GinAR: An End-To-End Multivariate Time Series Forecasting Model Suitable for Variable Missing

Chengqing Yu
Fei Wang*
Zezhi Shao
Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China
University of Chinese Academy of Sciences,
Beijing, China
{yuchengqing22b,wangfei,Shaozezhi19b}@ict.ac.cn

Tangwen Qian
Zhao Zhang
Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China
{qiantangwen,zhangzhao2021}@ict.ac.cn

Wei Wei
School of Computer Science and Technology, Huazhong
University of Science and Technology,
Wuhan, China
weiw@hust.edu.cn

Yongjun Xu*
Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China
xyj@ict.ac.cn

ABSTRACT

Multivariate time series forecasting (MTSF) is crucial for decision-making to precisely forecast the future values/trends, based on the complex relationships identified from historical observations of multiple sequences. Recently, Spatial-Temporal Graph Neural Networks (STGNNs) have gradually become the theme of MTSF model as their powerful capability in mining spatial-temporal dependencies, but almost of them heavily rely on the assumption of historical data integrity. In reality, due to factors such as data collector failures and time-consuming repairment, it is extremely challenging to collect the whole historical observations without missing any variable. In this case, STGNNs can only utilize a subset of normal variables and easily suffer from the incorrect spatial-temporal dependency modeling issue, resulting in the degradation of their forecasting performance. To address the problem, in this paper, we propose a novel Graph Interpolation Attention Recursive Network (named GinAR) to precisely model the spatial-temporal dependencies over the limited collected data for forecasting. In GinAR, it consists of two key components, that is, interpolation attention and adaptive graph convolution to take place of the fully connected layer of simple recursive units, and thus are capable of recovering all missing variables and reconstructing the correct spatial-temporal dependencies for recursively modeling of multivariate time series data, respectively. Extensive experiments conducted on five real-world datasets demonstrate that GinAR outperforms 11 SOTA baselines, and even when 90% of variables are missing, it can still accurately predict the future values of all variables.

*Fei Wang and Yongjun Xu are the corresponding authors.

CCS CONCEPTS

• **Information systems** → **Data mining**.

KEYWORDS

Multivariate time series forecasting, Variable missing, Adaptive graph convolution, Interpolation attention, Graph Interpolation Attention Recursive Network

1 INTRODUCTION

Multivariate time series forecasting (MTSF) is widely used in practice, such as transportation [48], environment [56] and others [71]. It predicts future values of multiple interlinked time series by using their historical observations, and contributes to decision-making [59, 62, 72]. Indeed, multivariate time series (MTS) can be formalized as a kind of classical spatial-temporal graph data [3], such as traffic flow [18], each variable of which is collected in chronological order, using a sensor deployed at an independent position. Naturally, they usually have two key factors, temporal dependency [63] and spatial correlation [57]. The former characterizes complex patterns (e.g., causal relationships) of instances in chronological order, and the later depicts the differences of time series corrected each other in spatial dimension. Therefore, effectively mining spatial-temporal dependencies is crucial for MTSF to precisely predict future values of the time series, or to better understand of how they interact [44, 54, 55].

Recently, 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 [10], but their superior performances heavily rely

(a) MTSF   (b) MTSF with missing values   (c) MTSF with variable missing (ours)

(d) Forecasting results of TGCN under different missing rates
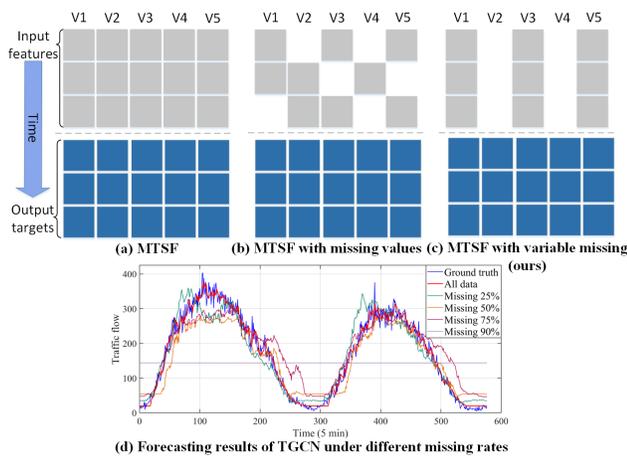
Figure 1: The principle and examples of multivariate time series forecasting with variable missing. V1 to V5 represent different variables. Compared to the other two tasks, our task can only use historical observations of certain variables to predict the future values of all variables. The forecasting performance of TGCN declines as the missing rate increases.

on the data quantity [83]. Since the time series data in practice is always incomplete, it is very challenging to obtain whole historical observations of all variables for accurate forecasting [42]. To make things worse, the data from some variables may be even unavailable for a long time under certain conditions [11]. We can take a classical MTS application (i.e., air quality forecasting) as an example, the data collectors may easily work anomaly owing to some unforeseen factors (e.g., horrible weather) [43]. Because equipment maintenance usually takes days or even months, corresponding data collectors only output outliers for a long time [75]. Thus, STGNNs need to address a problem, namely, whole history observations missing of some variables. This means that STGNNs only achieve MTSF using the remaining normal variables (shown in Figure 1 (c)), which severely limits their performance. To alleviate this problem, some works [6] only predict values of remaining normal variables by discarding all missing variables. However, if missing variables are key samples (e.g., important locations like hub nodes), the inability to predict their values will profoundly affect decision-making [64].

The above phenomenon shows that MTSF faces a significant practical challenge: **how to forecast MTS when missing part of variables?** By rethinking the characteristics of STGNNs and this task, the main problem is that STGNNs easily capture incorrect spatial-temporal dependencies during the modeling process, resulting in error accumulation and degraded forecasting performance. On the one hand, since each missing variable is usually a straight-line sequence composed of outliers, the sequence model in STGNNs [16] cannot mine any valuable pattern and information, resulting in incorrect temporal dependencies. On the other hand, existing STGNNs [22, 27, 34] need to use historical observations from all variables to construct spatial correlations. Because whole history observations of some variables are missing, existing STGNNs cannot establish spatial correlations between missing and

normal variables, leading to incorrect spatial correlations. In this case, as the missing rate increases, the above phenomena become more serious, leading to a significant decline in the performance of STGNNs. For example, a classic STGNN model, temporal graph convolutional network (TGCN) [81], is used for further analysis when given different missing rates on PEMS04. Figure 1 (d) shows that its performance deteriorates while increasing the missing rate.

At present, an intuitive strategy for addressing this challenge is to combine imputation and forecasting methods and propose two-stage models [33]. However, classic imputation methods [8, 65] primarily rely on the context information of time series to recover missing values. When history observations from some variables are unavailable for a long time, these methods cannot achieve reliable recovery effects since the missing variables do not have any normal value in the temporal dimension. In addition to the above classical methods, existing mainstream imputation methods [2, 46] combine the context information and spatial correlations of MTS for generating plausible missing values. However, they also have two problems: (1) Components that use context information in these imputation methods also introduce incorrect temporal dependencies, limiting the effectiveness of data recovery and leading to error accumulation [6]. (2) these imputation methods mainly rely on fixed spatial correlations (such as road network structure) to establish correspondences between missing variables and normal variables [73]. When the missing rate is significant, they cannot fully use all normal variables to recover missing variables, resulting in the ineffective recovery of missing variables that do not correspond with normal variables [79]. In general, due to the introduction of incorrect temporal dependencies and the lack of sufficient correspondences between missing variables and normal variables, two-stage models cannot work well in MTSF with variable missing.

To solve the above problems and realize MTSF with variable missing, forecasting models need to fully utilize historical observations of all normal variables to correct spatial-temporal dependencies during the modeling process. To this end, we propose an end-to-end framework called Graph Interpolation Attention Recursive Network (GinAR). Specifically, we use simple recursive units (SRU) based on the RNN framework as the backbone and propose two key components (interpolation attention (IA) and adaptive graph convolution (AGCN)) to replace all fully connected layers in SRU. This is done to realize end-to-end forecasting while correcting spatial-temporal dependencies. On the one hand, during the process of recursive modeling, for data at each time steps, IA first generates correspondences between normal variables and missing variables, then uses attention to restore all missing variables to plausible representations. In this way, the sequence model avoids directly mining missing variables that do not have any valuable patterns, thereby correcting temporal dependencies. On the other hand, for representations processed by IA, we use AGCN to reconstruct spatial correlations between all variables. Since all missing variables are recovered, AGCN can more accurately utilize their representations to generate a more reliable graph structure and obtain more accurate spatial correlations. In this way, GinAR mines more accurate spatial-temporal dependencies in the process of recursive modeling and effectively avoids the error accumulation problem. Thus, GinAR can implement MTSF with variable missing more accurately. **The main contributions of this paper are as follows:**

- To the best of our knowledge, this is the first work that challenges to achieve MTSF with variable missing. The proposed end-to-end framework can address the problem of error accumulation in the modeling process.
- To achieve this challenging task, 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-temporal dependencies during the process of recursive modeling.
- We design experiments on five real-world datasets. Results show that GinAR can outperform 11 baselines on all datasets. Even when 90% of variables are missing, it can still accurately predict the future values of all variables.

2 RELATED WORKS

2.1 Spatial-Temporal Forecasting Method

STGNNs combine the advantages of GCN [32] and sequence models [25] to fully mine spatial-temporal dependencies of MTS, and further improve the ability of spatial-temporal forecasting [15, 41, 51]. Li et al. [35] combine gated recursive unit (GRU) and GCN to propose the diffused convolutional recurrent neural network (DCRNN) and realize MTSF. Wu et al. [69] propose the graph wavenet (GWNET) by combining temporal convolutional network (TCN) and GCN. Compared with traditional methods, above two models achieves excellent results. However, these methods ignore hidden spatial correlations between variables, which limits their effectiveness [32]. To further improve the ability of STGNN to mine spatial correlations, graph learning has been widely studied [7]. Zheng et al. [82] design a spatial attention mechanism to learn attention scores by considering traffic features and variable embeddings in the graph structure. Shang et al. [47] use historical observations of all variables to learn the discrete probability graph structure. Shao et al. [52] propose decoupled spatial-temporal framework and dynamic graph learning to explore spatial-temporal dependencies between variables. Although STGNNs have made significant progress in MTSF, they need to use the variable features or prior knowledge to mine spatial-temporal dependencies [20, 30, 53]. However, in MTSF with variable missing, the graph structure based on prior knowledge and the graph learning based on variable features are affected by missing variables, which leads to inaccurate modeling of spatial correlations [21, 36, 76].

2.2 Imputation Method

Existing imputation methods include classical models [60] and deep learning-based models [19, 23, 24]. Compared with traditional models, deep learning can analyze hidden correlations between missing and normal data and improve performance [77]. Wu et al. [65] combine the matrix transformation with CNN to realize the missing data imputation, but it ignores correlations between different variables, limiting its performance. Mariscara et al. [28] combine cross-attention and temporal attention to achieve the recovery of missing data, but they do not take full advantage of the spatial correlations between variables, which leads to inadequate data recovery. In addition to the above methods, GNN-based methods [38] combine

GCN and sequence models to analyze spatial-temporal dependencies between missing data and normal data, and further recover all missing data [61]. Wu et al. [67] propose inductive graph neural network to recover missing data. Compared with classical methods, the proposed model has better performance. Chen et al. [12] use the adaptive graph recursive network to realize the imputation of missing data. Experiments show that the framework combining graph convolution with recurrent neural networks can better use temporal information and spatial correlation to recover missing data. Although imputation methods can recover missing data and improve the performance of forecasting models, they often suffer from several problems: (1) Classical imputation methods [1] need to reconstruct both missing data and normal data, resulting in the loss of effective information. (2) Existing imputation methods [4, 45] require full use of temporal information to recover missing data. When the data from some variables are unavailable for a long time, existing methods introduce incorrect temporal dependencies, resulting in limited recovery performance [13, 37].

3 METHODOLOGY

3.1 Preliminaries

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: $A \in R^{N*N}$.

Multivariate time series forecasting. Given a historical observation tensor $X \in R^{N*H*C}$ from H time slices in history, the model can predict the value $Y \in R^{N*L}$ 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 $X \in R^{N*H*C}$ and $Y \in R^{N*L}$.

Multivariate time series forecasting with variable missing. Compared with MTSF, the main difference of this task is that there are some variables with whole history data missing in historical observations $X \in R^{N*H*C}$. Thus, we mask M variables randomly from N variables of the historical observation $X \in R^{N*H*C}$. The values of these M variables are treated as 0, i.e. missing values and a new input feature $X_M \in R^{N*H*C}$ is obtained. The core goal of this task is to construct a mapping function between input $X_M \in R^{N*H*C}$ and output $Y \in R^{N*L}$.

3.2 Overall Framework of GinAR

The framework of GinAR is shown in Figure 2, which uses multiple GinAR layers as the encoder and MLP as the decoder. The GinAR layer adopts the idea of recursive modeling, and its core structure is the GinAR cell. By transmitting input features with variable missing to GinAR, it can predict future values of all variables. Next, we briefly introduce the design motivation of GinAR and the function of its components.

Firstly, we intuitively discuss the design idea for the GinAR cell. Specifically, we use IA and AGCN to replace all fully connected layers in SRU. IA can use normal variables to restore the missing variables to plausible representations, which can help the sequence model to better mine temporal dependencies of missing variables.
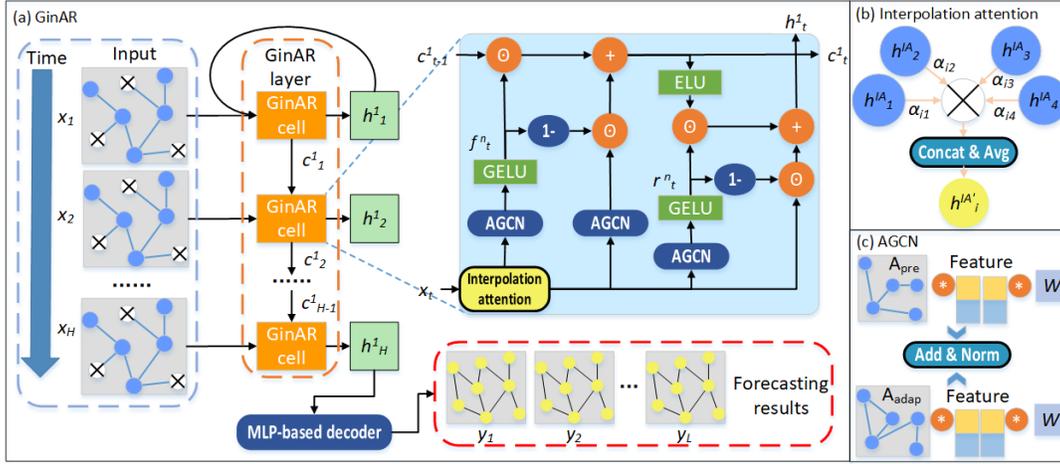
Figure 2: (a) The overall framework of GinAR. The GinAR layer adopts the RNN-based sequence framework and encodes historical observations of MTS with variable missing. The MLP-based decoder is used to predict future values of all variables. (b) The specific structure of the interpolation attention. (c) The specific structure of the adaptive graph convolution.

Furthermore, for all variables processed by IA, their spatial correlations cannot be determined by a predefined graph based on prior knowledge. Therefore, we use AGCN, which introduces graph learning, to reconstruct spatial correlations between all variables.

Then, we briefly discuss the encoder, which uses the recursive modeling framework. Specifically, at each time step T, the input features $x_T \in R^{N*C}$ of the current moment and the cell state c_{T-1} of the previous moment are transmitted to the GinAR cell. Then, the GinAR cell outputs the cell state c_T for the next cell and obtains the hidden feature h_T. In this way, the GinAR layer utilizes the GinAR cell to restore missing variables and reconstruct spatial correlations, while simultaneously capturing temporal dependencies through the recursive modeling framework. Besides, due to the introduction of skip connections in the GinAR cell, stacking multiple GinAR layers can capture deeper hidden information.

Finally, we discuss the decoder and the forecasting process. An important step in the forecasting process is to properly filter the hidden features obtained by the encoder. On the one hand, since the encoder takes the form of recursive modeling, the last hidden state of each GinAR layer contains all the information from the historical observation [31]. On the other hand, due to the introduction of skip connections, the hidden features obtained by each GinAR layer contains different information [26]. Therefore, we concatenate the last hidden state of all GinAR layers and use the concatenated tensor as the input to the decoder. Besides, we use the MLP, which is based on the direct multi-step (DMS) forecasting strategy [66], to predict future changes for all nodes. Compared with decoders based on auto-regressive [5] or iterated multi-step (IMS) [40] forecasting, the proposed method can solve the problem of error accumulation and improve the forecasting accuracy.

3.3 Interpolation Attention

For each missing variable, interpolation attention needs to select the normal variables for induction and give the corresponding weight for the selected normal variables. Thus, it contains two main steps:

(1) It first generates correspondences between missing variables and normal variables. (2) Based on above correspondences, attention is used to realize the induction of missing variables. The main schematic diagram of interpolation attention is shown in Figure 3. The specific modeling steps of IA are shown below:

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

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

where, softmax$(\cdot)$ is the activation function. ReLU$(\cdot)$ is the activation function. $A_{IA} \in R^{N*N}$ 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 variable i and the variable j. In other words, the interpolation attention can use the normal variable j to recover the missing variable i. Based on the above variable correlation matrix $A_{IA} \in R^{N*N}$, we can obtain the set of normal variables $N(i)$ associated with the missing variable i.

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

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

where, $FC(\cdot)$ is the fully connected layer. h_j^{IA} and W_j^{IA} represent the features and weight of variable 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 variables to achieve
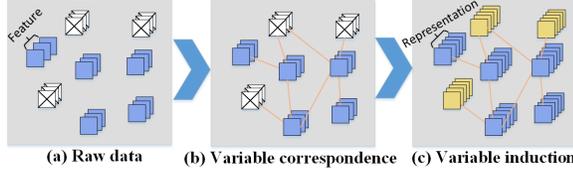
Figure 3: The schematic diagram of IA. Blue represents normal variables. White represents missing variables. Yellow represents the variables after induction.

(a) Raw data   (b) Variable correspondence   (c) Variable induction

the recovery of the missing variable i.

$$h_i^{IA'} = \text{ReLU}\left(\sum_{j \in N(i)} \alpha_{ij} W_{ij}^{IA} h_j^{IA}\right), \qquad (3)$$

Step 4: Repeat steps 2 through 3 until all missing variables are restored. At this time, all variables have representations in the new tensor obtained by the IA method. The original input feature $X_M \in R^{N*H*C}$ can be converted to $X_M^{IA} \in R^{N*H*C'}$. (Note: This step is completed by using matrix multiplication for parallel computation.)

3.4 Adaptive Graph Convolution

By introducing prior knowledge to define an adjacency matrix A, the predefined graph can help models establish a basic spatial correlation. However, for MTSF with missing variables, the predefined graph cannot adequately model the spatial correlation of all variables due to a large number of missing variables. To this end, we propose the data-based adaptive graph convolution, which consists of the predefined graph and the adaptive graph.

Predefined graph: In this paper, 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 [56] is used to form the adjacency matrix A. The predefined graph $A_{pre} \in R^{N*N}$ for the graph convolution network is obtained by the following formula:

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

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

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

$$E_n = FC(\text{concat}(W_x X_M^{IA}, W_e E_A)), \qquad (5)$$

where, W_x and W_e represent the weights of the variable representation $X_M^{IA} \in R^{N*H*C'}$ obtained by the interpolation attention and the variable-embedded matrix $E_A \in R^{N*d}$, respectively. concat($\cdot$) means concatenate two tensors. The adaptive graph can be obtained by the following formula:

$$A_{adap} = (I_N + \text{softmax}(\text{GeLU}(E_n E_n^T)), \qquad (6)$$

where, E_n^T represents the transpose of E_n.

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:

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

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.

3.5 GinAR

The main idea of GinAR is to integrate the proposed interpolation attention and adaptive graph convolution 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 variables. 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:

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

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

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

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

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

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 $X \in R^{N*H*C}$ is preprocessed and the input feature $X_M \in R^{N*H*C}$ for modeling is obtained. The values of the M variables in the N variables of the input feature $X_M \in R^{N*H*C}$ is 0.

$$X_M = [x_1, x_2, ..., x_H], x \in R^{N*C}, \qquad (13)$$

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. Each GinAR layer 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.

$$h^1 = [h_1^1, h_2^1, ..., h_H^1], \qquad (14)$$

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_{all}^n, which is shown as follows:

$$h_{all}^n = [h_H^1, h_H^2, ..., h_H^n], h_{all}^n \in R^{N*C'*n}, \qquad (15)$$

Step 6: h_{all}^n is passed to the MLP-based generative decoder. And the final forecasting result $Y \in R^{N*L}$ is obtained.

$$Y = FC(\text{ReLU}(FC(h_{all}^n))), \qquad (16)$$

4 EXPERIMENTAL STUDY

4.1 Experimental Design

Datasets. Five 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] and an air quality dataset (China AQI)[3].

Baselines. In order to fully compare and analyze the performance of the proposed GinAR, eleven existing SOTA methods are selected as the main baselines, which include forecasting models (MegaCRN [29], DSformer [78] and STID [50]), and forecasting models with data recovery components (LGnet [58], TriD-MAE [80], GC-VRNN [70], and BiTGraph [9]). Besides, we design two-phase models (DCRNN [35] + GPT4TS [84], DFDGCN [34] + TimesNet [65] and MTGNN [68] + GRIN [17], FourierGNN [74]+GATGPT[14]) as additional baselines to further demonstrate GinAR's effect.

Setting. Table 1 shows the main hyperparameters of the proposed model. We design experiments from the following aspects: (1) Our code is available at this link[4] . (2) All datasets are uniformly divided into training sets, validation sets and test sets according to the ratio in the reference [49]. (3) We set the history/future length based on the existing work [29, 34]. The history length and future length of GinAR are both 12. Metrics are the average of 12-step forecasting results. (4) We randomly set mask variables according to the ratio of 25%, 50%, 75% and 90%. Values of the masked variable are uniformly treated as 0. Besides, the experiment was repeated with 5 different random seeds for each missing rate. The final metrics are the mean values of repeated experiments. (5) To ensure the fairness of experiments, we train the two-stage model in two ways: first, train the two models separately. Second, based on the

[1]https://github.com/liyaguang/DCRNN
[2]https://github.com/guoshnBJTU/ASTGNN/tree/main/data
[3]https://quotsoft.net/air/
[4]https://github.com/ChengqingYu/GinAR

Table 1: Values of the corresponding hyperparameters for different missing rate.

Config	Values (25%, 50%, 75%, 90%)
loss function	L1 Loss
optimizer	Adam
learning rate	0.006
embedding size	32/32/16/16
variable embedding size	16/16/8/8
number of layers	2/2/3/3
dropout	0.15
learning rate schedule	MultiStepLR
clip gradient normalization	5
milestone	[1,15,40,70,90]
gamme	0.5
batch size	16
epoch	100

reference [70], the two models are spliced together for training. The final metrics are the optimal results.

Metrics. To comprehensively evaluate the forecasting performance of different models, this paper utilizes three classical metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) [39].

4.2 Main Results

Table 2 gives the performance comparison results of all baselines and GinAR on five datasets (The best results are shown in **bold**). Based on results, the following conclusions can be obtained: (1) Compared with SOTA forecasting models, all two-stage forecasting models can achieve better forecasting results. The main reason is that imputation methods use normal variables to recover missing variables, which reduces the impact of missing variables on the forecasting model. However, the error accumulation problem exists in two-stage models, which limits the performance of the downstream predictors. (2) The forecasting models with data recovery components can work better than other baselines. On the one hand, they address the problem that one-stage models cannot handle missing data. On the other hand, they avoid the error accumulation problem of two-stage models. (3) GinAR can achieve optimal experimental results on all datasets and all settings. Based on interpolation attention, adaptive graph convolution and RNN-based framework, the GinAR can realize missing variable recovery, spatial-temporal correlation reconstruction and end-to-end forecasting. Compared with one-stage models and two-stage models, GinAR can avoid the problem of error accumulation and produce more accurate spatial-temporal dependencies. Therefore, GinAR can achieve better results than all baselines in MTSF with variable missing. To further evaluate the effects of each component in GinAR, we conduct ablation experiments. Besides, to demonstrate the effect of the end-to-end framework, we analyze the performance recovery effect of interpolation attention on MLP-based models.

Table 2: Performance comparison results of all baselines and the proposed model on all 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	STID	10.31	17.85	5.05	10.73	18.88	5.35	11.35	20.57	5.84	12.06	23.17	6.24
	DSformer	7.69	11.05	4.02	8.27	12.78	4.38	9.21	14.59	4.77	10.15	17.69	5.02
	MegaCRN	7.43	10.47	3.81	7.87	11.02	3.94	8.28	12.13	4.24	8.72	13.54	4.58
	DCRNN+GPT4TS	7.41	10.72	3.78	7.91	11.61	3.98	8.16	11.93	4.15	8.31	12.18	4.29
	DFDGCN+TimesNet	7.42	10.42	3.72	7.68	11.45	3.89	8.11	11.75	4.14	8.33	12.24	4.34
	MTGNN+GRIN	7.28	10.48	3.69	7.43	11.22	3.77	8.05	11.58	4.12	8.29	12.14	4.25
	FourierGNN+GATGPT	7.40	10.87	3.71	7.84	11.75	3.82	8.25	12.25	4.21	8.37	12.28	4.33
	LGnet	7.52	10.97	3.95	8.03	11.83	4.17	8.52	13.09	4.42	9.15	14.38	4.79
	GC-VRNN	7.04	10.51	3.68	7.73	10.98	3.87	8.19	11.71	4.17	8.35	12.29	4.32
	TriD-MAE	7.15	10.37	3.64	7.58	11.07	3.79	7.92	11.13	3.92	8.22	11.92	4.11
	BiTGraph	6.74	10.25	3.61	7.32	10.79	3.69	7.63	11.04	3.74	8.03	11.78	3.91
	Proposed	**6.55**	**10.12**	**3.56**	**7.14**	**10.42**	**3.61**	**7.39**	**10.71**	**3.70**	**7.84**	**11.25**	**3.87**
PEMS-BAY	STID	7.13	7.78	3.01	7.86	8.21	3.39	8.26	9.24	3.51	8.65	10.07	3.78
	DSformer	6.32	7.15	2.91	6.46	7.73	3.08	8.15	9.06	3.45	9.06	10.23	3.72
	MegaCRN	5.93	7.03	2.85	7.36	7.75	3.02	7.75	8.77	3.35	8.25	9.23	3.54
	DCRNN+GPT4TS	5.54	6.24	2.63	6.22	6.75	2.87	7.14	7.82	3.09	7.69	8.82	3.31
	DFDGCN+TimesNet	5.39	6.17	2.58	6.17	7.03	2.74	6.96	7.59	3.07	7.15	8.34	3.27
	MTGNN+GRIN	5.78	5.73	2.65	6.19	6.55	2.83	7.06	7.75	3.02	7.26	8.45	3.22
	FourierGNN+GATGPT	5.21	5.46	2.40	5.93	5.96	2.71	6.85	7.58	2.98	7.44	9.07	3.28
	LGnet	6.02	7.19	2.88	6.74	8.15	3.14	8.08	9.18	3.43	8.92	9.83	3.67
	GC-VRNN	4.93	5.37	2.39	5.34	5.86	2.64	6.08	6.94	2.87	7.32	7.94	3.12
	TriD-MAE	5.17	5.48	2.46	5.53	6.12	2.69	5.97	6.85	2.79	7.02	7.63	3.02
	BiTGraph	4.52	5.16	2.17	5.06	6.07	2.44	5.79	6.68	2.61	6.75	7.42	2.83
	Proposed	**4.34**	**4.90**	**2.10**	**4.78**	**5.88**	**2.35**	**5.48**	**6.17**	**2.54**	**6.43**	**6.94**	**2.77**
PEMS04	STID	71.39	50.88	41.96	95.09	93.63	63.73	113.02	124.11	82.47	123.30	149.68	94.72
	DSformer	50.15	20.88	32.86	51.51	21.35	33.31	54.91	23.25	37.28	59.18	25.62	40.31
	MegaCRN	42.22	20.18	28.26	47.07	21.29	31.48	47.95	22.03	33.58	52.17	23.42	36.14
	DCRNN+GPT4TS	40.03	17.77	25.17	41.64	18.21	26.56	43.71	19.18	28.54	46.17	21.27	31.42
	DFDGCN+TimesNet	39.48	17.40	24.43	41.18	18.49	26.09	42.81	19.91	28.29	45.93	21.43	30.98
	MTGNN+GRIN	39.67	18.71	24.84	41.91	19.18	26.95	44.36	20.90	28.04	45.88	21.04	30.61
	FourierGNN+GATGPT	40.92	19.61	25.58	42.35	20.55	27.31	45.17	22.20	29.87	48.83	23.65	32.16
	LGnet	42.64	18.42	26.53	46.39	21.30	30.81	52.05	24.83	33.94	55.12	23.74	36.29
	GC-VRNN	39.75	16.82	23.57	41.34	17.83	26.43	43.82	18.67	27.72	45.12	20.43	30.06
	TriD-MAE	39.83	16.98	24.15	40.65	17.52	25.89	41.90	18.04	26.95	44.23	20.05	29.54
	BiTGraph	38.94	16.75	23.01	40.03	17.34	24.15	41.69	17.92	26.33	43.37	19.08	28.71
	Proposed	**38.22**	**16.45**	**22.52**	**39.02**	**17.04**	**23.78**	**41.53**	**17.58**	**25.98**	**42.82**	**18.31**	**28.20**
PEMS08	STID	58.81	31.71	32.88	79.41	51.74	49.71	101.63	74.47	69.06	113.20	86.54	81.64
	DSformer	38.38	19.24	27.74	42.49	23.79	30.47	51.57	25.18	35.21	55.34	32.61	38.79
	MegaCRN	39.29	17.42	26.04	43.43	21.30	30.68	48.37	21.34	32.23	52.75	24.64	34.52
	DCRNN+GPT4TS	36.58	15.96	24.64	41.79	18.73	27.96	44.82	19.79	29.62	46.22	22.96	31.78
	DFDGCN+TimesNet	36.29	15.56	24.26	39.05	19.42	25.39	42.67	20.74	28.30	45.83	21.59	30.46
	MTGNN+GRIN	36.65	15.17	24.08	37.64	17.41	25.78	40.51	18.52	27.45	42.79	21.33	29.15
	FourierGNN+GATGPT	35.54	15.35	23.77	37.42	16.86	25.53	39.44	17.95	26.89	41.44	20.61	29.11
	LGnet	37.54	22.18	26.51	47.23	21.91	32.04	50.38	21.43	33.65	52.06	25.71	35.48
	GC-VRNN	35.17	14.69	23.25	36.40	15.85	24.27	39.67	16.06	26.32	41.98	20.54	28.46
	TriD-MAE	33.15	14.25	21.53	35.95	15.32	23.18	37.64	15.58	24.42	39.25	16.43	26.18
	BiTGraph	31.89	14.05	20.65	35.06	14.62	22.44	36.98	15.04	23.38	39.06	16.18	25.01
	Proposed	**31.34**	**13.76**	**20.41**	**34.53**	**14.21**	**22.01**	**36.04**	**14.77**	**23.10**	**38.87**	**15.82**	**24.83**
China AQI	STID	31.97	45.19	18.54	34.28	48.48	20.32	36.36	55.39	22.79	40.28	62.41	25.97
	DSformer	28.22	44.81	17.73	30.35	48.09	19.06	33.22	52.05	20.63	35.72	57.31	23.17
	MegaCRN	28.41	33.35	15.32	29.52	35.86	16.51	33.09	48.96	19.66	35.74	53.28	22.61
	DCRNN+GPT4TS	28.48	32.18	15.14	30.83	35.24	16.82	31.99	37.22	17.82	34.28	50.64	21.78
	DFDGCN+TimesNet	26.33	29.73	14.62	29.30	32.85	15.76	30.51	38.68	17.85	33.19	51.28	21.06
	MTGNN+GRIN	27.13	33.29	14.89	30.02	37.82	16.37	31.97	39.29	17.94	34.15	50.23	20.02
	FourierGNN+GATGPT	27.17	32.82	14.65	30.77	38.37	16.13	31.86	39.90	18.01	34.17	50.94	20.18
	LGnet	27.76	38.92	16.02	31.03	44.95	18.39	34.09	49.63	20.36	35.54	57.18	23.24
	GC-VRNN	26.88	31.88	14.66	28.67	34.21	15.70	30.57	37.66	16.99	32.91	48.73	19.24
	TriD-MAE	26.18	29.09	14.51	28.96	32.94	15.74	29.84	35.76	16.79	32.68	45.76	18.04
	BiTGraph	25.79	28.94	13.85	27.45	31.36	14.52	29.01	33.58	15.62	31.85	38.17	17.06
	Proposed	**25.51**	**28.27**	**13.72**	**26.81**	**29.56**	**14.33**	**27.96**	**31.86**	**15.39**	**30.97**	**36.30**	**16.83**

4.3 Ablation Experiment

GinAR has three important components: interpolation attention, predefined graph, and adaptive graph learning. To demonstrate the importance of these components, ablation experiments are conducted from the following three perspectives: (1) **w/o ia**: We

remove the interpolation attention. (2) **w/o pg**: The predefined graph is deleted. It means that GinAR only uses the adaptive graph to construct spatial correlations. (3) **w/o ag**: The adaptive graph is removed. It means that spatial correlations are determined mainly through prior knowledge. Figure 4 shows the results of the ablation

(a) Results on PEMS-BAY dataset   (b) Results On China AQI dataset
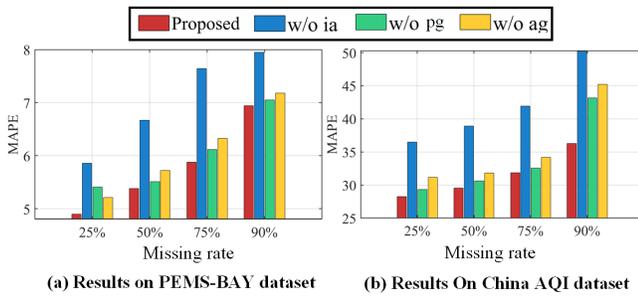
Figure 4: The results of the ablation experiment.

experiment. Based on the experimental results, the following conclusions can be drawn: (1) When the missing rate is low, deleting the predefined graph has a great impact on the forecasting result. However, when the missing rate is large, deleting the predefined graph has little impact on the result. (2) When the missing rate is large, deleting the adaptive graph can significantly reduce the forecasting accuracy. The main reason is that when there are more missing variables, the adaptive graph can better analyze the spatial correlation according to the characteristics of the data. Therefore, the adaptive graph plays an important role in this task. (3) 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 variables, which provides an important support for correcting spatial-temporal dependencies and avoiding error accumulation. To further analysis the effect of the IA, we compare IA with imputation methods in next section.

4.4 Performance Evaluation of IA

As one of the most important components proposed in this paper, it is important to further evaluate the effect of interpolation attention. In addition, it is important to further evaluate the effectiveness of the end-to-end framework. Therefore, this section compares the performance improvement effects of IA, GRIN, GATGPT, GPT4TS and TimesNet on STID. Specifically, TimesNet, GATGPT, GPT4TS and GRIN adopt the two-stage modeling framework (imputation and forecasting) to optimize the performance of STID. IA uses the end-to-end modeling framework to optimize the effects of STID. Table 3 shows the performance comparison results (MAE values) of these models (The best results are shown in **bold**). Based on the experimental results, the following conclusions can be drawn: (1) Compared with other methods, TimesNet has minimal performance improvements to STID. The main reason is that TimesNet uses temporal information to recover missing variables, without fully analyzing correspondences between missing variables and normal variables. (2) The proposed IA method and other imputation methods can effectively improve the forecasting effect of STID. (3) IA and GRIN can recover the performance of STID and obtain better forecasting results than other imputation methods. The main reason is that IA and GRIN adopt the graph-based framework to effectively reconstruct the spatial correlation between missing variables and normal variables, and then recover the missing variable data based on the normal variable. (4) Compared with other two-stage models, the end-to-end framework based on IA and STID can

Table 3: MAE values of interpolation attention and other imputation methods.

Datasets	Methods	Missing rate			
		25%	50%	75%	90%
METR-LA	STID+GPT4TS	4.05	4.32	4.76	5.02
	STID+TimesNet	4.54	4.64	5.13	5.44
	STID+GATGPT	3.96	4.19	4.39	4.57
	STID+GRIN	3.83	3.98	4.21	4.35
	STID+IA	**3.71**	**3.90**	**4.19**	**4.31**
PEMS08	STID+GPT4TS	23.84	24.91	26.79	28.51
	STID+TimesNet	24.55	25.24	27.70	29.69
	STID+GATGPT	22.49	24.18	26.43	27.76
	STID+GRIN	22.13	23.65	25.93	27.18
	STID+IA	**21.67**	**23.39**	**25.85**	**26.96**
China AQI	STID+GPT4TS	15.37	16.45	17.79	19.25
	STID+TimesNet	15.81	17.14	18.33	19.74
	STID+GATGPT	14.77	15.64	16.98	18.29
	STID+GRIN	14.25	15.13	16.48	17.92
	STID+IA	**13.75**	**14.87**	**16.25**	**17.83**

achieve good forecasting results. On the one hand, the two-stage models need to realize the feature reconstruction, and the problem of error accumulation results in the decline of forecasting accuracy. On the other hand, IA realizes adaptive induction by generating correspondences between normal variables and missing variables. Therefore, the end-to-end framework combining IA and STID can achieve better results.

4.5 Hyperparameter Experiment

The setting of the superparameter can affect the forecasting effect of the GinAR. In this section, we evaluate the influence of three main hyperparameters on the experimental results, including embedding size, variable embedding size, and number of layers. Figure 5 shows the influence of different hyperparameters on the experimental results (PEMS04 dataset). Based on the experimental results, 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 embedding size can be increased appropriately when the missing rate is small. And the embedding size cannot be too large when the missing rate is large. The main reason is that when the missing rate is large, the large embedding size can lead to overfitting, thus affecting the forecasting accuracy. (3) The number of layers has the greatest 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. Therefore, the best forecasting results is achieved when the number of layers is set to 2 or 3.

4.6 Visualization

To prove that GinAR can effectively predict the future values of all variables, this section visualizes the forecasting results from the spatial dimension. Figure 6 gives the visualization of the input features and forecasting results of GinAR on different missing rates
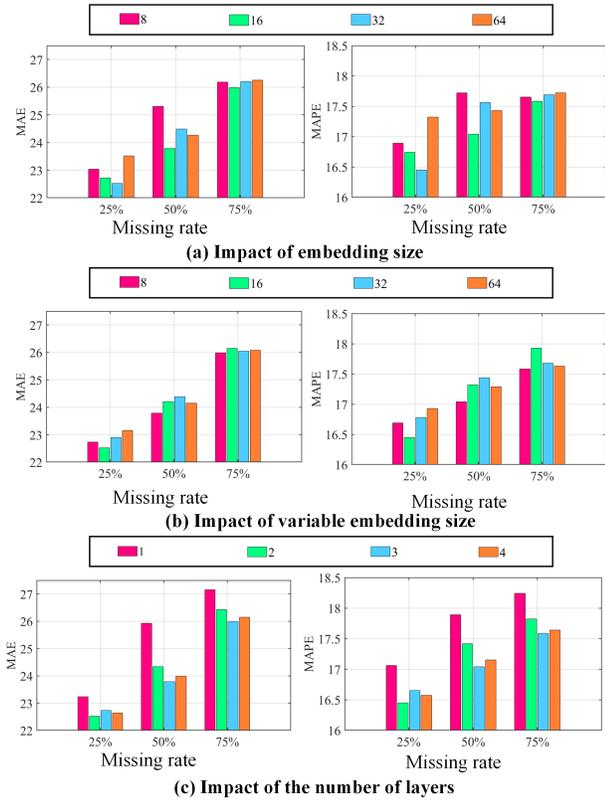
Figure 5: Hyperparameter experiment results (PEMS04).

(a) Real spatial distribution of China AQI

(b) The input and forecasting results of GinAR (Missing rate is 25%)

(c) The input and forecasting results of GinAR (Missing rate is 50%)

(d) The input and forecasting results of GinAR (Missing rate is 75%)

(e) The input and forecasting results of GinAR (Missing rate is 90%)

Figure 6: Visualization of the input features and forecasting results of GinAR on different missing rates (China AQI dataset). As the missing rate increases, the input features become more and more sparse. However, the forecasting performance of GinAR does not deteriorate significantly.

(China AQI dataset). Based on the visualization results, we can get the following conclusions: (1) As shown in Figure 6 (a), Figure 6 (b) and Figure 6 (c), GinAR can accurately predict the AQI value of all variables when the missing rate is not particularly large. (2) As shown in Figure 6 (a), Figure 6 (d) and Figure 6 (e), even though the number of normal variables is very sparse, GinAR can still accurately predict the spatial distribution of the AQI data. (3) GinAR can make full use of normal variables to realize accurate spatial-temporal forecasting for all variables. The visualization results can further demonstrate the practical value of GinAR.

5 CONCLUSION AND FUTURE WORK

In this paper, we try to address a new challenging task: MTSF with variable missing. In this task, to solve the problems of producing incorrect spatial-temporal dependencies and error accumulation in existing models, we carefully design two key components (Interpolation Attention and Adaptive Graph Convolution) and use them to replace all fully connected layers in the simple recursive unit. In this way, we propose the Graph Interpolation Attention Recursive Network based on the end-to-end framework, which can simultaneously recover all missing variables, correct spatial-temporal dependencies, and predict the future values of all variables. Experimental results on five real-world datasets demonstrate the practical value of our model, and even when only 10% of variables are normal, it can predict the future values of all the variables. In the future, we

will optimize the efficiency of GinAR and work on datasets with larger spatial dimensions and more complex spatial correlations.

ACKNOWLEDGMENTS

REFERENCES

[1] Dimitris Bertsimas, Agni Orfanoudaki, and Colin Pawlowski. 2021. Imputation of clinical covariates in time series. *Machine Learning* 110 (2021), 185–248.

[2] Ane Blázquez-García, Kristoffer Wickstrøm, Shujian Yu, Karl Øyvind Mikalsen, Ahcene Boubekki, Angel Conde, Usue Mori, Robert Jenssen, and Jose A Lozano. 2023. Selective imputation for multivariate time series datasets with missing values. *IEEE Transactions on Knowledge and Data Engineering* (2023).

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

[4] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. 2018. Brits: Bidirectional recurrent imputation for time series. *Advances in neural information processing systems* 31 (2018).

[5] Vitor Cerqueira, Nuno Moniz, and Carlos Soares. 2021. Vest: Automatic feature engineering for forecasting. *Machine Learning* (2021), 1–23.

[6] Jatin Chauhan, Aravindan Raghuveer, Rishi Saket, Jay Nandy, and Balaraman Ravindran. 2022. Multi-Variate Time Series Forecasting on Variable Subsets. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 76–86.

[7] Ling Chen, Donghui Chen, Zongjiang Shang, Binqing Wu, Cen Zheng, Bo Wen, and Wei Zhang. 2023. Multi-scale adaptive graph neural network for multivariate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[8] Xinyu Chen, Mengying Lei, Nicolas Saunier, and Lijun Sun. 2021. Low-rank autoregressive tensor completion for spatiotemporal traffic data imputation. *IEEE Transactions on Intelligent Transportation Systems* 23, 8 (2021), 12301–12310.

[9] Xiaodan Chen, Xiucheng Li, Bo Liu, and Zhijun Li. 2023. Biased Temporal Convolution Graph Network for Time Series Forecasting with Missing Values.. In *The Twelfth International Conference on Learning Representations*.

[10] Yuzhou Chen, Sotiris Batsakis, and H Vincent Poor. 2023. Higher-Order Spatio-Temporal Neural Networks for Covid-19 Forecasting. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[11] Yong Chen and Xiqun Michael Chen. 2022. A novel reinforced dynamic graph convolutional network model with data imputation for network-wide traffic flow prediction. *Transportation Research Part C: Emerging Technologies* 143 (2022), 103820.

[12] Yakun Chen, Zihao Li, Chao Yang, Xianzhi Wang, Guodong Long, and Guandong Xu. 2023. 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, 64–73.

[13] Yuanyuan Chen, Yisheng Lv, and Fei-Yue Wang. 2019. Traffic flow imputation using parallel data and generative adversarial networks. *IEEE Transactions on Intelligent Transportation Systems* 21, 4 (2019), 1624–1630.

[14] Yakun Chen, Xianzhi Wang, and Guandong Xu. 2023. Gatgpt: A pre-trained large language model with graph attention network for spatiotemporal imputation. *arXiv preprint arXiv:2311.14332* (2023).

[15] Yu Chengqing, Yan Guangxi, Yu Chengming, Zhang Yu, and Mi Xiwei. 2023. A multi-factor driven spatiotemporal wind power prediction model based on ensemble deep graph attention reinforcement learning networks. *Energy* 263 (2023), 126034.

[16] Ranak Roy Chowdhury, Xiyuan Zhang, Jingbo Shang, Rajesh K Gupta, and Dezhi Hong. 2022. Tarnet: Task-aware reconstruction for time-series transformer. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 212–220.

[17] Andrea Cini, Ivan Marisca, and Cesare Alippi. 2022. Filling the G_ap_s: Multivariate Time Series Imputation by Graph Neural Networks. In *International Conference on Learning Representations*.

[18] Razvan-Gabriel Cirstea, Bin Yang, Chenjuan Guo, Tung Kieu, and Shirui Pan. 2022. Towards spatio-temporal aware traffic time series forecasting. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2900–2913.

[19] Jinliang Deng, Xiusi Chen, Zipei Fan, Renhe Jiang, Xuan Song, and Ivor W Tsang. 2021. The pulse of urban transport: Exploring the co-evolving pattern for spatio-temporal forecasting. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15, 6 (2021), 1–25.

[20] Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W Tsang. 2021. 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*. 269–278.

[21] Jinliang Deng, Xiusi Chen, Renhe Jiang, Xuan Song, and Ivor W Tsang. 2022. A multi-view multi-task learning framework for multi-variate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[22] Jinliang Deng, Xiusi Chen, Renhe Jiang, Du Yin, Yi Yang, Xuan Song, and Ivor W Tsang. 2024. Disentangling Structured Components: Towards Adaptive, Interpretable and Scalable Time Series Forecasting. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[23] Wenjie Du, David Côté, and Yan Liu. 2023. Saits: Self-attention-based imputation for time series. *Expert Systems with Applications* 219 (2023), 119619.

[24] Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. 2020. Gp-vae: Deep probabilistic time series imputation. In *International conference on artificial intelligence and statistics*. PMLR, 1651–1661.

[25] Jingxuan Geng, Chunhua Yang, Yonggang Li, Lijuan Lan, and Qiwu Luo. 2022. MPA-RNN: a novel attention-based recurrent neural networks for total nitrogen prediction. *IEEE Transactions on Industrial Informatics* 18, 10 (2022), 6516–6525.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[27] Jia Hu, Xianghong Lin, and Chu Wang. 2022. MGCN: Dynamic Spatio-Temporal Multi-Graph Convolutional Neural Network. In *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–9.

[28] Marisca Ivan, Cini Andrea, and Cesare Alippi. 2022. Learning to Reconstruct Missing Data from Spatiotemporal Graphs with Sparse Observations. In *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*. 1–17.

[29] Renhe Jiang, Zhaonan Wang, Jiawei Yong, Puneet Jeph, Quanjun Chen, Yasumasa Kobayashi, Xuan Song, Shintaro Fukushima, and Toyotaro Suzumura. 2023. Spatio-temporal meta-graph learning for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 8078–8086.

[30] Renhe Jiang, Du Yin, Zhaonan Wang, Yizhuo Wang, Jiewen Deng, Hangchen Liu, Zekun Cai, Jinliang Deng, Xuan Song, and Ryosuke Shibasaki. 2021. 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*. 4515–4525.

[31] Tung Kieu, Bin Yang, Chenjuan Guo, Razvan-Gabriel Cirstea, Yan Zhao, Yale Song, and Christian S Jensen. 2022. Anomaly detection in time series with robust variational quasi-recurrent autoencoders. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 1342–1354.

[32] Thomas N Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

[33] Jinlong Li, Pan Wu, Hengcong Guo, Ruonan Li, Guilin Li, and Lunhui Xu. 2023. Multivariate Transfer Passenger Flow Forecasting with Data Imputation by Joint Deep Learning and Matrix Factorization. *Applied Sciences* 13, 9 (2023), 5625.

[34] Yujie Li, Zezhi Shao, Yongjun Xu, Qiang Qiu, Zhaogang Cao, and Fei Wang. 2023. Dynamic Frequency Domain Graph Convolutional Network for Traffic Forecasting. *arXiv preprint arXiv:2312.11933* (2023).

[35] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations*.

[36] Ke Liang, Yue Liu, Sihang Zhou, Wenxuan Tu, Yi Wen, Xihong Yang, Xiangjun Dong, and Xinwang Liu. 2023. Knowledge Graph Contrastive Learning Based on Relation-Symmetrical Structure. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[37] Ke Liang, Lingyuan Meng, Meng Liu, Yue Liu, Wenxuan Tu, Siwei Wang, Sihang Zhou, Xinwang Liu, and Fuchun Sun. 2022. Reasoning over different types of knowledge graphs: Static, temporal and multi-modal. *arXiv preprint arXiv:2212.05767* (2022), 7576–7584.

[38] Ke Liang, Jim Tan, Dongrui Zeng, Yongzhe Huang, Xiaolei Huang, and Gang Tan. 2023. Abslearn: a gnn-based framework for aliasing and buffer-size information retrieval. *Pattern Analysis and Applications* (2023), 1–19.

[39] Hui Liu, Chengqing Yu, Haiping Wu, Zhu Duan, and Guangxi Yan. 2020. A new hybrid ensemble deep reinforcement learning model for wind speed short term forecasting. *Energy* 202 (2020), 117794.

[40] Linfeng Liu, Michael C Hughes, Soha Hassoun, and Liping Liu. 2021. Stochastic iterative graph matching. In *International Conference on Machine Learning*. PMLR, 6815–6825.

[41] Yutian Liu, Soora Rasouli, Melvin Wong, Tao Feng, and Tianjin Huang. 2024. RT-GCN: Gaussian-based spatiotemporal graph convolutional network for robust traffic prediction. *Information Fusion* 102 (2024), 102078.

[42] Yonghong Luo, Ying Zhang, Xiangrui Cai, and Xiaojie Yuan. 2019. 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, 3094–3100.

[43] Soumen Pachal and Avinash Achar. 2022. Sequence Prediction under Missing Data: An RNN Approach without Imputation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 1605–1614.

[44] Tangwen Qian, Yile Chen, Gao Cong, Yongjun Xu, and Fei Wang. 2023. AdapTraj: A Multi-Source Domain Generalization Framework for Multi-Agent Trajectory Prediction. *arXiv preprint arXiv:2312.14394* (2023).

[45] Xiaobin Ren, Kaiqi Zhao, Patricia J Riddle, Katerina Taskova, Qingyi Pan, and Lianyan Li. 2023. DAMR: Dynamic Adjacency Matrix Representation Learning for Multivariate Time Series Imputation. *Proceedings of the ACM on Management of Data* 1, 2 (2023), 1–25.

[46] Siyuan Shan, Yang Li, and Junier B Oliva. 2023. Nrtsi: Non-recurrent time series imputation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.

[47] Chao Shang, Jie Chen, and Jinbo Bi. 2021. Discrete Graph Structure Learning for Forecasting Multiple Time Series. In *International Conference on Learning Representations*.

[48] Pan Shang, Xinwei Liu, Chengqing Yu, Guangxi Yan, Qingqing Xiang, and Xiwei Mi. 2022. A new ensemble deep graph reinforcement learning network for spatio-temporal traffic volume forecasting in a freeway network. *Digital Signal Processing* 123 (2022), 103419.

[49] Zezhi Shao, Fei Wang, Yongjun Xu, Wei Wei, Chengqing Yu, Zhao Zhang, Di Yao, Guangyin Jin, Xin Cao, Gao Cong, et al. 2023. Exploring Progress in Multivariate Time Series Forecasting: Comprehensive Benchmarking and Heterogeneity Analysis. *arXiv preprint arXiv:2310.06119* (2023).

[50] Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. 2022. 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*. 4454–4458.

[51] Zezhi Shao, Zhao Zhang, Fei Wang, and Yongjun Xu. 2022. 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*. 1567–1577.

[52] Zezhi Shao, Zhao Zhang, Wei Wei, Fei Wang, Yongjun Xu, Xin Cao, and Christian S Jensen. 2022. Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. *Proceedings of the VLDB Endowment* 15, 11 (2022), 2733–2746.

[53] Mengshuai Su, Hui Liu, Chengqing Yu, and Zhu Duan. 2023. A novel AQI forecasting method based on fusing temporal correlation forecasting with spatial correlation forecasting. *Atmospheric Pollution Research* 14, 4 (2023), 101717.

[54] Tao Sun, Fei Wang, Zhao Zhang, Lin Wu, and Yongjun Xu. 2022. Human mobility identification by deep behavior relevant location representation. In *International Conference on Database Systems for Advanced Applications*. Springer, 439–454.

[55] Xiangguo Sun, Hong Cheng, Bo Liu, Jia Li, Hongyang Chen, Guandong Xu, and Hongzhi Yin. 2023. Self-supervised hypergraph representation learning for sociological analysis. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[56] Jing Tan, Hui Liu, Yanfei Li, Shi Yin, and Chengqing Yu. 2022. A new ensemble spatio-temporal PM2. 5 prediction method based on graph attention recursive networks and reinforcement learning. *Chaos, Solitons & Fractals* 162 (2022), 112405.

[57] Peiwang Tang, Qinghua Zhang, and Xianchao Zhang. 2023. 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*. 181–189.

[58] Xianfeng Tang, Huaxiu Yao, Yiwei Sun, Charu Aggarwal, Prasenjit Mitra, and Suhang Wang. 2020. 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. 5956–5963.

[59] Fei Wang, Di Yao, Yong Li, Tao Sun, and Zhao Zhang. 2023. AI-enhanced spatial-temporal data-mining technology: New chance for next-generation urban computing. *The Innovation* 4, 2 (2023).

[60] Pu Wang, Zhihong Feng, Yan Tang, and Yuzhi Zhang. 2019. A fingerprint database reconstruction method based on ordinary kriging algorithm for indoor localization. In *2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*. IEEE, 224–227.

[61] Peixiao Wang, Tong Zhang, Yueming Zheng, and Tao Hu. 2022. A multi-view bidirectional spatiotemporal graph network for urban traffic flow imputation. *International Journal of Geographical Information Science* 36, 6 (2022), 1231–1257.

[62] Qi Wang, Tingting Li, Yongjun Xu, Fei Wang, Boyu Diao, Lei Zheng, and Jincai Huang. 2023. How to prevent malicious use of intelligent unmanned swarms? *The Innovation* 4, 2 (2023).

[63] Zhiyuan Wang, Fan Zhou, Goce Trajcevski, Kunpeng Zhang, and Ting Zhong. 2023. Learning Dynamic Temporal Relations with Continuous Graph for Multivariate Time Series Forecasting (Student Abstract). In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 16358–16359.

[64] Yuanyuan Wei, Julian Jang-Jaccard, Wen Xu, Fariza Sabrina, Seyit Camtepe, and Mikael Boulic. 2023. LSTM-autoencoder-based anomaly detection for indoor air quality time-series data. *IEEE Sensors Journal* 23, 4 (2023), 3787–3800.

[65] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *The Eleventh International Conference on Learning Representations*.

[66] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems* 34 (2021), 22419–22430.

[67] Yuankai Wu, Dingyi Zhuang, Aurelie Labbe, and Lijun Sun. 2021. Inductive graph neural networks for spatiotemporal kriging. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4478–4485.

[68] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 753–763.

[69] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 1907–1913.

[70] Yi Xu, Armin Bazarjani, Hyung-gun Chi, Chiho Choi, and Yun Fu. 2023. Uncovering the Missing Pattern: Unified Framework Towards Trajectory Imputation and Prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9632–9643.

[71] Yongjun Xu, Xin Liu, Xin Cao, Changping Huang, Enke Liu, Sen Qian, Xingchen Liu, Yanjun Wu, Fengliang Dong, Cheng-Wei Qiu, et al. 2021. Artificial intelligence: A powerful paradigm for scientific research. *The Innovation* 2, 4 (2021), 100179.

[72] Yongjun Xu, Fei Wang, Zhulin An, Qi Wang, and Zhao Zhang. 2023. Artificial intelligence for science—bridging data to wisdom. *The Innovation* 4, 6 (2023).

[73] Yongchao Ye, Shiyao Zhang, and James JQ Yu. 2021. Spatial-temporal traffic data imputation via graph attention convolutional network. In *International Conference on Artificial Neural Networks*. Springer, 241–252.

[74] Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Pengyang Wang, Ning An, Longbing Cao, and Zhendong Niu. 2023. FourierGNN: Rethinking Multivariate Time Series Forecasting from a Pure Graph Perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*.

[75] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. 2008. Wireless sensor network survey. *Computer networks* 52, 12 (2008), 2292–2330.

[76] Du Yin, Renhe Jiang, Jiewen Deng, Yongkang Li, Yi Xie, Zhongyi Wang, Yifan Zhou, Xuan Song, and Jedi S Shang. 2023. MTMGNN: Multi-time multi-graph neural network for metro passenger flow prediction. *GeoInformatica* 27, 1 (2023), 77–105.

[77] Jinsung Yoon, James Jordon, and Mihaela Schaar. 2018. Gain: Missing data imputation using generative adversarial nets. In *International conference on machine learning*. PMLR, 5689–5698.

[78] Chengqing Yu, Fei Wang, Zezhi Shao, Tao Sun, Lin Wu, and Yongjun Xu. 2023. Dsformer: A double sampling transformer for multivariate time series long-term prediction. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 3062–3072.

[79] Chengqing Yu, Guangxi Yan, Chengming Yu, Xinwei Liu, and Xiwei Mi. 2024. MRIformer: A multi-resolution interactive transformer for wind speed multi-step prediction. *Information Sciences* 661 (2024), 120150.

[80] Kai Zhang, Chao Li, and Qinmin Yang. 2023. 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*. 3164–3173.

[81] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. 2019. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE transactions on intelligent transportation systems* 21, 9 (2019), 3848–3858.

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

[83] Fan Zhou, Chen Pan, Lintao Ma, Yu Liu, Shiyu Wang, James Zhang, Xinxin Zhu, Xuanwei Hu, Yunhua Hu, Yangfei Zheng, et al. 2023. SLOTH: Structured Learning and Task-Based Optimization for Time Series Forecasting on Hierarchies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 11417–11425.

[84] Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. 2023. One Fits All: Universal Time Series Analysis by Pretrained LM and Specially Designed Adaptors. *arXiv preprint arXiv:2311.14782* (2023).

A EXPERIMENTAL DETAILS

A.1 Datasets

Table 4 shows the statistics of these datasets. A brief overview of all 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.

Table 4: The statistics of the five datasets.

Datasets	Variates	Timesteps	Granularity
MET-LA	207	34272	5minutes
PEMS-BAY	325	52116	5minutes
PEMS04	307	16992	5minutes
PEMS08	170	17856	5minutes
China AQI	350	59710	1hour

- **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.

A.2 Baselines

All baselines are introduced as follows:

- **STID**: It uses spatial-temporal identity embedding to improve the ability of MLP to mine multivariate time series.
- **DSformer**: It uses double sampling block and temproal variable attention block to mine spatiotemporal correlation and improve prediction performance.
- **MegaCRN**: This method uses memory back to improve the ability of AGCRN to model spatial correlation.
- **DCRNN+GPT4TS** : It first uses GPT4TS to realize the imputation of missing variables, and then uses DCRNN to model the processed data.
- **DFDGCN + TimesNet**: It first uses TimesNet to realize the imputation of missing variables, and then uses DFDGCN to model the processed data.
- **MTGNN+GRIN**: It first uses GRIN to realize the imputation of missing variables, and then uses MTGNN to model the processed data.
- **FourierGNN+GATGPT** : It first uses GATGPT to realize the imputation of missing variables, and then uses FourierGNN to model the processed data.
- **LGnet**: It uses the memory component to effectively improve the performance of long and short term memory networks.
- **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.

B EFFICIENCY

In this section, we compare the efficiency of GinAR with that of several baselines (GC-VRNN, TriD-MAE, MTGNN + GRIN, DFDGCN + TimesNet and DCRNN + GPT4TS) on the PEMS08 dataset. To ensure the fairness of the experiment, we compare the mean training time of each epoch of each model. 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. Based on Figure 7, it can be found that the training time of GinAR is not large. Compared with several two-stage models, the GinAR does not require the imputation stage, which reduces the overall training time. Besides, although the training time of GinAR is greater than that of the one-stage models, it solves the problem of variable missing, which can improve its forecasting performance.
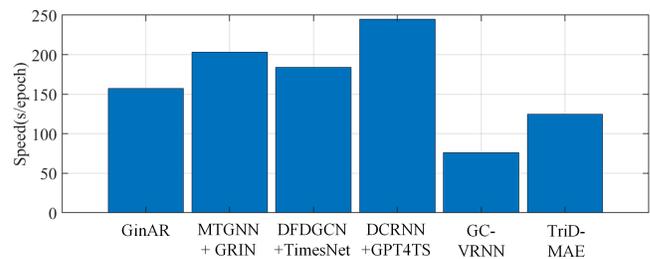
Figure 7: Training time for each epoch of different models.

C NOTATIONS

Some of the commonly used notations are presented in Table 5.

Table 5: Frequently used notation.

Notation	size	Definitions
H	Constant	The length of historical observation
L	Constant	The length of future forecasting results
B	Constant	Batch size
N	Constant	Number of variables
M	Constant	Number of missing variables
C	Constant	Embedding size
d	Constant	Variable embedding size
n	Constant	Number of GinAR layers
X	$N * H * C$	Input features
X_M	$N * H * C$	Input features with M missing variables
Y	$N * L$	Forecasting results
A_{pre}	$N * N$	Predefined graph
A_{adap}	$N * N$	Adaptive graph
E_a	$N * d$	Variable embedding of adaptive graph
F_{LN}	Functions	Layer normalization
FC	Functions	Fully connected layer
ReLU	Functions	Activation function ReLU
ELU	Functions	Activation function ELU
GeLU	Functions	Activation function GeLU
LeakyReLU	Functions	Activation function LeakyReLU
softmax	Functions	Activation function softmax