TimeCHEAT: A Channel Harmony Strategy for Irregularly Sampled Multivariate Time Series Analysis

Jiexi Liu[1,2], **Meng Cao**[1,2], **Songcan Chen**[1,2*]

[1]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics
[2]MIIT Key Laboratory of Pattern Analysis and Machine Intelligence
{liujiexi, meng.cao, s.chen}@nuaa.edu.cn

Abstract

Irregularly sampled multivariate time series (ISMTS) are prevalent in reality. Due to their non-uniform intervals between successive observations and varying sampling rates among series, the channel-independent (CI) strategy, which has been demonstrated more desirable for *complete multivariate time series forecasting* in recent studies, has failed. This failure can be further attributed to the sampling sparsity, which provides insufficient information for effective CI learning, thereby reducing its capacity. When we resort to the channel-dependent (CD) strategy, even higher capacity cannot mitigate the potential loss of diversity in learning similar embedding patterns across different channels. We find that existing work considers CI and CD strategies to be mutually exclusive, primarily because they apply these strategies to the global channel. However, we hold the view that *channel strategies do not necessarily have to be used globally*. Instead, by appropriately applying them locally and globally, we can create an opportunity to take full advantage of both strategies. This leads us to introduce the Channel Harmony ISMTS Transformer (TimeCHEAT), which *utilizes the CD strategy locally and the CI strategy globally*. Specifically, we segment the ISMTS into sub-series level patches. Locally, the CD strategy aggregates information within each patch for time embedding learning, maximizing the use of relevant observations while reducing long-range irrelevant interference. Here, we enhance generality by transforming embedding learning into an edge weight prediction task using bipartite graphs, eliminating the need for special prior knowledge. Globally, the CI strategy is applied across patches, allowing the Transformer to learn individualized attention patterns for each channel. Experimental results indicate our proposed TimeCHEAT demonstrates competitive state-of-the-art performance across three mainstream tasks including classification, forecasting and interpolation.

Introduction

Irregularly sampled multivariate time series (ISMTS) are ubiquitous in real-world scenarios, such as healthcare (Goldberger et al. 2000; Reyna et al. 2020), meteorology (Schulz and Stattegger 1997; Cao et al. 2018) and transportation (Chen et al. 2022; Tang et al. 2020). Due to sensor malfunctions, transmission distortions, cost-reduction strategies, etc,

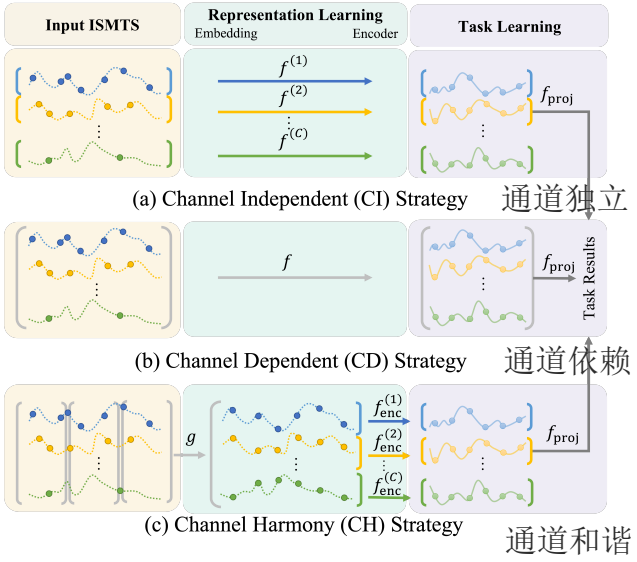

Figure 1: The difference between the 3 kinds of channel strategies.

ISMTS are characterized by inconsistent intervals between consecutive data points within a channel, asynchronous sampling across multiple channels and sometimes sampling sparsity. Such characteristics, arising from irregularities and multi-channels, pose a significant challenge to classical machine learning methods, which require the data to be defined in a consistent, fixed-dimensional feature space with constant intervals between successive timestamps.

Recent studies have sought to address this challenge using various approaches. One common method is a two-step process that treats ISMTS as synchronized, regularly sampled Normal Multivariate Time Series (NMTS) data with missing values, focusing on imputation strategies (Che et al. 2018; Yoon, Jordon, and Schaar 2018; Camino, Hammerschmidt, and State 2019; Tashiro et al. 2021; Le Morvan et al. 2021; Chen et al. 2022; Fan 2022; Du, Côté, and Liu 2023). However, accurate imputation is challenging, therefore separating imputation from downstream tasks may distort the underlying relationships and introduce substantial noise, leading to suboptimal results (Zhang et al. 2021a; Wu

*Corresponding Author

et al. 2021b; Agarwal et al. 2023; Sun et al. 2024). Since imputation needs to fully utilize information under the ISMTS scenario, two-step methods often use the channel-dependent (CD) strategy, merging all input dimensions in the learning process. While end-to-end models, which have gained considerable attention recently, have demonstrated superior performance over two-step approaches (Le Morvan et al. 2021). Some of these models treat ISMTS as discrete-time series, aggregating sample points from individual or multiple channels to create unified features (Wu et al. 2021b; Agarwal et al. 2023), while others preserve the continuous temporal dynamics of ISMTS data, often processing each channel independently (De Brouwer et al. 2019; Kidger et al. 2020; Schirmer et al. 2022; Jhin et al. 2022; Chowdhury et al. 2023). The end-to-end methods utilize CD and channel-independent (CI) strategies arbitrarily based on the existing models or experimental results. Therefore, despite the numerous time series models proposed to tackle irregular sampling in ISMTS, the challenge of effectively managing channel interactions still remains unresolved.

Since the interactions between multiple channels in ISMTS data are complex, a deeper understanding of these interactions lead to more accurate and insightful analysis. Given the advantages of end-to-end methods above, this paper focuses on these approaches. As shown in Fig.1, we divide the ISMTS analysis process into three main steps: ISMTS input, representation learning, and downstream tasks. Our primary focus is on representation learning, which includes both embedding learning and encoding. We do not separate these in Fig.1 (a) and (b) for two existing primary channel strategies CI and CD because the strategy remains consistent unchanged throughout the process. The **CI strategy** as in Fig.1(a), which uses individual models for each channel, works well in NMTS forecasting (Nie et al. 2023). In ISMTS analysis, CI allows channels to be processed according to their unique sampling patterns without forcing synchronization, preserving data integrity. However, CI struggles with limited generalizability and robustness on unseen channels (Han, Ye, and Zhan 2024) and the varying sampling rates across channels can result in the loss of crucial context, and channels with sparse sampling may fail to provide sufficient information for effective learning. On the other hand, the **CD strategy** in Fig.1(b) models all channels together, capturing complex temporal patterns but risks oversmoothing and struggles to fit individual channels, particularly when channel similarity is low.

To effectively manage channel interactions, this paper aims to balance necessary individual intra-channel treatment with inter-channel dependencies simultaneously. While existing work often treats CI and CD strategies as mutually exclusive, applying them only at a global level, we hold the view that these strategies can be more effective when used both locally and globally. This approach create an opportunity to fully leverage the advantages of both strategies. Accordingly, we propose the **C**hannel **H**armony Irr**E**gularly S**A**mpled Multivariate Time Series **T**ransformer (**TimeCHEAT**), which applies the CD strategy locally and the CI strategy globally, as illustrated in Fig.1(c). By applying the CD strategy locally and the CI strategy globally, the model achieves an effective balance between capturing detailed, context-specific information and preserving broader, channel-specific patterns. This hybrid approach enables more accurate and insightful analysis of ISMTS data, addressing the complex interactions between channels without sacrificing either detail or generality.

Specifically, we begin by dividing the ISMTS into subseries-level patches. Within each patch, we apply the CD strategy locally to effectively learn time embeddings. This approach aggregates nearby observations and neighborhood channel information, leveraging local smoothness to highlight their importance while avoiding interference from distant, irrelevant data. Traditional methods for time embedding learning often assume that larger time intervals weaken dependencies and relationships, leading to strong inductive biases. This can cause key timestamps with independent positions to be overlooked, thus affecting the extraction of important patterns. To address this, we transform the embedding learning process into an edge weight prediction task using bipartite graphs, which provides a more straightforward and generalizable learning method without above assumption. Additionally, we apply the CI strategy globally across patches, using the Transformer as the backbone encoder to capture individual attention patterns for each channel.

Our main contributions can be summarized as follows:

- We are the first to explore channel strategies for ISMTS analysis, proposing a novel and unified approach that better balances individual channel treatment with cross-channel modeling.
- We design a special time embedding method that can directly learn fix-length time embedding for ISMTS data without introducing special inductive bias.
- We are not limited to a specific forecasting task but attempt to propose a task-general channel strategy for ISMTS data, including classification, forecasting and interpolation.

Related Work

Irregularly Sampled Multivariate Time Series Analysis Models

An effective approach to analyzing ISMTS relies on understanding their unique properties. One natural idea is the two-step method, which treats ISMTS as NMTS with missing values and imputes the missingness before performing downstream tasks (Che et al. 2018; Yoon, Jordon, and Schaar 2018; Camino, Hammerschmidt, and State 2019; Tashiro et al. 2021; Chen et al. 2022; Fan 2022; Du, Côté, and Liu 2023; Wang et al. 2024). However, most two-step methods may distort the underlying relationships, introducing unsuitable assumptions and substantial noise due to incorrect imputation (Zhang et al. 2021a; Wu et al. 2021b; Agarwal et al. 2023), ultimately compromising the accuracy of downstream tasks. Therefore, recent work has shifted towards using end-to-end methods, which have been shown to outperform two-step methods both experimentally and theoretically (Le Morvan et al. 2021). One approach treats ISMTS as time series with discrete timestamps, focusing on

handling the irregularities by aggregating all sample points of either a single channel or all channels to extract a unified feature (Wu et al. 2021b; Agarwal et al. 2023). Others leverage the inherent continuity of time, thereby preserving the ongoing temporal dynamics present in ISMTS data (De Brouwer et al. 2019; Rubanova, Chen, and Duvenaud 2019; Kidger et al. 2020; Schirmer et al. 2022; Jhin et al. 2022; Chowdhury et al. 2023).

Due to these advancements, the issue of irregular sampling has been initially addressed. However, the relationships between channels have yet to be discussed. Since the channel strategies for most ISMTS analysis models simply merge all channels as input without considering the special inter-channel correlation nor treating different channels individually.

Channel Strategies for Time Series Analysis

Channel Strategies exist in all deep learning multivariate time series learning models, but the discussion of this topic mainly focuses on the *complete multivariate* time series *forecasting* task. The essential challenge of channel strategies for other downstream tasks as well as for ISMTS data from the model design perspective has not been solved. Before the discovery in Montero-Manso and Hyndman (2021) and the advent of PatchTST (Nie et al. 2023), the CD strategy was the mainstream approach in time series deep learning models (Wu et al. 2021a; Liu et al. 2021; Zhou et al. 2022) that tried to fully use information across channels. Conversely, the emergence of PatchTST breaks new ground in CI strategy, followed by several inspiring work (Li et al. 2023; Zeng et al. 2023; Chen et al. 2024; Han, Ye, and Zhan 2024) and conclude that CI has high capacity and low robustness, whereas CD is the opposite. Since existing work holds the view that CI and CD strategies are mutually exclusive, existing works focus on improving one of the strategies. Han, Ye, and Zhan (2024) design Predict Residuals with Regularization to incorporate a regularization term in the objective to address the non-robustness of the CD strategy and encourage smoothness in future forecasting. Moreover, Chen et al. (2024) dynamically group channels characterized by intrinsic similarities and leveraging cluster identity instead of channel identity. A similar idea has been proposed for ISMTS in LIFE (Zhang et al. 2021b) which collects credible and correlated channels to build individual features. Therefore, it remains challenging to develop a balanced channel strategy that contains advantages from both strategies. More importantly, the potential and effect of channel strategy for ISMTS remain under-explored.

Preliminaries

Notations. A set of ISMTS is a finite sequence of tuples $S = (X, M, Y) \in \mathbb{R}^d \times \{0,1\}^d \times \mathbb{R}^{d'}$, where X is the whole time series, M is a missingness indicator, and Y is a response of interest. Here, $d = N \times C \times T$ shows the dimensions of the input ISMTS, where N is the number of instances, C denotes the number of variates (i.e., channels) and T is the length of the time series. While $d' = 1$ for the classification task and $d' = N \times C \times T'$ for the forecasting problem, in which T' indicates the forecasting horizon. For each realization, (x, m, y), $m_{c,t} = 0$ indicates there is no observation at timestamp t in channel c, and $m_{c,t} = 1$ means it is observed. *Here, for brevity, we omit the data case index n for the n-th instance when the context is clear.*

Channel Dependent (CD) and Channel Independent (CI). We define multivariate time series in accordance with previous literature (Han, Ye, and Zhan 2024; Murtagh and Heck 2012; Zhou et al. 2021), which consists of two or more real-valued channels that depend on time. Other sequential data with extra dimensions like spatio-temporal data (Tan et al. 2023) or discrete values like natural language (Kenton and Toutanova 2019) are not included in this paper.

While different from existing work, our TimeCHEAT focuses on learning a set of nonlinear embedding functions F that can map the input ISMTS into best-described representations for further analysis. Under this scenario, the CI strategy models each channel separately and ignores the potential channel interactions. This approach needs C individual functions and each is typically denoted as $f^{(c)} : \mathbb{R}^T \to \mathbb{R}^T$, for $c = 1, \ldots, C$, where $f^{(c)}$ gets access to the ISMTS data specific to the c-th channel. On the contrary, the CD strategy models all the channels as a whole, using a single function $f : \mathbb{R}^{T \times C} \to \mathbb{R}^{T \times C}$. The learned representations $H \in \mathbb{R}^{T \times C}$ are utilized for further downstream tasks, including classification and forecasting in our paper.

Proposed TimeCHEAT Framework

In this paper, we propose an ISMTS analysis method that employs a novel channel harmony strategy for enhanced feature learning. As illustrated in Fig.2, the effectiveness of TimeCHEAT is largely ensured by 1) CD strategy for ISMTS embedding ensures comprehensive integration of multivariate data within a patch. 2) Bipartite graph modeling facilitates effective embedding learning by capturing complex relationships. 3) CI strategy for representation learning allows for tailored attention to individual channel dynamics. We will then discuss the above key points in the following subsections.

Locally Independent for Embedding Learning

Embedding learning is a crucial operation in end-to-end ISMTS learning. Some existing methods assume that large time intervals weaken dependencies (Che et al. 2018; Shukla and Marlin 2018, 2021, 2022). This assumption can lead to unrelated points being perceived as highly related while key points are overlooked, ultimately affecting the extraction of important patterns. While others rely on Neural Ordinary Differential Equations (NeuralODEs) (Chen et al. 2018; Jin et al. 2022; Scholz et al. 2022) which are known to be slow and often require additional features to handle irregularities.

To address these challenges, we observe that predicting edge weights in bipartite graphs can facilitate learning values based on query input, serving as an effective method for continuous-time embedding learning by the power of graph structures. It elegantly avoids the aforementioned assumptions and has been proven effective in ISMTS forecasting

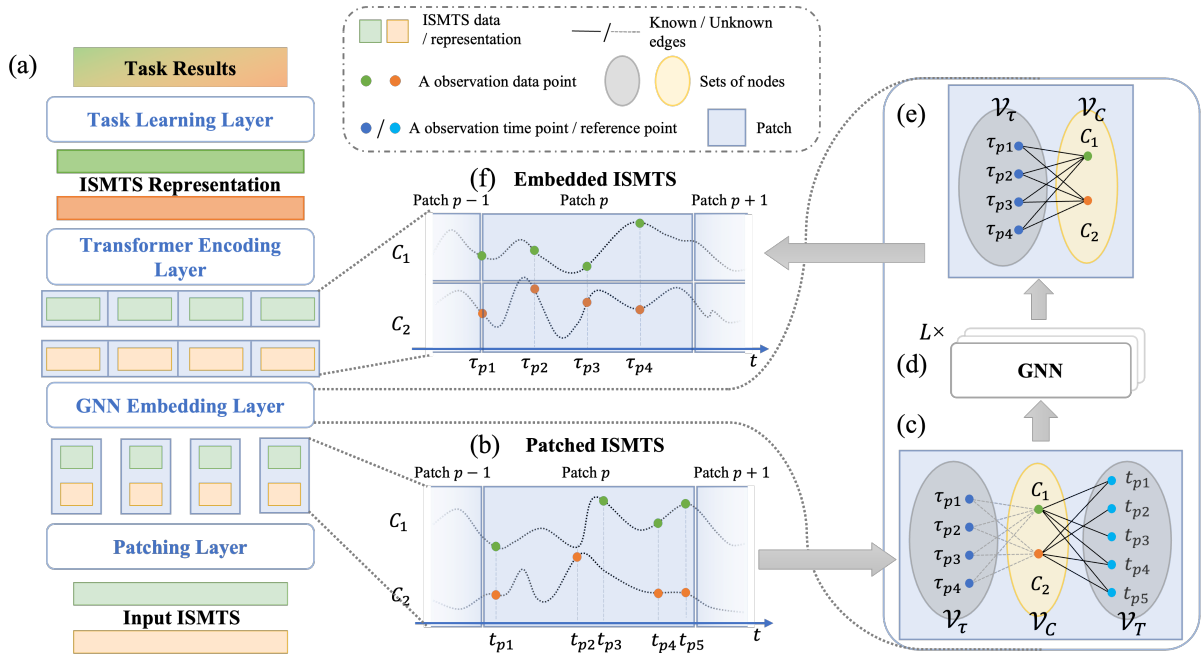

Figure 2: Overview of TimeCHEAT framework, shown in (a), containing 3 main steps, including ISMTS **embedding learning**, **Transformer encoding** and **task learning**. (b) is the patched ISMTS data with p patches. (c) is the initially established bipartite graph with known edges between channel and observation time step nodes and unknown edges between channel and reference point modes. (d) GNN module for unknown edges learning. (e) Learned edges between channel and reference point nodes. (f) Graph to embedding process to produce embedded ISMTS.

(Yalavarthi et al. 2024) and imputing missing data (You et al. 2020).

Graph Learning. We begin by segmenting the ISMTS into P equal-length, non-overlapping patches and constructing a bipartite graph for each patch p, as illustrated in Fig.2 (b) and (c). For simplicity, we omit the patch index p in this subsection, as all calculations are performed within a single patch. To effectively learn the local representation of the input ISMTS, each patch can be transferred into a bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consisting of two disjoint node sets and an edge set: the channel node set $\mathcal{V}_C$, the timestamp node set $\mathcal{V}_T$ and the observation measurement set $\mathcal{E} \subseteq \mathcal{V}_C \times \mathcal{V}_T$. Our goal is to learn from these edges to generalize relationships across channels and arbitrary timestamps.

To generate a fixed-dimensional embedding for each patch, making it suitable for subsequent neural network processing, we predefined a set of reference points $\boldsymbol{\tau} = [\tau_1, \cdots, \tau_K]$. These reference points are regular timestamps without observations, serving as a special subset of query nodes $\mathcal{V}_\tau$. The initial connection between reference points nodes set $\mathcal{V}_\tau$ and channels $\mathcal{V}_C$ are all 0, while the observation timestamp nodes set $\mathcal{V}_T$ and channels $\mathcal{V}_C$ have real observation value $x_{c,t}$. Here, we introduce an enhanced edge with an indicator i, denoted as

$$E(e,i) = \begin{cases} (x_{c,t}, 1), & e \in \mathcal{E}_T \\ (0,0), & e \in \mathcal{E}_\tau \end{cases} \tag{1}$$

when an edge is observed, i is set to 1, otherwise, 0.

We then introduce the Irregularity to Regularity Graph (I2RGraph), as shown in Fig.2 (c) to (e), aiming to transform irregularities into regularities by learning the edges $\mathcal{E}_\tau \subseteq \mathcal{V}_C \times \mathcal{V}_\tau$. This process can be formally expressed as follows:

$$\mathcal{E}_\tau := \text{I2RGraph}\left(\mathcal{V}_C, \mathcal{V}_T, \mathcal{E}_T, \mathcal{V}_\tau\right) \tag{2}$$

Patch Embedding Learning. When applying the CD strategy, it is insufficient to rely solely on channel ID numbers for learning. To address this, we encode the channel IDs to capture the underlying correlations between channels:

$$h_c^{\text{node},0} = \textbf{FFN}\left(\textbf{CM}\left(c\right)\right), c \in \mathcal{V}_C \tag{3}$$

Here, **CM** is a learnable matrix, initially set as an identity matrix, and c represents the channel ID used for encoding. **FFN** denotes a fully connected layer. Notably, if **CM** is not learnable, $h_c^{\text{node},0}$ cannot capture these underlying correlations due to the independence of one-hot encodings.

According to (Shukla and Marlin 2021), the timestamps can be encoded as a learned sinusoidal encoding and the initial edge embedding:

$$\begin{aligned} h_t^{\text{node},0} &= \sin\left(\textbf{FFN}\left(t\right)\right), t \in \mathcal{V}_T \cup \mathcal{V}_{\boldsymbol{\tau}} \\ h_e^{\text{edge},0} &= \textbf{FFN}\left(E(e,i)\right), e \in \mathcal{E} \end{aligned} \tag{4}$$

where $\cup$ denotes the union of two sets.

With initial embedding $(h^{\text{node},0})$ and $(h^{\text{edge},0})$ above, we then utilize Graph Attention Network (GNN) (Veličković et al. 2017; Yalavarthi et al. 2024) for further graph learning,

more details are in the supplementary. The proposed GNN architecture consists of L layers and the node features are updated layer wise from layer l to $l+1$ as follows:

$$\begin{aligned} &h_u^{\text{node},l+1} = \mathbf{MultiHead}^{(l)}\left(h_u^{\text{node},l}, H_u, H_u\right) \\ &s.t. \quad H_u = \left(\left[h_v^{\text{node},l} \| h_e^{\text{edge},l}\right]\right)_{v\in\mathcal{N}(u)} \end{aligned} \tag{5}$$

in which $\|$ represents the concatenation operation between two vectors or matrices, $\mathbf{MultiHead}$ is a multi-head attention block (Vaswani et al. 2017), $\mathcal{N}(u) := \{v | e_{u,v} \in \mathcal{E}, v \in \mathcal{V}\}$ indicates the neighborhood of u including all connected nodes of u though edges in $\mathcal{E}$. Eq. (5) means to search for the most relevant nodes among timestamp (channel) nodes to update the feature of a given channel (timestamp) node u.

The edge features are related to the corresponding channels, timestamps and edges themselves. Therefore, we adopt the following formula to update the edge features:

$$\begin{aligned} &h_e^{\text{edge},l+1} = \\ &\quad \alpha\left(h_e^{\text{edge},l} + \mathbf{FFN}^{(l)}\left(\left[h_c^{\text{node},l} \left\| h_t^{\text{node},l} \right\| h_e^{\text{edge},l}\right]\right)\right) \end{aligned} \tag{6}$$

in which, α is a non-linear activation, and a residual structure (He et al. 2016) has been adopted for stable learning.

Finally, the output embedding of the p-th patch is

$$H_p = \text{I2RGraph}(h^{\text{node},L}, h^{\text{edge},L}) \tag{7}$$

and we will use the overall embedding $H = \{H_p\}_{p=1}^{P}$ for further encoding.

Globally Independent for Transformer Encoding

After the embedding learning phase, we obtain several fixed-length patch embeddings H for the ISMTS, which can be directly fed into the encoder. In the case of symbol abuse, we denote the dimension of H as $P \times C \times T_P$. Subsequently, we apply a CI Transformer encoder to the patch time series, which maps the embedding H into a representation R for various downstream tasks. In other words, the subseries-level patches created during the embedding phase act as input tokens for the Transformer. This CI process, within each channel, contains a univariate time series that shares the same Transformer weights across all channels. To describe the temporal relationship, we provide the position encoding supported by the patch ID p and the position i for each channel as follows:

$$\begin{cases} PE_{p,2i} = \sin\left(\dfrac{p}{10000^{2i/T_P}}\right) \\ PE_{p,2i+1} = \cos\left(\dfrac{p}{10000^{2i/T_P}}\right) \end{cases} \tag{8}$$

To obtain the final representation R, we then feed the embedding H into a multi-head self-attention module with three learnable variable matrices, which are K, Q, and V.

$$R_c = \textbf{MultiHead}\left(KH_c^{PE}, QH_c^{PE}, VH_c^{PE}\right) \tag{9}$$

where R_c is the c-th channel of R, $H_c^{PE} = H_c + PE$, and $H_c \in \mathbb{R}^{P\times T_P}$ denotes c-th channel of H.

ISMTS Analysis Tasks. Here, for classification, the loss function is defined as:

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N} \ell_{CE}\left(\text{CLS}\left(R_n\right), y_n\right) \tag{10}$$

where $\text{CLS}(\cdot)$ denotes the projection head for classification, and $\ell_{CE}(\cdot)$ denotes the cross-entropy loss.

While for interpolation and forecasting, the associated loss function is defined as:

$$\mathcal{L} = \frac{1}{N}\sum_{n=1}^{N} \|M_n' \odot \left(\text{PRE}\left(R_n\right) - y_n\right)\|_2^2 \tag{11}$$

where $\text{PRE}(\cdot)$ denotes the projection head for forecasting or imputation. As observations might be missing also in the groundtruth data, to measure the accuracy we average an element-wise loss function over only valid values using M_n'.

Experiments

In this section, we show the effectiveness of the TimeCHEAT framework across 3 mainstream time series analysis tasks, including classification, interpolation, and forecasting. The results are reported as mean and standard deviation values calculated over 5 independent runs. The **bold** font highlights the top-performing method, while the underlined text marks the runner-up. Additional experimental setup details are provided in the Appendix due to space constraints.

Time Series Classification

Datasets and Experimental Settings. We use real-world datasets from healthcare to human activity domains for classification tasks: (1) **P19** (Reyna et al. 2020), with a missing ratio of 94.9%, includes $38,803$ patients monitored by 34 sensors. (2) **P12** (Goldberger et al. 2000) contains temporal measurements from 36 sensors of $11,988$ patients during the first 48 hours of ICU stay, with a missing ratio of 88.4%. (3) **PAM** (Reiss and Stricker 2012) consists of $5,333$ segments from 8 daily activities, measured by 17 sensors, with a missing ratio of 60.0%. *P19 and P12 are **imbalanced** binary label datasets* while PAM dataset contains 8 classes.

Following standard practice, we randomly split each dataset into training (80%), validation (10%), and test (10%) sets, using fixed indices across all methods. For the imbalanced P12 and P19 datasets, we evaluate performance using the area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPRC). For the nearly balanced PAM dataset, we use Accuracy, Precision, Recall, and F1 Score. For all of the above metrics, higher values indicate better performance.

Main Classification Results. We compare TimeCHEAT with 10 state-of-the-art methods for irregularly sampled time series classification, including Transformer (Vaswani et al. 2017), Trans-mean, GRU-D (Che et al. 2018), SeFT (Horn et al. 2020), mTAND (Shukla and Marlin 2021), IP-Net (Shukla and Marlin 2018), DGM^2-O (Wu et al. 2021b), MTGNN (Wu et al. 2020), Raindrop (Zhang et al. 2021a), and

Methods	P19		P12		PAM			
	AUROC	AUPRC	AUROC	AUPRC	Accuracy	Precision	Recall	F1 score
Transformer	80.7 ± 3.8	42.7 ± 7.7	83.3 ± 0.7	47.9 ± 3.6	83.5 ± 1.5	84.8 ± 1.5	86.0 ± 1.2	85.0 ± 1.3
Trans-mean	83.7 ± 1.8	45.8 ± 3.2	82.6 ± 2.0	46.3 ± 4.0	83.7 ± 2.3	84.9 ± 2.6	86.4 ± 2.1	85.1 ± 2.4
GRU-D	83.9 ± 1.7	46.9 ± 2.1	81.9 ± 2.1	46.1 ± 4.7	83.3 ± 1.6	84.6 ± 1.2	85.2 ± 1.6	84.8 ± 1.2
SeFT	81.2 ± 2.3	41.9 ± 3.1	73.9 ± 2.5	31.1 ± 4.1	67.1 ± 2.2	70.0 ± 2.4	68.2 ± 1.5	68.5 ± 1.8
mTAND	84.4 ± 1.3	50.6 ± 2.0	84.2 ± 0.8	48.2 ± 3.4	74.6 ± 4.3	74.3 ± 4.0	79.5 ± 2.8	76.8 ± 3.4
IP-Net	84.6 ± 1.3	38.1 ± 3.7	82.6 ± 1.4	47.6 ± 3.1	74.3 ± 3.8	75.6 ± 2.1	77.9 ± 2.2	76.6 ± 2.8
DGM^2-O	86.7 ± 3.4	44.7 ± 11.7	84.4 ± 1.6	47.3 ± 3.6	82.4 ± 2.3	85.2 ± 1.2	83.9 ± 2.3	84.3 ± 1.8
MTGNN	81.9 ± 6.2	39.9 ± 8.9	74.4 ± 6.7	35.5 ± 6.0	83.4 ± 1.9	85.2 ± 1.7	86.1 ± 1.9	85.9 ± 2.4
Raindrop	87.0 ± 2.3	51.8 ± 5.5	82.8 ± 1.7	44.0 ± 3.0	88.5 ± 1.5	89.9 ± 1.5	89.9 ± 0.6	89.8 ± 1.0
ViTST	89.2 ± 2.0	53.1 ± 3.4	**85.1** ± 0.8	**51.1** ± 4.1	95.8 ± 1.3	96.2 ± 1.3	96.1 ± 1.1	96.5 ± 1.2
TimeCHEAT	**89.5** ± 1.9	**56.1** ± 4.6	84.5 ± 0.7	48.2 ± 1.9	**96.5** ± 0.6	**97.1** ± 0.5	**96.9** ± 0.6	**97.0** ± 0.5

Table 1: Comparison with the baseline methods on ISMTS **classification** task.

Model	**Mean Squared Error** ($\times 10^{-3}$)				
Observed %	50%	60%	70%	80%	90%
RNN-VAE	13.418 ± 0.008	12.594 ± 0.004	11.887 ± 0.005	11.133 ± 0.007	11.470 ± 0.006
L-ODE-RNN	8.132 ± 0.020	8.140 ± 0.018	8.171 ± 0.030	8.143 ± 0.025	8.402 ± 0.022
L-ODE-ODE	6.721 ± 0.109	6.816 ± 0.045	6.798 ± 0.143	6.850 ± 0.066	7.142 ± 0.066
mTAND-Full	**4.139** ± 0.029	4.018 ± 0.048	4.157 ± 0.053	4.410 ± 0.149	4.798 ± 0.036
TimeCHEAT	4.185 ± 0.030	**3.981** ± 0.016	**3.657** ± 0.022	**3.642** ± 0.036	**3.686** ± 0.009

Table 2: Comparison with the baseline methods on ISMTS **interpolation** task on PhysioNet.

Methods	USHCN	MIMIC-III	MIMIC-IV	Physionet12
DLinear+	0.347 ± 0.065	0.691 ± 0.016	0.557 ± 0.001	0.380 ± 0.001
NLinear+	0.452 ± 0.101	0.726 ± 0.019	0.620 ± 0.002	0.382 ± 0.001
Informer+	0.320 ± 0.047	0.512 ± 0.064	0.420 ± 0.007	0.347 ± 0.001
FedFormer+	2.990 ± 0.476	1.100 ± 0.059	2.135 ± 0.304	0.455 ± 0.004
NeuralODE-VAE	0.960 ± 0.110	0.890 ± 0.010	–	–
GRU-Simple	0.750 ± 0.120	0.820 ± 0.050	–	–
GRU-D	0.530 ± 0.060	0.790 ± 0.060	–	–
T-LSTM	0.590 ± 0.110	0.620 ± 0.050	–	–
mTAND	0.300 ± 0.038	0.540 ± 0.036	ME	0.315 ± 0.002
GRU-ODE-Bayes	0.430 ± 0.070	0.480 ± 0.480	0.379 ± 0.005	0.329 ± 0.004
Neural Flow	0.414 ± 0.102	0.490 ± 0.004	0.364 ± 0.008	0.326 ± 0.004
CRU	0.290 ± 0.060	0.592 ± 0.049	ME	0.379 ± 0.003
GraFITi	0.272 ± 0.047	**0.396** ± 0.030	**0.225** ± 0.001	**0.286** ± 0.001
TimeCHEAT	**0.266** ± 0.069	0.462 ± 0.034	0.273 ± 0.002	0.290 ± 0.001

Table 3: Experimental results for **forecasting** next three time steps. – indicates no published results. ME indicates a Memory Error.

ViTST (Li, Li, and Yan 2023). Since *mTAND has demonstrated superiority over various recurrent models* such as RNNImpute (Che et al. 2018), Phased-LSTM (Neil, Pfeiffer, and Liu 2016), and ODE-based models like LATENT-ODE and ODE-RNN (Chen et al. 2018), our comparisons result contains mTAND, excluding results for the latter models.

As shown in Table 1, TimeCHEAT demonstrates competitive performance across the above three benchmark datasets, highlighting its effectiveness in typical time series classification tasks. In particular, for *imbalanced* binary classification, TimeCHEAT outperforms the leading baselines on the P19 dataset and achieves competitive results on the P12 dataset, trailing the top performer by only 0.5%. But it stands out due to its lower time and space complexity compared to ViTST though achieves SOTA performance, converting 1D time series into 2D images potentially leading to significant space inefficiencies due to the introduction of extensive blank areas, especially problematic in ISMTS. On the more complex task of 8-class classification in the PAM dataset, TimeCHEAT surpasses existing methods, with a 0.7% improvement in accuracy and a 0.9% increase in precision.

In almost all cases, our TimeCHEAT achieves consistent *low standard deviation* indicating it is a reliable model. Its performance remains steady across varying data samples and initial conditions, suggesting a strong potential for generalizing well to new, unseen data. This stability and predictability in performance enhance the confidence in the model's predictions, which is particularly crucial in sensitive areas such as medical diagnosis in clinical settings.

Time Series Interpolation

Datasets and experimental settings. **PhysioNet** (Silva et al. 2012) contains 37 variables recorded during the first 48 hours of ICU admission. For interpolation experiments, we utilize all $8,000$ instances, with a missing ratio of 78.0%.

The dataset is randomly split into 80% for training and 20% for testing, with 20% of the training data set aside for validation. Performance is evaluated using MSE, where lower values indicate better performance.

Methods	P19		P12	
	AUROC	AUPRC	AUROC	AUPRC
w/o correlation	88.0 ± 3.1	54.4 ± 5.0	83.5 ± 0.9	46.5 ± 2.3
w/ iTransformer	87.6 ± 2.4	54.7 ± 4.6	79.9 ± 1.5	39.2 ± 3.2
w/o correlation + w/ iTransformer	86.8 ± 2.7	54.2 ± 4.3	80.1 ± 1.7	39.4 ± 3.1
mTAND instead	87.4 ± 2.3	52.5 ± 3.4	84.3 ± 0.8	48.2 ± 1.0
TimeCHEAT	$\mathbf{89.5} \pm 1.9$	$\mathbf{56.1} \pm 4.6$	$\mathbf{84.5} \pm 0.7$	$\mathbf{48.2} \pm 1.9$

Table 4: Ablation studies on different strategies of TimeCHEAT in classification.

Main Interpolation Results. For the interpolation task, we compare TimeCHEAT with RNN-VAE, L-ODE-RNN (Chen et al. 2018), L-ODE-ODE (Rubanova, Chen, and Duvenaud 2019), and mTAND-full.

In this task, models are trained to predict or reconstruct values across the entire dataset based on a selected subset of observations. Experiments are conducted at varying observation levels, from 50% to 90% of observed points. During testing, models use the observed points to infer values at all time points within each test instance. We strictly follow the mTAND interpolation setup, where each column corresponds to a different setting. In each setting, a specific percentage of data is used to condition the model, which then predicts the remaining portion. Consequently, *results across columns are not directly comparable*, but each reflects the performance of interpolation under its respective conditions. As illustrated in Table 2, TimeCHEAT demonstrates great yet stable performance, highlighting its effectiveness in ISMTS interpolation.

Time Series Forecasting

Datasets and Experimental Settings. (1) **USHCN** (Menne, Williams Jr, and Vose 2015) is a preprocessed dataset with measurements of 5 variables from $1,280$ weather stations across the USA, featuring a missing ratio of 78.0%. (2) **MIMIC-III** (Johnson et al. 2016) contains recorded observations of 96 variables at 30-minute intervals, using data from the first 48 hours after ICU admission, with a missing ratio of 94.2%. (3) **MIMIC-IV** (Johnson et al. 2020) is built upon the MIMIC-III database with a missing ratio up to 97.8%. It adopts a modular approach to data organization, highlighting data provenance and facilitating both individual and combined use of disparate data sources. (4) **Physionet12** (Silva et al. 2012) includes medical records from $12,000$ ICU patients, capturing 37 vital signs during the first 48 hours of admission, with a missing ratio of 80.4%. The performance is evaluated using MSE.

Main Forecasting Results. We compare TimeCHEAT with various ISMTS forecasting models, including Grafiti (Yalavarthi et al. 2024), GRU-ODE-Bayes (De Brouwer et al. 2019), Neural Flows (Biloš et al. 2021), CRU (Schirmer et al. 2022), NeuralODE-VAE (Chen et al. 2018), GRUSimple, GRU-D, TLSTM (Baytas et al. 2017), mTAND, and enhanced versions of Informer (Zhou et al. 2021), Fedformer (Zhou et al. 2022), DLinear, and NLinear (Zeng et al. 2023), referred to as Informer+, Fedformer+, DLinear+, and NLinear+.

Following the GraFITi setup, for the USHCN dataset, the model observes the first 3 years and forecasts the next 3 time steps. For the other datasets, the model observes the first 36 hours and predicts the next 3 time steps. As indicated in Table 3, TimeCHEAT consistently shows competitive performance across all datasets, consistently ranking within the top two among baseline models. While GraFITi excels in scenarios where explicitly modeling the relationship between observation and prediction points is advantageous, TimeCHEAT remains highly competitive without relying on task-specific priors.

Ablation Study

We use two imbalance binary label datasets P12 and P19 in the classification task as an example to conduct the ablation study. We verify the necessity of three main designs: 1) learnable correlation between multiple channels in the embedding procedure, 2) channel-dependent embedding learning without special assumptions, and 3) channel-independent Transformer encoder.

As shown in Table 4, the full TimeCHEAT framework, which includes all original components (line 7), delivers the best performance. When local correlations between channels are removed (line 3), by replacing the channel encoding with a one-hot binary indicator vector (resulting in CI embedding learning), sparse sampling channels lack sufficient information and fail to aggregate crucial data from related channels for improved embedding. Discarding the CI encoder and switching to a vanilla Transformer (line 4) leads to a significant drop in classification accuracy, underscoring the effectiveness of the CI strategy in the encoding phase. Combining the above two changes results in a fully CI model (line 5), which yields nearly the worst accuracy among all tested conditions. Finally, replacing the local graph embedding with CI mTAND (line 5), while partially mitigating issues due to its assumptions about timestamp distances, still falls short of the best performance due to the limitations of the CI strategy.

Conclusion

In this paper, we introduce TimeCHEAT, a novel framework for ISMTS analysis. TimeCHEAT's innovative channel harmony strategy effectively balances individual channel processing with inter-channel interactions, improving ISMTS analysis performance. Our results show that combining the CD strategy locally with the CI strategy globally harnesses the strengths of both approaches, as well or better than a range of baseline and SOTA models. A key contribution of this work is to design the bipartite graphs with CD strategy locally to transform embedding learning into an edge weight prediction task, avoiding introducing inappropriate prior assumptions and enabling fixed-length embeddings for the encoder. Additionally, the CI strategy applied globally across patches allows the Transformer to learn individualized attention patterns for each channel, leading to superior representations for downstream tasks.

Acknowledgements

The authors wish to thank all the donors of the original datasets, and everyone who provided feedback on this work. This work is supported by the Key Program of NSFC under Grant No.62376126, Postgraduate Research & Practice Innovation Program of Jiangsu Province under Grant No.KYCX21_0225.

References

Agarwal, R.; Sinha, A.; Prasad, D. K.; Clausel, M.; Horsch, A.; Constant, M.; and Coubez, X. 2023. Modelling Irregularly Sampled Time Series Without Imputation. *arXiv preprint arXiv:2309.08698*.

Baytas, I. M.; Xiao, C.; Zhang, X.; Wang, F.; Jain, A. K.; and Zhou, J. 2017. Patient subtyping via time-aware LSTM networks. In *ACM SIGKDD*, 65–74.

Biloš, M.; Sommer, J.; Rangapuram, S. S.; Januschowski, T.; and Günnemann, S. 2021. Neural flows: Efficient alternative to neural ODEs. *NeurIPS*, 34: 21325–21337.

Camino, R. D.; Hammerschmidt, C. A.; and State, R. 2019. Improving missing data imputation with deep generative models. *arXiv preprint arXiv:1902.10666*.

Cao, W.; Wang, D.; Li, J.; Zhou, H.; Li, L.; and Li, Y. 2018. Brits: Bidirectional recurrent imputation for time series. *NeurIPS*, 31.

Che, Z.; Purushotham, S.; Cho, K.; Sontag, D.; and Liu, Y. 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1): 1–12.

Chen, J.; Lenssen, J. E.; Feng, A.; Hu, W.; Fey, M.; Tassiulas, L.; Leskovec, J.; and Ying, R. 2024. From Similarity to Superiority: Channel Clustering for Time Series Forecasting. *arXiv preprint arXiv:2404.01340*.

Chen, R. T.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018. Neural ordinary differential equations. *NeurIPS*, 31.

Chen, X.; Zhang, C.; Zhao, X.-L.; Saunier, N.; and Sun, L. 2022. Nonstationary temporal matrix factorization for multivariate time series forecasting. *arXiv preprint arXiv:2203.10651*.

Chowdhury, R. R.; Li, J.; Zhang, X.; Hong, D.; Gupta, R. K.; and Shang, J. 2023. Primenet: Pre-training for irregular multivariate time series. In *AAAI*, volume 37, 7184–7192.

De Brouwer, E.; Simm, J.; Arany, A.; and Moreau, Y. 2019. GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series. *NeurIPS*, 32.

Du, W.; Côté, D.; and Liu, Y. 2023. Saits: Self-attention-based imputation for time series. *Expert Systems with Applications*, 219: 119619.

Fan, J. 2022. Dynamic Nonlinear Matrix Completion for Time-Varying Data Imputation. In *AAAI*.

Goldberger, A. L.; Amaral, L. A.; Glass, L.; Hausdorff, J. M.; Ivanov, P. C.; Mark, R. G.; Mietus, J. E.; Moody, G. B.; Peng, C.-K.; and Stanley, H. E. 2000. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *circulation*, 101(23): e215–e220.

Han, L.; Ye, H.-J.; and Zhan, D.-C. 2024. The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting. *IEEE Transactions on Knowledge and Data Engineering*, (01): 1–14.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Horn, M.; Moor, M.; Bock, C.; Rieck, B.; and Borgwardt, K. 2020. Set functions for time series. In *ICML*, 4353–4363. PMLR.

Jhin, S. Y.; Lee, J.; Jo, M.; Kook, S.; Jeon, J.; Hyeong, J.; Kim, J.; and Park, N. 2022. Exit: Extrapolation and interpolation-based neural controlled differential equations for time-series classification and forecasting. In *ACM Web Conference*, 3102–3112.

Jin, M.; Zheng, Y.; Li, Y.-F.; Chen, S.; Yang, B.; and Pan, S. 2022. Multivariate time series forecasting with dynamic graph neural odes. *IEEE Transactions on Knowledge and Data Engineering*.

Johnson, A.; Bulgarelli, L.; Pollard, T.; Horng, S.; Celi, L. A.; and Mark, R. 2020. Mimic-iv. *PhysioNet. Available online at: https://physionet. org/content/mimiciv/1.0/(accessed August 23, 2021)*, 49–55.

Johnson, A.; Pollard, T. J.; Shen, L.; Lehman, L.-w. H.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Celi, L. A.; and Mark, R. G. 2016. MIMIC-III, a freely accessible critical care database Sci. *Data*, 3(1): 1.

Kenton, J. D. M.-W. C.; and Toutanova, L. K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, 4171–4186.

Kidger, P.; Morrill, J.; Foster, J.; and Lyons, T. 2020. Neural controlled differential equations for irregular time series. *NeurIPS*, 33: 6696–6707.

Le Morvan, M.; Josse, J.; Scornet, E.; and Varoquaux, G. 2021. What's a good imputation to predict with missing values? *NeurIPS*, 34: 11530–11540.

Li, Z.; Li, S.; and Yan, X. 2023. Time Series as Images: Vision Transformer for Irregularly Sampled Time Series. In *NeurIPS*.

Li, Z.; Qi, S.; Li, Y.; and Xu, Z. 2023. Revisiting long-term time series forecasting: An investigation on linear mapping. *arXiv preprint arXiv:2305.10721*.

Liu, S.; Yu, H.; Liao, C.; Li, J.; Lin, W.; Liu, A. X.; and Dustdar, S. 2021. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *ICLR*.

Menne, M. J.; Williams Jr, C.; and Vose, R. S. 2015. United States historical climatology network daily temperature, precipitation, and snow data. *Carbon Dioxide Information Analysis Center, Oak Ridge National Laboratory, Oak Ridge, Tennessee*.

Montero-Manso, P.; and Hyndman, R. J. 2021. Principles and algorithms for forecasting groups of time series: Locality and globality. *International Journal of Forecasting*, 37(4): 1632–1653.

Murtagh, F.; and Heck, A. 2012. *Multivariate data analysis*, volume 131. Springer Science & Business Media.

Neil, D.; Pfeiffer, M.; and Liu, S.-C. 2016. Phased lstm: Accelerating recurrent network training for long or event-based sequences. *NeurIPS*, 29.

Nie, Y.; Nguyen, N. H.; Sinthong, P.; and Kalagnanam, J. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *ICLR*.

Reiss, A.; and Stricker, D. 2012. Introducing a new benchmarked dataset for activity monitoring. In *16th international symposium on wearable computers*, 108–109. IEEE.

Reyna, M. A.; Josef, C. S.; Jeter, R.; Shashikumar, S. P.; Westover, M. B.; Nemati, S.; Clifford, G. D.; and Sharma, A. 2020. Early prediction of sepsis from clinical data: the PhysioNet/Computing in Cardiology Challenge 2019. *Critical care medicine*, 48(2): 210–217.

Rubanova, Y.; Chen, R. T.; and Duvenaud, D. K. 2019. Latent ordinary differential equations for irregularly-sampled time series. *NeurIPS*, 32.

Schirmer, M.; Eltayeb, M.; Lessmann, S.; and Rudolph, M. 2022. Modeling irregular time series with continuous recurrent units. In *ICML*, 19388–19405. PMLR.

Scholz, R.; Born, S.; Duong-Trung, N.; Cruz-Bournazou, M. N.; and Schmidt-Thieme, L. 2022. Latent Linear ODEs with Neural Kalman Filtering for Irregular Time Series Forecasting. *NeurIPS*.

Schulz, M.; and Stattegger, K. 1997. SPECTRUM: Spectral analysis of unevenly spaced paleoclimatic time series. *Computers & Geosciences*, 23(9): 929–945.

Shukla, S. N.; and Marlin, B. 2018. Interpolation-Prediction Networks for Irregularly Sampled Time Series. In *ICLR*.

Shukla, S. N.; and Marlin, B. 2021. Multi-Time Attention Networks for Irregularly Sampled Time Series. In *ICLR*.

Shukla, S. N.; and Marlin, B. 2022. Heteroscedastic Temporal Variational Autoencoder For Irregularly Sampled Time Series. In *ICLR*.

Silva, I.; Moody, G.; Scott, D. J.; Celi, L. A.; and Mark, R. G. 2012. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *2012 Computing in Cardiology*, 245–248. IEEE.

Sun, C.; Li, H.; Song, M.; Cai, D.; Zhang, B.; and Hong, S. 2024. Time pattern reconstruction for classification of irregularly sampled time series. *Pattern Recognition*, 147: 110075.

Tan, C.; Li, S.; Gao, Z.; Guan, W.; Wang, Z.; Liu, Z.; Wu, L.; and Li, S. Z. 2023. Openstl: A comprehensive benchmark of spatio-temporal predictive learning. *NeurIPS*, 36: 69819–69831.

Tang, X.; Yao, H.; Sun, Y.; Aggarwal, C.; Mitra, P.; and Wang, S. 2020. Joint modeling of local and global temporal dynamics for multivariate time series forecasting with missing values. In *AAAI*, volume 34, 5956–5963.

Tashiro, Y.; Song, J.; Song, Y.; and Ermon, S. 2021. CSDI: Conditional Score-based Diffusion Models for Probabilistic Time Series Imputation. *NeurIPS*, 34.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *NeurIPS*, 30.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Wang, J.; Du, W.; Cao, W.; Zhang, K.; Wang, W.; Liang, Y.; and Wen, Q. 2024. Deep Learning for Multivariate Time Series Imputation: A Survey. *arXiv preprint arXiv:2402.04059*.

Wu, H.; Xu, J.; Wang, J.; and Long, M. 2021a. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *NeurIPS*, 34: 22419–22430.

Wu, Y.; Ni, J.; Cheng, W.; Zong, B.; Song, D.; Chen, Z.; Liu, Y.; Zhang, X.; Chen, H.; and Davidson, S. B. 2021b. Dynamic gaussian mixture based deep generative model for robust forecasting on sparse multivariate time series. In *AAAI*, volume 35, 651–659.

Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; and Zhang, C. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *ACM SIGKDD*, 753–763.

Yalavarthi, V. K.; Madhusudhanan, K.; Scholz, R.; Ahmed, N.; Burchert, J.; Jawed, S.; Born, S.; and Schmidt-Thieme, L. 2024. GraFITi: Graphs for Forecasting Irregularly Sampled Time Series. In *AAAI*, 16255–16263.

Yoon, J.; Jordon, J.; and Schaar, M. 2018. Gain: Missing data imputation using generative adversarial nets. In *ICML*, 5689–5698. PMLR.

You, J.; Ma, X.; Ding, Y.; Kochenderfer, M. J.; and Leskovec, J. 2020. Handling missing data with graph representation learning. *NeurIPS*, 33: 19075–19087.

Zeng, A.; Chen, M.; Zhang, L.; and Xu, Q. 2023. Are transformers effective for time series forecasting? In *AAAI*, volume 37, 11121–11128.

Zhang, X.; Zeman, M.; Tsiligkaridis, T.; and Zitnik, M. 2021a. Graph-Guided Network for Irregularly Sampled Multivariate Time Series. In *ICLR*.

Zhang, Z.-Y.; Zhang, S.-Q.; Jiang, Y.; and Zhou, Z.-H. 2021b. LIFE: Learning individual features for multivariate time series prediction with missing values. In *ICDM*, 1511–1516. IEEE.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *AAAI*, volume 35, 11106–11115.

Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; and Jin, R. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *ICML*, 27268–27286. PMLR.