinAR+ can establish more accurate spatial correlations and only needs to be trained once to adapt to data with different missing rates, thereby achieving satisfactory results in MTSF with missing values.

This article is an extension of our conference version [36]. The new contributions are summarized as follows:

- We refine two core challenges in MTSF with missing values: the tendency to generate incorrect spatial correlations and poor robustness. To this end, we design a robust end-to-end forecasting framework.
- To prevent the model from being affected by missing values and generating incorrect spatial correlations, we carefully design Graph Interpolation Attention Recursive Network, which contains two key components (interpolation attention and adaptive graph convolution). We use above components to replace all FC layers in SRU and propose the GinAR cell, aiming to correct spatial correlations during the process of recursive modeling.
- We construct positive and negative data pairs using data with different missing rates. Based on these data pairs, we employ contrastive learning to train GinAR and subsequently propose GinAR+, which aims to enhance its robustness in handling different missing rates.
- We design experiments on nine real-world datasets. Results show that GinAR can outperform 11 baselines on all datasets. In addition, we provide more detailed experiments to verify our motivations.

II. Related Works

Considering that the mainstream models in MTSF with missing values are two-stage model (imputation and forecasting) and end-to-end model, we introduce spatial-temporal forecasting models, imputation models, and end-to-end forecasting models in this section. Besides, we also introduce contrastive learning methods used for time series forecasting.

A. Two-Stage Forecasting Models

Spatial-Temporal forecasting models. To establish the spatial correlation of MTS, GCN has been extensively studied in recent years [37]. STGNNs combine the advantages of GCN and sequence models, and improve the ability of modeling spatial-temporal correlation [38]. Li et al. [39] replace all FC layers in the gated recurrent unit (GRU) with GCN, and proposed the Diffusion Convolutional Recurrent Neural Network (DCRNN) to model multivariate time series. By combining the temporal convolutional network and GCN, Wu et al. [24] propose the graph wavenet (GWNET) to realize MTSF. Besides, to further improve the ability of STGNN to mine hidden spatial correlations between variables, adaptive graph learning strategies have been proposed [40]. Shang et al. [41] propose a discrete probabilistic graph structure to model spatial-temporal correlations. Shao et al. [42] propose a decoupled spatial-temporal framework along with a dynamic graph learning module to further improve the performance of STGNNs. The introduction of graph structure learning helps STGNNs fully analyze spatial correlations. However, missing values can affect the accuracy of graph structure learning and limit the forecasting performance of STGNNs.

Imputation models. GNN-based imputation models combines GCN and sequence models to analyze spatial-temporal correlations between the missing data and normal data, and further recover all missing data [43]. Wu et al. [44] propose a new spatial-temporal recovery framework based on inductive graph neural networks, achieving performance superior to that of traditional deep neural networks. Ivan et al. [45] combine temporal attention, cross attention, and graph structure to effectively recover missing values. Chen et al. [46] effectively combine adaptive graph convolution and recurrent neural networks to recover missing values in multivariate time series. By fully leveraging both temporal information and spatial correlations, the model can more accurately recover missing values. In summary, existing imputation methods require full use of prior knowledge to recover missing data. However, they are unable to fully utilize all the normal values, which limits their recovery performance under high missing rates [47], [48].

B. End-To-End Forecasting Models

Compared with the two-stage prediction model, the end-to-end prediction model improves the prediction performance and avoids the error accumulation by strengthening the model's ability to utilize normal values [32]. Tang et al. [49] introduce a query memory component to enhance the ability of LSTM in mining global information along the temporal dimension and propose the LGnet. In this way, the model can mitigate local anomalies caused by data missing and improve its forecasting performance. However, due to insufficient utilization of the spatial correlations, the performance of LGnet is not good when the missing rate is high. To fully utilize spatial correlations to further improve predictive performance, end-to-end models based on GNN have received widespread attention [30]. Xu et al. [50] propose a Graph-based Conditional Variational Recurrent Neural Network (GC-VRNN) that introduces a new Multi-Space Graph Neural Network (MS-GNN), which can extract spatial features from incomplete observations. Chen et al. [51] propose a biased Temporal Convolutional Network (TCN) over graphs that jointly captures temporal dependencies and spatial structures, which uses the Multi-Scale instance Partial TCN and the biased GCN for MTSF with missing values. In general, although existing end-to-end forecasting models introduce additional components to improve their utilization of normal values, they all require separate training of models for different miss rates, which limits their robustness [52], [53].

C. Contrastive Learning

The main role of contrastive learning is to help existing models enhance the difference in representations of negative data pairs and the similarity of representations of positive data pairs [54], [55]. In this way, the forecasting models can encode more discriminative representations for different samples and further improve their robustness [56]. Woo et al. [57] use trends and seasons from raw data to construct negative and positive data pairs to improve the model's ability to mine key representations. Zheng et al. [58] use forecasting results and labels as positive data pairs, aiming to improve the capabilities of both encoders and decoders. Luo et al. [59] design a set of adaptive signal enhancement strategies aimed at improving the quality of positive samples and enhancing the robustness of the model. In summary, if positive and negative data pairs can be correctly constructed and contrastive learning is fully utilized for training, the model can enhance its robustness to different samples and produce discriminative representations.

III. Methodology

A. Preliminaries

Multivariate time series. It represents the data composed of multiple sequences that change over time, and can be defined through a tensor $X \in \mathbb{R}^{N \times H \times C}$. N is the number of sequences. H is the number of time slices. C is the number of features.

Dependency graph. In multivariate time series, the change of each time series depends not only on itself but also on other time series. Such a dependency can be captured by the dependency graph $G = (V, E)$. V is the set of variables, and $|V| = N$. Each variable corresponds to a time series. E is the set of edges. The dependency graph can be represented by an adjacency matrix: $A \in \mathbb{R}^{N \times N}$.

Multivariate time series forecasting. Given a historical observation tensor $X \in \mathbb{R}^{N \times H \times C}$ from H time slices in history, the model can predict the value $Y \in \mathbb{R}^{N \times L}$ of the nearest L time steps in the future. C is the number of features. The goal of MTSF is to construct a mapping function between $X \in \mathbb{R}^{N \times H \times C}$ and $Y \in \mathbb{R}^{N \times L}$.

Multivariate time series forecasting with missing values. Compared with MTSF, the main difference of this task is that there are so much missing values in historical observations. In other words, we need to mask $M\%$ points from the historical observation tensor $X \in \mathbb{R}^{N \times H \times C}$. After the above processing, a new input feature $X_M \in \mathbb{R}^{N \times H \times C}$ is obtained. The core goal of this task is to construct mapping function between $X_M \in \mathbb{R}^{N \times H \times C}$ and $Y \in \mathbb{R}^{N \times L}$.

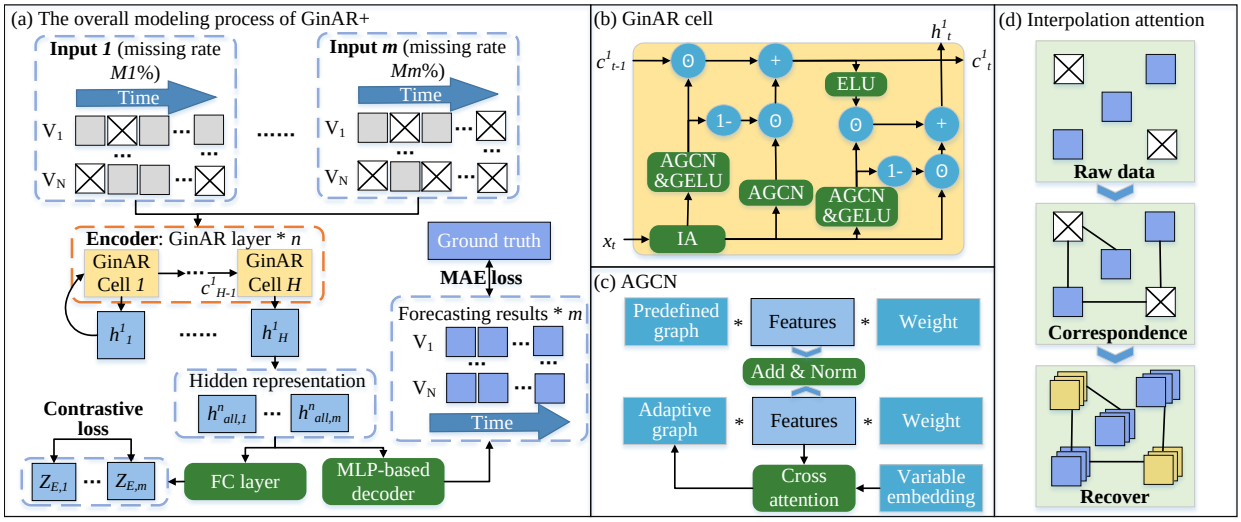


Fig. 2. The overall framework of GinAR+ and its components. (a) The modeling process of GinAR+. The GinAR layer adopts the SRU-based sequence framework and encodes historical observations with different missing values. The MLP-based decoder is used to predict future values. Contrastive loss and MAE loss are used to train GinAR. (b) The specific structure of the GinAR cell. It is achieved by replacing the FC layer in SRU with IA and AGCN. (c) The specific structure of the adaptive graph convolution. It includes predefined graphs and adaptive graphs. (d) The specific modeling process of the interpolation attention. Blue represents normal values. White represents missing values. Yellow indicates that the missing values are restored to the representation.

B. Overall Framework

The framework of GinAR+ is shown in Fig. 2. The input features are historical observations with missing values, and the outputs are the future values of all variables. GinAR+ uses multiple GinAR layers as the encoder and MLP [60], [61] as the decoder. The GinAR layer adopts the idea of recursive modeling, and its core structure is the GinAR cell. For each GinAR cell, we uses IA and AGCN as replacements for all FC layers within the SRU. IA utilizes the normal values to reconstruct missing values into plausible representations. AGCN incorporates graph learning to dynamically reconstruct the spatial correlations of the data processed by IA.

Besides, we combine contrastive loss with MAE loss to train GinAR, aiming to enhance its robustness and propose GinAR+. Specifically, since the operational status of different data collectors can change at different time steps, the missing rate is not constant [62]. This requires forecasting models to adapt to data with different missing rates in practical applications. To this end, we use data with different missing rates at the same time steps as positive data pairs and data from different time steps as negative data pairs, and train GinAR+ using contrastive loss. In this way, GinAR+ only needs to be trained once to adapt to data with different missing rates, thereby ensuring its robustness and practicality.

C. Interpolation Attention

For each missing value, the interpolation attention is tasked with selecting appropriate normal values for restoration and assigning corresponding weights to these selected normal values. This process involves two primary steps: (1) It initially establishes correspondences between missing and normal values, and (2) Leveraging the above correspondences along with the normal values, it employs attention mechanisms to reconstruct all missing values. The detailed modeling steps of Interpolation Attention (IA) are outlined as follows:

Step 1: First, we need to generate correspondences between missing values and normal values. Specifically, we initialize a diagonal matrix $I_N \in \mathbb{R}^{N\times N}$ and randomly initialize two embedding matrices $E_{IA1} \in \mathbb{R}^{N\times d}$ and $E_{IA2} \in \mathbb{R}^{d\times N}$. The values of these two embedding matrices can be iterated continuously during network training. Based on following formulas, correspondences between missing values and normal values can be obtained:

$$A_{IA} = (I_N + \text{softmax}(\text{ReLU}(E_{IA1}E_{IA2})), \tag{1}$$

where, softmax$(\cdot)$ is the activation function. ReLU$(\cdot)$ is the activation function. $A_{IA} \in \mathbb{R}^{N\times N}$ is a two-dimensional matrix. When the value of row i and column j in the A_{IA} is greater than 0, it means that there is a correlation between the missing value i and the normal value j. In other words, the interpolation attention can use the normal value j to recover the missing value i. Based on the above correlation matrix $A_{IA} \in \mathbb{R}^{N\times N}$, we can obtain the set of normal values $N(i)$ associated with the missing value i.

Step 2: Next, the missing value i is recovered by using attention mechanism and other associated normal values $j \in N(i)$. The attention coefficient α_{ij} between the missing value i and normal values $j \in N(i)$ can be calculated as follows:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(FC(W_j^{IA}h_j^{IA})\right)\right)}{\sum_{k\in N(i)}\exp\left(\text{LeakyReLU}\left(FC(W_k^{IA}h_k^{IA})\right)\right)}, \tag{2}$$

where, $FC(\cdot)$ is the fully connected layer. h_j^{IA} and W_j^{IA} represent the features and weight of normal values j, respectively. LeakyReLU$(\cdot)$ is the activation function. exp$(\cdot)$ stands for exponential function.

Step 3: The above attention coefficients are weighted and summed with the representations of all associated normal values to achieve the recovery of the missing value i.

$$h_i^{IA'} = \text{ReLU}(\sum_{j\in N(i)} \alpha_{ij}W_{ij}^{IA}h_j^{IA}). \tag{3}$$

Step 4: Repeat steps 2 through 3 until all missing values are restored. The original input feature $X_M \in \mathbb{R}^{N\times H\times C}$ can be converted to $X_M^{IA} \in \mathbb{R}^{N\times H\times C'}$. This step is completed by using matrix multiplication for parallel computation.

D. Adaptive Graph Convolution

By introducing prior knowledge to define an adjacency matrix A, we can utilize a predefined graph to help models establish foundational spatial correlations. However, in the case of Multivariate Time Series Forecasting (MTSF) with missing values, this predefined graph may not fully capture the spatial correlations among all variables due to the extensive amount of missing values. To address this issue, we introduce a data-based adaptive graph convolutional approach, which integrates both the predefined and the adaptive graphs. This combination enhances the model's ability to accurately represent spatial correlations even in the presence of missing values.

Predefined graph: The distance is used to construct adjacency matrix A for traffic data with road network information. For the data without road network information, the Pearson correlation coefficient [63] is used to form the adjacency matrix A. The predefined graph $A_{pre} \in \mathbb{R}^{N\times N}$ for the graph convolution network is obtained by the following formula:

$$A_{pre} = (I_N + D^{-1/2}AD^{-1/2}), \tag{4}$$

where, $I_N \in \mathbb{R}^{N\times N}$ represents the diagonal matrix with value 1. D is the degree matrix of A.

Adaptive graph: It needs to initialize a diagonal matrix $I_N \in \mathbb{R}^{N\times N}$ of value 1 and randomly initialize a variable-embedded matrix $E_A \in \mathbb{R}^{N\times d}$. The value of variable embedding matrix can be iterated continuously during neural network training. Then, based on the variable representation $X_M^{IA} \in \mathbb{R}^{N\times H\times C'}$ obtained by the interpolation attention and the variable-embedded matrix $E_A \in \mathbb{R}^{N\times d}$, the new variable embedding $E_n \in \mathbb{R}^{N\times d}$ are obtained.

$$E_n = \text{softmax}(\frac{W_q E_A * (W_k X_M^{IA})^T}{\sqrt{d_k}})W_v X_M^{IA}, \tag{5}$$

where, W represents the weight. softmax$(\cdot)$ stands for the normalized exponential function. d_k can let the outcome satisfy the distribution with expectation 0 and variance 1. The adaptive graph can be obtained by the following formula:

$$A_{adap} = (I_N + \text{softmax}(\text{TopK}(\text{ReLU}(E_n E_n^T))), \tag{6}$$

where, E_n^T represents the transpose of E_n. TopK$(\cdot)$ represents that we choose TopK neighbors for each variables.

Adaptive graph convolution: Based on the above formulas, the predefined graph and the adaptive graph can be obtained, which can reflect the spatial correlation of all variables from different perspectives. Then, we combine the adaptive graph convolution and layer normalization to fuse these graph information. The formula of the adaptive graph convolution is given as follows:

$$Z = F_{LN}(A_{pre}X_M^{IA}W_1 + b_1 + A_{adap}X_M^{IA}W_2 + b_2), \tag{7}$$

where, X_M^{IA} represents the variable representation obtained by the IA. W and b stand for weight and bias respectively. $F_{LN}(\cdot)$ stands for the layer normalization. Through above methods, the information of adaptive graph and predefined graph is fused.

E. GinAR

The main idea of GinAR is to integrate the IA and AGCN into the simple recursive units. Next, we introduce the composition of the GinAR cell and the overall modeling process of GinAR in detail.

GinAR cell: The GinAR cell is the most basic component of GinAR. Specifically, we introduce IA into the simple recursive unit cell to recover missing values. Besides, we use the AGCN to replace all full connected layers in the SRU cell, enhancing the ability to correct spatial-temporal dependencies. The formula for each GinAR cell is given as follows:

$$x_T^{IA} = F_{IA}(x_T), \tag{8}$$

$$\begin{aligned} f_T = \text{GeLU}(&F_{LN}(A_{pre}x_T^{IA}W_{f1}+ \\ &b_{f1} + A_{adap}x_T^{IA}W_{f2} + b_{f2})), \end{aligned} \tag{9}$$

$$\begin{aligned} r_T = \text{GeLU}(&F_{LN}(A_{pre}x_T^{IA}W_{r1}+ \\ &b_{r1} + A_{adap}x_T^{IA}W_{r2} + b_{r2})), \end{aligned} \tag{10}$$

$$\begin{aligned} c_T = (1-f_T)\odot F_{LN}(&A_{pre}x_T^{IA}W_{c1}+ \\ &A_{adap}x_T^{IA}W_{c2}) + f_T \odot c_{T-1}, \end{aligned} \tag{11}$$

$$h_T = r_T \odot \text{ELU}(c_T) + (1-r_T)\odot x_T^{IA}, \tag{12}$$

where, r_T stands for reset gate. f_T stands for forget gate. c_T represents the cell state of the current GinAR cell. h_T is the hidden state of the current GinAR cell. $\odot$ stands for the Hadamard product. GeLU$(\cdot)$ and ELU$(\cdot)$ are activation functions. $F_{IA}(\cdot)$ stands for the interpolation attention.

GinAR: The main components of GinAR include n GinAR layers and an MLP-based decoder. Each GinAR layer contains multiple GinAR cells. The modeling process of GinAR is given as follows:

Step 1: The original input feature $X \in \mathbb{R}^{N\times H\times C}$ is preprocessed and the input feature $X_M \in \mathbb{R}^{N\times H\times C}$ for modeling is obtained. There are M points in the input feature $X_M \in \mathbb{R}^{N\times H\times C}$ that are treated as missing values.

$$X_M = [x_1, x_2, ..., x_H], x \in \mathbb{R}^{N\times C}, \tag{13}$$

where, H is the length of historical observation. N is the number of variables. L is the length of future forecasting results. C stands for embedding size.

Step 2: X_M is passed to the first GinAR layer. It contains H GinAR cells, which are used to model x_1 to x_H.

Step 3: Initialize a cell state c_0. x_1 and c_0 are passed to the first GinAR cell in the GinAR layer. Based on the calculation formula of GinAR cell, the hidden state h_1^1 of the current cell and the cell state c_1 are obtained. c_1 and x_2 are passed to the next GinAR cell.

Step 4: Repeat Step 3 to obtain H hidden states of all GinAR cells in the first GinAR layer. These hidden states h^1 are used as the input features to the next GinAR layer.

$$h^1 = [h_1^1, h_2^1, ..., h_H^1]. \tag{14}$$

Step 5: Repeat steps 3 to 4 until all hidden states of n GinAR layers are obtained. The hidden state of the last cell

in each GinAR layer is extracted. These hidden states are concatenated together as a new tensor h^n_{all}.

$$h^n_{all} = [h^1_H, h^2_H, ..., h^n_H], h^n_{all} \in \mathbb{R}^{N \times C' \times n}. \quad (15)$$

Step 6: h^n_{all} is passed to the MLP-based generative decoder. The final forecasting result $Y \in \mathbb{R}^{N \times L}$ is obtained.

$$Y = FC(\text{ReLU}(FC(h^n_{all}))). \quad (16)$$

Next, we demonstrate how to use contrastive learning to train GinAR with the hidden representation h^n_{all} and the forecasting result Y, and propose GinAR+.

F. Contrastive Learning and GinAR+

Contrastive Loss: Considering that the missing rates of historical observations in reality are not fixed, we aim to enhance the robustness of GinAR by proposing a contrastive learning method. We treat historical observations with different missing rates at the same time steps as positive data pairs, and historical observations at different time points (other samples within the same batch) as negative data pairs. For representations $h^n_{all,1}$ to $h^n_{all,m}$, encoded by historical observations with different missing rates, we employ a pairwise contrastive learning approach to facilitate multi-view contrastive learning. The specific steps are given as follows:

Step I: A fully connected layer is used to decode the hidden representations $h^n_{all,1}$ to $h^n_{all,m}$, and get the representations $Z_{E,1}$ to $Z_{E,m}$ for contrastive learning.

$$Z_{E,1} = FC(h^n_{all,1}). \quad (17)$$

Step II: Firstly, we use the $Z_{E,1}$ and $Z_{E,2}$ to obtain $2N_s$ samples. The contrast loss between any two samples $z_{E,i}$ and $z_{E,j}$ is shown as follows:

$$l_{i,j} = -log(\frac{exp(sim(z_{E,i}, z_{E,j})/\tau)}{\sum^{2N_s}_{k=1\&k\neq i} exp(sim(z_{E,i}, z_{E,k})/\tau)}), \quad (18)$$

where, $exp(\cdot)$ is the exp function. $sim(\cdot)$ is the Cosine similarity. N_s is the number of samples. τ is the temperature parameter.

Step III: Then, the contrastive loss between $Z_{E,1}$ and $Z_{E,2}$ can be obtained by the following formula:

$$L_{Z1,Z2} = \frac{1}{2N_s}\sum_{k=1}^{N_s}(l_{2k-1,2k} + l_{2k,2k-1}). \quad (19)$$

Step IV: Repeat the above steps and obtain the contrastive loss between $Z_{E,1}$ to $Z_{E,m}$ pairwise. The final multi-view contrastive learning loss is shown below:

$$L_{CL} = \frac{2}{m(m-1)}(\sum_{Zj=Zi}^{m}\sum_{Zi=1}^{m-1} L_{Zi,Zj}). \quad (20)$$

Loss Function: In addition to the contrastive Loss, we also incorporate ground truth and MAE loss to train GinAR, enabling the realization of the supervised learning process. The formula is shown as follows:

$$L_{Pre} = \frac{1}{m}(\sum_{i=1}^{m} Mean(|Y_{tru} - Y_i|), \quad (21)$$

where, Y_{tru} is the ground truth. $|\cdot|$ stands for absolute value.

Finally, we need to effectively integrate Contrastive Loss and MAE Loss. There are two main ways to integrate these Loss functions: multi-stage training or stacking all Loss functions. Considering the problem of information forgetting caused by multi-stage training, we adopts the method of adding all Loss functions. The formula is given as follows:

$$L_{Finally} = L_{Pre} + L_{CL}. \quad (22)$$

Based on the above loss function, we propose GinAR+, which requires only one training session to adapt to data with different missing rates and exhibits good robustness.

IV. Experimental Study

A. Experimental Design

Datasets. Nine real-world datasets are selected to conduct comparative experiments, including two traffic speed datasets (METR-LA and PEMS-BAY)[1], two traffic flow datasets (PEMS04 and PEMS08)[2], two meteorological datasets (China AQI[3] and Weather[4]), two energy datasets (Electricity[5] and ETTh1[6]) and a financial dataset (Exchange)[7]. Table I shows the statistics of these datasets. A brief overview of these datasets is shown as follows:

- **METR-LA**: It is a traffic speed dataset collected by loop-detectors located on the LA County road network, which contains data collected by 207 sensors from Mar 1st, 2012 to Jun 30th, 2012. Each time series is sampled at a 5-minute interval, totaling 34272 time slices.
- **PEMS-BAY**: It is a traffic speed dataset collected by California Transportation Agencies (CalTrans) Performance Measurement System (PeMS), which contains data collected by 325 sensors from Jan 1st, 2017 to May 31th, 2017. Each time series is sampled at a 5-minute interval, totaling 52116 time slices.
- **PEMS04**: It is a traffic flow dataset collected by CalTrans PeMS, which contains data collected by 307 sensors from January 1st, 2018 to February 28th, 2018. Each time series is sampled at a 5-minute interval, totaling 16992 time slices.
- **PEMS08**: It is a traffic flow dataset collected by CalTrans PeMS, which contains data collected by 170 sensors from July 1st, 2018 to Aug 31th, 2018. Each time series is sampled at a 5-minute interval, totaling 17833 time slices.
- **China AQI**: It is an air quality dataset collected by China Environmental Monitoring Station, which contains data collected by 350 cities in China from January 2015 to December 2022. Each time series is sampled at a 1 hour interval, totaling 59710 time slices.
- **Electricity**: It is derived from The UCI Machine Learning Repository, and is the electricity consumption of 321

[1]https://github.com/liyaguang/DCRNN
[2]https://github.com/guoshnBJTU/ASTGNN/tree/main/data
[3]https://quotsoft.net/air/
[4]https://www.bgc-jena.mpg.de/wetter/
[5]https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams2011201
[6]https://github.com/zhouhaoyi/ETDataset
[7]https://github.com/laiguokun/multivariate-time-series-data

TABLE I
THE STATISTICS OF THE FIVE DATASETS.

Datasets	Variates	Timesteps	Granularity
MET-LA	207	34272	5 minutes
PEMS-BAY	325	52116	5 minutes
PEMS04	307	16992	5 minutes
PEMS08	170	17856	5 minutes
China AQI	350	59710	1 hour
Electricity	321	26304	1 hour
ETTh1	7	14400	1 hour
Weather	21	52696	10min
Exchange	8	7588	1 day

customers from 2012 to 2014. Each time series is sampled at a 1 hour interval, totaling 26304 time slices.

- **ETTh1**: It collected 20 months of Electricity Transformer data from China, covering 7 variables. Each time series is sampled at a 1-hour interval, totaling 14400 time slices.
- **Weather**: It recorded 21 meteorological indicators throughout the year 2020 in Germany. Each time series is sampled at a 10-minute interval, totaling 52696 time slices.
- **Exchange**: It collects daily exchange rates from eight countries (Australia, the United Kingdom, Canada, Switzerland, China, Japan, New Zealand, and Singapore) from 1990 to 2016. Each time series is sampled at a 1-day interval, totaling 7588 time slices.

Baselines. Eleven existing models are selected as the baselines, which include end-to-end forecasting models (LGnet [49], TriD-MAE [64], GC-VRNN [50], and BiTGraph [51]) and two-phase models (PatchTST [65] + SAITS [66], iTransformer [67] + CSDI [68], DCRNN [39] + GPT4TS [69], DFDGCN [70] + SPIN [45] and MTGNN [71] + GRIN [72], STID [73] + TimesNet [74], FourierGNN [75] + GATGPT [43]). The selection and combination of two-stage models are determined based on existing surveys [6], [76], [77] and experimental results, aiming to evaluate as many baselines as possible while ensuring their performance. All baselines are introduced as follows:

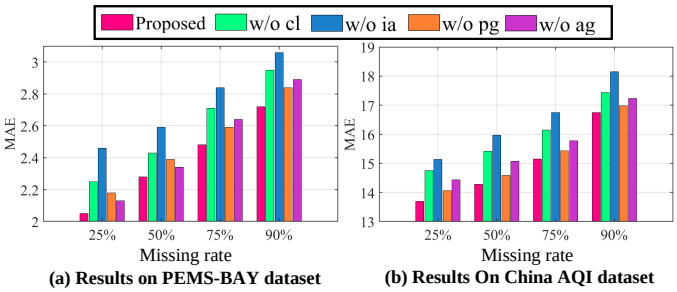
- **PatchTST+SAITS** : It first uses SAITS to realize the imputation of missing values, and then uses PatchTST to model the processed data.
- **iTransformer+CSDI** : It first uses CSDI to realize the imputation of missing values, and then uses iTransformer to model the processed data.
- **DCRNN+GPT4TS** : It first uses GPT4TS to realize the imputation of missing values, and then uses DCRNN to model the processed data.
- **DFDGCN+SPIN**: It first uses SPIN to realize the imputation of missing values, and then uses DFDGCN to model the processed data.
- **MTGNN+GRIN**: It first uses GRIN to realize the imputation of missing values, and then uses MTGNN to model the processed data.
- **STID+TimesNet**: It first uses TimesNet to realize the imputation of missing values, and then uses STID to model the processed data.
- **FourierGNN+GATGPT**: It first uses GATGPT to realize the imputation of missing values, and then uses FourierGNN to model the processed data.
- **LGnet**: It uses the memory component to effectively improve the performance of LSTM.
- **GC-VRNN**: It combines the Multi-Space Graph Neural Network with Conditional Variational Recurrent Neural to realize MTSF with missing values.
- **TriD-MAE**: It uses MAE to optimize the ability of the TCN model to realize MTSF with missing values.
- **BiTGraph**: It proposes a Biased Temporal Convolution Graph Network that jointly captures the temporal dependencies and spatial structure.

TABLE II
VALUES OF THE CORRESPONDING HYPERPARAMETERS OF GINAR+.

Config	Values
optimizer	Adam
learning rate	0.006
weight decay	0.0001
embedding size	16
variable embedding size	8
number of layers	3
TopK	12
temperature parameter	0.1
dropout	0.15
learning rate schedule	MultiStepLR
clip gradient normalization	5
milestone	[1,15,30,50,70,90]
gamme	0.5
batch size	16
epoch	100

Setting. Table II shows the main hyperparameters of GinAR+. We design experiments from the following aspects: (1) All datasets are uniformly divided into training sets, validation sets and test sets according to the ratio in the reference [77]. (2) We set the history/future length based on the existing work [78]. The history length and future length of GinAR+ are both 12. Metrics are the average of 12-step forecasting results. (3) We set missing values according to the ratio of 25%, 50%, 75% and 90%. We refer to references [72], [74] to convert the normal values into missing values. Besides, we construct data with different missing rates using two ways [79], [80]: the first way directly applies the random masking strategy to obtain data with different missing rates. In the second way, the data with high missing rates is obtained by further masking the data with low missing rates. In other words, the missing values in the data with low missing rates are a subset of those in data with high missing rates. The experiment was repeated with 5 different random seeds for each missing rate. The final metrics are the averages of repeated experiments and above two different masking ways. (4) For GinAR+, based on contrastive learning and MAE loss, we train only one model capable of adapting to different missing rates using data with multiple missing rates. For other baselines, we train the models individually for each missing rate.

Metrics. To comprehensively evaluate the forecasting performance of different models, we utilize three classical metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) [81].



(a) Results on PEMS-BAY dataset **(b) Results On China AQI dataset**

Fig. 3. The results of the ablation experiment.

B. Main Results

Table III and Table IV show the performance comparison results of all baselines and GinAR+ on different datasets (The best results are shown in **bold**). Since the number of variables is small and there are no predefined graphs, the three datasets in Table IV are mainly used to compare GinAR+ against four baselines that do not rely on predefined graphs. Based on results, the following conclusions can be obtained: (1) Although imputation models are practical and capable of recovering missing values, two-stage models are prone to error accumulation, which limits the performance of forecasting models. (2) End-to-end forecasting models demonstrate superior performance compared to two-stage baselines. These models not only remedy the shortcomings of one-stage models in handling missing data but also avoid the error accumulation typically associated with two-stage approaches. (3) The proposed model, GinAR+, consistently achieves optimal performance across all datasets and settings. By fully leveraging interpolation attention, adaptive graph convolution, and an RNN-based framework, GinAR+ effectively recovers missing values, reconstructs spatial-temporal correlations, and facilitates end-to-end forecasting. Moreover, by incorporating contrastive learning, GinAR+ adapts to data with different missing rates by training only one model, thereby outperforming all baseline models in multivariate time series forecasting with missing values and enhancing robustness.

C. Ablation Experiment

GinAR+ has four important components: interpolation attention, predefined graph, adaptive graph learning and contrastive loss. To demonstrate the importance of these components, ablation experiments are conducted from the following four perspectives: (1) **w/o ia**: We remove the interpolation attention. It means that GinAR+ directly analyzes the missing values. (2) **w/o pg**: The predefined graph is deleted. GinAR+ only uses the adaptive graph to construct spatial correlations. (3) **w/o ag**: The adaptive graph is removed. Spatial correlations are determined by the prior knowledge. (4) **w/o cl**: The contrastive loss is removed. It means that GinAR+ directly uses data with different missing rates to train itself.

Fig. 3 shows the results of the ablation experiment. The following conclusions can be drawn: (1) When the missing rate is low, the elimination of the predefined graph significantly affects the forecasting results. Conversely, when the missing rate is high, the removal of the predefined graph has a negligible impact on the forecasting results. (2) When the missing rate is high, removing the adaptive graph substantially diminishes the accuracy of the forecasting results. This is primarily because the adaptive graph is more effective at analyzing spatial correlations based on data characteristics when the missing values are abundant. Hence, the adaptive graph is crucial for achieving accurate forecasting in scenarios with high missing rates. (3) When the contrastive loss is deleted, the forecasting performance of GinAR+ decreases greatly. The main reason is that the contrastive loss introduces additional binary classification tasks, which helps GinAR+ enhance its ability to distinguish samples with different missing rates, thereby improving its robustness. (4) When IA is removed, the performance of GinAR+ decreases significantly, proving that IA is the most important component. The main reason is that IA realizes the recovery of missing values, which provides an important support for correcting spatial-temporal dependencies and avoiding error accumulation.

D. Performance Evaluation of IA and Contrastive Learning

This section compares the performance improvement effects of IA+CL, IA, GRIN, GATGPT, GPT4TS and SPIN on STID. Specifically, TimesNet, GATGPT, GPT4TS and GRIN adopt the two-stage modeling framework (imputation and forecasting) to optimize STID. IA uses the end-to-end modeling framework to optimize STID. Besides, STID+IA+CL trains only one model for all missing rates. Others train models separately for each missing rate. Table V shows the performance comparison results (MAE values) of these models (The best results are shown in **bold**). Based on the results, the following conclusions can be drawn: (1) Compared with other two-stage models, the end-to-end framework that integrates IA and STID delivers superior forecasting results. On the one hand, two-stage models often involve feature reconstruction, which can lead to error accumulation and the reduction in forecasting accuracy. On the other hand, IA within the end-to-end framework facilitates adaptive induction by establishing correspondences between normal and missing values. (2) Compared with STID+IA, STID+IA+CL not only achieves lower forecasting errors but also requires training only one model for all missing rates. The contrastive loss helps STID+IA effectively identify data with different different rates. The experimental results demonstrate the value of contrastive loss in enhancing the robustness of forecasting models.

E. Hyperparameter Analysis

We evaluate the influence of six hyperparameters on the results, including variable embedding size, number of TopK, embedding size, number of layers, temperature parameter, and batch size. Fig. 4 shows the influence of different hyperparameters (PEMS04 dataset). The following conclusions can be obtained: (1) The variable embedding size has the least influence on the forecasting results. It proves that the adaptive graph with good performance can be generated without a large number of parameters. (2) The number of TopK can be increased appropriately. The main reason is that properly increasing the number of TopK can reflect the spatial correlations more

TABLE III
PERFORMANCE COMPARISON RESULTS OF ALL BASELINES AND GINAR+ ON DIFFERENT DATASETS.

Datasets	Methods	Missing rate 25%			Missing rate 50%			Missing rate 75%			Missing rate 90%		
		RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE
METR-LA	PatchTST+SAITS	7.45	10.95	3.83	7.71	11.23	3.94	8.03	11.94	4.08	8.31	12.51	4.25
	iTransformer+CSDI	7.34	10.87	3.78	7.65	11.18	3.91	7.99	12.05	4.03	8.23	12.41	4.21
	DCRNN+GPT4TS	6.95	10.24	3.68	7.37	10.84	3.81	7.72	11.25	3.96	8.01	11.86	4.07
	DFDGCN+SPIN	6.92	10.21	3.65	7.35	10.86	3.79	7.69	11.18	3.93	7.96	12.01	4.02
	MTGNN+GRIN	6.89	10.15	3.63	7.34	10.79	3.75	7.68	11.24	3.91	7.97	11.94	4.04
	STID+TimesNet	6.94	10.25	3.66	7.33	10.81	3.76	7.69	11.24	3.95	7.99	11.92	4.05
	FourierGNN+GATGPT	6.97	10.31	3.70	7.42	10.91	3.84	7.86	11.44	3.97	8.14	12.19	4.11
	LGnet	7.04	10.78	3.76	7.57	11.15	3.88	7.94	11.71	4.01	8.25	12.38	4.15
	GC-VRNN	6.85	9.97	3.62	7.25	10.48	3.71	7.65	11.14	3.85	7.95	11.74	3.98
	TriD-MAE	6.83	10.04	3.61	7.31	10.54	3.73	7.62	11.09	3.86	7.92	11.69	3.96
	BiTGraph	6.72	9.88	3.54	7.23	10.36	3.62	7.44	10.73	3.71	7.86	11.36	3.89
	GinAR+	**6.64**	**9.83**	**3.51**	**7.15**	**10.23**	**3.59**	**7.34**	**10.58**	**3.67**	**7.79**	**11.17**	**3.85**
PEMS-BAY	PatchTST+SAITS	4.73	5.45	2.34	5.07	5.81	2.49	6.49	7.06	2.81	7.13	7.98	3.14
	iTransformer+CSDI	4.69	5.41	2.32	5.04	5.77	2.47	6.23	6.91	2.74	6.96	7.76	3.09
	DCRNN+GPT4TS	4.53	5.24	2.23	4.94	5.61	2.37	5.62	6.28	2.59	6.54	7.14	2.83
	DFDGCN+SPIN	4.54	5.21	2.24	4.91	5.57	2.34	5.59	6.19	2.57	6.56	7.22	2.85
	MTGNN+GRIN	4.51	5.23	2.21	4.92	5.63	2.35	5.61	6.25	2.58	6.51	7.08	2.82
	STID+TimesNet	4.55	5.25	2.25	4.95	5.62	2.36	5.58	6.23	2.56	6.52	7.12	2.83
	FourierGNN+GATGPT	4.57	5.28	2.27	4.98	5.68	2.40	5.67	6.35	2.64	6.63	7.35	2.91
	LGnet	4.61	5.39	2.31	5.02	5.71	2.45	5.83	6.44	2.71	6.71	7.53	3.06
	GC-VRNN	4.46	5.17	2.17	4.87	5.52	2.31	5.57	6.11	2.55	6.45	7.02	2.79
	TriD-MAE	4.42	5.08	2.13	4.89	5.54	2.32	5.54	6.13	2.54	6.42	6.98	2.78
	BiTGraph	4.33	4.96	2.08	4.73	5.43	2.30	5.40	6.07	2.51	6.25	6.94	2.75
	GinAR+	**4.28**	**4.85**	**2.05**	**4.69**	**5.37**	**2.28**	**5.35**	**5.94**	**2.49**	**6.22**	**6.86**	**2.72**
PEMS04	PatchTST+SAITS	39.74	17.45	24.37	41.17	18.17	26.04	42.36	18.67	27.23	45.62	20.38	30.76
	iTransformer+CSDI	39.47	17.31	24.15	40.97	18.06	25.97	42.25	18.61	27.15	45.31	20.07	30.54
	DCRNN+GPT4TS	39.07	17.04	23.17	40.17	17.41	25.38	41.97	18.45	26.84	44.10	19.45	29.75
	DFDGCN+SPIN	39.02	17.08	23.15	40.20	17.49	25.41	41.75	18.33	26.79	43.76	19.12	29.48
	MTGNN+GRIN	38.94	16.64	23.06	40.08	17.38	25.32	41.64	18.22	26.73	43.89	19.39	29.61
	STID+TimesNet	38.87	16.59	22.84	39.94	17.31	24.97	41.43	17.95	26.58	43.65	19.02	29.06
	FourierGNN+GATGPT	38.92	16.61	22.98	40.05	17.55	25.31	41.57	18.30	26.67	43.83	19.35	29.56
	LGnet	39.15	17.22	23.73	40.39	17.64	25.59	42.05	18.53	27.04	44.52	19.74	30.09
	GC-VRNN	38.45	16.52	22.81	39.52	17.13	24.33	41.23	17.59	26.43	43.01	18.94	28.76
	TriD-MAE	38.32	16.48	22.75	39.45	17.22	24.19	41.18	17.64	26.29	42.95	18.85	28.54
	BiTGraph	37.93	16.31	22.49	39.04	16.92	23.73	41.09	17.45	25.98	42.58	18.37	28.04
	GinAR+	**37.83**	**16.05**	**22.32**	**38.74**	**16.86**	**23.41**	**40.88**	**17.26**	**25.72**	**42.29**	**18.03**	**27.87**
PEMS08	PatchTST+SAITS	32.06	13.79	21.43	34.98	14.25	22.18	37.77	16.42	25.06	40.47	18.04	26.64
	iTransformer+CSDI	31.85	13.74	21.39	34.59	14.14	22.07	37.58	16.34	24.96	40.36	17.65	26.39
	DCRNN+GPT4TS	31.62	13.56	20.64	34.12	14.03	21.96	37.04	15.79	24.58	38.93	16.97	25.83
	DFDGCN+SPIN	31.66	13.67	20.71	33.97	13.93	21.84	36.95	15.74	24.45	38.85	16.75	25.62
	MTGNN+GRIN	31.57	13.34	20.59	34.02	13.98	21.78	36.97	15.68	24.51	38.77	16.58	25.56
	STID+TimesNet	31.43	13.28	20.52	33.89	13.86	21.76	36.59	15.34	23.97	38.52	16.18	25.24
	FourierGNN+GATGPT	31.64	13.45	20.77	34.08	14.08	21.91	37.01	15.85	24.89	39.04	17.06	26.03
	LGnet	32.34	13.85	21.51	34.83	14.31	22.14	37.38	16.26	24.95	40.15	17.71	26.48
	GC-VRNN	31.12	13.21	20.42	33.51	13.91	21.67	36.31	15.06	23.45	38.09	15.83	24.52
	TriD-MAE	31.05	13.17	20.37	33.47	13.86	21.62	36.25	14.98	23.27	38.05	15.74	24.39
	BiTGraph	31.02	13.04	20.21	33.45	13.82	21.59	35.91	14.79	23.06	37.97	15.56	24.27
	GinAR+	**30.79**	**12.97**	**20.03**	**33.27**	**13.75**	**21.55**	**35.68**	**14.52**	**22.93**	**37.81**	**15.35**	**24.04**
China AQI	PatchTST+SAITS	27.35	30.87	14.82	28.35	33.12	15.76	29.03	35.44	16.68	32.65	40.37	18.56
	iTransformer+CSDI	27.17	30.54	14.75	28.13	32.87	15.63	28.95	35.12	16.54	32.31	39.97	18.27
	DCRNN+GPT4TS	26.21	29.58	14.27	27.75	31.94	15.02	28.52	34.68	16.06	32.07	38.44	17.78
	DFDGCN+SPIN	26.32	29.71	14.31	27.72	31.85	14.96	28.45	34.49	15.99	31.96	38.28	17.74
	MTGNN+GRIN	26.15	29.82	14.24	27.66	31.76	14.93	28.49	34.34	16.04	31.92	38.23	17.69
	STID+TimesNet	25.89	29.04	14.02	27.34	31.56	14.78	28.15	33.26	15.64	31.15	37.06	17.32
	FourierGNN+GATGPT	27.05	30.46	14.71	28.06	32.37	15.42	28.86	34.90	16.37	32.17	39.84	18.18
	LGnet	27.58	31.63	15.24	28.53	33.75	16.09	29.09	35.63	16.72	32.54	40.18	18.74
	GC-VRNN	25.84	28.96	13.97	27.23	30.67	14.70	27.97	32.36	15.43	31.06	36.73	17.13
	TriD-MAE	25.78	28.87	13.92	27.18	30.35	14.64	27.89	32.45	15.38	30.92	36.56	17.04
	BiTGraph	25.55	28.36	13.81	26.98	30.06	14.47	27.69	32.33	15.22	30.58	36.17	16.86
	GinAR+	**25.43**	**28.15**	**13.69**	**26.71**	**29.42**	**14.27**	**27.52**	**31.75**	**15.16**	**30.47**	**35.95**	**16.74**
Electricity	PatchTST+SAITS	2246.18	30.25	278.43	2553.16	32.07	302.47	2769.81	33.53	336.45	3211.77	37.41	368.42
	iTransformer+CSDI	2273.18	30.72	280.78	2574.32	32.34	305.87	2801.32	33.87	337.42	3178.25	37.15	365.93
	DCRNN+GPT4TS	2085.32	29.64	269.14	2451.87	31.76	294.33	2713.68	33.29	326.62	3123.44	36.95	361.48
	DFDGCN+SPIN	2024.37	28.73	260.13	2349.56	31.03	284.63	2697.45	33.18	324.17	3059.28	36.24	357.83
	MTGNN+GRIN	1998.44	28.65	258.49	2243.12	30.27	279.65	2643.87	32.47	311.95	3027.14	35.74	353.61
	STID+TimesNet	2053.74	28.71	261.34	2378.92	31.25	286.82	2663.18	32.65	316.47	3029.86	35.97	355.27
	FourierGNN+GATGPT	2065.27	29.07	265.14	2403.35	31.67	289.15	2680.15	32.79	317.92	3067.54	36.12	358.47
	LGnet	2264.31	30.52	279.44	2568.12	32,27	304.46	2783.15	33.81	336.82	3247.66	37.75	368.79
	GC-VRNN	1957.28	28.57	253.44	2176.43	29.97	274.35	2583.69	32.56	306.88	3014.82	35.42	351.76
	TriD-MAE	1923,16	28.12	250.31	2189.42	30.06	275.31	2571.39	32.19	304.12	2987.16	34.78	348.16
	BiTGraph	1832.66	27.41	241.75	2092.16	29.13	269.73	2437.16	31.52	292.45	2834.16	33.94	342.79
	GinAR+	**1817.54**	**26.98**	**239.18**	**2063.41**	**29.04**	**265.28**	**2396.64**	**31.37**	**291.58**	**2784.62**	**33.76**	**337.48**

TABLE IV
MAE VALUES OF SEVERAL BASELINES AND GINAR+ ON DIFFERENT DATASETS.

Datasets	Methods	Missing rates			
		25%	50%	75%	90%
ETTh1	GinAR+	**0.518**	**0.548**	**0.584**	**0.615**
	BiTGraph	0.521	0.557	0.589	0.619
	STID+TimesNet	0.529	0.564	0.597	0.627
	iTransformer+CSDI	0.524	0.559	0.593	0.621
	PatchTST+SAITS	0.532	0.565	0.602	0.632
Weather	GinAR+	**0.127**	**0.142**	**0.157**	**0.173**
	BiTGraph	0.129	0.145	0.160	0.177
	STID+TimesNet	0.138	0.152	0.168	0.185
	iTransformer+CSDI	0.131	0.146	0.163	0.181
	PatchTST+SAITS	0.140	0.155	0.171	0.189
Exchange	GinAR+	**0.102**	**0.118**	**0.135**	**0.156**
	BiTGraph	0.107	0.121	0.138	0.163
	STID+TimesNet	0.113	0.130	0.143	0.168
	iTransformer+CSDI	0.108	0.124	0.139	0.165
	PatchTST+SAITS	0.115	0.132	0.147	0.170

TABLE V
MAE VALUES OF INTERPOLATION ATTENTION, CONTRASTIVE LEARNING AND OTHER IMPUTATION METHODS.

Datasets	Methods	Missing rates			
		25%	50%	75%	90%
METR-LA	STID+GPT4TS	3.64	3.73	3.91	4.01
	STID+SPIN	3.57	3.67	3.78	3.94
	STID+GATGPT	3.61	3.69	3.79	3.93
	STID+GRIN	3.59	3.70	3.81	3.96
	STID+IA	3.55	3.63	3.75	3.92
	STID+IA+CL	**3.54**	**3.61**	**3.72**	**3.88**
PEMS08	STID+GPT4TS	20.49	21.73	23.84	25.06
	STID+SPIN	20.30	21.67	23.49	24.58
	STID+GATGPT	20.34	21.69	23.54	24.63
	STID+GRIN	20.35	21.71	23.52	24.75
	STID+IA	20.29	21.63	23.41	24.52
	STID+IA+CL	**20.26**	**21.61**	**23.34**	**24.45**
China AQI	STID+GPT4TS	13.96	14.69	15.58	17.26
	STID+SPIN	13.83	14.45	15.42	16.97
	STID+GATGPT	13.92	14.54	15.49	17.06
	STID+GRIN	13.89	14.49	15.51	17.13
	STID+IA	13.76	14.41	15.34	16.91
	STID+IA+CL	**13.73**	**14.36**	**15.27**	**16.83**

comprehensively. (3) The embedding size has great influence on experimental results and cannot be too large. The main reason is that the large embedding size can lead to overfitting, thus affecting the forecasting accuracy. (4) The number of layers has great influence on the forecasting result. Too few layers can not adequately mine and analyze the data. Too many layers can lead to problems such as overfitting. (5) Achieving the right balance of the temperature parameter is crucial for enhancing the effectiveness of contrastive learning. On the one hand, appropriately lowering the temperature parameter can enhance the model's performance and accelerate convergence. On the other hand, setting the temperature parameter too low may result in the problem of local optimality. (6) Increasing the batch size can increase the number of negative data pairs, which helps GinAR+ generate more discriminative representations. However, an excessively large batch size may cause GinAR+ to converge prematurely, leading to underfitting.

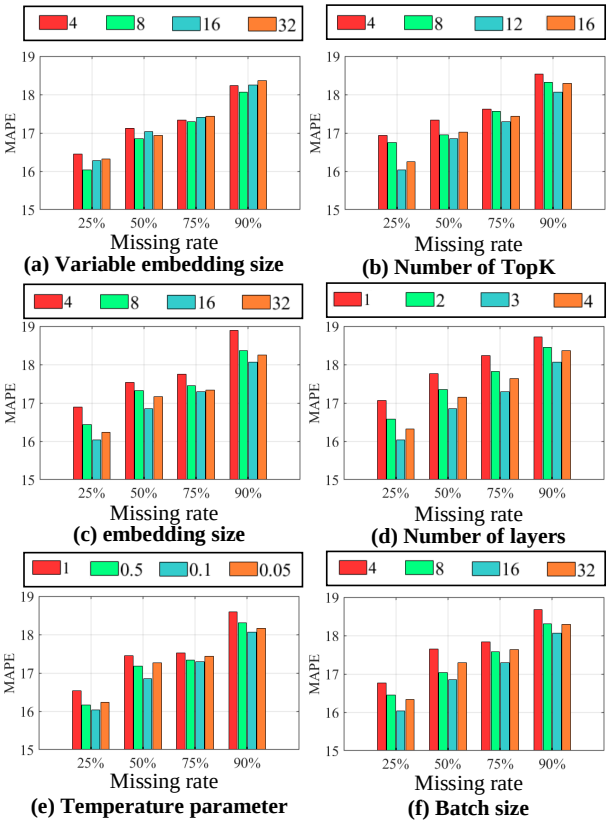


Fig. 4. The results of hyperparameter experiment (PEMS04 dataset).

TABLE VI
RMSE VALUES OF DIFFERENT LOSS FUNCTIONS

Datasets	Methods	Missing rates			
		25%	50%	75%	90%
METR-LA	Proposed	**6.64**	**7.15**	**7.34**	**7.79**
	Proposed+RL	6.69	7.18	7.39	7.81
	Two-stage training	6.71	7.22	7.42	7.84
PEMS04	Proposed	**37.83**	**38.74**	**40.88**	**42.29**
	Proposed+RL	37.89	38.91	40.97	42.41
	Two-stage training	37.93	39.04	41.06	42.53
China AQI	Proposed	**25.43**	**26.71**	**27.52**	**30.47**
	Proposed+RL	25.54	26.84	27.65	30.52
	Two-stage training	25.60	26.93	27.71	30.64

F. Loss Function Analysis

Considering that the training process and the loss function significantly impact the training effectiveness of the model, we adopt two different training methods and compares them with the proposed training approach: (1) Proposed+RL: Given that reconstruction loss [82] has been widely applied in MTSF with missing values, we incorporate the reconstruction loss for IA. IA restores the missing values to usable numerals, rather than a high-dimensional representation. (2) Two-stage training [83]: We first train the encoder of GinAR+ using contrastive loss, and then use MAE loss to optimize GinAR+. Table VI shows the RMSE values of different Loss functions (The best results are shown in **bold**). We can draw the following conclusions: (1) Compared to using reconstruction loss, directly mapping missing values to high-dimensional

TABLE VII
PARAMETERS OF THE BLOCK MISSING SCENARIO.

Config	Missing rates			
	25%	50%	75%	90%
Size of blocks	3	[2,4]	[2,4]	[2,4]
The number of blocks	1	2	3	4

TABLE VIII
MAE VALUES OF DIFFERENT MODELS (BLOCK MISSING)

Datasets	Methods	Missing rates			
		25%	50%	75%	90%
PEMS-BAY	GinAR+	**2.09**	**2.32**	**2.54**	**2.76**
	BiTGraph	2.14	2.37	2.60	2.80
	MTGNN+GRIN	2.25	2.41	2.64	2.88
	iTransformer+CSDI	2.38	2.53	2.81	3.15
PEMS08	GinAR+	**20.15**	**21.79**	**23.15**	**24.21**
	BiTGraph	20.53	21.94	23.45	24.65
	MTGNN+GRIN	20.94	22.24	24.85	25.83
	iTransformer+CSDI	21.58	22.62	25.43	26.87
Electricity	GinAR+	**243.76**	**271.58**	**296.15**	**342.98**
	BiTGraph	246.61	275.86	300.41	347.61
	MTGNN+GRIN	263.75	284.19	316.25	360.61
	iTransformer+CSDI	287.64	311.18	342.63	371.84

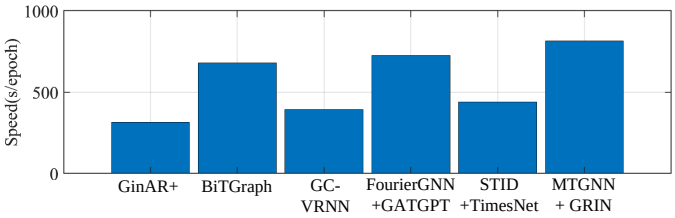


Fig. 5. Training time for each epoch of different models.

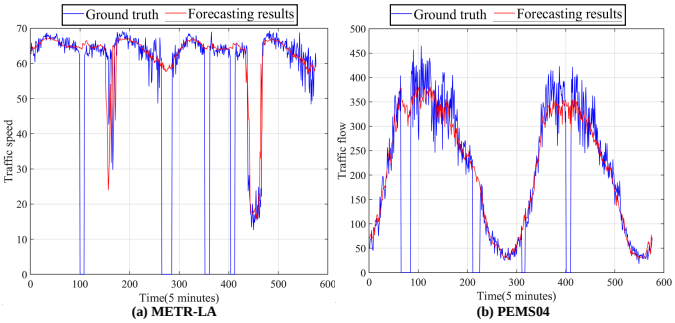


Fig. 6. Visualization of the forecasting results and ground truth.

representations can achieve better forecasting results. The main reason is the error accumulation caused by reconstruction loss. (2) Compared to two-stage training, directly adding all loss functions can achieve better forecasting results. The main reason is that two-stage training involves the issue of memory forgetting, which limits the overall performance of GinAR+.

G. Experiments on the Block Missing Scenario

Evaluating GinAR+'s adaptability to different missing data scenarios can better demonstrate its practical value. Based on related works [72], we conduct experiments on the block missing scenario which indicates that some continuous segments are randomly missing in the time series. Our experimental scenario involves block missing along the temporal dimension, where each time series is independently masked. Table VII shows the specific parameters of the block missing scenario, including the maximum and minimum block sizes, as well as the number of blocks in each historical observation. Moreover, considering that the baselines can be classified into three categories (end-to-end models, two-stage models based on STGNN, and two-stage models based on Transformer), we select the best-performing baseline from each category (BiTGraph, MTGNN+GRIN, and iTransformer+CSDI) and compare them with GinAR+. Table VIII shows the MAE values of different models (The best results are shown in **bold**). The results show that GinAR+ can still achieve the best performance on the block missing scenario, demonstrating its adaptability and superiority.

H. Efficiency

Since the main structure of GinAR+ is obtained by replacing all FC layers in SRU with IA and AGCN, its computational complexity is $O(N^2L^2)$. However, in practical modeling, we employed methods such as Top-K to improve the computational efficiency of GinAR+. In this section, we compare the efficiency of GinAR+ with that of several baselines (BiTGraph, GC-VRNN, FourierGNN+GATGPT, STID+TimesNet and MTGNN+GRIN) on the PEMS08 dataset. Because of the use of contrastive learning, GinAR+ only needs to be trained once to adapt to different missing rates, whereas other baselines require separate training for each missing rate. Therefore, for GinAR+, we directly recorded the training time for a single epoch. For other baselines, we summed up the training time across all missing rates. The experimental equipment is the Intel(R) Xeon(R) Gold 5217 CPU @ 3.00GHz, 128G RAM computing server with RTX 3090 graphics card. The batch size is set to 16. Fig. 5 displays the average training time per epoch for these models. The experimental results show that GinAR+ consumes fewer computational resources.

I. Visualization

To intuitively demonstrate the adaptability of GinAR+ to MTS with missing values, this section visualizes the model's forecasting results and ground truth on the METR-LA and PEMS04 datasets. Fig. 6(a) shows the forecasting results of GinAR+ and the actual traffic speed time series. Fig. 6(b) shows the forecasting results of GinAR+ and the actual traffic flow time series. It can be observed that GinAR+ is not affected by missing values and does not predict zero values, which proves its practical value.

V. CONCLUSION

In this paper, we identify and refine two core challenges that existing models need to address in multivariate time series forecasting with missing values: the tendency to generate incorrect spatial correlations and poor robustness. To generate accurate spatial correlations, we design two key components

(interpolation attention and adaptive graph convolution) and use them to replace all fully connected layers in simple recurrent units. In this way, we propose the Graph Interpolation Attention Recurrent Network, which can simultaneously recover all missing values, correct spatial correlations, and achieve end-to-end forecasting. Additionally, to enhance the robustness of GinAR, we construct positive and negative data pairs using data with different missing rates, and train GinAR using contrastive learning. In this way, we propose GinAR+, which can adapt to data with different missing rates by training only once. Experimental results on nine real-world datasets demonstrate the superiority and practical value of GinAR+.

ACKNOWLEDGMENTS

This work is supported by NSFC No. 62372430, the Postdoctoral Fellowship Program of CPSF under Grant Number GZC20241758, and the Youth Innovation Promotion Association CAS No.2023112.

REFERENCES

[1] H. Xu, Y. Wang, S. Jian, Q. Liao, Y. Wang, and G. Pang, "Calibrated one-class classification for unsupervised time series anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 5723–5736, 2024.

[2] F. Cheng and H. Liu, "Multi-step electric vehicles charging loads forecasting: An autoformer variant with feature extraction, frequency enhancement, and error correction blocks," *Applied Energy*, vol. 376, p. 124308, 2024.

[3] F. Wang, D. Yao, Y. Li, T. Sun, and Z. Zhang, "Ai-enhanced spatial-temporal data-mining technology: New chance for next-generation urban computing," *The Innovation*, vol. 4, no. 2, p. 100405, 2023.

[4] Z. Chen, M. Ma, T. Li, H. Wang, and C. Li, "Long sequence time-series forecasting with deep learning: A survey," *Information Fusion*, vol. 97, p. 101819, 2023.

[5] T. Zhao, S. Wang, C. Ouyang, M. Chen, C. Liu, J. Zhang, L. Yu, F. Wang, Y. Xie, J. Li *et al.*, "Artificial intelligence for geoscience: Progress, challenges and perspectives," *The Innovation*, vol. 5, no. 9, p. 100691, 2024.

[6] X. Qiu, J. Hu, L. Zhou, X. Wu, J. Du, B. Zhang, C. Guo, A. Zhou, C. S. Jensen, Z. Sheng, and B. Yang, "Tfb: Towards comprehensive and fair benchmarking of time series forecasting methods," *Proc. VLDB Endow.*, vol. 17, no. 9, pp. 2363–2377, 2024.

[7] K. Liang, L. Meng, M. Liu, Y. Liu, W. Tu, S. Wang, S. Zhou, X. Liu, F. Sun, and K. He, "A survey of knowledge graph reasoning on graph types: Static, dynamic, and multi-modal," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 9456–9478, 2024.

[8] C. Yu, F. Wang, Y. Wang, Z. Shao, T. Sun, D. Yao, and Y. Xu, "Mgsfformer: A multi-granularity spatiotemporal fusion transformer for air quality prediction," *Information Fusion*, vol. 113, p. 102607, 2025.

[9] D. Cao, Y. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, Y. Tong, B. Xu, J. Bai, J. Tong *et al.*, "Spectral temporal graph neural network for multivariate time-series forecasting," *Advances in neural information processing systems*, vol. 33, pp. 17 766–17 778, 2020.

[10] T. Qian, Y. Chen, G. Cong, Y. Xu, and F. Wang, "Adaptraj: a multi-source domain generalization framework for multi-agent trajectory prediction," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024, pp. 5048–5060.

[11] P. Tang, Q. Zhang, and X. Zhang, "A recurrent neural network based generative adversarial network for long multivariate time series forecasting," in *Proceedings of the 2023 ACM International Conference on Multimedia Retrieval*, 2023, pp. 181–189.

[12] P. Shang, X. Liu, C. Yu, G. Yan, Q. Xiang, and X. Mi, "A new ensemble deep graph reinforcement learning network for spatio-temporal traffic volume forecasting in a freeway network," *Digital Signal Processing*, vol. 123, p. 103419, 2022.

[13] X. Sun, H. Cheng, B. Liu, J. Li, H. Chen, G. Xu, and H. Yin, "Self-supervised hypergraph representation learning for sociological analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 11, pp. 11 860–11 871, 2023.

[14] X. Qiu, X. Wu, Y. Lin, C. Guo, J. Hu, and B. Yang, "Duet: Dual clustering enhanced multivariate time series forecasting," *arXiv preprint arXiv:2412.10859*, 2024.

[15] Z. Shao, T. Qian, T. Sun, F. Wang, and Y. Xu, "Spatial-temporal large models: A super hub linking multiple scientific areas with artificial intelligence," *The Innovation*, vol. 6, no. 2, p. 100763, 2025.

[16] F. Cheng, H. Liu, and X. Lv, "Lithium-ion batteries remaining useful life prediction via fourier-mixed window attention enhanced informer with decomposition and adaptive error correction strategy," *Advanced Engineering Informatics*, vol. 65, p. 103292, 2025.

[17] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 11, pp. 5415–5428, 2021.

[18] F. Zhou, C. Pan, L. Ma, Y. Liu, S. Wang, J. Zhang, X. Zhu, X. Hu, Y. Hu, Y. Zheng *et al.*, "Sloth: Structured learning and task-based optimization for time series forecasting on hierarchies," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 9, 2023, pp. 11 417–11 425.

[19] J. Deng, X. Chen, R. Jiang, X. Song, and I. W. Tsang, "A multi-view multi-task learning framework for multi-variate time series forecasting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 8, pp. 7665–7680, 2022.

[20] Y. Luo, Y. Zhang, X. Cai, and X. Yuan, "E2gan: End-to-end generative adversarial network for multivariate time series imputation," in *Proceedings of the 28th international joint conference on artificial intelligence*. AAAI Press Palo Alto, CA, USA, 2019, pp. 3094–3100.

[21] C. Zheng, X. Fan, C. Wang, J. Qi, C. Chen, and L. Chen, "Increase: Inductive graph representation learning for spatio-temporal kriging," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 673–683.

[22] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," *Advances in neural information processing systems*, vol. 33, pp. 17 804–17 815, 2020.

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

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

[25] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 01, 2020, pp. 1234–1241.

[26] Z. Shao, Z. Zhang, F. Wang, and Y. Xu, "Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 1567–1577.

[27] J. Deng, X. Chen, R. Jiang, X. Song, and I. W. Tsang, "St-norm: Spatial and temporal normalization for multi-variate time series forecasting," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 269–278.

[28] R. Jiang, D. Yin, Z. Wang, Y. Wang, J. Deng, H. Liu, Z. Cai, J. Deng, X. Song, and R. Shibasaki, "Dl-traff: Survey and benchmark of deep learning models for urban traffic prediction," in *Proceedings of the 30th ACM international conference on information & knowledge management*, 2021, pp. 4515–4525.

[29] A. Blázquez-García, K. Wickstrøm, S. Yu, K. Ø. Mikalsen, A. Boubekki, A. Conde, U. Mori, R. Jenssen, and J. A. Lozano, "Selective imputation for multivariate time series datasets with missing values," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 9, pp. 9490–9501, 2023.

[30] X. Ren, K. Zhao, P. J. Riddle, K. Taskova, Q. Pan, and L. Li, "Damr: Dynamic adjacency matrix representation learning for multivariate time series imputation," *Proceedings of the ACM on Management of Data*, vol. 1, no. 2, pp. 1–25, 2023.

[31] P. Wang, T. Zhang, Y. Zheng, and T. Hu, "A multi-view bidirectional spatiotemporal graph network for urban traffic flow imputation," *International Journal of Geographical Information Science*, vol. 36, no. 6, pp. 1231–1257, 2022.

[32] Y. Chen and X. M. Chen, "A novel reinforced dynamic graph convolutional network model with data imputation for network-wide traffic flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 143, p. 103820, 2022.

[33] Y. Chen, Y. Lv, and F.-Y. Wang, "Traffic flow imputation using parallel data and generative adversarial networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1624–1630, 2019.

[34] V. Fortuin, D. Baranchuk, G. Rätsch, and S. Mandt, "Gp-vae: Deep probabilistic time series imputation," in *International conference on artificial intelligence and statistics*. PMLR, 2020, pp. 1651–1661.

[35] X. Liu, Y. Liang, C. Huang, Y. Zheng, B. Hooi, and R. Zimmermann, "When do contrastive learning signals help spatio-temporal graph forecasting?" in *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, 2022, pp. 1–12.

[36] C. Yu, F. Wang, Z. Shao, T. Qian, Z. Zhang, W. Wei, and Y. Xu, "Ginar: An end-to-end multivariate time series forecasting model suitable for variable missing," in *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, 2024, pp. 3989–4000.

[37] M. Su, H. Liu, C. Yu, and Z. Duan, "A novel aqi forecasting method based on fusing temporal correlation forecasting with spatial correlation forecasting," *Atmospheric Pollution Research*, vol. 14, no. 4, p. 101717, 2023.

[38] Y. Chengqing, Y. Guangxi, Y. Chengming, Z. Yu, and M. Xiwei, "A multi-factor driven spatiotemporal wind power prediction model based on ensemble deep graph attention reinforcement learning networks," *Energy*, vol. 263, p. 126034, 2023.

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

[40] L. Chen, D. Chen, Z. Shang, B. Wu, C. Zheng, B. Wen, and W. Zhang, "Multi-scale adaptive graph neural network for multivariate time series forecasting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 10, pp. 10 748–10 761, 2023.

[41] C. Shang, J. Chen, and J. Bi, "Discrete graph structure learning for forecasting multiple time series," in *International Conference on Learning Representations*, 2021.

[42] Z. Shao, Z. Zhang, W. Wei, F. Wang, Y. Xu, X. Cao, and C. S. Jensen, "Decoupled dynamic spatial-temporal graph neural network for traffic forecasting," *Proceedings of the VLDB Endowment*, vol. 15, no. 11, pp. 2733–2746, 2022.

[43] Y. Chen, X. Wang, and G. Xu, "Gatgpt: A pre-trained large language model with graph attention network for spatiotemporal imputation," *arXiv preprint arXiv:2311.14332*, 2023.

[44] Y. Wu, D. Zhuang, A. Labbe, and L. Sun, "Inductive graph neural networks for spatiotemporal kriging," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021, pp. 4478–4485.

[45] I. Marisca, A. Cini, and C. Alippi, "Learning to reconstruct missing data from spatiotemporal graphs with sparse observations," *Advances in neural information processing systems*, vol. 35, pp. 32 069–32 082, 2022.

[46] Y. Chen, Z. Li, C. Yang, X. Wang, G. Long, and G. Xu, "Adaptive graph recurrent network for multivariate time series imputation," in *Neural Information Processing: 29th International Conference, ICONIP 2022, Virtual Event, November 22–26, 2022, Proceedings, Part V*. Springer, 2023, pp. 64–73.

[47] Y. Chen, R. Hu, Z. Li, C. Yang, X. Wang, G. Long, and G. Xu, "Exploring explicit and implicit graph learning for multivariate time series imputation," *Engineering Applications of Artificial Intelligence*, vol. 127, p. 107217, 2024.

[48] S. Shan, Y. Li, and J. B. Oliva, "Nrtsi: Non-recurrent time series imputation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[49] X. Tang, H. Yao, Y. Sun, C. Aggarwal, P. Mitra, and S. Wang, "Joint modeling of local and global temporal dynamics for multivariate time series forecasting with missing values," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 5956–5963.

[50] Y. Xu, A. Bazarjani, H.-g. Chi, C. Choi, and Y. Fu, "Uncovering the missing pattern: Unified framework towards trajectory imputation and prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9632–9643.

[51] X. Chen, X. Li, B. Liu, and Z. Li, "Biased temporal convolution graph network for time series forecasting with missing values." in *The Twelfth International Conference on Learning Representations*, 2023.

[52] K. Liang, Y. Liu, S. Zhou, W. Tu, Y. Wen, X. Yang, X. Dong, and X. Liu, "Knowledge graph contrastive learning based on relation-symmetrical structure," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 1, pp. 226–238, 2023.

[53] J. Deng, X. Chen, R. Jiang, D. Yin, Y. Yang, X. Song, and I. W. Tsang, "Disentangling structured components: Towards adaptive, interpretable and scalable time series forecasting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 8, pp. 3783–3800, 2024.

[54] B. U. Demirel and C. Holz, "Finding order in chaos: A novel data augmentation method for time series in contrastive learning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[55] Y. Yang, C. Zhang, T. Zhou, Q. Wen, and L. Sun, "Dcdetector: Dual attention contrastive representation learning for time series anomaly detection," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 3033–3045.

[56] L. Wu, H. Lin, C. Tan, Z. Gao, and S. Z. Li, "Self-supervised learning on graphs: Contrastive, generative, or predictive," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 4216–4235, 2021.

[57] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi, "Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting," in *International Conference on Learning Representations*, 2021.

[58] X. Zheng, X. Chen, M. Schürch, A. Mollaysa, A. Allam, and M. Krauthammer, "Simple contrastive representation learning for time series forecasting," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 6005–6009.

[59] D. Luo, W. Cheng, Y. Wang, D. Xu, J. Ni, W. Yu, X. Zhang, Y. Liu, Y. Chen, H. Chen *et al.*, "Time series contrastive learning with information-aware augmentations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, 2023, pp. 4534–4542.

[60] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in Neural Information Processing Systems*, vol. 34, pp. 22 419–22 430, 2021.

[61] J. Deng, F. Ye, D. Yin, X. Song, I. Tsang, and H. Xiong, "Parsimony or capability? decomposition delivers both in long-term time series forecasting," *Advances in Neural Information Processing Systems*, vol. 37, pp. 66 687–66 712, 2024.

[62] S. Pachal and A. Achar, "Sequence prediction under missing data: An rnn approach without imputation," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 1605–1614.

[63] J. Tan, H. Liu, Y. Li, S. Yin, and C. Yu, "A new ensemble spatiotemporal pm2. 5 prediction method based on graph attention recursive networks and reinforcement learning," *Chaos, Solitons & Fractals*, vol. 162, p. 112405, 2022.

[64] K. Zhang, C. Li, and Q. Yang, "Trid-mae: A generic pre-trained model for multivariate time series with missing values," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 3164–3173.

[65] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *The Eleventh International Conference on Learning Representations*, 2023.

[66] W. Du, D. Côté, and Y. Liu, "Saits: Self-attention-based imputation for time series," *Expert Systems with Applications*, vol. 219, p. 119619, 2023.

[67] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," in *The Twelfth International Conference on Learning Representations*, 2024.

[68] Y. Tashiro, J. Song, Y. Song, and S. Ermon, "Csdi: Conditional score-based diffusion models for probabilistic time series imputation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 24 804–24 816, 2021.

[69] T. Zhou, P. Niu, L. Sun, R. Jin *et al.*, "One fits all: Power general time series analysis by pretrained lm," *Advances in neural information processing systems*, vol. 36, pp. 43 322–43 355, 2023.

[70] Y. Li, Z. Shao, Y. Xu, Q. Qiu, Z. Cao, and F. Wang, "Dynamic frequency domain graph convolutional network for traffic forecasting," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 5245–5249.

[71] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 753–763.

[72] A. Cini, I. Marisca, and C. Alippi, "Filling the g_ap_s: Multivariate time series imputation by graph neural networks," in *International Conference on Learning Representations*, 2022.

[73] Z. Shao, Z. Zhang, F. Wang, W. Wei, and Y. Xu, "Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting," in *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, 2022, Conference Proceedings, pp. 4454–4458.

[74] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," in *The Eleventh International Conference on Learning Representations*, 2023.
[75] K. Yi, Q. Zhang, W. Fan, H. He, L. Hu, P. Wang, N. An, L. Cao, and Z. Niu, "Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective," *Advances in neural information processing systems*, vol. 36, pp. 69 638–69 660, 2023.
[76] Y. Wang, H. Wu, J. Dong, Y. Liu, M. Long, and J. Wang, "Deep time series models: A comprehensive survey and benchmark," *arXiv preprint arXiv:2407.13278*, 2024.
[77] Z. Shao, F. Wang, Y. Xu, W. Wei, C. Yu, Z. Zhang, D. Yao, T. Sun, G. Jin, X. Cao *et al.*, "Exploring progress in multivariate time series forecasting: Comprehensive benchmarking and heterogeneity analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 37, no. 1, pp. 291–305, 2025.
[78] R. Jiang, Z. Wang, J. Yong, P. Jeph, Q. Chen, Y. Kobayashi, X. Song, S. Fukushima, and T. Suzumura, "Spatio-temporal meta-graph learning for traffic forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 7, 2023, pp. 8078–8086.
[79] J. Cheng, C. Yang, W. Cai, Y. Liang, and Y. Wu, "Nuwats: Mending every incomplete time series," *arXiv preprint arXiv:2405.15317*, 2024.
[80] J. Dong, H. Wu, H. Zhang, L. Zhang, J. Wang, and M. Long, "Simmtm: A simple pre-training framework for masked time-series modeling," *Advances in Neural Information Processing Systems*, vol. 36, pp. 29 996–30 025, 2023.
[81] H. Liu, C. Yu, H. Wu, Z. Duan, and G. Yan, "A new hybrid ensemble deep reinforcement learning model for wind speed short term forecasting," *Energy*, vol. 202, p. 117794, 2020.
[82] T. H. Tran, L. M. Nguyen, K. Yeo, N. Nguyen, D. Phan, R. Vaculin, and J. Kalagnanam, "An end-to-end time series model for simultaneous imputation and forecast," *arXiv preprint arXiv:2306.00778*, 2023.
[83] S. Mukherjee and A. H. Awadallah, "Xtremedistil: Multi-stage distillation for massive multilingual models," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 2221–2234.

Chengqing Yu received the B.S. degree in Transport Equipment and Control Engineering from Central South University, Changsha, China, in 2019 and M.S. degree in Traffic and Transportation Engineering with Central South University, Changsha, China, in 2022, respectively. He is now PHD student in Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. His main research interests include deep learning, reinforcement learning, and time series forecasting.

Fei Wang , born in 1988, PhD, associate professor. He received the B.S. degree in computer science from the Beijing Institute of Technology, Beijing, China, in 2011. He received the PhD degree in computer architecture from Institute of Computing Technology, Chinese Academy of Sciences in 2017. From 2017 to 2020, he was a research assistant with the Institute of Technology, Chinese Academy of Sciences. Since 2020, he has been working as associate professor in Institute of Computing Technology, Chinese Academy of Sciences. His main research interest includes spatiotemporal data mining, Information fusion, graph neural networks.

Zezhi Shao is currently pursuing a Ph.D. degree with the Institute of Computing Technology, Chinese Academy of Sciences, China. He received the B.E. degree from Shandong University, Jinan, China, in 2019. His research interests include traffic condition forecasting, multivariate time series forecasting, graph neural networks, and spatial-temporal data mining. He has published many papers as the first author in top journals and conferences such as TKDE, KDD, VLDB, CIKM.

Tangwen Qian is currently an associate researcher at Institute of Computing Technology, Chinese Academy of sciences. She received her B.E. degree in computer science and technology from Beijing Institute of Technology, Beijing, China in 2017, and Ph.D. degree from Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China in 2024. Her research interests include spatial-temporal data mining and multi-agent trajectory prediction.

Zhao Zhang is currently an associate professor at the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. He received the BE degree in computer science and technology from the Beijing Institute of Technology (BIT), Beijing, China, in 2015, and the PhD degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2021. His research interests include data mining and applied machine learning, with a special focus on the representation and application of knowledge graphs.

Wei Wei received the PhD degree from the Huazhong University of Science and Technology, Wuhan, China, in 2012. He is currently a professor with the Department of Computer of Science and Technology, Huazhong University of Science and Technology. He was a research fellow with Nanyang Technological University, Singapore, and Singapore Management University, Singapore. His current research interests include information retrieval, natural language processing, social computing and recommendation, cross-modal/multimodal computing, deep learning, machine learning and artificial intelligence.

Zhulin An received the BEng and MEng degrees in computer science from the Hefei University of Technology, Hefei, China, in 2003 and 2006, respectively and the PhD degree from the Chinese Academy of Sciences, Beijing, China, in 2010. He is currently with the Institute of Computing Technology, Chinese Academy of Sciences, where he became an associate researcher in 2014. His current research interests include deep neural network acceleration and continual learning.

Qi Wang is an associate professor at Institute of Computing Technology, Chinese Academy of Sciences (ICT-CAS) in Beijing, China. She received the Ph.D. degree in computer science from University of Chinese Academy of Sciences, Beijing, China in 2015. In 2010, she received a 1-year fellowship from INRIA under the joint program with the Chinese Academy of Sciences to pursue her research within the SWING team of INRIA, at CITI laboratory, INSA Lyon, France. She visited IRIT laboratory at University of Toulouse because she was a recipient of the 2012 EIFFEL doctoral fellowship from the French Ministry of Foreign Affairs. Her research focuses on performance evaluation and optimization of wireless networks for delay sensitive applications.

Yongjun Xu is a professor at Institute of Computing Technology, Chinese Academy of Sciences (ICT-CAS) in Beijing, China. He received his B.Eng. and Ph.D. degree in computer communication from Xi'an Institute of Posts & Telecoms (China) in 2001 and Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China in 2006, respectively. His current research interests include artificial intelligence systems, and big data processing.