MULTIVARIATE TIME-SERIES IMPUTATION WITH DISENTANGLED TEMPORAL REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Missing value problem is very common in multivariate time-series data. Many time series imputation methods have been developed in the literature, with success in different applications such as traffic and electricity consumption. However, these state-of-the-art methods all rely on entangled representations to model dynamics, which may fail to fully exploit the meaningful explanatory factors contained in time series. In addition, most existing models are proposed to deal with holistic time-series to capture cross-instance correlations and to identify time-varying dependencies, but they are not scalable to large-scale datasets. Different from most existing approaches, we propose TIDER, a novel matrix factorization-based method with disentangled temporal representations that account for multiple explanatory factors, namely *trend*, *seasonality*, and *local bias*, to simulate complex dynamics more realistically. The disentanglement learnt makes imputation process more reasonable and explainable. Moreover, our proposed approach is scalable and can fit well in large datasets. Empirical results show that our method not only outperforms existing approaches by notable margins on three real-world datasets, but also scales well to large datasets on which the state-of-the-art methods struggle. Disentanglement validation experiments further demonstrate the power of our model in obtaining accurate and meaningful disentangled explanatory components.

1 INTRODUCTION

Multivariate time series data is ubiquitous and arises from various fields like marketing, finance, traffic, healthcare, etc. It serves as input signals for multivariate time series analysis (e.g., forecasting Zeng et al. (2021) and classification Li et al. (2022)) in a wide spectrum of applications such as traffic flow forecasting Liu et al. (2020), electricity demand prediction Kaur et al. (2021), motion detection Laddha et al. (2021), health monitoring Tonekaboni et al. (2021), etc. The great importance of multivariate time series analysis has led to the development of forecasting/classification approaches in computational statistics Taylor & Letham (2017), econometrics Kaneko (2021), machine learning and data mining Stankeviciute et al. (2021); Zeng et al. (2021). Most of these approaches typically assume intact input when building models. However, real-world multivariate time series data often has missing values caused by device malfunction, communication failure, or costly measurement, which will impair the performance of these approaches or even make them inapplicable.

In light of this, many time series imputation methods have been proposed to infer missing values from the observed ones. A multivariate time series, denoted as $\mathbf{X} \in \mathbb{R}^{N \times T}$, consists of N univariate time series (called channels or instances) spanning over T time steps. Hence, it provides us with two perspectives for imputing missing values: modeling cross-channel correlations and exploiting temporal dynamics. Earlier methods Batista et al. (2002); Acuna & Rodriguez (2004); Van Buuren & Groothuis-Oudshoorn (2011); Box et al. (2015) either simply aggregate observed entries across channels by estimating the similarity between two distinct instances, or solely exploit local smoothness or linear assumption to fill in missing values. Since these methods lack the ability of modeling nonlinear dynamics and complex correlations, they may not perform well in practice.

In recent years, there has been an increasing trend to employ deep neural networks Cao et al. (2018); Liu et al. (2019); Tashiro et al. (2021); Cini et al. (2022) for time series imputation. Recurrent Neural Networks (RNNs) are mostly chosen as backbone to jointly model nonlinear dynamics along temporal dimension (by updating hidden states nonlinearly over time steps) and cross-channel correlations (by

mapping $\mathbf{x}_t \in \mathbb{R}^N$ from data space to hidden state space). Combined with adversarial training Liu et al. (2019); Miao et al. (2021) or multi-task loss Cao et al. (2018); Cini et al. (2022), these RNN-based imputation models achieve promising results in various applications. Despite their success, these methods rely solely on entangled representations (hidden states) to model dynamics. However, for many real-world multivariate time series, their dynamics are rich combinations of multiple independent explanatory factors like trend, seasonality (periodicity), and local idiosyncrasy Woo et al. (2022). Modeling combinations of these multiple factors with a single entangled representation might lead to inferior performance That is because entangled representation has to compromise itself to explain multiple orthogonal patterns (local changes and global variations) together. Bengio et al. (2013). It becomes exacerbated when seasonal patterns dominate, as RNNs lack the inductive bias to initiatively capture periodicity Hewamalage et al. (2019). Moreover, in these methods, all N time series of time length T are required to be fed into their models at each computational step, to capture cross-channel correlations and time-span dependencies, which is prohibitively costly for large N and T. Thus, these methods are not applicable to relatively large datasets.

To overcome these limitations, in this paper, we propose a novel multivariate time series imputation method, Time-series Imputation with Disentangled tEmporal Representations (TIDER), in which we explicitly model the complex dynamics of multivariate time series with disentangled representations to account for different explanatory factors. These disentangled representations offer more flexibility and robustness in response to the shift of the explanatory factors. Some allow our model to correctly respond to local interventions without compromising global characteristics while some merely captures global patterns. This disentanglement is realized by imposing different forms of constraints on different representations. These different constraints inject different inductive biases into corresponding representations and make them more prone to capturing specific patterns. In particular, we introduce a neighboring-smoothness representation matrix to explain global trend, a Fourier series-based representation matrix to define periodic inductive bias, and a local bias representation matrix to capture idiosyncrasies specific to individual time step. Moreover, TIDER is a scalable model. It is allowed to optimize its parameters by randomly selecting a subset of entries from the $N \times T$ matrix as training samples for each computational step. In this case, we adopt a mini-batch training paradigm, where our method can be easily applied to datasets with large N and T.In summary, our contributions are as follows.

- We propose a new scalable multivariate time series imputation model–TIDER–which introduces disentangled representations to account for various explanatory factors that characterize the complex dynamics of time series data. The proposed model can handle higher-dimensional and longer-term multivariate time series.
- To the best of our knowledge, TIDER is the first imputation model that introduces a learnable Fourier series-based representation to capture periodic patterns inherent in time series data.
- Extensive experiments show that our proposed method outperforms the baseline imputation methods in effectiveness, scalability and interpretability.

2 Related Work

Time series imputation problem has been extensively studied in the literature. Earlier proposals based on simple statistical strategies focus on exploiting local smoothness of temporal dimension as well as the similarity between different instances. For example, SimpleMean/SimpleMedian Fung (2006) imputes missing values by averaging, and KNN Batista et al. (2002) aggregates cross-channel observations to fill in missing slots with k-nearest neighbors. Several approaches attempt to impute by building linear dynamics, including linear imputation and state-space models Durbin & Koopman (2012). MICE Van Buuren & Groothuis-Oudshoorn (2011) explores the idea of estimating missing slots with chain equations. Unfortunately, these methods typically lack the ability of exploiting nonlinear dynamics and complex correlation across channels, thus resulting in inferior performance.

A recent trend in imputation methods is to apply deep neural networks for better capturing the nonlinearity of time series. In particular, RNNs are widely adopted and have achieved remarkable success in practice. BRITS Cao et al. (2018) treats missing values as variables which participate in bidirectional RNN back-propagation and are updated with hidden states. GAIN Yoon et al. (2018) employs Generative Adversarial Nets (GAN) for better imputation. NAOMI Liu et al. (2019) combines RNNs and adversarial training strategies to generate realistic-looking sequences. SSGAN Miao et al. (2021)

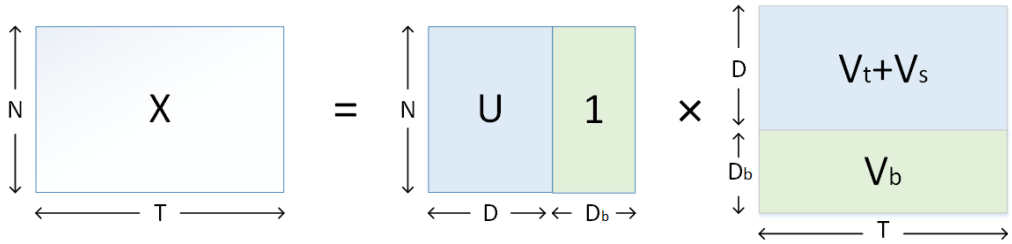

Figure 1: Architecture of the proposed method TIDER.

is a semi-supervised GAN-based model. It uses label information to assist imputation. GRIN Cini et al. (2022) fuse graph message passing into GRU structure to learn spatial-temporal patterns. These approaches are designed to model dynamics based on a single entangled representation, which is insufficient to capture multiple explanatory factors underlying time series, especially when seasonality emerges, since RNNs lack the inductive bias to initiatively capture periodicity Hewamalage et al. (2019). Moreover, these approaches are not scalable since they have to process all N time series of length T at each computation step to capture cross-channel correlations and identify dependencies in dynamics. Thus, they do not perform well, or are even not applicable to large time-series data.

Our proposed model is based on low-rank matrix factorization Yu et al. (2016); Bjorck et al. (2021). Vanilla matrix factorization imputes missing entries by learning two latent factors $\mathbf{U}$, $\mathbf{V}$ to exploit low-rank structure. However, it fails to consider the temporal continuity of time-series. To address this, TRMF Yu et al. (2016) imposes an autoregressive constraints on temporal factor $\mathbf{V}$. However, analogous to aforementioned methods, TRMF also relies on an entangled representation matrix to account for all explanatory factors underlying the dynamics. In contrast, our proposed TIDER tackles this issue by introducing multiple disentangled representations. These disentanglements are achieved by enforcing different forms of constraints into different representations. Particularly, Fourier-Series-based seasonality representation is explored for the first time in our approach, which plays a crucial role in accurate and meaningful imputation as shown in our experiments.

3 Problem Statement

Given N univariate time series data $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \in \mathbb{R}^T$ collected over T time steps. Put it into the multivariate time series matrix form $\mathbf{X} \in \mathbb{R}^{N \times T}$, whose n-th row represents n-th univariate time series (channel or instance) $\mathbf{x}_n$ and t-th column denotes the observation of all time series at time step t. Due to device malfunction, transmission failure or costly measurement, this multivariate time series matrix $\mathbf{X}$ is incomplete and a fraction of entries are missed randomly. We aim to infer the missing values from the observed ones, and we denote the mask matrix as $\mathbf{M} \in \{0, 1\}^{N \times T}$ where

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

4 Methodology

As mentioned in Section 1, existing deep neural network-based multivariate time series imputation methods mainly suffer from two limitations: 1) they model the temporal dynamics with a single entangled representation, which may not be capable of revealing the multiple explanatory factors underlying time series; 2) their scalability is limited by the fact that they have to feed all N instances into the model at each computation step to capture cross-channel correlations. In this paper, we intend to address these two limitations in a unified framework. The core idea behind our approach is to decompose multivariate time series data $\mathbf{X}$ into two latent factors $\mathbf{U}$ and $\mathbf{V}$, such that factor $\mathbf{U}$ only preserves characteristics that are unique to each instance whereas factor $\mathbf{V}$ is determined by multiple disentangled representations that jointly capture temporal dynamics. We employ such factorization for two reasons: 1) univariate time series instances (the rows of $\mathbf{X}$) are usually highly correlated; 2) observations at different time steps (the columns of $\mathbf{X}$) exhibit strong dependencies.

The benefits are twofold: 1) cross-channel correlations are decoupled since $\mathbf{U}$ only preserves instance-specific features, thus it allows us to optimize parameters by only selecting a subset of instances as training samples instead of considering all of them at each step, thereby enabling our approach to scale well to large datasets; 2) time-related information is isolated into $\mathbf{V}$, so it enables us to explain the potentially complex temporal dynamics with multiple disentangled representations.

4.1 Method Overview

Figure 1 shows the main architecture of TIDER. We adopt a low-rank matrix decomposition framework to factorize multivariate time series matrix $\mathbf{X}$ into two latent factors $\mathbf{U}$, $\mathbf{V}$. The latent factor $\mathbf{U}$ is a correlation-decoupled matrix which accounts for instance-specific characteristics, while the latent factor $\mathbf{V}$ is a temporal relevant matrix. Since the temporal dynamics underlying real-world time series can be complex and consist of rich interactions of multiple explanatory factors, describing them merely through an entangled representation matrix will lead to model degradation. Inspired by time series decomposition Wen et al. (2020) and disentanglement Woo et al. (2022), we propose to model $\mathbf{V}$ with multiple disentangled representations, each accounting for one particular explanatory factor. More specifically, we consider three important factors: *trend*, *seasonality*, and *bias* (Figure 2 illustrates an example of decomposing a time series into these three factors), which are specified by three representation matrix $\mathbf{V}_t$, $\mathbf{V}_s$, $\mathbf{V}_b$, respectively. Trend representation matrix $\mathbf{V}_t \in \mathbb{R}^{D\times T}$ captures the global trend shared by all instances, and seasonality representation matrix $\mathbf{V}_s \in \mathbb{R}^{D\times T}$ illustrates the periodic patterns hidden in temporal dynamics. $\mathbf{V}_t$ and $\mathbf{V}_s$ jointly determines the global dynamics. Bias representation matrix $\mathbf{V}_b \in \mathbb{R}^{D_b\times T}$ characterizes the local variation specific to each time step. Intuitively, $\mathbf{V}_b$ explains the individual idiosyncratic behaviors of time steps, which are orthogonal to global dynamics but shared across instances at a given time step. Hence, we treat it differently from $\mathbf{V}_t$, $\mathbf{V}_s$ and 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, the objective of TIDER is formulated as follows.

$$\begin{aligned} &\text{minimize}\, \|(\mathbf{X}-\mathbf{U}_a\mathbf{V})\odot\mathbf{M}\|^2 + \lambda_t f_t(\mathbf{V}_t) + \lambda_b f_b(\mathbf{V}_b) + \eta_1\|\mathbf{U}\|^2 + \eta_2\|\mathbf{V}\|^2, \\ &\mathbf{U}_a = [\ \mathbf{U}\ |\ \mathbf{1}\ ] \in \mathbb{R}^{N\times(D+D_b)}, \quad \mathbf{V} = \begin{bmatrix} \mathbf{V}_t+\mathbf{V}_s \\ \mathbf{V}_b \end{bmatrix} \in \mathbb{R}^{(D+D_b)\times T}, \end{aligned} \tag{2}$$

in which $\mathbf{U}_a$ is the augmented matrix of latent factor $\mathbf{U} \in \mathbb{R}^{N\times D}$, f_t and f_b are the corresponding inductive bias constraint functions imposed on $\mathbf{V}_t$ and $\mathbf{V}_b$, respectively. The last two terms $\eta_1\|\mathbf{U}\|^2$, $\eta_2\|\mathbf{V}\|^2$ are used to regularize the magnitude of latent factors, and λ_t, λ_b, η_1, $\eta_2 \in (0,1)$ are corresponding weights for each term. Once the optimal U_a and V are acquired, we use them to generate the imputed time-series matrix $\hat{\mathbf{X}}$ as

$$\hat{X}_{ij} = \begin{cases} X_{ij}, & M_{ij}=1, \\ (\mathbf{U}_a\mathbf{V})_{ij}, & M_{ij}=0. \end{cases} \tag{3}$$

We are now ready for elaborating the details of the three disentangled representation matrices.

4.2 Trend Representation Matrix

Trend representation matrix $\mathbf{V}_t$ characterizes the intrinsic trend of time series within a short period. Thus, the evolution patterns dominated by $\mathbf{V}_t$ are supposed to change gradually and smoothly in the absence of accidents or extreme events such as holidays (we will consider these external interventions and exogenous influences in $\mathbf{V}_b$). Based on this, we impose a smoothness constraint on trend representation matrix as

$$f_t(\mathbf{V}_t) = \sum_{j=2}^{T} \|\mathbf{v}_t^j - \mathbf{v}_t^{j-1}\|^2, \tag{4}$$

where $\mathbf{v}_t^j$ is the j-th column of $\mathbf{V}_t$. Equation 4 encourages close representations of two adjacent time steps in latent space, which will result in a smooth change in data space. Note that 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 Yu et al. (2016) which uses one temporal matrix to account for both short-term and long-term patterns by imposing regression constraint.

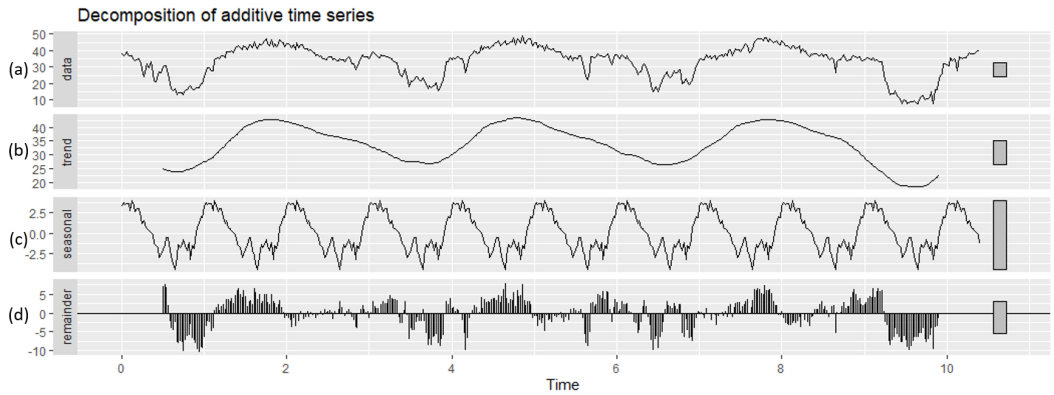

Figure 2: Visualization of time series decomposition. (a) is the raw time series, whereas (b), (c), and (d) are its decomposed trend, seasonality, and residual component. The grey bars in the right of each subplot show the relative scales of the components. Each grey bar represents the same length.

4.3 Seasonality Representation Matrix

Real-world time series data often demonstrate seasonal patterns. For instance, traffic flow exhibits strong daily and weekly seasonality due to regularity of human activities, such as commuting patterns. Solar power production present clear periodic characteristics caused by climate seasonality and meteorological conditions. Motivated by this and Fourier analysis in Section A.1.2, we propose to model the seasonality of time series by parameterizing representation matrix $\mathbf{V}_s$ with Fourier basis.

Recalling that $\mathbf{V}_s$ is a matrix with size $D \times T$, we represent each row with a superposition of $2K$ sinusoidal waves. More formally, let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{D \times K}$ be two coefficient matrices, and $\boldsymbol{\phi}_{\sin}$, $\boldsymbol{\phi}_{\cos} \in \mathbb{R}^{T \times K}$ be the corresponding Fourier basis matrices, which are defined as

$$\mathbf{A} = \begin{bmatrix} | & | & & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_K \\ | & | & & | \end{bmatrix}, \quad \boldsymbol{\phi}_{\sin} = \begin{bmatrix} | & | & & | \\ \sin(1\omega\mathbf{t}) & \sin(2\omega\mathbf{t}) & \dots & \sin(K\omega\mathbf{t}) \\ | & | & & | \end{bmatrix}, \tag{5}$$

$$\mathbf{B} = \begin{bmatrix} | & | & & | \\ \mathbf{b}_1 & \mathbf{b}_2 & \dots & \mathbf{b}_K \\ | & | & & | \end{bmatrix}, \quad \boldsymbol{\phi}_{\cos} = \begin{bmatrix} | & | & & | \\ \cos(1\omega\mathbf{t}) & \cos(2\omega\mathbf{t}) & \dots & \cos(K\omega\mathbf{t}) \\ | & | & & | \end{bmatrix}, \tag{6}$$

where $\mathbf{t} = [1, \dots, T]^\top$. For one specific time-series with period P, ω is calculated as $2\pi/P$.

Then the seasonality representation matrix $\mathbf{V}_s$ can be defined as

$$\mathbf{V}_s = \mathbf{A}\boldsymbol{\phi}_{\sin}^\top + \mathbf{B}\boldsymbol{\phi}_{\cos}^\top. \tag{7}$$

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

$$(\mathbf{v}_s^d)^\top = \sum_{k=1}^{K} \mathbf{A}_{d,k} \sin(k\omega\mathbf{t})^\top + \sum_{k=1}^{K} \mathbf{B}_{d,k} \cos(k\omega\mathbf{t})^\top, \tag{8}$$

which is a truncated Fourier series with coefficients $\mathbf{A}_{d,k}$, $\mathbf{B}_{d,k}$. This elaborate design of $\mathbf{V}_s$ provides meaningful periodic inductive bias, which enable our model to capture seasonal patterns more accurately and effectively, by learning coefficient matrices $\mathbf{A}$, $\mathbf{B}$ from data.

4.4 Bias Temporal Representation Matrix

The representation matrices $\mathbf{V}_t$, $\mathbf{V}_s$ presented so far jointly determine the global dynamics driven by endogenous explanatory factors of multivariate time series. However, there are also various external factors (e.g., holidays, weekdays/weekends) that could affect real-world time series. These external factors usually occur at specific time points and yield local variations within short time period, thus they are independent of global dynamics and impact nearly equally on all instances. To

account for this local variation, inspired by the idea of user and item bias explored in collaborative filtering Lü et al. (2012), we propose to learn another bias representation matrix $\mathbf{V}_b \in \mathbb{R}^{D_b \times T}$, where the representation of a specific time step is shared by all instances. In addition, we impose an autoregressive constraint on $\mathbf{V}_b$ in temporal dimension since the impact caused by local variation usually lasts for a short duration. Let $\mathbf{v}_t$ be the t-th column of $\mathbf{V}_b$ and L be the maximum time lag indicating duration, we define the constraint function as follows,

$$f_b(\mathbf{V}_b) = \sum_{t=L+1}^{T} \left\| \mathbf{v}_b^t - \sum_{l=1}^{L} \mathbf{W}_l \mathbf{v}_b^{t-l} \right\|^2, \tag{9}$$

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

4.5 Adaptive Weight for Trend and Seasonality

In Section 4.1, we use the additive form $\mathbf{V}_t + \mathbf{V}_s$ to characterize the influence of global impacting factors. This implicitly assumes that trend and seasonality components contribute equally to global dynamics. But in practice the importance of trend and seasonality can vary drastically across data sources, which is illustrated in Figure 2. To address this, we adopt a learnable parameter $\alpha \in (0, 1)$ to adaptively adjust the weight for these two components, which leads to a weighted additive form $\alpha\mathbf{V}_t + (1-\alpha)\mathbf{V}_s$. The temporal matrix $\mathbf{V}$ in Equation 2 then becomes:

$$\mathbf{V} = \begin{bmatrix} \alpha\mathbf{V}_t + (1-\alpha)\mathbf{V}_s \\ \mathbf{V}_b \end{bmatrix}. \tag{10}$$

5 Experiments

In this section, we evaluate the performance of our proposed model against existing multivariate time series imputation methods, by comparing their imputation accuracy and scalability. We upload the code of TIDER to `https://anonymous.4open.science/r/TIDER-527C`.

5.1 Experiment Setup

Datasets We conduct extensive experiments on three real world datasets. Guangzhou dataset is a relatively small one where all methods can fit in. Solar-energy shares an extremely long time-span while Westminster is one with large N. The latter two datasets are relatively large and thus some approaches struggle to handle them. For more details about datasets, please refer to Section A.2.

Baseline Methods We compare our model with popular baselines and recently-proposed methods, including statistical models (SimpleMean, KNN, MICE), MF-based methods (MF, MF+L2, SoftImpute, TRMF), and deep learning approaches (BRITS, GAIN, SAITS, CSDI, NAOMI, SingleRes). Details and settings of these baselines can be found in Section A.3. In addition, we also include TIDER (no W), a variant of TIDER without the learnable parameter α, to verify the effectiveness of the adaptive weight introduced in Section 4.5.

Evaluation Metrics We adopt RMSE, MAE and MAPE to evaluate imputation accuracy of all compared methods. Details of these three metrics can be found in Section A.4.

Training Setup We randomly remove a subset of entries from $\mathbf{X}$ as validation and test datasets separately. Let r be the missing rate variable, the ratio of training/validation/test is $(0.9-r)/0.1/r$. Since we randomly mask a certain proportion of observations, the experimental results fluctuate slightly. Thus, for each model we run experiments 7 rounds on every dataset and report the imputed results averaged over these 7 runs. All experiments are conducted on a Linux workstation with a 32GB Tesla V100 GPU. For more detailed hyperparameter settings, please refer to Table 5.

5.2 Imputation Performance Comparison

Table 1 and Table 2 shows the imputation accuracy of all methods on three datasets with different missing rate r. `OOM` indicates out of memory, the meaning of asterisk is illustrated in Section A.3.

Table 1: 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	OOM	OOM	OOM	1.893	1.366	0.119
CSDI	4.301	**2.991**	0.135	OOM	OOM	OOM	1.886	1.361	0.116
NAOMI	5.173	4.013	0.167	OOM	OOM	OOM	OOM	OOM	OOM
SingleRes	4.997	3.979	0.172	OOM	OOM	OOM	OOM	OOM	OOM
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**
Improvement (%)	13.3	/	0.3	94.1	87.4	9.7	1.9	0.7	0.1

Table 2: 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	OOM	OOM	OOM	1.998	1.453	0.129
CSDI	4.813	**3.202**	0.157	OOM	OOM	OOM	1.982	1.437	0.124
NAOMI	5.986	4.543	0.222	OOM	OOM	OOM	OOM	OOM	OOM
SingleRes	6.051	4.705	0.252	OOM	OOM	OOM	OOM	OOM	OOM
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**
Improvement (%)	10.5	/	0.5	85.7	98.5	13.9	2.3	1.5	0.3

First, we observe that the imputation accuracy of all methods drops as missing rate r raises , which is as expected since less patterns and information become available. Next, our proposed method, TIDER, achieve the best performance in most cases in terms of the three metrics. In particular, the superiority of our proposed method is much more significant on Solar-energy, the dataset with extremely large T.

Among all baseline methods, CSDI has the best performance on most datasets, and deep-learning methods (BRITS, GAIN, SAITS, CSDI) usually perform better than other baselines. However, our proposed method outperforms these models by large margins on each dataset. We hypothesize that this may be linked to our model's ability of capturing periodic patterns underlying time series where these models typically suffer because they lack the corresponding inductive bias. Moreover, when time span T increases to $52,560$ in solar-energy dataset, many deep learning methods run into OOM on a 32GB-memory GPU, even if we set the batch size to one. In contrast, TIDER is still applicable and results in the best performance. Because of their scalability, MF-based models can all suit datasets

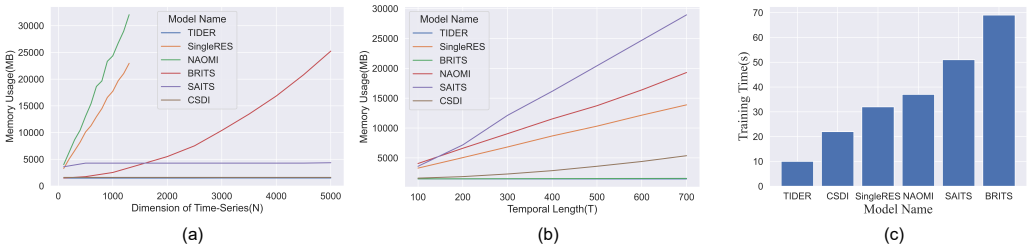

Figure 3: Scalability test on Westminster dataset: (a) Memory usage of different methods varies over N from 0 to 5000 with fixed $T = 100$; (b) Memory usage of different methods varies over T from 100 to 700 with fixed $N = 100$; (c) The average running time taken by different methods for every 100 epochs when processing a 100×100 matrix.

Table 3: Ablation Analysis of TIDER on Guangzhou dataset.

Name	RMSE	MAE	MAPE
$\mathbf{V}_s + \mathbf{V}_b$	7.241 (7.497)	5.388 (5.459)	0.272 (0.294)
$\mathbf{V}_t + \mathbf{V}_b$	5.680 (6.392)	4.311 (4.899)	0.190 (0.214)
$\mathbf{V}_s + \mathbf{V}_t$	5.297 (5.999)	4.014 (4.520)	0.170 (0.201)
TIDER	**4.168 (4.764)**	**3.098 (3.527)**	**0.132 (0.152)**

with large N and T. The difference between these models lies in constraints on $\mathbf{U}$ and $\mathbf{V}$. Compared with other MF-based baselines, TRMF outperforms others in most cases, for it intuitively captures auto-regressive dynamics. However, since it employs merely an entangled representation, TRMF is not enough to accurately model complex dynamics. TIDER (no W) and TIDER utilize disentangled features like trend, seasonality and local bias to rebuild patterns for time-series, which exploit more explanatory temporal factors, thus achieving better imputation result. Furthermore, TIDER performs better than TIDER (no W). This is in consistent with our previous analysis that trend and seasonality components might not contribute equally to global dynamics. To sum up, our proposed model TIDER outperforms all baselines, demonstrating its effectiveness in multivariate time-series imputation.

5.3 Scalability Analysis

We study the scalability of different methods in both memory usage and training time. In particular, we compare our method with state-of-the-art imputation methods, BRITS, NAOMI, SingleRes, SAITS and CSDI. Figure 3-(a) and Figure 3-(b) present the memory footprints of different methods against channel amount N and time span T. It can be seen from Figure 3-(a) that memory usage of NAOMI and SingleRes grow rapidly with very steep slops, and that of BRITS also grows quickly whereas our proposed TIDER consumes much less memory. Similarly, Figure 3-(b) shows that the memory consumption of NAOMI, SingleRes and SAITS also grows fast as T increases. Again, TIDER consumes the least amount of memory. There is also a gap between the curves of CSDI (BRITS) and TIDER in Figure 3, which is overwhelmed by the magnitude of NAOMI and SingleRes. We present these gaps in Section A.7. Lastly, Figure 3-(c) demonstrates the running time of different methods for 100 epochs when processing a 100×100 matrix. TIDER runs much faster than RNNs-based methods. Notably, TIDER outperforms BRITS by almost one order of magnitude. This is also true for total time taken by entire training process. Furthermore, the space complexity of BRITS is $O(N(N+T))$ Vaswani et al. (2017) while that of TIDER is $O(N+T)$. The complexity of NAOMI and SingleRes was not established and is tricky to analyze due to their complex divide-and-conquer strategy, but such complicated procedure yields more intermediate variables and thus results in more memory consumption than BRITS, as empirically validated by our experiments. In conclusion, this experiment demonstrates the scalability of our model, showing its ability to easily handle large datasets where existing RNN based methods struggle.

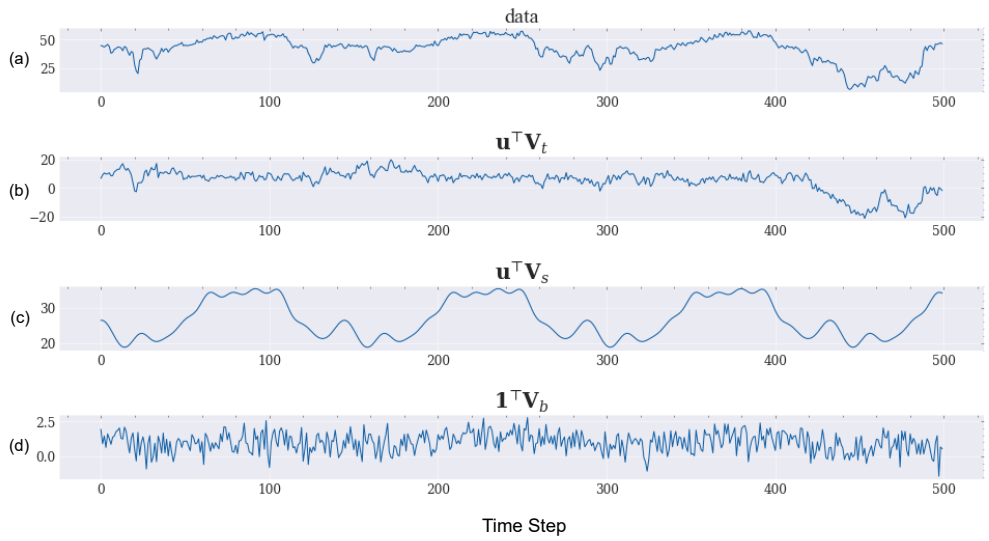

Figure 4: Case study of disentanglement. (a) is the original time-series, whereas (b),(c) and (d) represent the generated intrinsic trend, seasonality, and local variation within a period of time.

5.4 Ablation Analysis

To verify the effectiveness of our proposed disentangled representation matrices, we conduct an ablation study on Guangzhou dataset by removing one of the trend, seasonality, and bias representation matrices while leaving the rest of the model unchanged. The ablative results with missing rate $r = 0.2\,(0.4)$ are presented in Table 3. We find that the model performance drops no matter which representation is removed, which validates that all of our proposed disentangled representations play an important role in imputation and jointly enhance the performance of the final model.

5.5 Case Study of Disentanglement

In this experiment, we try to validate that the learned representations do achieve disentanglement. We select one time-series from Guangzhou dataset, and visualize it with the learnt patterns, namely, trend patterns $\mathbf{u}^\top\mathbf{V}_t$, seasonality patterns $\mathbf{u}^\top\mathbf{V}_s$, and bias patterns $\mathbf{1}^\top\mathbf{V}_b$, where $\mathbf{u}^\top$ is the time-series representation (a row of $\mathbf{U}$). From Figure 4, we can observe that the learned representations are able to generate semantically-meaningful patterns. Figure 4-(b) shows a gradually changing curve that corresponds to trend patterns, whereas Figure 4-(c) exhibits clearly periodicity and Figure 4-(d) manifests more idiosyncratic behaviors. These patterns are distinct and each of them captures information from a particular semantic perspective, which verifies that the learned representations $\mathbf{V}_t$, $\mathbf{V}_s$, $\mathbf{V}_b$ do exhibit independence and disentanglement. We also conduct experiments on synthetic data in Section A.5 to further show TIDER's power in meaningful and realistic disentanglement.

6 Conclusion

In this paper we propose a scalable multivariate time series imputation method, TIDER, with multiple novel inductive bias designs under the framework of low-rank matrix factorization. In contrast to existing imputation approaches, TIDER adopts semantically-meaningful disentangled representations to account for multiple explanatory factors of a time series. In particular, it enables capturing periodicity with a novel Fourier basis-based representation and allows to identify local time variation with a time bias representation. TIDER's superiority is verified by experimental results on three real-world datasets. Moreover, it scales well to large datasets where the existing methods struggle or even cannot fit. In the future, it is interesting to investigate how to use the disentangled representations for forecasting tasks and how to automatically learn period P in seasonality representation matrix V_s. Furthermore, when channel information is available, it is interesting to investigate how to design constraint on $\mathbf{U}$.

7 REPRODUCIBILITY STATEMENT

We provide an open-source implementation of our proposed model, TIDER, at https://anonymous.4open.science/r/TIDER-527C. Hyperparameter setting is shown in Section A.6. Users can download the code and run TIDER easily.

REFERENCES

Edgar Acuna and Caroline Rodriguez. The treatment of missing values and its effect on classifier accuracy. In *Classification, clustering, and data mining applications*. Springer, 2004.

Melissa J Azur, Elizabeth A Stuart, Constantine Frangakis, and Philip J Leaf. Multiple imputation by chained equations: what is it and how does it work? *International journal of methods in psychiatric research*, 20(1):40–49, 2011.

Gustavo EAPA Batista, Maria Carolina Monard, et al. A study of k-nearest neighbour as an imputation method. 2002.

Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE TPAMI*, 2013.

Johan Bjorck, Anmol Kabra, Kilian Q. Weinberger, and Carla P. Gomes. Characterizing the loss landscape in non-negative matrix factorization. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 6768–6776. AAAI Press, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/16836.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. Brits: Bidirectional recurrent imputation for time series. *NeurIPS*, 2018.

Jiawei Chen, Can Wang, Sheng Zhou, Qihao Shi, Jingbang Chen, Yan Feng, and Chun Chen. Fast adaptively weighted matrix factorization for recommendation with implicit feedback. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 3470–3477. AAAI Press, 2020. URL https://ojs.aaai.org/index.php/AAAI/article/view/5751.

Xinyu Chen, Yixian Chen, and Zhaocheng He. Urban traffic speed dataset of guangzhou, china, March 2018. URL https://doi.org/10.5281/zenodo.1205229.

Andrea Cini, Ivan Marisca, and Cesare Alippi. Filling the g_ap_s: Multivariate time series imputation by graph neural networks. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=kOu3-S3wJ7.

Wenjie Du, David Côté, and Yan Liu. Saits: Self-attention-based imputation for time series. *arXiv preprint arXiv:2202.08516*, 2022.

James Durbin and Siem Jan Koopman. *Time series analysis by state space methods*, volume 38. OUP Oxford, 2012.

David S Fung. Methods for the estimation of missing values in time series. 2006.

Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *CoRR*, 2019.

Yuta Kaneko. A time-series analysis of how google trends searches affect cryptocurrency prices for decentralized finance and non-fungible tokens. In *ICDM Workshops*, 2021.

Manpreet Kaur, Shalini Panwar, Ayush Joshi, and Kapil Gupta. Residential electricity demand prediction using machine learning. In Sarika Jain and Sven Groppe (eds.), *Proceedings of the International Semantic Intelligence Conference 2021 (ISIC 2021), New Delhi, India, February 25-27, 2021*, volume 2786 of *CEUR Workshop Proceedings*, pp. 331–340. CEUR-WS.org, 2021. URL http://ceur-ws.org/Vol-2786/Paper41.pdf.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2014.

Ankit Laddha, Shivam Gautam, Gregory P. Meyer, Carlos Vallespi-Gonzalez, and Carl K. Wellington. Rv-fusenet: Range view based fusion of time-series lidar data for joint 3d object detection and motion forecasting. In *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021, Prague, Czech Republic, September 27 - Oct. 1, 2021*, pp. 7060–7066. IEEE, 2021. doi: 10.1109/IROS51168.2021.9636083. URL https://doi.org/10.1109/IROS51168.2021.9636083.

Guozhong Li, Byron Choi, Jianliang Xu, Sourav S. Bhowmick, Daphne Ngar-yin Mah, and Grace Lai-Hung Wong. IPS: instance profile for shapelet discovery for time series classification. In *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022*, pp. 1781–1793. IEEE, 2022. doi: 10.1109/ICDE53745.2022.00179. URL https://doi.org/10.1109/ICDE53745.2022.00179.

Shuai Liu, Guojie Song, and Wenhao Huang. Real-time transportation prediction correction using reconstruction error in deep learning. *ACM Trans. Knowl. Discov. Data*, 14(2):17:1–17:20, 2020. doi: 10.1145/3369871. URL https://doi.org/10.1145/3369871.

Yukai Liu, Rose Yu, Stephan Zheng, Eric Zhan, and Yisong Yue. Naomi: Non-autoregressive multiresolution sequence imputation. *NeurIPS*, 2019.

Linyuan Lü, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. Recommender systems. *Physics reports*, 2012.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *JMLR*, 2010.

Xiaoye Miao, Yangyang Wu, Jun Wang, Yunjun Gao, Xudong Mao, and Jianwei Yin. Generative semi-supervised learning for multivariate time series imputation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 8983–8991. AAAI Press, 2021. URL https://ojs.aaai.org/index.php/AAAI/article/view/17086.

Kamile Stankeviciute, Ahmed M. Alaa, and Mihaela van der Schaar. Conformal time-series forecasting. In *NeurIPS*, 2021.

Elias M Stein and Rami Shakarchi. *Fourier analysis: an introduction*, volume 1. Princeton University Press, 2011.

Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. Matrix factorization and neighbor based algorithms for the netflix prize problem. In *RecSys*, 2008.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021.

Sean J. Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 2017.

Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. Unsupervised representation learning for time series with temporal neighborhood coding. In *ICLR*, 2021.

Stef Van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in r. *Journal of statistical software*, 2011.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, 2017.

Qingsong Wen, Zhe Zhang, Yan Li, and Liang Sun. Fast robuststl: Efficient and robust seasonal-trend decomposition for time series with complex patterns. *SIGKDD*, 2020.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven C. H. Hoi. Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=PilZY3omXV2.

Jinsung Yoon, James Jordon, and Mihaela Schaar. Gain: Missing data imputation using generative adversarial nets. In *International conference on machine learning*, pp. 5689–5698. PMLR, 2018.

Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. *NIPS*, 2016.

Sebastian Zeng, Florian Graf, Christoph Hofer, and Roland Kwitt. Topological attention for time series forecasting. In *NeurIPS*, 2021.

A Appendix

A.1 Preliminary

A.1.1 Low-rank Matrix Factorization

Given a matrix $\mathbf{X} \in \mathbb{R}^{N \times T}$ with rank $k \ll \min\{N, T\}$, the low-rank matrix factorization aims to factorize $\mathbf{X}$ as the product of two low-rank matrices $\mathbf{U} \in \mathbb{R}^{N \times k}$ and $\mathbf{V} \in \mathbb{R}^{k \times T}$,

$$\mathbf{X} = \mathbf{U}\mathbf{V}. \tag{11}$$

This low-rank matrix factorization can be extended to circumstance where the true rank of $\mathbf{X}$ is larger than k. In such a case, we use a metric ℓ to measure the discrepancy between $\mathbf{X}$ and the approximated product $\mathbf{U}\mathbf{V}$ as

$$\ell(\mathbf{X}, \mathbf{U}\mathbf{V}) = \|\mathbf{X} - \mathbf{U}\mathbf{V}\|^2, \tag{12}$$

where $\|\cdot\|$ denotes the Frobenius norm of a matrix. This low-rank approximation is fairly reasonable in practice, since many real-world structural observation matrices are low rank. This has been widely adopted and extensively studied in recommender systems Takács et al. (2008); Chen et al. (2020). In the context of multivariate time series data, we propose to adopt low-rank matrix factorization for two reasons: 1) univariate time series instances (the rows of $\mathbf{X}$) are usually highly correlated; 2) observations at different time steps (the columns of $\mathbf{X}$) exhibit strong dependencies.

A.1.2 Fourier Series

Fourier analysis Stein & Shakarchi (2011) states that any real-valued periodic function $f(x)$ can be represented by an infinite series of sinusoidal functions, known as Fourier series. These sinusoidal functions have their own coefficients and distinct frequencies. More formally, the Fourier series for a function $f(x)$ with its period P can be described as

$$f(x) = \sum_{n=0}^{\infty} a_n \cos(n\omega x) + \sum_{n=0}^{\infty} b_n \sin(n\omega x), \tag{13}$$

where $\omega = 2\pi/P$, and $a_n, b_n \in \mathbb{R}$ are the corresponding coefficients. Since $\sin(n\omega x), \cos(n\omega x)$ functions are orthogonal to each other and are able to span over the entire functional space, they are also called Fourier basis.

A.2 Dataset Details

- Guangzhou Traffic Data. This dataset Chen et al. (2018) contains traffic speed of 214 anonymous urban road segments during 5 days with a 10-minute sampling rate in Guangzhou, China. It results in a 214×500 multivariate time series matrix.

Table 4: Summary of dataset statistics.

Name	Number of instances (N)	Number of time steps (T)
Guangzhou[1]	214	500
Solar-Energy[2]	137	52,560
Westminster[3]	7,489	744

- Solar-Energy Production Data. This dataset consists of solar power production records of 137 PV plants in Alabama, USA sampled every 10 minutes. It results in a $137 \times 52,560$ data matrix.
- Westminster Uber Movement Data. This dataset contains hourly averaged speed of road segments in Westminster, in Jan 2020, released by Uber. Dimension of its data matrix is $7,489 \times 744$.

The dataset statistics is summarized in Table 4.

A.3 Details of Baseline Models

The details of baseline methods are briefly summarized as follows. For SimpleMean, KNN, and SoftImpute, we use their implementation provided by the package fancyimpute whereas for BRITS, GAIN, SAITS, CSDI, NAOMI, SingleRes, we use the source codes released by their authors.

- SimpleMean Acuna & Rodriguez (2004)[4] It imputes missing entries with mean values of corresponding columns.
- KNN Batista et al. (2002)[5] It first finds k nearest rows with the highest similarity score to the target row, and then uses the weighted sum of these k rows for imputation.
- SoftImpute Mazumder et al. (2010)[6]. It is a matrix completion approach based on iterative soft thresholding of Singular Value Decomposition (SVD).
- MICE Azur et al. (2011)[7]. It is the abbreviation for Multivariate Imputation by Chained Equations, a widely-used R package for imputation.
- MF Takács et al. (2008). It applies low-rank matrix factorization without any constraint on the latent factors.
- MF+L2 Takács et al. (2008). It applies low-rank matrix factorization with L2 regularization on $\mathbf{U}$ and $\mathbf{V}$.
- TRMF Yu et al. (2016). It applies low-rank matrix factorization with an autoregressive constraint imposed on temporal matrix $\mathbf{V}$.
- BRITS Cao et al. (2018)[8]. It is a time series imputation method based on Bidirectional Recurrent Neural Nets and time decay mechanism.
- GAIN Yoon et al. (2018)[9]. It employs Generative Adversarial Nets (GAN) for imputation.
- SAITS Du et al. (2022)[10]. It imputes missing values based on self-attention mechanism.
- CSDI Tashiro et al. (2021)[11]. It utilizes score-based diffusion models to explore correlations between observed values and for imputation.

[4] https://github.com/iskandr/fancyimpute
[5] https://github.com/iskandr/fancyimpute
[6] https://github.com/iskandr/fancyimpute
[7] https://github.com/amices/mice
[8] https://github.com/caow13/BRITS
[9] https://github.com/jsyoon0823/GAIN
[10] https://github.com/WenjieDu/SAITS
[11] https://github.com/ermongroup/CSDI

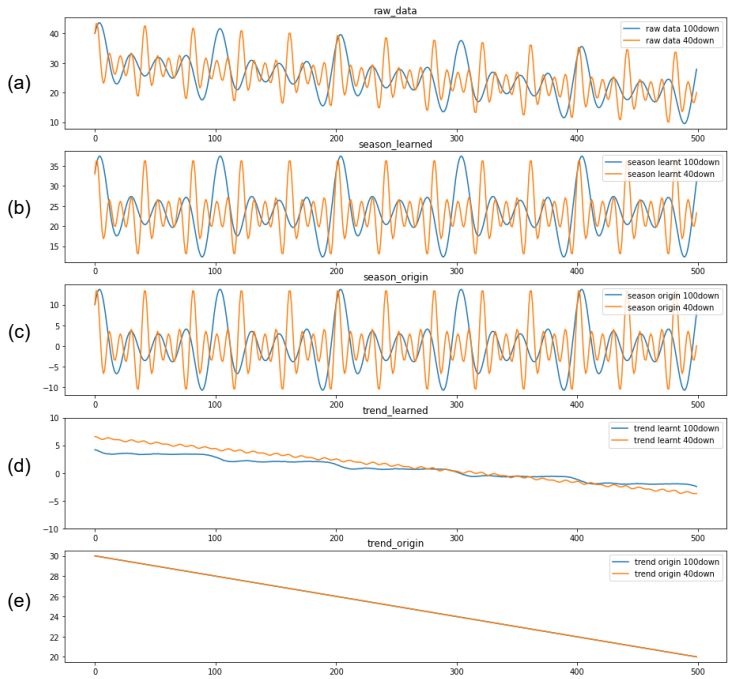

Figure 5: Disentanglement validation on synthetic dataset. Time-series are composed of the same trend but different seasonality. (a) shows the raw time-series. (b) and (d) show the trend and seasonality components TIDER has learnt. (c) and (e) depict the ground trend and seasonality components which compose the raw time-series.

- NAOMI Liu et al. (2019)[12]. It combines Bidirectional Recurrent Neural Nets with adversarial training to offer non-autoregressive style imputation. The divide-and-conquer strategy is adopted.
- SingleRes Liu et al. (2019)[13]. It is the autoregressive counterpart of NAOMI.

In our experiments, we find that BRITS with the suggested batch size and RNN hidden size will result in OOM (Out of Memory) on Westminster dataset. Thus we reduce the batch size to 1 and the dimension of RNN to 10 such that it could just fit the 32 GB GPU on that dataset, we use asterisk symbol to indicate its results in Table 1 and Table 2.

A.4 Details of Metrics

Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Precentage Error (MAPE) are adopted to evaluate the imputation accuracy of all compared methods. These three metrics are defined as

$$\text{RMSE} = \sqrt{\frac{\sum_{ij\in\Omega}(\mathbf{X}_{ij} - \hat{\mathbf{X}}_{ij})^2}{|\Omega|}}, \text{MAE} = \frac{\sum_{ij\in\Omega}|\mathbf{X}_{ij} - \hat{\mathbf{X}}_{ij}|}{|\Omega|}, \text{MAPE} = \sum_{ij\in\Omega}\frac{|\mathbf{X}_{ij} - \hat{\mathbf{X}}_{ij}|}{|\Omega|\cdot|\mathbf{X}_{ij}|}. \tag{14}$$

where $\mathbf{X}_{ij}$ denotes the ground-truth values, $\hat{\mathbf{X}}_{ij}$ is the imputed values, and Ω is the index set of missing entries to be evaluated.

[12] https://github.com/felixykliu/NAOMI
[13] https://github.com/felixykliu/NAOMI

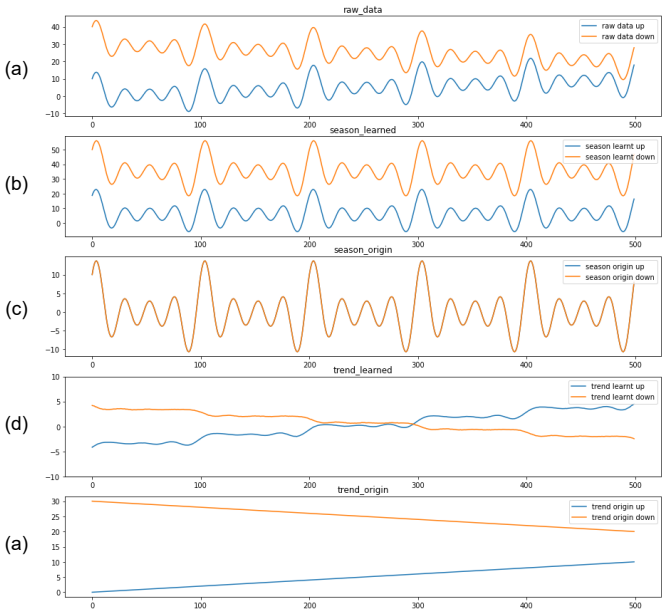

Figure 6: Disentanglement validation on synthetic dataset. Time-series are composed of different trend but the same seasonality. (a) shows the raw time-series. (b) and (d) show the trend and seasonality components TIDER has learnt. (c) and (e) depict the ground trend and seasonality components which compose the raw time-series.

A.5 Disentanglement Validation On Synthetic DataSet

We conducted supplementary experiments to validate TIDER's disentangling ability, where we build two sets of synthetic multivariate time-series data. Time-series in the first set are composed of same trend components but different seasonality components, while those in the second set consist of different trend components but same seasonality components. We use TIDER to capture explanatory factors of these time-series, and compare the disentangled components learnt with ground truth components used in constructing these synthetic data. Results of these disentanglement are depicted in Figure 5 and Figure 6. We can clearly find that the disentangled components TIDER learnt are very similar to the ground-truth ones. This further validation TIDER's power and might in achieve meaningful and realistic disentangled explanatory factors underlying time-series.

A.6 Hyperparameter Setting

We implement TIDER using Python 3.6 and Pytorch 1.9, and optimize the model parameters using Adam Kingma & Ba (2014) with a learning rate 0.001.

The general principle and guideline on parameter setting are as follows:

- ω: This hyperparameter is involved in the seasonality representation matrix. ω is in close relation with time-series' period P. By jointly applying our prior knowledge and trial, We set P for Guangzhou, Solar-Energy and Westminster dataset as 168, 168 and 24 respectively. In other words, the optimal ω for these datasets are $\frac{\pi}{84}$, $\frac{\pi}{84}$ and $\frac{\pi}{12}$.
- D: Dimension for matrix $\mathbf{U}$ and $\mathbf{V}_t$ is also an important hyperparameter. Small D will lack the ability to learn enough information while large D is prone to overfitting.
- λ_t, λ_b: These are the corresponding weights for trend representation matrix's and bias representation matrix's constraint function. We tune them by making a choice from a scale set $\{0.01, 0.1, 0.2, 0.5, 0.8\}$.

By following this principle, the optimal hyperparameters we obtained are listed in Table 5.

Table 5: Hyperparameters of TIDER for the three datasets.

Name ($r = 0.2$)	Guangzhou	Solar-Energy	Westminster
batch size	32	32	512
epoch	500	500	500
D	50	50	150
D_b	2	2	2
η_1	0.01	0.01	0.01
η_2	0.01	0.01	0.01
λ_t	0.1	0.01	0.01
λ_b	0.2	0.1	0.1
K	30	2	15
ω	$\frac{\pi}{84}$	$\frac{\pi}{84}$	$\frac{\pi}{12}$
Name ($r = 0.4$)	**Guangzhou**	**Solar-Energy**	**Westminster**
batch size	32	32	512
epoch	500	500	500
D	50	50	150
D_b	2	2	2
η_1	0.01	0.01	0.01
η_2	0.01	0.01	0.01
λ_t	0.1	0.01	0.01
λ_b	0.2	0.5	0.1
K	20	2	20
ω	$\frac{\pi}{84}$	$\frac{\pi}{84}$	$\frac{\pi}{12}$

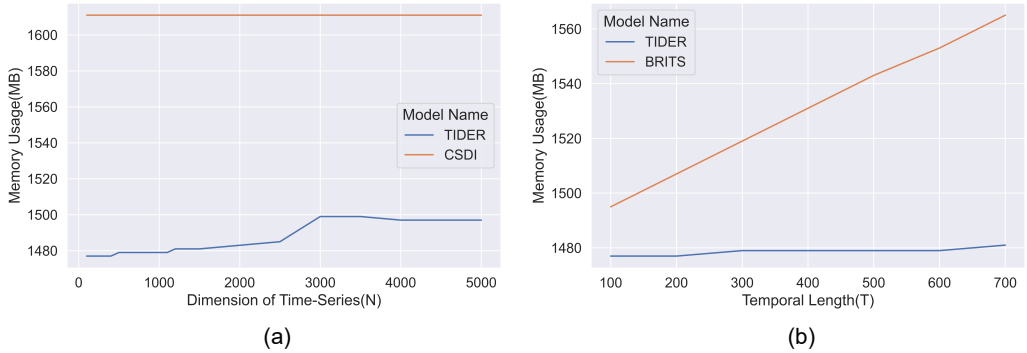

Figure 7: Supplementary scalability test on Westminster dataset. (a) Memory usage between TIDER and CSDI varies over N from 0 to 5000 with fixed $T = 100$; (b) Memory usage between TIDER and BRITS varies over T from 100 to 700 with fixed $N = 100$.

A.7 Supplemental Scalability Analysis

In Section 5.3, there is little gap between memory usage of CSDI and TIDER in Figure 3(a), and little gap between memory usage of BRITS and TIDER in Figure 3(b). In Figure 3-(a) (resp. Figure 3-(b)) the memory usage gap between CSDI (resp. BRITS) and TIDER is overwhelmed by the large magnitude of other methods. Thus, we zoom in to give a comparison in this section. As depicted in Figure 7, there are still large memory usage gaps between TIDER and them, which demonstrates the scalability superiority of TIDER.