BIASED TEMPORAL CONVOLUTION GRAPH NETWORK FOR TIME SERIES FORECASTING WITH MISSING VALUES.

Anonymous authors
Paper under double-blind review

ABSTRACT

Multivariate time series forecasting plays an important role in various applications ranging from meteorology study, traffic management to economics planning. In the past decades, many efforts have been made toward accurate and reliable forecasting by exploring both temporal dynamics and spatial correlation. Especially, the development of Transformer-based methods has significantly enhanced long-term forecasting accuracy in very recent years. The existing forecasting methods often assume intact input data, however, in practice the time series data is often partially observed due to device malfunction or costly data acquisition, which can seriously impede the performance of the existing approaches. A naive employment of imputation methods unavoidably involves error accumulation and leads to suboptimal solutions. Motivated by this, we propose a Biased Temporal Convolution Graph Network that jointly captures the temporal dependencies and spatial structure. In particular, we inject bias into the two carefully developed modules—the Multi-Scale Instance PartialTCN and Biased GCN—to account for missing patterns. The experimental results show that our proposed model is able to achieve up to 9.93% improvements over the existing methods on five real-world benchmark datasets. The code is available at this repository: https://anonymous.4open.science/r/BiaTCGNet-1F80/.

1 INTRODUCTION

Multivariate time series forecasting finds its applications in a wide spectrum of domains such as meteorology, traffic, energy consumption, economics, etc. The real-world demand has spurred various forecasting approaches being developed in the literature. From the generative perspective, the multivariate time series data is produced by a collection of N (correlated) instances (e.g., sensors) during a period of time. Thus, an accurate characterization of the underlying dynamics gives rise to a number of works to faithfully model both the temporal dependencies (intra-instance correlation) and the spatial structure (inter-instance correlation), The statistical methods—ARIMA (Nelson, 1998), VAR (Zivot & Wang, 2006)—have made early attempts by building autoregressive models to capture the temporal dependencies. However, their linear dependency assumption often leads to poor performance in practice. Inspired by their successes in Natural Language Processing, there has been an increasing trend in designing forecasting models based on RNNs and Transformers to explore their nonlinear modeling and complex pattern extraction capacity (Salinas et al., 2020; Zhou et al., 2021; Liu et al., 2021; Wu et al., 2021; Zhou et al., 2022). Especially, benefiting from the wide-range receptive fields enabled by the attention mechanism, the Transformer-based methods have exhibited excellent prediction performance on long-term forecasting tasks.

Apart from the methods dedicated to temporal dependencies modeling, there is another line of work toward exploiting the spatial correlation of multivariate time series. Many proposals (Salinas et al., 2020; Liu et al., 2021) model the spatial dependencies implicitly and simply rely on a hidden representation to capture the correlation. BRITS (Cao et al., 2018) proposes to use a dense connection layer to learn the correlation between every instance pair, which results in a high model complexity. The advent of graph neural networks (GNNs) (Kipf & Welling, 2016; Defferrard et al., 2016) enables us to effectively explore the non-Euclidean structure data. Indeed, the DCRNN (Li et al., 2018) proposes to build graphs and conduct graph convolution operations to capture the spatial correlation explicitly in traffic flow forecasting, in which the graphs are induced by spatial proximity. To apply GNNs to the more general forecasting scenarios, in which the graph structures are not available, the proposals (Bai et al., 2020; Wu et al., 2020) propose to learn the graphs adaptively by learning each node an embedding and building the graphs using the node embeddings, which achieves great progress in enhancing the prediction accuracy.

Despite the promising results achieved, the existing methods pay relatively less attention to multivariate time series forecasting with missing values. In the real world, the collected time series data is often partially observed as caused by device malfunction, communication failure, or data acquisition difficulty. One straightforward solution is to employ the existing time series imputation methods (Cao et al., 2018; Marisca et al., 2022; Cini et al., 2022) and then build the forecasting models on the imputed data. However, this two-step process separates the forecasting from the imputation, and the accumulated errors may impede the model performance and lead to suboptimal solutions. GRU-D (Che et al., 2018) proposes a decayed-GRU mechanism to handle the missing values for time series classification, and a similar idea is also adopted by BRITS (Cao et al., 2018) and GRUI (Luo et al., 2018) for time series imputation. However, these methods are not dedicated to time series forecasting. The neural ODE (Chen et al., 2018) is capable of handling irregularly-sampled time series data and many variants including LatentODE, NeuralCDE, and CRUs (Rubanova et al., 2019; Kidger et al., 2020; Schirmer et al., 2022) have been proposed. However, these methods often entail an ODE-solver computation for each iteration and have to align time steps of different time series, and thus cannot utilize the sparsity of the observations.

Motivated by the above observations, in this paper, we propose a Biased Temporal Convolution Graph Network, dubbed BiaTCGNet, to jointly capture the temporal dependencies and spatial structure by explicitly exploring the missing values in the model architecture design. We develop two core modules—the Multi-Scale Instance PartialTCN and Biased GCN. The Multi-Scale Instance PartialTCN integrates biased instance-specific convolution and multiple broadcasting paths to handle intricate missing patterns. Furthermore, the Biased GCN module constructs a biased graph to capture accurate global spatial correlations and facilitates effective graph diffusion. Additionally, the missing patterns will be updated progressively along the temporal and spatial dimensions to maximize information propagation and minimize the impacts of missing values. To summarize, our contributions are as follows.

- We present BiaTCGNet to jointly capture the temporal dependencies and spatial structure for the time series forecasting with missing values, the proposed model explicitly considers the missing patterns in its model design.
- To effectively model the temporal dependencies for partially observed time series, we introduce Multi-Scale Instance PartialTCN by injecting bias to account for missing patterns; we present Biased GCN to handle missing values aware in the information propagation process.
- BiaTCGNet achieves up to 9.93% improvements over the existing forecasting methods under various missing values scenarios as verified on five real-world benchmark datasets.

2 Related Work

Time series forecasting with complete data Due to its practical importance, a lot of efforts have been devoted to developing accurate time series forecasting methods. The classic ARIMA Nelson (1998), VAR Holden (2010) build the autoregressive models based on linear dependency assumption. RNNs-based methods (Salinas et al., 2020) and (Zaremba et al., 2014) exploit the expressive power of recurrent neural networks to relax the linear assumption. Very recently, various Transformer-based methods have been proposed to exploit the wide-range receptive fields of attention mechanism for long-term forecasting. To reduce the quadratic complexity of vanilla attention, Informer (Zhou et al., 2021), Pyraformer (Liu et al., 2021), Autoformer (Wu et al., 2021), and FEDformer (Zhou et al., 2022) have been proposed successively. Non-stationary Transformer (Liu et al., 2022) aims to renovate the attention mechanism to account for the non-stationary property of time series data. PatchTST (Nie et al., 2023) explores the patch and channel-independence design. Apart from enhancing the temporal dynamics modeling capability, many proposals are dedicated to exploring spatial correlation. DCRNN (Li et al., 2018), AGCRN (Bai et al., 2020), MTGNN (Wu et al., 2020), and GTS (Shang et al., 2021) model the spatial structure with the graph neural networks. In addition, CoST (Woo et al., 2022) and Ts2Vec (Yue et al., 2022) approach the time series forecasting from the self-supervised learning perspective.

Modeling time series with missing values Caused by device malfunction, communication failure, or costly data acquisition, the real-world collected time series data is often incomplete and partially observed. To handle filling missing entries, many time series imputation methods—BRITS (Cao et al., 2018), GRIN (Cini et al., 2022), CSDI (Tashiro et al., 2021), SPIN (Marisca et al., 2022), GRIN (Cini et al., 2022), and TIDER (Liu et al., 2023)—have been presented in the machine learning community. To deal with the partially observed time series, one may attempt to build the forecasting models with the imputed results produced by the imputation methods. However, the imputation is disparate from the forecasting in this two-step process, and thus the accumulated errors may seriously degrade the forecasting performance. GRU-D (Che et al., 2018) presents a decayed-GRU to handle the missing values for time series classification without resorting to the imputation. Tang et al. (2020); Zuo et al. (2023) attempt to capture local dependencies based on global statistic characteristics for the missing value forecasting. The neural ODE-based models NeuralCDE, LatentODE, and CRUs (Chen et al., 2018; Rubanova et al., 2019; Schirmer et al., 2022) are capable of handling irregularly-sampled time series data. Nonetheless, they have to align time steps of different time series and cannot utilize the sparsity of the samples.

3 PRELIMINARIES

In this paper, we consider the multivariate time series $\mathbf{X} \in \mathbb{R}^{N \times T \times D}$ consisting of N univariate time series $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(N)}$ collecting over T time steps with D-dimension observation. Due to the malfunction of devices, communication failure, or costly data acquisition, there may exist missing values in $\mathbf{X}$, and we use a mask matrix $\mathbf{M} \in \mathbb{R}^{N \times T}$ to represent the missing patterns, which is defined as follows.

$$M_{nt} = \begin{cases} 1, & \text{if } X_{nt} \text{ is observed,} \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where X_{nt} denotes the value of n-th instance (or channel) at time step t, we alternatively use $x_t^{(n)}$ or X_{nt} to represent the same entry. Similarly, we use $\mathbf{m}^{(n)} \in \mathbb{R}^T$ to denote the n-th row of the mask matrix $\mathbf{M}$, and both M_{nt} and $m_t^{(n)}$ represent the n-row, t-column element of $\mathbf{M}$. In addition, the slice notation $\mathbf{x}_{t-H:t} \in \mathbb{R}^{H \times D}$ or $\mathbf{X}_{t-H:t} \in \mathbb{R}^{N \times H \times D}$ denotes the values in a time window of size H from time step $t-H$ to $t-1$, i.e., the time interval $[t-H, t)$. In the subsequent discussion, we will also refer to the mask as the missing pattern.

Multivariate time series forecasting with missing values Given the partial observed multivariate time series $\mathbf{X}$ and the corresponding mask matrix $\mathbf{M}$, the multivariate time series forecasting with missing values problem aims to build a forecasting model ϕ to predict the future F-step values $\mathbf{Y} = \mathbf{X}_{t:t+F}$ by taking as inputs the historical observation $\mathbf{X}_{t-H:t}$ and its mask $\mathbf{M}_{t-H:t}$, that is, $\hat{\mathbf{Y}} = \phi(\mathbf{X}_{t-H:t}, \mathbf{M}_{t-H:t})$. In the training phase, we only resort to the observed values to provide the learning signals. More formally, the loss function $\mathcal{L}$ of the model can be described as follows.

$$\mathcal{L}(\mathbf{Y}, \hat{\mathbf{Y}}, \mathbf{M}_{t:t+F}) = \frac{\sum_{n=1}^{N} \sum_{\tau=t}^{t+F-1} m_\tau^{(n)} |\hat{y}_\tau^{(n)} - y_\tau^{(n)}|}{\sum_{n=1}^{N} \sum_{\tau=t}^{t+F-1} m_\tau^{(n)}}, \tag{2}$$

which measures the mean absolute error between the predicted values and ground truths.

4 METHODOLOGY

The framework of our proposed BiaTCGNet (Biased Temporal Convolution Graph Network) is shown in Figure 1-(a). It comprises L identical blocks, dubbed Biased TCGBlock (Biased Temporal Convolution Graph Block), which is the basic building block of our proposed method. The Biased TCGBlock consists of two key modules: the multi-Scale Instance PartialTCN module and the Biased GCN module. The two modules are responsible for fusing the information along the temporal dimension and spatial dimension, respectively. In contrast to the existing time series forecasting methods, we explicitly consider the missing values in the model design and inject bias to account for the different missing patterns, and the model also progressively updates the missing patterns

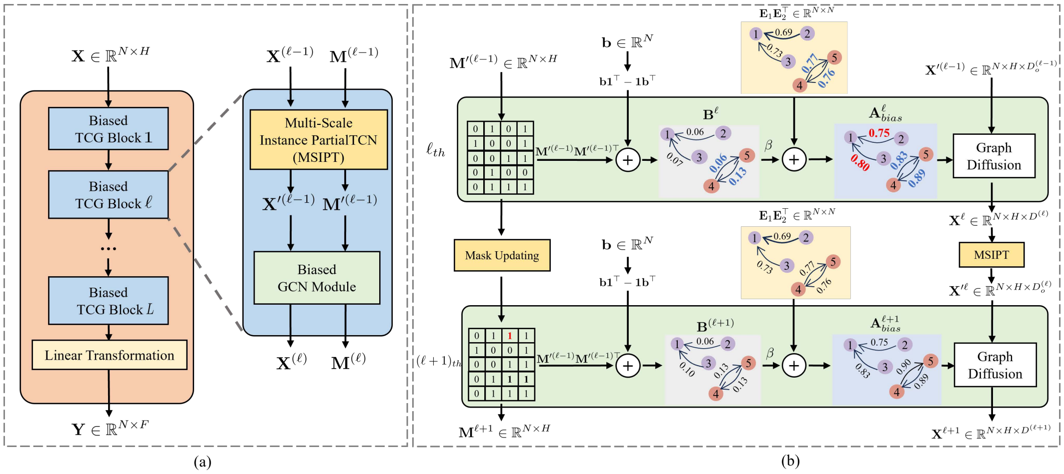

Figure 1: (a) The framework of our proposed BiaTCGNet, and (b) the illustration of Biased GCN module. It is noted that (b) disregards minor spatial connections and the temporal kernel size in (b) is specified as 1×1.

as the information diffusion proceeds. As the Figure 1-(a) shows, the ℓ-th block takes as inputs $\mathbf{X}^{(\ell-1)} \in \mathbb{R}^{N\times H\times D^{(\ell-1)}}$ and the missing patterns $\mathbf{M}^{(\ell-1)} \in \mathbb{R}^{N\times H}$, and it produces two transformed tensors $\mathbf{X}^{(\ell)} \in \mathbb{R}^{N\times H\times D^{(\ell)}}$ and $\mathbf{M}^{(\ell)} \in \mathbb{R}^{N\times H}$, where $D^{(\ell)}$ is the feature dimension of the ℓ-th block.

4.1 Multi-Scale Instance PartialTCN Module

In this paper, we opt for TCN as our backbone to capture the temporal dynamics for two main reasons: 1) it has been shown empirically that the TCN exhibits more favorable sequence modeling abilities in comparison to RNNs in a variety of tasks (Bai et al., 2018), 2) the convolution operation permits a simple modification to account for partial observations as evidenced in computer vision (Liu et al., 2018). We propose to apply the partial temporal convolution (PartialTCN) within each time series (instance). Different from vanilla partial convolution (PartialCNN), we focus on capturing temporal dependencies, enabling to explore temporal patterns invariant to instances and enhance statistical strengths, leading to **Instance PartialTCN**. On the other hand, we employ a parameter-efficient design that **Instance PartialTCN** is shareable across different instances. In addition, we further integrate multi-scale strategy to develop **Multi-Scale Instance ParitalTCN**, allowing our method can handle more intricate missing patterns. Next, we illustrate the module by focusing on a particular instance $\mathbf{x}^{(n)} \in \mathbb{R}^{H\times D}$ and its mask $\mathbf{m}^{(n)} \in \mathbb{R}^{H}$. To keep the notation uncluttered, we drop the upper script temporally.

Instance PartialTCN Given the kernel size k, the TCN applies the same linear transformation into different time windows under the time translation invariance assumption, i.e.,

$$\mathbf{x}' = \mathbf{x}_{t-k:t}\mathbf{W} + \mathbf{b}, \tag{3}$$

where $\mathbf{x}_{t-k:t} \in \mathbb{R}^{k\times D_i}$ is a time window sequence with D_i input features, $\mathbf{x}' \in \mathbb{R}^{D_o}$ is the output feature map at location $t-1$, and $\mathbf{W} \in \mathbb{R}^{k\times D_i\times D_o}$ and $\mathbf{b} \in \mathbb{R}^{D_o}$ are convolution parameters. Motivated by the success of partial convolutions in vision tasks, we introduce Instance PartialTCN to model the temporal dependencies of partially-observed time series to account for missing values as,

$$\mathbf{x}'_k = \begin{cases} \frac{k}{\mathrm{sum}(\mathbf{m}_{t-k:t})}\left(\mathbf{x}_{t-k:t} \odot \mathbf{m}_{t-k:t}\right)\mathbf{W_k} + \mathbf{b_k}, & \text{if } \mathrm{sum}(\mathbf{m}_{t-k:t}) > 0, \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{4}$$

where $\odot$ denotes the Hadamard product. The Instance PartialTCN only attends to the time steps with observations to compute the new feature maps and the factor $k/\,\mathrm{sum}(\mathbf{m}_{t-k:t})$ rescales the computation result to the same magnitude of convolutions on complete observations. In such a manner, the missing patterns are integrated into the temporal dynamics modeling. As the temporal convolution proceeds, the time steps with missing values will have chances to gather sufficient information from their surrounding neighbors. To account for this, the missing pattern $\mathbf{m}$ is updated as,

$$m_{t-1} = \begin{cases} 1, & \text{if } \mathrm{sum}(\mathbf{m}_{t-K:t}) > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

In other words, the time step $t-1$ is considered filled if we could collect values from the present time window $[t-K, t)$. The missing pattern $\mathbf{m}$ will be progressively filled as the convolution proceeds.

Multi-Scale Instance PartialTCN To capture the multi-scale temporal dependencies of the time series, we propose to integrate the multi-scale convolution with different kernel sizes into the Instance PartialTCN. Specifically, we adopt 1×3, 1×5, and 1×7 in this paper. Consequently, different kernels will yield multiple different updated missing patterns $\mathbf{m}_i^{(n)} \in \mathbb{R}^H$ for each instance n, $1 \leq i \leq n_{\text{ker}}$ and n_{ker} is the number of kernels. We propose to aggregate these missing patterns generated by different kernels to yield an eventual one by max pooling,

$$\mathbf{m}^{(n)} = \max(\mathbf{m}_i^{(n)}), \quad 1 \leq i \leq n_{\text{ker}}. \tag{6}$$

The aggregated $\mathbf{m}^{(n)}$ will then be used in the subsequent graph convolution module to diffuse information along spatial dimensions. By applying the Multi-Scale Instance PartialTCN to each instance $\mathbf{x}^{(n)}$ ($n = 1, 2, \ldots, N$), we transform the input feature map $\mathbf{X}^{(\ell-1)} \in \mathbb{R}^{N\times H\times D_i^{(\ell-1)}}$ and missing pattern $\mathbf{M}^{(\ell-1)} \in \mathbb{R}^{N\times H}$ into $\mathbf{X}'^{(\ell-1)} \in \mathbb{R}^{N\times H\times D_o^{(\ell-1)}}$ and updated missing pattern $\mathbf{M}'^{(\ell-1)} \in \mathbb{R}^{N\times H}$, respectively.

4.2 Biased GCN Module

The Multi-scale Instance PartialTCN focuses on capturing the temporal dynamics hidden in each instance without considering the inter-instance correlation. However, it is equally important to model both the spatial correlation and temporal dependencies for accurate multivariate time series forecasting. In this paper, we propose to use graph convolution networks to explore the spatial structure of the temporally fused feature map $\mathbf{X}'^{(\ell-1)}$ and updated $\mathbf{M}'^{(\ell-1)}$, produced by the Multi-scale Instance PartialTCN. The graph neural networks have been exploited to model the spatial correlation for time series forecasting in the literature either by using the predefined (Li et al., 2018) or adaptively-learned graph structures (Bai et al., 2020; Wu et al., 2020; Shang et al., 2021), in which each time series is treated as a graph node. In contrast to the existing approaches, we explicitly consider and incorporate a bias term (i.e., prior knowledge) into graph structure learning to account for the missing values, leading to **Biased GCN**. It is therefore able to deliver promising performance in the missing value scenarios. The Biased GCN module is shown in Figure 1-(b).

In this paper, we choose the adaptive graph structure learning approach since it is more flexible and applies to cases where the graph structures are unavailable. In particular, we learn the graph structure or adjacency matrix $\mathbf{A}$ by using two learnable embedding matrices $\mathbf{E}_1, \mathbf{E}_2 \in \mathbb{R}^{N\times D_{\text{node}}}$ as follows.

$$\mathbf{A} = \mathrm{ReLU}(\tanh(\mathbf{E}_1\mathbf{E}_2^\top)). \tag{7}$$

The i-th row of $\mathbf{E}_1$ (resp. $\mathbf{E}_2$), denoted by $\mathbf{e}_i^{(1)}$ (resp. $\mathbf{e}_i^{(2)}$), is the embedding of i-th time series and $\mathbf{e}_i^{(1)\top}\mathbf{e}_j^{(2)}$ quantifies the correlation strength from node i to node j. The reason we choose two embeddings instead of one $\mathbf{E}$ and computing $\mathbf{A} = \mathrm{ReLU}(\tanh(\mathbf{E}\mathbf{E}^\top))$ is that the spatial correlations are very likely to be asymmetric in practice. This learned adjacency matrix $\mathbf{A}$ will be used by the subsequent graph convolution operation to aggregate information and aid in the eventual forecasting task, and thus the embedding matrices $\mathbf{E}_1$ and $\mathbf{E}_2$ can be learned end-to-end.

However, Eq. 7 fails to account for missing patterns. Intuitively, the information propagation intensity should vary against the missing patterns and we choose the inner product to quantify it as

$$\mathbf{A}_{\textbf{bias}}^{\ell} = \mathrm{ReLU}(\tanh(\mathbf{E}_1\mathbf{E}_2^\top)) + \beta \mathrm{softmax}(\mathbf{M}_{t-H:t}\mathbf{M}_{t-H:t}^\top), \tag{8}$$

where the first term denotes global spatial correlation strength indicating the global message passing strengths among nodes, the second term is specific to a particular time window $[t-H, t)$ and can be considered as a time-window-specific bias that corrects the global message passing strength according to the current missing pattern in graph diffusion process, β denotes a learnable global parameter that controls the intensity of the correctness. By intuition, the information propagation should also be directed and more information should flow from nodes with fewer missing values to the ones with more missing values, but the second term is a symmetric matrix and cannot mirror this intuition. To correct this, we assign each node a learnable scalar bias b_i and use $b_i - b_j$ to adjust towards the asymmetries. Let $\mathbf{b} \in \mathbb{R}^N$ be the learnable bias term, we propose to learn the graph structure as,

$$\begin{aligned} \mathbf{B}^{\ell} &= \mathrm{softmax}(\mathbf{M}_{t-H:t}\mathbf{M}_{t-H:t}^{\top} + \mathbf{b}\mathbf{1}^{\top} - \mathbf{1}\mathbf{b}^{\top}) \\ \mathbf{A}_{\mathbf{bias}}^{\ell} &= \mathrm{ReLU}(\tanh(\mathbf{E}_1\mathbf{E}_2^{\top})) + \beta\mathbf{B}^{\ell}, \end{aligned} \tag{9}$$

where $\mathbf{1}$ is a length-N all-one vector and we omit the superscript of $\mathbf{M}'^{(\ell-1)}_{t-H:t}$ for brevity. As shown in Figure 1-(b), global message passing strengths between node 4 and node 5 are corrected by the time-window-specific bias.

To ensure the structure sparsity, we clamp the small entries of $\mathbf{A}_{\mathbf{bias}}^{\ell}$ to zeros by only preserving the neighbors of node i with the top-k correlation strengths and use the clamped $\mathbf{A}_{\mathbf{bias}}^{\ell}$ in the graph convolution operation to aggregate information (as will be shown shortly). Being analogous to the Instance PartialTCN, we propose to update the missing patterns of node i after aggregating the information from its spatial neighbors as follows,

$$\mathbf{m}^{(i)} = \max(\mathbf{m}^{(j)}), \quad j \in \{i\} \cup \mathcal{N}_i, \tag{10}$$

where $\mathcal{N}_i$ indicates the neighbors of node i in the graph. We illustrate the missing pattern updating of node 1 in Figure 1-(b).

Now considering the ℓ-th block of the model, it performs the graph convolution to diffuse information as follows.

$$\mathbf{X}^{(\ell)} = \left(\mathbf{I} + \mathbf{D}_{\mathrm{o}}^{-1}\mathbf{A}_{\mathbf{bias}}^{\ell} + \mathbf{D}_{\mathrm{i}}^{-1}(\mathbf{A}_{\mathbf{bias}}^{\ell})^{\top}\right)\mathbf{X}'^{(\ell-1)}\mathbf{\Theta}^{(\ell)} + \mathbf{b}^{(\ell)}, \tag{11}$$

where $\mathbf{X}'^{(\ell-1)}$ is the output of the Multi-Scale Instance PartialTCN in the ℓ-th block, $\mathbf{D}_{\mathrm{i}}$ and $\mathbf{D}_{\mathrm{o}}$ are the in-degree and out-degree matrix of $\mathbf{A}_{\mathbf{bias}}^{\ell}$, respectively, and $\mathbf{\Theta}^{(\ell)}$ and $\mathbf{b}^{(\ell)}$ are the graph convolution parameters of the ℓ-th block. $\mathbf{X}^{(\ell)}$ and $\mathbf{M}^{(\ell)}$ will then be fed to the next block as the inputs.

4.3 Hierarchical Biased TCGNet

By stacking L layers of Biased TCGBlock, we could enhance both the spatial and temporal receptive fields of the model. We initialize $\mathbf{X}^{(0)}$ and $\mathbf{M}^{(0)}$ with the original partial observation $\mathbf{X} \in \mathbb{R}^{N\times H\times D}$ and its corresponding missing pattern $\mathbf{M} \in \mathbb{R}^{N\times H}$, and the outputs of the L-th block are $\mathbf{X}^{(L)} \in \mathbb{R}^{N\times H\times D^{(L)}}$ and $\mathbf{M}^{(L)} \in \mathbb{R}^{N\times H}$. $\mathbf{X}^{(L)}$ fuses both the spatial and temporal features, which will be used to yield the future prediction by a linear transformation. Note that the mask of l-th layer $\mathbf{M}^{(l)}$ is hierarchically updated with information flows from bottom to up, aiming to ensure a correct mask. The model parameters are learned by optimizing the prediction loss in Eq. 2.

5 Experiments

We evaluate BiaTCGNet against the state-of-the-art forecasting methods under different missing rates on five real-world benchmark datasets. We first assess the forecasting performance of different methods in terms of three commonly used metrics, and then we verify the efficacy of our proposed modules by ablation study.

5.1 Experiment settings

Datasets We select five most commonly used time series forecasting datasets: Metr-LA, Electricity, PEMS, ETTh1, and BeijingAir, whose statistics are summarized in Table 1. The five datasets are collected from different domains and cover diverse magnitude ranges, sampling frequencies, and statistics. We randomly drop the data according to the missing rate r ranging from 0.1 to 0.8, including 0.1, 0.2, 0.4, 0.6, and 0.8.

Table 1: Dataset description.

	Metr-LA	**Electricity**	**PEMS**	**ETTh1**	**BeijingAir**
#Samples (T)	34272	26304	52116	17420	8759
#Instances (N)	207	321	325	7	36
Frequency	5 min	1 h	5 min	1 h	1 h
Mean	53.72	2538.79	62.62	4.58	72.01
Variance	410.53	2.26×10^8	92.05	42.68	79.07

Baseline methods We compare our proposed BiaTCGNet with the latest state-of-the-art forecasting methods as well as several classic methods. On the one hand, BRITS (Cao et al., 2018), SPIN (Marisca et al., 2022), GRIN (Cini et al., 2022), GCN-M (Zuo et al., 2023), CRUs (Schirmer et al., 2022) are representative forecasting methods designed specifically for time series with missing values. Meanwhile, we also include three Transformer-based methods, vanilla Transformer (Zerveas et al., 2021), STWA (Cirstea et al., 2022), and FEDformer (Zhou et al., 2022), as well as two Spatial-Temporal GNNs-based methods, AGCRN (Bai et al., 2020) and MTGNN (Wu et al., 2020). Since these five methods require complete input to perform prediction, we study their two variants, namely, filling the missing entries with zeros and the values imputed by TimesNet (Wu et al., 2023), the state-of-the-art time series imputation approach. We denote the corresponding variants as Model_0, and $\mathsf{Model}_\mathrm{t}$, respectively. It is noteworthy that in both of these variants, we introduce the missing mask as an additional feature to guide the forecasting task. The details of baseline methods are presented in Appendix A.

Implementation details The number of blocks L of BiaTCGNet is set to 3, the number of top-k nearest neighbors is set to 10 in all our experiments. The batch size is 32, the learning rate is 0.001. We split the datasets into training, validation, and test datasets with the ratio 0.6/0.2/0.2 chronologically. The future window size F is set to 24 for all methods, and the history window size H for our proposed method is 24. We select the best history window size from the set $\{24, 48, 96\}$ for the baseline methods and report their best results. All methods are trained on Nvidia V100 GPUs. Our method is implemented with PyTorch 2.0 and we use the source codes released by the authors for all baseline methods. Additionally, we adjust the hyperparameters of baseline methods to obtain the best performance on each dataset. In addition, we evaluate the performance of different methods in terms of Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE).

5.2 Overall Performance

Table 2 presents the forecasting performance on the two datasets (Metr-LA and Electricity) of different methods under the missing rates of 0.2, 0.4, 0.6, and 0.8, the results are averaged over five funs[1]. We move the results of the PEMS, ETTh1 and BeijingAir datasets and the results under the missing rates of 0.1 to Appendix B to save space. It can be seen from the table that our proposed BiaTCGNet is able to achieve the best results in most cases in terms of all three metrics. Its performance gains become more evident when the missing rate grows to 0.8, which benefits from the ability of the Multi-scale Instance PartialTCN module and the Biased GCN module in handling the missing patterns adaptively. It is worth noting that SPIN and GRIN, both of which are explicitly designed to address missing values, demonstrate a marked superiority. However, their practical applicability is constrained by the necessity of pre-defined graphs. Furthermore, The CRUs combines the Kalman filter and encoder-decoder framework; nevertheless, it still requires considerable time to reach convergence. Next, Table 2 reveals that SPIN, $\mathsf{MTGNN}_\mathrm{t}$, and STWA_t deliver the best results among all baseline methods. Our proposed BiaTCGNet always deliver reliable results. Notably, it achieves up to 9.93% improvement over the best baseline in terms of RMSE on the Electricity dataset.

5.3 Ablation Study

In this section, we conduct ablation studies to evaluate the effectiveness of our proposed modules, Multi-Scale Instance PartialTCN (MSIPT) and Biased GCN (BGCN) modules. The results are shown

[1]The model is trained with five different random seeds.

Table 2: The forecasting performance of different methods.

Method ($r = 0.2$)	Metr-LA MAE	Metr-LA RMSE	Metr-LA MAPE	Electricity MAE	Electricity RMSE	Electricity MAPE
BRITS	8.32 ± 0.02	13.18 ± 0.10	18.26 ± 0.71	1029.30 ± 1.10	10126.175 ± 30.57	47.73 ± 0.35
SPIN	6.46 ± 0.07	11.21 ± 0.05	12.98 ± 0.03	–	–	–
GRIN	6.80 ± 0.02	12.24 ± 0.12	16.18 ± 0.24	–	–	–
GCN-M	6.78 ± 0.03	11.12 ± 0.04	13.50 ± 0.02	–	–	–
CRUs	10.80 ± 0.02	12.49 ± 0.15	19.66 ± 0.54	464.66 ± 4.14	5276.49 ± 53.36	$\mathbf{25.64 \pm 0.53}$
AGCRN$_0$	14.88 ± 0.05	14.21 ± 0.04	28.94 ± 0.07	1307.62 ± 4.53	13217.78 ± 26.81	62.65 ± 0.22
Transformer$_0$	7.14 ± 0.06	13.08 ± 0.08	17.07 ± 0.09	296.03 ± 5.77	2432.09 ± 22.15	29.14 ± 0.27
FEDformer$_0$	7.09 ± 0.03	12.75 ± 0.14	16.73 ± 0.19	368.29 ± 3.71	2574.37 ± 25.89	31.29 ± 0.31
STWA$_0$	6.24 ± 0.07	10.99 ± 0.11	12.89 ± 0.13	272.60 ± 7.35	2263.55 ± 24.10	28.52 ± 0.26
MTGNN$_0$	6.34 ± 0.07	10.96 ± 0.10	12.51 ± 0.19	274.68 ± 5.56	2016.44 ± 13.77	28.54 ± 0.19
AGCRN$_t$	13.72 ± 0.06	13.11 ± 0.23	27.06 ± 0.18	1049.23 ± 12.06	11751.49 ± 20.67	57.76 ± 0.18
Transformer$_t$	6.90 ± 0.08	12.98 ± 0.13	16.49 ± 0.21	280.12 ± 6.78	2274.28 ± 25.18	28.74 ± 0.35
FEDformer$_t$	6.89 ± 0.06	11.75 ± 0.17	16.01 ± 0.09	313.59 ± 4.96	2666.93 ± 26.31	32.83 ± 0.23
STWA$_t$	6.20 ± 0.02	10.71 ± 0.11	12.26 ± 0.18	261.92 ± 4.65	2089.65 ± 19.35	27.37 ± 0.26
MTGNN$_t$	6.13 ± 0.02	10.76 ± 0.07	12.11 ± 0.19	269.25 ± 5.27	2175.24 ± 12.49	27.71 ± 0.78
BiaTCGNet	$\mathbf{6.04 \pm 0.02}$	$\mathbf{10.69 \pm 0.02}$	$\mathbf{11.69 \pm 0.11}$	$\mathbf{243.23 \pm 2.12}$	$\mathbf{1834.18 \pm 15.36}$	27.38 ± 0.46

Method ($r = 0.4$)	Metr-LA MAE	Metr-LA RMSE	Metr-LA MAPE	Electricity MAE	Electricity RMSE	Electricity MAPE
BRITS	8.38 ± 0.08	12.97 ± 0.11	18.39 ± 0.28	1029.73 ± 1.48	10136.39 ± 63.63	47.96 ± 0.56
SPIN	6.52 ± 0.07	11.94 ± 0.41	13.22 ± 1.00	–	–	–
GRIN	6.91 ± 0.09	12.60 ± 0.21	16.59 ± 0.18	–	–	–
GCN-M	7.09 ± 0.01	12.42 ± 0.03	17.06 ± 0.04	–	–	–
CRUs	10.94 ± 0.08	13.18 ± 0.44	20.13 ± 0.23	496.95 ± 6.03	5397.31 ± 52.52	27.94 ± 0.33
AGCRN$_0$	14.87 ± 0.04	14.30 ± 0.09	29.92 ± 0.06	1526.90 ± 13.77	14823.39 ± 21.68	68.73 ± 0.41
Transformer$_0$	7.25 ± 0.04	12.97 ± 0.06	17.72 ± 0.08	310.88 ± 4.67	2586.69 ± 22.73	31.79 ± 0.15
FEDformer$_0$	7.15 ± 0.02	12.89 ± 0.07	16.91 ± 0.12	406.17 ± 8.91	3606.49 ± 27.73	33.14 ± 0.33
STWA$_0$	6.37 ± 0.05	11.19 ± 0.06	13.13 ± 0.16	292.47 ± 4.64	2764.34 ± 20.06	29.07 ± 0.16
MTGNN$_0$	6.34 ± 0.05	11.10 ± 0.03	12.79 ± 0.08	305.46 ± 6.77	2576.44 ± 25.51	23.15 ± 0.37
AGCRN$_t$	12.73 ± 0.02	12.49 ± 0.14	24.13 ± 0.16	1283.27 ± 8.49	13743.42 ± 49.38	58.62 ± 0.36
Transformer$_t$	6.99 ± 0.06	12.49 ± 0.13	16.45 ± 0.08	300.43 ± 10.17	2529.26 ± 19.14	28.86 ± 0.20
FEDformer$_t$	7.10 ± 0.05	12.63 ± 0.13	16.62 ± 0.06	330.90 ± 7.76	2711.30 ± 22.31	29.24 ± 0.18
STWA$_t$	6.28 ± 0.03	10.93 ± 0.14	12.68 ± 0.07	289.59 ± 6.13	2355.34 ± 17.67	28.29 ± 0.31
MTGNN$_t$	6.26 ± 0.05	10.90 ± 0.10	12.49 ± 0.04	281.32 ± 6.82	2236.74 ± 16.81	28.46 ± 0.19
BiaTCGNet	$\mathbf{6.13 \pm 0.01}$	$\mathbf{10.76 \pm 0.02}$	$\mathbf{12.41 \pm 0.12}$	$\mathbf{270.14 \pm 3.77}$	$\mathbf{2091.88 \pm 30.49}$	$\mathbf{22.04 \pm 0.36}$

Method ($r = 0.6$)	Metr-LA MAE	Metr-LA RMSE	Metr-LA MAPE	Electricity MAE	Electricity RMSE	Electricity MAPE
BRITS	8.48 ± 0.02	12.94 ± 0.08	18.66 ± 0.22	1029.38 ± 1.84	10118.18 ± 33.04	48.25 ± 0.29
SPIN	6.61 ± 0.02	11.35 ± 0.17	13.31 ± 0.12	–	–	–
GRIN	7.04 ± 0.04	12.71 ± 0.14	17.04 ± 0.03	–	–	–
GCN-M	7.27 ± 0.02	11.55 ± 0.02	16.42 ± 0.03	–	–	–
CRUs	11.02 ± 0.02	13.38 ± 0.24	20.40 ± 0.04	664.07 ± 9.88	8126.82 ± 59.42	31.44 ± 0.45
AGCRN$_0$	14.87 ± 0.04	14.30 ± 0.09	29.92 ± 0.06	1945.61 ± 6.38	13891.03 ± 17.38	75.20 ± 0.29
Transformer$_0$	7.46 ± 0.01	12.03 ± 0.05	17.09 ± 0.07	346.43 ± 5.59	2952.28 ± 25.54	28.96 ± 0.37
FEDformer$_0$	7.50 ± 0.04	12.32 ± 0.03	17.31 ± 0.07	535.72 ± 7.67	5329.18 ± 26.71	42.09 ± 0.46
STWA$_0$	6.82 ± 0.02	11.72 ± 0.10	13.66 ± 0.04	325.47 ± 6.62	2479.75 ± 21.17	30.06 ± 0.22
MTGNN$_0$	6.95 ± 0.03	12.09 ± 0.02	13.87 ± 0.09	329.18 ± 4.61	2490.45 ± 23.38	28.20 ± 0.27
AGCRN$_t$	12.73 ± 0.02	12.49 ± 0.14	24.13 ± 0.16	1374.64 ± 7.11	12069.56 ± 19.73	61.92 ± 0.28
Transformer$_t$	7.22 ± 0.08	13.61 ± 0.17	16.75 ± 0.06	327.17 ± 8.68	2506.62 ± 23.17	29.27 ± 0.40
FEDformer$_t$	7.26 ± 0.04	13.08 ± 0.07	17.16 ± 0.03	341.66 ± 5.25	2682.73 ± 24.97	29.87 ± 0.09
STWA$_t$	6.55 ± 0.02	11.28 ± 0.07	13.57 ± 0.03	312.25 ± 5.36	2407.39 ± 23.05	29.05 ± 0.11
MTGNN$_t$	6.63 ± 0.02	11.10 ± 0.04	13.48 ± 0.05	309.59 ± 4.73	2399.51 ± 20.09	$\mathbf{25.37 \pm 0.13}$
BiaTCGNet	$\mathbf{6.32 \pm 0.01}$	$\mathbf{10.93 \pm 0.03}$	$\mathbf{12.67 \pm 0.11}$	$\mathbf{295.23 \pm 2.75}$	$\mathbf{2239.06 \pm 26.39}$	27.38 ± 0.49

Method ($r = 0.8$)	Metr-LA MAE	Metr-LA RMSE	Metr-LA MAPE	Electricity MAE	Electricity RMSE	Electricity MAPE
BRITS	8.56 ± 0.09	13.03 ± 0.18	18.92 ± 0.09	1027.28 ± 0.50	10150.54 ± 31.05	48.04 ± 0.02
SPIN	6.68 ± 0.31	11.42 ± 0.35	14.41 ± 1.20	–	–	–
GRIN	8.00 ± 0.02	12.68 ± 0.09	18.35 ± 0.05	–	–	–
GCN-M	7.75 ± 0.03	11.65 ± 0.04	17.94 ± 0.02	–	–	–
CRUs	11.35 ± 0.12	14.06 ± 0.70	22.08 ± 0.22	623.63 ± 13.07	7033.29 ± 17.85	33.29 ± 0.74
AGCRN$_0$	14.86 ± 0.01	14.27 ± 0.02	29.92 ± 0.08	2351.41 ± 26.79	16824.28 ± 29.33	207.77 ± 0.56
Transformer$_0$	8.06 ± 0.02	12.82 ± 0.05	18.37 ± 0.11	398.99 ± 6.62	3612.37 ± 24.19	30.07 ± 0.18
FEDformer$_0$	7.83 ± 0.05	12.97 ± 0.14	17.93 ± 0.06	676.93 ± 5.62	7859.76 ± 31.13	64.79 ± 0.35
STWA$_0$	7.57 ± 0.06	12.15 ± 0.07	17.31 ± 0.12	376.26 ± 5.36	3512.37 ± 22.09	31.15 ± 0.08
MTGNN$_0$	7.45 ± 0.03	12.21 ± 0.08	17.22 ± 0.09	383.89 ± 6.72	3539.74 ± 15.22	30.29 ± 0.11
AGCRN$_t$	14.88 ± 0.01	14.20 ± 0.05	29.92 ± 0.10	1841.76 ± 6.87	17376.51 ± 44.79	70.38 ± 0.56
Transformer$_t$	7.32 ± 0.04	12.96 ± 0.08	16.87 ± 0.05	391.83 ± 4.17	3451.33 ± 5.62	32.26 ± 0.17
FEDformer$_t$	7.33 ± 0.06	13.17 ± 0.06	16.71 ± 0.04	380.06 ± 3.39	3335.18 ± 20.10	31.56 ± 0.13
STWA$_t$	6.90 ± 0.03	11.30 ± 0.05	13.69 ± 0.07	362.25 ± 3.21	3156.68 ± 24.41	29.22 ± 0.15
MTGNN$_t$	6.79 ± 0.04	$\mathbf{11.05 \pm 0.07}$	13.54 ± 0.10	355.68 ± 5.11	3023.30 ± 11.46	28.78 ± 0.31
BiaTCGNet	$\mathbf{6.63 \pm 0.01}$	11.20 ± 0.00	$\mathbf{13.44 \pm 0.02}$	$\mathbf{347.35 \pm 1.76}$	$\mathbf{2839.79 \pm 25.49}$	$\mathbf{27.97 \pm 0.27}$

Table 3: The results of ablation studies on Metr, Electricity, and PEMS datasets under the missing rates of 0.2, 0.4, and 0.6.

Missing Rate	Model	Metr MAE	Metr RMSE	Electricity MAE	Electricity RMSE	PEMS MAE	PEMS RMSE
0.2	TCGNet	6.25 ± 0.01	11.01 ± 0.02	255.46 ± 2.86	2026.43 ± 35.79	1.94 ± 0.00	4.35 ± 0.02
	W.o.MSIPT	6.34 ± 0.04	11.25 ± 0.05	279.73 ± 4.91	2159.16 ± 28.74	1.97 ± 0.02	4.40 ± 0.04
	W.o.BGCN	6.26 ± 0.03	11.33 ± 0.03	263.69 ± 3.32	2029.16 ± 25.56	1.94 ± 0.00	4.37 ± 0.03
	W.o.Eq. 9	6.14 ± 0.02	10.72 ± 0.02	250.75 ± 5.01	2020.13 ± 36.75	1.94 ± 0.02	4.33 ± 0.03
	W.o.Eq. 4	6.12 ± 0.01	10.91 ± 0.03	246.18 ± 2.89	2003.22 ± 35.09	1.93 ± 0.01	4.30 ± 0.02
	BiaTCGNet	$\mathbf{6.04 \pm 0.02}$	$\mathbf{10.69 \pm 0.02}$	$\mathbf{243.23 \pm 2.12}$	$\mathbf{1834.18 \pm 15.36}$	$\mathbf{1.90 \pm 0.01}$	$\mathbf{4.28 \pm 0.01}$
0.4	TCGNet	6.41 ± 0.03	11.14 ± 0.04	284.39 ± 5.26	2323.03 ± 40.15	1.99 ± 0.01	4.47 ± 0.03
	W.o.MSIPT	6.48 ± 0.02	11.20 ± 0.03	299.34 ± 4.17	2361.79 ± 37.82	2.02 ± 0.02	4.50 ± 0.03
	W.o.BGCN	6.40 ± 0.02	11.20 ± 0.02	291.81 ± 3.87	2337.69 ± 31.98	1.98 ± 0.01	4.45 ± 0.02
	W.o.Eq. 9	6.18 ± 0.01	10.81 ± 0.04	282.12 ± 2.88	2236.82 ± 30.26	1.98 ± 0.01	$\mathbf{4.32 \pm 0.02}$
	W.o.Eq. 4	6.25 ± 0.00	10.87 ± 0.02	280.30 ± 2.73	2277.50 ± 28.49	1.97 ± 0.01	4.34 ± 0.01
	BiaTCGNet	$\mathbf{6.13 \pm 0.01}$	$\mathbf{10.76 \pm 0.02}$	$\mathbf{270.14 \pm 3.77}$	$\mathbf{2091.88 \pm 30.49}$	$\mathbf{1.96 \pm 0.00}$	4.34 ± 0.02
0.6	TCGNet	6.48 ± 0.02	11.10 ± 0.05	313.60 ± 3.29	2372.36 ± 36.19	2.04 ± 0.02	4.55 ± 0.02
	W.o.MSIPT	6.65 ± 0.03	11.50 ± 0.04	332.39 ± 3.82	2469.15 ± 33.63	2.09 ± 0.02	4.62 ± 0.04
	W.o.BGCN	6.65 ± 0.02	11.94 ± 0.01	322.68 ± 2.74	2487.22 ± 25.39	2.03 ± 0.00	4.49 ± 0.01
	W.o.Eq. 9	6.35 ± 0.03	11.06 ± 0.02	308.59 ± 3.97	2366.39 ± 32.16	2.03 ± 0.01	4.52 ± 0.02
	W.o.Eq. 4	6.38 ± 0.02	$\mathbf{10.84 \pm 0.02}$	301.25 ± 2.05	2312.39 ± 22.46	2.03 ± 0.02	4.54 ± 0.00
	BiaTCGNet	$\mathbf{6.32 \pm 0.01}$	10.93 ± 0.03	$\mathbf{295.23 \pm 2.75}$	$\mathbf{2239.06 \pm 26.39}$	$\mathbf{1.99 \pm 0.01}$	$\mathbf{4.47 \pm 0.01}$

in Table 3. We divide the MSIPT module or BGCN module into two distinct procedures. The first (Eq. 6 or Eq. 10) relates to the mask updating process (MUP). The second (Eq. 4 or Eq. 9) is involved in the information aggregation process (IAP). Firstly, we executed ablation studies (W.o.MSIPT, W.o.BGCN, and TCGNet) to assess the joint significance of UID and MUP across temporal and spatial dimensions. The results reveal that when we execute MSIPT or BGCN the performance diminishes even worse than that of the vanilla TCGNet. We analyze that because the MUP builds a complete information-passing path between the spatial and temporal dimensions, any disruption in a portion of the path of the spatial (or temporal) dimension invalidates the path of the temporal (or spatial) dimension.

To further prove the effectiveness of the IAP and MUP, we conducted ablation studies by modifying IAP. Firstly, we replaced the temporal convolution in the MSIPT module with the standard convolution operation (e.g., Eq. 3). Secondly, we altered the generation of adjacent matrix A using Eq. 7. The results are shown in the fourth and fifth rows of the table. It is crucial to emphasize that the MUP in the MSIPT and BGCN modules are still exited. As we can see, the replacement of either spatial or temporal operations within IAP leads to a notable drop in performance compared to BiaTCGNet, indicating the effectiveness of the IAP modules dedicated to handling missing values. The block missing scenarios are evaluated in Appendix C. The results of parameter sensitivity including window size H, the number of blocks L, and the number of nearest neighbors k are shown in Appendix D. We analyze the role of β in Appendix E. We visualize the prediction curves in Appendix F. The model complexity analysis is given in Appendix G.

6 CONCLUSIONS

In this paper, we present BiaTCGNet for the time series forecasting with missing values. BiaTCGNet jointly captures the temporal dynamics and spatial structure by explicitly taking the missing values into consideration. We inject bias into the two carefully designed modules, the Multi-Scale Instance PartialTCN and Biased GCN, to account for the missing patterns. The experimental results on five real-world benchmark datasets verify its superiority under various missing value scenarios. The ablation studies also show that its excellent performance stems from the two carefully designed Multi-Scale Instance PartialTCN and Biased GCN components. In the future, we would like to explore the Transformer architecture as the backbone of our temporal module to further enhance its long-term forecasting performance for partially observed time series data.

REFERENCES

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. Brits: Bidirectional recurrent imputation for time series. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 2018.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Andrea Cini, Ivan Marisca, and Cesare Alippi. Filling the g_ap_s: Multivariate time series imputation by graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2022.

Razvan-Gabriel Cirstea, Bin Yang, Chenjuan Guo, Tung Kieu, and Shirui Pan. Towards spatio-temporal aware traffic time series forecasting. In *International Conference on Data Engineering (ICDE)*, 2022.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Ken Holden. Vector auto regression modeling and forecasting. *International Journal of Forecasting*, 2010.

Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

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

Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *European Conference on Computer Vision (ECCV)*, 2018.

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations (ICLR)*, 2021.

Shuai Liu, Xiucheng Li, Gao Cong, Yile Chen, and Yue Jiang. Multivariate time-series imputation with disentangled temporal representations. In *International Conference on Learning Representations (ICLR)*, 2023.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Yonghong Luo, Xiangrui Cai, Ying Zhang, Jun Xu, et al. Multivariate time series imputation with generative adversarial networks. *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Ivan Marisca, Andrea Cini, and Cesare Alippi. Learning to reconstruct missing data from spatiotemporal graphs with sparse observations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Brian K Nelson. Time series analysis using autoregressive integrated moving average (arima) models. *Academic Emergency Medicine*, 1998.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *International Conference on Learning Representations (ICLR)*, 2023.

Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent odes for irregularly-sampled time series. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 2020.

Mona Schirmer, Mazin Eltayeb, Stefan Lessmann, and Maja Rudolph. Modeling irregular time series with continuous recurrent units. In *International Conference on Machine Learning (ICML)*, 2022.

Chao Shang, Jie Chen, and Jinbo Bi. Discrete graph structure learning for forecasting multiple time series. In *International Conference on Learning Representations (ICLR)*, 2021.

Xianfeng Tang, Huaxiu Yao, Yiwei Sun, Charu Aggarwal, Prasenjit Mitra, and Suhang Wang. Joint modeling of local and global temporal dynamics for multivariate time series forecasting with missing values. In *International Joint Conference on Artificial Intelligence(AAAI)*, 2020.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. In *International Conference on Learning Representations (ICLR)*, 2022.

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

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet:temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations (ICLR)*, 2023.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2020.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2022.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.

George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *ACM SIGKDD Conference on Knowledge Discovery Data Mining (SIGKDD)*, 2021.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning (ICML)*, 2022.

Eric Zivot and Jiahui Wang. Vector autoregressive models for multivariate time series. In *Modeling Financial Time Series with S-PLUS®*, 2006.

Jingwei Zuo, Karine Zeitouni, Yehia Taher, and Sandra Garcia-Rodriguez. Graph convolutional networks for traffic forecasting with missing values. *Data Mining and Knowledge Discovery*, 2023.