Self-Supervised Learning of Time Series Representation via Diffusion Process and Imputation-Interpolation-Forecasting Mask

Zineb Senane*
Motherbrain, EQT Group
KTH Royal Institute of Technology
Stockholm, Sweden
senane@kth.se

Lele Cao*
Motherbrain, EQT Group
Stockholm, Sweden
caolele@gmail.com
lele.cao@eqtpartners.com

Valentin Leonhard Buchner
Motherbrain, EQT Group
Stockholm, Sweden
vlbuchner@gmail.com

Yusuke Tashiro
Mitsubishi UFJ Trust Investment Technology Institute
Tokyo, Japan
yusu.tashi@gmail.com

Lei You
Technical University of Denmark
Ballerup, Denmark
leiyo@dtu.dk

Pawel Andrzej Herman
KTH Royal Institute of Technology
Stockholm, Sweden
paherman@kth.se

Mats Nordahl
KTH Royal Institute of Technology
Stockholm, Sweden
mnordahl@kth.se

Ruibo Tu
KTH Royal Institute of Technology
Stockholm, Sweden
ruibo@kth.se

Vilhelm von Ehrenheim
Motherbrain, EQT Group
QA.tech
Stockholm, Sweden
vilhelm.vonehrenheim@eqtpartners.com

ABSTRACT

Time Series Representation Learning (TSRL) focuses on generating informative representations for various Time Series (TS) modeling tasks. Traditional Self-Supervised Learning (SSL) methods in TSRL fall into four main categories: reconstructive, adversarial, contrastive, and predictive, each with a common challenge of sensitivity to noise and intricate data nuances. Recently, diffusion-based methods have shown advanced generative capabilities. However, they primarily target specific application scenarios like imputation and forecasting, leaving a gap in leveraging diffusion models for generic TSRL. Our work, Time Series Diffusion Embedding (TSDE), bridges this gap as the first diffusion-based SSL TSRL approach. TSDE segments TS data into observed and masked parts using an Imputation-Interpolation-Forecasting (IIF) mask. It applies a trainable embedding function, featuring dual-orthogonal Transformer encoders with a crossover mechanism, to the observed part. We train a reverse diffusion process conditioned on the embeddings, designed to predict noise added to the masked part. Extensive experiments demonstrate TSDE's superiority in imputation, interpolation, forecasting, anomaly detection, classification, and clustering. We also conduct an ablation study, present embedding visualizations, and compare inference speed, further substantiating TSDE's efficiency and validity in learning representations of TS data.

*Zineb Senane and Lele Cao contributed equally as first authors. For correspondence, please reach out to either of them. The source code and models for reproduction purposes are available at https://github.com/llcresearch/TSDE.

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08.
https://doi.org/10.1145/3637528.3671673

CCS CONCEPTS

• **Computing methodologies → Unsupervised learning**; **Learning latent representations**; • **Mathematics of computing → Time series analysis**; **Probabilistic algorithms**.

KEYWORDS

multivariate time series, diffusion model, representation learning, self-supervised learning, imputation, interpolation, forecasting, anomaly detection, clustering, classification, time series modeling

ACM Reference Format:
Zineb Senane*, Lele Cao*, Valentin Leonhard Buchner, Yusuke Tashiro, Lei You, Pawel Andrzej Herman, Mats Nordahl, Ruibo Tu, and Vilhelm von Ehrenheim. 2024. Self-Supervised Learning of Time Series Representation via Diffusion Process and Imputation-Interpolation-Forecasting Mask. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3671673

1 INTRODUCTION

Time Series (TS) data is a sequence of data points collected at regular time intervals. It is prevalent in various real-world applications, such as understanding human behavioral patterns [9], conducting in-depth financial market analyses [5], predicting meteorological phenomena [34], and enhancing healthcare diagnostics [46]. In this work, we focus on Multivariate TS (MTS) data, which refers to a TS with multiple variables or features recorded at each time point, where these variables may have inter-dependencies. This is in contrast to Univariate TS (UTS), which only involves a single variable. It should be noted that Multiple TS (Multi-TS) differs from MTS as it pertains to the simultaneous monitoring of several UTSs, each operating independently without any interrelation among them. While this paper primarily concentrates on MTS data, our methodology and insights are also applicable to UTS and Multi-TS, ensuring the versatility and broad applicability of our approach.

To effectively extract and interpret valuable information from intricate raw MTS data, the field of Time Series Representation Learning (TSRL) has become increasingly pivotal. TSRL focuses on learning latent representations that encapsulate critical information within the time series, thereby uncovering the intrinsic dynamics of the associated systems or phenomena [52]. Furthermore, the learned representations are crucial for a variety of downstream applications, such as time series imputation, interpolation, forecasting, classification, clustering and anomaly detection. TSRL can be conducted in a supervised manner; however, the need for extensive and accurate labeling of vast time series data presents a significant bottleneck, often resulting in inefficiencies and potential inaccuracies. Consequently, our focus lies in unsupervised learning techniques, which excel in extracting high-quality MTS representations without the constraints of manual labeling.

Self-Supervised Learning (SSL), a subset of unsupervised learning, has emerged as a highly effective methodology for TSRL. SSL utilizes innovative pretext tasks[1] to generate supervision signals from unlabeled TS data, thereby facilitating the model's ability to autonomously learn valuable representations without relying on external labels. The four main designs of SSL pretext tasks – reconstructive, adversarial, contrastive, and predictive [18, 42, 52, 100] – will be elaborated in Section 2. These designs have demonstrated notable success in addressing TSRL across a diverse range of applications, yet they often struggle with capturing the full complexity of MTS data, particularly in modeling intricate long-term dependencies and handling high-dimensional, noisy datasets.

Due to their advanced generative capabilities, diffusion models [28, 71, 73–76] have emerged as a promising solution for TS modeling, adept at handling the complexities and long-term dependencies often found in MTS data. While these methods have shown success in specific tasks like forecasting [62] and imputation [80], their adoption in SSL TSRL remains largely unexplored, leaving a gap in the related research literature. Our work, Time Series Diffusion Embedding (TSDE), pioneers in this area by integrating conditional diffusion processes with crossover Transformer encoders and introducing an Imputation-Interpolation-Forecasting (IIF) mask strategy. This unique combination allows TSDE to generate versatile representations that are applicable to a wide range of tasks, including imputation, interpolation, forecasting, classification, anomaly detection, and clustering. Our main contributions are:

- We propose a novel SSL TSRL framework named TSDE, which optimizes a denoising (reverse diffusion) process, conditioned on a learnable MTS embedding function.
- We develop dual-orthogonal Transformer encoders integrated with a crossover mechanism, which learns MTS embeddings by capturing temporal dynamics and feature-specific dependencies.
- We design a novel SSL pretext task, the IIF masking strategy, which creates pseudo observation masks designed to simulate the typical imputation, interpolation, and forecasting tasks.
- We experimentally show that TSDE achieves superior performance over existing methods across a wide range of MTS tasks, thereby validating the universality of the learned embeddings.

[1]A pretext task in SSL is a self-generated learning challenge designed to facilitate the extraction of informative representations for downstream tasks, encompassing various methods such as transformation prediction, masked prediction, instance discrimination, and clustering, tailored to the specific data modality involved [21, 33, 100].

2 RELATED WORK

This research addresses the problem of TSRL using a SSL approach. Inspired by the taxonomies adopted by [18, 42, 52, 100], we structure our review of SSL-based TSRL around four primary methodologies: reconstructive, adversarial, contrastive, and predictive methods.

Reconstructive methods focus on minimizing the discrepancy between original and reconstructed MTS data, mostly using an encoder-decoder Neural Network (NN) architecture to emphasize salient features and filter out noise, thereby training the NN to learn meaningful representations [27]. Recent mainstream methods in this category predominantly employ Convolutional NN (CNN) [72, 99], Recurrent NN (RNN) [47, 65] or Transformer [15, 102] as their architectural backbone. In this category, deep clustering stands out by simultaneously optimizing clustering and reconstruction objectives. It has been implemented through various clustering algorithms, including k-means [7, 88], Gaussian Mixture Model (GMM) [4, 32], and spectral clustering [79]. Reconstructive methods might face limitations in addressing long-term dependencies and adequately representing complex features such as seasonality, trends, and noise in extensive, high-dimensional datasets.

Adversarial methods utilize Generative Adversarial Network (GAN) to learn TS representations by differentiating between real and generated data [50, 58]. These methods often integrate advanced NN architectures or autoregressive models to effectively capture temporal dependencies and generate realistic TS data. For instance, TimeGAN [93] combines GANs with autoregressive models for temporal dynamics replication, while RGAN [22] uses RNN to enhance the realism of generated TS. Furthermore, approaches like TimeVAE [16] and DIVERSIFY [44] innovate in data generation, with the former tailoring outputs to user-specified distributions and latter employing adversarial strategies to maximize distributional diversity in generated TS data. However, the intricate training process of GANs, potential for mode collapse, and reliance on high-quality datasets are notable drawbacks of adversarial methods, potentially generating inconsistent or abnormal samples [100].

Contrastive methods distinguish themselves by optimizing self-discrimination tasks, contrasting positive samples with similar characteristics against negative samples with different ones [106]. These methods learn representations by generating augmented views of TS data and leveraging the inherent similarities and variations within the data [100]. They include instance-level models [11, 12, 78, 91] that treat each sample independently, using data augmentations to form positive and negative pairs. Prototype-level models [8, 37, 51, 98], on the other hand, break this independence by clustering semantically similar samples, thereby capturing higher-level semantic information. Additionally, temporal-level models [19, 78, 90, 95] address TS-specific challenges by focusing on scale-invariant representations at individual timestamps, enhancing the understanding of complex temporal dynamics. However, a common disadvantage across these contrastive methods is their potential to overlook higher-level semantic information, especially when not integrating explicit semantic labels, leading to the generation of potentially misleading negative samples.

Predictive methods excel in capturing shared information from TS data by maximizing mutual information from various data slices or augmented views. These methods, like TST [97], wave2vec [67], CaSS [14] and SAITS [17], focus on predicting future, missing, or contextual information, thereby bypassing the need for full input reconstruction. Most recent advancements in this category, such as TEMPO [3] and TimeGPT [25], leverage LLM (Large Language Model) architectures to effectively decompose and predict complex TS components. TimeGPT, in particular, stands out as a foundation model specifically for TS forecasting, yet it only treats MTS as Multi-TS. Lag-Llama [61], another notable predictive model, demonstrates strong univariate probabilistic forecasting, trained on a vast corpus of TS data. However, the challenge in these methods is their focus on local information, which can limit their capacity to capture long-term dependencies and make them susceptible to noise and outliers, thus affecting their generalization ability.

Diffusion-based methods in TS modeling have recently gained traction, leveraging the unique abilities of diffusion models to model the data distribution through a process of injecting and reversing noise [100]. These models, like TimeGrad [62] and CSDI [80], have been effectively applied to tasks such as forecasting and imputation, employing innovative techniques like RNN-conditioned diffusion and multiple Transformer encoders. Recent developments like SSSD [2] have further evolved the field by integrating structured state space models [26] with diffusion processes. These advancements have showcased the flexibility and potential of diffusion models in handling diverse TS data, with applications ranging from electrical load forecasting with DiffLoad [83] to predicting spatio-temporal graph evolutions using DiffSTG [84]. Despite these significant advancements, a notable gap remains in the application of diffusion models for TSRL. While a recent study [13] demonstrates the efficacy of diffusion models as robust visual representation extractors, their specific adaptation and optimization for TSRL have not been explored. Our work aims to fill this gap with the innovative TSDE framework, synergistically integrating conditional diffusion processes and crossover Transformer encoders, coupled with an innovative IIF mask strategy, to effectively tackle a wide range of downstream tasks.

3 THE APPROACH

The task entails learning general-purpose embeddings for MTS that has K features/variables and L time steps. Formally, given a multivariate time series $\mathbf{x}$:

$$\mathbf{x} = \{x_{1:K,1:L}\} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,L} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ x_{K,1} & x_{K,2} & \cdots & x_{K,L} \end{bmatrix} \in \mathbb{R}^{K\times L}, \quad (1)$$

we aim to learn a $\boldsymbol{\phi}$-parameterized embedding function $\boldsymbol{f_\phi}(\cdot)$ that maps the input MTS $\mathbf{x}$ to a latent representation $\mathbf{Z}$:

$$\mathbf{Z} = \{\mathbf{z}_{1:K,1:L}\} = \boldsymbol{f_\phi}(\mathbf{x}) = \begin{bmatrix} \mathbf{z}_{1,1} & \cdots & \mathbf{z}_{1,L} \\ \vdots & \ddots & \vdots \\ \mathbf{z}_{K,1} & \cdots & \mathbf{z}_{K,L} \end{bmatrix} \in \mathbb{R}^{K\times L\times C}, \quad (2)$$

where each element $\mathbf{z}_{k,l} \in \mathbb{R}^C$ represents the embedding vector for the k-th feature and l-th step, with C denoting the dimensionality of the embedding space. We propose to learn $\boldsymbol{f_\phi}$ by leveraging a conditional diffusion process trained in a self-supervised fashion.

3.1 Unconditional Diffusion Process

The unconditional diffusion process assumes a sequence of latent variables $\mathbf{x}_t$ $(t \in \mathbb{Z} \cap [1, T])$ in the same space as $\mathbf{x}$. For unification, we will denote $\mathbf{x}$ as $\mathbf{x}_0$ henceforth. The objective is to approximate the ground-truth MTS distribution $q(\mathbf{x}_0)$ by learning a $\boldsymbol{\theta}$-parameterized model distribution $p_{\boldsymbol{\theta}}(\mathbf{x}_0)$. The entire process comprises both forward and reverse processes.

3.1.1 Forward process. In this process, Gaussian noise is gradually injected to $\mathbf{x}_0$ in T steps until x_T is close enough to a standard Gaussian distribution, which can be expressed as a Markov chain:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \textstyle\prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad (3)$$

where $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ is a diffusion transition kernel, and is defined as

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \quad (4)$$

which is a conditional Gaussian distribution with a mean of $\sqrt{1-\beta_t}\mathbf{x}_{t-1}$ and a covariance matrix of $\beta_t\mathbf{I}$, and $\beta_t \in (0, 1)$ indicates the noise level at each diffusion step t. Because of the properties of Gaussian kernels, we can sample any $\mathbf{x}_t$ from $\mathbf{x}_0$ directly with

$$q(\mathbf{x}_t|\mathbf{x}_0) := \mathcal{N}(\mathbf{x}_t; \sqrt{\tilde{\alpha}_t}\mathbf{x}_0, (1-\tilde{\alpha}_t)\mathbf{I}), \text{where } \tilde{\alpha}_t := \textstyle\prod_{i=1}^{t}(1-\beta_i), \quad (5)$$

and $\mathbf{x}_t = \sqrt{\tilde{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\tilde{\alpha}_t}\boldsymbol{\epsilon}$, and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

3.1.2 Reverse process. This process, modeled by a NN parameterized with $\boldsymbol{\theta}$, recovers $\mathbf{x}_0$ by progressively denoising $\mathbf{x}_T$:

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\textstyle\prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t), \quad (6)$$

where $p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is the reverse transition kernel with a form of

$$p_{\boldsymbol{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t), \Sigma_{\boldsymbol{\theta}}(\mathbf{x}_t, t)). \quad (7)$$

To approximate the reverse transition kernel, Ho et al. [29] propose the following reparametrization of the mean and variance:

$$\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) := (1-\beta_t)^{-\frac{1}{2}}(\mathbf{x}_t - \beta_t(1-\tilde{\alpha}_t)^{-\frac{1}{2}}\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)), \quad (8)$$

$$\Sigma_{\boldsymbol{\theta}}(\mathbf{x}_t, t) := \boldsymbol{\sigma}_{\boldsymbol{\theta}}(\mathbf{x}_t, t)\mathbf{I} = \sigma_t^2\mathbf{I}, \quad (9)$$

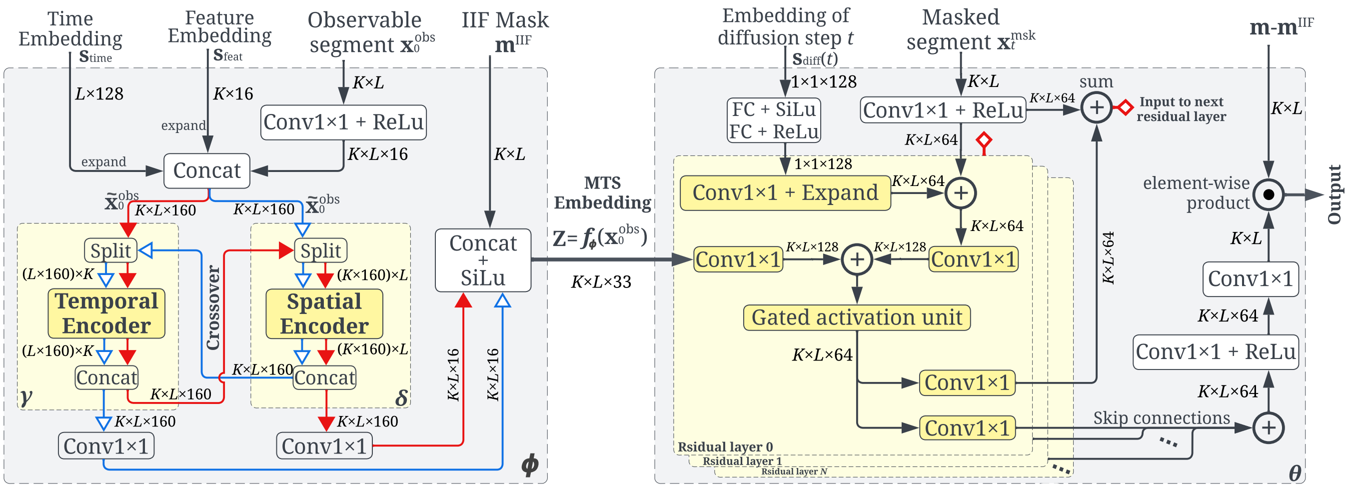
where $\sigma_t^2 = \beta_t(1-\tilde{\alpha}_{t-1})/(1-\tilde{\alpha}_t)$ when $t > 1$, otherwise $\sigma_t^2 = \beta_1$; $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$ is a trainable network predicting the noise added to input $\mathbf{x}_t$ at diffusion step t. Specifically, $\tilde{\alpha}_T \approx 0$ such that $q(\mathbf{x}_T) \approx \mathcal{N}(\mathbf{x}_T; 0, \mathbf{I})$, thus the starting point of the backward chain is a Gaussian noise.

3.2 Imputation-Interpolation-Forecasting Mask

The reverse process of unconditional diffusion facilitates the generation of MTS from noise. However, our objective is to create general-purpose embeddings for unlabeled MTS, which can be leveraged in many popular downstream tasks such as imputation, interpolation, and forecasting. Consequently, we propose an Imputation-Interpolation-Forecasting (IIF) mask strategy, producing a pseudo observation mask $\mathbf{m}^{\text{IIF}} = \{m^{\text{IIF}}_{1:K,1:L}\} \in \{0,1\}^{K\times L}$ where $m^{\text{IIF}}_{k,l} = 1$ if $x_{k,l}$ in Equation (1) is observable, and $m^{\text{IIF}}_{k,l} = 0$ otherwise. Algorithm 1 details the implementation and combination of imputation, interpolation, and forecasting masks[2]. During training, given any original MTS $\mathbf{x}_0$, we extract the observed ($\mathbf{x}_0^{\text{obs}}$) and masked ($\mathbf{x}_0^{\text{msk}}$) segments by

$$\mathbf{x}_0^{\text{obs}} := \mathbf{x}_0 \odot \mathbf{m}^{\text{IIF}} \quad \text{and} \quad \mathbf{x}_0^{\text{msk}} := \mathbf{x}_0 \odot (\mathbf{m} - \mathbf{m}^{\text{IIF}}), \quad (10)$$

[2]The *imputation mask* simulates random missing values; the *interpolation mask* mimics the MTS interpolation tasks by masking all values at a randomly selected timestamp; and the *forecasting mask* assumes all values post a specified timestamp unknown.



Embedding Function/Block: $f_\phi(\mathbf{x}_0^{obs})$ Conditional Reverse Diffusion (denoising): $\epsilon_\theta(\mathbf{x}_t^{msk}, t\,|\,f_\phi(\mathbf{x}_0^{obs}))\odot(\mathbf{m}\text{-}\mathbf{m}^{IIF})$

Figure 1: The TSDE architecture comprises an embedding function (left) and a conditional reverse diffusion block (right): the temporal and spatial encoders are implemented as one-layer Transformer.

where $\odot$ represents element-wise product; and $\mathbf{m} = \{m_{1:K,1:L}\} \in \{0,1\}^{K\times L}$ is a mask with zeros indicating originally missing values in $\mathbf{x}_0$. We now reformulate our self-supervised learning objective to generate the masked version of MTS, denoted as $\mathbf{x}_0^{msk}$, from a **corrupted input** $\mathbf{x}_t^{msk}$, through a diffusion process, **conditioned on the embedding of the observed MTS** $\mathbf{x}_0^{obs}$, i.e., $f_\phi(\mathbf{x}_0^{obs})$. Both the diffusion process (parameterized by θ) and the embedding function (parameterized by ϕ) are approximated with a trainable NN.

3.3 Conditional Reverse Diffusion Process

Our conditional diffusion process estimates the ground-truth conditional probability $q(\mathbf{x}_0^{msk}|f_\phi(\mathbf{x}_0^{obs}))$ by re-formulating (6) as

$$p_\theta(\mathbf{x}_{0:T}^{msk}|f_\phi(\mathbf{x}_0^{obs})) := p(\mathbf{x}_T^{msk})\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}^{msk}|\mathbf{x}_t^{msk}, f_\phi(\mathbf{x}_0^{obs})). \quad (11)$$

Similar to (7), the reverse kernel $p_\theta(\mathbf{x}_{t-1}^{msk}|\mathbf{x}_t^{msk}, f_\phi(\mathbf{x}_0^{obs})) :=$

$$\mathcal{N}(\mathbf{x}_{t-1}^{msk}; \boldsymbol{\mu}_\theta(\mathbf{x}_t^{msk}, t, f_\phi(\mathbf{x}_0^{obs})), \Sigma_\theta(\mathbf{x}_t^{msk}, t, f_\phi(\mathbf{x}_0^{obs}))). \quad (12)$$

According to DDPM [29], the variance $\Sigma_\theta(\mathbf{x}_t^{msk}, t, f_\phi(\mathbf{x}_0^{obs}))$ can be formulated in the same way as (9), i.e., $\boldsymbol{\sigma}_\theta(\mathbf{x}_t^{msk}, t, f_\phi(\mathbf{x}_0^{obs}))\mathbf{I} = \sigma_t^2\mathbf{I}$. Similar to Equation (8), the conditional mean $\boldsymbol{\mu}_\theta(\mathbf{x}_t^{msk}, t, f_\phi(\mathbf{x}_0^{obs}))$:=

$$(1-\beta_t)^{-\frac{1}{2}}(\mathbf{x}_t^{msk} - \beta_t(1-\tilde{\alpha}_t)^{-\frac{1}{2}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t^{msk}, t|f_\phi(\mathbf{x}_0^{obs}))). \quad (13)$$

3.4 Training Loss and Procedure

It has been shown in [29] that the reverse process of unconditional diffusion can be trained by minimizing the following loss:

$$\mathcal{L}(\theta) := \mathbb{E}_{\mathbf{x}_0\sim q(\mathbf{x}_0), \epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}), t}\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|_2^2 . \quad (14)$$

Inspired by [80], we replace the noise prediction NN $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)$ with the conditioned version $\boldsymbol{\epsilon}_\theta(\mathbf{x}_t^{msk}, t|f_\phi(\mathbf{x}_0^{obs}))$ in (14), obtaining

$$\mathcal{L}(\theta,\phi) := \mathbb{E}_{\mathbf{x}_0\sim q(\mathbf{x}_0), \epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}), t}\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t^{msk}, t|f_\phi(\mathbf{x}_0^{obs}))\|_2^2 . \quad (15)$$

Given the focus of training is solely on predicting the noise at the non-missing and masked locations, we actually minimize $\widetilde{\mathcal{L}}(\theta,\phi) :=$

$$\mathbb{E}_{\mathbf{x}_0\sim q(\mathbf{x}_0), \epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}), t}\|(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t^{msk}, t|f_\phi(\mathbf{x}_0^{obs})))\odot(\mathbf{m}-\mathbf{m}^{IIF})\|_2^2. \quad (16)$$

The self-supervised and mini-batch training procedure, detailed in Algorithm 2, essentially attempts to solve $\min_{\theta,\phi}\widetilde{\mathcal{L}}(\theta,\phi)$. In each iteration i of the training process, a random diffusion step t is chosen, at which point the denoising operation is applied.

3.5 Embedding Function

The left part of Figure 1 illustrates the architectural design of the embedding function $f_\phi(\mathbf{x}_0^{obs})$. This figure highlights that the function not only processes the input $\mathbf{x}_0^{obs}$, but also incorporates additional side information (namely, time embedding $\mathbf{s}_{time}(l)$, feature embedding $\mathbf{s}_{feat}(k)$, and the mask $\mathbf{m}^{IIF}$) into its computations. Consequently, the notation $f_\phi(\mathbf{x}_0^{obs})$ is succinctly used to represent the more extensive formulation $f_\phi(\mathbf{x}_0^{obs}, \mathbf{s}_{time}, \mathbf{s}_{feat}, \mathbf{m}^{IIF})$, which accounts for all the inputs processed by the function. To obtain 128-dimensional $\mathbf{s}_{time}(l)$, we largely follow [80, 107]:

$$\mathbf{s}_{time}(l) = \left(\sin\frac{l}{\tau^{\frac{0}{64}}}, \ldots, \sin\frac{l}{\tau^{\frac{63}{64}}}, \cos\frac{l}{\tau^{\frac{0}{64}}}, \ldots, \cos\frac{l}{\tau^{\frac{63}{64}}}\right), \quad (17)$$

where τ=10,000 and $l \in \mathbb{Z}\cap[1, L]$. For $\mathbf{s}_{feat}(k)$, a 16-dimensional feature embedding is obtained by utilizing the categorical feature embedding layer available in PyTorch. The observable segment $\mathbf{x}_0^{obs}$ undergoes a nonlinear transformation and is then concatenated with time and feature embeddings, resulting in $\tilde{\mathbf{x}}_0^{obs} \in \mathbb{R}^{K\times L\times 160}$:

$$\tilde{\mathbf{x}}_0^{obs} = Concat(ReLu(Conv(\mathbf{x}_0^{obs})), \mathbf{s}_{time}, \mathbf{s}_{feat}), \quad (18)$$

where $Concat(\cdot)$, $ReLu(\cdot)$ and $Conv(\cdot)$ represent concatenation, ReLu activation, and 1×1 convolution operation [39] respectively.

To accurately capture the inherent temporal dependencies and feature correlations in MTS data, thereby enabling clearer data interpretation and a customizable, modular design, we devise separate temporal and feature embedding functions: $g_\gamma(\tilde{\mathbf{x}}_0^{obs})$ and $h_\delta(\tilde{\mathbf{x}}_0^{obs})$,

Algorithm 1: Imputation-Interpolation-Forecasting Mask

Input: Mask $\mathbf{m} = \{m_{1:K,1:L}\} \in \{0,1\}^{K\times L}$ indicating the missing values in $\mathbf{x}_0$
Output: A pseudo observation mask $\mathbf{m}^{\text{IIF}} \in \{0,1\}^{K\times L}$

$r \leftarrow$ random value from the range of [0.1, 0.9]; // imputation mask ratio
$N \leftarrow \sum_{k=1}^{K}\sum_{l=1}^{L} m_{k,l}$; // total number of observed values
$\mathbf{m}^{\text{IIF}} \leftarrow \mathbf{m}$ and randomly set $\lfloor N \times r \rceil$ 1s to 0; // apply imputation mask
Sample a probability p uniformly from the range of [0, 1];
if $1/3 < p < 2/3$ **then**
  $l' \leftarrow$ uniformly sample a time step from $\mathbb{Z} \cap [1, L]$;
  $\mathbf{m}^{\text{IIF}}[:, l'] \leftarrow 0$; // mix with interpolation mask
else if $p >= 2/3$ **then**
  $l' \leftarrow$ uniformly sample a time window length from $\mathbb{Z} \cap [1, \lfloor \frac{L}{3} \rceil]$;
  $\mathbf{m}^{\text{IIF}}[:, -l':] \leftarrow 0$; // mix with forecasting mask
return $\mathbf{m}^{\text{IIF}}$;

parameterized by γ and δ respectively. Inspired by [80], both the temporal $\boldsymbol{g}_\gamma(\cdot)$ and feature $\boldsymbol{g}_\delta(\cdot)$ encoders are simply implemented as a one-layer Transformer encoder that takes an input tensor shaped $K\times L\times 160$, as shown in Figure 1. Specifically, the temporal encoder operates on tensors shaped $1\times L\times 160$, representing a feature across all timestamps; and the feature encoder handles tensors shaped $K\times 1\times 160$, representing a feature vector corresponding to a time stamp.

To integrate temporal and feature embeddings in varying orders without adding to the model's trainable parameters, we have developed a crossover mechanism. This mechanism is depicted by the red and blue arrows in Figure 1. It facilitates the generation of $\boldsymbol{g}_\gamma(\boldsymbol{h}_\delta(\tilde{\mathbf{x}}_0^{\text{obs}}))$ and $\boldsymbol{h}_\delta(\boldsymbol{g}_\gamma(\tilde{\mathbf{x}}_0^{\text{obs}}))$, which are subsequently transformed and concatenated along with $\mathbf{m}^{\text{IIF}}$, resulting in the final embedding $\mathbf{Z} = \boldsymbol{f}_{\boldsymbol{\phi}}(\mathbf{x}_0^{\text{obs}}) :=$

$$SiLu\Big(Concat\Big(Conv(\boldsymbol{g}_\gamma(\boldsymbol{h}_\delta(\tilde{\mathbf{x}}_0^{\text{obs}}))), Conv(\boldsymbol{h}_\delta(\boldsymbol{g}_\gamma(\tilde{\mathbf{x}}_0^{\text{obs}}))), \mathbf{m}^{\text{IIF}}\Big)\Big), \quad (19)$$

where $SiLu(\cdot)$ is the Sigmoid-weighted Linear Unit (SiLU) activation function [20]. Once the model is trained, the embedding for any MTS $\mathbf{x}_0$ is computed following Equations (18) and (19), where $\mathbf{x}_0^{\text{obs}}$ and $\mathbf{m}^{\text{IIF}}$ are substituted with $\mathbf{x}_0$ and $\mathbf{m}$, respectively.

3.6 The Overall Architecture

Figure 1 provides a comprehensive depiction of the various components within the TSDE architecture. The process begins by applying the IIF mask $\mathbf{m}^{\text{IIF}}$ to partition the input MTS into observable ($\mathbf{x}_0^{\text{obs}}$) and masked ($\mathbf{x}_0^{\text{msk}}$) segments. The entire architecture primarily consists of two key elements: (1) an embedding function $\boldsymbol{f}_{\boldsymbol{\phi}}(\mathbf{x}_0^{\text{obs}})$ thoroughly introduced in Section 3.5; and (2) a conditional reverse diffusion module, illustrated on the right side of Figure 1.

The conditional reverse diffusion, introduced in Section 3.3, functions as a noise predictor, effectively implementing $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\text{msk}}, t|\boldsymbol{f}_{\boldsymbol{\phi}}(\mathbf{x}_0^{\text{obs}}))$. During the i-th training step, as outlined in Algorithm 2, the sampled diffusion step t is first transformed into a 128-dimensional vector, denoted as $\mathbf{s}_{\text{diff}}(t) :=$

$$\Big(\sin(10^{\frac{0\cdot 4}{63}}t), \ldots, \sin(10^{\frac{63\cdot 4}{63}}t), \cos(10^{\frac{0\cdot 4}{63}}t), \ldots, \cos(10^{\frac{63\cdot 4}{63}}t)\Big). \quad (20)$$

Subsequently, the MTS embedding $\mathbf{Z}$, along with $\mathbf{s}_{\text{diff}}(t)$ and $\mathbf{x}_0^{\text{msk}}$, are input into a residual block composed of N residual layers. The outputs of these layers are aggregated (summation), processed through some transformations, and combined with $\mathbf{x}_t^{\text{msk}}$. This results in $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\text{msk}}, t|\boldsymbol{f}_{\boldsymbol{\phi}}(\mathbf{x}_0^{\text{obs}})) \odot (\mathbf{m} - \mathbf{m}^{\text{IIF}})$, which is then utilized to compute the loss $\widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi})$, as formulated in Equation (16).

Algorithm 2: TSDE Training Procedure

Input: Ground-truth MTS data distribution $q(\mathbf{x}_0)$, noise scheduler $\{\tilde{\alpha}_t\}$, the denoising and embedding functions (approx. by NN): $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\cdot)$ and $\boldsymbol{f}_{\boldsymbol{\phi}}(\cdot)$
Output: The trained NN parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$
Parameter: The total number of training iterations N_{train} and learning rate τ

for $(i = 1; i \leq N_{train}; i{+}{+})$ **do**
  Sample a diffusion step $t \sim \text{Uniform}(\{1, \ldots, T\})$ and a MTS $\mathbf{x}_0 \sim q(\mathbf{x}_0)$;
  Obtain IIF Masking $\mathbf{m}^{\text{IIF}}$ by following Algorithm 1;
  Obtain the observed ($\mathbf{x}_0^{\text{obs}}$) and masked ($\mathbf{x}_0^{\text{msk}}$) parts using Equation (10);
  Sample a noise matrix $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ that has the same shape as $\mathbf{x}_0^{\text{msk}}$;
  Compute $\mathbf{x}_t^{\text{msk}} \leftarrow \sqrt{\tilde{\alpha}_t}\mathbf{x}_0^{\text{msk}} + \sqrt{1-\tilde{\alpha}_t}\boldsymbol{\epsilon}$;
  Compute loss $\widetilde{\mathcal{L}} := \|(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t^{\text{msk}}, t|\boldsymbol{f}_{\boldsymbol{\phi}}(\mathbf{x}_0^{\text{obs}}))) \odot (\mathbf{m} - \mathbf{m}^{\text{IIF}})\|_2^2$, cf. (16);
  $\boldsymbol{\theta} := \boldsymbol{\theta} - \tau\frac{\partial\widetilde{\mathcal{L}}}{\partial\boldsymbol{\theta}}$ and $\boldsymbol{\phi} := \boldsymbol{\phi} - \tau\frac{\partial\widetilde{\mathcal{L}}}{\partial\boldsymbol{\phi}}$;
return $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$;

3.7 Downstream Tasks and Model Efficiency

The trained model can be utilized in two scenarios: (1) the embedding function, as a standalone component, can be used to generate comprehensive MTS representations, which are suitable for various downstream applications including anomaly detection, clustering, and classification as demonstrated in Section 4.2, 4.3, and 4.4, respectively. (2) When combined with the trained conditional reverse diffusion process, the model is capable of predicting missing values (for imputation and interpolation) as well as future values (for forecasting) in MTS data. In the second scenario, a notable increase in speed can be achieved compared to the existing diffusion-based methods such as those in [62, 80]. This efficiency, confirmed in Section 4.1.5, stems from simplifying the conditional reverse diffusion (the right block of Figure 1, i.e., $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}$) to use only Conv1×1 operators. This streamlining significantly accelerates the T=50 steps reverse diffusion process.

4 EXPERIMENTS

Our evaluation of the TSDE framework includes thorough experiments across six tasks (imputation, interpolation, forecasting, anomaly detection, classification, and clustering) accompanied by additional analyses on inference efficiency, ablation study, and embedding visualization. For experiment details, dataset specifications, hyperparameters, and metric formulas, refer to Appendix of [68].

4.1 Imputation, Interpolation and Forecasting

4.1.1 Imputation. We carry out imputation experiments on PhysioNet[3] [70] and PM2.5[4] [92]. TSDE is benchmarked against several state-of-the-art TS imputation models. These include BRITS [6], a deterministic method using bi-directional RNN for correlation capture; V-RIN [53], employing variational-recurrent networks with feature and temporal correlations for uncertainty-based imputation; GP-VAE [24], integrating Gaussian Processes with VAEs; and CSDI [80], the top-performing model among the diffusion-based

[3] PhysioNet, a healthcare dataset with 4,000 records of 35 variables over 48 hours, is processed and hourly sampled as [66, 80], leading to ~80% missing rate. For testing, we randomly mask 10%, 50%, and 90% of observed values to create ground-truth scenarios. On this dataset, we pretrain TSDE for 2,000 epochs, followed by a 200-epoch finetuning with an imputation mask.

[4] PM2.5, an air quality dataset, features hourly readings from 36 Beijing stations over 12 months with artificially generated missing patterns. Adapting [80], each series spans 36 consecutive timestamps. On this dataset, we pretrain for 1,500 epochs and finetune for 100 epochs using a history mask as detailed in Algorithm 5 in [68].

Table 1: Probabilistic MTS imputation and interpolation benchmarking results, featuring TSDE's pretraining-only and task-specific finetuned (TSDE+ft) models against established baselines. We present mean and standard deviation (SD) from three iterations, with baseline results primarily derived or reproduced according to [80].

	Models	PhysioNet									PM2.5		
		10% masking ratio			50% masking ratio			90% masking ratio					
		CRPS	MAE	RMSE	CRPS	MAE	RMSE	CRPS	MAE	RMSE	CRPS	MAE	RMSE
Imputation	BRITS [6]	-	0.284(0.001)	0.619(0.022)	-	0.368(0.002)	0.693(0.023)	-	0.517(0.002)	0.836(0.015)	-	14.11(0.26)	24.47(0.73)
	V-RIN [53]	0.808(0.008)	0.271(0.001)	0.628(0.025)	0.831(0.005)	0.365(0.002)	0.693(0.022)	0.922(0.003)	0.606(0.006)	0.928(0.013)	0.526(0.025)	25.4(0.062)	40.11(1.14)
	GP-VAE [24]	0.558(0.001)*	0.449(0.002)*	0.739(0.001)*	0.642(0.003)*	0.566(0.004)*	0.898(0.005)*	0.748(0.002)*	0.690(0.002)*	1.008(0.002)*	0.397(0.009)	-	-
	unc. CSDI [80]	0.360(0.007)	0.326(0.008)	0.621(0.020)	0.458(0.008)	0.417(0.010)	0.734(0.024)	0.671(0.007)	0.625(0.010)	0.940(0.018)	0.135(0.001)	12.13(0.07)	22.58(0.23)
	CSDI [80]	0.238(0.001)	0.217(0.001)	0.498(0.020)	0.330(0.002)	0.301(0.002)	**0.614(0.017)**	0.522(0.002)	0.481(0.003)	0.803(0.012)	0.108(0.001)	**9.60(0.04)**	19.30(0.13)
	TSDE	**0.226(0.002)**	**0.208(0.001)**	**0.446(0.003)**	**0.316(0.000)**	**0.290(0.000)**	0.641(0.007)	**0.488(0.001)**	**0.450(0.001)**	**0.801(0.001)**	0.13(0.001)	11.41(0.60)	27.02(2.91)
	TSDE+ft	0.230(0.001)	0.211(0.001)	0.4718(0.013)	0.318(0.001)	0.292(0.001)	0.644(0.001)	0.490(0.001)	0.452(0.001)	0.803(0.001)	**0.107 (0.000)**	9.71(0.04)	**18.76(0.02)**
Interpolation	Latent ODE [64]	0.700(0.002)	0.522(0.002)	0.799(0.012)	0.676(0.003)	0.506(0.003)	0.783(0.012)	0.761(0.010)	0.578(0.009)	0.865(0.017)			
	mTANs [69]	0.526(0.004)	0.389(0.003)	0.749(0.037)	0.567(0.003)	0.422(0.003)	0.721(0.014)	0.689(0.015)	0.533(0.005)	0.836(0.018)			
	CSDI [80]	0.380(0.002)	0.362(0.001)	0.722(0.043)	0.418(0.001)	0.394(0.002)	0.700(0.013)	0.556(0.003)	0.518(0.003)	0.839(0.009)			
	TSDE	**0.365(0.001)**	**0.331(0.001)**	**0.597(0.002)**	**0.403(0.001)**	**0.371(0.001)**	**0.657(0.001)**	**0.517(0.001)**	**0.476(0.001)**	**0.775(0.001)**			
	TSDE+ft	0.374(0.001)	0.338(0.001)	0.610(0.003)	0.421(0.001)	0.385(0.001)	0.677(0.003)	0.570(0.004)	0.522(0.006)	0.821(0.006)			

* Results reproduced using GP-VAE original implementation available at https://github.com/ratschlab/GP-VAE. We report the mean and standard deviation of three runs.

TS imputation models. The model performance is evaluated using continuous ranked probability score (CRPS) to assess the fit of predicted outcomes with original data distributions, and two deterministic metrics – mean absolute error (MAE) and the root mean square error (RMSE). Deterministic metrics are calculated using the median across all samples, and CRPS value is reported as the normalized average score for all missing values distributions (approximated with 100 samples).

The imputation results, as detailed in the upper part of Table 1, highlight TSDE's superior performance over almost all metrics, outperforming all baselines. Notably, the pretraining-only variant (i.e., "TSDE") excels on the PhysioNet dataset, underpinning its robustness and enhanced generalization capability, even without the need of any imputation-specific finetuning. For the PM2.5 dataset, finetuning TSDE (i.e., "TSDE+ft") yields improved outcomes, likely attributable to its capability to adapt to the dataset's structured missing value patterns. Overall, TSDE's improvement in CRPS by 4.2%-6.5% over CSDI, a leading diffusion-based TS imputation model, signifies a notable advancement in the field. For a qualitative illustration of imputation results, refer to Figure 2(a).

4.1.2 Interpolation. For interpolation analysis, we utilized the same PhysioNet dataset [70], adopting the processing methods from [64, 69, 80]. Ground truth scenarios were created by masking all values at randomly selected timestamps, sampled at rates of 10%, 50% and 90%. TSDE is pretrained for 2,000 epochs, and then further finetuned using an interpolation-only mask for another 200 epochs. In our benchmarking, TSDE is compared against three TS interpolation methods: (1) Latent ODE [64], an RNN-based model leveraging ODE (ordinary differential equation) for dynamic, continuous and irregular TS handling; (2) mTANs [69], utilizing time embeddings and attention mechanisms, noted for its strong performance in irregular TS interpolation; and (3) CSDI [80] which has also reported competitive result in interpolation tasks.

The results in the lower section of Table 1 demonstrate TSDE's exceptional performance in interpolation, outperforming CSDI by 3.6%-7.0% in CRPS, 5.8%-8.6% in MAE, and 6.1%-17.3% in RMSE. These findings highlight TSDE's adeptness in managing irregular timestamp gaps, a likely factor behind the observation that finetuning does not enhance the pretraining-only TSDE's performance. Comparatively, while CSDI also operates on a similar diffusion model backbone, TSDE's edge lies in its unique embedding learning ability via IIF masking, adeptly capturing intricate TS characteristics and dynamics for improved results. A qualitative illustration of interpolation results can be found in Figure 2(b).

4.1.3 Forecasting. We conducted two sets of benchmarking experiments. The first was a benchmarking for probabilistic multivariate time series forecasting. We employ five real-world datasets: (1) *Electricity*, tracking hourly consumption across 370 customers; (2) *Solar*, detailing photovoltaic production at 137 Alabama stations; (3) *Taxi*, recording half-hourly traffic from 1,214 New York locations; (4) *Traffic*, covering hourly occupancy rates of 963 San Francisco car lanes; and (5) *Wiki*, monitoring daily views of 2,000 Wikipedia pages. Adapting the practices from [55, 66, 80], each dataset is converted into a series of multivariate sequences, with L_1 historical timestamps followed by L_2 timestamps for forecasting. Training data apply a rolling window approach with a stride of 1, while validation and testing data employ a stride of L_2, ensuring distinct, non-overlapping series for evaluation. Specific L_1 and L_2 values are outlined in Table 9 in [68]. For evaluation metrics, we use CRPS and MSE, supplemented by CRPS-Sum, as introduced in [66]. CRPS-Sum is computed by summing across different features, capturing the joint impact of feature distributions. As of benchmarking baselines, we include several *state-of-the-art probabilistic MTS forecasting models*: GP-copula [66], TransMAF [63] and TLAE [55]. Additionally, in the realm of *diffusion-based methods*, we include CSDI [80] and TimeGrad [62].

For the second benchmarking, which is a deterministic benchmarking including recent baselines in the time series library [86], we conducted five experiments for each baseline following the same setting in [86] with history-prediction window lengths of {8-8, 16-16, 32-32, 96-96, 96-192} on the Electricity dataset. We report the averaged performance in terms of MAE and MSE. We compared TSDE with the following baselines: TimesNet [86], ETSformer [85], LightTS [101], DLinear [96], FEDformer [104], Nonstationary Transformer [43], Autoformer [87], Pyraformer [41], Informer [103], Reformer [36], and PatchTST [56].

The forecasting results, as detailed in Table 3, showcase TSDE's robust performance, especially when finetuned with a forecasting

Table 2: Forecasting task results on Electricity following [86] setting. We compare extensive competitive models under five different history-prediction lengths {8-8, 16-16, 32-32, 96-96, 96-192}. *Avg* is averaged from all five history-prediction lengths results. See Table 13 in [68] for full results.

Models	**TSDE (Ours)**		TimesNet [2023]		ETSformer [2023]		LightTS* [2022]		DLinear* [2023]		FEDformer [2022]		Stationary [2022a]		Autoformer [2021]		Pyraformer [2022b]		Informer [2021]		Reformer [2020]		PatchTST [2023]	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Avg	**0.169**	**0.253**	0.195	0.290	0.241	0.355	0.227	0.329	0.359	0.408	0.193	0.310	0.191	0.285	0.192	0.311	0.299	0.367	0.295	0.390	0.287	0.382	0.342	0.362

∗ means that there are some mismatches between our input-output setting and their papers. We adopt their official codes and only change the length of input and output sequences for a fair comparison.

Table 3: Probabilistic MTS forecasting results embodying both TSDE (pretraining-only) and finetuned (TSDE+ft) variants. Baseline results are either sourced or reproduced from [55, 63, 66]. For TSDE-related experiments, we report the mean and SD across three iterations.

	Models	Electricity	Solar	Taxi	Traffic	Wiki
CRPS	GP-copula	0.056(0.002)	0.371(0.022)	0.360(0.201)	0.133(0.001)	0.236(0.000)
	TransMAF	0.052(0.000)	0.368(0.001)	0.377(0.002)	0.134(0.001)	0.274(0.007)
	TLAE	0.058(0.003)	**0.335(0.044)**	0.369(0.011)	0.097(0.002)	0.298(0.002)
	CSDI	0.043(0.001)*	0.396(0.021)*†	0.277(0.006)*	**0.076(0.000)***	0.232(0.006)*
	TSDE	0.043(0.000)	0.400(0.025)†	**0.277(0.001)**	0.091(0.001)	**0.222(0.003)**
	TSDE+ft	**0.042(0.000)**	0.375(0.013)†	0.282(0.001)	0.081(0.001)	0.226(0.003)
CRPS-sum	GP-copula	0.024(0.002)	0.337(0.024)	0.208(0.183)	0.078(0.002)	0.086(0.004)
	TransMAF	0.021(0.000)	0.301(0.014)	0.179(0.002)	0.056(0.001)	0.063(0.003)
	TimeGrad	0.021(0.001)	0.287(0.020)	**0.114(0.020)**	0.044(0.006)	**0.049(0.002)**
	TLAE	0.040(0.003)	**0.124(0.057)**	0.130(0.010)	0.069(0.002)	0.241(0.001)
	CSDI	0.019(0.001)*	0.345(0.029)*†	0.138(0.008)*	**0.020(0.000)***	0.084(0.013)*
	TSDE	0.020(0.001)	0.453(0.026)†	0.136(0.003)	0.038(0.003)	0.064(0.002)
	TSDE+ft	**0.017(0.001)**	0.345(0.012)†	0.153(0.006)	0.025(0.001)	0.059(0.003)
MSE	GP-copula	2.4e5(5.5e4)	9.8e2(5.2e1)	3.1e1(1.4e0)	6.9e-4(2.2e-5)	4.0e7(1.6e9)
	TransMAF	2.0e5	9.3e2	4.5e1	5.0e-4	**3.1e7**
	TLAE	2.0e5(1.6e4)	**6.8e2(1.3e2)**	2.6e1(1.4e0)	4.0e-4(5.0e-6)	3.8e7(7.2e4)
	CSDI	1.23e5(9.7e3)*	1.12e3(1.2e2)*†	**1.82e1(7.8e-1)***	**3.64e-4(0.0e0)***	4.43e7(1.0e7)*
	TSDE	1.20e5(3.5e3)	1.07e3(9.8e1)†	1.89e1(3.7e-1)	4.34e-4(0.0e0)	3.59e7(7.2e4)
	TSDE+ft	**1.16e5(6.0e3)**	9.25e2(4.9e1)†	1.92e1(2.4e-1)	3.88e-4(0.0e0)	3.62e7(1.8e5)

* We replace the linear Transformers [82] in CSDI with the Pytorch TransformerEncoder [57].
† We take the training MTS dataset and split it into training, validation and testing sets.

mask. Its effectiveness is notable when compared to CSDI, which is the most closely related method, sharing a diffusion backbone. TSDE particularly excels in the Electricity, Taxi, and Wiki datasets, especially as evaluated by the CRPS metric. However, it is important to note a discrepancy in the Solar dataset performance between TSDE/CSDI and other baselines, likely due to a data split issue: the actual test set, per the source code, is identical to the training set, which contradicts the details reported in the corresponding paper. Table 2 demonstrates that TSDE outperforms the recent baselines in terms of average MSE and MAE, highlighting its robustness and superiority compared to recent methods. The detailed results for each window length are available in the appendix of [68], Table 13. For a qualitative illustration, refer to Figure 2(c).

4.1.4 Ablation Study. In an ablation study on TSDE across imputation, interpolation, and forecasting, evaluated on PhysioNet (10% missing ratio) and Electricity datasets, two configurations were tested: one without crossover, and another without IIF mask (replaced by an imputation mask detailed in Algorithm ??). Table 4 underscores the positive contribution of the crossover mechanism across all three tasks. The impact of IIF masking, while less pronounced for imputation and interpolation, becomes noticeable in the forecasting task. This can be attributed to the random PhysioNet missing values, which are distributed fundamentally differently from a typical forecasting scenario. Thus, IIF strategy is important for TSDE to gain a generalization ability across various settings. The contrast between "TSDE" and "TSDE+ft" in Tables 1 and 3 serves as an ablation study for finetuning; it reveals that pretrained TSDE can achieve competitive results without the necessity of finetuning.

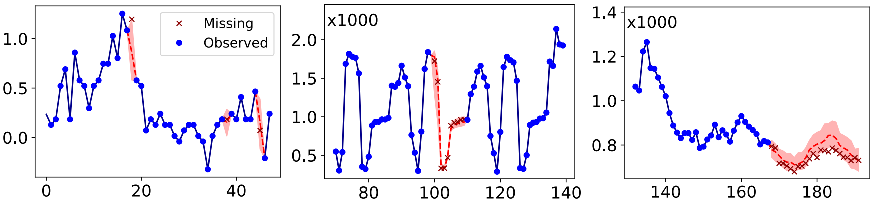
Table 4: Ablation study on PhysioNet (imputation and interpolation) and Electricity (forecasting) datasets.

Ablation Configuration	Imputation (MAE/CRPS)	Interpolation (MAE/CRPS)	Forecasting (CRPS-sum/CRPS)
w/o crossover	0.252(0.001)/0.274(0.001)	0.339(0.000)/0.373(0.000)	0.021(0.001)/0.046(0.001)
w/o IIF mask	0.207(0.001)/0.225(0.001)	0.330(0.001)/0.364(0.001)	0.028(0.004)/0.053(0.003)
TSDE	0.208(0.001)/0.226(0.002)	0.331(0.001)/0.365(0.001)	**0.020**(0.001)/**0.043**(0.000)

4.1.5 Inference Efficiency. Similar to CSDI [80], TSDE performs inference by gradual denoising from the last diffusion step T=50 to the initial step t=1, to approximate the true data distribution of missing or future values for imputation/interpolation/forecasting tasks. Typically, this iterative process can become computationally expensive. TSDE achieves a substantial acceleration in this process as illustrated in Table 5, where TSDE is ten times faster than CSDI under the same experimental setup. This is primarily owing to its globally shared, efficient dual-orthogonal Transformer encoders with a crossover mechanism, merely requiring approximately a quarter of the parameters used by CSDI for MTS encoding.

4.2 Anomaly Detection

For anomaly detection, we adopt an unsupervised approach using reconstruction error as the anomaly criterion, aligning with [86, 105]. We evaluate TSDE on five benchmark datasets: SMD [77], MSL [31], SMAP [31], SWaT [48] and PSM [1]. Once TSDE is pretrained, a projection layer, designed to reconstruct MTS from TSDE embeddings, is finetuned by minimizing MSE reconstruction loss. Our anomaly detection experiments align with TimesNet [105], utilizing preprocessed datasets from [89]. Following their method, we segment datasets into non-overlapping MTS instances of 100 timestamps each, labeling timestamps as anomalous based on a MSE threshold. This threshold is set according to the anomaly proportion in the validation dataset, ensuring consistency with baseline anomaly ratios for a fair comparison.



(a) Imputation on PhysioNet (b) Interpolation on Electricity (c) Forecasting on Electricity

Figure 2: Comparison of predicted and ground truth values for (a) imputation (10% missing), (b) interpolation, and (c) forecasting. The line is the median of the predictions and the red shade indicates 5%~95% quantile for missing/future values. See Appendix in [68] for more results.

Datasets	CSDI* (sec.)	TSDE (sec.)
Electricity	1,997	163
Solar	608	62
Taxi	27,533	1,730
Traffic	7,569	422
Wiki	9,138	391

* For fair comparison, the linear Transformer encoders in CSDI [80] is replaced with the TransformerEncoder [57] implementation in Pytorch.

Table 5: Inference time comparison for forecasting tasks between TSDE and CSDI.

Table 6: Anomaly detection: baseline results are cited from Table 27 of [105]; higher scores indicate better performance; the best and second best results are in bold and underlined, respectively.

Models	SMD P	SMD R	SMD F1	MSL P	MSL R	MSL F1	SMAP P	SMAP R	SMAP F1	SWaT P	SWaT R	SWaT F1	PSM P	PSM R	PSM F1	Avg. F1
Transformer	83.6	76.1	79.6	71.6	87.4	78.7	89.4	57.1	69.7	68.8	96.5	80.4	62.7	96.6	76.1	76.9
LogSparseT.	83.5	70.1	76.2	73.0	87.4	79.6	89.1	57.6	70.0	68.7	**97.3**	80.5	63.1	**98.0**	76.7	76.6
Reformer	82.6	69.2	75.3	85.5	83.3	84.4	90.9	57.4	70.4	72.5	96.5	82.8	59.9	95.4	73.6	77.3
Informer	86.6	77.2	81.6	81.8	86.5	84.1	90.1	57.1	69.9	70.3	96.7	81.4	64.3	96.3	77.1	78.8
AnomalyT.†	**88.9**	82.2	85.5	79.6	87.4	83.3	91.8	58.1	71.2	72.5	**97.3**	83.1	68.3	94.7	79.4	80.5
Pyraformer	85.6	80.6	83.0	83.8	85.9	84.9	92.5	57.7	71.1	87.9	96.0	91.8	71.7	96.0	82.1	82.6
Autoformer	88.1	82.3	85.1	77.3	80.9	79.0	90.4	58.6	71.1	89.8	95.8	92.7	99.1	88.1	93.3	84.3
NonStation.	88.3	81.2	84.6	68.5	**89.1**	77.5	89.4	59.0	71.1	68.0	96.7	79.9	97.8	96.8	97.3	82.1
DLinear	83.6	71.5	77.1	84.3	85.4	84.9	92.3	55.4	69.3	80.9	95.3	87.5	98.3	89.3	93.5	82.5
LightTS	87.1	78.4	82.5	82.4	75.8	78.9	**92.6**	55.3	69.2	92.0	94.7	93.3	98.4	96.0	97.1	84.2
FEDformer	87.9	82.4	85.1	77.1	80.1	78.6	90.5	58.1	70.8	90.2	96.4	93.2	97.3	97.2	97.2	85.0
ETSformer	87.4	79.2	83.1	85.1	84.9	85.0	92.2	55.7	69.5	90.0	80.4	84.9	**99.3**	85.3	91.8	82.9
PatchTS.	87.3	82.1	84.6	88.3	71.0	78.7	90.6	55.5	68.8	91.1	80.9	85.7	98.8	93.5	96.1	82.8
TimesNet*	87.9	81.5	84.6	89.5	75.4	81.8	90.1	56.4	69.4	90.7	95.4	93.0	98.5	96.2	**97.3**	85.2
GPT4TS‡	88.9	**85.0**	**86.9**	82.0	82.9	82.4	90.6	**60.9**	**72.9**	92.2	96.3	**94.2**	98.6	95.7	97.1	**86.7**
TSDE‡	87.5	82.2	84.8	**90.1**	84.5	**87.2**	91.4	56.9	70.1	**98.2**	92.9	92.5	98.6	90.7	94.5	85.8

* Reproduced with https://github.com/thuml/Time-Series-Library. † Reconstruction error is used as joint criterion for fair comparison.
‡ GPT4TS leverage a pretrained LLM (GPT-2) with 1.5B parameters, while TSDE merely uses two single-layer Transformer encoders.

Table 7: Classification performance on PhysioNet measured with AUROC. The baseline results are sourced from Table 2 of [6] and Table 3 of [24].

Models	AUROC
Mean imp. [24, 40]	0.70 ± 0.000
Forward imp. [24, 40]	0.71 ± 0.000
GP [60]	0.70 ± 0.007
VAE [35, 40]	0.68 ± 0.002
HI-VAE [54]	0.69 ± 0.010
GRUI-GAN [45]	0.70 ± 0.009
GP-VAE [24]	0.73 ± 0.006
GRU-D [10]	0.83 ± 0.002
M-RNN [94]	0.82 ± 0.003
BRITS-LR [6]†	0.74 ± 0.008
BRITS-RF [6]*	0.81 ± (N/A)
BRITS [6]	**0.85 ± 0.002**
TSDE	**0.85 ± 0.001**

† Logistic Regression (LR) on imputed PhysioNet data.
* Train Random Forest (RF) on imputed PhysioNet data.

In this task, TSDE is benchmarked against an extensive set of baselines featuring diverse backbones, including a) *Frozen pretrained LLM-based models*: GPT4TS [105]; b) *Task-agnostic foundation models*: TimesNet [86]; c) *MLP (multi-layer perceptron) based models*: LightTS [101] and DLinear [96]; and finally d) *Transformer-based models*: Transformer [81], Reformer [36], Informer [103], Autoformer [87], Pyraformer [41], LogSparse Transformer [38], FEDformer [104], Non-stationary Transformer [43], ETSformer [85], PatchTST [56] and Anomaly Transformer [89]. The results in Table 6 reveal that TSDE's anomaly detection performance surpasses nearly all baselines, with less than a 1% F1 score difference from GPT4TS. Notably, while TSDE doesn't outperform GPT4TS, it's important to consider that GPT4TS benefits from a pretrained LLM (GPT-2) with about 1.5 billion model parameters. TSDE, in contrast, relies on just two single-layer Transformer encoders (<0.3 million parameters), demonstrating its competitive edge despite having significantly fewer model parameters.

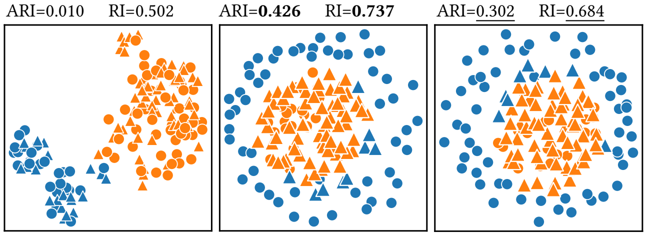
4.3 Classification

To further inspect the discriminative power of the pretrained TSDE embedding, we utilize the labeled PhysioNet dataset to evaluate TSDE's performance on a binary classification downstream task. This dataset, marked by in-hospital mortality labels for each patient, features MTS with over 80% missing values. To address this, we pretrain TSDE for 2,000 epochs to impute the raw MTS. Subsequently, we train a simple MLP for 40 epochs to perform mortality classification. Given the imbalanced nature of PhysioNet labels, we assess our model's efficacy with AUROC as in [6, 24]. We benchmark TSDE against a diverse range of established MTS classification methods, categorized into 3 groups with a total of 12 methods: (1) heuristic methods: mean/forward imputation [24, 40], (2) GP/VAE based models: GP [60], VAE [35], HI-VAE [54], GP-VAE [24], and (3) RNN based models: GRUI-GAN [45], GRU-D [10], M-RNN [94] and BRITS variants [6].

As shown in Table 7, TSDE surpasses all existing baselines and is on par with the state-of-the-art BRITS baseline. It is worth noting that BRITS achieves that performance by employing a sophisticated multi-task learning mechanism tailored for classification tasks. In contrast, our method achieves top-tier results by simply finetuning a simple MLP. TSDE's remarkable performance, especially in challenging classification scenarios with significant class imbalance (~10% positive classes), highlights its ability to learn generic embeddings well-suited for downstream MTS classification tasks.

4.4 Clustering

MTS data often lack annotations, making supervised learning inapplicable. In such scenarios, unsupervised clustering is a valuable method for uncovering intrinsic patterns and classes. We utilize the same pretrained TSDE model from our classification experiments (trained on PhysioNet with a 10% missing ratio) to evaluate the clustering performance of TSDE embeddings. Initially, we generate MTS embeddings using TSDE's pretrained embedding function. For simplicity and visual clarity, these embeddings are projected into a 2D space using UMAP (uniform manifold approximation and projection) [49]. Subsequently, DBSCAN (density-based spatial clustering of applications with noise) [23] is applied to these 2D projections to obtain clusters.



(a) Raw MTS (b) Embed raw MTS (c) Embed imputed MTS

Figure 3: Clustering of (a) raw MTS, (b) TSDE embedding of raw MTS, and (c) TSDE embedding of TSDE-imputed MTS. Marker shapes denote ground-truth binary labels; colors indicate DBSCAN [23] clusters after UMAP [49] dimension reduction.

As shown in Figure 3, the clustering quality is assessed across three data types: (a) raw MTS, (b) TSDE embeddings of raw MTS, and (c) TSDE embeddings of TSDE-imputed MTS. The ground truth binary labels are indicated using two distinct marker shapes: circles and triangles. When using raw MTS as seen in 3(a), the clusters are unfavourably intertwined, with data points from both classes intermingling. However, the TSDE embeddings, whether derived from raw or imputed MTS, exhibit substantially improved cluster differentiation. These embeddings enable more precise alignment with ground truth classifications, implying the capability of TSDE in capturing data nuances. Furthermore, the negligible performance disparity between Figures 3(b) and 3(c) suggests that TSDE embeddings can be directly used for MTS clustering without the need of imputation. This consistency is likely because our encoders proficiently encapsulate missing data traits, seamlessly integrating these subtleties into the embeddings. To provide a quantitative assessment of clustering, given the presence of labels, we calculate RI (rand index) [59] and ARI (adjusted RI) [30]. These metrics are reported on top of each setup in Figure 3. Notably, the RI and ARI values align with the qualitative observations discussed earlier, further substantiating our findings.

4.5 Embedding Visualization

To substantiate the representational efficacy of TSDE embeddings, we undertake a visualization experiment on synthetic MTS data, as showcased in Figure 4. The data comprises three distinct UTS: (a) a consistently ascending trend, (b) a cyclical seasonal signal, and (c) a white noise component. Each UTS embedding has two dimensions ($L \times 33$); for a lucid depiction, we cluster the second dimension by treating it as 33 samples each of length L, and visualize the centroid of these clusters. Intriguingly, the embeddings, which were pretrained on the entire synthetic MTS, vividly encapsulate the joint encoding effects of all series. The trend's embedding delineates the series' progression, evident from the gradual color saturation changes, embodying the steady evolution. The seasonal signal's embedding mirrors its inherent cyclicity, with color oscillations reflecting its periodic nature. Finally, the noise component's embeddings exhibit sporadic color band patterns (with subtle traces of seasonal patterns), capturing the inherent randomness.

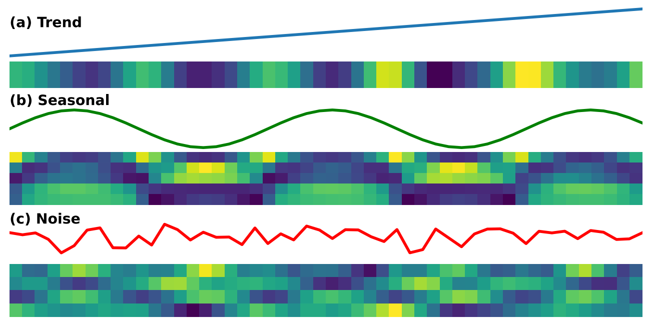


Figure 4: TSDE embedding visualization of (a) Trend, (b) Seasonal, and (c) Noise components from synthetic MTS.

5 CONCLUSION

In this paper, we propose TSDE, a novel SSL framework for TSRL. TSDE, the first of its kind, effectively harnesses a diffusion process, conditioned on an innovative dual-orthogonal Transformer encoder architecture with a crossover mechanism, and employs a unique IIF mask strategy. Our comprehensive experiments across diverse TS analysis tasks, including imputation, interpolation, forecasting, anomaly detection, classification, and clustering, demonstrate TSDE's superior performance compared to state-of-the-art models. Specifically, TSDE shows remarkable results in handling MTS data with high missing rates and various complexities, thus validating its effectiveness in capturing the intricate MTS dynamics. Moreover, TSDE not only significantly accelerates inference speed but also showcases its versatile embeddings through qualitative visualizations, encapsulating key MTS characteristics. This positions TSDE as a robust, efficient, and versatile advancement in MTS representation learning, suitable for a wide range of MTS tasks. Future work will focus on several key directions to address the limitation of slower inference for IIF tasks. Particularly, we will explore simplifying TSDE's architecture with a simple MLP without the need for the diffusion block, enabling the pretrained TSDE to execute IIF tasks independently.

ACKNOWLEDGMENTS

We would like to thank Yang Song (OpenAI & Stanford University) for his help to connect us with CSDI authors and the insightful discussions regarding time series representation learning.

R. Tu would like to acknowledge the support of Gustav Eje Henter, Hedvig Kjellström and the Wallenberg AI, Autonomous Systems and Software Program (WASP). R. Tu was also funded by the Industrial Strategic Technology Development Program (grant no. 20023495) from MOTIE, Korea.

REFERENCES

[1] Ahmed Abdulaal, Zhuanghua Liu, and Tomer Lancewicki. 2021. Practical Approach to Asynchronous Multivariate Time Series Anomaly Detection and Localization. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 2485–2494. https://doi.org/10.1145/3447548.3467174

[2] Juan Lopez Alcaraz and Nils Strodthoff. 2023. Diffusion-based Time Series Imputation and Forecasting with Structured State Space Models. *Transactions on Machine Learning Research* (2023).

[3] Defu Cao, Furong Jia, Sercan O Arik, Tomas Pfister, Yixiang Zheng, Wen Ye, and Yan Liu. 2023. TEMPO: Prompt-based Generative Pre-trained Transformer for Time Series Forecasting. *arXiv preprint arXiv:2310.04948* (2023).

[4] Lele Cao, Sahar Asadi, Wenfei Zhu, Christian Schmidli, and Michael Sjöberg. 2020. Simple, scalable, and stable variational deep clustering. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 108–124.

[5] Lele Cao, Sonja Horn, Vilhelm von Ehrenheim, Richard Anselmo Stahl, and Henrik Landgren. 2022. Simulation-informed revenue extrapolation with confidence estimate for scaleup companies using scarce time-series data. In *Proceedings of the 31st ACM international conference on information & knowledge management*. 2954–2963.

[6] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. 2018. BRITS: Bidirectional Recurrent Imputation for Time Series. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc.

[7] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*. 132–149.

[8] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems* 33 (2020), 9912–9924.

[9] Sravan Kumar Challa, Akhilesh Kumar, and Vijay Bhaskar Semwal. 2022. A multibranch CNN-BiLSTM model for human activity recognition using wearable sensor data. *The Visual Computer* 38, 12 (2022), 4095–4109.

[10] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. 2018. Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Scientific Reports* 8, 1 (2018), 6085.

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[12] Xinlei Chen and Kaiming He. 2021. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 15750–15758.

[13] Xinlei Chen, Zhuang Liu, Saining Xie, and Kaiming He. 2024. Deconstructing Denoising Diffusion Models for Self-Supervised Learning. *arXiv preprint arXiv:2401.14404* (2024).

[14] Yijiang Chen, Xiangdong Zhou, Zhen Xing, Zhidan Liu, and Minyang Xu. 2022. CaSS: A Channel-Aware Self-supervised Representation Learning Framework for Multivariate Time Series Classification. In *International Conference on Database Systems for Advanced Applications*. Springer, 375–390.

[15] Ranak Roy Chowdhury, Xiyuan Zhang, Jingbo Shang, Rajesh K Gupta, and Dezhi Hong. 2022. Tarnet: Task-aware reconstruction for time-series transformer. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 212–220.

[16] Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver. 2021. TimeVAE: A variational auto-encoder for multivariate time series generation. *arXiv preprint arXiv:2111.08095* (2021).

[17] Wenjie Du, David Côté, and Yan Liu. 2023. SAITS: Self-attention-based imputation for time series. *Expert Systems with Applications* 219 (2023), 119619.

[18] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee-Keong Kwoh, and Xiaoli Li. 2023. Label-efficient time series representation learning: A review. *arXiv preprint arXiv:2302.06433* (2023).

[19] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. 2021. Time-Series Representation Learning via Temporal and Contextual Contrasting. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. International Joint Conferences on Artificial Intelligence Organization, 2352–2359.

[20] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. 2018. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks* 107 (2018), 3–11.

[21] Linus Ericsson, Henry Gouk, Chen Change Loy, and Timothy M Hospedales. 2022. Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine* 39, 3 (2022), 42–62.

[22] Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. 2017. Real-valued (medical) time series generation with recurrent conditional GANs. *arXiv preprint arXiv:1706.02633* (2017).

[23] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Vol. 96. 226–231.

[24] Vincent Fortuin, Dmitry Baranchuk, Gunnar Raetsch, and Stephan Mandt. 2020. GP-VAE: Deep Probabilistic Time Series Imputation. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 108)*, Silvia Chiappa and Roberto Calandra (Eds.). PMLR, 1651–1661.

[25] Azul Garza and Max Mergenthaler-Canseco. 2023. TimeGPT-1. *arXiv preprint arXiv:2310.03589* (2023).

[26] Albert Gu, Karan Goel, and Christopher Ré. 2022. Efficiently Modeling Long Sequences with Structured State Spaces. In *The International Conference on Learning Representations (ICLR)*.

[27] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science* 313, 5786 (2006), 504–507.

[28] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

[29] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 6840–6851.

[30] Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of Classification* 2, 1 (1985), 193–218. https://doi.org/10.1007/BF01908075

[31] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (London, United Kingdom) *(KDD '18)*. Association for Computing Machinery, New York, NY, USA, 387–395. https://doi.org/10.1145/3219819.3219845

[32] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. 2017. Variational deep embedding: an unsupervised and generative approach to clustering. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 1965–1972.

[33] Longlong Jing and Yingli Tian. 2020. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence* 43, 11 (2020), 4037–4058.

[34] Zahra Karevan and Johan AK Suykens. 2020. Transductive LSTM for time-series prediction: An application to weather forecasting. *Neural Networks* 125 (2020), 1–9.

[35] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.).

[36] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The Efficient Transformer. In *The Eighth International Conference on Learning Representations*. https://openreview.net/pdf?id=rkgNKkHtvB

[37] Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. 2020. Prototypical Contrastive Learning of Unsupervised Representations. In *International Conference on Learning Representations*.

[38] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. 2019. *Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting*. Curran Associates Inc., Red Hook, NY, USA.

[39] Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in network. *arXiv preprint arXiv:1312.4400* (2013).

[40] Roderick J A Little and Donald B Rubin. 1986. *Statistical analysis with missing data*. John Wiley & Sons, Inc., USA.

[41] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X. Liu, and Schahram Dustdar. 2022. Pyraformer: Low-Complexity Pyramidal Attention for Long-Range Time Series Modeling and Forecasting. In *International Conference on Learning Representations*. https://openreview.net/forum?id=0EXmFzUn5I

[42] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *IEEE transactions on knowledge and data engineering* 35, 1 (2021), 857–876.

[43] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2022. Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 9881–9893.

[44] Wang Lu, Jindong Wang, Xinwei Sun, Yiqiang Chen, and Xing Xie. 2022. Out-of-distribution Representation Learning for Time Series Classification. In *The Eleventh International Conference on Learning Representations*.

[45] Yonghong Luo, Xiangrui Cai, Ying ZHANG, Jun Xu, and Yuan xiaojie. 2018. Multivariate Time Series Imputation with Generative Adversarial Networks. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2018/

file/96b9bff013acedfb1d140579e2fbeb63-Paper.pdf
[46] Liantao Ma, Chaohe Zhang, Yasha Wang, Wenjie Ruan, Jiangtao Wang, Wen Tang, Xinyu Ma, Xin Gao, and Junyi Gao. 2020. Concare: Personalized clinical feature embedding via capturing the healthcare context. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 833–840.
[47] Pankaj Malhotra, Vishnu TV, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. 2017. TimeNet: Pre-trained deep recurrent neural network for time series classification. In *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. 607–612.
[48] Aditya P. Mathur and Nils Ole Tippenhauer. 2016. SWaT: a water treatment testbed for research and training on ICS security. In *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*. 31–36. https://doi.org/10.1109/CySWater.2016.7469060
[49] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. 2018. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software* 3, 29 (2018), 861. https://doi.org/10.21105/joss.00861
[50] Mehran Mehralian and Babak Karasfi. 2018. RDCGAN: Unsupervised representation learning with regularized deep convolutional generative adversarial networks. In *2018 9th conference on artificial intelligence and robotics and 2nd Asia-pacific international symposium*. IEEE, 31–38.
[51] Qianwen Meng, Hangwei Qian, Yong Liu, Lizhen Cui, Yonghui Xu, and Zhiqi Shen. 2023. MHCCL: masked hierarchical cluster-wise contrastive learning for multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 9153–9161.
[52] Qianwen Meng, Hangwei Qian, Yong Liu, Yonghui Xu, Zhiqi Shen, and Lizhen Cui. 2023. Unsupervised Representation Learning for Time Series: A Review. arXiv:2308.01578 [cs.LG]
[53] Ahmad Wisnu Mulyadi, Eunji Jun, and Heung-Il Suk. 2022. Uncertainty-Aware Variational-Recurrent Imputation Network for Clinical Time Series. *IEEE Transactions on Cybernetics* 52, 9 (2022), 9684–9694. https://doi.org/10.1109/TCYB.2021.3053599
[54] Alfredo Nazabal, Pablo M. Olmos, Zoubin Ghahramani, and Isabel Valera. 2020. Handling Incomplete Heterogeneous Data using VAEs. arXiv:1807.03653 [cs.LG]
[55] Nam Nguyen and Brian Quanz. 2021. Temporal latent auto-encoder: A method for probabilistic multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 9117–9125.
[56] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2023. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *The Eleventh International Conference on Learning Representations*.
[57] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc.
[58] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).
[59] William M. Rand. 1971. Objective Criteria for the Evaluation of Clustering Methods. *J. Amer. Statist. Assoc.* 66, 336 (1971), 846–850. http://www.jstor.org/stable/2284239
[60] Carl Edward Rasmussen. 2004. *Gaussian Processes in Machine Learning*. Springer Berlin Heidelberg, Berlin, Heidelberg, 63–71. https://doi.org/10.1007/978-3-540-28650-9_4
[61] Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, Marin Biloš, Hena Ghonia, Nadhir Vincent Hassen, Anderson Schneider, et al. 2023. Lag-Llama: Towards foundation models for time series forecasting. *arXiv preprint arXiv:2310.08278* (2023).
[62] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. 2021. Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting. arXiv:2101.12072
[63] Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs M Bergmann, and Roland Vollgraf. 2021. Multivariate Probabilistic Time Series Forecasting via Conditioned Normalizing Flows. In *International Conference on Learning Representations*. https://openreview.net/forum?id=WiGQBFuVRv
[64] Yulia Rubanova, Ricky T. Q. Chen, and David K Duvenaud. 2019. Latent Ordinary Differential Equations for Irregularly-Sampled Time Series. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc.
[65] Alaa Sagheer and Mostafa Kotb. 2019. Unsupervised pre-training of a deep LSTM-based stacked autoencoder for multivariate time series forecasting problems. *Scientific reports* 9, 1 (2019), 19038.
[66] David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. 2019. High-dimensional multivariate forecasting with low-rank Gaussian Copula Processes. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc.
[67] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised Pre-Training for Speech Recognition. In *Proc. Interspeech 2019*. 3465–3469.
[68] Zineb Senane, Lele Cao, Valentin Leonhard Buchner, Yusuke Tashiro, Lei You, Pawel Herman, Mats Nordahl, Ruibo Tu, and Vilhelm von Ehrenheim. 2024. Self-Supervised Learning of Time Series Representation via Diffusion Process and Imputation-Interpolation-Forecasting Mask. (2024). arXiv:2405.05959
[69] Satya Narayan Shukla and Benjamin Marlin. 2021. Multi-Time Attention Networks for Irregularly Sampled Time Series. In *International Conference on Learning Representations*. https://openreview.net/forum?id=4c0J6lwQ4_
[70] I Silva, G Moody, DJ Scott, LA Celi, and RG Mark. 2012. Predicting In-Hospital Mortality of ICU Patients: The PhysioNet/Computing in Cardiology Challenge 2012. *Computing in Cardiology* 39 (2012), 245–248.
[71] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*. PMLR, 2256–2265.
[72] Wei Song, Lu Liu, Minghao Liu, Wenxiang Wang, Xiao Wang, and Yu Song. 2020. Representation learning with deconvolution for multivariate time series classification and visualization. In *Data Science: 6th International Conference of Pioneering Computer Scientists, Engineers and Educators, ICPCSEE 2020, Taiyuan, China, September 18-21, 2020, Proceedings, Part I 6*. Springer, 310–326.
[73] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. 2021. Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems* 34 (2021), 1415–1428.
[74] Yang Song and Stefano Ermon. 2019. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems* 32 (2019).
[75] Yang Song and Stefano Ermon. 2020. Improved techniques for training score-based generative models. *Advances in neural information processing systems* 33 (2020), 12438–12448.
[76] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*.
[77] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 2019. Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 2828–2837. https://doi.org/10.1145/3292500.3330672
[78] Chenxi Sun, Yaliang Li, Hongyan Li, and Shenda Hong. 2023. TEST: Text Prototype Aligned Embedding to Activate LLM's Ability for Time Series. *arXiv preprint arXiv:2308.08241* (2023).
[79] Yaling Tao, Kentaro Takagi, and Kouta Nakata. 2020. Clustering-friendly Representation Learning via Instance Discrimination and Feature Decorrelation. In *International Conference on Learning Representations*.
[80] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. 2021. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems* 34 (2021), 24804–24816.
[81] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc.
[82] Phil Wang. 2020. *Linear attention transformer*. Technical Report. https://github.com/lucidrains/linear-attention-transformer
[83] Zhixian Wang, Qingsong Wen, Chaoli Zhang, Liang Sun, and Yi Wang. 2023. DiffLoad: Uncertainty Quantification in Load Forecasting with Diffusion Model. *arXiv preprint arXiv:2306.01001* (2023).
[84] Haomin Wen, Youfang Lin, Yutong Xia, Huaiyu Wan, Roger Zimmermann, and Yuxuan Liang. 2023. DiffSTG: Probabilistic spatio-temporal graph forecasting with denoising diffusion models. *arXiv preprint arXiv:2301.13629* (2023).
[85] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. 2023. ETSformer: Exponential Smoothing Transformers for Time-series Forecasting. https://openreview.net/forum?id=5m_3whfo483
[86] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=ju_Uqw384Oq
[87] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (Eds.), Vol. 34. Curran Associates, Inc., 22419–22430.
[88] Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*. PMLR, 478–487.

[89] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2022. Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy. In *International Conference on Learning Representations*. https://openreview.net/forum?id=LzQQ89U1qm_

[90] Ling Yang and Shenda Hong. 2022. Unsupervised time-series representation learning with iterative bilinear temporal-spectral fusion. In *International Conference on Machine Learning*. PMLR, 25038–25054.

[91] Xinyu Yang, Zhenguo Zhang, and Rongyi Cui. 2022. TimeCLR: A self-supervised contrastive learning framework for univariate time series representation. *Knowledge-Based Systems* 245 (2022), 108606.

[92] Xiuwen Yi, Yu Zheng, Junbo Zhang, and Tianrui Li. 2016. ST-MVL: filling missing values in geo-sensory time series data. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* (New York, New York, USA) *(IJCAI'16)*. AAAI Press, 2704–2710.

[93] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. 2019. Time-series generative adversarial networks. *Advances in neural information processing systems* 32 (2019).

[94] Jinsung Yoon, William R. Zame, and Mihaela van der Schaar. 2017. Multi-directional Recurrent Neural Networks : A Novel Method for Estimating Missing Data. https://api.semanticscholar.org/CorpusID:39429412

[95] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. 2022. TS2Vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 8980–8987.

[96] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are Transformers Effective for Time Series Forecasting? *Proceedings of the AAAI Conference on Artificial Intelligence.*

[97] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. 2021. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 2114–2124.

[98] Dejiao Zhang, Feng Nan, Xiaokai Wei, Shang-Wen Li, Henghui Zhu, Kathleen Mckeown, Ramesh Nallapati, Andrew O Arnold, and Bing Xiang. 2021. Supporting Clustering with Contrastive Learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 5419–5430.

[99] Kexin Zhang and Yong Liu. 2021. Unsupervised Feature Learning with Data Augmentation for Control Valve Stiction Detection. In *2021 IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS)*. IEEE, 1385–1390.

[100] Kexin Zhang, Qingsong Wen, Chaoli Zhang, Rongyao Cai, Ming Jin, Yong Liu, James Zhang, Yuxuan Liang, Guansong Pang, Dongjin Song, et al. 2023. Self-Supervised Learning for Time Series Analysis: Taxonomy, Progress, and Prospects. *arXiv preprint arXiv:2306.10125* (2023).

[101] Tianping Zhang, Yizhuo Zhang, Wei Cao, Jiang Bian, Xiaohan Yi, Shun Zheng, and Jian Li. 2022. Less Is More: Fast Multivariate Time Series Forecasting with Light Sampling-oriented MLP Structures. arXiv:2207.01186 [cs.LG]

[102] Wenrui Zhang, Ling Yang, Shijia Geng, and Shenda Hong. 2023. Self-supervised time series representation learning via cross reconstruction transformer. *IEEE Transactions on Neural Networks and Learning Systems* (2023).

[103] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 11106–11115.

[104] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 27268–27286.

[105] Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. 2023. One Fits All: Power General Time Series Analysis by Pretrained LM. In *Thirty-seventh Conference on Neural Information Processing Systems*.

[106] Yanqiao Zhu, Yichen Xu, Qiang Liu, and Shu Wu. 2021. An Empirical Study of Graph Contrastive Learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

[107] Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. 2020. Transformer hawkes process. In *International conference on machine learning*. PMLR, 11692–11702.

A SUPPLEMENTARY MATERIAL

In our research, we have included an extensive appendix in the preprint, which provides supplementary material essential for a comprehensive understanding of our methodologies and results. The appendix encompasses detailed descriptions of the datasets, hyperparameters, and metric formulas used in our evaluations with their definitions and explanations. These descriptions ensure that readers can fully grasp the context and settings under which our experiments were conducted. Additionally, the appendix offers pseudo-code for the masking strategies employed in imputation, interpolation, and forecasting tasks, enabling reproducibility and a deeper understanding of our approach. Furthermore, it provides architectural details of the projection head for anomaly detection and the classifier head for classification, highlighting the technical design choices of applying our TSDE framework to different downstream tasks.

Moreover, we present additional results for forecasting, comparing TSDE to recent baselines and demonstrating the robustness and versatility of our proposed TSDE framework across various scenarios. The appendix also includes comprehensive documentation on the implementation aspects, offering insights into the practical considerations and challenges faced during the development and evaluation of the TSDE framework. This detailed supplementary material is crucial for understanding the depth and rigor of the experiments conducted, as well as the significance of our findings. For a detailed exploration of these methodologies and findings, readers are encouraged to refer to the full appendix in the preprint [68].

B KEY FINDINGS AND LIMITATIONS

This work explored the integration of diffusion models and transformer encoders for TSRL. Our results indicate that conditioning the diffusion model on learned embeddings improves performance across tasks such as imputation, interpolation, and forecasting. Additionally, the embedding block within TSDE proves highly effective in handling sparse data by leveraging the SSL task of imputing missing values. The use of IIF masking facilitated the model's robustness to sparse data, and handling of different missingness scenarios.

Despite these advancements, the TSDE model has few limitations. The iterative nature of the denoising diffusion probablistic model can lead to slower inference, presenting a trade-off between enhanced quality and efficiency in real-world applications. Incorporating the SSL pretext task of IIF masking requires additional training epochs, which can be a constraint for rapid model deployment and retraining. Furthermore, while TSDE outperforms other methods, there remains a gap in perfectly matching the ground truth in highly noisy scenarios. Addressing these limitations will be crucial for further improving the model's robustness and applicability.