Irregular Multivariate Time Series Forecasting: A Transformable Patching Graph Neural Networks Approach

Weijia Zhang [1] **Chenlong Yin** [1] **Hao Liu** [1 2] **Xiaofang Zhou** [2] **Hui Xiong** [1 2]

Abstract

Forecasting of Irregular Multivariate Time Series (IMTS) is critical for numerous areas, such as healthcare, biomechanics, climate science, and astronomy. Despite existing research addressing irregularities in time series through ordinary differential equations, the challenge of modeling correlations between asynchronous IMTS remains underexplored. To bridge this gap, this study proposes Transformable Patching Graph Neural Networks (T-PATCHGNN), which transforms each univariate irregular time series into a series of transformable patches encompassing a varying number of observations with uniform temporal resolution. It seamlessly facilitates local semantics capture and inter-time series correlation modeling while avoiding sequence length explosion in aligned IMTS. Building on the aligned patching outcomes, we then present time-adaptive graph neural networks to model dynamic inter-time series correlation based on a series of learned time-varying adaptive graphs. We demonstrate the remarkable superiority of T-PATCHGNN on a comprehensive IMTS forecasting benchmark we build, which contains four real-world scientific datasets covering healthcare, biomechanics and climate science, and seventeen competitive baselines adapted from relevant research fields.[1]

1. Introduction

While the forecasting of Multivariate Time Series (MTS) has been extensively investigated, most research focuses on regularly sampled and fully observed MTS (Lim & Zohren, 2021). The forecasting challenges associated with Irregular Multivariate Time Series (IMTS), characterized by their irregular sampling intervals and missing data, have received significantly less attention. Indeed, IMTS are prevalent across a wide range of subject areas, such as healthcare, biomechanics, climate science, astronomy, and finance (Rubanova et al., 2019; De Brouwer et al., 2019; Yao et al., 2018; Vio et al., 2013; Engle & Russell, 1998; Zhang et al., 2021a). Accurate forecasting of IMTS serves as the foundation to support various significant activities from making informed decisions to planning with foresight.

Unlike regular MTS, the modeling and analysis for IMTS is more challenging due to the inherent irregularity within the series and asynchrony between them (Horn et al., 2020). As illustrated in Figure 1(a), given a set of historical IMTS observations and forecasting queries, the IMTS forecasting problem aims to accurately predict the values in correspondence to these queries. Although a few proactive efforts have been made for IMTS forecasting (Rubanova et al., 2019; De Brouwer et al., 2019; Biloš et al., 2021; Schirmer et al., 2022), these works mainly focus on handling irregularity within the time series based on neural Ordinary Differential Equations (ODEs) (Chen et al., 2018), failing to explicitly consider the crucial correlations between multiple series. Moreover, calculating ODE solvers is computationally expensive due to the numerical integration process, leading to poor efficiency in both training and inference stages (Biloš et al., 2021; Shukla & Marlin, 2020).

It is a non-trivial task for accurate IMTS forecasting, which faces three major challenges. (1) The first challenge is the *irregularity in intra-time series dependency modeling*. The varying time intervals between adjacent observations disrupt the consistent flow of time series data, making it difficult for classical time series forecasting models (Lim & Zohren, 2021) to accurately capture the underlying temporal dynamics and dependencies (Rubanova et al., 2019; Che et al., 2018). (2) The second challenge is the *asynchrony in inter-time series correlation modeling*. While there are always considerable correlations between time series of different variables, the observations among IMTS can be significantly misaligned at time due to irregular sampling or missing data.

[1]The Hong Kong University of Science and Technology (Guangzhou) [2]The Hong Kong University of Science and Technology. Correspondence to: Hao Liu <liuh@ust.hk>, Hui Xiong <xionghui@ust.hk>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[1]Code and datasets are available at https://github.com/usail-hkust/t-PatchGNN.

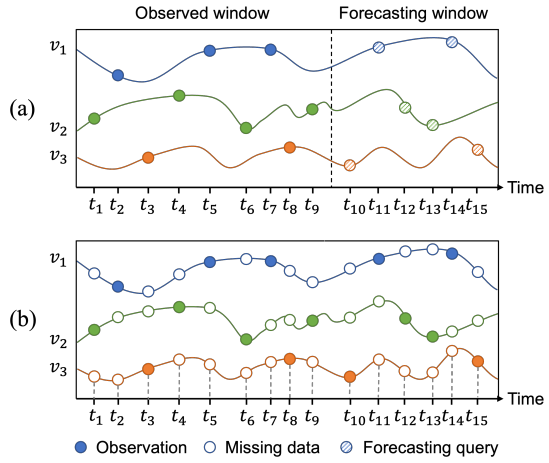


Figure 1: (a) Irregular multivariate time series forecasting problem, where v_1, v_2, and v_3 represent three different variables. (b) Canonical pre-alignment representation causes the average sequence length to increase from 5 to 15, an explosive growth proportional to the variable count.

This asynchrony complicates direct comparisons and correlations at specific time points and potentially obscures or distorts the actual relationships between the time series, resulting in a significant challenge to model inter-time series correlations (Zhang et al., 2021b). (3) The last challenge is the *sequence length explosion with the increase of variables*. As shown in Figure 1(b), to facilitate IMTS modeling, current studies typically represent IMTS in a time-aligned format which involves extending each univariate irregular time series to a uniform length corresponding to the count of all unique timestamps among IMTS observations (Che et al., 2018). However, such a canonical pre-alignment representation may lead to the sequence length explosively growing proportional to the addition of variables, which raises severe scalability concerns on both computation and memory overhead when encountering a large number of variables.

To this end, we propose a Transformable Patching Graph Neural Networks (T-PATCHGNN) approach for IMTS forecasting. T-PATCHGNN initially transforms each univariate irregular time series into a series of transformable patches, which vary in observation count but maintain a unified time horizon resolution. This process for IMTS offers three major advantages: (1) The independent patching process for each univariate irregular time series bypasses the canonical pre-alignment representation for IMTS, eliminating the risk of sequence length explosion in the representation of IMTS with large-scale variables; (2) local semantics of irregular time series can be better captured by putting each individual observation into patches with richer context (Nie et al., 2022); (3) after transformable patching, the IMTS is naturally aligned in a consistent patch-level temporal resolution. It addresses the asynchrony problem, seamlessly facilitating subsequent inter-time series correlation modeling.

Along this line, a transformable time-aware convolution network is introduced to encode each transformable patch into a latent embedding, which subsequently serves as input tokens to a Transformer for intra-time series dependency modeling. Furthermore, we present time-adaptive graph neural networks to model the inter-time series correlation. To explicitly represent the dynamic correlations between IMTS, we learn a series of time-varying adaptive graphs constructed based on both the learnable inherent variable embedding and dynamic patch embedding, and consequently, these graphs keep the same temporal resolution as transformable patches. Then, graph neural networks are applied to these learned graphs to model patch-level dynamic correlations between IMTS. Finally, a Multi-Layer Perception (MLP) output layer is employed to generate predicted results in terms of forecasting queries based on the obtained comprehensive latent representation of IMTS.

Our major contributions are summarized as follows:

- We propose a new transformable patching method to transform each univariate irregular time series of IMTS into a series of variable-length yet time-aligned patches. This tactfully bypasses the canonical pre-alignment representation for IMTS while aligning IMTS in a consistent temporal resolution. It prevents the sequence length of aligned IMTS from explosively growing proportional to the increasing variables, and meanwhile, seamlessly facilitates local semantics capture and inter-time series correlation modeling for IMTS.
- Based on the transformable patching outcomes, we propose time-adaptive graph neural networks to model the dynamic inter-time series correlation within IMTS.
- We build a benchmark for IMTS forecasting evaluation. Seventeen state-of-the-art baseline models from various relevant research fields, *i.e.*, IMTS forecasting, interpolation, classification, and MTS forecasting, are taken for a fair comparison on four public scientific IMTS datasets, which cover areas of healthcare, biomechanics, and climate science. Extensive experiments demonstrate remarkable superiority of T-PATCHGNN.

2. Related Works

2.1. Irregular Multivariate Time Series Forecasting

Existing efforts on IMTS primarily focus on classification tasks (Che et al., 2018; Shukla & Marlin, 2021; Zhang et al., 2021b; 2023a; Horn et al., 2020; Shukla & Marlin, 2018; Li et al., 2023; Baytas et al., 2017). Only a few proactive studies (Rubanova et al., 2019; De Brouwer et al., 2019; Biloš et al., 2021; Schirmer et al., 2022) have made efforts

on the IMTS forecasting. Specifically, these works primarily rely on neural ODEs (Chen et al., 2018) and focus on handling the continuous dynamics and irregularity within the time series. For instance, Latent-ODE (Rubanova et al., 2019) enables Recurrent Neural Networks (RNNs) to have continuous-time hidden state dynamics specified by neural ODEs. GRU-ODE-Bayes (De Brouwer et al., 2019) incorporates neural ODEs to develop a continuous-time Gated Recurrent Unit (GRU) and introduces a Bayesian update network to process the sparse observations. CRU (Schirmer et al., 2022) handles irregular intervals between observations by evolving the hidden state based on a linear stochastic differential equation and the continuous-discrete Kalman filter. However, calculating ODE solvers is known to be low-efficient due to the expensive numerical integration computation. To address this, Neural Flows (Biloš et al., 2021) models the solution curves of ODEs through neural networks to mitigate the expensive numerical solvers in neural ODEs. While these works have made big efforts to handle the irregularity within irregular time series, it is still underexplored to effectively model the inter-time series correlations within asynchronous IMTS.

2.2. Irregular Multivariate Time Series Representation

To represent IMTS in a time-aligned manner and facilitate the subsequent modeling, existing works predominantly adopt a pre-alignment representation method (Che et al., 2018; Shukla & Marlin, 2021; Zhang et al., 2021b; 2023a; Baytas et al., 2017; Rubanova et al., 2019; De Brouwer et al., 2019; Biloš et al., 2021; Schirmer et al., 2022). It involves extending all univariant series in IMTS to a consistent sequence length that equals the number of all unique timestamps in IMTS and indicating the missing values with mask terms (Che et al., 2018). However, with the number of variables increasing, such a representation method may suffer from the sequence length explosion problem, which is detailed in Section 3.2, raising severe scalability concerns on both computation and memory overhead. Beyond the pre-alignment representation, Horn et al. (2020) introduce a more scalable representation method by regarding observations of IMTS as a set of tuples comprising of time, value, and variable indicator, and then these tuples are summarized for the IMTS classification. However, this representation method may not be suitable for the forecasting task that requires each variable to be more meticulously and distinctively analyzed.

2.3. Graph Neural Networks for Multivariate Time Series

Graph Neural Networks (GNNs) are introduced to MTS for their powerful capability to model complicated correlations between variables (Li et al., 2018; Yu et al., 2018; Wu et al., 2019; 2020b; Huang et al., 2023; Yi et al., 2023; Cao et al., 2020; Liu et al., 2022). DCRNN (Li et al., 2018) and STGCN (Yu et al., 2018) apply GNNs to the pre-defined graph structures, which may be difficult to obtain in some domains. Therefore, some studies (Wu et al., 2019; 2020b; Huang et al., 2023; Yi et al., 2023; Cao et al., 2020) propose to learn graph structures from data, enabling automatic modeling of variables' topological relationships. However, when it comes to IMTS, the observations can be notably misaligned at times, raising challenges for the inter-time series correlation modeling. Raindrop (Zhang et al., 2021b) addresses it by propagating the asynchronous observations at all the timestamps when an observation appears at an arbitrary variable, which involves the IMTS pre-alignment and may suffer from the sequence length explosion problem.

Another line of works associated with us applies GNNs for modeling regular MTS with missing data (Cini et al., 2022; Marisca et al., 2022; Chen et al., 2024), which usually necessitate aligning the missing MTS at times like the aforementioned pre-alignment representation and focus on handling the data missing issues. However, our work emphasizes bypassing the canonical pre-alignment representation to address both the irregularity and asynchrony challenges within IMTS modeling.

3. Preliminary

3.1. Problem Definition

Definition 1 (Irregular Multivariate Time Series). *An IMTS can be represented as* $\mathcal{O} = \{\mathbf{o}^n_{1:L_n}\}_{n=1}^N = \{[(t_i^n, x_i^n)]_{i=1}^{L_n}\}_{n=1}^N$*, where there are* N *variables, the* n*-th variable contains* L_n *observations, and the* i*-th observation of* n*-th variable is composed of the recorded time* t_i^n *and value* x_i^n*.*

Definition 2 (Forecasting Query). *A forecasting query is represented as* q_j^n*, denoting* j*-th query on* n*-th variable to predict its corresponding value at a future time* q_j^n*.*

Problem 1 (Irregular Multivariate Time Series Forecasting). *Given historical IMTS observations* $\mathcal{O} = \{[(t_i^n, x_i^n)]_{i=1}^{L_n}\}_{n=1}^N$*, and a set of IMTS forecasting queries* $\mathcal{Q} = \{[q_j^n]_{j=1}^{Q_n}\}_{n=1}^N$*, the problem is to accurately forecast recorded values* $\hat{\mathcal{X}} = \{[\hat{x}_j^n]_{j=1}^{Q_n}\}_{n=1}^N$ *in correspondence to the forecasting queries:*

$$\mathcal{F}(\mathcal{O}, \mathcal{Q}) \longrightarrow \hat{\mathcal{X}}, \tag{1}$$

where $\mathcal{F}(\cdot)$ *denotes the forecasting model we aim to learn.*

3.2. Canonical Pre-Alignment Representation for IMTS

To facilitate IMTS modeling, a pre-alignment representation method (Che et al., 2018) has been widely adopted as the standard in current studies (Che et al., 2018; Shukla & Marlin, 2021; Zhang et al., 2021b; 2023a; Rubanova et al., 2019;

De Brouwer et al., 2019; Biloš et al., 2021; Schirmer et al., 2022). In this method, an IMTS $\mathcal{O}$ is represented by three matrix $(\mathcal{T}, \mathcal{X}, \mathcal{M})$. $\mathcal{T} = [t_l]_{l=1}^{L} = \cup_{n=1}^{N}[t_i^n]_{i=1}^{L_n} \in \mathbb{R}^L$ denotes the chronological unique timestamps of all observations within $\mathcal{O}$. $\mathcal{X} = [[\tilde{x}_l^n]_{n=1}^{N}]_{l=1}^{L} \in \mathbb{R}^{L\times N}$ are variable's values corresponding to the timestamps, where $\tilde{x}_l^n = x_i^n$ if the value of n-th variable is observed at time t_l, otherwise $\tilde{x}_l^n$ would be filled 'NA'. $\mathcal{M} = [[m_l^n]_{n=1}^{N}]_{l=1}^{L} \in \mathbb{R}^{L\times N}$ represents a masking matrix, where $m_l^n = 1$ if $\tilde{x}_l^n$ is observed at time t_l, otherwise zero.

We can observe that the sequence length L depends on the number of unique timestamps among $\mathcal{O}$. Let $L_{avg} = \frac{1}{N}\sum_{n=1}^{N} L_n$ and $L_{max} = \max[L_n]_{n=1}^{N}$ respectively denote the averaged and maximal number of observations for N variables in an IMTS, then the sequence length L after pre-aligned representation theoretically falls into:

$$L_{max} \leq \left|\cup_{n=1}^{N}[t_i^n]_{i=1}^{L_n}\right| \leq N \times L_{avg}, \tag{2}$$

which could be explosively growing proportional to the number of variables, thereby posing significant scalability concerns when dealing with large-scale variables.

4. Methodology

The overview of T-PATCHGNN is illustrated in Figure 2. In subsequent sections, we sequentially introduce the technical details of irregular time series patching, intra- and inter-time series modeling, and the IMTS forecasting process.

4.1. Irregular Time Series Patching

In this section, as a unified patching operation is applied to all univariate irregular time series, we take the n-th variable for illustration and omit the superscript n for simplicity in the presentation.

4.1.1. TRANSFORMABLE PATCHING

Time series patching has been demonstrated effective in MTS forecasting tasks due to its benefits in capturing local semantic information, reducing computation and memory usage, and modeling longer-range historical observations (Nie et al., 2022). The standard time series patching segments regular time series into a series of subseries-level patches, each of which consists of a fixed number of consecutive observations. However, in the context of IMTS, this approach will lead to patches spanning across diverse time horizons due to the varying time intervals between observations. For instance, a patch composed of five sequential observations might span merely a few minutes for densely sampled scenarios and could cover several days in cases of sparse sampling. This variability in the patch's temporal resolution can even exacerbate the inherent irregularity and asynchrony characteristics in IMTS modeling.

To address this problem, we propose to divide each univariate irregular time series $\mathbf{o}_{1:L}$ as a series of transformable patches $[\mathbf{o}_{l_p:r_p}]_{p=1}^{P}$ with variable-length consecutive observations, where P is the number of resulting patches, and $l_1 = 1, r_P = L$. Each transformable patch spans a patch window size s with a unified time horizon (*e.g.*, 2 hours) to guarantee a consistent temporal resolution across time and variables. The division can be overlapped or disjoint between two consecutive transformable patches. Along this line, the resulting patches of IMTS are aligned in a consistent time horizon resolution. As each univariate irregular time series is patched independently, this bypasses the canonical pre-alignment process on IMTS, preventing sequence length explosion from the increasing variable count.

4.1.2. PATCH ENCODING

After transforming each univariate irregular time series into a series of transformable patches, we encode each patch into a latent embedding to capture the local semantics within time series.

Continuous time embedding. To model the time information in IMTS, we first adopt a continuous time embedding (Shukla & Marlin, 2021) to encode the continuous time of observations:

$$\phi(t)[d] = \begin{cases} \omega_0 \cdot t + \alpha_0, & \text{if} \quad d = 0 \\ \sin\left(\omega_d \cdot t + \alpha_d\right), & \text{if} \quad 0 < d < D_t \end{cases}, \tag{3}$$

where the ω_d and α_d are learnable parameters and D_t is embedding's dimension. The linear term captures non-periodic patterns that evolve over time and the periodic terms capture periodicity among time series data, where ω_d and α_d represent the frequency and phase of the sine function.

By incorporating continuous time embedding via concatenation, we derive observations in the patch:

$$\mathbf{z}_{l_p:r_p} = [z_i]_{i=l_p}^{r_p} = [\phi(t_i)\|x_i]_{i=l_p}^{r_p}. \tag{4}$$

Transformable time-aware convolution. As each transformable patch is essentially a sub-irregular time series, we introduce the Transformable Time-aware Convolution Network (TTCN) (Zhang et al., 2023b) to capture the semantics within it. TTCN employs a meta-filter to derive the time-aware convolution filter, featuring adaptively generated parameters and transformable filter size that matches the input sequence's length, formulated as:

$$\mathbf{f}_d = \left[\frac{\exp(\mathbf{F}_d(z_i))}{\sum_{j=1}^{L_p}\exp(\mathbf{F}_d(z_j))}\right]_{i=1}^{L_p}, \tag{5}$$

where L_p is the sequence length of patch $\mathbf{z}_{l_p:r_p}$, $\mathbf{f}_d \in \mathbb{R}^{L_p\times D_{in}}$ is the derived filter for d-th feature map, D_{in} is dimension of inputs, and $\mathbf{F}_d$ denotes the meta-filter that can be

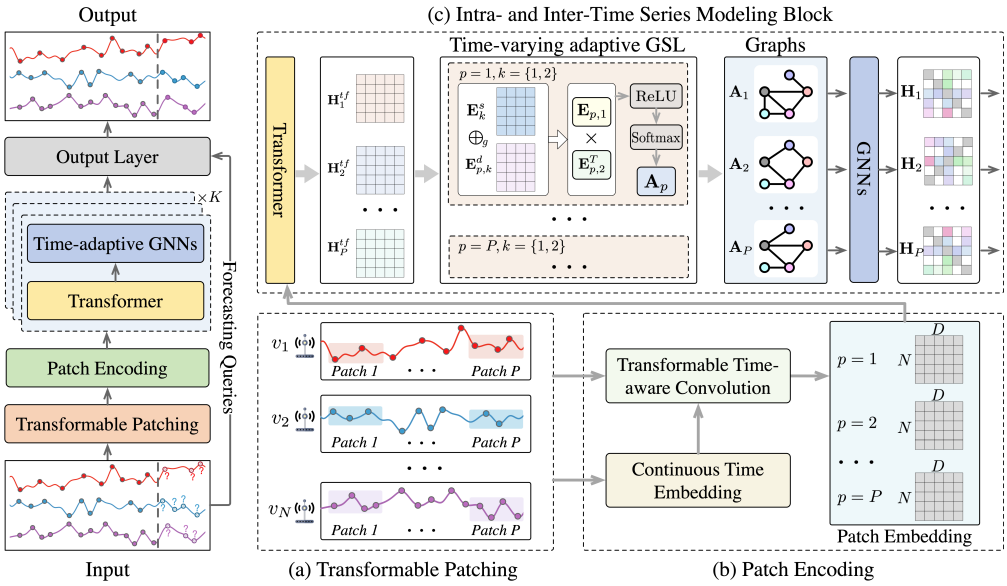


Figure 2: Overview of T-PATCHGNN, which initially divides each univariate irregular time series into a series of transformable patches with varying number of consecutive observations but maintains a unified time horizon resolution. Then the patching outcomes can be seamlessly modeled by Transformer and time-adaptive GNNs, which incorporate the time-varying adaptive graph structure learning (GSL), to realize an effective intra- and inter-time series modeling for IMTS. $\oplus_g$ represents a gated adding operation.

instantiated by learnable neural networks. By normalizing the derived filter parameters along the temporal dimension, TTCN ensures consistent scaling of the convolution results for sequences with varying lengths.

With $D-1$ filters derived based on Eq. (5), we attain the latent patch embedding $h_p^c \in \mathbb{R}^{D-1}$ through the following temporal convolution:

$$h_p^c = \left[\sum_{i=1}^{L_p} \mathbf{f}_d[i]^\top \mathbf{z}_{l_p:r_p}[i] \right]_{d=1}^{D-1}. \tag{6}$$

TTCN is applicable to encode transformable patches as it offers flexibility to adapt to variable-length sequences through transformable filters, customs parameterization for varying time intervals in irregular time series, and the ability to model arbitrarily long sequences without additional learnable filter parameters.

Considering that some patches may have no observations in the cases of sparse time series or high time horizon resolution, we additionally incorporate a patch masking term into the patch embedding:

$$h_p = [h_p^c \| m_p], \tag{7}$$

where m_p equals one if the patch has observations, otherwise zero, and we have $\mathbf{h}_{1:P} = [h_p]_{p=1}^P \in \mathbb{R}^{P\times D}$.

4.2. Intra- and Inter-Time Series Modeling

This section elaborates on how applying transformable patching to irregular time series can seamlessly facilitate both intra- and inter-time series modeling.

4.2.1. Transformer to Model Sequential Patches

With the patches encoded, they can be utilized as input tokens in a Transformer (Vaswani et al., 2017) to model the dependencies within the irregular time series. The position encodings $\mathbf{PE}_{1:P} \in \mathbb{R}^{P\times D}$ are added to indicate the temporal order of patches: $\mathbf{x}_{1:P}^{tf,n} = \mathbf{h}_{1:P}^n + \mathbf{PE}_{1:P}$. After that, the multi-head attention is applied by transforming them into query matrices $\mathbf{q}_h^n = \mathbf{x}_{1:P}^{tf,n}\mathbf{W}_h^Q$, key matrices $\mathbf{k}_h^n = \mathbf{x}_{1:P}^{tf,n}\mathbf{W}_h^K$ and value matrices $\mathbf{v}_h^n = \mathbf{x}_{1:P}^{tf,n}\mathbf{W}_h^V$, where $\mathbf{W}_h^Q, \mathbf{W}_h^K, \mathbf{W}_h^V \in \mathbb{R}^{D\times(D/H)}$ are learnable parameters, and H is the number of heads. A scaled dot-product attention is adopted to obtain the outputs of intra-time series modeling:

$$\mathbf{h}_{1:P}^{tf,n} = \|_{h=1}^H \text{Softmax}\left(\frac{\mathbf{q}_h^n \mathbf{k}_h^{nT}}{\sqrt{D/H}}\right)\mathbf{v}_h^n \in \mathbb{R}^{P\times D}. \tag{8}$$

4.2.2. Time-varying Adaptive Graph Structure Learning

Time series of different variables often exhibit substantial correlations. Insights from other variables can be highly in-

formative and significantly enhance the forecasting of each variable. For instance, there is a significant correlation between a patient's heart rate and blood pressure that changes in one can be indicative of changes in the other, reflecting the body's cardiovascular status (Obrist et al., 1978). However, observations within IMTS can be notably misaligned at times, raising obstacles for the inter-time series correlation modeling. Existing work (Zhang et al., 2021b) addresses this by propagating the asynchronous observations at all the timestamps when an observation appears at an arbitrary variable, which also involves the IMTS pre-alignment and may suffer from the sequence length explosion problem.

Fortunately, the asynchrony problem among IMTS can be seamlessly addressed after applying transformable patching to IMTS. Each variable has a consistent number of patches that are aligned to a uniform time horizon resolution. Along this line, we present time-adaptive graph neural networks to model inter-time series correlation within IMTS.

To shed light on the dynamic correlations underlying IMTS, we propose to learn a series of time-varying adaptive graphs, which keep the same temporal resolution as the patches. Specifically, inspired by studies (Wu et al., 2019; 2020b), we first maintain two embedding dictionaries with learnable parameters for all variables $\mathbf{E}_1^s, \mathbf{E}_2^s \in \mathbb{R}^{N \times D_g}$. This learns to capture the inherent characteristics of variables. While the above variable embedding can be updated during training, they will be static in the inference and remain invariable across all the periods in time series. However, the correlations between variables can dynamically change along with time (Zhang et al., 2021b). To address this, we incorporate patch embedding $\mathbf{H}_p^{tf} = [\mathbf{h}_p^{tf,n}]_{n=1}^N \in \mathbb{R}^{N \times D}$, which implies the time-varying semantics of time series at the patch-level temporal resolution, into the static variable embedding through a gated adding operation:

$$\begin{aligned} \mathbf{E}_{p,k} &= \mathbf{E}_k^s + g_{p,k} * \mathbf{E}_{p,k}^d, \\ \mathbf{E}_{p,k}^d &= \mathbf{H}_p^{tf} \mathbf{W}_k^d, \\ g_{p,k} &= \mathrm{ReLU}(\tanh([\mathbf{H}_p^{tf} \| \mathbf{E}_k^s] \mathbf{W}_k^g)), \\ k &= \{1, 2\}, \end{aligned} \tag{9}$$

where $\mathbf{W}_k^d \in \mathbb{R}^{D \times D_g}, \mathbf{W}_k^g \in \mathbb{R}^{(D+D_g) \times 1}$ are learnable parameters. In this way, we obtain the time-varying adaptive graph structure for each patch's time horizon to explicitly characterize the dynamic correlations underlying IMTS:

$$\mathbf{A}_p = \mathrm{Softmax}\left(\mathrm{ReLU}\left(\mathbf{E}_{p,1} \mathbf{E}_{p,2}^T\right)\right) \tag{10}$$

4.2.3. GNNs TO MODEL INTER-TIME SERIES CORRELATION

Based on the learned graph structures, we introduce GNNs (Kipf & Welling, 2016; Wu et al., 2020a; Zhou et al., 2020) to model the dynamic inter-time series correlations at a patch-level resolution:

$$\mathbf{H}_p = \mathrm{ReLU}\left(\sum_{m=0}^{M} (\mathbf{A}_p)^m \mathbf{H}_p^{tf} \mathbf{W}_m^{gnn}\right) \in \mathbb{R}^{N \times D}, \tag{11}$$

where M is the number of layers for GNNs, and $\mathbf{W}_m^{gnn} \in \mathbb{R}^{D \times D}$ are learnable parameters at m-th layer.

In practical usage, we can flexibly stack multiple K intra- and inter-time series modeling blocks to effectively address diverse IMTS modeling scenarios.

4.3. IMTS Forecasting

Subsequently, a flattened layer with a linear head is used to obtain the final latent representation for each variable:

$$\mathbf{H} = \mathrm{Flatten}([\mathbf{H}_p]_{p=1}^P) \mathbf{W}^f \in \mathbb{R}^{N \times D_o}, \tag{12}$$

where $\mathbf{W}^f \in \mathbb{R}^{PD \times D_o}$ are learnable parameters.

Given $\mathbf{H}^n \in \mathbf{H}$ for n-th variable and a set of forecasting queries $\{[q_j^n]_{j=1}^{Q_n}\}_{n=1}^N$, an MLP projection layer is used to generate the predicted results for these queries:

$$\hat{x}_j^n = \mathrm{MLP}([\mathbf{H}^n \| \phi(q_j^n)]). \tag{13}$$

The model is trained by minimizing the Mean Squared Error (MSE) loss between the prediction and the ground truth:

$$\mathcal{L} = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{Q_n} \sum_{j=1}^{Q_n} \left(\hat{x}_j^n - x_j^n\right)^2. \tag{14}$$

4.4. Analysis on Scalability

As the proposed transformable patching independently processes each univariate irregular time series to achieve alignment of IMTS, the average sequence length to be processed for IMTS equals to their average number of observations, *i.e.*, $L_{avg} = \frac{1}{N} \sum_{n=1}^{N} L_n$. Based on the analysis in Eq. (2), it is evident that by using transformable patching, the average sequence length to be processed, denote as L_{tp}, serves as a lower bound relative to the resulting average sequence length, L_{cpr}, derived by canonical pre-alignment representation:

$$L_{tp} = L_{avg} \leq L_{max} \leq L_{cpr} \leq N \times L_{avg}. \tag{15}$$

It prevents L_{tp} from explosively growing proportional to the number of variables, enhancing the model's scalability as the variable count increases. We also provide empirical evidence to analyze the scalability in Section 5.4 and Appendix A.2.

Table 1: Overall performance evaluated by MSE and MAE (mean ± std). The best-performing and second-best results are highlighted in **bold** and underline, respectively.

Algorithm	PhysioNet		MIMIC		Human Activity		USHCN	
	MSE×10^{-3}	MAE×10^{-2}	MSE×10^{-2}	MAE×10^{-2}	MSE×10^{-3}	MAE×10^{-2}	MSE×10^{-1}	MAE×10^{-1}
DLinear	41.86 ± 0.05	15.52 ± 0.03	4.90 ± 0.00	16.29 ± 0.05	4.03 ± 0.01	4.21 ± 0.01	6.21 ± 0.00	3.88 ± 0.02
TimesNet	16.48 ± 0.11	6.14 ± 0.03	5.88 ± 0.08	13.62 ± 0.07	3.12 ± 0.01	3.56 ± 0.02	5.58 ± 0.05	3.60 ± 0.04
PatchTST	12.00 ± 0.23	6.02 ± 0.14	3.78 ± 0.03	12.43 ± 0.10	4.29 ± 0.14	4.80 ± 0.09	5.75 ± 0.01	3.57 ± 0.02
Crossformer	6.66 ± 0.11	4.81 ± 0.11	2.65 ± 0.10	9.56 ± 0.29	4.29 ± 0.20	4.89 ± 0.17	5.25 ± 0.04	3.27 ± 0.09
Graph Wavenet	6.04 ± 0.28	4.41 ± 0.11	2.93 ± 0.09	10.50 ± 0.15	2.89 ± 0.03	3.40 ± 0.05	5.29 ± 0.04	3.16 ± 0.09
MTGNN	6.26 ± 0.18	4.46 ± 0.07	2.71 ± 0.23	9.55 ± 0.65	3.03 ± 0.03	3.53 ± 0.03	5.39 ± 0.05	3.34 ± 0.02
StemGNN	6.86 ± 0.28	4.76 ± 0.19	1.73 ± 0.02	7.71 ± 0.11	8.81 ± 0.37	6.90 ± 0.02	5.75 ± 0.09	3.40 ± 0.09
CrossGNN	7.22 ± 0.36	4.96 ± 0.12	2.95 ± 0.16	10.82 ± 0.21	3.03 ± 0.10	3.48 ± 0.08	5.66 ± 0.04	3.53 ± 0.05
FourierGNN	6.84 ± 0.35	4.65 ± 0.12	2.55 ± 0.03	10.22 ± 0.08	2.99 ± 0.02	3.42 ± 0.02	5.82 ± 0.06	3.62 ± 0.07
GRU-D	5.59 ± 0.09	4.08 ± 0.05	1.76 ± 0.03	7.53 ± 0.09	2.94 ± 0.05	3.53 ± 0.06	5.54 ± 0.38	3.40 ± 0.28
SeFT	9.22 ± 0.18	5.40 ± 0.08	1.87 ± 0.01	7.84 ± 0.08	12.20 ± 0.17	8.43 ± 0.07	5.80 ± 0.19	3.70 ± 0.11
RainDrop	9.82 ± 0.08	5.57 ± 0.06	1.99 ± 0.03	8.27 ± 0.07	14.92 ± 0.14	9.45 ± 0.05	5.78 ± 0.22	3.67 ± 0.17
Warpformer	5.94 ± 0.35	4.21 ± 0.12	1.73 ± 0.04	7.58 ± 0.13	2.79 ± 0.04	3.39 ± 0.03	5.25 ± 0.05	3.23 ± 0.05
mTAND	6.23 ± 0.24	4.51 ± 0.17	1.85 ± 0.06	7.73 ± 0.13	3.22 ± 0.07	3.81 ± 0.07	5.33 ± 0.05	3.26 ± 0.10
Latent-ODE	6.05 ± 0.57	4.23 ± 0.26	1.89 ± 0.19	8.11 ± 0.52	3.34 ± 0.11	3.94 ± 0.12	5.62 ± 0.03	3.60 ± 0.12
CRU	8.56 ± 0.26	5.16 ± 0.09	1.97 ± 0.02	7.93 ± 0.19	6.97 ± 0.78	6.30 ± 0.47	6.09 ± 0.17	3.54 ± 0.18
Neural Flow	7.20 ± 0.07	4.67 ± 0.04	1.87 ± 0.05	8.03 ± 0.19	4.05 ± 0.13	4.46 ± 0.09	5.35 ± 0.05	3.25 ± 0.05
T-PATCHGNN	**4.98 ± 0.08**	**3.72 ± 0.03**	**1.69 ± 0.03**	**7.22 ± 0.09**	**2.66 ± 0.03**	**3.15 ± 0.02**	**5.00 ± 0.04**	**3.08 ± 0.04**

5. Experiments

5.1. Experimental Setup

5.1.1. Datasets

We involve four datasets, including PhysioNet, MIMIC, Human Activity, and USHCN, across diverse subject areas, such as healthcare, biomechanics, and climate science, to comprehensively evaluate models' performance on IMTS forecasting tasks. Consistently, we randomly divide all the instances among each dataset into training, validation, and test sets adhering to ratios of 60%, 20%, and 20%. Please refer to Appendix Section A.5 for details of these datasets.

5.1.2. Implementation Details

All experiments are performed on a Linux server with 20-core Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50GHz and NVIDIA Tesla V100 GPU. To ensure a fair comparison, for all compared models, we consistently set the hidden dimensions to 64 for PhysioNet and MIMIC, and 32 for Human Activity and USHCN. The batch size is chosen as 192 for USHCN and 32 for the other datasets. We employ Adam optimizer for these models' training and apply early stopping when the validation loss doesn't decrease over 10 epochs. To mitigate randomness, we perform each experiment using five different random seeds and present the mean and standard deviation of the results.

For detailed setups of T-PATCHGNN, we chose the patch window size s as 8 hours for PhysioNet and MIMIC, 300 milliseconds for Human Activity, and 2 months for USHCN. To reduce the number of resulting patches, we do not make the patch segmenting overlap and maintain a sliding stride for the patch window equal to its size. The dimension of time embedding D_t and variable embedding D_g is set to 10. The number of heads H in Transformer, layers M in GNNs, and the number of block K is selected as 1. We adopt three layers MLPs to instantiate the meta-filters in TTCN and the output projection layer. We set the learning rate to 0.001 for the entire model training.

5.1.3. Evaluation Metrics

Current IMTS forecasting studies predominantly utilize Mean Square Error (MSE) for the evaluation, which tends to be sensitively affected by outliers and is hard to interpret (Chai & Draxler, 2014). To offer a more comprehensive assessment of model performance, we also incorporate Mean Absolute Error (MAE), a metric extensively employed in evaluations for classical time series forecasting (Lim & Zohren, 2021; Fan et al., 2023). These two metrics are formally defined as follows: $\text{MSE} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{Q_n}\sum_{j=1}^{Q_n}\left(\hat{x}_j^n - x_j^n\right)^2$, $\text{MAE} = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{Q_n}\sum_{j=1}^{Q_n}\left|\hat{x}_j^n - x_j^n\right|$.

5.1.4. Baselines

To establish a thorough benchmark for the under-explored IMTS forecasting task, we incorporate seventeen relevant baselines for a fair comparison, covering the SOTA models from (1) MTS forecasting: DLinear (Zeng et al., 2023), TimesNet (Wu et al., 2022), PatchTST (Nie et al., 2022), Crossformer (Zhang & Yan, 2022), Graphwavenet (Wu et al., 2019), MTGNN (Wu et al., 2020b), StemGNN (Cao et al., 2020), CrossGNN (Huang et al., 2023) and FourierGNN (Yi et al., 2023), (2) IMTS classification: GRU-D (Che et al., 2018), SeFT (Horn et al., 2020), RainDrop (Zhang et al., 2021b), Warpformer (Zhang et al., 2023a), (3) IMTS interpolation: mTAND (Shukla & Marlin, 2021), and (4) IMTS forecasting: Latent ODEs (Rubanova et al., 2019),

Table 2: Ablation results of T-PATCHGNN on four datasets.

Ablation	PhysioNet		MIMIC		Human Activity		USHCN	
	MSE$\times 10^{-3}$	MAE$\times 10^{-2}$	MSE$\times 10^{-2}$	MAE$\times 10^{-2}$	MSE$\times 10^{-3}$	MAE$\times 10^{-2}$	MSE$\times 10^{-1}$	MAE$\times 10^{-1}$
Complete	$\mathbf{4.98 \pm 0.08}$	$\mathbf{3.72 \pm 0.03}$	$\mathbf{1.69 \pm 0.03}$	$\mathbf{7.22 \pm 0.09}$	$\mathbf{2.66 \pm 0.03}$	$\mathbf{3.15 \pm 0.02}$	$\mathbf{5.00 \pm 0.04}$	$\mathbf{3.08 \pm 0.04}$
w/o Patch	5.27 ± 0.06	3.88 ± 0.03	1.73 ± 0.02	7.41 ± 0.06	2.85 ± 0.05	3.28 ± 0.02	5.32 ± 0.02	3.30 ± 0.09
rp Patch	5.64 ± 0.13	3.97 ± 0.05	1.72 ± 0.01	7.29 ± 0.02	2.82 ± 0.02	3.29 ± 0.04	5.20 ± 0.07	3.17 ± 0.07
w/o VE	10.53 ± 0.15	5.30 ± 0.01	3.71 ± 0.02	12.06 ± 0.03	2.79 ± 0.03	3.18 ± 0.03	5.08 ± 0.03	3.25 ± 0.08
w/o PE	5.10 ± 0.02	3.83 ± 0.06	1.71 ± 0.01	7.42 ± 0.11	2.79 ± 0.02	3.21 ± 0.04	5.08 ± 0.02	3.22 ± 0.08
w/o Transformer	5.10 ± 0.12	3.83 ± 0.07	1.71 ± 0.02	7.27 ± 0.06	2.78 ± 0.04	3.20 ± 0.04	5.22 ± 0.06	3.19 ± 0.11

CRU (Schirmer et al., 2022), Neural Flows (Biloš et al., 2021). The details of these baseliens are provided in Appendix Section A.6.

5.2. Main Results

Table 1 reports the models' forecasting performance evaluated by MSE and MAE on four datasets. As can be seen, T-PATCHGNN archives the consistently best performance on all datasets and even outperforms the second-best baseline over 10%. Besides, we observe the MTS forecasting models, including patching-based models and GNN-based models, do not attain consistently competitive performance on IMTS forecasting. It indicates that straightforwardly applying these two techniques to IMTS fails to effectively handle the challenging intra- and inter-time series modeling. Moreover, the existing IMTS forecasting models do not achieve satisfactory performance, probably because they fail to effectively model inter-time series correlations to enhance forecasting performance. We also test these models' performance on longer and shorter forecasting windows, where the results are provided in Appendix Section A.1.

5.3. Ablation Study

We evaluate the performance of T-PATCHGNN and its several variants on four datasets. (1) **Complete** represents the model without any ablation; (2) **w/o Patch** removes transformable patching and adopts the canonical pre-alignment representation; (3) **rp Patch** replaces transformable patching with standard time series patching (Nie et al., 2022); (4) **w/o VE** removes the variable embedding in Eq. (9) when constructing adaptive graph; (5) **w/o PE** removes the patch embedding when constructing adaptive graph; (6) **w/o Transformer** removes the Transformer module in the model.

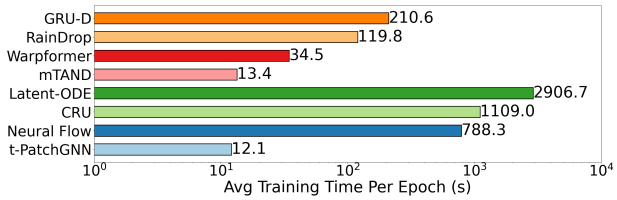
Table 2 shows the results of model ablation. As can be seen, removing any component can lead to a performance descent compared to the complete model. From these results, we observe **w/o Patch** causes notable performance degradation for all datasets, which proves that patching irregular time series can indeed facilitate the subsequent intra- and inter-time series modeling for IMTS. However, directly using standard time series patching can even lead to worse performance than **w/o Patch** in some datasets like PhysioNet. It verifies our claims that the standard patching faces troubles with the variability in the patch's temporal resolution, which may even exacerbate the inherent irregularity and asynchrony characteristics in IMTS modeling. Comparing the results of **w/o VE** and **w/o PE**, we can find that variable's inherent characteristic is more important than its dynamic patterns to characterize the inter-time series correlations for physiological signals forecasting tasks (PhysioNet and MIMIC). This makes sense because there is a remarkable semantic discrepancy between these signals, it is difficult to accurately characterize their interrelation without effectively identifying them. However, we observe that the dynamic patch embedding plays a significant role on human motion and climate forecasting, which indicates the variables' correlation in these tasks can usually dynamically vary along with time. For instance, temperature decreases in winter often lead to increased snow cover, but this correlation does not necessarily apply in summer.

Table 3: Evidence of sequence length explosion. Aligned length represents the sequence length after canonical pre-alignment. Amplification indicates the multiple of growth by comparing the aligned length of IMTS to the original number of observations.

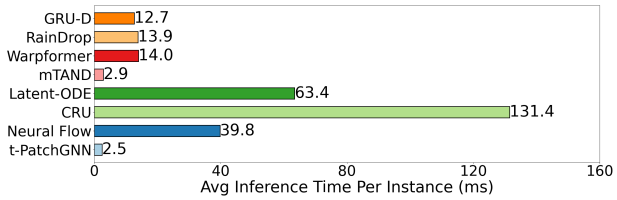
Description	PhysioNet	MIMIC	Human Activity	USHCN
# Variable	41	96	12	5
Avg # observations	10.7	1.5	30.2	36.1
Avg aligned length	75	46.1	120.6	163.9
Max aligned length	216	643	130	214
Avg amplification	$\times 7.0$	$\times 21.9$	$\times 4.0$	$\times 4.5$
Max amplification	$\times 16.8$	$\times 96.0$	$\times 4.0$	$\times 5.0$

5.4. Scalability and Efficiency Analysis

Table 3 showcases the extent of the sequence length explosion issue following canonical pre-alignment representation across four datasets. It is evident that, on average, sequence lengths can expand more than 20-fold from the original number of observations, particularly when dealing with a larger number of variables. In extreme cases, the sequence length may increase explosively proportional to the number of variables (revealed by max amplification), posing significant scalability challenges. However, our transformable patching effectively circumvents this issue by processing the original

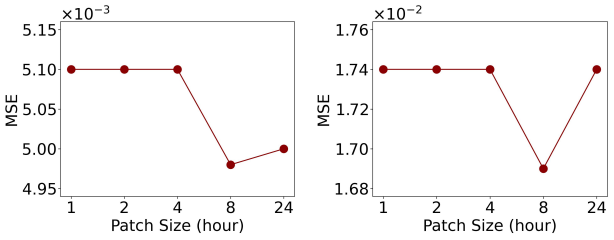


(a) Average training time per epoch

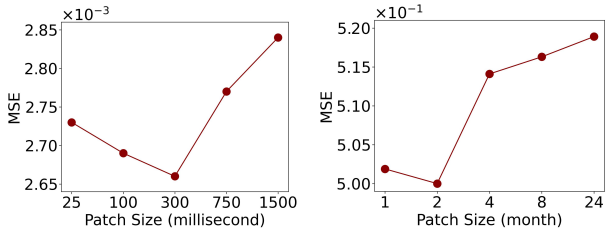


(b) Average inference time per instance

Figure 3: Efficiency comparison of training and inference.



(a) PhysioNet (b) MIMIC



(c) Human Activity (d) USHCN

Figure 4: Effect of different patch sizes.

observation sequences without requiring pre-alignment.

To further study the benefits of transformable patching on the model's efficiency, we present the average training time per epoch and average inference time per IMTS instance on MIMIC in Figure 3. We can observe T-PATCHGNN outperforms all models that employ canonical pre-alignment representation in terms of efficiency during both training and inference phases. Furthermore, when compared to current predominant ODE-based IMTS forecasting models, T-PATCHGNN even achieves at least 65 times faster training speeds and 15 times quicker inference speeds. More analytical testing on models' scalability with increased variables is provided in Appendix A.2.

5.5. Effect of Patch Size

Figure 4 depicts the effect of different patch window sizes on various datasets. We can observe the impact of patch size on performance varies across datasets from different areas. Specifically, for PhysioNet and MIMIC, performance remains relatively stable with smaller patch sizes and peaks when the patch size reaches 8 hours. This could be attributed to the sparse nature of many physiological signals, where a timespan shorter than four hours may not encompass sufficient observations to capture local patterns effectively within sub-series. However, as patch size increases beyond this point, we observe a decline in model performance. An excessively large patch size results in a reduced patch-level temporal resolution, adversely affecting the detailed intra- and inter-time series analysis. When it comes to Human Activity and USHCN, a relatively small patch size would be preferred. As the IMTS from these areas usually exhibit highly dynamic patterns, a relatively small patch size can enable finer-grained modeling of dynamics within IMTS.

From another perspective, the optimal patch size can be selected by comprehensively considering the forecasting and observed window sizes. Long-range forecasting and observation usually involve a larger patch size to better capture the trend semantics within patches and long-range dependencies across time series (*e.g.*, PhysioNet and MIMIC), whereas short-range forecasting (*e.g.*, Human Activity and USHCN) are more recommended to choose a relatively smaller patch size for finer-grained resolution modeling. The sensitivity analysis on more hyper-parameters is provided in Appendix A.3.

6. Conclusion

This paper presented a Transformable Patching Graph Neural Networks approach, T-PATCHGNN, to address the IMTS forecasting problem. T-PATCHGNN achieved the alignment between asynchronous IMTS by transforming each univariate irregular time series into a series of transformable patches with varying observation counts but maintaining unified time horizon resolution. This transformation enabled the capture of local semantics within IMTS and seamlessly facilitated intra- and inter-time series modeling without a canonical pre-alignment representation process, preventing the aligned sequence length from explosively growing proportional to the increasing variables. Building on transformable patching, we then presented the time-adaptive graph neural networks to model dynamic inter-time series correlations based on a series of learned time-varying adaptive graphs. We demonstrated the remarkable superiority of T-PATCHGNN on a comprehensive IMTS forecasting benchmark we build.

Acknowledgements

This work was partially supported by the National Key Research and Development Program of China (No.2023YFF0725001), National Natural Science Foundation of China (Grant No.92370204, No.62102110), Guangzhou-HKUST(GZ) Joint Funding Program (Grant No.2023A03J0008), Education Bureau of Guangzhou Municipality, Guangdong Science and Technology Department, and CCF-Baidu Open Fund.

Impact Statement

This paper details efforts to advance time series analysis and its applications across various scientific domains. While our research may lead to many societal impacts, we do not find it necessary to single out any particular consequences for emphasis here.

References

A. Johnson, T. Pollard, L. S. H. L. L.-W. M. F. M. G. B. M. P. S. L. A. C. and Mark, R. G. Mimic-iii, a freely accessible critical care database. *Scientific data, 3(1):1–9*, 2016.

Baytas, I. M., Xiao, C., Zhang, X., Wang, F., Jain, A. K., and Zhou, J. Patient subtyping via time-aware lstm networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 65–74, 2017.

Biloš, M., Sommer, J., Rangapuram, S. S., Januschowski, T., and Günnemann, S. Neural flows: Efficient alternative to neural odes. *Advances in neural information processing systems*, 34:21325–21337, 2021.

Cao, D., Wang, Y., Duan, J., Zhang, C., Zhu, X., Huang, C., Tong, Y., Xu, B., Bai, J., Tong, J., et al. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in neural information processing systems*, 33:17766–17778, 2020.

Chai, T. and Draxler, R. R. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, pp. 1247–1250, 2014.

Che, Z., Purushotham, S., Cho, K., Sontag, D., and Liu, Y. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085, 2018.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Chen, X., Li, X., Liu, B., and Li, Z. Biased temporal convolution graph network for time series forecasting with missing values. In *International Conference on Learning Representations*, 2024.

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

De Brouwer, E., Simm, J., Arany, A., and Moreau, Y. Gru-ode-bayes: Continuous modeling of sporadically-observed time series. *Advances in neural information processing systems*, 32, 2019.

Engle, R. F. and Russell, J. R. Autoregressive conditional duration: a new model for irregularly spaced transaction data. *Econometrica*, pp. 1127–1162, 1998.

Fan, W., Wang, P., Wang, D., Wang, D., Zhou, Y., and Fu, Y. Dish-ts: a general paradigm for alleviating distribution shift in time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7522–7529, 2023.

Horn, M., Moor, M., Bock, C., Rieck, B., and Borgwardt, K. Set functions for time series. In *International Conference on Machine Learning*, pp. 4353–4363. PMLR, 2020.

Huang, Q., Shen, L., Zhang, R., Ding, S., Wang, B., Zhou, Z., and Wang, Y. Crossgnn: Confronting noisy multivariate time series via cross interaction refinement. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Ikaro Silva, George Moody, D. S. L. C. and Mark, R. Predicting in-hospital mortality of icu patients: The physionet computing in cardiology challenge 2012. *Computing in cardiology, 39:245–248*, 2012.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2016.

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

Li, Z., Li, S., and Yan, X. Time series as images: Vision transformer for irregularly sampled time series. *arXiv preprint arXiv:2303.12799*, 2023.

Lim, B. and Zohren, S. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

Liu, F., Liu, H., and Jiang, W. Practical adversarial attacks on spatiotemporal traffic forecasting models. *Advances in Neural Information Processing Systems*, 35:19035–19047, 2022.

Marisca, I., Cini, A., and Alippi, C. Learning to reconstruct missing data from spatiotemporal graphs with sparse observations. *Advances in Neural Information Processing Systems*, 35:32069–32082, 2022.

Menne, M., W. J. C. and Vose, R. Long-term daily climate records from stations across the contiguous united states.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2022.

Obrist, P. A., Gaebelein, C. J., Teller, E. S., Langer, A. W., Grignolo, A., Light, K. C., and McCubbin, J. A. The relationship among heart rate, carotid dp/dt, and blood pressure in humans as a function of the type of stress. *Psychophysiology*, pp. 102–115, 1978.

Rubanova, Y., Chen, R. T., and Duvenaud, D. K. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.

Schirmer, M., Eltayeb, M., Lessmann, S., and Rudolph, M. Modeling irregular time series with continuous recurrent units. In *International Conference on Machine Learning*, pp. 19388–19405. PMLR, 2022.

Shukla, S. N. and Marlin, B. Interpolation-prediction networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2018.

Shukla, S. N. and Marlin, B. Multi-time attention networks for irregularly sampled time series. In *International Conference on Learning Representations*, 2021.

Shukla, S. N. and Marlin, B. M. A survey on principles, models and methods for learning from irregularly sampled time series. *arXiv preprint arXiv:2012.00168*, 2020.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Vio, R., Diaz-Trigo, M., and Andreani, P. Irregular time series in astronomy and the use of the lomb–scargle periodogram. *Astronomy and Computing*, 1:5–16, 2013.

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2022.

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

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020a.

Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., and Zhang, C. 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*, pp. 753–763, 2020b.

Yao, Z.-J., Bi, J., and Chen, Y.-X. Applying deep learning to individual and community health monitoring data: A survey. *Machine Intelligence Research*, 15:643–655, 2018.

Yi, K., Zhang, Q., Fan, W., He, H., Hu, L., Wang, P., An, N., Cao, L., and Niu, Z. Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

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

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Zhang, J., Zheng, S., Cao, W., Bian, J., and Li, J. 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*, pp. 3273–3285, 2023a.

Zhang, W., Liu, H., Zha, L., Zhu, H., Liu, J., Dou, D., and Xiong, H. Mugrep: A multi-task hierarchical graph representation learning framework for real estate appraisal. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 3937–3947, 2021a.

Zhang, W., Zhang, L., Han, J., Liu, H., Zhou, J., Mei, Y., and Xiong, H. Irregular traffic time series forecasting based on asynchronous spatio-temporal graph convolutional network. *arXiv preprint arXiv:2308.16818*, 2023b.

Zhang, X., Zeman, M., Tsiligkaridis, T., and Zitnik, M. Graph-guided network for irregularly sampled multivariate time series. In *International Conference on Learning Representations*, 2021b.

Zhang, Y. and Yan, J. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2022.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.

Table 4: Performance of varying observation and forecast horizons.

Algorithm	History=3h, Forecast=45h		History=12h, Forecast=36h		History=36h, Forecast=12h		History=45h, Forecast=3h	
	MSE$\times 10^{-3}$	MAE$\times 10^{-2}$	MSE$\times 10^{-3}$	MAE$\times 10^{-2}$	MSE$\times 10^{-3}$	MAE$\times 10^{-2}$	MSE$\times 10^{-3}$	MAE$\times 10^{-2}$
DLinear	51.82	17.13	43.56	15.73	41.63	15.48	41.23	15.51
TimesNet	57.30	10.70	24.95	7.62	13.57	5.50	13.86	5.65
PatchTST	42.18	13.67	18.56	7.80	9.85	5.11	8.53	4.64
Crossformer	9.48	5.86	8.57	5.70	5.70	4.47	5.33	4.44
Graph Wavenet	9.43	5.86	7.23	4.82	4.71	3.90	<u>4.10</u>	<u>3.73</u>
MTGNN	9.83	5.95	7.48	5.01	5.08	3.99	5.22	4.19
StemGNN	8.70	5.37	7.46	4.84	6.65	4.69	5.47	4.56
CrossGNN	10.44	6.56	7.97	5.37	6.87	4.73	4.80	4.24
FourierGNN	9.59	5.61	7.95	4.99	6.35	4.61	5.37	4.34
GRU-D	<u>8.18</u>	<u>4.99</u>	<u>6.89</u>	<u>4.55</u>	<u>4.42</u>	<u>3.66</u>	4.44	3.79
SeFT	9.78	5.55	9.30	5.41	9.15	5.15	8.76	5.57
RainDrop	10.47	5.72	9.89	5.62	9.70	5.40	9.28	5.62
Warpformer	8.48	5.13	7.57	4.83	5.60	4.09	6.44	4.67
mTAND	8.45	5.23	7.11	4.67	5.71	4.17	5.44	4.33
Latent-ODE	8.25	5.04	7.20	4.69	6.70	4.36	7.10	5.33
CRU	9.20	5.38	9.20	5.31	9.50	5.41	11.60	6.98
Neural Flow	8.30	<u>4.99</u>	8.50	5.27	7.70	4.68	7.40	5.10
T-PATCHGNN	**8.01**	**4.87**	**6.48**	**4.19**	**4.14**	**3.31**	**3.69**	**3.25**

Table 5: Training and inference time with increased variables.

Algorithm \ # Variable	100		500		1000		2000	
	Training(s)	Inference(ms)	Training(s)	Inference(ms)	Training(s)	Inference(ms)	Training(s)	Inference(ms)
GRU-D	870.79	469.43	4178.09	2228.06	8839.45	4819.39	>10000	12073.38
Warpformer	138.66	208.31	out of memory	out of memory	out of memory	out of memory	out of memory	out of memory
mTAND	6.32	4.15	22.71	30.92	78.92	126.40	out of memory	out of memory
T-PATCHGNN	5.05	2.30	5.65	2.45	6.53	3.08	7.45	3.19

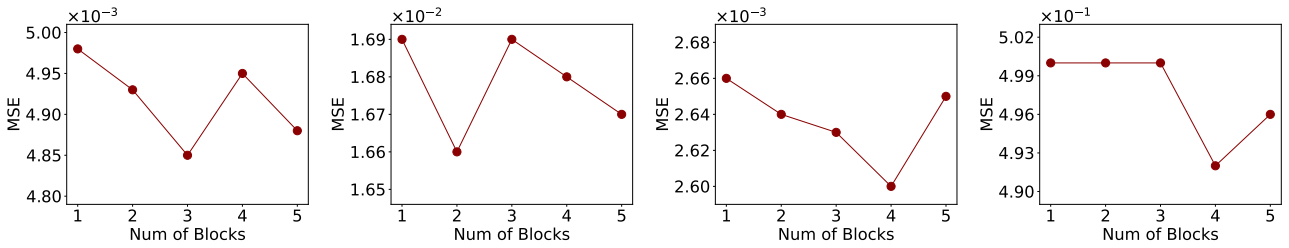
A. Additional Experiment

A.1. Varying Observation and Forecast Horizons

Table 4 presents the model's performance on longer- (forecast next 36 / 45 hours using historical 12 / 3 hours) and shorter-horizon (forecast next 12 / 3 hours using historical 36 / 45 hours) forecasting on PhysioNet. We can observe T-PATCHGNN achieves the consistently best performance for different forecasting horizons. Moreover, our model showcases larger superiority over baselines to process longer historical windows (*e.g.*, 24h, 36h, and 45h), which is probably attributed to the transformable patching to facilitate long-range time series dependencies modeling. Furthermore, it shows that the performance of these algorithms tends to be closer when the historical observed window becomes very short (*e.g.*, 3h). This may be because a shorter historical window contains less semantics and is thus easier to capture by different models. While most models perform better when forecasting horizon window reduction, the ODE-based models yet perform worse when using a longer history to predict shorter horizons. This may be because the too-long sequence and less labeled data degrade the performance of this type of method.

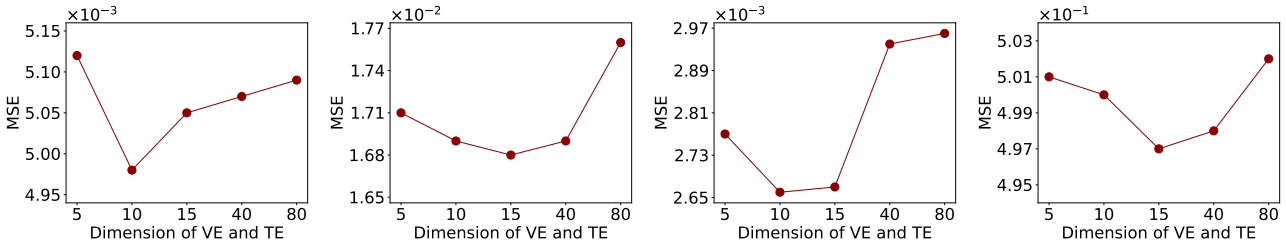
A.2. Model Scalability with Increased Variables

To further analyze the impact of pre-alignment representation on scalability, we created a synthetic dataset designed to flexibly test the influence of increased variables and report the average training time per epoch and inference time per instance. In this test, we generated 1,000 IMTS instances with multiple variables. Each variable comprises an average of 10 observations, randomly distributed at different timestamps (in seconds) within a day. To conserve memory and enable testing with larger variables, we set the batch size to 1 and restricted the hidden dimension to 2. As illustrated in Table 5, we observe that the sequence length explosion problem deteriorates as the number of variables increases. This leads to pronounced scalability issues affecting both computational efficiency and memory usage for the pre-alignment methods, *i.e.*, GRU-D, Warpformer and mTAND. However, our T-PATCHGNN with transformable patching effectively mitigates this problem, maintaining high training and inference efficiency despite the increased variables.



(a) PhysioNet (b) MIMIC (c) Human Activity (d) USHCN

Figure 5: Effect of different block K.



(a) PhysioNet (b) MIMIC (c) Human Activity (d) USHCN

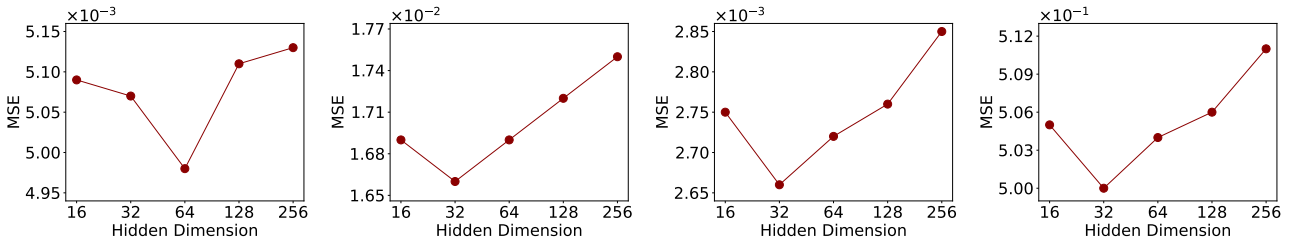
Figure 6: Effect of different variable embedding (VE)&time embedding (TE) dimensions D_t&D_g.

A.3. Parameter Sensitivity

Figure 5 displays the effect of different intra- and inter-time series modeling blocks. We observe stacking multiple K blocks always has the potential to achieve better performance. However, it also costs more expensive computational overheads. Therefore, we choose $K = 1$ for the major experiments.

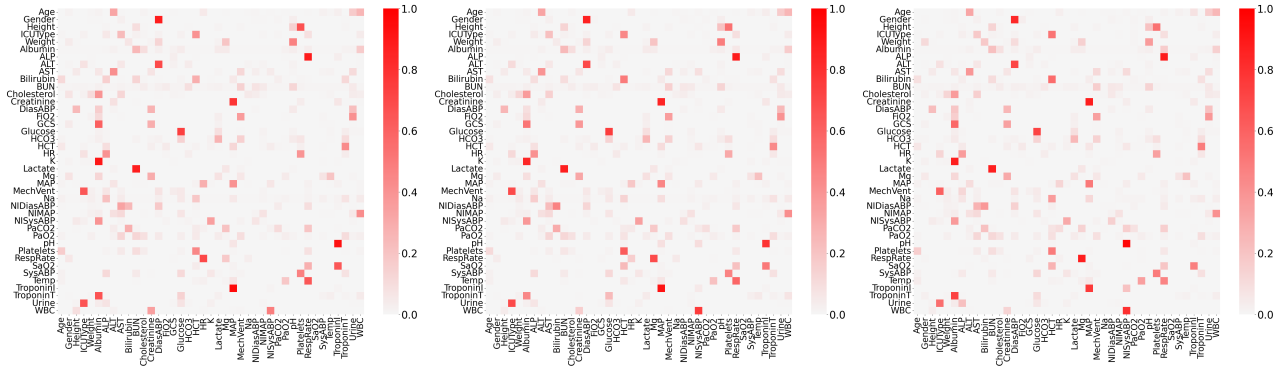
Figure 6 shows the effect of dimensions D_t&D_g of variable&time embeddings. It indicates relatively small sizes (*e.g.*, 10 or 15) usually perform better. A too-large size may lead to performance collapse due to the potential data sparsity issues for some variables to learn semantic embedding.

Figure 7 reports the effect of different hidden dimension D. We find setting the hidden dimension to 32 for smaller datasets (*e.g.*, Human Activity and USHCN) and 64 for larger datasets (*e.g.*, PhysioNet) would be a good choice. However, this is not absolute. While MIMIC is a large-scale dataset, its best setting is 32 due to a notable sparsity in its measurements. The choice of hidden dimension should comprehensively consider the number of training data and the sparsity of measurements.



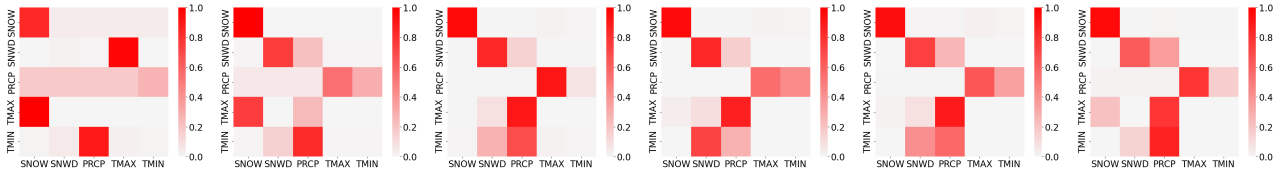
(a) PhysioNet (b) MIMIC (c) Human Activity (d) USHCN

Figure 7: Effect of different hidden dimension D.

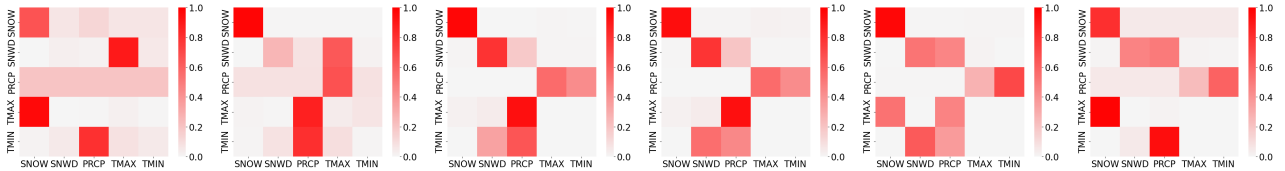


(a) Patch 1 (0-7h) (b) Patch 2 (8-15h) (c) Patch 3 (16-23h)

Figure 8: Adjacent matrices of time-adaptive graphs learned from PhysioNet.



(a) Patch 1 (Jan, Feb) (b) Patch 2 (Mar, Apr) (c) Patch 3 (May, Jun) (d) Patch 4 (Jul, Aug) (e) Patch 5 (Sep, Oct) (f) Patch 6 (Nov, Dec)



(g) Patch 7 (Jan, Feb) (h) Patch 8 (Mar, Apr) (i) Patch 9 (May, Jun) (j) Patch 10 (Jul, Aug) (k) Patch 11 (Sep, Oct) (l) Patch 12 (Nov, Dec)

Figure 9: Adjacent matrices of time-adaptive graphs learned from USHCN.

A.4. Visualization on Learned Adaptive Graphs

Figure 8 and Figure 9 provide visualizations on the learned adjacent matrices of adaptive graph structures to analyze how they work in different contexts. Overall, we find the learned adjacent matrices are usually sparse, which implies our model attempts to learn the real correlations from data instead of simply aggregating these variables. Moreover, we observe remarked and insightful time-varying correlations learned from dynamic contexts (*e.g.*, USHCN), further underscoring the necessity of learning time-adaptive graph structures.

For PhysioNet, as illustrated in Figure 8, we observe our model can learn insightful correlations between different indicator variables of patients. For example, the adjacent matrix indicates that heart rate (HR) and respiratory rate (RespRate) are highly correlated because they usually simultaneously increase during physical activity or stress to meet the body's higher demand for oxygen. A high correlation is also displayed between RespRate and body temperature (Temp) as they usually increase together in many situations like when the body is fighting an infection. In addition, some underlying and more complex correlations may be automatically discovered from data through the graph structure learning process. For instance, it indicates that there is a high correlation between blood urea nitrogen (BUN) and Lactate levels. Typically, BUN levels reflect renal function. Impaired renal function can lead to reduced clearance of both urea and lactate, and thus cause lactate to accumulate.

The cases from USHCN are illustrated in Figure 9, where the learned graph structures exhibit pronounced seasonal variation.

For instance, in winter (Patch 1 (Jan, Feb)), snowfall (SNOW) markedly influences maximum temperature forecasts (TMAX). This effect gradually wanes from winter to summer, correlating with the reduction or absence of snowfall. As seasons cycle from summer back to winter, this influence progressively strengthens. A similar trend is also showcased between TMAX and snow depth (SNWD). Furthermore, we discover that these correlations exhibit cyclical changes on an annual scale. This highlights our model's ability to learn the temporal dynamics of variable correlations within data.

A.5. Description on Datasets

PhysioNet[2] (Ikaro Silva & Mark, 2012) contains 12,000 IMTS corresponding to different patients, where each consists of a total of 41 clinical signal variables irregularly collected during the initial 48 hours following the patient's admission to the ICU. For each IMTS, we use the first 24 hours as the observed data to predict the queried values in the next 24 hours.

MIMIC[3] (A. Johnson & Mark, 2016) is a widely accessible clinical database that houses electronic health records of patients in critical care. Following the pre-processing provided by (Biloš et al., 2021), we obtain 23,457 patients' IMTS collected from the first 48 hours after the patient's admission, and each with 96 variables. Similar to the PhysioNet, we utilize the initial 24-hour period as the observed data to forecast the target values for the subsequent 24-hour time frame.

Human Activity[4] comprises 12 variables consisting of irregularly measured 3D positional records of four different sensors worn in the human left ankle, right ankle, belt, and chest. The dataset is gathered from five individuals executing a diverse range of activities such as walking, sitting, lying down, standing, and others. To better align with the requirements of realistic forecasting scenarios, we chunk the original time series to obtain a total of 5,400 IMTS, each of which contains 4,000 milliseconds span, and we leverage the first 3,000 milliseconds as the observed data to predict the positional value of sensors in the next 1,000 milliseconds.

Given that data missing is a frequent occurrence in climate research possibly due to sensor malfunctions, measurement errors, or data acquisition issues, we follow the previous works (De Brouwer et al., 2019; Schirmer et al., 2022) and have chosen the USHCN[5] (Menne & Vose) as one of our evaluation datasets. USHCN encompasses daily measurements for 5 climate variables spanning over 150 years collected by widespread meteorological stations throughout the United States. We follow the pre-processing adopted by (De Brouwer et al., 2019) to obtain 1,114 stations and four-year observational periods between 1996 and 2000. To meet realistic forecasting requirements, we chunk the data to acquire a total of 26,736 IMTS, each of which uses the previous 24 months' climate data to forecast the following month's climate conditions.

A.6. Baseline Details

We incorporate seventeen relevant baselines for a fair comparison, covering the SOTA models from MTS forecasting, and IMTS classification, interpolation, and forecasting. We carefully search the key hyper-parameters of these models around their recommended setups. For a fair comparison across all models, we standardize the hidden dimensions to 64 for PhysioNet and MIMIC, and 32 for Human Activity and USHCN. We select a batch size of 192 for USHCN and 32 for the other datasets. The Adam optimizer is used for training, with early stopping implemented if there is no reduction in validation loss after 10 epochs.

A.6.1. Models for MTS forecasting

For MTS forecasting models, we input sequences after canonical pre-alignment and incorporate the observed time, mask information, and forecasting queries as additional features into these models.

DLinear (Zeng et al., 2023) decomposes time series into trend series and remainder series, subsequently employing two single-layer linear networks to model each of these sequences to accomplish the forecasting task. We use the following setting in our experiment: The window size of moving average is 25. The learning rate is 1×10^{-4}.

TimesNet (Wu et al., 2022) disassembles complex sequential changes into different periods through a modular structure, and achieves unified modeling of both inter-period and intra-period representation by transforming the original one-dimensional time series into a two-dimensional space to capture cross-time dependency for forecasting. We use the following setting in

[2] https://archive.physionet.org/challenge/2012
[3] https://mimic.mit.edu/
[4] https://archive.ics.uci.edu/dataset/196/localization+data+for+person+activity
[5] https://www.osti.gov/biblio/1394920

our experiment: The number of top-k for period is 5. The number of the encoder layers is 2. The learning rate is 1×10^{-4}.

PatchTST (Nie et al., 2022) is a Transformer-based model using Patch and Channel Independence to capture cross-time dependency for forecasting. We use the following setting in our experiment: The number of multi-heads is 2. The length of the patch is 16. The length of the stride is 8. The number of the encoder layers is 1. The learning rate is 1×10^{-4}.

Crossformer (Zhang & Yan, 2022) is a Transformer-based model using Cross-Time Attention and Cross-Dimension Attention to capture cross-dimension dependency and cross-time dependency for forecasting. We use the following setting in our experiment: The length of the segment is 12. The size of the window is 2. The number of the encoder layers is 1 for the Human Activity and 2 for other datasets. The number of the multi-heads is 3 for the Human Activity and 8 for other datasets. The learning rate is 1×10^{-3}.

We use the implementation provided by https://github.com/thuml/TimesNet to reproduce the above four baseline models.

GraphWavenet (Wu et al., 2019) leverages the self-adaptive adjacency matrix and diffusion convolution to capture the cross-dimension dependency and uses gated mechanism and dilated casual convolution to capture the cross-time dependency for forecasting. We use the following setting in our experiment: The dilation exponential is 3 for MIMIC and 2 for other datasets. The size of the kernel is 5 for PhysioNet and Human Activity, 7 for USHCN and 9 for MIMIC. The number of the blocks is 2 for Human Activity and 3 for other datasets. The number of convolution layers is 3 for MIMIC and 4 for other datasets. The learning rate is 1×10^{-3}. We use the official implementation at https://github.com/nnzhan/Graph-WaveNet.

MTGNN (Wu et al., 2020b) integrates graph convolutional networks and temporal convolutional networks to capture cross-dimensional relationships and cross-temporal dependencies in a direct and explicit manner. We use the following setting in our experiment: The dilation exponential is 2 for PhysioNet and Human Activity, 3 for USHCN and 4 for MIMIC. The size of the kernel is 7. The number of the convolution layers is 4 for MIMIC and USHCN, 5 for PhysioNet and Human Activity. The size of the subgraph is 5 for USHCN, 12 for Human Activity and 20 for PhysioNet and MIMIC. The learning rate is 1×10^{-3}. We use the official implementation at https://github.com/nnzhan/MTGNN.

StemGNN (Cao et al., 2020) transfers the spatiotemporal domain to the frequency domain through discrete Fourier transform and graph Fourier transform while capturing spatiotemporal dependencies in the frequency domain. We use the following setting in our experiment: The number of layers is 5 and the learning rate is 1×10^{-4}. We use the official implementation at https://github.com/microsoft/StemGNN.

CrossGNN (Huang et al., 2023) uses adaptive multi-scale identifier to construct multi-scale time series with different noise levels, subsequently utilize cross-scale GNN to capture the cross-time dependency and cross-variable GNN to capture the cross-dimension dependency for forecasting. We use the following setting in our experiment: The dimension of the scale vector and variable vector is 10. The scale number is 4. The number of cross-scale neighbors is 10. The learning rate is 1×10^{-3}. We use the official implementation at https://github.com/hqh0728/CrossGNN.

FourierGNN (Yi et al., 2023) initially constructs a hypervariate graph and transforms features into the Fourier space. Subsequently, it stacks Fourier graph operators in the Fourier domain and finally maps the convolved results back to the original feature space for forecasting. We use the following setting in our experiment: The number of frequency is 1. The scale is 0.02. The hidden size factor is 1. The sparsity threshold is 0.01. The learning rate is 1×10^{-3}. We use the official implementation at https://github.com/aikunyi/FourierGNN.

A.6.2. Models for irregular time series classification

To adapt these classification baseline models for forecasting tasks, we substitute their classification output layer with a MLP-based forecasting output layer that is the same as us.

GRU-D (Che et al., 2018) is a GRU-based model using time decay and missing data imputation strategies to handle irregularly sampled time series. We set learning rate to 1×10^{-3} and follows the implementation at https://github.com/zhiyongc/GRU-D.

SEFT (Horn et al., 2020) converts the time series into a set encoding, then using set functions to model them. We use the following setting in our experiment: The number of layers is 2. The learning rate is 1×10^{-3}. We use the implementation provided by https://github.com/mims-harvard/Raindrop.

RainDrop (Zhang et al., 2021b) employs neural message passing and temporal self-attention to model the dependencies among sensors, considering cross-sample shared relationships between sensors and adaptively estimates unaligned observations based on neighboring measurements. We use the following setting in our experiment: The dimension of the

observation is 4. The number of layers and heads for the Transformer is 2. The learning rate is 1×10^{-3}. We use the official implementation at https://github.com/mims-harvard/Raindrop.

Warpformer (Zhang et al., 2023a) is a Transformer-based model that adopts a tailored input representation, explicitly encapsulating both the within-series irregularities and inter-series variations. It further incorporates a warping module to flexibly synchronize irregular time series at a predefined scale, along with a custom-designed attention module for advanced representation learning. We use the following setting in our experiment: The number of warp is 0-0.2-1. The number of the heads is 1 and layers is 2. The learning rate is 1×10^{-3}. We use the official implementation at https://github.com/imJiawen/Warpformer.

A.6.3. Models for irregular time series interpolation and forecasting

mTAND (Shukla & Marlin, 2021) is an IMTS interpolation model that can be easily applied to forecasting tasks by only replacing the queries for interpolation with forecasting. It learns embeddings for numerical values corresponding to continuous time steps and generates fixed-length representations for variable-length sequential data using an attention mechanism. We use the following setting in our experiment: The encoder and the decoder is mTAND-rnn, the k-iwae is 5, the std is 0.01, the number of the ref-points is 64. The learning rate is 1×10^{-3}. We use the official implementation at https://github.com/reml-lab/mTAN.

Latent-ODE (Rubanova et al., 2019) is an ODE-based model that improves RNNs with continuous-time hidden state dynamics specified by neural ODEs. We use the following setting in our experiment: The number of the rec-layers and gen-layers is 3 for PhysioNet and USHCN and 1 for MIMIC and Human Activity. The learning rate is 1×10^{-3}. We use the official implementation at https://github.com/YuliaRubanova/latent-ode.

CRU (Schirmer et al., 2022) integrates the Kalman Filter with an encoder-decoder architecture to facilitate updates of the latent states in ODEs. We use the following setting in our experiment: The scaling factor of timestamps for numerical stability is 0.2 for PhysioNet and MIMIC and 0.3 for USHCN and Human Activity. The variance activation function in encoder is square and the variance activation function in decoder is exp. The activation function for transition net is relu. The number of bias is 15 for USHCN and Human Activity and 20 for PhysioNet and MIMIC. The bandwidth is 3 for USHCN and Human Activity and 10 for PhysioNet and MIMIC. The learning rate is 1×10^{-3}. We use the official implementation at https://github.com/boschresearch/Continuous-Recurrent-Units.

Neural Flow (Biloš et al., 2021) models the solution curves of ODEs through neural networks. We use the following setting in our experiment: The number of the flow-layers is 4 for MIMIC and 2 for other datasets. The number of the hidden-layers is 2 for MIMIC and 3 for other datasets. The rec-dims is 20 for MIMIC and 40 for other datasets. The flow model is GRU for MIMIC and coupling for other datasets. The time-net is TimeTanh for MIMIC and TimeLinear for other datasets. The time-hidden-dim is 8. The activation is ReLU for MIMIC and Tanh for other datasets. The learning rate is 1×10^{-3}. The decay of the learning rate is 0.33 for MIMIC and 0.5 for other datasets. We use the official implementation at https://github.com/mbilos/neural-flows-experiments.