Expert Systems With Applications 331 (2026) 133204

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A transformer-based transfer learning algorithm for time series imputation and forecasting with data scarcity

Rui Ye [a], Yu Ding [a], Jing Zhang [b,*], Yi-Heng Zhu [a,*]

[a] College of Artificial Intelligence, Nanjing Agricultural University, NanJing, 211800, China
[b] College of Information Science and Technology & Artificial Intelligence, Nanjing Forestry University, NanJing, 210037, China

ARTICLE INFO

Keywords:
Time series forecasting
Time series imputation
Transformer
Transfer learning

ABSTRACT

To tackle the prevalent challenges of missing values and data scarcity in time series analysis, this paper introduces a Transformer-based transfer learning algorithm (TTL-TS), which can simultaneously address both the two challenges. It involves a Transformer-based Representation Learning module (TRL) and a Transfer Learning module (TLM). TRL tackles the missing data distribution problem by designing a joint-optimization technique. To preserve critical temporal dynamics, this technique evaluates reconstruction performance at both the holistic data level and the finer trend-season level. TLM addresses the data scarcity bottleneck by designing a flexible transfer learning framework. It facilitates knowledge transfer through useful features selection and intermediate domains construction. Effectiveness of this proposed algorithm is underpinned by extensive experiments in our paper.

1. Introduction

Due to all kinds of reasons, such as equipment failures or unexpected malfunctions (Pratama et al., 2016), time series issues with missing values are particularly prevalent in scientific and industrial fields (Li et al., 2020c; Ruan et al., 2016). These missing values hamper the interpretation of time series and pose challenge to machine learning models that require fully-observed data. To address this challenge, several significant studies have been proposed, which can be broadly categorized into traditional deep learning-based methods, diffusion-based methods, and attention-mechanism-based methods. In the realm of deep learning-based approaches, Che et al. (2018) first introduced the concept of time decay and handled missing values by using RNN architecture. Other RNN-based approaches, such as MRNN (Yoon et al., 2018) and BRITS (Cao et al., 2018), have also been widely employed for missing time series issues. Due to the superior performance of diffusion technique, diffusion models have shown success in time series imputation and forecasting task (Feng et al., 2024b; Tashiro et al., 2021). Recently, attention mechanisms have also been applied to missing input time series (Du et al., 2023; Nayak et al., 2024). Most of these approaches design an encoding of time and use self-attention to capture temporal and feature relations within the data (Du et al., 2023).

Because of data privacy or data missing reasons, data scarcity is also a common problem that hinders time series analysis. Transfer learning offers a popular solution to this problem by leveraging useful knowledge from a known source domain to target domain (He et al., 2023; Iman et al., 2023).

In this work, to tackle the mentioned two challenges in time series analysis, i.e. missing values and data scarcity, we propose a Transformer-based transfer learning algorithm (TTL-TS) for time series imputation and forecasting. It comprises two main modules: the Transformer-based Representation Learning module (TRL) and the Transfer Learning module (TLM). TRL is designed to estimate the distribution of time series with missing values. TLM constructs a transfer learning framework to overcome the further challenge of data scarcity.

Specifically, inspired by the superior performance of the self-attention mechanism, TRL adopts the Transformer encoder structure as its base architecture and further explores the intrinsic characteristics of time series by integrating a seasonal-trend decomposition mechanism. A joint-optimization technique is proposed for model training, which involves three learning tasks: observed values reconstruction, missing values reconstruction and seasonal-trend characteristics reconstruction. The first two tasks consider the reconstruction performance on the whole data, while the third task provides a more refined learning objective by emphasizing seasonal and trend components, which enables the model to notice some potential characteristics of time series. These leaning tasks work in synergy to enhance the imputation and forecasting performance of the model.

TLM aims to address the challenge of data scarcity. It adopts the idea of transfer learning, which entails transferring knowledge from a related

* Corresponding authors.
E-mail addresses: yerui@njau.edu.cn (R. Ye), yuding@njau.edu.cn (Y. Ding), jzhangnjfu@njfu.edu.cn (J. Zhang), yihzhu@njau.edu.cn (Y.-H. Zhu).

https://doi.org/10.1016/j.eswa.2026.133204
Received 26 April 2025; Received in revised form 5 June 2026; Accepted 7 June 2026
Available online 9 June 2026

and data-rich source domain to the data-insufficient target domain. It involves two major steps: constructing the source domain and model, and extracting useful knowledge from the source domain. Existing TL tasks usually assume that the related source domain is known and available. This may be hard to achieve in many practical scenes. In TLM, we first construct a related source domain only based on limited target samples. The resulting source model is then leveraged to support robust training of the target model for downstream tasks. This approach allows leveraging external temporal patterns for the target task. Specifically, TLM accomplishes knowledge transfer through two primary mechanisms: useful features selection and intermediate domains construction.

Considering the difference between source and target tasks, not all source information is beneficial to the target task. Therefore, in the first mechanism, we assign varying weights to different source features according to their effects on the target task. Valid source features are prioritized, while useless ones are disregarded. As demonstrated in Dai et al. (2024), intermediate domains existing along the path connecting source and target domains can reveal some inter-domain correlations. In the second mechanism, we hypothesize that these intermediate domains contain additional valuable information. We aim to explore the valid knowledge to enhance the training of target model.

It is worth noting that TTL-TS does not fully align with the standard transfer learning setting, where a naturally available external source domain is assumed. But it still satisfies the broad definition of transfer learning (Pan & Yang, 2009) and can be categorized as a subclass of transfer learning where no natural source domain is available, which is well-supported by the following works. For instance, Du et al. (2024) explicitly construct a pseudo-source domain from target samples to enable transfer without accessing source data. Ding et al. (2022) further prove that the source domain need not be accessible. It can be effectively instantiated as a synthetic proxy derived from target-side information constrained by source priors. Thus, TTL-TS is still characterized as a transfer learning method.

Although TTL-TS constructs the source domain from limited target data, it is distinct from two superficially similar approaches: synthetic-data augmentation and self-distillation. In synthetic-data augmentation, the augmented samples are usually directly fed into the training set, with no independent source model and no cross-domain alignment loss. This is different from TTL-TS. TTL-TS constructs a source domain and then trains a source model on it, which distills higher-level feature patterns rather than directly augmenting the training data.

Self-distillation usually involves a teacher model trained on the target data, which aims to produce soft labels to supervise a student model. The knowledge transfer path is different from that in TTL-TS. In TTL-TS, the source model is trained on the constructed source domain. Then the TLM module uses useful features selection and intermediate domains construction to accomplish feature-level alignment. There is no soft-label supervision between teacher and student, and the entire alignment is implemented through cross-domain loss terms (as shown in Eqs. (19)–(23)), which satisfies the mechanism of transfer learning. We provide empirical comparisons with Gaussian-noise augmentation and distillation baselines in Section 4.5.3 to further demonstrate the effectiveness of the transfer learning mechanisms employed by TTL-TS.

Fig. 1 depicts the key components of the proposed TTL-TS.

Our main contributions can be summarized as follows:

- To alleviate the challenges of missing values and data scarcity in time series analysis, we design a Transformer-based transfer learning algorithm for imputation and forecasting task. Though recently corresponding methods have been proposed to tackle these two challenges individually, there has been little research considering both of these two challenges.
- To better estimate the distribution of missing values, a joint-optimization technique composed of three learning tasks is designed. It evaluates reconstruction performance from both a holistic data perspective and a finer seasonal-trend perspective.
- To mitigate the plight of target data scarcity, a transfer learning framework is designed. It creates a source domain based on the limited target data and extracts useful source knowledge in two ways: automatically selecting effective source features and learning additional information from the intermediate domains. This TL framework is flexible to different networks.

2. Related work

In this section, we respectively review the prior work of time series imputation and time series forecasting.

2.1. Time series imputation

Various imputation techniques have been developed to address missing values in time series analysis. We categorize these imputation methods into three main approaches: Traditional deep learning-based methods, Diffusion-based methods, and Attention-mechanism-based methods.

Traditional deep learning-based methods: Che et al. (2018) propose a Gated Recurrent Unit (GRU) based model to alleviate data missing challenge in time series classification tasks. It firstly introduces the concept of time decay, which is widely used in subsequent researches. In Yoon et al. (2018), to alleviate the data missing problem in medical fields, a new RNN-based method that simultaneously trains an interpolation block and an imputation block is proposed. Cao et al. (2018) design an RNN-based method for time series imputation. It treats imputed values as variables of RNN graph and considers the feature correlations. In Shukla and Marlin (2021b), authors design a variational autoencoder (VAE) architecture with a time attention mechanism to learn temporal relationships in irregularly sampled time series.

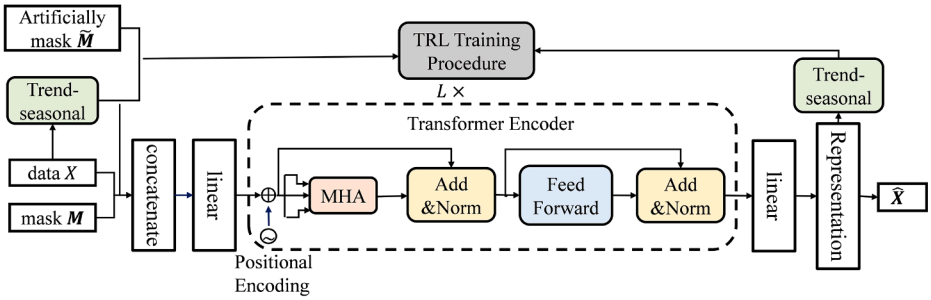
Diffusion-based methods: Tashiro et al. (2021) propose a conditional score-based diffusion technique for time series imputation. It learns useful correlations from observed values and employs a self-supervised method to train the diffusion model. Alcaraz and Strodthoff (2022) design a novel time series imputation model by tactfully combining state-space models with diffusion models.

Attention-mechanism-based methods: The first application of self-attention mechanisms to multivariate time series imputation is presented by Ma et al. (2019). It jointly captures the self-attention across different dimensions with high efficiency. Shukla and Marlin (2021a) propose an attention-based approach including a temporal VAE architecture with a heteroscedastic layer,which is utilized to interpolate the irregularly sampled time series. Du et al. (2023) design a self-attention-based method with two diagonally-masked self-attention blocks. It explicitly explores the temporal and feature relations of time series.

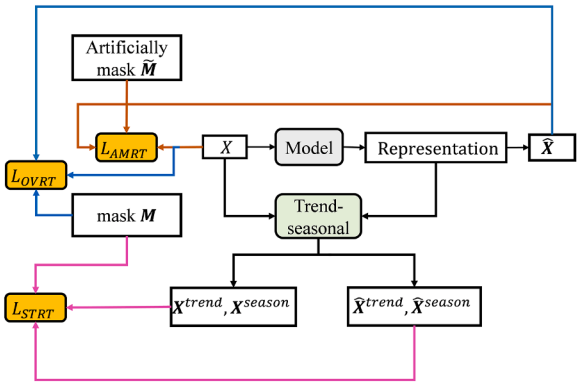
2.2. Transfer learning and meta-learning for time series imputation

To address the challenge of data scarcity, time series imputation algorithms based on transfer learning and meta-learning have been proposed recently. For example, a transfer learning-based algorithm that imputes long-interval consecutive missing values is proposed (Ma et al., 2020). It employs transfer learning to leverage patterns from the most similar complete sequence to the target sequence with missing values. Liang et al. (2025) redefine the imputation under variable subset forecasting problem as a cross-domain knowledge transfer problem, where invariant patterns from complete source data are adapted to target data with missing values. Zhang et al. (2025) propose a novel framework to tackle cross-domain imputation, which can complete the imputation task under high missing rates and domain shifts.

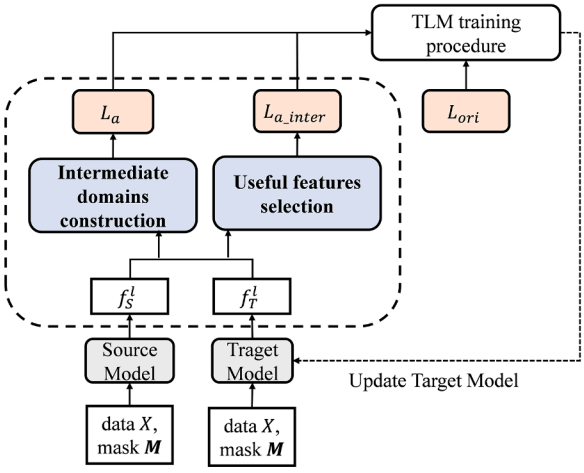
Zhu and Zhao (2025) design a bidirectional meta-learning approach, which can be applied to time series imputation tasks. Almeida et al. (2025) propose a meta-learning-based framework for univariate time



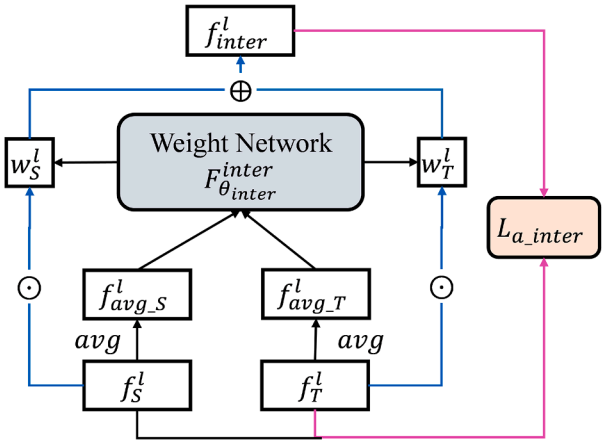
(a) TRL module



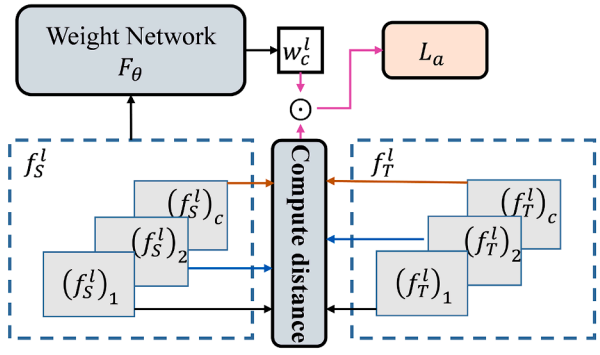
(b) Training procedure of TRL



(c) TLM module



(d) Intermediate domains construction of TLM



(e) Useful features selection of TLM

Fig. 1. Key components of the proposed TTL-TS. (a) The overall TRL module; (b) The training procedure of TRL. It contains three learning tasks, which are introduced in Section 3.1. (c) The overall TLM module. It cotains two parts: (d) Intermediate domains construction of TLM and (e) Useful features selection of TLM.

series imputation. It introduces a novel HybridLSTM neural network architecture to recommend the most suitable imputation technique within a given time series.

2.3. Transfer learning optimization

As an important branch of machine learning, transfer learning has made significant progress. However, in practical applications, the optimization of transfer learning faces multiple challenges. To address these challenges, many novel transfer learning methods have been proposed.

Zhuang et al. (2020) provide a survey on transfer learning, which systematically reviews existing transfer learning research from data and model perspectives. To address the "reckless loss" of discarding task-specific layers during fine-tuning, You et al. (2020) propose a two-step transfer learning framework. With the rise of pre-trained models, transfer learning methods tailored for large language models, i.e., Parameter-Efficient Fine-Tuning (PEFT) algorithms have been developed in recent years. Liao et al. (2023) introduce a novel adapter technique that is applied directly to pre-trained parameters instead of hidden representations. To address the challenge of large data distribution differences, Feng et al. (2024a) present a framework including an adaptive layer

selection strategy and a collaborative loss function to enhance the model's adaptability to the target task.

In data-driven evolutionary transfer optimization filed, many high-quality studies have been proposed. In DETO (Li et al., 2023), a data-driven evolutionary transfer optimization framework is designed for expensive problems in dynamic environments. This framework employs hierarchical multi-output Gaussian processes and adaptive source-task selection to warm-start optimization upon environmental changes. For expensive multi-objective settings, Li and Chen (2022) develop batched data-driven evolutionary multi-objective optimization based on manifold interpolation, exploiting Karush-Kuhn-Tucker structure to explore diversified solutions along the approximated Pareto set. Chen and Li (2023) further address disconnected Pareto fronts via multiple-gradient descent on surrogate landscapes. Complementary lines include transfer Bayesian optimization for expensive black-box problems in dynamic environments (Chen & Li, 2021) and surrogate-assisted evolutionary algorithms with transfer learning for dynamic expensive multi-objective optimization (Fan et al., 2020).

Recent efforts also aim to operationalize transfer optimization in software platforms. Mao and Li (2024) present OpenTOS, an open-source system for transfer-learning Bayesian optimization that supports modular design, benchmarking, and application of transfer-learning Bayesian optimization (TLBO) algorithms.

Transfer optimization has also been studied in bilevel programming and software engineering. Chen et al. (2021) propose a novel bilevel optimization framework integrating Parallel Computing and Transfer Learning. In cross-project defect prediction (CPDP), limited target-project data motivate transferring models from historical projects. BiLO-CPDP (Li et al., 2020a) formulates automated CPDP model discovery as bilevel programming with combinatorial upper-level pipeline selection and lower-level hyperparameter tuning; MBL-CPDP (Chen et al., 2025) extends this paradigm to multi-objective bilevel optimization for pipeline and hyperparameter co-optimization. Li et al. (2020b) empirically study how automated parameter optimization interacts with transfer learning in CPDP.

2.4. Time series forecasting

Time series forecasting has been a popular topic in many fields (Casolaro et al., 2023; Chen et al., 2023; Liu et al., 2024a). An RNN-based methodology is proposed for probabilistic forecasting (Salinas et al., 2020). It offers forecasts for items with little or even no historical data. NBeats model (Oreshkin et al., 2020) designs an interpretable deep neural architecture and is flexible enough to solve a wide range of forecasting tasks. In Zhou et al. (2022), a frequency enhanced decomposed Transformer mechanism combing seasonal-trend decomposition is used to extract valuable information from both global profile and detailed structures. iTransformer (Liu et al., 2024a) design an inverted version of the Transformer architecture for time series forecasting. This approach alleviates the dilemma of traditional Transformer in time series analysis and allows for a better capture of multivariate correlations.

2.5. Time series foundation models

Inspired by the success of large models, time series foundation models based on large-scale pre-training have emerged as the prominent research focus in the time series field. Zhou et al. (2023) propose a novel framework for time series analysis based on the pre-trained language model. It freezes the pre-trained block and handles different time series tasks through fine-tuning. Goswami et al. (2024) propose a family of open-source foundation models for general-purpose time series analysis. It curates the Time Series Pile and pre-train the model based on transformer architectures on this corpus. Timer (Liu et al., 2024b) is a large-scale time series model grounded in a generative Transformer architecture. It is pre-trained on a unified dataset (UTSD) and can be fine-tuned for different downstream time series tasks. TimerXL (Liu et al., 2025) is a decoder-only transformer for unified time series forecasting. A novel mechanism based on causal self-attention is designed to improve the model's capability.

3. Proposed algorithm

We begin by presenting a structural overview of the proposed model. Each major module will be introduced in the following sections. As presented in Fig. 1, TTL-TS consists of two major modules: the Transformer-based Representation Learning module (TRL) and the Transfer Learning module (TLM). TRL aims to handle data imputation and forecasting for time series with missing values. TLM aspires to be suited to more complex and general scenarios, where data imputation and forecasting are performed with limited time series data.

3.1. Transformer-based representation learning module (TRL)

TRL aims to derive a model that performs imputation and forecasting for time series bearing missing values. Specifically, the proposed TRL is built on the Transformer encoder architecture, and incorporates a seasonal-trend decomposition mechanism to help further exploit the overall characteristics of time series. Formally, we consider a time series dataset $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\} \in R^{T\times D}$, where $\boldsymbol{x}_t = \{x_t^0, x_t^1, \ldots, x_t^{D-1}\} \in R^D$ represents the tth step observation with D dimensions. Each x_t^d is associated with a missing mask $\boldsymbol{M}_t^d$, where $\boldsymbol{M}_t^d = 1$ if x_t^d is observed and $\boldsymbol{M}_t^d = 0$ otherwise. Time series imputation aims to predict the missing values by using the observed values in $\boldsymbol{X}$. Time series forecasting is the task of predicting future values based on past observations. It can be seen as a particular case of imputation, where the imputation region spans across the last H time steps of each series. For example, $\boldsymbol{X}_{t+1:t+H}$ means the set of H observations from time step $t+1$ to $t+H$ in $\boldsymbol{X}$. We additionally mask $\boldsymbol{X}_{t+1:t+H}$ and predict the values of $\boldsymbol{X}_{t+1:t+H}$ based on $\boldsymbol{X}_{1:t}$.

Notably, in forecasting tasks, the entire pipeline strictly adheres to causal constraints, and no future information is used. Specifically, in the input masking phase, the future values $\boldsymbol{X}_{t+1:t+H}$ of each sample are additionally masked (i.e., $\boldsymbol{M}_{t+1:t+H} = 0$). During the training phase, the input-visible set only contains x_t^d where $M_t^d = 1$. Seasonal-trend decomposition and the associated reconstruction loss are computed exclusively on the input sequence $\boldsymbol{X}_{1:t}$, without any future values. Source domain construction and normalization statistics are derived from the training set. No future test values are used. At inference time, only observed values are fed into the trained model. No future information leaks into the prediction.

Model Architecture: Inspired by the superior performance of Transformer in time series fields, TRL leverages the Transformer encoder architecture to learn meaningful representations for downstream tasks with missing values. Akin to standard Transformer encoder, TRL starts by embedding original time series to a d_{model}-dimensional vector space:

$$e = Embed_{\theta_{emb}}(concat(\boldsymbol{X}, \boldsymbol{M})) \tag{1}$$

where the original time series $\boldsymbol{X}$ and its missing mask $\boldsymbol{M}$ are concatenated as the input. The $Embed_{\theta_{emb}}$ is given by a linear transformation with parameter θ_{emb}. Sinusoidal position encoding is used to capture the sequential order of time series:

$$\begin{cases} PE(pos, 2i) = \sin\left(\dfrac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \\ PE(pos, 2i+1) = \cos\left(\dfrac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \end{cases} \tag{2}$$

$$e' = e + PE \tag{3}$$

where pos, i respectively represent the time step position and the dimension, e' is the final embedding including position information. Then, similar to Du et al. (2023), the e' is passed through a stack of layers that include multi-head self-attention and layer normalization to attain the

corresponding encoding. A feed-forward network made up of two linear transformations with ReLU activate function is utilized to learn the final encoding $\mathbf{z}$:

$$FFN(x) = ReLU\left(\boldsymbol{W}_1 x + b_1\right)\boldsymbol{W}_2 + b_2 \tag{4}$$

$$\mathbf{z} = \left\{ FFN\left(MultiHeadAttention(e^{'})\right)\right\}^{L} \tag{5}$$

$$\tilde{\mathbf{z}} = F_z(\mathbf{z}) = \boldsymbol{W}_z\mathbf{z} + b_z \tag{6}$$

$$\hat{\boldsymbol{X}} = F_{\tilde{\mathbf{z}}}(\tilde{\mathbf{z}}) = \boldsymbol{X} \odot \boldsymbol{M} + (1 - \boldsymbol{M}) \odot \tilde{\mathbf{z}} \tag{7}$$

In Eq. (4), $\boldsymbol{W}_1 \in R^{d_{model} \times d_{ffn}}$, $\boldsymbol{W}_2 \in R^{d_{ffn} \times d_{model}}$, $b_2 \in R^{d_{model}}$. In Eq. (5), L is the number of stacking layers. In Eq. (6), encoding $\mathbf{z}$ is fed to a linear output layer with parameters $\boldsymbol{W}_z \in R^{d_{model} \times D}$ and $b_z \in R^D$ to obtain the learned representation $\tilde{\mathbf{z}}$. In Eq. (7), we generate the completed imputation $\hat{\boldsymbol{X}}$ by retaining the observed values of original $\boldsymbol{X}$ and replacing the missing part in $\boldsymbol{X}$ with $\tilde{\mathbf{z}}$.

To better exploit the complex temporal patterns of time series, we leverage the idea of seasonal-trend decomposition used in Wu et al. (2022) to assist the training of TRL. Since in real-world scenes, trend component is usually coupled with complex periodic patterns, it may be hard to be captured by fixed window average pooling.

To ameliorate this limitation, we utilize a mixture of auto-regressive experts to explore the underlying trend signal (Woo et al., 2022). It consists of K auto-regressive experts, where $K = \lfloor \log_2\left(\frac{T}{2}\right) \rfloor$. Each expert is implemented as a 1D convolution. To capture multi-scale temporal patterns, the kernel sizes of K experts are different. Specifically, the ith expert employs a convolution kernel of size 2^i. Then the final trend representation is generated through average-pooling of all experts' outputs. Convolution weights of all experts are trained end-to-end with the TRL module (i.e., minimizing the combined loss function in Eq. (17)). Formally, we have:

$$\boldsymbol{X}^{trend} = Avgpool\left(conv(x, 2^i)\right), i = 1, 2, \ldots K \tag{8}$$

$$\boldsymbol{X}^{season} = \boldsymbol{X} - \boldsymbol{X}^{trend} \tag{9}$$

where K is the number of auto-regressive experts, $\boldsymbol{X}^{trend}$, $\boldsymbol{X}^{season}$ denote the trend and seasonal part of original time series $\boldsymbol{X}$. Ideally, the seasonal and trend components $\hat{\boldsymbol{X}}^{season}$,$\hat{\boldsymbol{X}}^{trend}$ extracted from the final learned representation $\tilde{z}$ should be close to $\boldsymbol{X}^{trend}$,$\boldsymbol{X}^{season}$. Detailed training procedure are introduced in the following subsection.

Training Procedure: To well train the imputation and forecasting model on time series with missing values, we design three learning tasks: the Observed Values Reconstruction Task (OVRT), the Artificially Masked Values Reconstruction Task (AMRT) and the Seasonal-Trend Representation Reconstruction Task (STRT).

The OVRT puts emphasis on the reconstruction of observed values. It forces the model to well simulate the distribution of observed data. Inspired by the idea in Du et al. (2023), we use Mean Absolute Error (MAE) to evaluate the reconstruction loss between the observed values and the reconstructions:

$$L_{OVRT} = MAE\left(\boldsymbol{X}, \hat{\boldsymbol{X}}, M\right) = \frac{\sum_{t=1}^{T}\sum_{d=1}^{D}\left|\left(x_t^d - \hat{x}_t^d\right) \odot \boldsymbol{M}_t^d\right|}{\sum_{t=1}^{T}\sum_{d=1}^{D}\boldsymbol{M}_t^d} \tag{10}$$

One disadvantage of OVRT is that it only focuses on the observed values while ignores the missing points. To alleviate this problem, AMRT is designed to strengthen the prediction capability of missing values. AMRT artificially masks a proportion of observed points and forces the model to make accurate predictions on these extra masked values. This urges the model to attend to contextual information in time series, and encourages the model to learn inner-dependencies between variables. Formally, AMRT loss is defined as follows:

$$L_{AMRT} = MAE\left(\boldsymbol{X}, \hat{\boldsymbol{X}}, \tilde{\boldsymbol{M}}\right) = \frac{\sum_{t=1}^{T}\sum_{d=1}^{D}\left|\left(x_t^d - \hat{x}_t^d\right) \odot \tilde{\boldsymbol{M}}_t^d\right|}{\sum_{t=1}^{T}\sum_{d=1}^{D}\tilde{\boldsymbol{M}}_t^d} \tag{11}$$

where $\tilde{\boldsymbol{M}}$ is the artificially missing mask. $\tilde{\boldsymbol{M}}_t^d = 1$ if x_t^d is artificially masked and $\tilde{\boldsymbol{M}}_t^d = 0$ otherwise.

Both OVRT and AMRT consider the reconstruction performance from the perspective of holistic time series. STRT is designed to optimize the reconstruction loss more finely from the standpoint of seasonal-trend components. Inspired by the idea in Wang et al. (2022), we respectively use autocorrelation and temporal correlation to measure the seasonal and trend characteristics similarity between actual and reconstructed series. Since standard autocorrelation and temporal correlation estimators are only directly applicable to regularly sampled time series, we make some changes to reorient them to the scene with irregularly sampling (i.e. with missing values). Formally, the autocorrelation is defined as follows:

$$\rho(\boldsymbol{X}, \kappa) = \sum_{t_i + \kappa < \max(T_I)}\left(\boldsymbol{X}_{t_i} - \overline{\boldsymbol{X}}\right)\left(\boldsymbol{X}_{S(t_i+\kappa)} - \overline{\boldsymbol{X}}\right) \tag{12}$$

where $\overline{\boldsymbol{X}}$ is the mean of time series values, T_I is the set of time steps where the values $\boldsymbol{X}_{T_I}$ are not missing. $\max\left(T_I\right)$ $(\min\left(T_I\right))$ is the maximal (minimal) time step in T_I. $\kappa \in [0, \max\left(T_I\right) - \min\left(T_I\right)]$ is a generalized lag value. The main difference between Eq. (12) and standard autocorrelation is the selective function $S\left(t_i + \kappa\right)$. It projects an arbitrary time step $t_i + \kappa$ to the closest time step t' where $\boldsymbol{X}_{t'}$ is not missing. For example, there is an irregular time series $x = [0.13, 0, 0, 0.31, 0.35, 0, 0.41]$, where values are missing at time steps 1, 2 and 5. When calculating the lag $\kappa = 2$ autocorrelation at position $t_i = 0$, the conventional autocorrelation measuring method requires comparing x_0 and x_2, while x_2 is missing. For this situation, $S(t_i + \kappa)$ finds the nearest time step $t' = 3$ where x_3 is not missing. Then the autocorrelation calculation uses the observed value at $t' = 3$ for the lag $\kappa = 2$ at position $t_i = 0$, enabling autocorrelation computation on irregular data.

Based on Eq. (12), the autocorrelation loss is defined as follows:

$$L_{seasonal} = \sum_{\kappa \in [0, \max(T_I) - \min(T_I)]} \left\| \rho(\boldsymbol{X}^{season}, \kappa) - \rho\left(\hat{\boldsymbol{X}}^{season}, \kappa\right)\right\| \tag{13}$$

Eq. (13) reflects the autocorrelation similarity of seasonal patterns between actual and reconstructed time series. A smaller $L_{seasonal}$ indicates a greater similarity in the seasonal characteristic between the actual and reconstructed series.

To further depict the property in trend patterns, temporal correlation is defined to measure the simultaneity of rising or falling tendency between actual and reconstructed trend:

$$\varrho\left(\boldsymbol{X}, \hat{\boldsymbol{X}}\right) = \frac{\sum_{t_i \in T_I}\left(\boldsymbol{X}_{t_i} - \boldsymbol{X}_{S'(t_i)}\right)\left(\hat{\boldsymbol{X}}_{t_i} - \hat{\boldsymbol{X}}_{S'(t_i)}\right)}{\sqrt{\sum_{t_i \in T_I}\left(\boldsymbol{X}_{t_i} - \boldsymbol{X}_{S'(t_i)}\right)}\sqrt{\sum_{t_i \in T_I}\left(\hat{\boldsymbol{X}}_{t_i} - \hat{\boldsymbol{X}}_{S'(t_i)}\right)}} \tag{14}$$

$$L_{trend} = \varrho\left(\boldsymbol{X}^{trend}, \hat{\boldsymbol{X}}^{trend}\right) \tag{15}$$

In Eq. (14), T_I has the same definition as that in Eq. (12). To accommodate the situation of time series with missing values, a selective function S' is defined. It projects time step t_i to the closest time step t_i', where $t_i' < t_i, t_i' \in T_I$, and $\boldsymbol{X}_{t_i'}$ is observed. The STRT loss is defined as follows:

$$L_{STRT} = L_{seasonal} + \lambda L_{trend} \tag{16}$$

As shown in Eq. (16), autocorrelation similarity (i.e., Eq. (13)) aims to measure how well $\hat{\boldsymbol{X}}^{season}$ reproduces the periodic regularity of $\boldsymbol{X}^{season}$. This ensures the accurate reconstruction of cyclic characteristics in imputation task. Temporal correlation (i.e., Eq. (14)) aims to quantify the simultaneity of rising/falling tendencies between $\hat{\boldsymbol{X}}^{trend}$ and $\mathbf{X}^{trend}$, which preserves the underlying long-term trajectory between imputed data and original data.

The imputation and forecasting model is trained by optimizing the final loss:

$$L\left(\boldsymbol{X}, \hat{\boldsymbol{X}}, \boldsymbol{M}\right) = L_{OVRT} + L_{AMRT} + L_{STRT} \tag{17}$$

In summary, these three learning tasks operate complementarily to enhance the model's ability of temporal feature extraction. The OVRT loss focuses on reconstructing actual observed values, providing a strong foundational fit to the available information. The AMRT loss prioritizes the reconstruction of artificially masked values, enabling the model to learn robust representations for missing observations. Both OVRT and AMRT losses address the reconstruction at the global time-series level, this holistic perspective lacks explicit constraints for capturing the inherent patterns underlying temporal data. To alleviate this limitation, the STRT loss disentangles the series into seasonal and trend representations, guiding the model to capture interpretable and physics-informed (i.e., periodic variations and trend evolution) representations.

The main difference of the training procedure between our model and other existing imputation models is the incorporation of seasonal-trend reconstruction. It finely considers the reconstruction performance from seasonal and trend perspective, and assists the model to take note of the inherent seasonal-trend characteristics in time series. Effectiveness of this reconstruction loss is demonstrated in the Experiments.

3.2. Transfer learning module (TLM)

As mentioned above, the TRL module focuses on time series imputation and forecasting in the presence of missing values. Here we consider a more challenging scenario, where the imputation and forecasting tasks are implemented with limited and missing time series data. The TLM is designed to cope with the dilemma where only limited time series data are available. Specifically, we first construct a related source domain to help train the source model. Then, a transfer learning framework is designed to explore useful knowledge from the trained source model, aiding in the training of the target model. The detailed procedure is explained below.

Source domain and source model construction: Different from existing TL tasks that the related source domain is known and available, TLM needs to construct a source dataset which is connected to the target domain based on limited target time series. Specifically, according to TRL module, we use the model trained with L_{OVRT} and L_{AMRT} as the data construction model. Then we randomly mask several values and add Gaussian noise to the actual target data. The obtained noisy data is fed to the data construction model and the output is regarded as the source domain. Since the source data is constructed based on the target dataset, if the construction model performs exceptionally well on the target dataset, the samples it generates may be too similar to the existing target data, potentially limiting the acquisition of additional useful knowledge. Thus we simply train the construction model only with L_{OVRT} and L_{AMRT}. Although the constructed source domain is related to the target domain and may have additional effective knowledge, there is no guarantee that the two domains are closely associated at the data level. Hence, we use the source domain to train a source model and extract useful information through this source model rather than directly from the source data. The source model is constructed on the source domain according to the complete training procedure in TRL module.

Notably, though the source domain is constructed based on the limited target domain, it is not a simple corruption of the sparse target data. We minimize the risk of circularity and overfitting from the following aspects: 1) The construction model is deliberately restricted in training process with only OVRT loss and AMRT loss, which limits the model's capacity to memorize the target data. This restriction can to some extent prevent overfitting to sparse target samples; 2) Target data is perturbed with random masking and Gaussian noise. Novel patterns that are not present in the original target data can be provided, ensuring the generated source data diverges from the target data; 3) Critically, the generated source data are not directly used for feature extraction. Instead, we train a source model on them. This source model acts as a "filter " to distill useful patterns from the source domain. This operation provides higher-level insights that are more generalizable than raw source data, mitigating the chance of overfitting.

Transfer learning framework: As introduced above, though the generated source data is associated with the target domain, there is no assurance that the source and target domains are closely related. Directly aligning the target domain against the source domain may be counterproductive. Thus, here we provide a different perspective of the target data by feeding them to the trained source model. The output feature space should have useful knowledge for the target task and we regard it as the newly source domain. TLM aims to promote the target model's performance by aligning the newly source domain and the target domain. We implement the alignment in two ways: useful features selection and intermediate domains construction. The former one aims to encourage the learning of valid source features. The latter one offers a different alignment perspective by using the constructed intermediate domains. Detailed introductions are exhibited as follows.

Useful features selection: Since the newly source domain offers a different viewpoint of target data, the source feature spaces may include valid knowledge for the target task. Mimicking the source features might be helpful for the training of target model. Formally, we denote the trained source model mentioned above as G_S. Model trained on target domain is denoted as G_T. $f_S^l = G_S^l(x), f_T^l = G_T^l(x), f_S^l, f_T^l \in R^{d_{model} \times c}$ respectively represent source and target hidden features of the lth stacking layer. The original feature alignment task can be formulated as follows:

$$L_\alpha = \sum\nolimits_l \left\| f_S^l - f_T^l \right\| \tag{18}$$

However, as a result of the difference between source and target tasks, not all the source features are favorable to the training of target model. A weighting strategy is designed to select useful features automatically. Different weights are assigned to different features according to their effects on the target task. Eq. (19) shows the modified feature alignment task:

$$L_\alpha = \sum\nolimits_l \sum\nolimits_c w_c^l \left\| \left(f_S^l\right)_c - \left(f_T^l\right)_c \right\|^2 \tag{19}$$

where $\left(f_S^l\right)_c, \left(f_T^l\right)_c$ respectively represent the cth channel of source and target features in the lth layer, w_c^l is the corresponding weight. Since different source features show different importance for target task, we predict the weight by using a simple network F_θ which takes source features as the input. This network consists of an average pooling layer, a fully-connected (FC) layer and a softmax normalization layer:

$$w_c^l = F_\theta\left(f_S^l\right) = softmax\left(FC\left(AvgPool\left(f_S^l\right)\right)\right) \tag{20}$$

where θ represents the parameters of the network, w_c^l is the learned feature weight, and $\sum_c w_c^l = 1$.

Intermediate domains construction: As demonstrated in Dai et al. (2024), if the target domain is not closely related to the source domain, directly transferring knowledge between these two domains can be difficult. There exists a special path that potentially connects the source and target domains. Intermediate domains existing along this path can reflect some inter-domain connection characterizations. Therefore, we aim to minimize the discrepancy between source and target data by virtue of intermediate domains. Inspired by the idea of Dai et al. (2024), we construct intermediate domains by linearly mixing the source and target features with adaptively updated weighting coefficients $[w_S, w_T]$. As a primary component of Intermediate domains construction, how to attain appropriate weighting coefficients is considerable.

Here we predict the weighting coefficients by using a designed network $F_{\theta_{\text{inter}}}^{\text{inter}}$ with input of source and target features. It contains an average pooling layer, a FC layer, a Multi-Layer Perceptron (MLP) layer and a softmax normalization layer. Specifically, we respectively feed $f_{\text{ave\_}S}^l$ and $f_{\text{ave\_}T}^l$ into the FC layer. Summation of the outputs are then fed into the MLP layer. The output is finally transformed as probabilistic importance scores through the softmax normalization layer.

Specifically, in each batch, n source features and n target features are paired at random to form n source-target pairs. For each feature pair $\left(f_S^l, f_T^l\right)$, an average pooling operation is added to transform each feature into a $1 \times c$ dimensional vector, which is denoted as $\left(f_{ave\_S}^l, f_{ave\_T}^l\right)$.

Here f_S^l, f_T^l have the same meanings as that mentioned above. Then we respectively feed $f_{ave\_S}^l$ and $f_{ave\_T}^l$ into a fully-connected layer *FC*. Summation of the outputs are then fed into a multi-layer perception (MLP) to predict the weighting coefficients:

$$\begin{aligned}\left[w_S^l, w_T^l\right] &= F_{\theta_{inter}}^{inter}\left(f_S^l, f_T^l\right) \\ &= softmax\left(MLP\left(FC\left(f_{avg\_S}^l\right) + FC\left(f_{avg\_T}^l\right)\right)\right)\end{aligned} \tag{21}$$

$$f_{inter}^l = w_S^l \cdot f_S^l + w_T^l \cdot f_T^l \tag{22}$$

where θ_{inter} represents the parameters of the designed sub-network, $\left[w_S^l, w_T^l\right] \in R^2, w_S^l + w_T^l = 1$. w_S^l, w_T^l respectively denote the weighting coefficient assigned to source and target features of the lth layer. In Eq. (22), f_{inter}^l represents the intermediate domain, which is obtained by linearly mixing the source and target features. $w_S(w_T)$ controls the relevance between the intermediate domain and source (target) domain. The source-target alignment task is then reformulated to proportionally minimizing distances between the intermediate domains and source (target) domain:

$$L_{\alpha\_inter} = \sum_l w_S^l \cdot \left\| f_S^l - f_{inter}^l \right\|^2 + w_T^l \cdot \left\| f_T^l - f_{inter}^l \right\|^2 \tag{23}$$

In Eq. (23), if f_{inter}^l is closer to f_S^l than f_T^l, more penalization is allocated to the distance between f_{inter}^l and f_S^l (i.e., $w_S^l > w_T^l$).

To effectively transfer knowledge from the source task to the target task, Useful features selection and Intermediate domains construction are employed. The former one pays attention on the importance of each source feature. The latter one concerns about the knowledge embedded in the intermediate domains along the path connecting the source and target domains. The final loss used to train the target model is given as follows:

$$L_{total} = L_{ori} + L_\alpha + L_{\alpha\_inter} \tag{24}$$

$$L_{ori} = w_T^l \cdot L_{tgt} + \left(1 - w_T^L\right) \cdot L_{inter} \tag{25}$$

$$\begin{cases} L_{tgt} = L\left(\boldsymbol{X}^T, \hat{\boldsymbol{X}}^T, \boldsymbol{M}\right) \\ L_{inter} = L\left(\boldsymbol{X}^T, \hat{\boldsymbol{X}}^{inter}, \boldsymbol{M}\right) \\ \hat{\boldsymbol{X}}^{inter} = F_{\tilde{\mathbf{z}}}\left(F_z\left(f_{inter}^L\right)\right) \end{cases} \tag{26}$$

In Eq. (24), L_{ori} is the original imputation loss. L is the reconstruction loss defined in Eq. (17). $\boldsymbol{X}^T, \hat{\boldsymbol{X}}^T$ respectively represent the actual and reconstructed target domains. $\hat{\boldsymbol{X}}^{inter}$ is the intermediate domain obtained by feeding f_{inter}^L to the layers defined in Eqs. (6)-(7). The weighting coefficients are also used to measure the intermediate domain's impact on the target task. L_α and L_{α_inter} optimize the alignment task, which is utilized to assist the training of target model with limited target data. Detailed training procedure is exhibited below.

In summary, the "Useful features selection" strategy is theoretically grounded in representation learning. It treats the source features as a complementary perspective of the target data that may be underrepresented in the sparse target samples. Selecting the complementary source features via Eq. (19) can enhance the target model's ability of capturing temporal patterns.

The "Intermediate domains construction" strategy is built on the following theoretical foundation: When the source and target domains are not closely related, direct knowledge transfer often suffers from negative transfer. To alleviate this problem, this strategy constructs an intermediate space that bridges the two domains. It can help capture characteristics critical to target task and mitigate negative transfer.

Training procedure: The primary objective of the target model is to achieve high performance on imputation and forecasting task with missing and limited target data. To alleviate the target data scarcity challenge, L_α and L_{α_inter} are used to encourage the learning of useful source knowledge. The training procedure needs to learn three networks' parameters: the parameter φ of the target model, the parameter θ of the sub-network F_θ (defined in Eq. (20)), and the parameter θ_{inter} of the sub-network $F_{\theta_{inter}}^{inter}$ (defined in Eq. (21)). Borrowing the idea proposed in Jang et al. (2019), we update the parameters by using the following scheme:

1) Update φ to minimize L_α and L_{α_inter} for P times;
2) Update φ to minimize L_{ori} once;
3) Given parameter φ of the target model, then update θ and θ_{inter} to minimize L_{ori}.

As shown above, the target model and two sub-networks are trained jointly by alternatively updating the parameters φ and θ, θ_{inter}. First, the target model's parameter φ is updated by minimizing L_{total}. Next, with the parameter φ fixed, the parameters of sub-networks are updated by minimizing L_{ori}. Specifically, in the first step, φ is learned based on the alignment task. It promotes the learning of useful source knowledge. In the second step, the reconstruction loss is used to further train the parameter φ for the imputation and forecasting task. In the third step, given the target model parameter φ, the sub-networks parameters F_θ and $F_{\theta_{inter}}^{inter}$ are updated by minimizing L_{ori}. Detailed process of our proposed method is exhibited in Algorithm 1.

4. Experimental results

In this section, we evaluate the effectiveness of the proposed TTL-TS for time series imputation and forecasting task under the challenges of missing values and data scarcity.

4.1. Evaluation metric

To evaluate the practicability of the proposed method in imputation and forecasting tasks, we utilize the Mean Absolute Error (MAE) metric, Root Mean Square Error (RMSE) metric and Mean Relative Error (MRE) metric to assess the performance of the model. Following the setting in Du et al. (2023), the imputation errors are computed on the values indicated by mask:

$$MAE\left(x, \hat{x}, \boldsymbol{M}\right) = \frac{\sum_{d=1}^{D}\sum_{t=1}^{T}\left|\left(x - \hat{x}\right) \odot \boldsymbol{M}\right|_t^d}{\sum_{d=1}^{D}\sum_{t=1}^{T}\mathbf{M}_t^d} \tag{27}$$

$$MRE\left(x, \hat{x}, \boldsymbol{M}\right) = \frac{\sum_{d=1}^{D}\sum_{t=1}^{T}\left|\left(x - \hat{x}\right) \odot \boldsymbol{M}\right|_t^d}{\sum_{d=1}^{D}\sum_{t=1}^{T}\left|x \odot \boldsymbol{M}\right|_t^d} \tag{28}$$

$$RMSE\left(x, \hat{x}, \boldsymbol{M}\right) = \sqrt{\frac{\sum_{d=1}^{D}\sum_{t=1}^{T}\left(\left(\left(x - \hat{x}\right) \odot \boldsymbol{M}\right)^2\right)_t^d}{\sum_{d=1}^{D}\sum_{t=1}^{T}\boldsymbol{M}_t^d}} \tag{29}$$

where $\boldsymbol{M}$ is the missing mask, $\hat{x}, x$ respectively denote the predicted and actual data.

4.2. Experimental settings

In the experiment, the model architecture (i.e., TRL module, TLM module) and training loss functions are identical for both forecasting and imputation tasks. The key differences between them lie in two aspects: 1) Input masking strategy: The method for constructing artificial masks is the same for forecasting and imputation tasks. But for forecasting task, we additionally mask the last H time steps (the target to predict) in the input sequence; 2) Evaluation difference: For imputation task, metrics (MAE, RMSE, MRE) are calculated between the full reconstructed sequence and the ground truth. For forecasting task, metrics are computed only on the last H time steps.

In the imputation task, the window sizes for Physio2012, Air-Quality, Electricity, ILI and Exchange are respectively set as 48, 24, 100, 48 and 48 in TTL-TS. In the forecasting task, the window sizes for Physio2012, ILI, Exchange and Wind are respectively set as 48, 48, 48 and 100. During training process, TTL-TS uses the same batch size 64 for both forecasting and imputation tasks.

Algorithm 1 TTL-TS algorithm.

Input: Limited target time series $\boldsymbol{X}^T$ with missing values;
Output: Results of target time series imputation or forecasting.

Define missing mask $\boldsymbol{M}$ of $\boldsymbol{X}^T$;
procedure TRL:
  for $epoch = 1 \rightarrow epoch_{TRL}$ **do**
    Feed $\boldsymbol{X}^T$ and $\boldsymbol{M}$ to the target model G_T and obtain the reconstructed data $\hat{\boldsymbol{X}}^T$ according to Eqs. (1)-(7);
    Train target model G_T by optimizing Eq. (17).
  end for
end procedure TRL
procedure TLM:
  Construct source domain and source model G_S;
  for $epoch = 1 \rightarrow epoch_{TRL}$ **do**
    Useful features selection:
      Predict the weighting coefficients of source features according to Eq. (20);
      Construct the corresponding alignment objective according to Eq. (19);
    Intermediate domains construction:
      Predict weighting coefficients for the construction of intermediate domains according to Eq. (21);
      Construct intermediate features according to Eq. (22);
      Construct the source (target)-intermediate features alignment objective according to Eq. (23);
      Update target model parameters φ and two sub-networks parameters θ, θ_{inter} according to the training scheme introduced above.
  end for
end procedure TLM
Return: Feed target data to the trained target model to obtain the imputation or forecasting results.

4.3. Time series imputation

Datasets: We run imputation experiments on five datasets. The first one is PhysioNet2012 Mortality Prediction Challenge (Physio2012) dataset[1] (Goldberger et al., 2020). It consists of clinical multivariate time series extracted from intensive care unit records. According to different patients, there are up to 37 variables. We process the dataset following the procedure in Che et al. (2018), each sample is processed to hourly time series with 48 time steps. The processed dataset is very sparse and has 80% missing values. Similar to Du et al. (2023), we randomly divide the dataset into 80% training set and 20% test set. Then 20% samples randomly selected from the training set are regarded as the validation set. To simulate the data scarcity situation, we randomly select 10% samples from the training set to form the training set in practice. To measure the imputation performance, we randomly mask 10% observed values of the validation set and the test set and regard them as ground truth for validation and test.

The second one is Air-Quality dataset[2] (Zhang et al., 2017). It records the hourly sampled air pollutants from 12 stations in Beijing. Following Du et al. (2023), data from 2013/03/01 to 2017/02/28 are selected as the whole data. There are 132 features in total by aggregating all the variables of 12 stations. In the original dataset, there are 1.6% missing values. To increase difficulty, we artificially add 60% extra missing values. Similar to Du et al. (2023), data from 2014/01-2014/10 are selected as the validation set. Data from 2013/03-2013/12 are used as the test set. Data from 2014/11-2017/02 are used as the original training set. Analogous to Physio2012, to simulate the situation of data scarcity, we randomly select 10% samples from the training set to form the practical training set. We respectively mask 10% observed values in the validation and test sets and use them as the ground truth for evaluation.

The third dataset is the Electricity dataset[3] (Dua & Graff, 2017). It records the electricity consumption of 370 clients. We use the data from 2011/11 to 2012/08 as the validation set and use the data from 2011/01 to 2011/10 as the test set. Data from 2012/09 to 2014/12 are used as the original training set. Since there are no missing values in original electricity dataset, we artificially eliminate 60% observed values in the dataset. Similar to the first two datasets, we randomly select 10% samples from the original training set to form the practical training set. The artificially added missing values in the test set are used for imputation performance evaluation.

The fourth dataset is the Influenza-like illness (ILI) dataset.[4] It collects the weekly proportion of patients with influenza-like symptoms from the Centers for Disease Control and Prevention. We firstly preprocess ILI following the method in Challu et al. (2022). Similar to Electricity dataset, ILI has no missing values. We artificially mask 60% observed values. According to Challu et al. (2022), we randomly split the dataset into original training/validation/test sets according to 70%, 10%, 20%. Since the number of samples in original training set is quite small, we randomly select 50% samples from it to construct the practical training set. The artificially masked values in test data are used for evaluation.

The fifth dataset is the Exchange dataset.[5] It contains the records of daily exchange rates of eight currencies from 1990 to 2016. We artificially mask 60% observed values of the data. Then we use the same method in Physio2012 dataset to construct the original training/validation/test sets. For the same reason as mentioned in ILI dataset, we randomly select 50% samples from the original training set to construct the practical training set.

In summary, we use five datasets for imputation experiments. PhysioNet2012 has 80% naturally missing data and no artificially masked data. Air-Quality has 1.6% naturally missing data and 60% artificially masked data in the training set. Electricity, ILI and Exchange contain no naturally missing data and we artificially mask 60% data in their training sets. We simulate data scarcity by constructing reduced training subsets. For PhysioNet2012, Air-Quality, and Electricity, 10% samples are randomly selected from their original training sets. To account for the limited scale of ILI and Exchange, we randomly retain 50% samples from the original training sets.

Experimental Results: In this part, we evaluate the imputation performance of the proposed method. To make a comprehensive comparison, we adopt several outstanding imputation methods, including BRITS (Cao et al., 2018), MRNN (Yoon et al., 2018), SAITS (Du et al., 2023), CSDI (Tashiro et al., 2021), SpectraNet (Challu et al., 2022) as benchmarks. We implement the compared approaches via the codes provided by authors.

Tables 1–3 respectively exhibit the MAE, RMSE and MRE results of different methods on five datasets. All results are averaged over 3 independent runs with different random seeds. The proportions of missing values and selected training samples in the original training data are introduced in Datasets part. From Tables 1–3 we can find that the proposed TTL-TS outperforms all other baselines on MAE and MRE. The RMSE metric of SAITS is better than that of TTL-TS on dataset Air-Quality. While on other datasets, RMSE values of TTL-TS are also the best. By analyzing the ablations of TTL-TS, we can easily find that TRL with the trend-season reconstruction part behaves better than that without this part. On some datasets, e.g., Electricity, ILI, Exchange, performance of TRL with fine-tune is worse than that of TRL. That may be because the source model differs considerably from target model. But the TTL-TS also has satisfactory performance. Thus, we can roughly conclude that both the trend-season reconstruction part and the designed

[1] https://www.physionet.org/content/challenge-2012
[2] https://archive.ics.uci.edu/ml/datasets/Beijing+Multi-Site+Air-Quality+Data
[3] https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014
[4] https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html
[5] https://github.com/laiguokun/multivariate-time-series-data

Table 1
Imputation MAE results of different methods on five datasets.

	Physio2012	Air-Quality	Electricity	ILI	Exchange
BRITS	0.2824 ± 0.0013	0.2962 ± 0.0041	1.1917 ± 0.0058	0.9361 ± 0.0029	0.0504 ± 0.0023
MRNN	0.5338 ± 0.0004	0.3126 ± 0.0029	1.2974 ± 0.0110	1.0268 ± 0.0030	0.1403 ± 0.0009
SAITS	0.2608 ± 0.0017	0.2470 ± 0.0047	1.0104 ± 0.0094	0.7395 ± 0.0043	0.0499 ± 0.0010
CSDI	0.3199 ± 0.0008	0.2607 ± 0.0015	1.0118 ± 0.0101	1.0317 ± 0.0027	0.1246 ± 0.0024
SpectraNet	0.3192 ± 0.0015	0.2727 ± 0.0023	1.3462 ± 0.0089	0.9730 ± 0.0037	0.0569 ± 0.0018
TRL/wo trend-season[a]	0.2499 ± 0.0010	0.2719 ± 0.0020	1.0024 ± 0.0064	0.7176 ± 0.0026	0.0447 ± 0.0012
TRL[a]	0.2469 ± 0.0011 (Δ: −1.20%)	0.2614 ± 0.0008 (Δ: −3.86%)	0.9821 ± 0.0075 (Δ: −2.03%)	0.6717 ± 0.0032 (Δ: −6.39%)	0.0441 ± 0.0014 (Δ: −1.34%)
TRL + finetune[a]	0.2353 ± 0.0007	0.2472 ± 0.0017	0.9907 ± 0.0072	0.7463 ± 0.0025	0.0449 ± 0.0008
TTL-TS	**0.2312** ± 0.0005	**0.2460** ± 0.0014	**0.9669** ± 0.0086	**0.6451** ± 0.0027	**0.0402** ± 0.0004

[a] Ablations of TTL-TS. **TRL**: model trained without TLM module; Δ: the relative performance improvement ratio of TRL over TRL/wo trend-season; **TRL/wo trend-season**: The variant of TRL where only L_{OVRT} and L_{AMRT} are used (i.e., L_{STRT} is removed); **TRL + finetune**: Replace TLM module with the fine-tune technique.

Table 2
Imputation RMSE results of different methods on five datasets.

	Physio2012	Air-Quality	Electricity	ILI	Exchange
BRITS	0.5594 ± 0.0016	0.6043 ± 0.0039	1.7604 ± 0.0115	1.3354 ± 0.0118	0.0758 ± 0.0030
MRNN	0.7752 ± 0.0007	0.5847 ± 0.0020	1.9064 ± 0.0133	1.4498 ± 0.0120	0.1713 ± 0.0007
SAITS	0.5014 ± 0.0012	**0.5182** ± 0.0031	1.6842 ± 0.0083	1.0794 ± 0.0127	0.0621 ± 0.0009
CSDI	0.6654 ± 0.0009	0.8735 ± 0.0010	8.1830 ± 0.0140	1.5521 ± 0.0115	0.1608 ± 0.0018
SpectraNet	0.5306 ± 0.0016	0.5702 ± 0.0026	1.9740 ± 0.0139	1.3765 ± 0.0088	0.0714 ± 0.0011
TRL/wo trend-season	0.4976 ± 0.0011	0.5464 ± 0.0023	1.6300 ± 0.0117	1.0623 ± 0.0124	0.0552 ± 0.0014
TRL	0.4942 ± 0.0012 (Δ: −0.68%)	0.5348 ± 0.0006 (Δ: −2.12%)	1.5744 ± 0.0127 (Δ: −3.41%)	1.0300 ± 0.0132 (Δ: −3.04%)	0.0544 ± 0.0010 (Δ: −1.45%)
TRL + finetune	0.4680 ± 0.0003	0.5326 ± 0.0018	1.6345 ± 0.0119	1.1258 ± 0.0129	0.0559 ± 0.0009
TTL-TS	**0.4657** ± 0.0001	0.5320 ± 0.0016	**1.5477** ± 0.0121	**1.0137** ± 0.0138	**0.0503** ± 0.0013

Table 3
Imputation MRE results of different methods on five datasets.

	Physio2012	Air-Quality	Electricity	ILI	Exchange
BRITS	40.99% ± 0.0023	41.85% ± 0.0021	63.82% ± 0.0111	52.93% ± 0.0100	6.08% ± 0.0029
MRNN	77.48% ± 0.0002	44.17% ± 0.0017	69.48% ± 0.0125	58.06% ± 0.0018	16.92% ± 0.0010
SAITS	37.86% ± 0.0011	34.91% ± 0.0022	54.11% ± 0.0117	41.82% ± 0.0113	6.01% ± 0.0013
CSDI	46.44% ± 0.0006	36.84% ± 0.0010	54.19% ± 0.0120	58.34% ± 0.0019	15.03% ± 0.0021
SpectraNet	46.18% ± 0.0020	38.54% ± 0.0019	72.09% ± 0.0128	55.02% ± 0.0017	6.86% ± 0.0022
TRL/wo trend-season	36.28% ± 0.0012	38.42% ± 0.0011	53.68% ± 0.0123	40.58% ± 0.0008	5.40% ± 0.0016
TRL	35.84% ± 0.0014 (Δ: −1.21%)	36.93% ± 0.0005 (Δ: −3.88%)	52.60% ± 0.0115 (Δ: −2.01%)	37.98% ± 0.0013 (Δ: −6.40%)	5.32% ± 0.0010 (Δ: −1.48%)
TRL + finetune	34.15% ± 0.0009	34.94% ± 0.0021	53.05% ± 0.0098	42.20% ± 0.0018	5.42% ± 0.0008
TTL-TS	**33.56%** ± 0.0007	**34.76%** ± 0.0019	**51.78%** ± 0.0104	**36.48%** ± 0.0015	**4.85%** ± 0.0017

transfer learning framework (i.e. TLM module) are beneficial to the imputation task under the challenge of data scarcity (i.e., only limited training data are available).

Figs. 2–3 describe the imputation performances of the proposed method intuitively. Due to space limitations, here we only present some of the visualization results.

Figs. 2 and 3 respectively exhibit a section of the results obtained by TTL-TS and each baseline on ILI dataset with dimension 7 and Exchange dataset with dimension 4. As shown in Fig. 2, compared to all the baselines, TTL-TS has a better fitting effect on the real data. In Fig. 3, we can also find that TTL-TS also lead to superior result. Fitting performance of TTL-TS is obviously better than all other baselines. By analyzing Tables 1–3 and Figs. 2–3, we can roughly conclude that the proposed TTL-TS excels at the time series imputation task under the challenges of data scarcity and missing values.

4.4. Time series forecasting

As mentioned in Section 3, Time series forecasting can be regarded as a particular case of imputation, where the imputation region spans across the last H time steps of each series. Thus, in the forecasting task, for each series, we additionally mask the last H time steps.

Datasets: We run forecasting experiments on four datasets: Physio2012 dataset, Exchange dataset, ILI dataset and Wind dataset[6] (Dane, 2015). The first three datasets are the same as that mentioned in Section 4.2. The Wind dataset is commonly utilized in time series forecasting task. It contains 28 countries' wind energy which is daily estimated from 1986 to 2015. We preprocess Wind dataset according to Moreno-Pino et al. (2023). Since ILI, Exchange and Wind datasets have no missing values, we artificially mask 60% observed values. Then rolling window strategy, which is widely applied in time series forecasting task, is utilized to construct forecasting samples. The proportions of original training/validation/test sets of Physio2012, Exchange and ILI datasets are the same as that used in the imputation task. To simulate the data scarcity situation, the practical training set is made up of 10% samples selected from the original training set. For Wind dataset, we randomly divide the dataset into 80% original training set and 20% test set. Then we randomly select 20% samples from the original training set to construct the validation set. 10% samples are selected from the rest original training set to form the practical training set. Forecasting horizons of Physio2012, ILI, Exchange and Wind are respectively 5, 24, 24, 24.

In forecasting experiments, PhysioNet2012, ILI and Exchange adopt the same natural missing ratio and artificial masking ratio as those used in the imputation experiments. Wind has no naturally missingness and we artificially mask 60% data in the training set. To simulate data scarcity situation, the training set size is reduced to 10% of the original for all datasets.

Experimental Results: In this part, we evaluate the forecasting performance of the proposed method. To make a comprehensive comparison, we use several state-of-the-art forecasting methods as bench-

[6] https://www.kaggle.com/sohier/

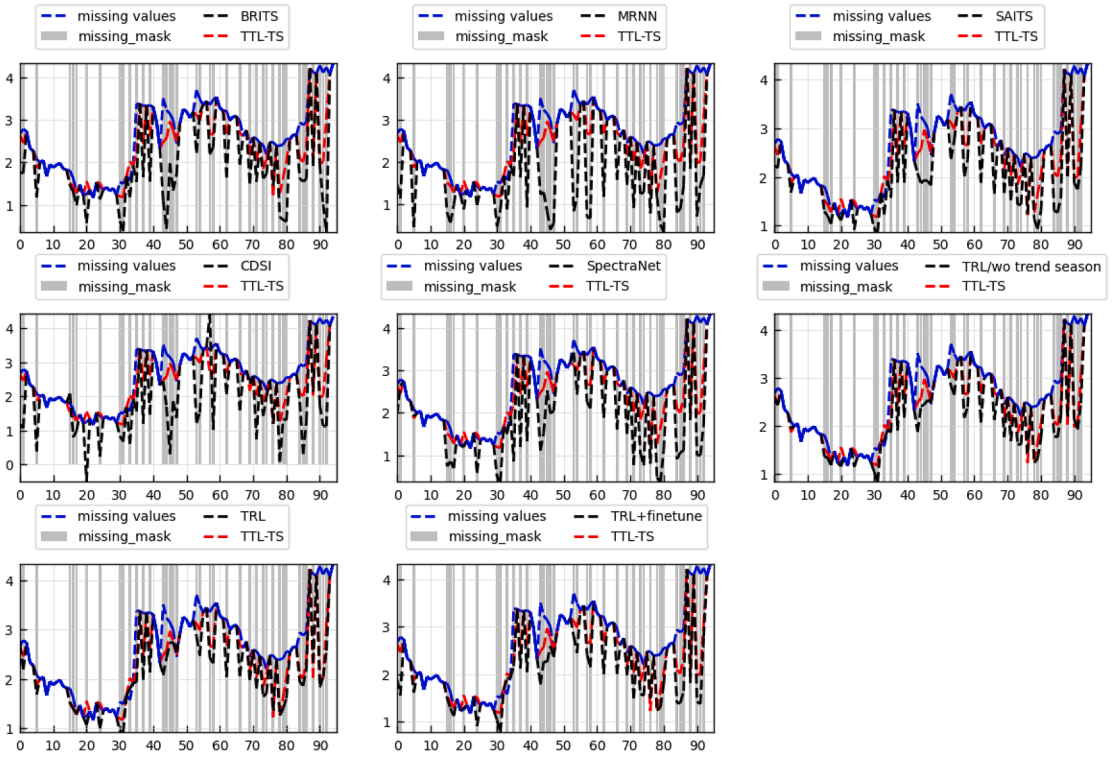


Fig. 2. Results of TTL-TS and each baseline vs. true data on ILI dataset with dimension 7.

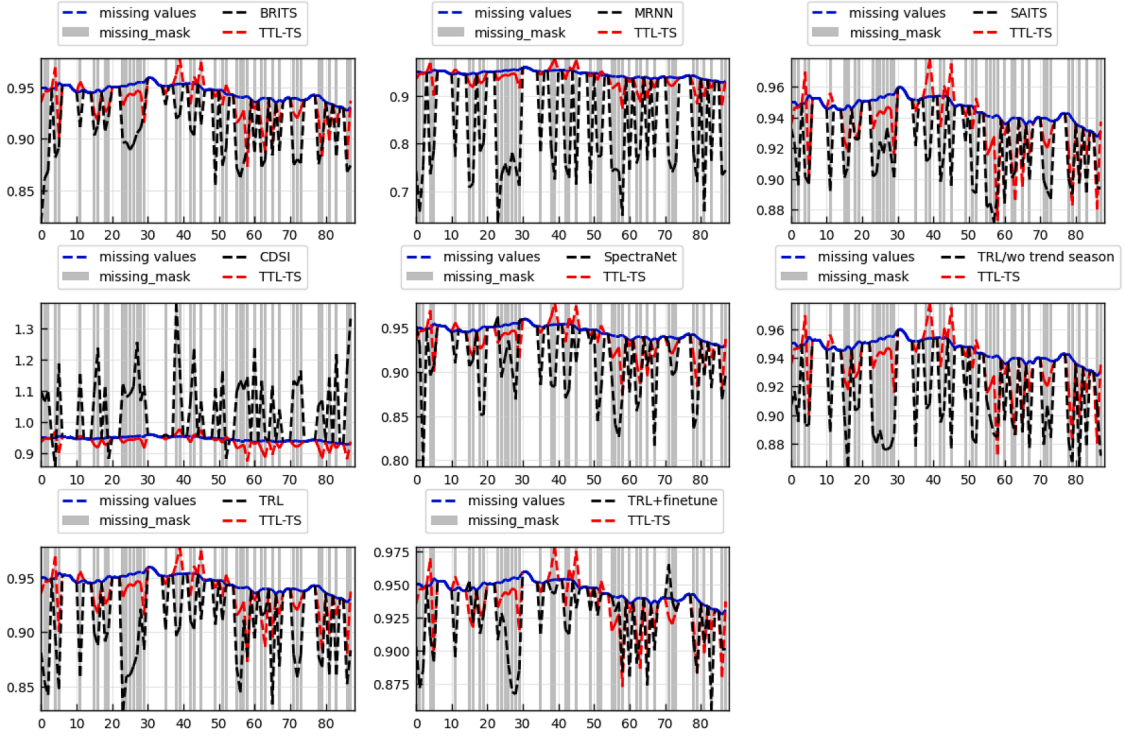


Fig. 3. Results of TTL-TS and each baseline vs. true data on exchange dataset with dimension 4.

marks, including DeepAR (Salinas et al., 2020), NBeats (Oreshkin et al., 2020), FEDFormer (Zhou et al., 2022), SAITS (Du et al., 2023), and SpectraNet (Challu et al., 2022). The first three are widely recognized for their effectiveness in time series forecasting, while SpectraNet and SAITS stand out for their versatility, handling both time series imputation and forecasting tasks (Challu et al., 2022; Du et al., 2023). We implement the compared approaches via the codes provided by authors.

Tables 4–6 present the forecasting accuracy of different methods on four datasets. We find that the proposed TTL-TS achieves satisfactory performances on all the datasets. It's evident that imputation methods, such as SpectraNet and SAITS, significantly outperform traditional

Table 4
Forecasting MAE results of different methods on four datasets.

	Physio2012 (h-5)[b]	ILI (h-24)[b]	Exchange (h-24)[b]	Wind (h-24)[b]
DeepAR	0.5563 ± 0.0025	1.5889 ± 0.0020	0.3736 ± 0.0013	0.6777 ± 0.0009
NBeats	0.5182 ± 0.0007	1.5533 ± 0.0023	0.4102 ± 0.0031	0.6724 ± 0.0010
FEDFormer	0.5582 ± 0.0015	1.8360 ± 0.0009	0.2926 ± 0.0038	0.7941 ± 0.0010
SpectraNet	0.4343 ± 0.0008	1.4919 ± 0.0016	0.0665 ± 0.0012	0.6062 ± 0.0021
SAITS	0.4274 ± 0.0010	1.3388 ± 0.0033	0.0436 ± 0.0015	0.5939 ± 0.0022
TRL/wo trend-season[a]	0.4257 ± 0.0011	1.3350 ± 0.0003	0.0492 ± 0.0028	0.6202 ± 0.0004
TRL[a]	0.4112 ± 0.0005 (Δ: −3.41%)	1.3010 ± 0.0007 (Δ: −2.55%)	0.0532 ± 0.0011 (Δ: 8.13%)	0.5660 ± 0.0015 (Δ: −8.74%)
TRL + finetune[a]	0.4109 ± 0.0003	1.3352 ± 0.0009	0.0587 ± 0.0010	0.5673 ± 0.0012
TTL-TS	**0.4018** ± 0.0006	**1.2698** ± 0.0013	**0.0411** ± 0.0005	**0.5604** ± 0.0016

[a] Ablations of TTL-TS.
[b] Forecasting horizons: e.g. 'h-5' means that the forecasting horizon is 5

Table 5
Forecasting RMSE results of different methods on four datasets.

	Physio2012 (h-5)	ILI (h-24)	Exchange (h-24)	Wind (h-24)
DeepAR	0.7852 ± 0.0033	2.3301 ± 0.0025	0.3918 ± 0.0020	0.8100 ± 0.0012
NBeats	0.7342 ± 0.0016	2.2795 ± 0.0038	0.4732 ± 0.0031	0.8027 ± 0.0015
FEDFormer	0.7712 ± 0.0020	2.4546 ± 0.0019	0.3758 ± 0.0043	0.9667 ± 0.0018
SpectraNet	0.6737 ± 0.0019	2.1710 ± 0.0026	0.0847 ± 0.0017	0.7655 ± 0.0014
SAITS	0.6530 ± 0.0021	2.0622 ± 0.0035	0.0527 ± 0.0028	0.7571 ± 0.0040
TRL/wo trend-season	0.6491 ± 0.0019	2.0263 ± 0.0013	0.0635 ± 0.0042	0.7811 ± 0.0022
TRL	0.6412 ± 0.0014 (Δ: −1.22%)	1.9724 ± 0.0018 (Δ: −2.66%)	0.0636 ± 0.0026 (Δ: 0.16%)	0.7290 ± 0.0016 (Δ: −6.67%)
TRL + finetune	0.6459 ± 0.0010	2.0081 ± 0.0021	0.0715 ± 0.0015	0.7334 ± 0.0020
TTL-TS	**0.6315** ± 0.0015	**1.9569** ± 0.0020	**0.0515** ± 0.0010	**0.7229** ± 0.0013

Table 6
Forecasting MRE results of different methods on four datasets.

	Physio2012 (h-5)	ILI (h-24)	Exchange (h-24)	Wind (h-24)
DeepAR	80.40% ± 0.0019	84.25% ± 0.0031	45.19% ± 0.0022	77.54% ± 0.0010
NBeats	74.91% ± 0.0015	82.36% ± 0.0029	49.63% ± 0.0018	76.93% ± 0.0011
FEDFormer	123.61% ± 0.0022	174.77% ± 0.0015	44.38% ± 0.0052	189.03% ± 0.0046
SpectraNet	61.96% ± 0.0012	79.47% ± 0.0025	8.05% ± 0.0022	69.36% ± 0.0028
SAITS	61.71% ± 0.0018	71.14% ± 0.0022	5.28% ± 0.0037	67.95% ± 0.0037
TRL/wo trend-season	61.46% ± 0.0019	70.94% ± 0.0016	5.94% ± 0.0033	70.96% ± 0.0035
TRL	59.37% ± 0.0007 (Δ: −3.40%)	69.13% ± 0.0019 (Δ: −2.55%)	6.43% ± 0.0025 (Δ: 8.25%)	64.76% ± 0.0024 (Δ: −8.74%)
TRL + finetune	59.32% ± 0.0009	70.95% ± 0.0018	7.10% ± 0.0023	64.91% ± 0.0020
TTL-TS	**58.01%** ± 0.0011	**67.47%** ± 0.0034	**4.97%** ± 0.0017	**64.12%** ± 0.0022

forecasting methods like DeepAR, NBeats and FEDFormer. That may be because traditional forecasting approaches rely heavily on sufficient training data. Missing values and data scarcity vastly diminish their forecasting capability.

By analyzing the ablations of TTL-TS, we find that TRL with the trend-season reconstruction part performs better than that without this part on most of the datasets. Incorporation of the TLM module also has obviously positive impact on the forecasting task, which is particularly obvious on Exchange dataset. On this dataset, though the forecasting performance of TRL is inferior to that of SAITS and TRL/wo trend-season, the incorporation of TLM significantly increases the forecasting accuracy.

Fig. 4 intuitively describes the forecasting performances of the proposed TTL-TS and several effective baselines. Due to space limitations, here we only present some of the visualization results.

The picture above displays the results for the 6th dimension of the 56th sample in Exchange dataset. The picture below displays the results for 4th dimension of the 5th sample in Exchange dataset. We can find that the forecasting result of TTL-TS consistently meets or exceeds the performances of other methods. TRL/wo trend-season and SAITS also have satisfactory forecasting results. While compared to these methods, TRL and SpectraNet perform slightly worse. This conclusion is coincident with that in Tables 4–6.

As shown above, on most of the datasets, the TLM module and trend-season reconstruction part are beneficial to the imputation/forecasting performance of TTL-TS. Above results demonstrate the effectiveness of the proposed TTL-TS method in time series imputation and forecasting tasks with limitations of missing values and data scarcity.

4.5. Additional experiments

4.5.1. Comparison with time series foundation models

To further evaluate the performance of the proposed method for time series forecasting and imputation tasks under data-missing and data-scarce conditions, we expand the comparative analysis by incorporating several state-of-the-art time series foundation models. These models are pre-trained on extensive datasets and demonstrate strong transferability to few-shot/zero-shot scenarios.

For the forecasting task, Timer (Liu et al., 2024b),[7] MOMENT (Goswami et al., 2024),[8] and TimerXL (Liu et al., 2025)[9] are selected as benchmark methods. For the imputation task, Timer, MOMENT and GPT4TS (Zhou et al., 2023)[10] are used as benchmark methods. (Since TimerXL lacks explicit support for imputation tasks, here we substituted it with GPT4TS).

Specifically, according to the instructions provided in the papers of these foundation models, we fine-tune the pre-trained base models (official checkpoints provided in the papers) with limited target samples.

[7] https://github.com/thuml/Large-Time-Series-Model
[8] https://huggingface.co/AutonLab/MOMENT-1-small
[9] https://huggingface.co/thuml/timer-base-84m
[10] https://github.com/DAMO-DI-ML/NeurIPS2023-One-Fits-All

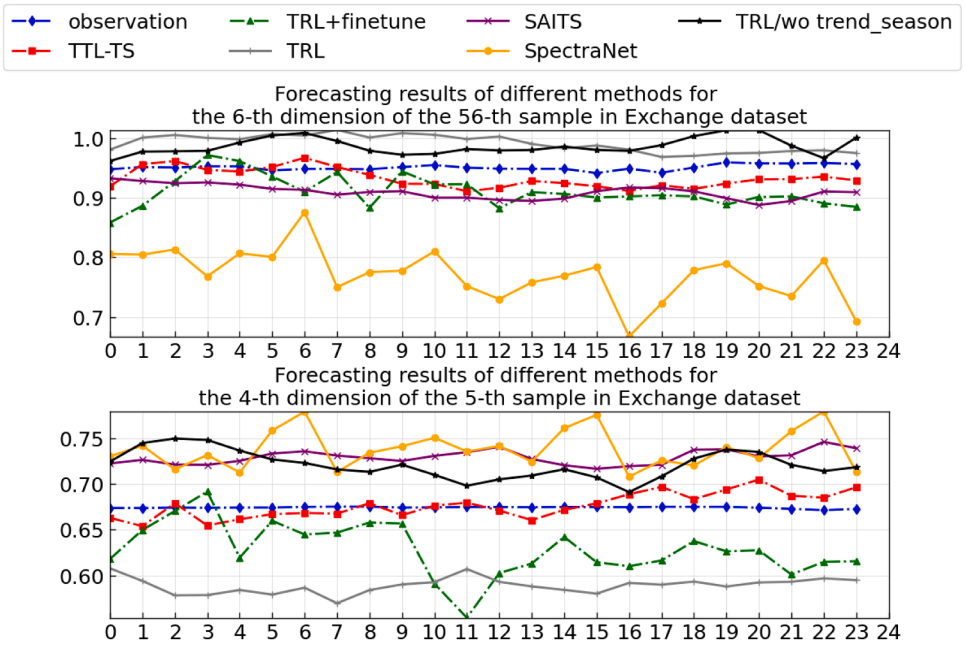


Fig. 4. Forecasting results of different methods for the 6th dimension of the 56th sample in exchange dataset (ABOVE), the 4th dimension of the 5th sample in exchange dataset (BELOW).

Table 7
Forecasting results of TTL-TS and time series foundation models.

	Metrics	Physio2012 (h-5)	ILI (h-24)	Exchange (h-24)	Wind (h-24)
Timer	MAE	0.4380 ± 0.0010	1.2840 ± 0.0024	0.0417 ± 0.0011	0.5702 ± 0.0022
	RMSE	0.6600 ± 0.0015	1.8062 ± 0.0038	0.0421 ± 0.0012	0.7929 ± 0.0032
	MRE	63.23% ± 0.0012	68.22% ± 0.0020	5.04% ± 0.0014	65.24% ± 0.0025
MOMENT	MAE	0.4279 ± 0.0016	1.3026 ± 0.0021	0.0433 ± 0.0007	0.5689 ± 0.0019
	RMSE	0.6511 ± 0.0012	1.8324 ± 0.0027	0.0437 ± 0.0010	0.7915 ± 0.0029
	MRE	61.77% ± 0.0009	69.21% ± 0.0025	5.23% ± 0.0014	65.08% ± 0.0017
TimerXL	MAE	0.4183 ± 0.0013	1.2747 ± 0.0032	**0.0409** ± 0.0008	0.5671 ± 0.0015
	RMSE	0.6442 ± 0.0021	**1.7931** ± 0.0029	**0.0413** ± 0.0010	0.7905 ± 0.0023
	MRE	60.39% ± 0.0016	67.79% ± 0.0026	**4.95%** ± 0.0013	64.88% ± 0.0019
TTL-TS	MAE	**0.4018** ± 0.0006	**1.2698** ± 0.0013	0.0411 ± 0.0005	**0.5604** ± 0.0016
	RMSE	**0.6315** ± 0.0015	1.9569 ± 0.0020	0.0515 ± 0.0010	**0.7229** ± 0.0013
	MRE	**58.01%** ± 0.0011	**67.47%** ± 0.0034	4.97% ± 0.0017	**64.12%** ± 0.0022

All foundation models are fine-tuned on the same preprocessed data as TTL-TS, with the following settings: The learning rate of the foundation models is set as 1e-4. Batch size is selected from the range $\{32, 64, 128\}$. Training is performed for 100 epochs with early stopping based on validation loss (patience = 5). Hyperparameters were tuned on the validation set independently for each dataset.

Tables 7, 8 respectively present the forecasting and imputation performances of different methods under data-missing and data-scarce conditions. All results are averaged over 3 independent runs with different random seeds.

From Table 7, we find that time series foundation models exhibit competitive results compared to conventional time series forecasting baselines (shown in Tables 4–6). Large-scale pre-training on diverse datasets enables these models to learn universal temporal patterns and dependencies across domains. Overall, TTL-TS performs competitively and is superior to foundation models on both Wind and PhysioNet2012 datasets. That maybe because TTL-TS is explicitly optimized for the joint challenge of data scarcity and missingness. Under this protocol, limited fine-tuning data may degrade the feature extraction capability of foundation models. Missing values may disrupt data continuity, making models harder to infer underlying dynamics. Nevertheless, foundation models such as TimerXL still maintain strong performance on specific datasets and metrics (e.g., Exchange).

Table 8 shows the imputation performances of models. From Table 8, all foundation models achieve competitive results. Timer presents the strongest overall performance among foundation models. TTL-TS also performs competitively. For example, TTL-TS achieves better MAE and MRE on PhysioNet2012, ILI and Exchange. That maybe because significant domain shift exists between the target data and the pre-trained data, which limits effective fine-tuning of large pre-trained models. On Air-Quality and Electricity datasets, Timer achieves lower MAE and MRE, indicating that when temporal patterns are compatible with pre-training distributions, foundation models may remain highly competitive.

Consequently, TTL-TS is preferable when extreme training-sample scarcity and high missing rates coexist. When the target domain is well aligned with pre-training data, foundation models may achieve superior results.

4.5.2. *Comparison with other prominent transfer learning strategies*

To further measure the effectiveness of the TLM module, we replace TLM with two prominent and model-Agnostic transfer learning framework strategies: Domain-Adversarial Neural Network (DANN) (Ganin et al., 2016) and Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017). All other model architectures are unchanged. DANN is a classic domain-adversarial based method which is widely used for domain

Table 8
Imputation results of TTL-TS and time series foundation models.

	Metrics	Physio2012	Air-Quality	Electricity	ILI	Exchange
Timer	MAE	0.2550 ± 0.0011	**0.2391** ± 0.0017	**0.9571** ± 0.0068	0.7129 ± 0.0019	0.0416 ± 0.0008
	RMSE	**0.4398** ± 0.0023	0.5606 ± 0.0020	1.6610 ± 0.0097	1.2797 ± 0.0072	**0.0440** ± 0.0016
	MRE	37.01% ± 0.0009	**33.78%** ± 0.0011	**51.25%** ± 0.0088	40.31% ± 0.0056	5.02% ± 0.0015
MOMENT	MAE	0.2615 ± 0.0008	0.2474 ± 0.0019	0.9702 ± 0.0089	0.7428 ± 0.0023	0.0490 ± 0.0006
	RMSE	0.4510 ± 0.0012	0.5664 ± 0.0036	1.6695 ± 0.0130	1.2778 ± 0.0132	0.0515 ± 0.0015
	MRE	37.96% ± 0.0010	34.96% ± 0.0021	51.96% ± 0.0108	42.01% ± 0.0016	5.90% ± 0.0011
GPT4TS	MAE	0.2643 ± 0.0019	0.2551 ± 0.0017	0.9812 ± 0.0079	0.7721 ± 0.0031	0.0453 ± 0.0010
	RMSE	0.4535 ± 0.0034	0.5721 ± 0.0028	1.6763 ± 0.0124	1.4223 ± 0.0042	0.0478 ± 0.0018
	MRE	38.37% ± 0.0021	36.05% ± 0.0013	52.54% ± 0.0100	43.66% ± 0.0039	5.46% ± 0.0020
TTL-TS	MAE	**0.2312** ± 0.0005	0.2460 ± 0.0014	0.9669 ± 0.0086	**0.6451** ± 0.0027	**0.0402** ± 0.0004
	RMSE	0.4657 ± 0.0001	**0.5320** ± 0.0016	**1.5477** ± 0.0121	**1.0137** ± 0.0138	0.0503 ± 0.0013
	MRE	**33.56%** ± 0.0007	34.76% ± 0.0019	51.78% ± 0.0104	**36.48%** ± 0.0015	**4.85%** ± 0.0017

Table 9
Forecasting results of TTL-TS and TRL with two prominent transfer learning strategies.

	Metrics	Physio2012 (h-5)	ILI (h-24)	Exchange (h-24)	Wind (h-24)
TRL + DANN	MAE	0.4097 ± 0.0009	1.3249 ± 0.0016	0.0448 ± 0.0010	0.5675 ± 0.0019
	RMSE	0.6362 ± 0.0018	1.8638 ± 0.0024	**0.0452** ± 0.0017	0.7915 ± 0.0026
	MRE	59.15% ± 0.0010	70.40% ± 0.0029	5.43% ± 0.0015	64.92% ± 0.0030
TRL + MAML	MAE	0.4079 ± 0.0007	1.3212 ± 0.0012	0.4698 ± 0.0013	0.5650 ± 0.0015
	RMSE	0.6365 ± 0.0011	**1.8586** ± 0.0023	0.6318 ± 0.0023	0.7902 ± 0.0022
	MRE	58.89% ± 0.0008	70.20% ± 0.0033	5.68% ± 0.0018	64.64% ± 0.0027
TTL-TS	MAE	**0.4018** ± 0.0006	**1.2698** ± 0.0013	**0.0411** ± 0.0005	**0.5604** ± 0.0016
	RMSE	**0.6315** ± 0.0015	1.9569 ± 0.0020	0.0515 ± 0.0010	**0.7229** ± 0.0013
	MRE	**58.01%** ± 0.0011	**67.47%** ± 0.0034	**4.97%** ± 0.0017	**64.12%** ± 0.0022

Table 10
Imputation results of TTL-TS and TRL with two prominent transfer learning strategies.

	Metrics	Physio2012	Air-Quality	Electricity	ILI	Exchange
TRL + DANN	MAE	0.2361 ± 0.0015	**0.2428** ± 0.0035	0.9922 ± 0.0063	0.7311 ± 0.0028	0.0435 ± 0.0007
	RMSE	0.5150 ± 0.0013	0.5635 ± 0.0051	1.6838 ± 0.0096	1.2182 ± 0.0101	0.0459 ± 0.0016
	MRE	34.27% ± 0.0020	**34.32%** ± 0.0043	53.13% ± 0.0115	41.34% ± 0.0020	5.24% ± 0.0012
TRL + MAML	MAE	0.2342 ± 0.0009	0.2462 ± 0.0013	0.9854 ± 0.0073	0.6988 ± 0.0034	0.0424 ± 0.0010
	RMSE	**0.3474** ± 0.0010	0.5629 ± 0.0018	1.6792 ± 0.0127	1.2114 ± 0.0127	0.0447 ± 0.0021
	MRE	34.00% ± 0.0016	34.79% ± 0.0021	52.77% ± 0.0110	39.51% ± 0.0058	5.11% ± 0.0019
TTL-TS	MAE	**0.2312** ± 0.0005	0.2460 ± 0.0014	**0.9669** ± 0.0086	**0.6451** ± 0.0027	**0.0402** ± 0.0004
	RMSE	0.4657 ± 0.0001	**0.5320** ± 0.0016	**1.5477** ± 0.0121	**1.0137** ± 0.0138	**0.0503** ± 0.0013
	MRE	**33.56%** ± 0.0007	34.76% ± 0.0019	**51.78%** ± 0.0104	**36.48%** ± 0.0015	**4.85%** ± 0.0017

adaptation. MAML is a classic meta-learning framework widely applied to few-shot tasks. Tables 9 and 10 respectively present the forecasting and imputation results of different models. The experimental datasets are the same as that in Tables 1–6.

As shown in Tables 9, 10, TTL-TS outperforms TRL + DANN and TRL + MAML on most datasets for both forecasting and imputation tasks. That maybe because the source data are generated by simply-trained model and contains noise (as mentioned in "Source domain and source model construction" section). It maybe not closely associated with target task. DANN aims to mitigate domain shift by aligning feature distributions between source and target domains, while large domain divergence makes it hard to capture effective cross-domain representations. For MAML, the large domain divergence corrupts the task similarity assumption, degrading the model's knowledge transfer efficiency. The experiment validates the effectiveness of TLM module in time series forecasting and imputation tasks with data-scarce and data-missing.

4.5.3. Further empirical validation of the transfer learning mechanism

To test whether the performance gains of TTL-TS arise from knowledge transfer rather than circular exploitation of target-specific patterns, we conduct four sets of controlled experiments:

(1) **Comparison with Gaussian-noise augmentation without TLM (GN_w/o TLM)**: Gaussian noise is used to augment the training data without using the TLM module.
(2) **Comparison with teacher-student distillation (Teacher-Student-Distill)**: The source model is pre-trained on the same target-derived synthetic samples as in TTL-TS. Then it serves as the teacher to guide the target model's training via the standard distillation objective without TLM.
(3) **Evaluation with completely unrelated source domain**: A completely unrelated external dataset is used as the source domain. Specifically, we use Air-Quality as the unrelated source domain for the time series forecasting tasks, and Wind dataset as the unrelated source domain for the imputation tasks. To better validate the effectiveness of the TLM, two variants are tested:
  (3-1) **Unrelated Source with full TTL-TS (UR_src-TTL-TS)**: The full TTL-TS pipeline is applied with an unrelated external dataset as the source domain.;
  (3-2) **Unrelated Source with simple distillation (UR_src-Distill)**: The same unrelated source domain is used. But the TLM module of TTL-TS is substituted by a teacher-student distillation mechanism.

Table 11
Validation of the transfer learning mechanism on forecasting tasks.

	Metric	Physio2012(h-5)	ILI(h-24)	Exchange(h-24)	Wind(h-24)
GN_w/o TLM	MAE	0.4722 ± 0.0002	1.3821 ± 0.1168	0.0806 ± 0.0132	0.5727 ± 0.0023
	RMSE	0.6913 ± 0.0017	2.0811 ± 0.1674	0.1039 ± 0.0223	0.7384 ± 0.0001
	MRE	68.17% ± 0.0003	73.44% ± 0.0620	9.75% ± 0.0144	65.52% ± 0.0026
Teacher-Student-Distill	MAE	0.4745 ± 0.0013	1.3816 ± 0.0045	0.0451 ± 0.0010	0.5785 ± 0.0136
	RMSE	0.6966 ± 0.0022	2.0862 ± 0.0243	0.0586 ± 0.0002	0.7495 ± 0.0208
	MRE	68.51% ± 0.0019	73.41% ± 0.0024	5.46% ± 0.0012	66.19% ± 0.0156
UR_src-TTL-TS	MAE	0.4905 ± 0.0012	1.3525 ± 0.0120	0.0513 ± 0.0067	0.5841 ± 0.0098
	RMSE	0.7134 ± 0.0032	2.0536 ± 0.0203	0.0621 ± 0.0076	0.7478 ± 0.0104
	MRE	70.81% ± 0.0017	71.87% ± 0.0064	6.21% ± 0.0081	66.83% ± 0.0112
UR_src-Distill	MAE	0.5309 ± 0.0031	1.4721 ± 0.0140	0.0812 ± 0.0100	0.6015 ± 0.0017
	RMSE	0.7614 ± 0.0018	2.1638 ± 0.0091	0.0995 ± 0.0127	0.7663 ± 0.0026
	MRE	76.64% ± 0.0045	78.22% ± 0.0074	9.83% ± 0.0120	68.81% ± 0.0020
TTL_TS	MAE	**0.4018** ± 0.0006	**1.2698** ± 0.0013	**0.0411** ± 0.0005	**0.5604** ± 0.0016
	RMSE	**0.6315** ± 0.0015	**1.9569** ± 0.0020	**0.0515** ± 0.0010	**0.7229** ± 0.0013
	MRE	**58.01**% ± 0.0011	**67.47%** ± 0.0034	**4.97%** ± 0.0017	**64.12%** ± 0.0022
MMD/p-value		0.4590/$p \leq 0.01$	0.4806/$p \leq 0.01$	0.5966/$p \leq 0.01$	0.1436/$p \leq 0.01$

Table 12
Validation of the transfer learning mechanism on imputation tasks.

	Metric	Physio2012	Air-Quality	Electricity	ILI	Exchange
GN_w/o TLM	MAE	0.2626 ± 0.0011	0.3115 ± 0.0019	1.0253 ± 0.0180	0.7409 ± 0.0026	0.0449 ± 0.0012
	RMSE	0.5098 ± 0.0020	0.6910 ± 0.0031	1.6460 ± 0.0012	1.2423 ± 0.0070	0.0572 ± 0.0002
	MRE	38.12% ± 0.0015	44.01% ± 0.0026	54.91% ± 0.0096	38.27% ± 0.0036	5.42% ± 0.0015
Teacher-Student-Distill	MAE	0.2574 ± 0.0017	0.2528 ± 0.0020	1.0406 ± 0.0119	0.7186 ± 0.0189	0.0503 ± 0.0027
	RMSE	0.5079 ± 0.0015	0.4701 ± 0.0018	1.6271 ± 0.0057	1.2025 ± 0.0311	0.0599 ± 0.0034
	MRE	37.36% ± 0.0024	35.84% ± 0.0029	55.73% ± 0.0117	37.12% ± 0.0097	6.08% ± 0.0032
UR_src-TTL-TS	MAE	0.2507 ± 0.0026	**0.2110** ± 0.0016	1.0191 ± 0.0282	0.6721 ± 0.0074	**0.0386** ± 0.0064
	RMSE	0.4958 ± 0.0088	**0.3942** ± 0.0022	1.6166 ± 0.0061	1.1102 ± 0.0374	0.0509 ± 0.0093
	MRE	36.40% ± 0.0038	**29.91%** ± 0.0023	54.57% ± 0.0151	**34.72%** ± 0.0038	**4.65%** ± 0.0077
UR_src-Distill	MAE	0.2695 ± 0.0006	0.2151 ± 0.0005	1.0737 ± 0.0077	0.8584 ± 0.0048	0.0856 ± 0.0021
	RMSE	0.5149 ± 0.0027	0.4520 ± 0.0032	1.6706 ± 0.0158	1.4120 ± 0.0053	0.1043 ± 0.0016
	MRE	39.12% ± 0.0009	30.49% ± 0.0007	57.50% ± 0.0041	44.34% ± 0.0025	10.33% ± 0.0026
TTL_TS	MAE	**0.2312** ± 0.0005	0.2460 ± 0.0014	**0.9669** ± 0.0086	**0.6451** ± 0.0027	0.0402 ± 0.0004
	RMSE	**0.4657** ± 0.0001	0.5320 ± 0.0016	**1.5477** ± 0.0121	**1.0137** ± 0.0138	**0.0503** ± 0.0013
	MRE	**33.56%** ± 0.0007	34.76% ± 0.0019	**51.78%** ± 0.0104	36.48% ± 0.0015	4.85% ± 0.0017
MMD/p-value		0.4583/$p \leq 0.01$	0.7876/$p \leq 0.01$	0.7604/$p \leq 0.01$	0.6003/$p \leq 0.01$	0.5986/$p \leq 0.01$

1. **Discrepancy between target and constructed source data**: We use Maximum Mean Discrepancy (MMD) and permutation tests to validate the domain shift between the target domain and the constructed source domain. MMD quantifies the divergence between domains, where higher values indicate greater distributional disparity. The p-value, derived from a permutation test, measures the statistical significance of the observed distributional discrepancy under the null hypothesis that the target and constructed source domains share the same distribution. The p-value ≤ 0.01 indicates that the null hypothesis can be rejected at the 1% significance level, implying a statistically significant distributional shift.

Tables 11 and 12 respectively represent the results for time series forecasting task and imputation task. All results are averaged over 3 independent runs with different random seeds.

As shown in Tables 11–12, GN_w/o TLM underperforms TTL-TS across all datasets in both forecasting and imputation tasks. These results indicate that simply augmenting the target training data with Gaussian noise cannot replace the efficacy of TTL-TS, which validates the effectiveness of TLM module.

Teacher-Student-Distill also falls short of TTL-TS in time series forecasting tasks. In imputation task, Teacher-Student-Distill is competitive on Air-Quality with lower RMSE, but still inferior on others. The key distinction is that TTL-TS selectively transfers useful source features and constructs intermediate domains for gradual alignment in TLM module, whereas distillation mimics all teacher outputs. This confirms that the performance gains of TTL-TS over TRL are not attributable to simple knowledge distillation.

The unrelated-source experiments aim to test whether TTL-TS merely memorizes target-specific patterns. To better validate the effectiveness of the TLM, we divide this baseline into the following two variants: UR_src-TTL-TS and UR_src-Distill. From Tables 11–12, UR_src-TTL-TS outperforms UR_src-Distill across all datasets, demonstrating that TLM's transfer mechanisms can extract useful knowledge even from unrelated source domains.

Besides, TTL-TS with its target-derived constructed source domain outperforms UR_src-TTL-TS on most datasets, which is expected because the constructed source domain is more task-relevant than an unrelated external dataset in most situations. An interesting case arises on Air-Quality in the imputation task, where UR_src-TTL-TS outperforms the TTL-TS. That maybe because as multivariate environmental time series, though Wind and Air-Quality are unrelated at the task semantics level, there are similar structural properties between them. In addition, the comparison between UR_src-TTL-TS and UR-src-Distill on the same dataset shows that the gain is not merely due to using an external dataset. TLM remains necessary to extract transferable knowledge.

To further verify that the constructed source domain is statistically distinct from the target domain, we use the Maximum Mean

Table 13
Ablation results of each TLM component on forecasting tasks.

	Metric	Physio2012(h-5)	ILI(h-24)	Exchange(h-24)	Wind(h-24)
w/o_ufs	MAE	0.4941 ± 0.0082	1.4261 ± 0.0036	0.0510 ± 0.0135	0.5764 ± 0.0023
	RMSE	0.7191 ± 0.0110	2.1465 ± 0.0017	0.0632 ± 0.0155	0.7420 ± 0.0008
	MRE	71.33% ± 0.0119	75.78% ± 0.0019	6.16% ± 0.0163	65.95% ± 0.0026
w/o_idc	MAE	0.4795 ± 0.0027	1.3664 ± 0.0222	0.0473 ± 0.0063	0.5718 ± 0.0008
	RMSE	0.7048 ± 0.0026	2.0562 ± 0.0043	0.0576 ± 0.0069	0.7395 ± 0.0015
	MRE	0.6922 ± 0.0039	72.61% ± 0.0118	5.72% ± 0.0076	65.42% ± 0.0009
direct-align	MAE	0.4943 ± 0.0026	1.3820 ± 0.0732	0.0443 ± 0.0073	0.5753 ± 0.0012
	RMSE	0.7171 ± 0.0028	2.0765 ± 0.0898	0.0542 ± 0.0064	0.7443 ± 0.0055
	MRE	71.36% ± 0.0038	73.44% ± 0.0389	5.36% ± 0.0089	65.81% ± 0.0013
random-weights	MAE	0.4944 ± 0.0067	1.5123 ± 0.0483	0.0438 ± 0.0031	0.5759 ± 0.0024
	RMSE	0.7194 ± 0.0108	2.2155 ± 0.0381	0.0549 ± 0.0047	0.7465 ± 0.0019
	MRE	71.37% ± 0.0096	80.36% ± 0.0257	5.30% ± 0.0038	65.88% ± 0.0028
TTL_TS	MAE	**0.4018** ± 0.0006	**1.2698** ± 0.0013	**0.0411** ± 0.0005	**0.5604** ± 0.0016
	RMSE	**0.6315** ± 0.0015	**1.9569** ± 0.0020	**0.0515** ± 0.0010	**0.7229** ± 0.0013
	MRE	**58.01%** ± 0.0011	**67.47%** ± 0.0034	**4.97%** ± 0.0017	**64.12%** ± 0.0022

Table 14
Ablation results of each TLM component on imputation tasks.

	Metric	Physio2012	Air-Quality	Electricity	ILI	Exchange
w/o_ufs	MAE	0.2518 ± 0.0038	0.2485 ± 0.0042	1.0226 ± 0.0155	0.6658 ± 0.0121	0.0433 ± 0.0036
	RMSE	0.4976 ± 0.0010	0.4633 ± 0.0039	1.6698 ± 0.0018	1.0139 ± 0.0192	0.0561 ± 0.0049
	MRE	36.55% ± 0.0055	35.22% ± 0.0059	54.76% ± 0.0083	0.3765 ± 0.0289	5.23% ± 0.0044
w/o_idc	MAE	0.2565 ± 0.0014	0.2491 ± 0.0018	1.0293 ± 0.0419	0.6524 ± 0.0158	0.0411 ± 0.0007
	RMSE	0.5000 ± 0.0033	**0.4621** ± 0.0003	1.6247 ± 0.0112	**0.9630** ± 0.0154	0.0520 ± 0.0002
	MRE	37.24% ± 0.0020	35.31% ± 0.0026	55.12% ± 0.0225	0.3689 ± 0.0272	4.90% ± 0.0008
direct-align	MAE	0.2612 ± 0.0333	0.2516 ± 0.0024	1.0254 ± 0.0036	0.7048 ± 0.0169	0.0423 ± 0.0011
	RMSE	0.5164 ± 0.0391	0.4671 ± 0.0016	1.6351 ± 0.0057	1.0127 ± 0.0046	0.0542 ± 0.0026
	MRE	37.91% ± 0.0484	35.66% ± 0.0015	54.91% ± 0.0019	39.85% ± 0.0095	5.10% ± 0.0014
random-weights	MAE	0.2509 ± 0.0017	0.2537 ± 0.0031	1.0159 ± 0.0146	0.6703 ± 0.0393	0.0457 ± 0.0074
	RMSE	0.4963 ± 0.0017	0.4682 ± 0.0015	1.6408 ± 0.0274	0.9841 ± 0.0428	0.0589 ± 0.0113
	MRE	36.41% ± 0.0025	35.97% ± 0.0022	54.41% ± 0.0078	37.90% ± 0.0222	5.51% ± 0.0089
TTL_TS	MAE	**0.2312** ± 0.0005	**0.2460** ± 0.0014	**0.9669** ± 0.0086	**0.6451** ± 0.0027	**0.0402** ± 0.0004
	RMSE	**0.4657** ± 0.0001	0.5320 ± 0.0016	**1.5477** ± 0.0121	1.0137 ± 0.0138	**0.0503** ± 0.0013
	MRE	**33.56%** ± 0.0007	**34.76%** ± 0.0019	**51.78%** ± 0.0104	**36.48%** ± 0.0015	**4.85%** ± 0.0017

Discrepancy (MMD) and permutation tests to measure the discrepancy between the target domain and the constructed source domain. As shown in Tables 11–12, the MMD distances are all larger than zero, with all p-values ≤ 0.01. This statistically confirms significant distributional disparities across source and target domains. Thus, the distribution of the constructed source domain is distinct from that of the target domain, rather than being a near-duplicate.

4.5.4. Detailed ablation study on TLM components

To evaluate the independent contributions of each TLM component, we conduct ablation studies with four variants:

(1) **TTL-TS without useful features selection (w/o_ufs)**: Measure the contribution of "useful features selection" part.
(2) **TTL-TS without intermediate-domains construction (w/o_idc)**: Measure the contribution of "intermediate domains construction" part.
(3) **TTL-TS with direct source-target feature alignment only (direct-align)**: Use direct source-target alignment (Eq. (18)) without weighting learning (Eq. (20)) or intermediate domains construction part. This tests whether the TLM mechanisms can be replaced by naive direct alignment.
(4) **TTL-TS with random feature weights instead of learned weights (random-weights)**: The weights w_c^l in Eq. (20) are randomly initialized. Unlike the learnable weighting mechanism in TTL-TS that utilizes a neural network. This tests whether performance gains depend on learned, task-relevant weighting rather than arbitrary weighting.

Tables 13 and 14 respectively represent the ablation results for time series forecasting task and imputation task. All results are averaged over 3 independent runs with different random seeds.

As shown in Tables 13–14, removing useful features selection part (w/o_ufs) degrades the performance of TTL-TS in both forecasting and imputation tasks. These results confirm that not all source features are equally beneficial, and the useful features selection part is essential for selecting transferable knowledge.

Removing intermediate domain construction (w/o_idc) also degrades the performance of TTL-TS. For instance, w/o_idc yields a relative MAE increase of 19.33% on PhysioNet2012 dataset for forecasting, and 10.94% on PhysioNet2012 for imputation. Based on these results, we can infer that the intermediate domains construction part plays a valid role in bridging disparate domains.

Direct source-target alignment (direct-align) underperforms the full TTL-TS on most datasets. This indicates that naive direct alignment cannot adequately replace the weighting learning and intermediate domains construction.

Replacing learned weights with random weights (random-weights) leads to performance degradation on most datasets. This confirms that the learned weighting network is capable of identifying task-relevant features.

5. Conclusion

In this paper, we consider a more complex and general situation of time series, where data missing and training data scarcity are coexisting. To overcome these two challenges, a Transformer-based transfer learning algorithm (TTL-TS) is designed for time series imputation and forecasting task. Though many researches have been proposed respectively for data missing and data scarcity challenges, there is little research considering both of these two challenges.

The proposed TTL-TS has two major components: the TRL module and the TLM module. In TRL, a joint-optimization technique is designed to assess the distribution of missing values. It considers the reconstruction loss from both the holistic data aspect and the finer trend-season aspect. In TLM, a transfer learning framework is designed to alleviate the data scarcity problem. It focuses on capturing useful knowledge from the source model to assist the target task.

While the proposed TTL-TS demonstrates improvements in handling missing values and data scarcity in time series analysis, there exists several limitations: (1) In TLM module, the source domain is constructed via implicit association with the target domain. This may limit the source domain's ability to provide complementary information under extreme scenarios. (2) Although TLM mitigates data scarcity challenge through transfer learning, it does not fully consider the scenarios where significant heterogeneity or divergence exists between the source and target domains, which may degrade the generalization performance of the transfer learning framework. (3) Although TTL-TS effectively handles training data scarcity and missingness in both forecasting and imputation tasks, its performance should not be regarded as fully representative of real-world conditions. This is primarily because a substantial portion of the empirical evidence presented here is derived from controlled simulated settings rather than naturally occurring low-resource deployments. In the future work, we will explore solutions to these limitations.

Moreover, recent surveys on multi-fidelity optimization (Li & Li, 2026) and decomposition-based multi-objective evolutionary algorithms (Li, 2024) suggest promising directions for future extensions. Incorporating multi-fidelity methods could reduce the cost of tuning TRL/TLM hyperparameters or evaluating alternative source-construction strategies by combining cheap proxy runs with a limited number of high-fidelity training cycles.

CRediT authorship contribution statement

Rui Ye: Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing; **Yu Ding:** Resources, Writing – review & editing; **Jing Zhang:** Validation, Resources; **Yi-Heng Zhu:** Supervision, Funding acquisition, Project administration.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China under Grant 62402227; Fundamental Research Funds for the Central Universities under Grant YDZX2025024.

References

Alcaraz, J. M. L., & Strodthoff, N. (2022). Diffusion-based time series imputation and forecasting with structured state space models. arXiv preprint arXiv:2208.09399, .

Almeida, M. M., Almeida, J. D. S., Quintanilha, D. B. P., Junior, G. B., & Silva, A. C. (2025). A meta-learning based neural network and LSTM for univariate time series missing data imputation. *Applied Soft Computing*, *172*, 112845–112859.

Cao, W., Wang, D., & Li, J., et al. (2018). Brits: Bidirectional recurrent imputation for time series. *Advances in neural information processing systems*, *vol. 31*.

Casolaro, A., Capone, V., Iannuzzo, G., & Camastra, F. (2023). Deep learning for time series forecasting: Advances and open problems. *Information*, *14*(11), 598–633.

Challu, C., Jiang, P., Wu, Y. N., & Callot, L. (2022). SpectraNet: Multivariate forecasting and imputation under distribution shifts and missing data. arXiv preprint arXiv:2210.12515, .

Che, Z., Purushotham, S., & Cho, K., et al. (2018). Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, *8*(1), 6085–6097.

Chen, J., Ding, J., Tan, K. C., Qian, J., & Li, K. (2025). MBL-CPDP : A multi-objective bilevel method for cross-project defect prediction. *IEEE Transactions on Software Engineering*, *51*(8), 2305–2328.

Chen, L., Liu, H.-L., Tan, K. C., & Li, K. (2021). Transfer learning-based parallel evolutionary algorithm framework for bilevel optimization. *IEEE Transactions on Evolutionary Computation*, *26*(1), 115–129.

Chen, R., & Li, K. (2021). Transfer Bayesian optimization for expensive black-box optimization in dynamic environment. In *2021 IEEE International conference on systems, man, and cybernetics (SMC)* (pp. 1374–1379). IEEE.

Chen, R., & Li, K. (2023). Data-driven evolutionary multi-objective optimization based on multiple-gradient descent for disconnected pareto fronts. In *International conference on evolutionary multi-criterion optimization* (pp. 56–70). Springer.

Chen, Z., Ma, M., Li, T., Wang, H., & Li, C. (2023). Long sequence time-series forecasting with deep learning: A survey. *Information Fusion*, *97*, 1–36.

Dai, Y., Sun, Y., Liu, J., & Tong, et al. (2024). Bridging the source-to-target gap for cross-domain person re-identification with intermediate domains. *International Journal of Computer Vision*, *219*, 1–25.

Dane, S. (2015). 30 Years of European wind generation, [Online]. Available: https://www.kaggle.com/sohier/30-years-of-european-wind-generation.

Ding, N., Xu, Y., Tang, Y., Xu, C., Wang, Y., & Tao, D. (2022). Source-free domain adaptation via distribution estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 7212–7222).

Du, W., Côté, D., & Liu, Y. (2023). Saits: Self-attention-based imputation for time series. *Expert Systems with Applications*, *219*, 119619–119634.

Du, Y., Yang, H., Chen, M., Luo, H., Jiang, J., Xin, Y., & Wang, C. (2024). Generation, augmentation, and alignment: A pseudo-source domain based method for source-free domain adaptation. *Machine Learning*, *113*(6), 3611–3631.

Dua, D., & Graff, C. (2017). UCI machine learning repository, [Online]. Available: http://archive.ics.uci.edu/ml.

Fan, X., Li, K., & Tan, K. C. (2020). Surrogate assisted evolutionary algorithm based on transfer learning for dynamic expensive multi-objective optimisation problems. In *2020 IEEE Congress on evolutionary computation (CEC)* (pp. 1–8). IEEE.

Feng, L., Yang, Y., Tan, M., Zeng, T., Tang, H., Li, Z., Niu, Z., & Feng, F. (2024a). Adaptive multi-source domain collaborative fine-tuning for transfer learning. *PeerJ Computer Science*, *10*, 1–27.

Feng, S., Miao, C., Zhang, Z., & Zhao, P. (2024b). Latent diffusion transformer for probabilistic time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence* (pp. 11979–11987). (*vol. 38*).

Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning (ICML)* (pp. 1126–1135). PMLR.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., March, M., & Lempitsky, V. (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, *17*(59), 1–35.

Goldberger, L. A. A., Glass, L., & Hausdorff, J. M., et al. (2020). Physiobank, physiotoolkit, and physionet: Components of a new research resource for complex physiologic signals. *Circulation*, *101*(3), 215–220.

Goswami, M., Szafer, K., Choudhry, A., Cai, Y., Li, S., & Dubrawski, A. (2024). MOMENT: A family of open time-series foundation models. In *Forty-first international conference on machine learning, ICML*.

He, Y., Tao, L., & Zhang, Z. (2023). A transfer learning enhanced decomposition-based hybrid framework for forecasting multiple time-series. In *CCF conference on big data* (pp. 16–31). Springer.

Iman, M., Arabnia, H. R., & Rasheed, K. (2023). A review of deep transfer learning and recent advancements. *Technologies*, *11*(2), 40–54.

Jang, Y., Lee, H., Hwang, S. J., & Shin, J. (2019). Learning what and where to transfer. In K. Chaudhuri, & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning (ICML), USA* (pp. 3030–3039). PMLR (*vol. 97*).

Li, K. (2024). A survey of multi-objective evolutionary algorithm based on decomposition: Past and future. *IEEE Transactions on Evolutionary Computation*, *30*(3), 957–978.

Li, K., & Chen, R. (2022). Batched data-driven evolutionary multiobjective optimization based on manifold interpolation. *IEEE Transactions on Evolutionary Computation*, *27*(1), 126–140.

Li, K., Chen, R., & Yao, X. (2023). A data-driven evolutionary transfer optimization for expensive problems in dynamic environments. *IEEE Transactions on Evolutionary Computation*, *28*(5), 1396–1411.

Li, K., & Li, F. (2026). Multi-fidelity methods for optimization: A survey. *ACM Computing Surveys*, *58*(12), 1–38.

Li, K., Xiang, Z., Chen, T., & Tan, K. C. (2020a). BiLO-CPDP: Bi-level programming for automated model discovery in cross-project defect prediction. In *Proceedings of*

the 35th IEEE/ACM international conference on automated software engineering (pp. 573–584).

Li, K., Xiang, Z., Chen, T., Wang, S., & Tan, K. C. (2020b). Understanding the automated parameter optimization on transfer learning for cross-project defect prediction: An empirical study. In *Proceedings of the ACM/IEEE 42nd international conference on software engineering* (pp. 566–577).

Li, L., Du, B., & Wang, Y., et al. (2020c). Estimation of missing values in heterogeneous traffic data: Application of multimodal deep learning model. *Knowledge-Based Systems*, *194*, 105592–105605.

Liang, R., Hao, Q., Gao, Y., Liu, K., Jiang, L., Wang, P., & Yin, M. (2025). Imputation via domain adaptation: Rethinking variable subset forecasting from knowledge transfer. In *Proceedings of the 31st ACM SIGKDD conference on knowledge discovery and data mining* (pp. 1683–1694).

Liao, B., Meng, Y., & Monz, C. (2023). Parameter-efficient fine-tuning without introducing new latency. arXiv preprint arXiv:2305.16742, .

Liu, Y., Hu, T., Zhang, H., Wu, H., Wang, S., Ma, L., & Long, M. (2024a). iTransformer: Inverted transformers are effective for time series forecasting. In *The twelfth international conference on learning representations (ICLR), Austria*.

Liu, Y., Qin, G., Huang, X., Wang, J., & Long, M. (2025). Timer-XL: Long-context transformers for unified time series forecasting. In *The thirteenth international conference on learning representations, ICLR, Singapore, April 24–28*.

Liu, Y., Zhang, H., Li, C., Huang, X., Wang, J., & Long, M. (2024b). Timer: Generative pre-trained transformers are large time series models. In *Forty-first international conference on machine learning, ICML, Vienna, Austria, July 21–27*.

Ma, J., Cheng, J. C. P., Ding, Y., Lin, C., Jiang, F., Wang, M., & Zhai, C. (2020). Transfer learning for long-interval consecutive missing values imputation without external features in air pollution time series. *Advanced Engineering Informatics*, *44*, 101092–10203.

Ma, J., Shou, Z., Zareian, A., Mansour, H., & Vetro, et al. (2019). CDSA: Cross-dimensional self-attention for multivariate, geo-tagged time series imputation. arXiv preprint arXiv:1905.09904, .

Mao, P., & Li, K. (2024). OpenTOS: Open-source system for transfer learning Bayesian optimization. In *Proceedings of the 33rd ACM international conference on information and knowledge management* (pp. 5254–5259).

Moreno-Pino, F., Olmos, P. M., & Artés-Rodríguez, A. (2023). Deep autoregressive models with spectral attention. *Pattern Recognition*, *133*, 109014–109026.

Nayak, S., Dwivedi, D., & Babu, K. V. S. M., et al. (2024). Data imputation using self attention based model for enhancing distribution grid monitoring and protection systems. *IEEE Transactions on Instrumentation and Measurement*, *73*, 1–11.

Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2020). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *8th International conference on learning representations (ICLR), Ethiopia*.

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359.

Pratama, I., Permanasari, A. E., & Ardiyanto, I., et al. (2016). A review of missing values handling methods on time-series data. In *2016 International conference on information technology systems and innovation (ICITSI)* (pp. 1–6). IEEE.

Ruan, W., Xu, P., & Sheng, Q. Z., et al. (2016). When sensor meets tensor: Filling missing sensor values through a tensor approach. In *Proceedings of the 25th ACM international on conference on information and knowledge management* (pp. 2025–2028).

Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, *36*(3), 1181–1191.

Shukla, S. N., & Marlin, B. M. (2021a). Heteroscedastic temporal variational autoencoder for irregularly sampled time series. arXiv preprint arXiv:2107.11350, .

Shukla, S. N., & Marlin, B. M. (2021b). Multi-time attention networks for irregularly sampled time series. In *9th International conference on learning representations (ICLR), Austria*.

Tashiro, Y., Song, J., Song, Y., & Ermon, S. (2021). CDSI: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in neural information processing systems*, *vol. 34*, 24804–24816.

Wang, Z., Xu, X., Zhang, W., Trajcevski, G., & et, a. (2022). Learning latent seasonal-trend representations for time series forecasting. In *Advances on neural information processing systems (neurIPS) USA*.

Woo, G., Liu, C., Sahoo, D., & Kumar, A., et al. (2022). CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. In *International conference on learning representations* (pp. 1–18).

Wu, H., Xu, J., Wang, J., & Long, M. (2022). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. In *Neural information processing systems* (pp. 1–20).

Yoon, J., Zame, W. R., & van der Schaar, M. (2018). Estimating missing data in temporal data streams using multi-directional recurrent neural networks. *IEEE Transactions on Biomedical Engineering*, *66*(5), 1477–1490.

You, K., Kou, Z., Long, M., & Wang, J. (2020). Co-tuning for transfer learning. *Advances in neural information processing systems*, *vol. 33*, 17236–17246.

Zhang, K., Jing, B., Candan, K. S., Zhou, D., Wen, Q., Liu, H., & Ding, K. (2025). Cross-domain conditional diffusion models for time series imputation. arXiv preprint arXiv:2506.12412, .

Zhang, S., Guo, B., Dong, A., He, J., Xu, Z., & Chen, S. (2017). Cautionary tales on air-quality improvement in Beijing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *473*, 1–14.

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., & Jin, R. (2022). Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning* (pp. 27268–27286). PMLR.

Zhou, T., Niu, P., Sun, L., Jin, R. et al. (2023). One fits all: Power general time series analysis by pretrained LM. *Advances in neural information processing systems (NIPS)*, *vol. 36*, 43322–43355.

Zhu, K., & Zhao, C. (2025). When causality meets missing data: Fusing key information to bridge causal discovery and imputation in time series via bidirectional meta-learning. *Information Fusion*, *117*, 102811–102823.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, *109*(1), 43–76.