# Score-CDM: Score-Weighted Convolutional Diffusion Model for Multivariate Time Series Imputation

**Shunyang Zhang**[1] , **Senzhang Wang**[1] , **Hao Miao**[2] , **Hao Chen**[3]
**Changjun Fan**[4] and **Jian Zhang**[1]
[1]Central South University
[2]Aalborg University
[3]The Hong Kong Polytechnic University
[4]National University of Defense Technology
{224712166, szwang}@csu.edu.cn, haom@cs.aau.dk, sundaychenhao@gmail.com,
fanchangjun@nudt.edu.cn, jianzhang@csu.edu.cn

## Abstract

Multivariant time series (MTS) data are usually incomplete in real scenarios, and imputing the incomplete MTS is practically important to facilitate various time series mining tasks. Recently, diffusion model-based MTS imputation methods have achieved promising results by utilizing CNN or attention mechanisms for temporal feature learning. However, it is hard to adaptively trade off the diverse effects of local and global temporal features by simply combining CNN and attention. To address this issue, we propose a Score-weighted Convolutional Diffusion Model (Score-CDM for short), whose backbone consists of a Score-weighted Convolution Module (SCM) and an Adaptive Reception Module (ARM). SCM adopts a score map to capture the global temporal features in the time domain, while ARM uses a Spectral2Time Window Block (S2TWB) to convolve the local time series data in the spectral domain. Benefiting from the time convolution properties of Fast Fourier Transformation, ARM can adaptively change the receptive field of the score map, and thus effectively balance the local and global temporal features. We conduct extensive evaluations on three real MTS datasets of different domains, and the result verifies the effectiveness of the proposed Score-CDM.

## 1 Introduction

The perpetual integration of sensing technologies results in the generation of increasingly voluminous time series data, contributing to various practical applications such as urban planning [Acevedo and Masuoka, 1997], city renewal [Lim and Zohren, 2021], and traffic management [Wang *et al.*, 2020; Miao *et al.*, 2024; Wang *et al.*, 2022; Yao *et al.*, 2023; Fang *et al.*, 2021],. Nevertheless, in practical scenarios, the time series data are usually incomplete due to various issues including sensor failure and communication errors. To this end, multivariate time series (MTS) imputation has attracted rising research interest in both academic and industrial communities in recent years, finding applications across diverse domains such as finance, healthcare, and industrial manufacturing [Lim and Zohren, 2021].

Traditionally, statistical-based machine learning methods are widely adopted for MTS imputation, such as ARIMA [Nelson, 1998] and KNN [Peterson, 2009]. These methods typically can capture the linear properties of time series, but may not be effective to model the complex and nonlinear temporal correlations. Recent progress in deep learning has brought about more effective MTS imputation techniques, including RNN, CNN, and Attention. RNNs continuously update the hidden states to capture the temporal information for time series data. Some RNN-based methods [Cini *et al.*, 2022; Wang *et al.*, 2022; Li *et al.*, 2022] adopt attention to further consider the temporal feature correlations. A major issue of RNN-based methods is the accumulation of errors [Liu *et al.*, 2023] that can result in inaccurate imputations. Attention-based models [Marisca *et al.*, 2022] can effectively model the long-term temporal features [Si *et al.*, 2022], but may overlook the local and short-term temporal correlation. Although TCN can effectively capture the long-term dependencies of time series due to its larger receptive field [Liu *et al.*, 2022], its temporal feature learning capacity is still largely limited by the size of the receptive field.

Recently, diffusion models (DM) [Ho *et al.*, 2020], recognized by their powerful generative learning capability and great success in data generation tasks, have also been applied to impute MTS and achieved SOTA performance [Tashiro *et al.*, 2021; Liu *et al.*, 2023]. These models tackle the data imputation task by creating conditional guidance, aiming to bring the diffusion process and backward process to an accurate outcome. In simpler terms, they create a process that learns a map from the ground truth to noise and another map to reconstruct data from noise [Yang *et al.*, 2022]. The flexibility of DM in neural network architecture allows it to incorporate different deep learning models (e.g. CNN and attention) as its denoising function [Tashiro *et al.*, 2021].

Despite their effectiveness, existing diffusion models have overlooked the design of a suitable denoising function that can effectively capture the global and local temporal features

for time series imputation [Liu *et al.*, 2023]. Directing adopting attention mechanism as the denoising function can effectively capture global temporal features (e.g., weekly traffic flow patterns), but the local temporal features (e.g., hourly traffic flow features) may not be effectively learned [Si *et al.*, 2022]. TCN can model larger and structured receptive fields by combining structured transformation with CNN. However, it still lacks of a broader receptive field compared to attention mechanisms. Approaches like attention-free transformer [Zhai *et al.*, 2021] attempt to integrate CNN to capture the local correlations. However, this type of method handles the local and global temporal features separately, and thus is hard to adaptively balance the diverse effects of the two types of features.

In this paper, we propose a **Score**-Weighted **C**onvolutional **D**iffusion **M**odel named Score-CDM to effectively and adaptively learn the local and global time series features for MTS imputation. Specifically, Score-CDM contains a Score-weighted Convolution Module (SCM) and an Adaptive Reception module (ARM). SCM can generate a globally attentive convolution kernel, and ARM can construct a time window that regulates the receptive field of this kernel. We derive a globally attentive convolution operation by multiplying this kernel with the elements within the receptive field. To delve deeper, we propose a Spectral2time Window Block (S2TWB) in ARM, which can adaptively change the receptive field in the spectral domain by using the Fast Fourier Transform (FFT). By leveraging the convolution properties of FFT, S2TWB establishes a flexible receptive field to achieve a balance between the global and local temporal features by learning to weight each time step in the time window. SCM and ARM together work as the denoising function of Score-CDM for more effective MTS imputation. We conduct extensive evaluations on three real-world MTS datasets. The results show the superior performance of our proposed method by comparison with current SOTA models. We summarize our contributions as follows.

- We introduce Score-CDM, a score-weighted convolutional diffusion model whose denoising function contains the novel SCM and ARM. Score-CDM can more effectively capture the global and local temporal features for MTS imputation.
- A spectral2time window block S2TWB is proposed to adaptively change the receptive field of the kernel generated by SCM on the spectral domain by adopting FFT. Due to the time convolution properties of FFT, the extracted receptive field is structured and flexible (e.g. continuous or discretized).
- We conduct extensive evaluations on three real-world MTS datasets of different domains. The result verifies the superior performance of our proposal, and also demonstrates the effectiveness of the two proposed modules SCM and ARM in temporal feature learning.

## 2 Related Work

Multivariate time series imputation attracts increasing interest due to the increasing availability of time series data and rich applications, such as statistical methods [Lee and Fambro, 1999] and deep models [Gu *et al.*, 2022]. In the early stage, time series imputation methods are mostly based on statistical models, such as Matrix Factorization (MF) [Lee and Fambro, 1999], and Multiple Imputation using Chained Equations. However, their linear properties make them hard to capture the dynamic features of time series. Recently, various deep learning-based methods are proposed to address time series imputation [Che *et al.*, 2018; Wang *et al.*, 2021; Yoon *et al.*, 2017]. BRITS utilized a simple linear regression layer to incorporate spatial information and adopted bidirectional RNNs architecture for time series imputation [Cao *et al.*, 2018]. SAITS introduced self-attention to capture the global temporal relations of time series [Du *et al.*, 2023]. SPIN adopted a joint attention that combined spatial and temporal attention to model information exchange between time series variants [Marisca *et al.*, 2022].

Motivated by the great success of generative models, multiple generative model based MTS imputation methods are also proposed and achieved SOTA performance. GAINFilling used GAN models to generate sequences by matching the underlying data distribution [Luo *et al.*, 2018]. Conditional Score-based Diffusion models for Imputation (CSDI) is a paradigmatic example of applying the diffusion models in MTS imputation [Tashiro *et al.*, 2021]. CSDI presented a novel time series imputation method that leveraged score-based diffusion models. Following CSDI, Structured State Space Diffusion (SSSD) integrated conditional diffusion models and structured state-space models to particularly capture long-term dependencies in time series [Alcaraz and Strodthoff, 2022]. PriSTI applied spatial information to guide the generation of the missing time series values [Liu *et al.*, 2023]. Unlike CSDI and SSSD, TimeDiff [Shen and Kwok, 2023] introduced additional inductive bias in the conditioning module to achieve long-time series forecasting. Diffusion model based methods generally perform well in time series imputation. However, existing diffusion model based methods still suffer from the issue of lacking sufficient temporal and global temporal feature learning capacity because their backbone denoising function directly adopts attention or CNN models. How to design a new denoising function that is more suitable to deal with the MTS data is not well studied.

## 3 Preliminary and Problem Definition

In this section, we will first define the studied problem, and then briefly introduce Fourier transformation and diffusion probabilistic models.

**Problem Definition 1.** *Given the incomplete multivariate time series $X \in \mathbb{R}^{N \times C \times L}$ with some missing values, we aim to build a model $\epsilon_\theta$ to impute $X \in \mathbb{R}^{N \times C \times L}$ and obtain the complete data $\widetilde{X}$, where $N$ is the number of time series variables, $C$ is the number of channels and $L$ is the length of the time series.*

**Fourier Operator** We define $\mathcal{S} = \mathcal{F}(\kappa) \in \mathbb{C}^{N \times C \times L}$ as a Fourier Operator (FO), where $\mathcal{F}$ denotes Discrete Fourier Transform (DFT). According to the convolution theorem (see Appendix), we can write the multiplication between $\mathcal{F}(X)$

and $\mathcal{S}$ in Fourier space as follows,

$$\begin{aligned}\mathcal{F}(X)\mathcal{F}(\kappa) &= \mathcal{F}((X * \kappa)[i]) \\ &= \mathcal{F}(\sum_{j=1}^{n} X[j]\kappa[i-j]) \\ &= \mathcal{F}(\sum_{j=1}^{n} X[j]\kappa[i,j]), \qquad \forall i \in [1, ..., n]\end{aligned} \tag{1}$$

where $(X * \kappa)[i]$ denotes the convolution of $X$ and $\kappa$. As defined $\kappa[i,j] = W$ ($W \in R^{n\times n}$), it yields $\sum_{j=1}^{n} X[j]\kappa[i,j] = \sum_{j=1}^{n} X[j]W = XW$. Accordingly, we can get the convolution equation as follows,

$$\mathcal{F}(X)\mathcal{S} = \mathcal{F}(XW). \tag{2}$$

From Eq.2, one can observe that performing the multiplication between $\mathcal{F}(X)$ and $\mathcal{S}$ in Fourier space corresponds to a time shift operation in Eq.1 (i.e., a temporal convolution) in the time domain. Since the multiplication in Fourier space has much lower complexity ($\mathcal{O}(t \log t)$) than the above shift operations ($\mathcal{O}(t^2)$) in the time domain, we adopt FFT to make a more efficient convolution on the time domain.

**Diffusion process and reverse process.** The *diffusion process* for MTS imputation adds Gaussian noise into the original data, which can be formalized as follows,

$$\begin{aligned}q(\widetilde{X}^{1:T}|\widetilde{X}^0) &= \prod_{t=1}^{T} q(\widetilde{X}^t|\widetilde{X}^{t-1}), \\ q(\widetilde{X}^t|\widetilde{X}^{t-1}) &= \mathcal{N}(\widetilde{X}^t; \sqrt{1-\beta_t}\widetilde{X}^{t-1}, \beta_t \boldsymbol{I}),\end{aligned} \tag{3}$$

where $\beta_t$ is a small constant hyperparameter that controls the variance of the added noise. $\widetilde{X}^t$ is sampled by $\widetilde{X}^t = \sqrt{\bar{\alpha}_t}\widetilde{X}^0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, where $\alpha_t = 1-\beta_t$, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, and $\epsilon$ is the sampled standard Gaussian noise. When $T$ is large enough, $q(\widetilde{X}^T|\widetilde{X}^0)$ is close to the standard normal distribution.

The *reverse process* for MTS imputation gradually converts random noise to the missing values with spatiotemporal consistency based on conditional information. In this work, the reverse process is conditioned on the interpolated conditional information $\mathcal{X}$ (conditional guidance) that enhances the observed values, which can be formalized as follows,

$$\begin{aligned}p_\theta(\widetilde{X}^{0:T-1}|\widetilde{X}^T, \mathcal{X}) &= \prod_{t=1}^{T} p_\theta(\widetilde{X}^{t-1}|\widetilde{X}^t, \mathcal{X}), \\ p_\theta(\widetilde{X}^{t-1}|\widetilde{X}^t, \mathcal{X}) &= \mathcal{N}(\widetilde{X}^{t-1}; \mu_\theta(\widetilde{X}^t, \mathcal{X}, t), \sigma_t^2 \boldsymbol{I}).\end{aligned} \tag{4}$$

# 4 Methodology

The schematic representation of the denoising function of Score-CDM in the diffusion process is depicted in Figure 1. The designed denoising function mainly contains a Score-weighted convolution module (SCM) and a Adaptive Reception Module. The SCM undergoes two key steps: matrix projection and information exchange (the upper red line part). ARM undergoes one step: receptive field generation (the lower blue line part). The matrix projection involves

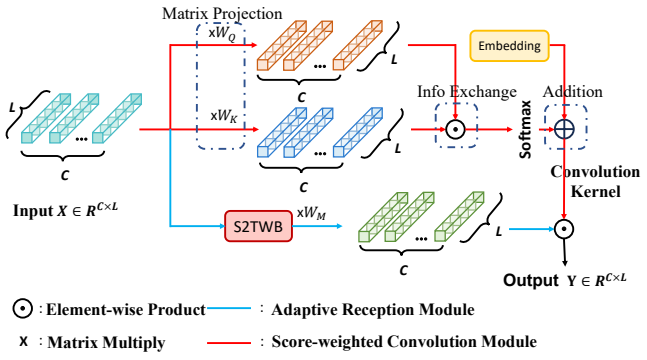



Figure 1: The framework of the Score-CDM denoising function.

the operation of multiplying raw data $X_i \in R^{C\times L}$ with two learnable matrices $W_K$ and $W_Q$, whose function is similar to the attention mechanism. In the information exchange phase, the element-wise product of $Q$ and $K$ is computed, facilitating the comprehensive multiplication of time series elements. This process enables the learning of a globally attentive score map through the application of Softmax and an additional embedding. For ARM, it contains a S2TWB block whose detailed illustration is given in Figure 2. As indicated by the blue line in Figure 1, the raw data undergoes processing through the S2TWB block and matrix $W_M$, resulting in data aggregation within a specified time window. The time window (receptive field) from ARN and the score map from SCM are then multiplied using the element-wise product. Essentially, the score map acts as a convolution kernel to convolve the elements in the time window (receptive field) for this kernel. Next, we will provide a detailed exploration of these components.

## 4.1 Score-weighted Convolution Module

**Matrix Projection.** This step aims to aggregate the information of each time step in a time series. $X$ is the input multivariant time series, and $W$ is the injection matrix. $X_i$ is the $i$-th row of $X$ and $W_j$ is the $j$-th column of $W$.

$$\begin{aligned}&Q_{i,j} = X_i W_j^Q, K_{i,j} = X_i W_j^K, M_{i,j} = X_i W_j^M \\ &X \in \mathbb{R}^{C\times L}, W \in \mathbb{R}^{L\times L},\end{aligned} \tag{5}$$

We learn $Q$ and $K$ in a similar way as attention. After this operation, we send $Q$ and $K$ to the next nonlinear element-wise mixing step for information exchange.

**Information Exchange.** In this step, the product of element pairs allows for a thorough computation on the features of $X_{i,j}$. To consider the global correlation between element pairs, the output of information exchange undergoes a Softmax layer to weight the importance of all $X_{i,j}, j = 1, ..., C$. However, traditional dot product operation considers all time series of $X_i$, and the combination with Softmax leads to redundant computation on element pairs. To address this issue, we generate a series of convolutional kernels whose size is equal to $Q$ as follows

$$Q_{i,j} \odot K_{i,j} = (X_i W_j^Q)(X_i W_j^K), \tag{6}$$

based on which we can further derive

$$\sum X_i W_j \times \sum X_i W_j \rightarrow \sum_k \sum_m X_{i,k} X_{i,m} WW. \quad (7)$$

For the attention mechanism, its computation of time series is equal to our model but greater on channel aspect (different channel of time series) in element view (more details will presented in Discussion):

$$\sum_k (\sum X_i W_k \times \sum X_k W_j) \rightarrow \sum_k \sum_p \sum_q X_{i,p} X_{k,q} W. \quad (8)$$

Then, through the Softmax function, we gain the weighted kernel (score map) as follows,

$$Kernel = softmax(Q_{i,j} \odot K_{i,j}). \quad (9)$$

Finally, we add an embedding to the kernel as a random shift operation similar to Attention-Free Transformer [Zhai *et al.*, 2021],

$$Kernel = Kernel + Embedding. \quad (10)$$

The embedding is initialized in random and it is similar to an attention weights shift as in [Zhai *et al.*, 2021].

## 4.2 Adaptive Reception Module

**Spectral2Time Window Block.** This module uses a self-attention kernel named **S**pectral**2T**ime **W**indow **B**lock (S2TWB) to convolve the time series based on the convolution theorem in Eq. 1 as follows

$$\mathcal{F}(K * X) = \mathcal{F}(K) \cdot \mathcal{F}(X), \quad (11)$$

where $*$ is a convolution operation and $\cdot$ is a multiply operation. $\mathcal{F}$ is the Fast Fourier Transform. We aim to generate a series of kernels as $\mathcal{K}_{\theta_i}$ and aggregate them together to generate the kernel $\mathcal{K}_\theta$ as follows

$$\mathcal{K}_\theta = \sum_{i=1}^{L} w_i \mathcal{K}_{\theta_i}, \quad (12)$$

where $L$ is the time series length and $\mathcal{K}_{\theta_i}$ ($i = 1$ to $L$) are basis operators with learnable parameters $\{w_i\}_{i=1}^{L}$. Here we use $sin()$ function to present the basic operator, and we have

$$\mathcal{K}_{\theta_i}(X) = \sin(iX). \quad (13)$$

Then the kernel $\mathcal{K}_\theta$ of S2TWB can be reformulated as follows

$$\mathcal{K}_\theta = \mathbf{w}_{\sin} \begin{bmatrix} \sin(x) \\ \vdots \\ \sin(ix) \end{bmatrix} \quad (14)$$

where $\mathbf{w}_{\sin}[i] = w_i$. Finally, we apply FFT on kernel $\mathcal{K}_\theta$ and $X$ as follows,

$$\mathcal{K}_\theta * X = \mathcal{F}^{-1}(\mathcal{F}(\mathcal{K}_\theta) \cdot \mathcal{F}(X)). \quad (15)$$

Next, we will demonstrate how to use FFT and kernel $\mathcal{K}_\theta$ to adaptively change the receptive field. FFT is a convolutional method for time series as mentioned in section 3. As shown in Figure 2, S2TWB uses FFT to project kernel and time series $X$ in the time domain to the spectral domain and multiply them, after which the kernel generated in the time domain convolves time series like the CNN kernel. Specifically, the kernel $\mathcal{K}_\theta$ is combined linearly with several waves as shown in the left side of Figure 2. Then the kernel allocates different attention weights to different positions of a time series based on the distance between itself and the target position in the time domain as shown in the left part of Figure 2. Convolution via FFT only relies on the relative position but not on the absolute position, which is more flexible.

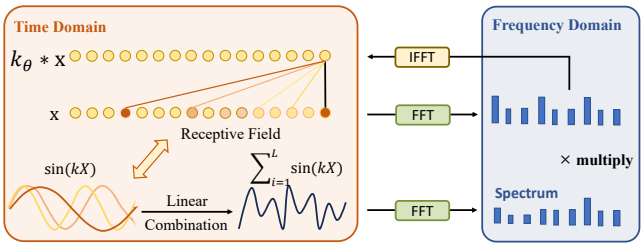



Figure 2: Illustration of Spectral2Time Window Block

**Overall formulation.** The overall mathematical presentation of the denoising function can be written as follows

$$Y = f(X);$$
$$Y_{c,t'} = \frac{\mathcal{K}_\theta * ((\exp(Q_{c,t'} \odot K_{c,t'}) + w_{c,t'}) \odot M_{c,t'})}{\sum_{t'=1}^{T} \exp(Q_{c,t'} \odot K_{c,t'})}, \quad (16)$$

where $Y$ is the output of the representation, $X$ is the input and $t_\tau$ is the structured receptive field. Then we learn a convolution operation as follows

$$\mathcal{K}_{\theta(i,j)} Q_{i,j} = \mathcal{K}_{\theta(i,j)} X_{i,t_\tau} W^M_{j,t_\tau}, \quad (17)$$

which presents the $\mathcal{K}_{\theta(i,j)}$ convolves the $X_{i,t_\tau}$.

## 4.3 Discussion

**Relationship with Convolution.** The convolutional kernel $K$ projects onto the time series data $X$, essentially performing a dot product between two vectors. In Score-CDM, the transformation matrix $M$ projects a portion of the vector $X_{i,t_0:t_1}$ onto unit elements, and the result is multiplied by a convolutional kernel with a size of $1 \times 1$. This is equivalent to convolving $X_{i,t_0:t_1}$ with a kernel of size $(t_1 - t_0) \times 1$.

**Relationship with Transformer.** For the attention mechanism, after time-mixing and element-mixing, the temporal values on the nodes are thoroughly blended, resulting in $\sum X \times \sum X \rightarrow \sum_i \sum_j X_i X_j$. At this point, applying Softmax yields attention weights that span the entire time series.

We compare Score-CDM with four classic models including Attention Free Transformer, Transformer, TCN, and Dlinear (MLP-based Model) from four aspects, whether can be coded as attention or convolution, the computational metrics (element-wise or dot-product), and through full or structured transformation. Through the comparison, one can see that Score-CDM can be considered as a convolution model with a global attention score map, and its receptive field is flexible.

For a clearer comparison among these methods, one can refer to the Appendix.

### 4.4 Overview Diffusion Architecture

We design a similar training process and backward process as PriSTI [Liu *et al.*, 2023]. In the training process, we train a map from diffusion step $t$ to noise $\epsilon$. In other words, our model learns to predict noise intensity in each diffusion step. We finally get a noise estimation function $\epsilon_\theta$ to denoise data step by step. To capture intervariate correlation, we additionally add an attention mechanism into $\epsilon_\theta$ to directly compute the data in variate dimensions. Its input is $Y$ and its output is the predicted noise.

**Algorithm 1** Training process of Score-CDM.

**Require:** Incomplete MTS data $X$, the adjacency matrix $A$, the number of iterations $ite$, the number of diffusion steps $T$, noise levels sequence $\bar{\alpha}_t, \bar{\beta}_t\epsilon$.
**Ensure:** Optimized noise prediction model $\epsilon_\theta$.
1: Insert the designed backbone into $\epsilon_\theta$ as the time exaction module;
2: **for** $i = 1$ **to** $ite$ **do**
3: $\widetilde{X}^0 \leftarrow \text{Mask}(X)$;
4: $\mathcal{X} \leftarrow \text{Interpolate}(\widetilde{X}^0)$;
5: Sample $t \sim \text{Uniform}(\{1, \cdots, T\})$, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$;
6: Calculate noise $\widetilde{X}^t \leftarrow \bar{\alpha}_t \widetilde{X}^0 + \bar{\beta}_t\epsilon$;
7: Update the gradient $\nabla_\theta \left\| \epsilon - \epsilon_\theta(\widetilde{X}^t, \mathcal{X}, A, t) \right\|^2$.
8: **end for**

**Algorithm 2** Imputation process with Score-CDM.

**Require:** The incomplete MTS data $X$, the adjacency matrix $A$, the number of diffusion steps $T$, the optimized noise prediction model $\epsilon_\theta$, and noise levels sequence $\bar{\alpha}_t, \bar{\beta}_t$.
**Ensure:** Missing values of the imputation target $\widetilde{X}^0$, where $\widetilde{X}$ is equal to $\widetilde{X}^0$.
1: $\mathcal{X} \leftarrow \text{Interpolate}(X)$;
2: Set $\widetilde{X}^T \sim \mathcal{N}(0, \mathbf{I})$;
3: **for** $t = T$ **to** $1$ **do**
4: $\mu_\theta(\widetilde{X}^t, t) \leftarrow \frac{1}{\sqrt{\bar{\alpha}_t}} \left( \widetilde{X}^t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\widetilde{X}^t, \mathcal{X}, A, t) \right)$;
5: $\widetilde{X}^{t-1} \leftarrow \mathcal{N}(\mu_\theta(\widetilde{X}^t, t), \sigma_t^2 \boldsymbol{I})$;
6: **end for**

When using the trained noise prediction model $\epsilon_\theta$ for imputation, we aim to impute the incomplete MTS data $X$, and the interpolated conditional information $\mathcal{X}$ is constructed based on all observed values. The model receives $\widetilde{X}^T$ and $\mathcal{X}$ as inputs and generates samples of the imputation results through the reverse process in Eq. (4).

## 5 Experiment

### 5.1 Experiment Setup

**Dataset.** We evaluate the performance of our model on three real-world datasets METR-LA, AQI-36, and PEMS-BAY that are widely adopted for MTS imputation in previous works. METR-LA and PEMS-BAY are the traffic flow datasets collected from traffic sensors in Los Angeles County Highway and Bay Areas in California. AQI-36 is collected from 36 AQI sensors distributed across the city of Beijing. The detailed dataset statistics are given in the appendix.

For the dataset AQI-36, we adopt the same evaluation strategy as the previous work [Yi *et al.*, 2016]. For the traffic datasets METR-LA and PEMS-BAY, more details will show in Appendix.

**Baselines.** We compare Score-CDM with the following baselines.

- **Transformer** [Vaswani *et al.*, 2017] is based on multi-head self-attention mechanism.
- **BRITS** [Cao *et al.*, 2018] employs bidirectional RNN and MLP to learn spatio-temporal information for MTS imputation.
- **CSDI** [Tashiro *et al.*, 2021] is a recent SOTA MTS imputation method that is based on the conditional diffusion probability model.
- **TimesNet** [Wu *et al.*, 2022] contains a self-organized convolution model for time series imputation.
- **SPIN** [Marisca *et al.*, 2022] employs threshold graph attention and temporal attention to jointly model the spatio-temporal dependencies of time series.
- **GRIN** [Cini *et al.*, 2022] is a bidirectional GRU-based method with a graph neural network.
- **SAITS** [Du *et al.*, 2023] is based on diagonally-masked self-attention mechanism for MTS imputation.
- **PriSTI** [Liu *et al.*, 2023] incorporates spatial conditions into attention to reduce the discrepancy between the missing time series values and the ground truth.

To study whether each module of Score-CDM is useful, we also compare Score-CDM with the following two variants.

- **w/o(S2TWB)**: This variant of Score-CDM removes the S2TWB block.
- **w/o(SCM)**: This variant of Score-CDM removes the score-weighted convolution module SCM.

In the experiment, the baselines Transformer, SAITS, BRITS, GRIN, and SPIN are implemented by the code[1] provided by the work [Marisca *et al.*, 2022].

**Evaluation metrics.** We apply Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) defined as follows to evaluate the model performance.

$$MAE = \frac{1}{T} \sum_{k=0}^{L} \left\| X_k - \widetilde{X}_k \right\|$$

$$RMSE = \sqrt{\frac{1}{T} \sum_{k=0}^{L} (\left\| X_k - \widetilde{X}_k \right\|_F^2}$$

where $X_t$ is the imputed time series at time $t$ and $\widetilde{X}_t$ is the corresponding ground truth.

[1] https://github.com/Graph-Machine-Learning-Group/spin

| Model | AQI-36 | | | | PEMS-BAY | | | | METR-LA | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $p\%$ = 25% | | $p\%$ = 50% | | $p\%$ = 25% | | $p\%$ = 50% | | $p\%$ = 25% | | $p\%$ = 50% | |
| | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE |
| Transformer | 29.46 | 16.26 | 31.49 | 17.45 | 2.98 | 1.63 | 3.22 | 1.74 | 7.01 | 2.82 | 7.16 | 2.89 |
| BRITS | 28.76 | 15.72 | 29.12 | 16.01 | 2.85 | 1.59 | 3.02 | 1.67 | 6.93 | 2.80 | 7.13 | 2.85 |
| SAITS | 29.85 | 16.24 | 30.97 | 17.18 | 2.34 | 1.35 | 2.57 | 1.42 | 6.23 | 2.66 | 6.51 | 2.73 |
| CSDI | 14.52 | 7.71 | 16.93 | 8.87 | 1.49 | 0.76 | 1.77 | 0.85 | 4.05 | 2.27 | 4.38 | 2.32 |
| TimesNet | 15.01 | 12.38 | 17.49 | 13.22 | 1.88 | 0.92 | 2.14 | 0.97 | 5.33 | 2.49 | 5.76 | 2.55 |
| GRIN | 12.93 | 7.93 | 15.81 | 9.02 | 1.70 | 0.85 | 1.92 | 0.91 | 4.21 | 2.30 | 4.47 | 2.35 |
| SPIN | 12.98 | 7.56 | 16.53 | 9.11 | 1.62 | 0.81 | 1.83 | 0.86 | 4.25 | 2.32 | 4.51 | 2.39 |
| PriSTI | 12.57 | 7.05 | 14.68 | 8.25 | 1.32 | 0.71 | 1.54 | 0.78 | 3.84 | 2.03 | 4.16 | 2.09 |
| **Score-CDM** | **12.14** | **6.78** | **14.56** | **7.72** | **1.21** | **0.65** | **1.33** | **0.69** | **3.59** | **1.93** | **3.85** | **2.02** |

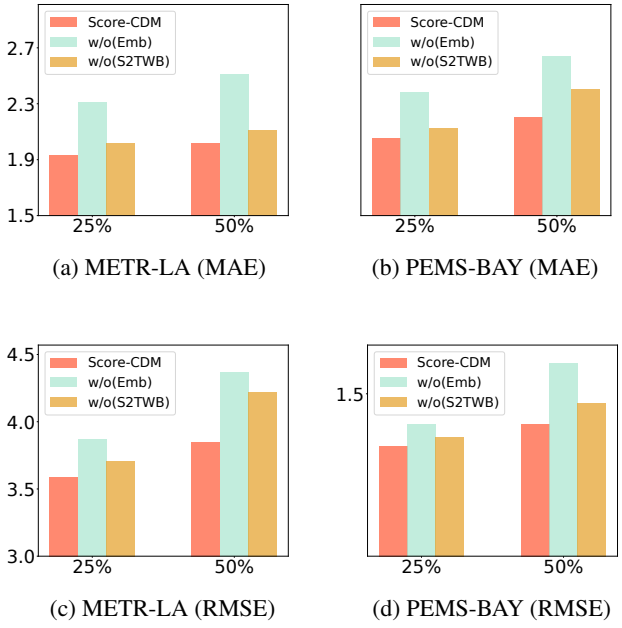
Table 1: Performance comparison of different methods on the three datasets in point missing scenario

| Model | METR | PEMS | AQI-36 |
|---|---|---|---|
| | Sensor failure probability 5% | | |
| BRITS | 5.87 | 4.14 | 24.09 |
| SAITS | 4.73 | 3.88 | 20.78 |
| Transformer | 6.03 | 3.69 | 29.21 |
| GRIN | 3.05 | 2.26 | 15.62 |
| SPIN | 2.74 | 1.78 | 14.29 |
| PriSTI | 2.70 | 1.66 | 14.01 |
| **Score-CDM** | **2.60** | **1.55** | **13.74** |

Table 2: MAE comparison of different methods with a sensor failures probability $q\%$ in block missing scenario

| Model | METR-LA | | PEMS-BAY | |
|---|---|---|---|---|
| | 75 % | 95 % | 75 % | 95 % |
| BRITS | 3.02 | 5.19 | 2.17 | 3.91 |
| SAITS | 3.74 | 6.72 | 2.96 | 7.40 |
| Transformer | 2.71 | 5.13 | 1.13 | 2.70 |
| GRIN | 2.39 | 4.08 | 1.09 | 2.70 |
| SPIN | 2.24 | 2.89 | 1.09 | 2.26 |
| PriSTI | 2.21 | 2.89 | 1.08 | 2.27 |
| **Score-CDM** | **2.14** | **2.86** | **1.06** | **2.23** |

Table 3: MAE comparison of different methods with high data missing percentage (75% and 95%) in the point missing scenario




(a) METR-LA (MAE) (b) PEMS-BAY (MAE)

(c) METR-LA (RMSE) (d) PEMS-BAY (RMSE)

Figure 3: Performance comparison between Score-CDM and two variant models on the point data missing scenarios

## 5.2 Experiment Result

We first compare the performance of different methods in the point missing scenario with the data missing rate $p\%$ setting to $25\%$ and $50\%$, respectively. The experiment result is shown in Table 1. The best result is highlighted in bold font, and the second-best result is underlined. From Table 1, one can observe that Score-CDM consistently outperforms all the baseline methods in both cases and over all the datasets. Specifically, Score-CDM improves the performance of the best baseline PriSTI by 3%-5% in terms of MAE on AQI-36, by 9%-12% on PEMS-BAY dataset, and by 5%-7% on METR-LA dataset. This demonstrates that Score-CDM can effectively balance the local and the global temporal information of time series. Compared with attention-based diffusion models CSDI and PriSTI, Score-CDM performs better in all the datasets, verifying the effectiveness of the extracted score map by SCM and the self-attention kernel of S2TWB to construct the flexible receptive field. Compared with RNN-based methods BRITS, the performance improvement of Score-CDM is much more significant. For example, the RMSE of BRITS on AQI-36 is 28.76 when $p\% = 25\%$, while the RMSE of Score-CDM is dropped to only 12.14. GRIN is also an RNN model, but its performance is much better than BRITS by incorporating GNN models. One can see that SPIN performs best among all the attention-based methods, indicating that integration of both temporal and spatial information can significantly enhance model performance for MTS imputation. However, the performance of SPIN is still inferior to PriSTI, which suggests that diffusion models is truly powerful in MTS imputation due to their strong generative capability. TimesNet's performance is moderate among all the methods. This is because its receptive field is smaller than attention methods, and thus is less effective to capture long-term temporal features in time series data.

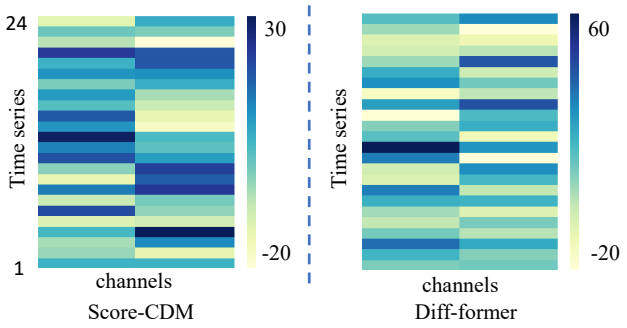



Figure 4: A case study to visualize the score maps of Score-CDM and Diff-former.

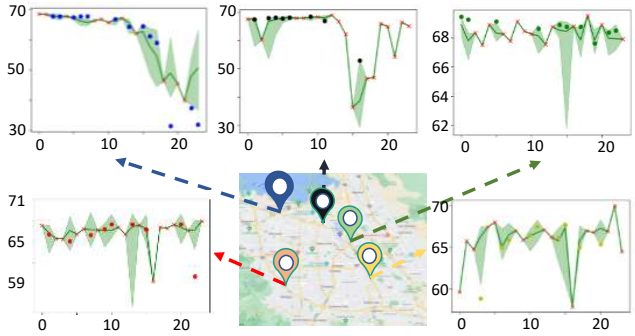


Figure 5: A case study to show the traffic flow time series data imputation results on 5 road sensors in METR-LA dataset.

For the block missing scenario, we set the sensor missing probability $q\% = 5\%$ to mimic that $5\%$ sensors fail in 1 to 4 hours without any time series data observations. We compare Score-CDM with six strong baselines. The result is shown in Table 2. It shows that Score-CDM still outperforms the baseline methods on the three datasets, demonstrating its superior performance in the block data missing scenario. PriSTI achieves the best performance among all the baselines, but it is still inferior to Score-CDM. For example, Score-CDM outperforms PriSTI by more than 3% in terms of MAE on METR-LA & AQI-36, and by more than 12% on PEMS-BAY. To further evaluate the performance of different methods under very high point data missing percentages, we compare Score-CDM against the baselines when $p\% = 75\%$ and $p\% = 95\%$. The result is shown in Table 3. It demonstrates again that Score-CDM outperforms all the baselines when the available time series observations are very sparse.

### 5.3 Ablation Study

To examine whether the designed two modules SCM and S2TWB work, we conduct the ablation study to compare Score-CDM with its two variant models w/o[S2TWB] and w/o[SCM]. Figure 3 shows the result. One can see both modules are useful to the model, as removing each one of them will lead to remarkably performance drop in all four cases. One can also see that SCM has a larger impact on the model performance compared with S2TWB, because removing it leads to a more significant performance decline. This implies that the global temporal features are critical to MTS imputation and the proposed SCM can effectively capture the global features. When 25% point data are missing on the PEMS-BAY dataset, the performance of w/o[S2TWB] drops by over 5% compared with Score-CDM in terms of MAE, and the performance drop is up to 8.7% when the 50% data are missing, which verifies S2TWB is also important to the performance improvement. This indicates a pronounced periodicity of the traffic flow time series in the PEMS-BAY dataset, characterized by a prevalence of local temporal features. The proposed S2TWB in Score-CDM can effectively capture this periodicity by extracting the corresponding frequencies in the spectral domain.

### 5.4 Case Study

To further show the effectiveness of Score-CDM, we give a case study to visualize the learned score map in Figure 4. To make a comparison, we also present the score map learned by Diff-former. Diff-former applies the attention mechanism as the denoising function of the diffusion model to learn the score map and extract time features. We select a traffic flow time series whose length is 24 and with two channels from METR-LA. The darker color in the figure represents a higher score, while the light color represents a smaller score. It shows that Score-CDM can better capture both local and global temporal features compared with Diff-former as the high attention scores are distributed over different locations on the score map, while the high score of Diff-former only located at one or two areas in its score map.

In Figure 5, we give a case study to show the traffic flow time series imputation results of 5 road sensors by Score-CDM in METR-LA dataset. Each subfigure represents the imputation result of a sensor, and the time windows of all sensors are in one day (24 hours). The red crosses represent observations, and dots of various colors represent the ground truth of the missing values. The solid green line is the imputation result by Score-CDM, and the green shadow represents the quantile between 0.05 to 0.95. One can see that the imputed values denoted by the green curves are very close to the ground truth missing time series points and the observations, which demonstrates the desirable time series data imputation performance of Score-CDM.

## 6 Conclusion

This paper proposed a Denoising Diffusion Probabilistic Model based method Score-CDM for multivariate time series imputation. Significantly different from existing methods, Score-CDM employed the specially designed denoising function to adaptively capture and balance the global and local temporal features in time series. The designed denoising function contained two modules, SCM and ARM. SCM generated a score map that contained a global weighting of the entire time series, and ARM module adapted S2TWB to generate a flexible receptive field of the score map. Extensive evaluations over three real-world datasets showed the effectiveness of the proposed model.

# A Complexity Analysis

As shown in Table 4, we compare our method with five transformer models including Transformer, Reformer [Kitaev *et al.*, 2019], Linear Transformer [Katharopoulos *et al.*, 2020], Performer [Choromanski *et al.*, 2020], and Attention Free Transformer in terms of time complexity and space complexity. The comparison shows the efficiency of Score-CDM and its variant.

| Model | Time | Space |
|---|---|---|
| Transformer | $O(T^2d)$ | $O(T^2 + Td)$ |
| Reformer | $O(T \log Td)$ | $O(T \log T + Td)$ |
| Linear Transformer | $O(Td^2)$ | $O(Td + d^2)$ |
| Performer | $O(Td^2 \log d)$ | $O(Td \log d + d^2 \log d)$ |
| AFT | $O(T^2d)$ | $O(Td)$ |
| **Score-CDM** | $O(Td + TlogT)$ | $O(Td)$ |
| (w/o[S2TWB]) | $O(Td)$ | $O(Td)$ |

Table 4: Complexity comparison with different Transformers. $T$ and $d$ denote the sequence length and channel dimension, respectively.

# B Comparison of Different Models

We compare Score-CDM with four classic models including Attention Free Transformer, Transformer, TCN, and Dlinear (MLP-based Model) from four aspects, whether can be coded as attention or convolution, the computational metrics (element-wise or dot-product), and through full or structured receptive field. Through the comparison, one can see that Score-CDM can be considered a convolution model with a global attention score map, and its receptive field is flexible. For a clearer comparison among these methods, one can refer to Table 5.

| Model | Attent | Field | Comput | Conv |
|---|---|---|---|---|
| AFT | ✗ | full | element | ✗ |
| Transformer | ✓ | full | dot | ✗ |
| TCN | ✗ | **local & structured** | element | ✓ |
| DLinear | ✗ | full | dot | ✗ |
| Score-CDM | ✓ | **full & structured** | element | ✓ |

Table 5: Comparison of different models. In this table, we shorthand Attention as Attent, receptive field as Field, Computation as Comput, and Convolution as Conv.

As the table shows, the receptive field of TCN is structured but local. Compared to TCN, the receptive field of Score-CDM is structured and global.

# C Convolution Theorem

To use the Fast Fourier Transform, we need to introduce the convolution theorem first, which is a communication theory as follows,

$$\mathcal{F}(K * X) = \mathcal{F}(K) \cdot \mathcal{F}(X)$$

where $*$ is a convolution operation, $\cdot$ is a multiply operation. $\mathcal{F}$ represents Fast Fourier Transform, which can be used to convolve time series. For time series $X$ and convolution kernel $K$, they can be computed by using FFT which is the same as a time convolution operation.

# D Dataset

For the traffic datasets METR-LA and PEMS-BAY, we artificially inject some missing values by following [Cini *et al.*, 2022] to construct the imcomplete data. We evaluate the model on two data missing scenarios, block missing and point missing. In the block missing scenario, we first randomly mask 5% of the time series data, and then for each sensor we mask its data ranging from 1 to 4 hours with a probability $q\%$ as in [Marisca *et al.*, 2022] to mimic sensor failure. For the point missing case, we randomly mask $p\%$ of all the time series observations.

Additionally, as shown in Table 6, We evaluate the performance of our model on three spatial time series datasets, METR-LA, AQI-36, and PEMS-BAY. METR-LA is a dataset used in traffic flow prediction and imputation. It contains 207 traffic sensor nodes in Los Angeles County Highway with a minute-level sampling rate. AQI-36 is collected from 36 AQI sensors distributed across the city of Beijing. This dataset serves as a widely recognized benchmark for imputation techniques and includes a mask used for evaluation that simulates the distribution of actual missing data [Yi *et al.*, 2016]. For a specific month, such as January, this mask replicates the patterns of missing values from the preceding month. Across all scenarios, the valid observations that have been masked out are employed as targets for evaluation. PEMS-BAY is an open dataset used for traffic flow prediction and analysis, primarily covering the transportation network of the Bay Area in California, USA. The dataset comprises 325 sensor nodes with a sampling interval of 5 minutes, and it contains a total of 16,937,700 data points.

| Dataset | Node | Time step |
|---|---|---|
| METR-LA | 207 | 34272 |
| PEMS-BAY | 325 | 52116 |
| AQI-36 | 36 | 52116 |

Table 6: Comparison of different datasets.

In total, each dataset will be artificially masked 25% or 50% values at random. For the two datasets METR-LA and PEMSBAY, we partition the entire data into training, validation, and testing sets by a ratio of 8 : 1: 1. We evaluate our model performance under two metrics Mean Absolute Error (MAE) and Root Mean Square Error (RMSE).

# E Experiment Settings

For the hyperparameters of Score-CDM, the batch size is 16. The hyperparameter for diffusion model includes a minimum noise level $\beta_1$ and a maximum noise level $\beta_T$. We adopted the quadratic schedule for other noise levels following [Tashiro *et al.*, 2021], which is formalized as:

$$\beta_t = \left( \frac{T-t}{T-1} \sqrt{\beta_1} + \frac{t-1}{T-1} \sqrt{\beta_T} \right)^2. \tag{18}$$

| Description | AQI-36 | METR-LA | PEMS-BAY |
|---|---|---|---|
| Batch size | 16 | 16 | 16 |
| Time length $L$ | 24 | 24 | 24 |
| Epochs | 200 | 200 | 200 |
| Learning rate | 0.001 | 0.001 | 0.001 |
| Channel size $d$ | 64 | 64 | 64 |
| Minimum noise level $\beta_1$ | 0.0001 | 0.0001 | 0.0001 |
| Maximum noise level $\beta_T$ | 0.5 | 0.2 | 0.2 |
| Diffusion steps $T$ | 50 | 50 | 50 |

Table 7: The hyperparameters of Score-CDM for all datasets.

We summarize the hyperparameters of Score-CDM in Table 7.

# F Discussion

As mentioned in Section 4.1, Eq.8 shows the additional part, which we call channel mixing, compared to the attention-free transformer like AFT, which leads to enough message passing through time and channel dimension on element view.

To address the challenge of how to reduce complexity with a comparable result, recent works like RWKV [Peng *et al.*, 2023] have designed an additional module named channel mixing to fill the missing part which is serial to time mixing.

For our work, the S2TWB & ARM can be regarded as a channel-mixing module, which is paralleled with SCM. The whole design ensures a comparable result in evaluation.


## Acknowledgements

This research was funded by the National Science Foundation of China (No. 62172443 and 62206303), Hunan Provincial Natural Science Foundation of China (No. 2022JJ30053) and Science and Technology Innovation Program of Hunan Province(No.2023RC3009).


## Contribution Statement

Shunyang Zhang and Senzhang Wang are the co-first authors who contribute equally. Jian Zhang is the Corresponding author.

## References


[Acevedo and Masuoka, 1997] William Acevedo and Penny Masuoka. Time-series animation techniques for visualizing urban growth. *Computers & Geosciences*, 23(4):423–435, 1997.

[Alcaraz and Strodthoff, 2022] Juan Miguel Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *arXiv preprint arXiv:2208.09399*, 2022.

[Cao *et al.*, 2018] Wei Cao, Dong Wang, Jian Li, Hao Zhou, Lei Li, and Yitan Li. Brits: Bidirectional recurrent imputation for time series. *Advances in neural information processing systems*, 31, 2018.

[Che *et al.*, 2018] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085, 2018.

[Choromanski *et al.*, 2020] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers, 2020.

[Cini *et al.*, 2022] Andrea Cini, Ivan Marisca, and Cesare Alippi. Filling the g_ap_s: Multivariate time series imputation by graph neural networks. In *ICLR*, 2022.

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

[Fang *et al.*, 2021] Ziquan Fang, Lu Pan, Lu Chen, Yuntao Du, and Yunjun Gao. Mdtp: A multi-source deep traffic prediction framework over spatio-temporal trajectory data. *Proceedings of the VLDB Endowment*, 14(8):1289–1297, 2021.

[Gu *et al.*, 2022] Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. On the parameterization and initialization of diagonal state space models. *Advances in Neural Information Processing Systems*, 35:35971–35983, 2022.

[Ho *et al.*, 2020] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020.

[Katharopoulos *et al.*, 2020] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

[Kitaev *et al.*, 2019] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2019.

[Lee and Fambro, 1999] Sangsoo Lee and Daniel B Fambro. Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting. *Transportation research record*, 1678(1):179–188, 1999.

[Li *et al.*, 2022] Jiyue Li, Senzhang Wang, Jiaqiang Zhang, Hao Miao, Junbo Zhang, and S Yu Philip. Fine-grained urban flow inference with incomplete data. *IEEE Transactions on Knowledge and Data Engineering*, 35(6):5851–5864, 2022.

[Lim and Zohren, 2021] Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

[Liu *et al.*, 2022] Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35:5816–5828, 2022.

[Liu *et al.*, 2023] Mingzhe Liu, Han Huang, Hao Feng, Leilei Sun, Bowen Du, and Yanjie Fu. Pristi: A conditional diffusion framework for spatiotemporal imputation.

*2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 1927–1939, 2023.

[Luo *et al.*, 2018] Yonghong Luo, Xiangrui Cai, Ying Zhang, Jun Xu, et al. Multivariate time series imputation with generative adversarial networks. *Advances in neural information processing systems*, 31, 2018.

[Marisca *et al.*, 2022] Ivan Marisca, Andrea Cini, and Cesare Alippi. Learning to reconstruct missing data from spatiotemporal graphs with sparse observations. *Advances in Neural Information Processing Systems*, 35:32069–32082, 2022.

[Miao *et al.*, 2024] Hao Miao, Yan Zhao, Chenjuan Guo, Bin Yang, Zheng Kai, Feiteng Huang, Jiandong Xie, and Christian S Jensen. A unified replay-based continuous learning framework for spatio-temporal prediction on streaming data. In *ICDE*, 2024.

[Nelson, 1998] Brian K Nelson. Time series analysis using autoregressive integrated moving average (arima) models. *Academic emergency medicine*, 5(7):739–744, 1998.

[Peng *et al.*, 2023] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, et al. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*, 2023.

[Peterson, 2009] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[Shen and Kwok, 2023] Lifeng Shen and James Kwok. Non-autoregressive conditional diffusion models for time series prediction. *arXiv preprint arXiv:2306.05043*, 2023.

[Si *et al.*, 2022] Chenyang Si, Weihao Yu, Pan Zhou, Yichen Zhou, Xinchao Wang, and Shuicheng Yan. Inception transformer. *Advances in Neural Information Processing Systems*, 35:23495–23509, 2022.

[Tashiro *et al.*, 2021] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *NeurIPS*, 34:24804–24816, 2021.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

[Wang *et al.*, 2020] Senzhang Wang, Jiannong Cao, and S Yu Philip. Deep learning for spatio-temporal data mining: A survey. *IEEE transactions on knowledge and data engineering*, 34(8):3681–3700, 2020.

[Wang *et al.*, 2021] Qinfen Wang, Siyuan Ren, Yong Xia, and Longbing Cao. Bicmts: Bidirectional coupled multivariate learning of irregular time series with missing values. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3493–3497, 2021.

[Wang *et al.*, 2022] Senzhang Wang, Jiyue Li, Hao Miao, Junbo Zhang, Junxing Zhu, and Jianxin Wang. Generative-free urban flow imputation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2028–2037, 2022.

[Wu *et al.*, 2022] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The Eleventh International Conference on Learning Representations*, 2022.

[Yang *et al.*, 2022] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 2022.

[Yao *et al.*, 2023] Yuanyuan Yao, Dimeng Li, Hailiang Jie, Hailiang Jie, Tianyi Li, Jie Chen, Jiaqi Wang, Feifei Li, and Yunjun Gao. Simplets: An efficient and universal model selection framework for time series forecasting. *Proceedings of the VLDB Endowment*, 16(12):3741–3753, 2023.

[Yi *et al.*, 2016] Xiuwen Yi, Yu Zheng, Junbo Zhang, and Tianrui Li. St-mvl: filling missing values in geo-sensory time series data. In *IJCAI*, 2016.

[Yoon *et al.*, 2017] Jinsung Yoon, William R Zame, and Mihaela van der Schaar. Multi-directional recurrent neural networks: A novel method for estimating missing data. In *Proceedings of Time series workshop in international conference on machine learning*, 2017.

[Zhai *et al.*, 2021] Shuangfei Zhai, Walter Talbott, Nitish Srivastava, Chen Huang, Hanlin Goh, Ruixiang Zhang, and Josh Susskind. An attention free transformer. *arXiv preprint arXiv:2105.14103*, 2021.