Beyond Sequences: A Dynamic Hierarchical Heterogeneous Spatio-Temporal Graph for Irregular Multivariate Time Series Forecasting

Xiaowei Yan[1] , **Zhuo Li**[1] , **Junjie Zhang**[1] , **Bing Li**[1] , **Jun Yan**[2] and **Buzhou Tang**[1,3†]
[1]Harbin Institute of Technology, Shenzhen
[2]City University of Hong Kong
[3]Peng Cheng Laboratory, Shenzhen
{24s151172, 220110918}@stu.hit.edu.cn, {junjzhang7, lbing9411}@gmail.com, yan.jun@cityu.edu.hk, tangbuzhou@gmail.com

Abstract

Irregular Multivariate Time Series (IMTS) analysis is a challenging task as asynchronous irregular sampling disrupts intra-variable temporal consistency and cross-variable alignment. Most existing methods model multivariate correlations at either the variable or observation level in static ways. They suffer from correlation loss or cross-variable misalignment inevitably. In this paper, we propose a novel method DyH2-STGraph, Dynamic Hierarchical Heterogeneous Spatio-Temporal Graph, for IMTS forecasting. Within the graph, irregular observations are represented as nodes with spatio-temporal coordinates and observed values, connected with other neighbor observation nodes dynamically by spatio-temporal neighbor selection, while variables are treated as hyper nodes, connected with their constituent observation nodes as well as other variable nodes. The multivariate correlations are classified into the following three categories in specific manners: (1) Intra-variable coarse-to-fine correlations between variable nodes and their constituent observation nodes captured by hierarchical message propagation, (2) Inter-variable fine-grained correlations among neighboring observation nodes captured by spatio-temporal attention, and (3) Inter-variable coarse-grained correlations among variable nodes captured by attention. Extensive experiments on four benchmark datasets demonstrate that DyH2-STGraph significantly outperforms state-of-the-art methods while maintaining competitive efficiency.

1 Introduction

Irregular Multivariate Time Series (IMTS) forecasting is a critical task in various domains, including clinical medicine, financial analysis, and industrial monitoring [Yao *et al.*, 2018; Vio *et al.*, 2013; Shukla and Marlin, 2020; De Brouwer *et al.*, 2019]. Unlike regularly sampled time series, IMTS are characterized by asynchronous observation timestamps across variables, heterogeneous sampling frequencies, and pervasive missing values. Despite recent advancements, accurately capturing the latent dependencies within such irregularly-sampled data remains a significant challenge.

To address the inherent irregularity, conventional methods predominantly rely on alignment-based preprocessing, such as interpolation, resampling, or patching [De Gooijer and Hyndman, 2006; Zeng *et al.*, 2022]. However, these manual imputation or resampling strategies introduces non-trivial inductive biases, which can distort the underlying data distribution [Ansari *et al.*, 2023]. While patching avoids explicit point-wise alignment, it inevitably leads to boundary effects and data truncation. These issues obscure high-frequency dynamics and, in scenarios of extreme sparsity, result in the irretrievable loss of inherent irregular patterns [Nie *et al.*, 2023; Zhang *et al.*, 2023]. Standard sequence-modeling architectures, such as Recurrent Neural Networks (RNNs) and Transformers, exhibit limitations when applied to IMTS [Li and Marlin, 2020]. RNNs typically assume uniform time steps between transitions, failing to account for temporal decay or state evolution during variable intervals. Similarly, standard Transformers utilize discrete positional encodings that are unsuitable for representing the continuous nature of time [Vaswani *et al.*, 2017].

While graph-based representations have emerged as a promising alternative, they often encounter a structural trade-off between modeling precision and computational complexity [Yalavarthi *et al.*, 2024; Li *et al.*, 2025]. Modeling IMTS as a global fully-connected graph incurs prohibitive computational costs and risks diluting of salient information by global noise [Luo *et al.*, 2025]. Conversely, localized graph models, while more efficient, suffer from restricted receptive fields and a lack of global contextual awareness [Liu *et al.*, 2026a]. Consequently, neither approach effectively captures the multi-scale dependencies required for robust IMTS analysis. Although some advanced models adopt dimensional decoupling strategy to process temporal and variable dependencies separately, this approach restricts correlations to axis-parallel directions [Liu *et al.*, 2024]. Such a design ignores the intrinsic spatio-temporal coupling of local dynamics, forcing the model to rely on multiple stacked layers to indirectly reconstruct cross-variable interactions. This architectural constraint significantly reduces sensitivity to fine-grained, instantaneous dependencies.

[†]Corresponding author

To address these limitations, we propose the Dynamic Hierarchical Heterogeneous Spatio-Temporal Graph (DyH2-STGraph). Instead of constraining irregular observations to a rigid grid, DyH2-STGraph adopts a graph-centric perspective, where observations are represented as nodes with spatio-temporal coordinates. Meanwhile, to reconcile the trade-off between local precision and global context, our framework introduces a variable-level hypernode. These two types of nodes are connected through three categories of edges, enabling a three-fold correlation decomposition: (1) Hierarchical edges bridge observation nodes with their corresponding variable hypernodes, facilitating intra-variable coarse-to-fine correlation capture. (2) Spatio-temporal edges are dynamically established between neighboring observation nodes via a neighbor selection mechanism, facilitating inter-variable fine-grained correlation capture. (3) Horizontal edges connect all hypernodes, enabling inter-variable coarse-grained correlation capture. By leveraging these heterogeneous nodes and edges, DyH2-STGraph effectively captures the multi-grained, non-stationary correlations inherent in IMTS. Our major contributions are summarized as follows:

- We introduce a dynamic hierarchical heterogeneous spatio-temporal graph paradigm specifically tailored for IMTS. This approach naturally accommodates asynchronous observations across variables, effectively bypassing the need for traditional alignment or imputation.
- We propose a comprehensive mechanism to disentangle IMTS correlations into three levels: intra-variable hierarchical, inter-variable fine-grained, and inter-variable coarse-grained correlations. Facilitated by three distinct categories of edges of the graph, this decomposition enables more robust representation learning.
- We conducted extensive experiments on four public benchmark datasets. The results demonstrate that DyH2-STGraph consistently outperforms state-of-the-art baselines. In-depth analyses further validate the effectiveness of each proposed component in modeling multivariate correlations.

2 Related Work

2.1 Irregular Multivariate Time Series Modeling

IMTS modeling faces unique challenges from asynchronous sampling and heterogeneous temporal gaps. Existing approaches fall into four paradigms: (1) Continuous-time Dynamics Models: Neural ODEs and flows-based frameworks parameterize latent dynamics as continuous functions, naturally handling arbitrary temporal gaps [Chen *et al.*, 2018; Kidger *et al.*, 2020; Biloš *et al.*, 2021; Mercatali *et al.*, 2024; Schirmer *et al.*, 2022]. However, they incur substantial computational overhead and may struggle with high-dimensional dependencies. (2) Standard MTS Adaptation Models: These methods adapt conventional architectures through time-aware mechanisms: RNN-based methods use trainable decay or gating [Che *et al.*, 2018], attention-based approaches map observations to reference points [Shukla and Marlin, 2021], and Transformers employ discrete encodings or warping [Zhang *et al.*, 2023]. Despite effectiveness, their reliance on fixed-step transitions, and require alignment operations that may distort intrinsic temporal patterns. (3) Set-based Models: These methods treat observations as permutation-invariant sets, bypassing temporal alignment [Horn *et al.*, 2020]. While efficient, they sacrifice precise temporal ordering, limiting their capacity for high-frequency dynamics. (4) Graph-based Models: Graph provide a flexible paradigm for modeling complex relational structures in IMTS [Wu *et al.*, 2020; Han *et al.*, 2024]. They typically represent variables as nodes and their dependencies as edges, using sparse representations to handle IMTS samples without padding. While showing promise for IMTS modeling, they still suffer from architectures restricts. We discussed in detail in Section 2.2.

2.2 Using Graphs for IMTS

Graph are increasingly popular in IMTS analysis[Li *et al.*, 2025; Yi *et al.*, 2023; Wen *et al.*, 2023; Cini *et al.*, 2021]. Early GNN-based approaches required a predefined adjacency matrix to establish relationships between variables [Wu *et al.*, 2020]. However, these methods often rely on fixed topologies that cannot adapt to transient, non-stationary dependencies. To address this challenge, recent models have integrated graph learning components to dynamically capture dependencies between variables [Cao *et al.*, 2020; Chen *et al.*, 2021]. Existing graph-based models primarily focus on learning correlations at either the variable level [Zhang *et al.*, 2022] or the node level [Yalavarthi *et al.*, 2024]. Nevertheless, they often fail to fully exploit complex graph structures, leading to inevitable correlation loss. Furthermore, while some recent approaches utilize hypergraphs or bipartite graphs to capture correlations flexibly [Li *et al.*, 2025; Satorras *et al.*, 2022; Yalavarthi *et al.*, 2024], they typically process temporal and variable dependencies separately, which restricts correlations to axis-aligned directions and results in suboptimal performance. Some methods initialize data-driven contexts to capture the intrinsically coupled nature of local spatio-temporal dynamics [Liu *et al.*, 2026a]. However, their graph structures often remain static during the message-passing process, limiting their capacity to capture evolving correlations.

3 Problem Definition

We consider an IMTS dataset $\mathcal{D}$. With a total of T timestamps and M variables, each IMTS sample $S \in \mathcal{D}$ can be denoted as a collection of observation triplets: $S = \{(t_i, v_i, x_i) \mid i = 1, \ldots, N\}$, where $t_i \in \{1, \ldots, T\}$, $v_i \in \{1, \ldots, M\}$, $x_i \in \mathbb{R}$ represent the timestamp, variable indicator, observed value respectively, and N represent the total number of observations in the sample S. For the IMTS forecasting task, given a split timestamp t_S, an IMTS sample is divided into a lookback window $\mathcal{X} = \{(t_i, v_i, x_i) \in S \mid t_i \leq t_S\}$ and a forecast window $\mathcal{Y} = \{(t_i, v_i, x_i) \in S \mid t_i > t_S\}$. The forecast query $q_i \in \mathcal{Q}$ is derived by combining (t_i, v_i) of the i-th observation triplet within the forecast window. We aim to learn a forecasting model $\mathcal{F}(\cdot)$, such that given the lookback window $\mathcal{X}$ and forecast query $\mathcal{Q}$ as input, it accurately predicts the corresponding observed values $\mathcal{Z}$:

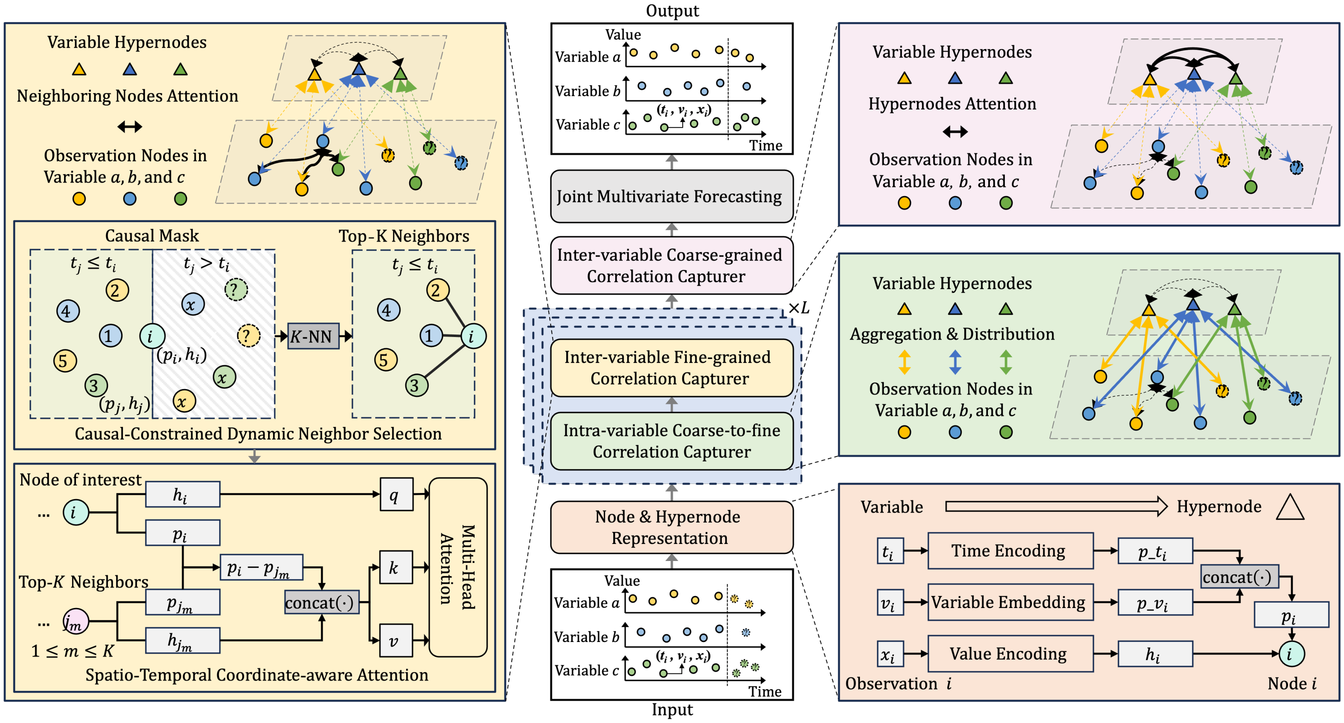

Figure 1: Overview of the DyH2-STGraph framework. Node & Hypernode Representation encodes the two types of nodes. Intra-variable Coarse-to-fine Correlation Capturer, Inter-variable Fine-grained Correlation Capturer and Inter-variable Coarse-grained Correlation Capturer encode multivariate correlations via the three categories of edges. Joint Multivariate Forecasting decodes multivariate for joint forecasting.

$$\mathcal{F}(\mathcal{X}, \mathcal{Q}) \rightarrow \mathcal{Z}. \tag{1}$$

4 Methodology

Figure 1 shows the DyH2-STGraph framework. All modules are introduced in the following subsections in detail.

4.1 Node & Hypernode Representation

For each IMTS sample, irregular observations are represented as nodes with spatio-temporal coordinates and observed values, while variables are treated as hyper nodes. The two types of nodes can be represented as follows.

Observation Node Representation. Each observation node i denoted by (t_i, v_i, x_i) is represented by:

(1) Temporal Embeddings, which is used to capture temporal distances in a continuous space by projecting the normalized timestamp t_i:

$$\mathbf{p}_{t_i} = \Phi_T(t_i) \in \mathbb{R}^{d_t}, \tag{2}$$

(2) Spatial Embeddings, which is used to distinguish nodes across different variables by mapping the variable index v_i:

$$\mathbf{p}_{v_i} = f_{\text{emb}}(v_i) \in \mathbb{R}^{d_v}, \tag{3}$$

(3) Value Embeddings, which is used to encode the node's feature. To distinguish between known observations and forecast targets, a binary indicator $m_i \in \{0, 1\}$ is concatenated with the numerical value x_i before projection, with the value of forecast target initialized to 0. The value embedding of node i is given by:

$$\mathbf{h}_i^{(0)} = \text{ReLU}(\text{FF}_{\text{obs}}(x_i \oplus m_i)) \odot \mathbb{I}_{\text{mask}}, \tag{4}$$

where FF_{obs} is a learnable projection, and $\oplus$ denotes the concatenation operation. $\mathbb{I}_{\text{mask}} \in \{0, 1\}$ indicates whether node i is a padding node.

Finally, the representation of node i is $(\mathbf{p}_i, \mathbf{h}_i^{(0)})$, with the spatio-temporal coordinate defined as:

$$\mathbf{p}_i = \mathbf{p}_{t_i} \oplus \mathbf{p}_{v_i} \in \mathbb{R}^{d_t + d_v}. \tag{5}$$

Hypernode Representation. We define a set of learnable global parameters $\mathbf{W}_s \in \mathbb{R}^{M \times d_{model}}$ to initialize the hypernodes corresponding to M variables. The initial state of the v-th variable hypernode is given by:

$$\mathbf{s}_v^{(0)} = \text{ReLU}([\mathbf{W}_s]_{v,:}), v = 1, \ldots, M. \tag{6}$$

4.2 Intra-variable Coarse-to-fine Correlation Capturer

We propose a bidirectional hierarchical message propagation to capture intra-variable coarse-to-fine correlations between variable hypernodes and their constituent observation nodes, that is, Bottom-up Coarse-grained Aggregation first, and then Top-down Fine-grained Refinement.

Bottom-up Coarse-grained Aggregation. In this phase, observation nodes associated with the same variable are aggregated to obtain a representative hypernode embedding.

The membership is strictly governed by the variable incidence matrix $\mathcal{I} \in \{0,1\}^{M\times N}$, where $\mathcal{I}_{v,i} = 1$ indicates that node i belongs to variable v (either as an observation or a forecast target). Notably, we employ a causal mask $\mathbf{M} \in \{0,1\}^N$ at the first layer to ensure that each variable hypernode aggregates only historical observation nodes initially, where $\mathbf{M}_i = 1$ denotes that node i is an observation, and $\mathbf{M}_i = 0$ denotes that node i is a forecast target. The representation of hypernode v is updated by:

$$\begin{aligned}\mathbf{s}_v^{(l+1)} = \mathrm{MHA}\Big(&\mathbf{Q} = \mathbf{s}_v^{(l)}, \mathbf{K} = \mathbf{h}^{(l)}, \mathbf{V} = \mathbf{h}^{(l)}, \\ &\mathrm{mask} = [\mathcal{I}]_{v,:} \odot \mathbf{M}^{(l=0)}\Big).\end{aligned} \tag{7}$$

This step transforms discrete, irregular nodes into coarse-grained hypernodes representing the overall semantic evolution of the variable.

Top-down Fine-grained Refinement. After aggregation, hypernode representations are redistributed to their constituent observation nodes. The representation of observation node i is refined by:

$$\begin{aligned}\mathbf{z}_i^{(l+1)} = \mathrm{MHA}\Big(&\mathbf{Q} = \mathbf{h}_i^{(l)}, \mathbf{K} = (\mathbf{s}_{v_i}^{(l+1)} \oplus \mathbf{h}_i^{(l)}), \\ &\mathbf{V} = (\mathbf{s}_{v_i}^{(l+1)} \oplus \mathbf{h}_i^{(l)})\Big),\end{aligned} \tag{8}$$

$$\mathbf{h}_i^{(l+1)} = \mathrm{LayerNorm}\left(\mathbf{h}_i^{(l)} + \mathrm{FC}\left(\mathbf{z}_i^{(l+1)} \oplus \mathbf{s}_{v_i}^{(l+1)}\right)\right), \tag{9}$$

where s_{v_i} denotes the hypernode for node i, $\oplus$ denotes concatenation, and FC denotes a linear fusion layer.

4.3 Inter-variable Fine-grained Correlation Capturer

We first introduce a Dynamic Neighbor Selection mechanism to determine which observation nodes across variables should be connected, and then employ Spatio-Temporal Coordinate-aware Attention to encode fine-grained correlations among observation nodes across variables.

Causal-Constrained Dynamic Neighbor Selection. For each observation node, the dynamic neighbor selection mechanism only adaptively selects the most relevant K neighbors from all observation nodes to connect it. The distance between nodes is measured by a hybrid metric based on the static spatio-temporal coordinates $\mathbf{p}$ and dynamic latent states $\mathbf{h}$. At layer l, the distance $\mathbf{D}_{ij}^{(l)}$ between node i and node j is defined as:

$$\mathbf{D}_{ij}^{(l)} = \hat{d}(\mathbf{p}_i, \mathbf{p}_j) + \omega^{(l)} \cdot \hat{d}(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}), \tag{10}$$

where $\hat{d}(\cdot,\cdot)$ denotes the squared Euclidean distance, and $\omega^{(l)}$ denotes the balancing weight which is defined as a linearly increasing function of network depth: $\omega^{(l)} = \lambda \times l$, where λ is a hyperparameter. This design reflects the increasing discriminability of latent representations at greater network depths. Subsequently, we utilize a causal mask to prevent information leakage from future time steps, along with a padding mask to filter out padding nodes. The final distance matrix is given by:

$$\mathbf{D}_{ij}^{(l)} = \begin{cases}\infty, & \text{if } t_j > t_i \text{ or node } j \text{ is padded}, \\ \mathbf{D}_{ij}^{(l)}, & \text{otherwise}.\end{cases} \tag{11}$$

Based on the distance matrix, we obtain a set of K neighbors $\mathcal{N}_i^{(l)}$ closest to each observation node i.

Spatio-Temporal Coordinate-aware Attention. For each observation node i, the Spatio-Temporal Coordinate-aware Attention mechanism is used to aggregate messages within the neighbors $\mathcal{N}_i^{(l)}$.

Specifically, we employ the multi-head attention to model the spatio-temporal correlation between node i and its neighbor j based on the relative distance $\mathbf{p}_{rel} = \mathbf{p}_i - \mathbf{p}_j$. The Query, Key, and Value representations are formulated as:

$$\begin{aligned}\mathbf{Q}_i &= \mathbf{h}_i^{(l)}\mathbf{W}_Q, \\ \mathbf{K}_j &= (\mathbf{h}_j^{(l)} \oplus \mathbf{p}_{rel})\mathbf{W}_K, \\ \mathbf{V}_j &= (\mathbf{h}_j^{(l)} \oplus \mathbf{p}_{rel})\mathbf{W}_V.\end{aligned} \tag{12}$$

And the node representation is updated by:

$$\mathbf{z}_i^{(l)} = \sum_{j\in\mathcal{N}_i^{(l)}} \mathrm{Softmax}\left(\frac{\mathbf{Q}_i\mathbf{K}_j^\top}{\sqrt{d_k}}\right)\mathbf{V}_j, \tag{13}$$

$$\mathbf{h}_i^{(l+1)} = \mathrm{LayerNorm}\left(\mathbf{h}_i^{(l)} + \mathrm{MLP}(\mathbf{z}_i^{(l)})\right). \tag{14}$$

4.4 Inter-variable Coarse-grained Correlation Capturer

After the iterations of the Intra-variable Coarse-to-fine and Inter-variable Fine-grained Correlation Capturers, we employ a self-attention mechanism to capture inter-variable coarse-grained correlation among variable hypernodes, defined as:

$$\mathbf{s} = \mathrm{MHA}\Big(\mathbf{Q} = \mathbf{s}^{(L)}, \mathbf{K} = \mathbf{s}^{(L)}, \mathbf{V} = \mathbf{s}^{(L)}\Big), \tag{15}$$

where $\mathbf{s}^{(L)} \in \mathbb{R}^{M\times d_{model}}$ represents the hidden states of the M variable hypernodes output by the Inter-variable Fine-grained Correlation Capturer after L layers of processing.

4.5 Joint Multivariate Forecasting

In the final stage, we employ a joint decoder for multivariate forecasting. For target node $i \in \mathcal{Y}$, its final representation $\mathbf{h}_i^{(L)}$ and the final representation of its associated hypernode $\mathbf{s}_{v_i}$ are fused together to predict its value $\hat{y}_i$:

$$\hat{y}_i = \mathrm{FC}_{\mathrm{out}}\left(\mathbf{h}_i^{(L)} \oplus \mathbf{s}_{v_i}\right), \tag{16}$$

where $\mathrm{FC}_{\mathrm{out}}$ denotes a linear projection layer. The model is trained by minimizing the Mean Squared Error (MSE) between the predicted values $\hat{\mathcal{Z}}$ and the corresponding ground truth $\mathcal{Z}$ for the forecast queries.

Dataset	**PhysioNet'12**		**MIMIC-III**		**HumanActivity**		**USHCN**	
Metric	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
PrimeNet	0.803±0.000	0.688±0.000	0.875±0.000	0.646±0.000	4.372±0.006	1.728±0.001	0.494±0.002	0.496±0.002
SeFT	0.781±0.002	0.679±0.002	0.899±0.001	0.649±0.002	1.410±0.013	0.990±0.005	0.335±0.004	0.408±0.013
GNeuralFlow	0.788±0.141	0.660±0.054	0.880±0.040	0.631±0.018	0.411±0.328	0.461±0.155	0.223±0.055	0.331±0.051
mTAN	0.388±0.006	0.433±0.004	0.946±0.136	0.674±0.053	0.096±0.006	0.223±0.010	0.666±0.153	0.523±0.068
CRU	0.631±0.005	0.585±0.004	0.728±0.008	0.575±0.004	0.139±0.012	0.262±0.013	0.214±0.028	0.319±0.027
NeuralFlows	0.408±0.007	0.449±0.006	0.634±0.021	0.531±0.009	0.176±0.014	0.317±0.015	0.209±0.042	0.316±0.030
GRU_D	0.349±0.002	0.405±0.001	0.492±0.022	0.453±0.009	0.174±0.045	0.312±0.037	0.213±0.058	0.308±0.039
Raindrop	0.349±0.003	0.406±0.003	0.583±0.191	0.496±0.093	0.089±0.009	0.209±0.012	0.209±0.044	0.288±0.015
Hi-Patch	0.307±0.003	0.366±0.002	0.453±0.008	0.411±0.005	0.049±0.001	0.128±0.002	0.189±0.023	0.275±0.034
tPatchGNN	0.306±0.003	0.364±0.001	0.447±0.013	0.406±0.009	0.045±0.002	0.125±0.003	0.182±0.073	0.299±0.070
Warpformer	0.300±0.006	0.362±0.009	0.424±0.007	0.393±0.006	0.046±0.001	0.125±0.003	0.191±0.105	0.296±0.102
APN	0.308±0.002	0.364±0.003	0.433±0.004	0.402±0.002	<u>0.041±0.013</u>	**0.116±0.006**	0.163±0.022	0.263±0.032
HyperIMTS	<u>0.294±0.002</u>	<u>0.356±0.002</u>	<u>0.363±0.002</u>	<u>0.365±0.003</u>	0.047±0.003	0.132±0.007	0.185±0.016	0.280±0.014
GraFITi	0.300±0.003	0.359±0.006	0.377±0.002	0.373±0.003	0.044±0.001	0.124±0.002	0.174±0.018	0.272±0.016
ASTGI	0.297±0.004	0.357±0.003	0.408±0.015	0.392±0.011	**0.041±0.001**	0.118±0.001	<u>0.159±0.009</u>	<u>0.261±0.019</u>
Ours	**0.290±0.001**	**0.352±0.001**	**0.350±0.003**	**0.355±0.004**	**0.041±0.001**	<u>0.117±0.004</u>	**0.155±0.012**	**0.254±0.019**

Table 1: Forecasting performance on four IMTS datasets. Overall performance is evaluated by MSE and MAE (mean ± std). The best and second-best results are highlighted in **bold** and with an <u>underline</u>, respectively.

5 Experiment

5.1 Experimental Setup

Datasets. We conduct experiments on four widely-used public IMTS datasets from various domains, including healthcare, biomechanics, and climate science. The PhysioNet'12 dataset provides clinical time series from the first 48 hours of ICU stays, which is rounded for 1 hour. MIMIC-III is also a clinical database collected from ICU patients during the first 48 hours of admission, which is rounded for 30 minutes. The HumanActivity dataset consists of biomechanics data with 3D positional variables captured from subjects performing various activities, which is rounded for 1 millisecond. USHCN includes climate data over 150 years collected by meteorological stations scattered over the United States, and we focus on a subset of 4 years between 1996 and 2000. To ensure fairness and comparability of the results, all datasets are preprocessed by strictly following the standard procedures established in prior state-of-the-art works [Li *et al.*, 2025; Yalavarthi *et al.*, 2024]. In line with established practices, all datasets are partitioned into training, validation, and test sets using a standard 80%, 10%, and 10% ratio, respectively. Training and validation sets are shuffled, while test sets are not.

Baselines. To evaluate the performance of DyH2-STGraph, we selected a total of fifteen state-of-the-art models designed for irregular time series as baselines. These include: (1) RNN and ODE-based methods, including GRU-D [Che *et al.*, 2018], CRU [Schirmer *et al.*, 2022], NeuralFlows [Biloš *et al.*, 2021] and GNeuralFlow [Mercatali *et al.*, 2024], (2) Set-based and Kernel-attention methods, including mTAN [Shukla and Marlin, 2021] and SeFT [Horn *et al.*, 2020], (3) Patch-based methods, including tPatchGNN [Zhang *et al.*, 2024], Hi-Patch [Luo *et al.*, 2025] and APN [Liu *et al.*, 2026b], (4) Warping and hybrid Transformer methods, including Warpformer [Zhang *et al.*, 2023], (5) Graph-based methods, including Raindrop [Zhang *et al.*, 2022], PrimeNet [Chowdhury *et al.*, 2023], GraFITi [Yalavarthi *et al.*, 2024], ASTGI [Liu *et al.*, 2026a] and HyperIMTS [Li *et al.*, 2025].

Implementation Details. All models are trained on a Linux server with PyTorch version 2.6.0+cu124 using four NVIDIA Tesla V100-DGXS-32GB GPUs. All experiments run with a maximum epoch number of 300 and early stopping patience of 10 epochs. All models are trained using the Mean Squared Error (MSE) as the loss function and optimized with the Adam optimizer. The learning rate $\mathcal{L}_n$ for the n-th epoch is kept unchanged when $n \leq 3$, and adjusted according to $\mathcal{L}_n = \mathcal{L}_0 \times 0.8^{n-3}$ when $n > 3$. To ensure a fair comparison, we primarily adopted the hyperparameter settings reported in the original papers for the baseline models. To mitigate randomness, each experiment is run independently with five different random seeds from 2024 to 2028, and we report the mean and standard deviation. Detailed hyperparameter configurations for all models are provided in the Appendix A.

5.2 Main Results

Table 1 shows the models' forecasting performance across four datasets. We have the following observations: (1) DyH2-STGraph achieves leading prediction accuracy across the board. On all datasets, DyH2-STGraph achieves the best or second-best forecasting performance. Compared to the overall next-best model, ASTGI, DyH2-STGraph reduces the

Dataset	PhysioNet'12		MIMIC-III		HumanActivity		USHCN	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
w/o Inter-F	0.387 ± 0.002	0.429 ± 0.001	0.410 ± 0.006	0.402 ± 0.002	0.070 ± 0.001	0.165 ± 0.003	0.162 ± 0.015	0.267 ± 0.038
w/o Intra	0.300 ± 0.002	0.358 ± 0.001	0.376 ± 0.003	0.368 ± 0.004	0.043 ± 0.001	0.123 ± 0.004	0.176 ± 0.024	0.279 ± 0.024
w/o SD	0.290 ± 0.002	0.352 ± 0.001	0.353 ± 0.007	0.358 ± 0.005	0.042 ± 0.002	0.118 ± 0.008	0.158 ± 0.011	0.257 ± 0.023
w/o Inter-C	0.292 ± 0.002	0.353 ± 0.001	0.353 ± 0.007	0.357 ± 0.004	0.042 ± 0.001	0.117 ± 0.003	0.157 ± 0.007	0.261 ± 0.019
Complete	$\mathbf{0.290 \pm 0.001}$	$\mathbf{0.352 \pm 0.001}$	$\mathbf{0.350 \pm 0.003}$	$\mathbf{0.355 \pm 0.004}$	$\mathbf{0.041 \pm 0.000}$	$\mathbf{0.117 \pm 0.004}$	$\mathbf{0.155 \pm 0.012}$	$\mathbf{0.254 \pm 0.019}$

Table 2: Ablation results of DyH2-STGraph and its four variants on four irregular multivariate time series datasets.

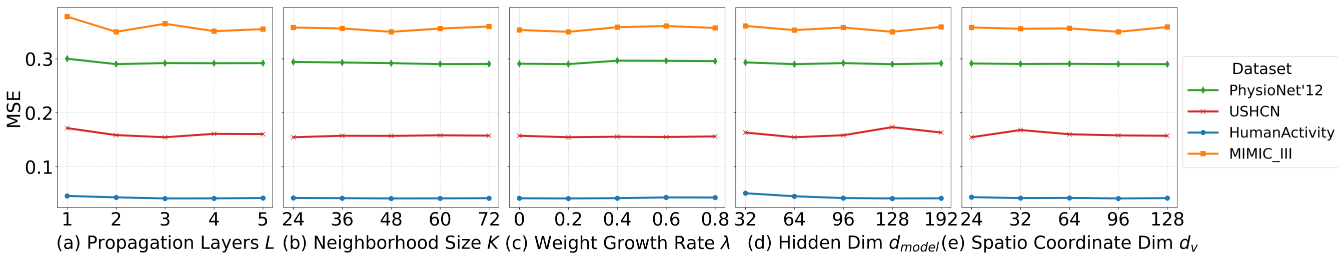

Figure 2: Parameter sensitivity studies of five hyper-parameters on four irregular multivariate time series datasets.

MSE and MAE metrics by approximately 6.9% and 5.0%, respectively. (2) DyH2-STGraph exhibits remarkable stability. This is evidenced by its consistently lower standard deviations across all metrics and datasets, suggesting the model is less sensitive to initialization or data noise. (2) DyH2-STGraph demonstrates exceptional generalization and robustness across diverse domains. Whether on IMTS datasets with different characteristics in healthcare, biomechanics, or climate science, DyH2-STGraph consistently remains at the top of the leaderboard. The outstanding performance of DyH2-STGraph can be attributed to its dynamic hierarchical heterogeneous spatio-temporal graph structure. By effectively decoupling and capturing the intricate dependencies of IMTS through three distinct categories of edge components, our model successfully addresses the challenges of cross-variable misalignment and missing correlations.

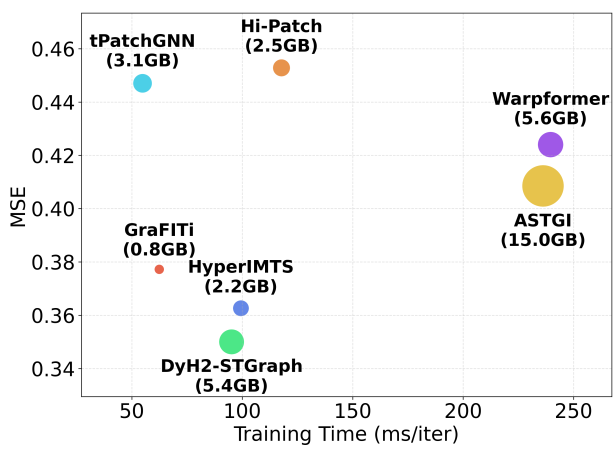

Figure 3: Model efficiency comparison on MIMIC-III, with 36 hours of lookback length, 3 forecast timestamps, 96 variables, and a batch size of 16.

5.3 Ablation Study

To investigate the contribution of each component, we compare DyH2-STGraph with four variants across all benchmark datasets. (1) **w/o Intra** removes the Intra-variable Coarse-to-fine Correlation Capturer. (2) **w/o Inter-F** removes the Inter-variable Fine-grained Correlation Capturer. (3) **w/o Inter-C** removes the Inter-variable Coarse-grained Correlation Capturer. (4) **w/o SD** disables the dynamic feature distance within the neighbor selection process to rely solely on static spatio-temporal coordinates. As summarized in Table 2, DyH2-STGraph outperforms all variants, confirming that every component is essential. Specifically, **w/o Inter-F** causes the most significant performance degradation, identifying the capture of fine-grained dependencies between neighboring observation nodes as the primary driver of our model. **w/o Intra** highlights the necessity of hierarchical message passing for maintaining temporal consistency within individual variables, while **w/o Inter-C** proves that correlations between variable hypernodes provide important high-level guidance for final forecasting. Finally, the performance gap in **w/o SD** confirms the effectiveness of dynamic neighbor selection mechanism, where the incorporation of feature distance (d_h) allows the model to adaptively capture evolving relationships as the network depth increases.

5.4 Parameter Sensitivity

To evaluate the robustness of the proposed DyH2-STGraph framework, we conducted a sensitivity analysis on five core hyperparameters: the number of propagation layers L, the number of neighbors K, the weight growth rate λ, the hidden dimension d_{model}, and the spatial coordinate embedding dimension d_v. As illustrated in Figure 2, the model maintains consistent performance across diverse configurations. (1) L governs the depth of stacking interaction layers. Performance

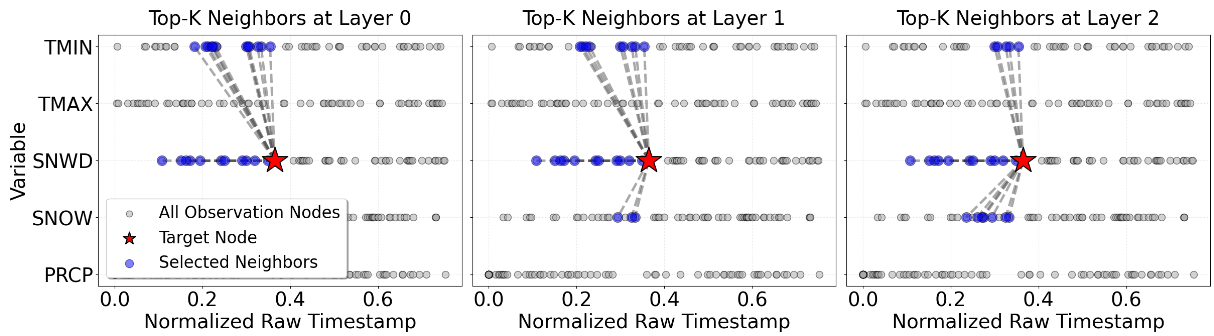

Figure 4: Case study of the dynamic neighbor selection mechanism. The subplots illustrate the evolution of top-K neighbors of a target observation node across three successive layers on the USHCN dataset.

typically peaks at $L \in \{2, 3\}$, while $L = 1$ is insufficient and $L = 5$ leads to over-smoothing. (2) K determines the neighborhood size. MSE remains relatively stable as K increases, indicating the model's ability to identify informative neighbors while filter out noise. (3) λ is the growth rate of the layer-wise balancing weight $\omega^{(l)} = \lambda \times l$ (defined in Eq. 10), which controls the relative contribution of dynamic latent states to the distance metric across layers. Performance peaks at $\lambda = 0.2$, while $\lambda = 0$ degenerates to purely static distance and larger values over-emphasize dynamic states. (4) d_{model} and d_v define the model's expressive power. An optimal balance is achieved where the dimensions are large enough to encode complex patterns but avoid overfitting. Overall, DyH2-STGraph demonstrates strong stability across various configurations, confirming its practical applicability without the need for exhaustive hyperparameter tuning.

5.5 Efficiency Analysis

To evaluate the efficiency of DyH2-STGraph, we conduct a comparative analysis against the most competitive baselines on MIMIC-III dataset. From a data preprocessing perspective, these methods can be categorized as follows: (1) Canonical padding methods: Warpformer, (2) Patch-aligned padding method: tPatchGNN and Hi-Patch, (3) Non-padding methods: HyperIMTS, GraFITi, ASTGI and DyH2-STGraph. All models are evaluated based on their predictive accuracy (MSE), training time (ms/iter) and peak GPU memory usage. As illustrated in Figure 3, DyH2-STGraph achieves the state-of-the-art MSE, outperforming all baseline models while maintaining a highly competitive computational efficiency. Specifically, compared to heavy-duty architectures like ASTGI and Warpformer, DyH2-STGraph yields a significant reduction in training time from over 230 ms/iter to approximately 95 ms/iter, while requiring substantially less GPU memory (5.4 GB vs. 15.0 GB). Although lightweight models such as GraFITi and HyperIMTS exhibit lower memory footprints and training time, they suffer from higher prediction errors. Overall, DyH2-STGraph achieves a superior trade-off between predictive precision and computational efficiency. By avoiding the prohibitive costs of heavy-duty models and the fidelity loss of lightweight methods, it establishes a practical and robust paradigm for high-dimensional IMTS forecasting task.

5.6 Case Study

To qualitatively evaluate the dynamic neighbor selection in DyH2-STGraph, we conduct a case study on the USHCN dataset. We track the top-K neighbors of a target Snow Depth (SNWD) node ($t \approx 0.36$) across successive layers, as illustrated in Figure 4. The visualization confirms that all selected neighbors strictly adhere to causal constraints. In Layer 0, the neighbors are primarily concentrated within the SNWD variable itself and a few Minimum Temperature (TMIN) nodes. As the network deepens, the target node broadens its receptive field to incorporate an increasing number of neighbors from the Snowfall (SNOW) variable, while the connections with TMIN relatively decreases. This evolution aligns with meteorological principles: snow depth is governed by both snowfall (SNOW) for accumulation and minimum temperature (TMIN) for melting effects, demonstrating the model's capacity to select relevant neighbors across variables. Meanwhile, the model's increasing focus on SNOW in deeper layers highlights its ability to prioritize the most direct physical drivers. In summary, the proposed neighbor selection mechanism allows the model to adaptively balance static spatio-temporal relations with dynamic feature similarity, enhancing its capacity to effectively capture crucial, fine-grained inter-variable dependencies within the spatio-temporal graph.

6 Conclusion

This paper introduces DyH2-STGraph to address the fundamental challenges of IMTS forecasting. Representing variables as hypernodes and observations as spatio-temporal nodes allows the framework to preserve data fidelity, bypassing the distortions of traditional alignment-first strategies. By employing a three-fold decoupling strategy, the model captures intra-variable coarse-to-fine dependencies alongside multi-grained inter-variable correlations, overcoming the correlation loss prevalent in existing static models. Furthermore, the dynamic neighbor selection mechanism enables the model to adaptively capture evolving correlations as the network depth increases. Extensive experiments across four benchmark datasets demonstrate that DyH2-STGraph achieves state-of-the-art accuracy and stability, establishing a robust new paradigm for high-dimensional, asynchronous time series analysis.

Acknowledgments

This study is partially supported by National Natural Science Foundation of China (62276082), National Key R&D Program of China (2023YFC3502900), Shenzhen Science and Technology Research and Development Fund (KJZD20240903102802003), Shenzhen Science and Technology Research and Development Fund for Sustainable Development Project (GXWD20231128103819001, 20230706140548006) and Guangdong Provincial Key Laboratory Grant (2023B1212060076).

Contribution Statement

Xiaowei Yan and Zhuo Li contributed equally to this manuscript and are recognized as joint first authors.

References

[Ansari *et al.*, 2023] Abdul Fatir Ansari, Alvin Heng, Andre Lim, and Harold Soh. Neural continuous-discrete state space models for irregularly-sampled time series. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 926–951. PMLR, 2023.

[Biloš *et al.*, 2021] Marin Biloš, Johanna Sommer, Syama Sundar Rangapuram, Tim Januschowski, and Stephan Günnemann. Neural flows: Efficient alternative to neural odes. *Advances in neural information processing systems*, 34:21325–21337, 2021.

[Cao *et al.*, 2020] Defu Cao, Yujing Wang, Jianhui Duan, Chao Zhang, Xiaoping Zhu, Chang Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, et al. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in Neural Information Processing Systems*, 33:17766–17778, 2020.

[Che *et al.*, 2018] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8(1):6085, 2018.

[Chen *et al.*, 2018] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[Chen *et al.*, 2021] Yuzhou Chen, Ignacio Segovia, and Yulia R. Gel. Z-gcnets: Time zigzags at graph convolutional networks for time series forecasting. In *International Conference on Machine Learning*, PMLR, pages 1684–1694, 2021.

[Chowdhury *et al.*, 2023] Rakshith R Chowdhury, Jialu Li, Xiyuan Zhang, Di Hong, Rajesh K Gupta, and Jingbo Shang. Primenet: Pre-training for irregular multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7184–7192, 2023.

[Cini *et al.*, 2021] 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*, 2021.

[De Brouwer *et al.*, 2019] Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. Gru-ode-bayes: Continuous modeling of sporadically-observed time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7377–7388, 2019.

[De Gooijer and Hyndman, 2006] Jan G. De Gooijer and Rob J. Hyndman. 25 years of time series forecasting. *International Journal of Forecasting*, 22(3):443–473, 2006.

[Han *et al.*, 2024] Jialin Han, Wenxuan Zhang, Haotian Liu, Tiantian Tao, Ning Tan, and Hui Xiong. Bigst: Linear complexity spatio-temporal graph neural network for traffic forecasting on large-scale road networks. *Proceedings of the VLDB Endowment*, 17(5):1081–1090, 2024.

[Horn *et al.*, 2020] M. Horn, M. Moor, C. Bock, B. Rieck, and K. M. Borgwardt. Set functions for time series. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119, pages 4353–4363, 2020.

[Kidger *et al.*, 2020] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in neural information processing systems*, 33:6696–6707, 2020.

[Li and Marlin, 2020] Steven Cheng-Xian Li and Benjamin M. Marlin. Learning from irregularly-sampled time series: A missing data perspective. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 5937–5946. PMLR, 2020.

[Li *et al.*, 2025] Boyuan Li, Yicheng Luo, Zhen Liu, Junhao Zheng, Jianming Lv, and Qianli Ma. Hyperimts: Hypergraph neural network for irregular multivariate time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

[Liu *et al.*, 2024] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

[Liu *et al.*, 2026a] Xvyuan Liu, Xiangfei Qiu, Hanyin Cheng, Xingjian Wu, Chenjuan Guo, Bin Yang, and Jilin Hu. Astgi: Adaptive spatio-temporal graph interactions for irregular multivariate time series forecasting, 2026.

[Liu *et al.*, 2026b] Xvyuan Liu, Xiangfei Qiu, Xingjian Wu, Zhengyu Li, Chenjuan Guo, Jilin Hu, and Bin Yang. Rethinking irregular time series forecasting: A simple yet effective baseline. In *Proceedings of the 40th AAAI Conference on Artificial Intelligence*, 2026. Oral Presentation.

[Luo *et al.*, 2025] Yicheng Luo, Bowen Zhang, Zhen Liu, and Qianli Ma. Hi-patch: Hierarchical patch gnn for irregular multivariate time series. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

[Mercatali *et al.*, 2024] G. Mercatali, A. Freitas, and J. Chen. Graph neural flows for unveiling systemic interactions among irregularly sampled time series. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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

[Satorras *et al.*, 2022] Victor Garcia Satorras, Syama Sundar Rangapuram, and Tim Januschowski. Multivariate time series forecasting with latent graph inference, 2022.

[Schirmer *et al.*, 2022] M. Schirmer, M. Eltayeb, S. Lessmann, and M. Rudolph. Modeling irregular time series with continuous recurrent units. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162, pages 19388–19405, 2022.

[Shukla and Marlin, 2020] Satya Narayan Shukla and Benjamin M. Marlin. A survey on principles, models and methods for learning from irregularly sampled time series: From discretization to attention and invariance. *CoRR*, abs/2012.00168, 2020.

[Shukla and Marlin, 2021] Satya Narayan Shukla and Benjamin Marlin. Multi-time attention networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2021.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[Vio *et al.*, 2013] Roberto Vio, Mara Daz-Trigo, and Paola Andreani. Irregular time series in astronomy and the use of the lomb-scargle periodogram. *Astronomy and Computing*, 1:5–16, 2013.

[Wen *et al.*, 2023] Haomin Wen, Youfang Lin, Yutong Xia, Huaiyu Wan, Qingsong Wen, Roger Zimmermann, and Yuxuan Liang. Diffstg: Probabilistic spatio-temporal graph forecasting with denoising diffusion models. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, pages 1–12. ACM, 2023.

[Wu *et al.*, 2020] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 753–763, 2020.

[Yalavarthi *et al.*, 2024] Vijaya Krishna Yalavarthi, Kiran Madhusudhanan, Randolf Scholz, Nourhan Ahmed, Johannes Burchert, Shayan Jawed, Stefan Born, and Lars Schmidt-Thieme. Grafiti: Graphs for forecasting irregularly sampled time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16255–16263, 2024.

[Yao *et al.*, 2018] Zhenjie Yao, Jie Bi, and Yixin Chen. Applying deep learning to individual and community health monitoring data: A survey. *International Journal of Automation and Computing*, 15(6):643–655, 2018.

[Yi *et al.*, 2023] Kun Yi, Qi Zhang, Wei Fan, Hui He, Liang Hu, Peng Wang, Ning An, Longbing Cao, and Zhendong Niu. Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective. In *Advances in Neural Information Processing Systems*, volume 36, pages 15147–15159, 2023.

[Zeng *et al.*, 2022] Ailing Zeng, Mu-Hwa Chen, L. Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *AAAI Conference on Artificial Intelligence*, 2022.

[Zhang *et al.*, 2022] Xiang Zhang, Marko Zeman, Theodoros Tsiligkaridis, and Marinka Zitnik. Graph-guided network for irregularly sampled multivariate time series. In *International Conference on Learning Representations (ICLR)*, 2022.

[Zhang *et al.*, 2023] Jiawen Zhang, Shun Zheng, Wei Cao, Jiang Bian, and Jia Li. Warpformer: A multi-scale modeling approach for irregular clinical time series. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 3273–3285, 2023.

[Zhang *et al.*, 2024] Weijia Zhang, Chenlong Yin, Hao Liu, Xiaofang Zhou, and Hui Xiong. Irregular multivariate time series forecasting: A transformable patching graph neural networks approach. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.