

Disentangling Dynamics: Advanced, Scalable and Explainable Imputation for Multivariate Time Series

Shuai Liu, Xiucheng Li, Yile Chen, Yue Jiang, Gao Cong

Abstract—Missing values pose a formidable obstacle in multivariate time series analysis. Existing imputation methods rely on entangled representations that struggle to simultaneously capture multiple orthogonal time-series patterns, leading to suboptimal performance and limited interpretability. Meanwhile, requiring the entire data span as input renders these models impractical for long time series. To address these issues, we propose **TIDER** and its enhanced version, **AdaTIDER**. **TIDER** employs low-rank matrix factorization and disentangled temporal representations to model intricate dynamics like trend, seasonality, and local bias. However, **TIDER** is limited to single-period modeling and does not explicitly capture dependencies between channels. To overcome these limitations, **AdaTIDER** incorporates adaptive cross-channel dependency modeling and multi-period seasonality representations. These advancements enable it to dynamically capture variable relationships and complex multi-period patterns, significantly enhancing imputation accuracy and interpretability, while maintaining **TIDER**'s scalability. Extensive experiments on real-world datasets validate the superiority of our models in imputation accuracy, scalability, interpretability, and robustness.

Index Terms—Time Series, Imputation, Disentangled Representation, Matrix Factorization, Scalability, and Interpretability

I. INTRODUCTION

Recent progress in data acquisition and storage have resulted in the accumulation of time series across diverse fields, including traffic [1], healthcare [2], meteorology [3], and power demand [4]. This abundance has led to the development of numerous models [5]–[9]. However, as most models rely on complete time series, they face challenges with missing values. Therefore, they are susceptible to suboptimal performance or outright failure when meeting missing values [10]–[12]. In practice, missing data frequently arises due to equipment malfunctions, transmission errors, or high data collection costs.

Given these issues, multivariate time series imputation has become a pivotal preliminary, as its quality directly impacts the effectiveness of downstream tasks. Accurate imputation restores original data's temporal patterns, enabling reliable analyses, while arbitrary imputation distorts inherent patterns, leading to erroneous insights and degraded analytical performance. Intuitively, multivariate time series can be represented as an $N \times T$ matrix, where N is the number of channels,

S. Liu, G. Cong, Y. L. Chen, and Y. Jiang are with School of Computer Science and Engineering, Nanyang Technological University, Singapore, 639798. E-mail: {SHUAI004@e., gaocong@, yile001@e., yue013@e.}ntu.edu.sg

X. C. Li is with School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Guangdong, China, 518055. E-mail: lixiucheng@ hit.edu.cn

and T is the time span. This matrix-based representation facilitates the exploration of cross-channel correlations (rows) and temporal dynamics (columns) for imputation. Early imputation approaches [13]–[15] employed simplistic strategies like aggregating observed entries across channels based on estimated similarities or relying on local smoothness or linear assumptions within the same channel. However, these methods have inherent limitations in capturing nonlinearities and intricate correlations, resulting in suboptimal performance.

Recent years have witnessed a surge in deep learning-based time series imputation [16]–[18]. These methods use RNNs or Transformers to capture nonlinear dynamics. By mapping input data to hidden state space and updating nonlinearly, such models have achieved notable outcomes. Some methods further incorporate multi-task loss [18], [19] or generative models like GANs, VAEs, and diffusion models [16], [17], [20], [21] to improve performance. Despite their success, these methods rely on a single entangled representation, typically the hidden state, to capture underlying dynamics. These approaches are suboptimal for real-world multivariate time series, where dynamics often involve multiple independent factors like trend, seasonality (periodicity), and local idiosyncrasies [22], [23]. Attempting to model these factors using a single entangled representation can lead to suboptimal results, as such entangled representation needs to compromise itself to explain multiple orthogonal patterns (e.g., exogenous interventions versus global patterns) simultaneously [24]. Another drawback is their scalability, as they require the entire time span of time series as input in each forward step, leading to computational inefficiency for long time series [25]. Additionally, the hidden states learned by these models represent complex entangled combinations of various factors, which are difficult to extract interpretable information to explain the imputation process.

To address these limitations, we propose **TIDER**, as detailed in our prior ICLR publication [26]. It explicitly models time-series dynamics through a low-rank matrix decomposition-based structure with distinct disentangled representations to capture different components. Specifically, **TIDER** introduces three distinct representations: a trend representation for global patterns, a Fourier series-based seasonality representation to incorporate periodic inductive bias, and a local bias representation captures time step-specific idiosyncrasies. This disentanglement provides **TIDER** with great flexibility and robustness in modeling complex time-series dynamics. However, **TIDER** has its limitations. It lacks regularization for channel representations, where different channels often exhibit inherent correlations [27]. Additionally,

TIDER fails to capture multi-periodic patterns, as it merely model a single predefined period's seasonality pattern. As highlighted in [22], [28], many time series exhibit multiple seasonal patterns, such as daily, weekly, and yearly cycles.

To address these issues, we present AdaTIDER (Adaptive TIDER), an enhanced version of TIDER that achieves adaptive learning across both temporal and cross-channel dimensions. A key innovation is its fast Fourier transform-based approach to adaptively capture multi-period seasonality features. Unlike TIDER, which models periodic features under a single pre-defined period, AdaTIDER incorporates several adaptive single-period seasonality representations, each targeting a specific periodicity. This empowers AdaTIDER to effectively impute time series with varying periodicities. Additionally, AdaTIDER leverages an adaptive adjacency matrix to model correlations between channels. By leveraging this matrix, AdaTIDER generates more accurate and reasonable channel representations. Specifically, each channel representation is constructed by concatenating a residual and a non-residual component. The residual component computes cross-channel correlations, which are then utilized to regularize the non-residual part. This design effectively mitigates the 'bootstrapping' issue, where correlations derived from channel information are directly imposed back on the same channel, leading to interference and weakening learning effectiveness [29], [30]. By incorporating correlation regularization from the residual component, AdaTIDER enhances its ability to capture and represent inter-channel relationships. While maintaining scalability, AdaTIDER achieves higher imputation accuracy and conveys richer semantic information. Comprehensive quantitative and qualitative evaluations demonstrate the significant performance of AdaTIDER in terms of imputation effectiveness, scalability, and interpretability. Overall, the contributions of this paper can be summarized as follows:

- We propose TIDER, a novel method which utilizes effective and explainable disentangled representations to capture complex time series dynamics. To the best of our knowledge, TIDER is the first model to learn disentangled representations for time series imputation.
- We introduce AdaTIDER as an improved version of TIDER. It achieves greater temporal and cross-channel adaptivity by learning multi-period seasonality representations and adaptive adjacency matrix, enabling more accurate and interpretable imputation.
- Extensive experiments thoroughly evaluate TIDER's and AdaTIDER's performance. The results demonstrate that TIDER exhibits excellent performance in accuracy, scalability, and interpretability. Moreover, AdaTIDER outperforms TIDER, showcasing its superiority and effectiveness in imputing multivariate time series.

II. RELATED WORK

Early time series imputation methods use simple statistical techniques, adapting inter-channel similarities and local temporal smoothness. SimpleMean and SimpleMedian [31] use averages and medians, MICE [32] applies chain equations, and KNN [13] aggregates similar channels. These methods

overlook nonlinear dynamics and complex correlations across channels, limiting their imputation performance.

Deep learning imputation models are recently used to capture nonlinearity. RNN-based models excel at modeling sequential data. BRITS [19] uses bidirectional RNNs, while NAOMI [16] and GAIN [33] combine GAN and RNN. VAEs also play a significant role. PoGeVON [34] introduces a position-aware graph-enhanced VAE to model spatial and temporal dependencies. Meanwhile, diffusion models have gained attention for conditional imputation. CSDI [17] employs score-based diffusion models, while SADI [21] incorporates similarity-aware diffusion. However, these methods rely on a single entangled representation, limiting their ability to model multiple temporal patterns like trends, seasonality, and local fluctuations separately [22], [23]. Furthermore, they face scalability challenges when applied to long time series, as they need to process the entire length of time series in each forward step to capture temporal dynamics [25]. Additionally, the learned representations contain complex and entangled components, making it challenging to extract explainable information. This lack of interpretability hinders the understanding of imputation process and underlying dynamics.

Fourier transform is widely used to capture periodic patterns by decomposing data into distinct frequency components, effectively identifying and reconstructing recurring patterns, making it suitable for imputation [35], [36]. However, real-world time series often exhibit more than periodic dynamics. They frequently incorporate non-stationary behaviors like trends and localized biases, which traditional Fourier methods struggle to capture [22], [23]. As the classical Fourier transform assumes stationarity and a single consistent periodicity, its applicability is limited in complex, practical scenarios.

Many time series are composed of various periods [22]. To address the presence of multiple periodicities, various approaches have been explored. MuSDRI [37] combines seasonal-trend decomposition using Loess with RNNs to address multiple seasonal patterns in its imputation framework. TimesNet [38] transforms 1D time series into multiple 2D tensors, thereby capturing multi-period signals for imputation in an end-to-end neural architecture. While these methods effectively exploit multiple seasonal factors, they primarily rely on deep learning backbones, which lack explicit disentanglement of individual temporal components, making interpretation more challenging. Additionally, these deep learning models often require substantial computational resources for long sequences, further limiting their scalability.

Our proposed TIDER and AdaTIDER aim to overcome the above limitations by using a low-rank matrix factorization (MF) framework [39]. Unlike deep learning based methods—which typically encode entire sequences end-to-end for imputation—MF-based models decompose the observed time series into latent factors U and V. This decomposition is particularly advantageous for long time series, where RNN-based models suffer from vanishing gradients and sequential dependencies, while Transformer-based models face quadratic computational complexity with respect to sequence length. Additionally, MF naturally allows for the integration of explicit structural constraints to enhance interpretability,

while deep learning methods tend to learn complex entangled representations that lack explainability. However, these models often overlook temporal continuity in time series. TRMF [40] imposes autoregressive constraints on temporal factor \mathbf{V} , while LCR [41] introduces a Laplacian kernel for temporal regularization. Although these approaches capture some temporal dependencies, they still rely on an entangled representation to capture all dynamics, leading to similar issues as encountered in deep learning-based methods. In contrast, our models introduce multiple disentangled representations with specialized constraints for different temporal components. TIDER explicitly models key time-series dynamics like trend, seasonality, and local bias through distinct representations. However, TIDER lacks cross-channel constraints and captures only one single pre-defined period's seasonality. To overcome this, AdaTIDER further fuses fast Fourier transform for adaptive multi-period seasonality learning, enabling precise decomposition of temporal patterns across varying periods. Additionally, an adaptive adjacency matrix is employed for cross-channel dependency modeling, further improving imputation accuracy and interpretability.

III. PROBLEM STATEMENT

Given N univariate time series collected in T time steps, denoted as $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N \in \mathbb{R}^T$, we organize them into a matrix $\mathbf{X} \in \mathbb{R}^{N \times T}$, whose n -th row represents n -th univariate time series \mathbf{x}_n (channel) and t -th column denotes all observations at time step t . \mathbf{X} is incomplete, with some entries missing. We aim to infer missing values from the observed ones, and we denote the mask matrix as $\mathbf{M} \in \{0, 1\}^{N \times T}$, where

$$M_{ij} = \begin{cases} 1, & \text{if } X_{ij} \text{ is observed,} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

IV. TIME-SERIES IMPUTATION WITH DISENTANGLED TEMPORAL REPRESENTATIONS

A. Method Overview

The core idea of TIDER is to decompose multivariate time series matrix \mathbf{X} into two latent factors \mathbf{U} and \mathbf{V} , where \mathbf{U} retains channel-specific features and \mathbf{V} represents multiple disentangled temporal components. This factorization, as shown in Fig. 1, is motivated by two primary reasons: 1) strong correlations among time series channels (rows of \mathbf{X}), 2) significant temporal variations across time steps (columns of \mathbf{X}). The benefits of this design are twofold: 1) cross-channel correlations are decoupled as \mathbf{U} only preserves channel-specific features 2) temporal dynamics are isolated into \mathbf{V} , and it enables us to model the complex temporal dynamics with multiple explainable disentangled representations. Inspired by the principle that seasonality and trend can be treated as independent factors in time-series generation under mild assumptions [22], and independence serves as a proxy criterion for disentanglement [42], we enforce distinct constraints on different representations to achieve disentanglement and introduce specific inductive biases to capture semantically-independent patterns. Specifically, we consider *trend*, *seasonality*, and *bias*, represented by \mathbf{V}_t , \mathbf{V}_s , and \mathbf{V}_b . Trend representation matrix $\mathbf{V}_t \in \mathbb{R}^{D \times T}$ captures gradual and smooth intrinsic trends,

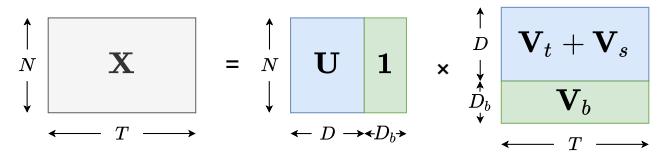


Fig. 1: Architecture of TIDER. \mathbf{X} is the time series matrix. \mathbf{V}_t , \mathbf{V}_s , and \mathbf{V}_b are trend, seasonality, and bias representation.

while seasonality representation matrix $\mathbf{V}_s \in \mathbb{R}^{D \times T}$ reveals periodic patterns. \mathbf{V}_t and \mathbf{V}_s jointly determine endogenous dynamics. Bias representation matrix $\mathbf{V}_b \in \mathbb{R}^{D_b \times T}$ captures time-specific variations, which are orthogonal to global dynamics but shared across channels. Hence, we interpret it as a residual term matrix, i.e., $\mathbf{X} - \mathbf{U}(\mathbf{V}_t + \mathbf{V}_s) \approx \mathbf{1}_{N \times D_b} \mathbf{V}_b$. Mathematically, we formulate the objective of TIDER as

$$\min_{\mathbf{U}_a, \mathbf{V}} \|(\mathbf{X} - \mathbf{U}_a \mathbf{V}) \odot \mathbf{M}\|^2 + \lambda_t f_t(\mathbf{V}_t) + \lambda_b f_b(\mathbf{V}_b) + \eta_1 \|\mathbf{U}\|^2 + \eta_2 \|\mathbf{V}\|^2, \quad (2)$$

$$\mathbf{U}_a = [\mathbf{U} \mid \mathbf{1}] \in \mathbb{R}^{N \times (D+D_b)}, \quad (3)$$

$$\mathbf{V} = \begin{bmatrix} \mathbf{V}_t + \mathbf{V}_s \\ \mathbf{V}_b \end{bmatrix} \in \mathbb{R}^{(D+D_b) \times T}, \quad (4)$$

where \mathbf{U}_a is the augmented matrix of $\mathbf{U} \in \mathbb{R}^{N \times D}$, f_t and f_b are corresponding constraint functions imposed on \mathbf{V}_t and \mathbf{V}_b . $\eta_1 \|\mathbf{U}\|^2$, $\eta_2 \|\mathbf{V}\|^2$ are used to regularize the magnitude of latent factors, and λ_t , λ_b , η_1 , η_2 are corresponding weights for each term. Once training is completed, we use the learned \mathbf{U} to get \mathbf{U}_a , and the learnt \mathbf{V}_t , \mathbf{V}_s , and \mathbf{V}_b to form \mathbf{V} . \mathbf{U}_a and \mathbf{V} are then used to generate the imputed time series $\tilde{\mathbf{X}}$ as

$$\tilde{X}_{ij} = \begin{cases} X_{ij}, & M_{ij} = 1, \\ (\mathbf{U}_a \mathbf{V})_{ij}, & M_{ij} = 0. \end{cases} \quad (5)$$

B. Trend Representation Matrix

Trend representation matrix \mathbf{V}_t captures the intrinsic trend of time series, whose evolution patterns change gradually and smoothly. We impose a smoothness constraint on \mathbf{V}_t as

$$f_t(\mathbf{V}_t) = \sum_{j=2}^T \|\mathbf{v}_t^j - \mathbf{v}_t^{j-1}\|^2, \quad (6)$$

where \mathbf{v}_t^j is the j -th column of \mathbf{V}_t . Eq. 6 encourages close representations of two adjacent time steps, which will result in a smooth change in data space. We only impose constraints on two consecutive time steps to account for short-term patterns whereas long-term patterns are explained by \mathbf{V}_s . This is in contrast with TRMF [40] which uses one temporal matrix to account for both short-term and long-term patterns.

C. Seasonality Representation Matrix

Real-world time series often demonstrate seasonal patterns [22]. Motivated by this, we model seasonality by parameterizing representation matrix $\mathbf{V}_s \in \mathbb{R}^{D \times T}$ with Fourier basis. We represent each row with a superposition of $2K$ sinusoidal waves ($K \ll T$). Formally, let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{D \times K}$ be two

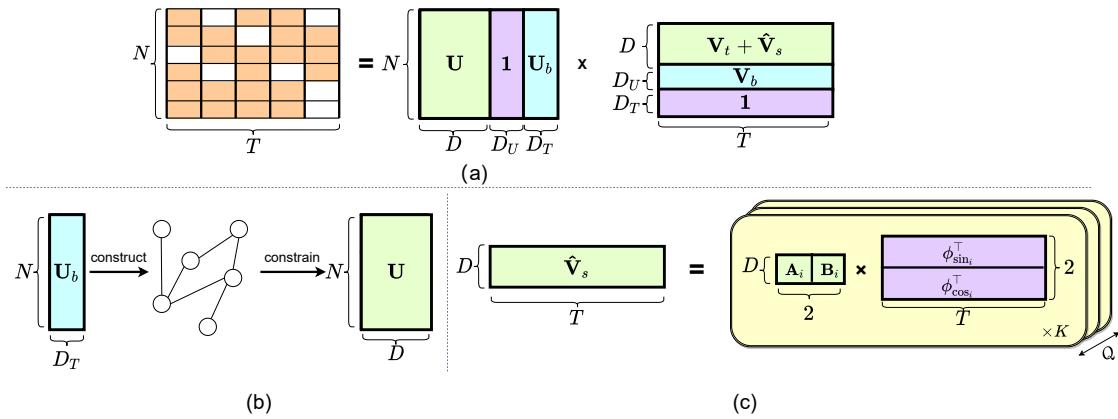


Fig. 2: Architecture of AdaTIDER. Colors of the blocks are detailed in the main paragraph. (a) Main framework. (b) Adaptive learning of the correlations between channels. (c) Multi-period seasonality representation matrix.

learnable coefficient matrices, and $\phi_{\sin}, \phi_{\cos} \in \mathbb{R}^{T \times K}$ be the corresponding Fourier basis matrices, defining as

$$\mathbf{A} = \begin{bmatrix} | & | & & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_K \\ | & | & & | \end{bmatrix}, \quad (7)$$

$$\mathbf{B} = \begin{bmatrix} | & | & & | \\ \mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_K \\ | & | & & | \end{bmatrix}, \quad (8)$$

$$\begin{aligned} \phi_{\sin} &= \begin{bmatrix} | & | & & | \\ \sin(1\omega t) & \sin(2\omega t) & \dots & \sin(K\omega t) \\ | & | & & | \end{bmatrix}, \quad (9) \\ \phi_{\cos} &= \begin{bmatrix} | & | & & | \\ \cos(1\omega t) & \cos(2\omega t) & \dots & \cos(K\omega t) \\ | & | & & | \end{bmatrix}, \quad (10) \end{aligned}$$

where $t = [1, \dots, T]^\top$. For time-series with period P , ω is calculated as $2\pi/P$. Then \mathbf{V}_s can be defined as

$$\mathbf{V}_s = \mathbf{A}\phi_{\sin}^\top + \mathbf{B}\phi_{\cos}^\top. \quad (11)$$

In other words, \mathbf{V}_s is spanned by Fourier basis ϕ_{\sin} and ϕ_{\cos} . The d -th row of \mathbf{V}_s , denoted by $(\mathbf{v}_s^d)^\top$, has the form

$$(\mathbf{v}_s^d)^\top = \sum_{k=1}^K \mathbf{A}_{d,k} \sin(k\omega t)^\top + \sum_{k=1}^K \mathbf{B}_{d,k} \cos(k\omega t)^\top, \quad (12)$$

which is a truncated Fourier series with coefficients $\mathbf{A}_{d,k}$, $\mathbf{B}_{d,k}$. This elaborate design of \mathbf{V}_s imparts meaningful periodic inductive bias, empowering TIDER to capture seasonal patterns more accurately and effectively.

D. Bias Temporal Representation Matrix

\mathbf{V}_t , \mathbf{V}_s discussed so far jointly determine time-series dynamics driven by endogenous factors. However, there are also various external factors (e.g., holidays, sudden accidents) that affect real-world time series at specific time points and influence all channels within short time periods. These external factors are independent of endogenous dynamics captured by \mathbf{V}_t and \mathbf{V}_s and cannot be represented by them. To address these variations, inspired by the idea of user and item bias

in collaborative filtering [43], we propose bias representation matrix $\mathbf{V}_b \in \mathbb{R}^{D_U \times T}$, where the representation of a specific time step is shared by all channels. Specifically, we impose a short-term autoregressive constraint on \mathbf{V}_b in temporal dimension as the impact of external factors usually lasts shortly. Let \mathbf{v}_b^t be the t -th column of \mathbf{V}_b and L be the event influencing duration, we define the constraint function as follows,

$$f_b(\mathbf{V}_b) = \sum_{t=L+1}^T \left\| \mathbf{v}_b^t - \sum_{l=1}^L \mathbf{W}_l \mathbf{v}_b^{t-l} \right\|^2, \quad (13)$$

where $\mathcal{W} = \{\mathbf{W}_l \in \mathbb{R}^{D_U \times D_U} \mid l = 1 \dots L\}$ is a group of learnable parameters. In our setting, D_b and L are small numbers, thus group \mathcal{W} only incurs very few extra parameters.

E. Adaptive Weight for Trend and Seasonality

In Sec. IV-A, we characterize the influence of endogenous factors using additive form $\mathbf{V}_t + \mathbf{V}_s$, which assumes that trend and seasonality contribute equally to endogenous dynamics. However, in practice, their importance vary drastically across data []. To introduce greater flexibility, we adopt a learnable parameter $\alpha \in (0, 1)$ to adaptively adjust the contribution of the two components., leading to a weighted additive form $\alpha\mathbf{V}_t + (1 - \alpha)\mathbf{V}_s$. Eq. 4 then becomes:

$$\mathbf{V} = \begin{bmatrix} \alpha\mathbf{V}_t + (1 - \alpha)\mathbf{V}_s \\ \mathbf{V}_b \end{bmatrix}. \quad (14)$$

V. ADAPTIVE TIME-SERIES IMPUTATION WITH DISENTANGLED TEMPORAL REPRESENTATIONS

A. Overview of the Model

As discussed in Sec. I, TIDER faces specific limitations. It lacks explicit regularization for channel representations, which are crucial for capturing inherent cross-channel correlations. Additionally, its seasonality modeling is restricted to a single predefined period, limiting its ability to handle datasets with complex multi-periodic patterns commonly observed in reality. To address these issues, we introduce **AdaTIDER** (Adaptive TIDER), which builds upon TIDER and is more adaptive in temporal and cross-channel dimensions. Key components

in TIDER, like trend and bias temporal representation, are retained due to their effectiveness in capturing temporal dynamics. AdaTIDER introduces tailored regularization to gain better control over channel representations. Additionally, the construction of seasonality representation matrix is enhanced by adopting a flexible, multi-period framework. These adaptive approaches allows AdaTIDER to effectively capture multiple seasonality patterns, resulting in more accurate imputation.

Fig. 2 presents an overview of AdaTIDER. Original time series \mathbf{X} is divided into missing (white blocks) and observed parts (red blocks). Purple blocks do not retain gradients while blue and green blocks participate in gradient updates. Similarly, AdaTIDER adopts a low-rank matrix factorization framework to decompose \mathbf{X} into \mathbf{U}_a and \mathbf{V}_a . One key advancement is an adaptive adjacency matrix for Laplacian regularization on channel representations (Fig 2-(b)), which allows channel features to incorporate both channel-specific patterns and inter-channel relationships. Additionally, AdaTIDER incorporates a multi-period seasonality representation (Fig 2-(c)), which enables independent modeling of different periodicities, allowing for a more comprehensive understanding of various periodic factors. Formally, AdaTIDER's objective is:

$$\begin{aligned} \min_{\mathbf{U}_a, \mathbf{V}_a} & \|(\mathbf{X} - \mathbf{U}_a \mathbf{V}_a) \odot \mathbf{M}\|^2 + \lambda_t f_t(\mathbf{V}_t) + \lambda_b f_b(\mathbf{V}_b) \\ & + \lambda_u f_u(\mathbf{U}) + \eta_1 \|\mathbf{U}_a\|^2 + \eta_2 \|\mathbf{V}_a\|^2, \end{aligned} \quad (15)$$

$$\mathbf{U}_a = [\mathbf{U} \mid \mathbf{1} \mid \mathbf{U}_b] \in \mathbb{R}^{N \times (D+D_U+D_T)}, \quad (16)$$

$$\mathbf{V}_a = \begin{bmatrix} \alpha \mathbf{V}_t + (1-\alpha) \hat{\mathbf{V}}_s \\ \mathbf{V}_b \\ \mathbf{1} \end{bmatrix} \in \mathbb{R}^{(D+D_U+D_T) \times T}, \quad (17)$$

where \mathbf{V}_t , \mathbf{V}_b , f_t , f_b , f_u , $\eta_1 \|\mathbf{U}_a\|^2$, and $\eta_2 \|\mathbf{V}_a\|^2$ are identical to those in TIDER. The difference lies in $\hat{\mathbf{V}}_s$, \mathbf{U}_a and \mathbf{U}_b , representing the novel advancements of AdaTIDER, which will be elaborated in subsequent sections. Upon the conclusion of the training phase, we use the learned \mathbf{U} and \mathbf{U}_b to obtain \mathbf{U}_a , and the learnt \mathbf{V}_t , \mathbf{V}_s , and \mathbf{V}_b to form \mathbf{V}_a . The generation process of imputed time series $\tilde{\mathbf{X}}$ is similar as Eq. 5.

B. Adaptive Cross-channel Correlation

In multivariate time series, different channels often exhibit complex dependencies, and accurately modeling their relationships is crucial for effective analysis. Previous approaches [44] rely on adjacency matrices based on spatial proximity to approximate their dependencies. However, this approach faces notable challenges because spatial relationships are only defined in geographical contexts, making it unsuitable for abstract spaces like financial market. Additionally, spatial distance does not always reflect true correlations, as strongly correlated time-series may be geographically distant, while adjacent time-series may exhibit distinct patterns. [45], [46] employ adaptive adjacency matrices computed from learned representations. These matrices are then integrated into graph neural networks. However, directly applying such adaptive matrices risks a "bootstrapping" issue, where using channel representations to constrain themselves will weaken and distort regularization effect [29], [30]. AdaTIDER innovatively

addresses it by decomposing \mathbf{U}_a into a residual term \mathbf{U}_b and a non-residual term \mathbf{U} . \mathbf{U} contains channel-specific information, whereas \mathbf{U}_b is used to impose regularization and produce the adaptive adjacency matrix \mathbf{G} as:

$$\mathbf{G} = \text{softmax}(\text{ReLU}(\mathbf{U}_b \mathbf{U}_b^\top)), \quad (18)$$

$$\hat{\mathbf{G}} = \mathbf{D}^{-1/2} \mathbf{G} \mathbf{D}^{-1/2}, \quad (19)$$

$$\hat{\mathbf{L}} = \hat{\mathbf{D}} - \hat{\mathbf{G}}, \quad (20)$$

$$f_u(\mathbf{U}) = \text{tr}(\mathbf{U}^\top \hat{\mathbf{L}} \mathbf{U}), \quad (21)$$

where $\mathbf{D}, \hat{\mathbf{D}}$ are degree matrices of $\mathbf{G}, \hat{\mathbf{G}}$ separately and the operator tr denotes the matrix trace calculation operation. The Laplacian regularization—Eq. 21—can be further expanded as

$$f_u(\mathbf{U}) = \text{tr}(\mathbf{U}^\top \hat{\mathbf{L}} \mathbf{U}) = \sum_{i=1}^N \sum_{j=1}^N G_{ij} \|\mathbf{u}_i^\top - \mathbf{u}_j^\top\|_2^2, \quad (22)$$

where G_{ij} is the ij -th element of \mathbf{G} , and \mathbf{u}_i^\top and \mathbf{u}_j^\top are rows of \mathbf{U} . G_{ij} aims to minimize the gap between nearby points according to adjacency matrix \mathbf{G} , so as to promote spatial smoothness and penalize variations between nearby points.

The adaptive adjacency matrix \mathbf{G} in AdaTIDER has distinct advantages over adjacency matrices derived solely from spatial proximity, for \mathbf{G} captures context-specific inter-channel dependencies that may not be driven by physical proximity. This adaptability enables the model to identify meaningful relationships when spatial distance is not an accurate measure of correlation. Furthermore, \mathbf{G} is applicable to time-series data without explicit spatial information. In such domains, the learned adjacency matrix can be interpreted as capturing correlations in a semantic space, reflecting latent interdependencies between channels. Overall, the adaptive adjacency matrix offers AdaTIDER enhanced power and flexibility in capturing inter-channel dependencies, making it applicable to a wide range of data types, including both geographically meaningful and non-geographical time series.

C. Adaptive Multi-period Seasonality Representation Matrix

In real-world time series, recurring patterns within specific time intervals, known as seasonality, are commonly observed. TIDER models this through \mathbf{V}_s , but this approach is limited to capturing periodicity of a single pre-specified period. In practice, time series often exhibit multiple periodic patterns simultaneously [22], [28]. For example, traffic flow data may reflect daily and weekly patterns due to commuting activities, while solar power generation data may show periodicities influenced by seasonal changes and weather conditions. This makes single-period modeling, as used in TIDER, insufficient. To address this, AdaTIDER introduces an adaptive approach to capture multiple inherent periods. Given time series $\mathbf{X} \in \mathbb{R}^{N \times T}$, we perform fast Fourier transform on each channel, calculate the average amplitude of each frequency across all these channels, and identify Q frequencies with the highest

average amplitudes along with their corresponding periods. Mathematically, this can be formatted as,

$$\text{AMP} = \text{Mean}(\text{Amp}(\text{FFT}(\mathbf{X}))), \quad (23)$$

$$\{\omega_1, \dots, \omega_Q\} = \arg \underset{\omega}{\text{TopQ}}(\text{AMP}), \quad (24)$$

$$\{p_1, \dots, p_Q\} = \{\lfloor \frac{T}{\omega_1} \rfloor, \dots, \lfloor \frac{T}{\omega_Q} \rfloor\}. \quad (25)$$

For each period p_i , corresponding \mathbf{A}_i and \mathbf{B}_i are learned, resulting in a single-period seasonality matrix \mathbf{V}_{s_i} . These single-period seasonality matrices are then summed to obtain the multi-periods seasonality representation $\hat{\mathbf{V}}_s$:

$$\mathbf{V}_{s_i} = \mathbf{A}_i \phi_{\sin_i}^\top + \mathbf{B}_i \phi_{\cos_i}^\top, \quad (26)$$

$$\hat{\mathbf{V}}_s = \sum_{i=1, \dots, Q} \mathbf{V}_{s_i}. \quad (27)$$

D. Computational Complexity and Scalability Analysis

Training TIDER primarily involves $O(NTD)$ matrix operations, with $O(D(N+T))$ parameters. Its structure supports efficient parallelization and batching along channel dimension, ensuring scalability for large datasets. AdaTIDER further incorporates adaptive adjacency matrix and multi-period seasonality representation. The adjacency matrix results in an additional complexity term of $O(N(D_T)^2)$, while multi-period seasonality introduces another term of $O(NQD)$. Since D_T and Q are significantly smaller than N and T , these terms are negligible compared to the main complexity term. Combining these factors, AdaTIDER's total time complexity becomes $O(NTD + N(D_T)^2 + NQD)$, which is asymptotically equivalent to $O(NTD)$. Similarly, the space complexity extends to $O(D(N+T) + D_T N + QD)$, still dominated by the original term $O(D(N+T))$. As a result, AdaTIDER remains on the same complexity level as TIDER. In contrast, RNN-based methods like BRITS sequentially update hidden states, struggling to parallelize computations over long time series, and thus exhibit significantly longer training times. Diffusion-based methods like CSDI require multiple diffusion and sampling steps, and rely on large neural networks to capture complex distributions, resulting in substantial time and resource consumption.

VI. EXPERIMENTS

We evaluate TIDER and AdaTIDER on imputation accuracy, scalability, interpretability, disentanglement, and stability. Since TIDER and AdaTIDER demonstrate similar scalability and stability, we present AdaTIDER's results to represent both in these aspects. However, in terms of interpretability and disentanglement, AdaTIDER outperforms TIDER, as evidenced by multiple case studies. The code of TIDER and AdaTIDER is available at <https://github.com/liuwj2000/TIDER>.

A. Experimental Setup

Baseline Methods We select statistical models like SimpleMean [14], KNN [13], and MICE [47], MF-based methods like MF [48], MF+L2 [48], SoftImpute [49], LCR [50], and TRMF [40], and deep learning approaches

including BRITS [19], GAIN [33], NAOMI [16], SingleRes [16], SAITS [51], CSDI [17], TimeCIB [52], SADI [21], PoGeVON [34], PITC [53], MTSCI [54], and TimesNet [38]. We also introduced a variant of TIDER called TIDER (no W), which excludes the learnable parameter α , allowing us to assess the effectiveness of this weight.

Datasets We use three real-world datasets, representing different scenarios: small, large, and long time series. These datasets enable us to demonstrate the robustness and effectiveness of our models across diverse conditions. Details are as follows:

- **Guangzhou**¹. It consists of traffic speed measured every 10 minutes from 214 anonymous urban road segments in Guangzhou, China. We select the last 5000 minutes, resulting in a time series matrix of size 214×500 .
- **Solar-Energy**². It comprises records of solar power production sampled every 10 minutes from 137 PV plants in Alabama, USA, resulting in a matrix of size 137×52560 .
- **Westminster**³. It includes hourly averaged speed measurements of road segments in Westminster, collected by Uber Movement in January 2020, with a size of 7489×744 .

Metrics We adopt Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) for evaluation.

Training Setup Data is randomly split into training, validation, and testing sets. Given a missing rate r , the ratio of training/validation/testing is $(0.9 - r)/0.1/r$. Each model is evaluated over 7 rounds on each dataset, with the imputed results averaged across these runs. All experiments are conducted on a Linux workstation with a 32GB Tesla V100 GPU.

B. Imputation Accuracy Comparison

Tab. I and II display the imputation accuracy with varying missing rates r . "OOM" refers to Out Of Memory. Deep-learning methods generally outperform other baseline methods. Our methods surpass all baseline methods, demonstrating their effectiveness in accurate imputation. A notable advantage of our models is their superiority in the solar-energy dataset, with a long time span of 52560. Many deep learning methods encounter OOM errors when executed on a 32GB GPU. In contrast, our models remain applicable and achieves better imputation performance. This can be attributed to their ability to disentangle multiple factors, allowing the representations to capture and leverage underlying patterns and dependencies.

All MF-based models are well-suited for long time series. Their main difference lies in their constraints imposed on \mathbf{U} and \mathbf{V} . LCR stands out among MF-based baselines as it utilizes Laplacian convolutional representations to incorporate spatial relationships into imputation process. However, LCR does not explicitly model temporal dynamics, which limits its ability to capture temporal patterns. In contrast, TIDER employs disentangled features such as trend, seasonality, and local bias to reconstruct temporal patterns. This approach

¹<https://zenodo.org/record/1205229>

²<https://www.nrel.gov/grid/solar-power-data.html>

³This dataset is no longer open to public.

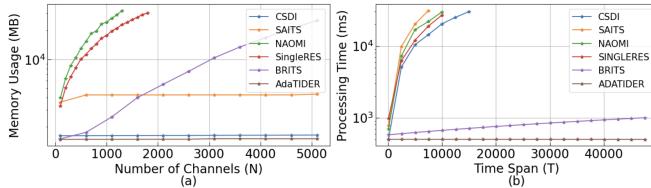


Fig. 3: Different methods' memory usage with logarithmic scaling on y-axis: (a) varied N and fixed T on Westminster; (b) varied T and fixed N on the Solar-Energy.

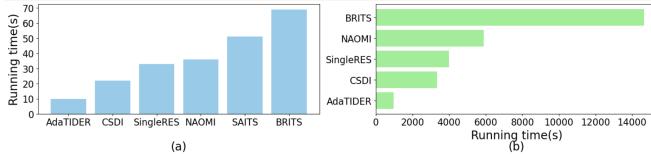


Fig. 4: (a) Average running time for 100 epochs on a 100×100 matrix. (b) Total training time on Westminster-Central.

leverages explanatory factors, resulting in significantly improved imputation results. Furthermore, TIDER performs better than TIDER (no W), validating that trend and seasonality components may not contribute equally to global dynamics. AdaTIDER, which further considers adaptive cross-channel correlations and adaptive multi-period seasonality patterns, outperforms TIDER in most cases. However, there is an exception in the Westminster dataset with missing rate of 0.2. AdaTIDER performs slightly worse than TIDER, and its performance is also worse than that in the same dataset with a larger missing rate. We hypothesize that this may be due to a combination of factors, including an extremely large number of channels (N), a small missing rate (r), and a short time span (T). With such factors combined, AdaTIDER may inadvertently learn many outliers and suspicious cross-channel correlations, leading to overfitting. Furthermore, the small time span of the Westminster dataset may also limit AdaTIDER's ability to effectively capture recurrent patterns within the time series, such as the weekly pattern in this traffic dataset.

To validate our hypothesis, we conduct an experiment on modified version of Westminster dataset, named Westminster-Central. We select a subset of road segments (as shown in Fig. 5) and the time length is extended to six months, resulting in a matrix of size 284×4273 . Imputation results in Tab. III, confirm our hypothesis: as number of channels decreases and time span increases, AdaTIDER mitigates the overfitting issues. Fewer channels alleviate the impact of suspicious cross-channel correlations, while a longer time span helps to better capture recurrent patterns. Furthermore, most methods' performance declined. This is because fewer channels limit the inter-variable interaction information, while an extended time span introduces greater complexity into time series dynamics. These factors make the imputation task more challenging, causing most methods to struggle. AdaTIDER still outperforms all other methods. This can be attributed to its flexible dynamic disentanglement framework, which effectively models trends, periodicity, and local deviations, enabling AdaTIDER to maintain robust performance even when faced with fewer channels and longer time spans.

C. Scalability Analysis

We evaluate the scalability of different methods in memory usage and training time. Fig. 3-(a) and 3-(b) present memory footprints against number of channels (N) and time span (T). As shown in Fig. 3-(a), NAOMI, SingleRes, and BRITS exhibit rapid memory growth. In contrast, AdaTIDER requires substantially less memory. Similarly, Fig. 3-(b) illustrates that NAOMI, SingleRes, and SAITS experience fast memory growth. Once again, AdaTIDER requires the lowest memory. Notably, in Fig. 3(a), CSDI's memory usage is slightly higher than AdaTIDER's, while in Fig. 3(b), BRITS' memory usage lies slightly above AdaTIDER's. Fig. 4(a) further shows the running time of different methods when processing a 100×100 matrix over 100 epochs, while Fig. 4(b) presents their total training time on Westminster-Central dataset. The results clearly show AdaTIDER's superior efficiency.

D. Ablation Analysis for Temporal Disentanglements

To assess the effectiveness of disentangled temporal representations in AdaTIDER, we conduct an ablation study on the Guangzhou dataset, where we remove each representation matrix individually while retaining the remaining parts. Additionally, we compare the imputation results of using \mathbf{V}_s with using $\hat{\mathbf{V}}_s$. The ablative results are presented in Tab. IV. We observe a decrease in performance when any representation matrix is removed. This demonstrates that each representation plays a crucial role. Among these components, \mathbf{V}_t is the most critical, with its removal causing the largest decline in RMSE, MAE, and MAPE metrics. This highlights its importance in capturing long-term temporal dynamics and ensuring stable imputation. $\hat{\mathbf{V}}_s$ also demonstrates significant importance. Replacing \mathbf{V}_s with $\hat{\mathbf{V}}_s$ consistently improves the model's performance, showcasing its ability to model complex and adaptive periodic patterns. Lastly, V_b contributes to the model's robustness by capturing short-term fluctuations.

E. Ablation Analysis for Adaptive Adjacency Matrix Modules

To assess the effectiveness of each computational module in calculating the adaptive adjacency matrix, we compare different variations of AdaTIDER to evaluate their performance in imputing missing values. The variations we explored include:

- Geo-Ada. It obtains the adjacency matrix based on the geographical adjacency of road segments.
- No- \mathbf{U}_b . No- \mathbf{U}_b directly uses \mathbf{U} to calculate adaptive adjacency matrix and constrains \mathbf{U}_a itself using this adaptive adjacency. There is no \mathbf{U}_b in this case.

Tab V reveals that none of the variations can match the performance of AdaTIDER. There are clear reasons behind this observation. When using geographical adjacency, the model is forced to use inappropriate connections while ignoring hidden correlations that are essential for accurate imputation. On the other hand, No- \mathbf{U}_b suffers from the "bootstrapping" problem, where channels rely on their own information to constrain themselves, leading to self-referential constraints that may not be optimal for regularization. As a result, the model may not effectively capture true underlying relationships between channels, thus affecting the imputation performance.

TABLE I: Imputation accuracy of different methods with missing rate $r = 0.2$

Method ($r = 0.2$)	Guangzhou			Solar-energy			Westminster		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
SimpleMean	10.24	8.023	0.396	3.352	2.433	4.527	5.233	4.247	0.410
KNN	6.540	4.731	0.242	2.701	1.748	2.324	2.249	1.611	0.147
MF	5.642	4.301	0.188	3.769	2.447	4.243	2.275	1.641	0.137
MF+L2	5.400	4.074	0.177	2.761	1.918	2.760	2.109	1.523	0.134
SoftImpute	5.876	4.465	0.217	2.692	1.903	3.051	2.347	1.645	0.141
MICE	5.097	3.859	0.161	2.647	1.763	2.372	2.638	2.014	0.169
TRMF	4.563	3.407	0.148	3.212	2.010	2.937	1.903	1.370	0.119
BRITS	4.416	3.003	0.139	2.617	1.861	2.677	2.154	1.488	0.136
GAIN	4.976	3.451	0.154	2.803	1.917	2.501	1.947	1.403	0.121
SAITS	4.407	3.025	0.140	OOD	OOD	OOD	1.893	1.366	0.119
CSDI	4.301	2.991	0.135	OOD	OOD	OOD	1.886	1.361	0.116
TimeCIB	4.698	3.371	0.152	OOD	OOD	OOD	1.891	1.357	0.123
SADI	4.933	3.394	0.155	OOD	OOD	OOD	2.062	1.447	0.129
PoGeVON	3.854	2.765	0.117	1.608	1.005	2.292	OOD	OOD	OOD
PITC	5.262	4.297	0.188	OOD	OOD	OOD	2.207	1.822	0.177
LCR	4.601	3.458	0.136	1.837	1.189	2.382	2.070	1.581	0.176
MTSCI	3.739	2.693	0.114	OOD	OOD	OOD	1.943	1.392	0.117
NAOMI	5.173	4.013	0.167	OOD	OOD	OOD	OOD	OOD	OOD
SingleRes	4.997	3.979	0.172	OOD	OOD	OOD	OOD	OOD	OOD
TimesNet	5.211	3.620	0.139	1.514	0.753	2.193	2.036	1.475	0.139
TIDER (no W)	4.431	3.229	0.142	1.872	0.893	2.522	1.981	1.426	0.127
TIDER	4.168	3.098	0.132	1.676	0.874	2.227	1.867	1.354	0.115
AdaTIDER	3.553	2.586	0.106	1.029	0.515	2.096	1.885	1.361	0.114
Improvement(%)	4.975	3.973	7.018	32.03	31.61	4.423	3.911	2.730	2.564

TABLE II: Imputation accuracy of different methods with missing rate $r = 0.4$

Method ($r = 0.4$)	Guangzhou			Solar-energy			Westminster		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
SimpleMean	9.949	7.901	0.384	3.259	2.412	4.889	5.243	4.259	0.407
KNN	6.712	4.955	0.252	2.724	1.835	2.874	2.391	1.714	0.159
MF	7.671	5.844	0.246	4.111	2.887	5.851	2.476	1.807	0.153
MF+L2	7.324	5.553	0.235	2.617	1.80	2.977	2.157	1.593	0.134
SoftImpute	5.811	4.465	0.209	2.536	1.849	3.311	2.333	1.637	0.145
MICE	7.119	5.417	0.235	2.756	1.823	3.517	2.755	2.113	0.175
TRMF	6.119	4.617	0.198	3.407	2.199	3.586	2.014	1.453	0.126
BRITS	4.874	3.335	0.158	2.842	1.985	3.146	2.180	1.527	0.138
GAIN	5.550	3.671	0.192	2.639	1.883	3.117	2.337	1.711	0.145
SAITS	4.839	3.391	0.159	OOD	OOD	OOD	1.998	1.453	0.129
CSDI	4.813	3.202	0.157	OOD	OOD	OOD	1.982	1.437	0.124
TimeCIB	4.927	3.399	0.162	OOD	OOD	OOD	1.993	1.472	0.135
SADI	5.001	3.414	0.160	OOD	OOD	OOD	2.062	1.447	0.129
PoGeVON	4.158	2.907	0.137	1.898	1.029	2.421	OOD	OOD	OOD
PITC	6.892	4.976	0.203	OOD	OOD	OOD	2.512	2.013	0.163
LCR	4.923	3.724	0.139	1.916	1.272	2.805	2.162	1.697	0.183
MTSCI	4.528	3.336	0.133	OOD	OOD	OOD	1.924	1.403	0.119
NAOMI	5.986	4.543	0.222	OOD	OOD	OOD	OOD	OOD	OOD
SingleRes	6.051	4.705	0.252	OOD	OOD	OOD	OOD	OOD	OOD
TimesNet	5.528	3.926	0.148	1.998	1.052	2.956	2.115	1.608	0.143
TIDER (no W)	4.708	3.469	0.155	1.697	0.878	3.152	2.013	1.466	0.125
TIDER	4.764	3.527	0.152	1.679	0.838	2.735	1.959	1.422	0.121
AdaTIDER	3.996	2.880	0.122	1.050	0.522	2.071	1.717	1.247	0.107
Improvement(%)	3.896	0.929	8.271	44.68	97.12	14.46	10.76	11.12	10.08

TABLE III: Imputation accuracy of different methods for Westminster-Central

Method	Westminster-Central ($r = 0.2$)			Westminster-Central ($r = 0.4$)		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE
TRMF	2.485	1.802	0.146	2.681	1.907	0.155
BRITS	2.743	2.045	0.133	3.215	2.378	0.134
SAITS	OOD	OOD	OOD	OOD	OOD	OOD
CSDI	2.180	1.571	0.129	2.194	1.603	0.133
TimeCIB	2.093	1.547	0.124	2.117	1.552	0.133
SADI	2.266	1.639	0.130	2.512	1.704	0.141
PoGeVON	2.032	1.432	0.122	2.175	1.542	0.128
PITC	2.607	2.092	0.186	2.864	2.344	0.197
LCR	2.109	1.627	0.185	2.233	1.709	0.188
MTSCI	2.091	1.542	0.124	2.168	1.597	0.133
TimesNet	2.393	1.568	0.147	2.323	1.563	0.168
TIDER	2.173	1.567	0.128	2.159	1.600	0.131
AdaTIDER	1.953	1.419	0.116	2.038	1.488	0.122
Improvement(%)	3.887	0.908	4.918	3.731	3.502	4.688

TABLE IV: Ablation Analysis for temporal components of AdaTIDER on Guangzhou dataset ($r = 0.2$).

Name	RMSE	MAE	MAPE
$\mathbf{V}_t + \mathbf{V}_b$	3.616	2.652	0.114
$\hat{\mathbf{V}}_s + \mathbf{V}_b$	4.405	3.279	0.143
$\mathbf{V}_s + \mathbf{V}_b$	6.963	5.281	0.250
$\mathbf{V}_t + \hat{\mathbf{V}}_s$	3.804	2.805	0.116
$\mathbf{V}_t + \mathbf{V}_s$	3.900	2.848	0.118
TIDER	4.168	3.098	0.132
AdaTIDER	3.553	2.586	0.106

TABLE V: Ablation Analysis for channel-correlation components of AdaTIDER on Westminster-Central dataset ($r = 0.2$).

Name	RMSE	MAE	MAPE
Geo-Ada	2.098	1.542	0.129
no- U_r	2.045	1.505	0.123
AdaTIDER	1.953	1.419	0.116

F. Case Study of Adaptive adjacency matrix

We visualize the adjacency matrix learned by AdaTIDER on Westminster-Central dataset. Specifically, we select a road segment colored in blue (Fig. 5). By applying suitable thresholds, we identify other related road segments based on their association strengths, represented in red and black. The results reveal that spatial adjacency plays a crucial role, as nearby segments (Rd1–Rd5, Bk1) show strong correlations, indicating likely traffic flow interactions with the blue segment. However, spatial proximity does not guarantee strong associations, as Y1 lacks a clear connection. Interestingly, distant segments like Rd6 and Rd8 exhibit notable associations despite their spatial separation. These correlations likely arise from shared functional roles, such as being part of bus routes or having traffic dependencies along upstream and downstream segments.

Fig. 6 further contrasts the geographical adjacency matrix (Fig. 6(a)) with the adaptive adjacency matrix (Fig. 6(b)). Due to the large number of road segments, we focus on the segments highlighted in Fig. 5. The geographical adjacency matrix reflects direct spatial connections based on physical proximity. However, it overlooks complex dynamic interactions. In contrast, AdaTIDER’s learned adjacency matrix reveals a broader range of associations, capturing both explicit spatial adjacency and implicit relationships.

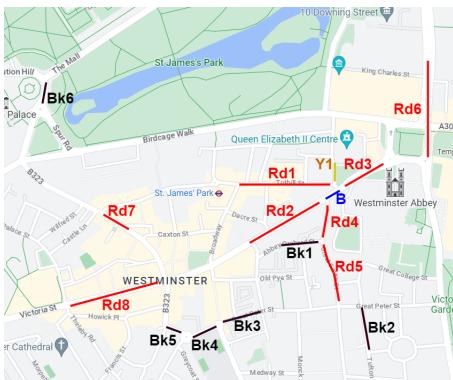


Fig. 5: Visualization of the learned adaptive adjacency matrix.

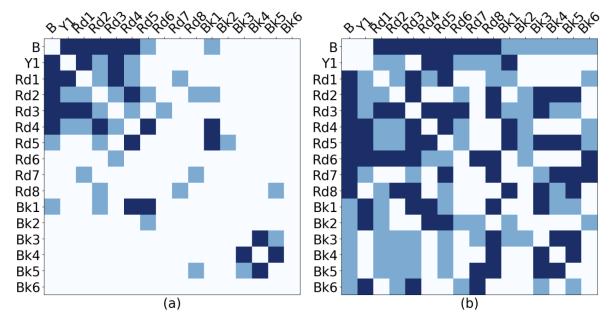


Fig. 6: Comparison of different adjacency matrices for road segments in Fig. 5. (a) Geographical adjacency matrix (b) Adaptive adjacency matrix learned by AdaTIDER.

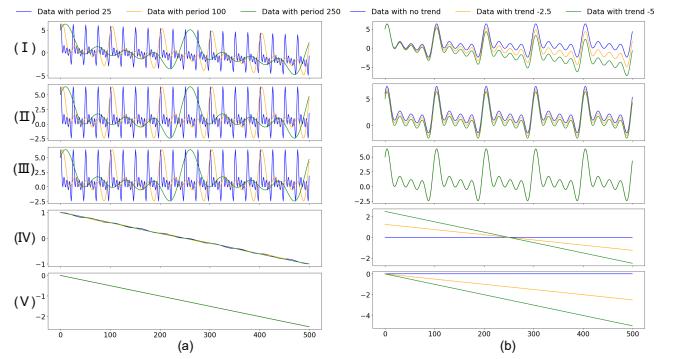


Fig. 7: Disentanglement validation on synthetic dataset. I is the raw time series. II and IV show trend and seasonality components learned. III and V depict the ground-truth trend and seasonality components from the raw time series.

G. Case Studies of Temporal Disentanglement

We demonstrate the disentanglement and interpretability of AdaTIDER’s learned temporal representations through synthetic time series. Two sets of time series are created: one with the same trend but different seasonality, and another with the same seasonality but different trend. Fig. 7 visualizes the ground-truth factors alongside the corresponding learned

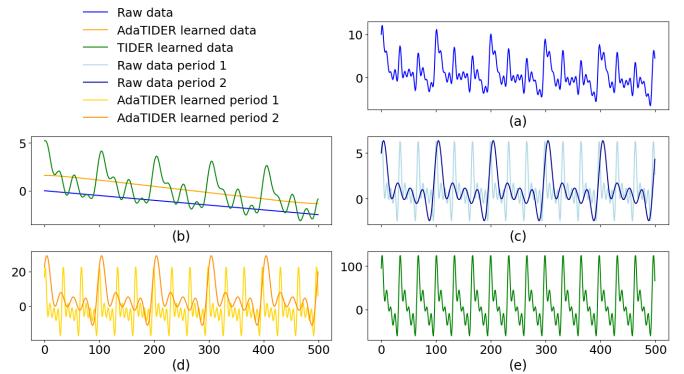


Fig. 8: Disentanglement comparison between TIDER and AdaTIDER. (a) is the raw time series. (b) shows the different trends. (c),(d), and (e) are seasonality components.

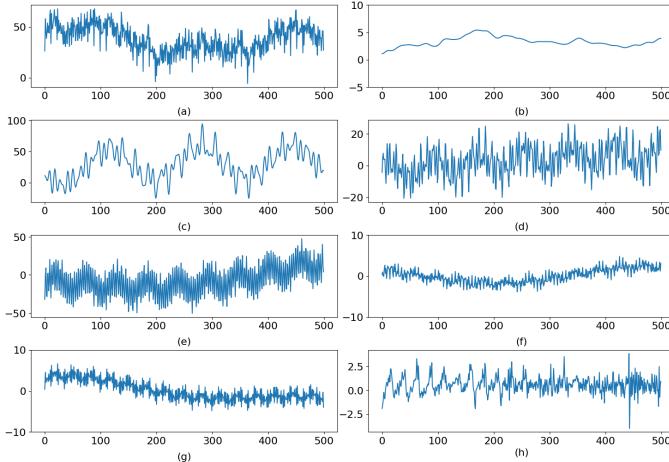


Fig. 9: Learned Temporal components of AdaTIDER from Westminster-Central. (a) Raw Data, (b)Learned trend, (c)–(g) Learned Seasonality, (h) Learned local Bias.

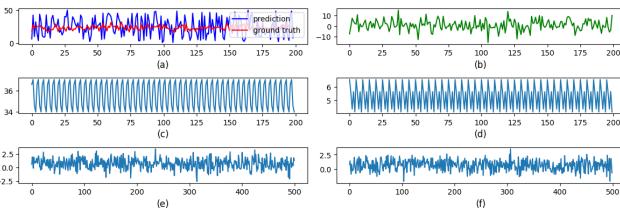


Fig. 10: Failure case study of AdaTIDER from a randomly-created time series. (a) Raw Data and learned time series. (b) Learned trend. (c)–(d) Learned Seasonality. (e)–(f) Local Bias.

representations. It's clear that the learned representations closely match the ground-truth factors. This validates the interpretability and disentanglement of AdaTIDER.

AdaTIDER's superiority over TIDER is evident in Fig. 8. In multi-period scenarios, AdaTIDER effectively separate trend from seasonality and disentangle multiple periodic patterns. It closely aligns with the true trend (Fig. 8 (b)) and captures multiple periodic patterns (Fig. 8 (c), (d)). In contrast, TIDER is limited to single-period modeling, failing to capture multi-frequency periodic patterns or disentangle trend and periodicity. It may even misattribute parts of the seasonality as trends.

Fig. 9 visualizes the temporal components AdaTIDER learned. To show its robustness, Q is set to 5, larger than the actual number of periods. The results reveal that AdaTIDER successfully captures major periodic patterns, including weekly (168), daily (24), and short-term fluctuations (8), while the amplitudes of the fourth and fifth components is minimal. This underscores AdaTIDER's ability to adaptively identify and disentangle meaningful temporal features.

For completeness, we explore a failure case where AdaTIDER is applied to a time series without trend and seasonality (Fig 10). Although most real-world time series contain trends and periodicity [22], [28], certain datasets, such as financial or stock market data, may lack such patterns. Using a randomly initialized time series without trends and periodic features, we find that AdaTIDER struggles to extract

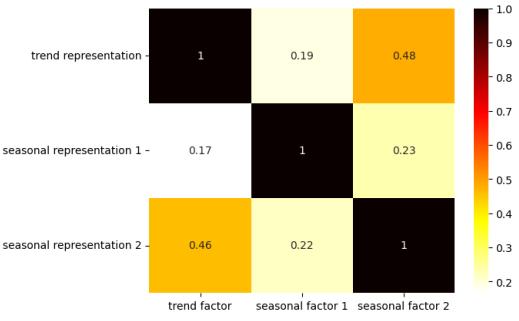


Fig. 11: Maximal Information Coefficient (MIC) matrix for time series factors and representations AdaTIDER has learnt.

meaningful components, resulting in significant deviations in the decomposed trend and seasonality terms. Addressing the imputation of such datasets remains an open challenge.

H. Quantitative Evaluation of Temporal Disentanglements

To evaluate AdaTIDER's disentanglement, we employed Maximal Information Coefficient (MIC) to measure relationships between original time series factors and representations learned by AdaTIDER. MIC is a non-parametric measure that captures both linear and nonlinear associations without assuming a specific functional form. A higher MIC value signifies a stronger association. Formally, MIC can be computed as:

$$\text{MIC}[x; y] = \max_{|X||Y| < B} \frac{\text{I}[x; y]}{\log_2(\min(|X|, |Y|))}, \quad (28)$$

where x and y are two variables. $|X|$ and $|Y|$ represent the number of grids in x-axis and y-axis, and B is a predefined threshold that limits the number of grids. $\text{I}[x; y]$ is the mutual information between x and y . We employ the same time series as in Fig. 8. MIC values are computed between factors and representations, and the results are organized into a 3×3 matrix, as visualized in Fig. 11. It's observed that MIC between trend factor and trend representation, as well as between the two seasonality factors and their respective seasonality representations, exhibit substantial values. Conversely, MIC between different types of factors and representations show small values. This suggests that AdaTIDER effectively disentangles the learned components, as evidenced by the strong associations between factors and corresponding representations.

I. Hyperparameter Sensitivity

We assess the impact of key hyperparameters (D , D_T , D_U , Q) on AdaTIDER using Guangzhou dataset, as shown in Fig. 12. AdaTIDER maintains stable performance across various hyperparameter settings, attributed to its model design, including an adaptive adjacency matrix and disentangled temporal representations. Additionally, AdaTIDER employs a feature separation mechanism, enabling independent modeling of different time-series components, including periodicity, trends, and local biases. This design mitigates the impact of individual hyper-parameters on overall model performance.

Specifically, Q has small impact within a reasonable range. Even if Q slightly exceeds the actual number of periodic

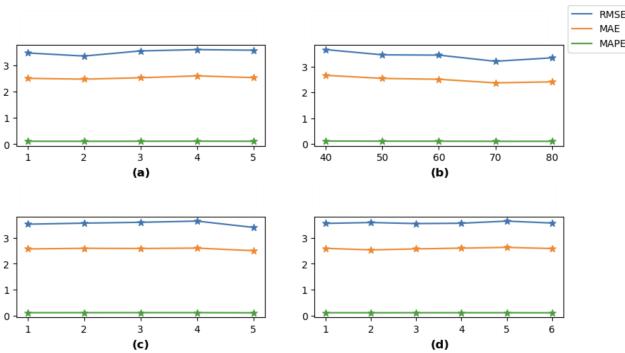


Fig. 12: Performance of AdaTIDER against different hyperparameters.

patterns, the parameters associated with these extra periods (A and B) will converge toward zero, minimizing their influence (as Fig 9(f), (g) shows). However, setting Q excessively high (e.g., above 20 in Guangzhou) may cause these small-amplitude periodic components to accumulate, reducing imputation accuracy. Similarly, D traditionally plays a critical role, but its impact in AdaTIDER is mitigated by the disentangled modeling of features. Variations in D show limited influence on the model's performance as other mechanisms adaptively capture the essential dynamics. These findings highlight AdaTIDER's robustness to hyperparameter variations, providing reliable performance while minimizing the burden of hyperparameter selection. In practice, we recommend setting hyperparameters as follows. Starting with a small Q (e.g., 2 or 3) and gradually increasing it until no further performance improvement is observed, while D depends on the dataset's complexity. We suggest using $D = 50$ for simpler datasets and gradually increasing it to 100 for more complex scenarios.

VII. CONCLUSION

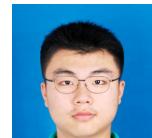
In this paper, we introduce two innovative multivariate time series imputation methods, TIDER and AdaTIDER, which are the first models to leverage low-rank matrix factorization with semantically meaningful disentangled representations to account for various factors influencing time series. TIDER employs neighboring-smoothness, Fourier basis-based periodicity, and time bias representations to explain trend, capture periodic patterns, and identify external factors, respectively. AdaTIDER builds upon TIDER by incorporating adaptive cross-channel correlations and adaptive multi-period seasonality matrix, leading to improved effectiveness in capturing hidden characteristics and enhanced imputation accuracy. Experimental results demonstrate the superiority of TIDER and AdaTIDER in terms of imputation accuracy, scalability, explainability, and robustness, especially in handling long time series. However, in scenarios where the underlying data lack distinct trends or seasonal patterns, the trend and seasonality modules fail to extract meaningful structures, leading to imputation performance degradation. Our failure case experiment (Fig 10) empirically illustrates this phenomenon. Consequently, TIDER and AdaTIDER are better suited for time series with trends and seasonal patterns,

such as traffic flow, meteorological, and energy-consumption data. Moving forward, an open research challenge lies in developing specific flexible feature representations within a disentangled framework to handle time series without clear trends or seasonality, such as those in finance and economics. Additionally, we propose other potential future research directions, including applying TIDER and AdaTIDER to other time-series analysis tasks, accommodating tensor-shaped time series, exploring their applicability in blackout time series imputation, and integrating contrastive learning to further enhance their capabilities and performance. By addressing these directions, we can advance the field of multivariate time series analysis and imputation, enhancing the applicability and effectiveness of TIDER and AdaTIDER.

REFERENCES

- [1] S. Liu, G. Song, and W. Huang, "Real-time transportation prediction correction using reconstruction error in deep learning," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2020.
- [2] S. Tonekaboni, D. Eytan, and A. Goldenberg, "Unsupervised representation learning for time series with temporal neighborhood coding," in *9th International Conference on Learning Representations (ICLR)*, 2021.
- [3] Z. Karevan and J. A. Suykens, "Transductive lstm for time-series prediction: An application to weather forecasting," *Neural Networks*, vol. 125, pp. 1–9, 2020.
- [4] M. Kaur, S. Panwar, A. Joshi, and K. Gupta, "Residential electricity demand prediction using machine learning," in *Proceedings of the International Semantic Intelligence Conference (ISIC)*, 2021.
- [5] S. Zeng, F. Graf, C. Hofer, and R. Kwitt, "Topological attention for time series forecasting," in *Annual Conference on Neural Information Processing Systems (NeurIPS), December 6-14, 2021, virtual*.
- [6] G. Li, B. Choi, J. Xu, S. S. Bhowmick, D. N.-y. Mah, and G. L. Wong, "Ips: Instance profile for shapelet discovery for time series classification," in *IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia, May 9-12, 2022*.
- [7] A. Laddha, S. Gautam, G. P. Meyer, and C. K. Wellington, "Rv-fusenet: Range view based fusion of time-series lidar data for joint 3d object detection and motion forecasting," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 7060–7066.
- [8] K. Yan, C. Long, H. Wu, and Z. Wen, "Multi-resolution expansion of analysis in time-frequency domain for time series forecasting," *IEEE Transactions on Knowledge and Data Engineering*, 2024.
- [9] H. Xue and F. D. Salim, "Promptcast: A new prompt-based learning paradigm for time series forecasting," *IEEE Transactions on Knowledge and Data Engineering*, 2023.
- [10] B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey," *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, p. 20200209, 2021.
- [11] Y. Hu, Z. Yang, and W. Hou, "Multiple receding imputation of time series based on similar conditions screening," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 3, pp. 2837–2846, 2021.
- [12] X. Miao, Y. Wu, L. Chen, Y. Gao, and J. Yin, "An experimental survey of missing data imputation algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 7, pp. 6630–6650, 2022.
- [13] G. E. Batista, M. C. Monard *et al.*, "A study of k-nearest neighbour as an imputation method," in *Soft Computing Systems - Design, Management and Applications, (HIS) 2002, December 1-4, 2002, Santiago, Chile*.
- [14] E. Acuna and C. Rodriguez, "The treatment of missing values and its effect on classifier accuracy," Springer, 2004.
- [15] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*, 2015.
- [16] Y. Liu, R. Yu, S. Zheng, E. Zhan, and Y. Yue, "Naomi: Non-autoregressive multiresolution sequence imputation," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2019.
- [17] Y. Tashiro, J. Song, Y. Song, and S. Ermon, "Csd: Conditional score-based diffusion models for probabilistic time series imputation," in *Annual Conference on Neural Information Processing Systems (NeurIPS), December 6-14, 2021, virtual*.
- [18] A. Cini, I. Marasca, and C. Alippi, "Filling the g_ap_s: Multivariate time series imputation by graph neural networks," *arXiv preprint*, 2021.

- [19] W. Cao, D. Wang, J. Li, H. Zhou, L. Li, and Y. Li, "Brits: Bidirectional recurrent imputation for time series," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2018.
- [20] X. Miao, Y. Wu, J. Wang, Y. Gao, X. Mao, and J. Yin, "Generative semi-supervised learning for multivariate time series imputation," in *Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI)*, 2021.
- [21] Z. Dai, E. Getzen, and Q. Long, "Sadi: Similarity-aware diffusion model-based imputation for incomplete temporal ehr data," in *International Conference on Artificial Intelligence and Statistics*, 2024.
- [22] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi, "Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting," in *The Tenth International Conference on Learning Representations (ICLR), Virtual Event, April 25-29, 2022*.
- [23] Z. Wang, X. Xu, W. Zhang, G. Trajcevski, T. Zhong, and F. Zhou, "Learning latent seasonal-trend representations for time series forecasting," *Advances in Neural Information Processing Systems*, 2022.
- [24] Y. Bengio, A. C. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI)*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [25] J. Wang, W. Du, W. Cao, K. Zhang, W. Wang, Y. Liang, and Q. Wen, "Deep learning for multivariate time series imputation: A survey," *arXiv preprint arXiv:2402.04059*, 2024.
- [26] S. Liu, X. Li, G. Cong, Y. Chen, and Y. Jiang, "Multivariate time-series imputation with disentangled temporal representations," in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*.
- [27] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.
- [28] K. Bandara, C. Bergmeir, and H. Hewamalage, "Lstm-msnet: Leveraging forecasts on sets of related time series with multiple seasonal patterns," *IEEE transactions on neural networks and learning systems*, 2020.
- [29] S. Mohammadpour, E. Freijinger, and P.-L. Bacon, "Decoupling regularization from the action space," *arXiv preprint arXiv:2406.05953*, 2024.
- [30] K. Wang, H. He, T. D. Nguyen, P. Kumar, and D. Roth, "On regularization and inference with label constraints," in *International Conference on Machine Learning*. PMLR, 2023, pp. 35740–35762.
- [31] H. Junninen, H. Niska, K. Tuppurainen, J. Ruuskanen, and M. Kolehmainen, "Methods for imputation of missing values in air quality data sets," *Atmospheric environment*, 2004.
- [32] S. Van Buuren and K. Groothuis-Oudshoorn, "mice: Multivariate imputation by chained equations in r," *Journal of statistical software*, 2011.
- [33] J. Yoon, J. Jordon, and M. Schaar, "Gain: Missing data imputation using generative adversarial nets," in *International conference on machine learning (ICML), Stockholm, Sweden, July 10-15, 2018*.
- [34] D. Wang, Y. Yan, R. Qiu, Y. Zhu, K. Guan, A. Margenot, and H. Tong, "Networked time series imputation via position-aware graph enhanced variational autoencoders," in *the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 2256–2268.
- [35] K. Yi, Q. Zhang, H. He, G. Long, and Z. Niu, "Neural time series analysis with fourier transform: A survey," *arXiv preprint*, 2023.
- [36] Y.-c. Park, J.-G. Jang, and U. Kang, "Fast and accurate partial fourier transform for time series data," in *the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 1309–1318.
- [37] Y. Zhou, J. Jiang, S.-H. Yang, L. He, and Y. Ding, "Musdri: Multi-seasonal decomposition based recurrent imputation for time series," *IEEE Sensors Journal*, vol. 21, no. 20, pp. 23213–23223, 2021.
- [38] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," in *The Eleventh International Conference on Learning Representations (ICLR), Kigali, Rwanda, May 1-5, 2023*.
- [39] J. Bjorck, A. Kabra, K. Q. Weinberger, and C. Gomes, "Characterizing the loss landscape in non-negative matrix factorization," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021.
- [40] H.-F. Yu, N. Rao, and I. S. Dhillon, "Temporal regularized matrix factorization for high-dimensional time series prediction," in *Annual Conference on Neural Information Processing Systems (NIPS)*, 2016.
- [41] X. Chen, Z. Cheng, N. Saunier, and L. Sun, "Laplacian convolutional representation for traffic time series imputation," *arXiv preprint*, 2022.
- [42] T. Cohen, "Learning transformation groups and their invariants," Ph.D. dissertation, PhD thesis, University of Amsterdam, 2013.
- [43] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, "Recommender systems," *Physics reports*, 2012.
- [44] K. Takeuchi, H. Kashima, and N. Ueda, "Autoregressive tensor factorization for spatio-temporal predictions," in *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2017, pp. 1105–1110.
- [45] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining (KDD)*, 2020.
- [46] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI), Macao, China, August 10-16, 2019*, pp. 1907–1913.
- [47] M. J. Azur, E. A. Stuart, C. Frangakis, and P. J. Leaf, "Multiple imputation by chained equations: what is it and how does it work?" *International journal of methods in psychiatric research*, 2011.
- [48] G. Takács, I. Pilászy, B. Németh, and D. Tikk, "Matrix factorization and neighbor based algorithms for the netflix prize problem," in *Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys 2008, Lausanne, Switzerland, October 23-25, 2008*, pp. 267–274.
- [49] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," 2010.
- [50] X. Chen, Z. Cheng, H. Cai, N. Saunier, and L. Sun, "Laplacian convolutional representation for traffic time series imputation," *IEEE Transactions on Knowledge and Data Engineering*, 2024.
- [51] W. Du, D. Côté, and Y. Liu, "Sait: Self-attention-based imputation for time series," *Expert Syst. Appl.*, vol. 219, p. 119619, 2023.
- [52] M. Choi and C. Lee, "Conditional information bottleneck approach for time series imputation," in *The Twelfth International Conference on Learning Representations*, 2023.
- [53] S. Kim, H. Kim, E. Yun, H. Lee, J. Lee, and J. Lee, "Probabilistic imputation for time-series classification with missing data," in *International Conference on Machine Learning*. PMLR, 2023, pp. 16654–16667.
- [54] J. Zhou, J. Li, G. Zheng, X. Wang, and C. Zhou, "Mtsci: A conditional diffusion model for multivariate time series consistent imputation," in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024, pp. 3474–3483.



Shuai Liu is a PhD student in the School of Computer Science and Engineering, Nanyang Technological University (NTU). He obtained his B.Sc degree in computer science from Peking University (PKU), China. Shuai's ongoing research is focused on spatial-temporal data mining and time series analysis.



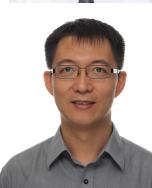
Xiucheng Li is an Assistant Professor of School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen). Prior to joining HITSZ, he was a Research Fellow at SCALE@NTU. His research interest focuses on representation learning, generative models and probabilistic inference, spatial and time series data analysis.



Yile Chen received the Ph.D. degree from Nanyang Technological University, Singapore, in 2022. He is currently a research fellow with the Singtel Cognitive and Artificial Intelligence Lab for Enterprises, Nanyang Technological University, Singapore. His current research interests include data mining and data management.



Yue Jiang received the B.Sc. degree from Nanyang Technological University and the M.Sc. degree from National University of Singapore. Currently, Yue Jiang is pursuing the Ph.D. degree in the School of Computer Science and Engineering at Nanyang Technological University. His current research interests include machine learning on multivariate time series and spatial temporal data mining.



Gao Cong is a professor with the School of Computer Science and Engineering, Nanyang Technological University (NTU). He is the director of Singtel Cognitive and Artificial Intelligence Lab for Enterprises at NTU. Prior to joining NTU, he worked with Aalborg University, Microsoft Research Asia, and the University of Edinburgh. His current research interests include geospatial data management, spatio-temporal data mining, recommendation, and mining social media.