MTSCI: A Conditional Diffusion Model for Multivariate Time Series Consistent Imputation

Jianping Zhou
Shanghai Jiao Tong University
Shanghai, China
jianpingzhou@sjtu.edu.cn

Junhao Li
Shanghai Jiao Tong University
Shanghai, China
Lijunhao_hz@sjtu.edu.cn

Guanjie Zheng*
Shanghai Jiao Tong University
Shanghai, China
gjzheng@sjtu.edu.cn

Xinbing Wang
Shanghai Jiao Tong University
Shanghai, China
xwang8@sjtu.edu.cn

Chenghu Zhou
Chinese Academy of Sciences
Beijing, China
zhouch@lreis.ac.cn

ABSTRACT

Missing values are prevalent in multivariate time series, compromising the integrity of analyses and degrading the performance of downstream tasks. Consequently, research has focused on multivariate time series imputation, aiming to accurately impute the missing values based on available observations. A key research question is how to ensure imputation consistency, i.e., *intra-consistency* between observed and imputed values, and *inter-consistency* between adjacent windows after imputation. However, previous methods rely solely on the inductive bias of the imputation targets to guide the learning process, ignoring imputation consistency and ultimately resulting in poor performance. Diffusion models, known for their powerful generative abilities, prefer to generate consistent results based on available observations. Therefore, we propose a conditional diffusion model for **M**ultivariate **T**ime **S**eries **C**onsistent **I**mputation (MTSCI). Specifically, MTSCI employs a contrastive complementary mask to generate dual views during the forward noising process. Then, the intra contrastive loss is calculated to ensure intra-consistency between the imputed and observed values. Meanwhile, MTSCI utilizes a mixup mechanism to incorporate conditional information from adjacent windows during the denoising process, facilitating the inter-consistency between imputed samples. Extensive experiments on multiple real-world datasets demonstrate that our method achieves the state-of-the-art performance on multivariate time series imputation task under different missing scenarios. Code is available at https://github.com/JeremyChou28/MTSCI.

CCS CONCEPTS

• **Information systems → Data mining**; • **Computing methodologies → Artificial intelligence**.

KEYWORDS

Multivariate Time Series Imputation, Intra consistency, Inter consistency

ACM Reference Format:
Jianping Zhou, Junhao Li, Guanjie Zheng, Xinbing Wang, and Chenghu Zhou. 2024. MTSCI: A Conditional Diffusion Model for Multivariate Time Series Consistent Imputation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM '24), October 21–25, 2024, Boise, ID, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3627673.3679532

1 INTRODUCTION

Multivariate time series data widely exists in various real-world applications, e.g., transportation [29, 36], meteorology [1, 17], healthcare [37], energy [2], etc. The integrity of time series plays a crucial role on tasks such as forecasting [26, 48] and classification [22]. However, missing data is a common issue in real-world datasets due to device failures, communication interruptions, and human errors [16, 34], which impairs the downstream task performance and renders the integrity analysis approaches inapplicable. Consequently, multivariate time series imputation, which aims to accurately impute missing values using available observations, arises as an important research question.

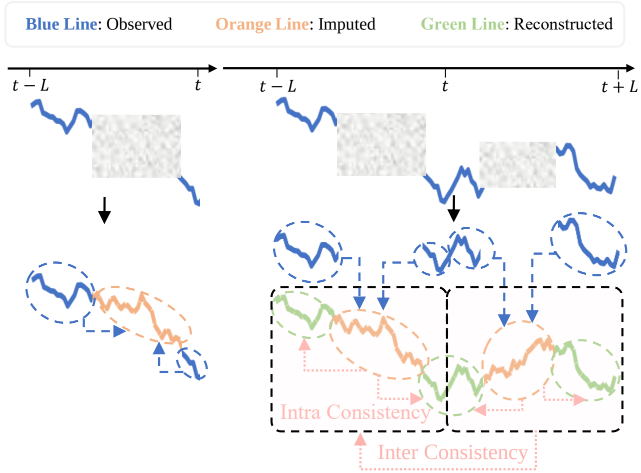
Early studies are based on statistical learning [4, 13] and machine learning methods [20, 40, 46]. Subsequently, numerous deep learning-based methods have been proposed. Some approaches [5, 6, 9, 33] treat the imputation task as a deterministic point estimation problem, while others [3, 8, 12, 35, 45] view it as a probabilistic generative problem. However, these methods rely solely on the inductive bias of artificially simulated imputation targets to guide the learning process, as illustrated in Figure 1 (a), neglecting the crucial aspect of imputation consistency.

The imputation consistency can be divided into two categories: (i)***intra-consistency*** and (ii)***inter-consistency***. As shown in Figure 1 (b), consider two samples of adjacent windows in incomplete time series. *Intra-consistency* implies that the imputed values, guided by observed values, should facilitate the reconstruction of observed values, ensuring consistency between imputed and observed values, thereby reducing imputation bias. *Inter-consistency* means that when imputing a single sample, the sample from the adjacent window should be considered to ensure that the *complete* sample maintains temporal consistency with the adjacent window.

[*] Corresponding author

CIKM '24, October 21–25, 2024, Boise, ID, USA.

ACM ISBN 979-8-4007-0436-9/24/10
https://doi.org/10.1145/3627673.3679532



(a) Previous methods (b) Ours

Figure 1: (a) Previous methods only use the observed values in a single window to impute missing values. (b) Our method uses the adjacent window to assist in imputation to maintain *inter-consistency*, and constraints the imputed targets reciprocally reconstruct the observed values for *intra-consistency*.

This concept coincides with the idea of good continuity in the temporal samples of adjacent windows [37]. However, no existing imputation method addresses the issue of imputation consistency in multivariate time series imputation task.

Nowadays, diffusion models exhibit powerful generative abilities in synthesizing images [31], audio [7], text [14], and time series [35]. Compared to other models, diffusion models are more inclined to generate data consistent with the distribution of observed data, which aligns well with the concept of imputation consistency [18]. Therefore, to address the aforementioned issues, we propose **MTSCI**, a conditional diffusion model for **M**ultivariate **T**ime **S**eries **C**onsistent **I**mputation. MTSCI employs a contrastive complementary mask strategy during the noising process and a mixup mechanism that incorporates conditional information from adjacent windows, effectively guiding the model to maintain consistency and improving imputation performance. Specifically, (i) the complementary mask strategy, which self-supervises the generation of complementary pairs of samples when simulating the imputation targets, imposes contrastive loss to teach the model to maintain *intra-consistency* between the observed and imputed values. (ii) The mixup mechanism incorporates the temporal characteristics of the adjacent window as conditional information to assist in the imputation of current window during the training stage, thus preserving *inter-consistency* between adjacent windows. Experimental results on three real-world datasets demonstrate that our method achieves state-of-the-art performance with average improvements of 17.88% in MAE, 15.09% in RMSE, and 13.64% in MAPE compared to baseline methods.

In summary, the main contributions of our work are as follows:

- Motivated by the neglect of imputation consistency in existing time series imputation methods, we systematically summarize the concept of imputation consistency in multivariate time series imputation task as *intra-consistency* and *inter-consistency*.
- We propose MTSCI, a conditional diffusion model for multivariate time series consistent imputation, incorporating a complementary mask strategy and a mixup mechanism to realize intra-consistency and inter-consistency.
- We conduct extensive experiments on multiple real-world datasets to demonstrate the effectiveness of MTSCI. The results indicate that our method achieves the state-of-the-art performance on multivariate time series imputation task under different missing data scenarios.

2 RELATED WORK

Existing multivariate time series imputation methods can be divided into four categories: *statistical-based methods*, *machine learning-based methods*, *deterministic deep models* and *probabilistic generative models*. (i) *Statistical methods*, such as Mean, Median [13] and KNN [4], utilize the statistical indicators to impute missing values. (ii) *Machine learning methods* like linear imputation, state-space models [10] and MICE [40] impute missing values based on linear dynamics assumptions. Other machine learning methods like low-rank matrix factorization, e.g., NMF [20], TRMF [46], TIDER [25], factorize incomplete data into low-rank matrices and impute missing values using the product of these matrices. However, these methods struggle with capturing the nonlinear dynamics and handling large datasets. (iii) *Deterministic deep models* treat imputation as a deterministic point estimation problem. For instance, GRUD [6] uses the last observation and mean of observations to represent missing patterns. BRITS [5] employs bidirectional recurrent neural networks to capture temporal features. mTAN [33] and SAITS [9] design attention mechanisms to capture temporal dependencies. TimesNet [42] transforms 1D time series to 2D space to capture complex temporal variations, achieving state-of-the-art performance in multiple time series tasks, including imputation. However, deterministic methods fall short of modeling imputation uncertainties. (iv) *Probabilistic generative models* view imputation as a missing values generation problem. For example, GP-VAE [19] combines the VAE [19] and Gaussian process to model incomplete time series. GAIN [45] uses a GAN [15] with a hint mechanism to aid imputation. McFlow [30] leverages normalizing flow generative models and Monte Carlo sampling for imputation. CSDI [35] and MIDM [39] utilize conditional diffusion models to generate missing values by treating the observed values as conditional information. CSBI [8] employs the Schrödinger bridge algorithm for imputation. PriSTI [24] extracts conditional information for spatio-temporal imputation using geographic data. However, these methods only use the inductive bias on imputation targets to guide the learning process, which is insufficient to maintain *intra-consistency* between observed and imputed values within a sequence, as well as *inter-consistency* between imputed sequences.

Although there are no imputation methods considering the imputation consistency, other studies have explored consistency strategies in time series analysis. Time series representation methods [37, 44] propose a consistency strategy in the sampling process, where time segments within adjacent windows exhibit high pattern consistency, such as consistent trends, periods and amplitudes.

TS2Vec [47] and TS-TCC [11] utilize data augmentation to generate multiple views for contrastive learning, thereby learning consistent representations. TF-C [50], CoST [41] consider that sequence dependencies of a time series remain consistent during transformations between the time and frequency domains. However, these methods concentrate on complete time series, falling short in addressing multivariate time series imputation task.

3 PRELIMINARIES

In this section, we define the problem of multivariate time series imputation, and then introduce the background of diffusion models and conditional diffusion models.

3.1 Problem Definition

Definition 1 (**Multivariate Time Series.**). *The multivariate time series denoted as $\mathbf{X} \in \mathbb{R}^{T\times C}$ contains C features with length T. The mask matrix denoted as $\mathbf{M} \in \{0,1\}^{T\times C}$ indicates whether the values is observed or missing, where $\mathbf{M}^{i,j} = 1$ indicates $\mathbf{X}^{i,j}$ is observed, and $\mathbf{M}^{i,j} = 0$ indicates $\mathbf{X}^{i,j}$ is missing. Then, the observed values in $\mathbf{X}$ is denoted as $\mathbf{X}^o = \mathbf{X} \odot \mathbf{M}$, the missing values in $\mathbf{X}$ is denoted as $\mathbf{X}^m = \mathbf{X} \odot (\mathbf{1} - \mathbf{M})$.*

Problem 1 (**Multivariate Time Series Imputation.**). *Given the multivariate time series $\mathbf{X}$ and the mask matrix $\mathbf{M} \in \{0,1\}^{T\times C}$ over T time slices, our task of multivariate time series imputation is to estimate the missing values or corresponding distributions in $\mathbf{X}$. The problem can be formulated as learning a probabilistic imputation function p_θ:*

$$\max_{\theta} p_\theta(\mathbf{X}^m|\mathbf{X}^o), \tag{1}$$

where the goal is to approximate the real the conditional distribution $p_\theta(\mathbf{X}^m|\mathbf{X}^o)$ or minimize the estimation error on missing positions.

3.2 Diffusion Models

A well-known diffusion model is the denoising diffusion probabilistic model (DDPM) [18], which contains the forward noising process and the backward denoising process. During the forward noising process, an input $\mathbf{x}_0$ is gradually corrupted to a Gaussian noise vector, which can be defined by the following Markov chain:

$$q(\mathbf{x}_T|\mathbf{x}_0) := \prod_{k=1}^{T} q(\mathbf{x}_k|\mathbf{x}_{k-1}), q(\mathbf{x}_k|\mathbf{x}_{k-1}) := \mathcal{N}(\sqrt{1-\beta_k}\mathbf{x}_{k-1}, \beta_k\mathbf{I}), \tag{2}$$

where $\beta_k \in [0,1]$ represents the noise level. Then, sampling of $\mathbf{x}_k$ can be written as $q(\mathbf{x}_k|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_k; \sqrt{\bar{\alpha}_k}\mathbf{x}_0, (1-\bar{\alpha}\mathbf{I})$, where $\bar{\alpha}_k = \prod_{s=1}^{k}\alpha_s$, and $\alpha_k = 1-\beta_k$. Thus, $\mathbf{x}_k$ can be simply obtained as:

$$\mathbf{x}_k = \sqrt{\bar{\alpha}_k}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_k}\epsilon, \tag{3}$$

where ϵ is a Gaussian noise. During the backward denoising process, DDPM considers the following specific parameterization of $p_\theta(\mathbf{x}_{k-1}|\mathbf{x}_k)$:

$$p_\theta(\mathbf{x}_{k-1}|\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_{k-1}; \mu_\theta(\mathbf{x}_k, k), \sigma_\theta(\mathbf{x}_k, k)), \tag{4}$$

where the variance $\sigma_\theta(\mathbf{x}_k, k)$ is usually fixed as $\sigma^2\mathbf{I}$ and the mean $\mu_\theta(\mathbf{x}_k, k)$ is defined by a denoising network ($\mathbf{x}_\theta$ or ϵ_θ). For noising estimation, the denoising network ϵ_θ predicts the noise, and then obtains the mean $\mu_\theta(\mathbf{x}_k, k)$:

$$\mu_\theta(\mathbf{x}_k, k) = \frac{1}{\sqrt{\alpha_k}}\mathbf{x}_k - \frac{1-\alpha_k}{\sqrt{1-\bar{\alpha}_k}\sqrt{\alpha_k}}\epsilon_\theta(\mathbf{x}_k, k). \tag{5}$$

The denoising network ϵ_θ is trained by minimizing the loss $\mathcal{L}_\epsilon$:

$$\mathcal{L}_\epsilon = \mathbb{E}_{k,\mathbf{x}_0,\epsilon}\|\epsilon - \epsilon_\theta(\mathbf{x}_k, k)\|_2^2. \tag{6}$$

For $\mathbf{x}_0$ estimation, the denoising network $\mathbf{x}_\theta$ predicts the value $\mathbf{x}_0$, and then obtains the mean $\mu_\theta(\mathbf{x}_k, k)$:

$$\mu_\theta(\mathbf{x}_k, k) = \frac{\sqrt{\alpha}_k(1-\bar{\alpha}_{k-1})}{1-\bar{\alpha}_k}\mathbf{x}_k + \frac{\sqrt{\bar{\alpha}_{k-1}}\beta_k}{1-\bar{\alpha}_k}\mathbf{x}_\theta(\mathbf{x}_k, k). \tag{7}$$

The denoising network $\mathbf{x}_\theta$ is trained by minimizing the loss $\mathcal{L}_\mathbf{x}$:

$$\mathcal{L}_\mathbf{x} = \mathbb{E}_{k,\mathbf{x}_0,\epsilon}\|\mathbf{x}_0 - \mathbf{x}_\theta(\mathbf{x}_k, k)\|_2^2. \tag{8}$$

3.3 Conditional Diffusion Models

The multivariate time series imputation task aims to impute missing values $\mathbf{X}^m \in \mathbb{R}^{T\times C}$ based on the conditional information of observed values $\mathbf{X}^o \in \mathbb{R}^{T\times C}$. The forward noising process and backward denoising process of conditional diffusion model to impute missing values are defined as follows:

$$p_\theta(\mathbf{x}_0^m|\mathbf{x}_0^o) := p(\mathbf{x}_T^m)\prod_{k=1}^{T} p_\theta(\mathbf{x}_{k-1}^m|\mathbf{x}_k^m, \mathbf{c}), \mathbf{x}_T^m \in \mathcal{N}(0, \mathbf{I}), \tag{9}$$

$$p_\theta(\mathbf{x}_{k-1}^m|\mathbf{x}_k^m, \mathbf{x}_0^o) := \mathcal{N}(\mathbf{x}_{k-1}^m; \mu_\theta(\mathbf{x}_k^m, k|\mathbf{c}), \sigma_\theta(\mathbf{x}_k^m, k|\mathbf{c})\mathbf{I}), \tag{10}$$

where $\mathbf{c} = \mathcal{F}(\mathbf{x}_0^o)$ is the conditional information output of the conditioning network $\mathcal{F}$. By repeatedly running the denoising step in (10) till $k = 1$, the imputed value $\hat{\mathbf{x}}_0^m$ is obtained.

4 METHODOLOGY

In this section, we elaborate on our model, MTSCI, which is designed to tackle the multivariate time series imputation task. We start with an overview of MTSCI in Section 4.1. Then, we introduce the contrastive consistency on forward noising process in Section 4.2. Following that, we explain the consistency-assured denoising process in Section 4.3. Finally, we outline the algorithm procedures for both the training and inference stages.

4.1 Overview

MTSCI is a conditional diffusion model for multivariate time series consistent imputation. As shown in Figure 2, the "sampled" window $x_{t-L:t}$ is sampled for imputation. First, we use contrastive complementary mask to generate two views for contrastive consistency on noising process. Then, we utilize the intra contrastive module and inter-consistency condition network for consistency-assured denoising process, where the intra contrastive module is to calculate contrastive loss between two views, ensuring the intra-consistency between imputed and observed values. The inter-consistency condition network is to incorporate the conditional information from "context" window for consistency between adjacent windows.

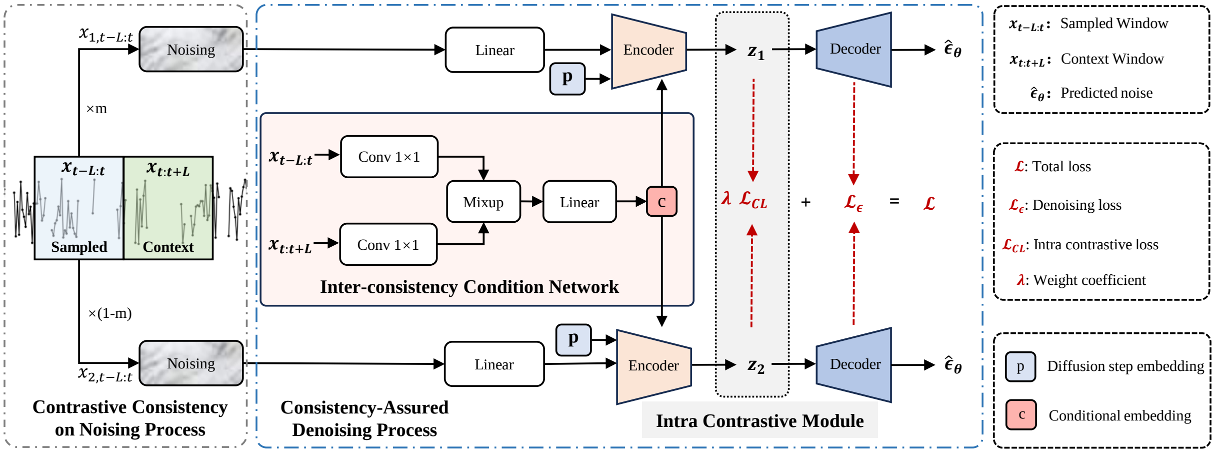

$x_{1,t-L:t}$
Noising
×m
$x_{t-L:t}$
$x_{t:t+L}$
Sampled
Context
×(1-m)
Noising
$x_{2,t-L:t}$
Contrastive Consistency on Noising Process
Linear
p
Encoder
z_1
Decoder
$\hat{\epsilon}_\theta$
$x_{t-L:t}$
Conv 1×1
Mixup
Linear
c
$x_{t:t+L}$
Conv 1×1
Inter-consistency Condition Network
λ $\mathcal{L}_{CL}$
+
$\mathcal{L}_\epsilon$
=
$\mathcal{L}$
Linear
p
Encoder
z_2
Decoder
$\hat{\epsilon}_\theta$
Consistency-Assured Denoising Process
Intra Contrastive Module
$x_{t-L:t}$: Sampled Window
$x_{t:t+L}$: Context Window
$\hat{\epsilon}_\theta$: Predicted noise
$\mathcal{L}$: Total loss
$\mathcal{L}_\epsilon$: Denoising loss
$\mathcal{L}_{CL}$: Intra contrastive loss
λ: Weight coefficient
p Diffusion step embedding
c Conditional embedding

Figure 2: Overview of MTSCI. The intra contrastive module is used to maintain intra-consistency between imputed and observed values. The inter-consistency condition network is utilized to facilitate the inter-consistency between adjacent windows. The model is optimized using a combination of denoising loss $\mathcal{L}_\theta$ and intra contrastive loss $\mathcal{L}_{CL}$.

4.2 Contrastive Consistency on Noising Process

Goal: Previous works directly use a random mask strategy to generate imputation targets and utilize the remaining observations to impute them. However, this approach does not guarantee consistency between the imputed and the observed values, meaning the imputed values cannot guide the network to reconstruct the observed values. This results in overfitting some outliers and then leads to deviations between the imputed values and their ground-truth. To address this, we utilize a contrastive complementary mask strategy for the "sampled" window of the incomplete time series, generating a pair of samples where the imputation targets and observations are complementary. This approach ensures that MTSCI learns to impute missing values consistent with the observed values during training.

4.2.1 Contrastive Complementary Mask. We refer to the input into the MTSCI as the "sampled" window, denoted as $x_{t-L:t}$, and the "context" window is denoted as $x_{t:t+L}$. Using the complementary mask strategy, we generate two views of the "sampled" window. Based on a random mask matrix m from self-supervised mask modeling process, we consider $x_{1,t-L:t} = m \odot x_{t-L:t}$ and $x_{2,t-L:t} = (1-m) \odot x_{t-L:t}$ as the two views, where $\odot$ denotes element-wise production. The selection of mask patterns and mask ratios in the self-supervised mask modeling process is discussed in Section 5.1.

4.2.2 Noising. Take the $x_{1,t-L:t}$ as an example, the imputation target and conditional observation are denoted as $\mathbf{x}_0^{ta} = x_{1,t-L:t}^{ta}$, $\mathbf{x}_0^{co} = x_{1,t-L:t}^{co}$, respectively. Based on (3), we obtain the noised $\mathbf{x}_k^{ta}$:

$$\mathbf{x}_k^{ta} = \sqrt{\bar{\alpha}_k}\mathbf{x}_0^{ta} + \sqrt{1-\bar{\alpha}_k}\epsilon, \qquad (11)$$

where ϵ is sampled from $\mathcal{N}(0, \mathbf{I})$ with the same size as $\mathbf{x}_k^{ta}$.

4.3 Consistency-Assured Denoising Process

Goal: The goal is to introduce how consistency strategies can be ensured in the backward denoising process to enhance imputation performance. We first introduce the intra contrastive loss to constrain the two views generated in Section 4.2, enabling the imputed and observed values to reconstruct each other(***intra consistency***). Subsequently, we utilize the "context" window to provide supplemental conditional information, combined with the observed conditional information in the "sampled" window via a mixup mechanism, to teach the model to impute the "sampled" window while considering the "context" window(***inter-consistency***).

*4.3.1 Intra Contrastive Module(**Intra-consistency**).* The noised contrastive pairs obtained from Section 4.2.2 are input into the denoising network, producing the embedding $z_1$and z_2 after the encoder. Then, we calculate the representation similarity between z_1 and z_2. To be specific, suppose that the number of samples in one batch is N, then after applying the complementary mask, we have $2N$ views of samples. Inspired by the cross-entropy formulation of contrastive loss in previous works [11, 21], we calculate the intra contrastive loss as follows:

$$\mathcal{L}_{CL} = \frac{1}{N}\log\frac{exp(sim(z_1^i, z_2^i)/\tau)}{\sum_{m=1}^{2N}\mathbb{1}_{[m\neq i]}exp(sim(z_1^i, z_2^m)/\tau)}, \qquad (12)$$

where $sim(u,v) = u^T v/\|u\|\|v\|$ is the cosine similarity, $\mathbb{1}_{m\neq i} \in \{0,1\}$ is an indicator function, τ is a temperature parameter. By constraining the representation similarity between two views generated by complementary mask strategy, the model learns to ensure that the imputed values can also reconstruct the observed values during the imputation process.

*4.3.2 Inter-consistency Condition Network(**Inter-consistency**).* Existing methods only use the observed values to impute imputation targets in "sampled" window, ignoring the conditional information of the "context" window actually. To address this problem, we utilize the "context" window to provide supplemental conditional information, teaching the model to impute the "sampled" window while maintaining the contextual consistency on adjacent windows. Although the "context" window is accessible during training, it is not available during inference. Therefore, we incorporate a mixup mechanism [32, 49] to combine the conditional information in the "sampled" window with that in the "context" window as follows:

$$\mathbf{x}_{mix}^{co} = \mathbf{m}^k \odot x_{1,t-L:t}^{co} + (\mathbf{1} - \mathbf{m}^k) \odot \mathcal{F}(x_{2,t:t+L}^{co}), \tag{13}$$

where $\mathbf{x}_{mix}^{co}$represents the mixed conditional information, $\mathbf{m}^k$ is a mixing coefficient matrix sampled from the uniform distribution $\mathcal{U}(0,1)$, and $\mathcal{F}$ is a convolution function with 1×1 kernel size. Notably, we only utilize "context" windows during the training stage, while during the inference stage, we set all elements in $\mathbf{m}^k$ to 1. Then, $\mathbf{x}_{mix}^{co}$ is passed through a linear layer to obtain the hidden representations $\mathbf{c}$:

$$\mathbf{c} = \text{Linear}(\mathbf{x}_{mix}^{co}). \tag{14}$$

Subsequently, the conditional information representation $\mathbf{c}$, which incorporates information from adjacent windows, is used in the encoder of the denoising network to capture dependencies between observed and missing values.

4.3.3 Denoising Network. The denoising network accepts the noised "sampled" window and the conditional information representation $\mathbf{c}$ to predict the noise at missing positions, and then generate the imputed values.

Embedding: First, the noised "sampled" window $\mathbf{x}_k^{ta}$ is embed on a linear layer, plus with the diffusion step representation $\mathbf{p}^k$ to acquire the hidden representation h:

$$h = \text{Linear}(\mathbf{x}_k^{ta}) + \mathbf{p}^k, \tag{15}$$

$$\mathbf{p}^k = \text{FeedForward}(k_{embedding}(k)), \tag{16}$$

$$\begin{aligned} k_{embedding}(k) = [&sin(10^{\frac{0\times4}{w-1}}k), \cdots, sin(10^{\frac{w\times4}{w-1}}k), \\ &cos(10^{\frac{0\times4}{w-1}}k), \cdots, cos(10^{\frac{w\times4}{w-1}}k)], \end{aligned} \tag{17}$$

where FeedForward$(\cdot)$ is a two fully-conncted layers with the *SiLU* activation function, $k_{embedding}$ is d-dimension vectors, $w = \frac{d}{2}$. Then, the h is sent to the encoder to learn the temporal and variable dependencies.

Encoder: First, a linear layer is utilized to fuse the input, and then, we utilize the single-layer vanilla transformer block [38] to capture the temporal dependency. Inspired by the recent work iTransformer [26], we introduce another single-layer invert transformer block to capture the variable dependency. The encoder is stacked by multiple encoder-layers. For a single encoder-layer, the dependency extraction process is defined as follows:

$$\begin{aligned} \mathbf{H} &= \text{Trm}(\text{Linear}(h) + \text{TE}(h)), \\ \mathbf{H}_{inv} &= \text{iTrm}(\text{Transpose}(\mathbf{H}) + \text{FE}(\mathbf{H}))), \end{aligned} \tag{18}$$

where $\mathbf{H}$ is the output of transformer, $\mathbf{H}_{inv}$ is the output of inverted transformer, Trm represents the transformer block, iTrm represents the itransformer block, TE$(\cdot)$ represents the temporal position embedding, FE$(\cdot)$ represents the feature position embedding.

Decoder: The decoder is to merge multiple output from the encoder to acquire $\mathbf{H}_{out}$, and then generate the predicted noise $\hat{\epsilon}$ through a feed-forward network implemented by two fully-connected layers with *ReLU* activation function:

$$\begin{aligned} \mathbf{H}_{out} &= \text{LN}(\text{Concat}(\mathbf{H}_{inv}^1, \cdots, \mathbf{H}_{inv}^L)), \\ \epsilon_\theta(\mathbf{x}_0^{ta}, k|\mathbf{c}) &= \text{FeedForward}(\mathbf{H}_{out}), \end{aligned} \tag{19}$$

$$\hat{\epsilon}_{k-1}^{ta} = \frac{1}{\sqrt{\alpha_k}}\mathbf{x}_k^{ta} - \frac{1-\alpha_k}{\sqrt{1-\bar{\alpha}_k}\sqrt{\alpha_k}}\epsilon_\theta(\mathbf{x}_k^{ta}, k|\mathbf{c}) + \sigma_k\epsilon, \tag{20}$$

where $\mathbf{H}_{inv}^l$ represents the output of l-th layer, L is the number of encoder layers.

4.4 Training

During the training process, for each predicted noise $\hat{\epsilon}$, we calculate the denoising loss $\mathcal{L}_\epsilon$ as follows:

$$\begin{aligned} \mathcal{L}_\epsilon &= \mathbb{E}_{\mathbf{x}_0^{ta}, \epsilon\sim\mathcal{N}(\mathbf{0},\mathbf{I}),k}\mathcal{L}_\epsilon(k), \\ \mathcal{L}_\epsilon(k) &= \|(1-\mathbf{M}) \odot (\epsilon - \epsilon_\theta(\mathbf{x}_k^{ta}, k|\mathbf{c}))\|_2^2. \end{aligned} \tag{21}$$

Then, the total loss $\mathcal{L}$ is obtained by adding the denoising loss $\mathcal{L}_\epsilon$ and the intra contrastive loss $\mathcal{L}_{CL}$ with weighted coefficient λ:

$$\mathcal{L} = \mathcal{L}_\epsilon + \lambda\mathcal{L}_{CL}. \tag{22}$$

The complete training procedure is shown in Algorithm 1.

Algorithm 1: Training process of MTSCI.

```
Input: Incomplete time series $\mathbf{X}$, mask matrix $\mathbf{M}$, the
       number of diffusion steps $K$, the noise levels $\bar{\alpha}_k$.
Output: The optimized denoising network $\epsilon_\theta$
repeat
    sample the "sampled" window $x_{t-L:t}$ and its "context"
      window $x_{t:t+L}$;
    generate two views by complementary mask based on
      random mask matrix $m$: $m \odot x_{t-L:t}$ and
      $(1-m) \odot x_{t-L:t}$ ;
    sample $k \sim$ Uniform(1,$\cdots$,K), $\epsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ ;
    obtain the noised $\mathbf{x}_k^{ta}$ using (11);
    obtain the diffusion step embedding using (16) and (17);
    obtain the mixed conditional information $\mathbf{x}_{mix}^{co}$ by (13);
    use the $\mathbf{x}_{mix}^{co}$ to acquire conditional embedding $\mathbf{c}$;
    use the denoising network to predict the noise $\hat{\epsilon}$ by (19)
      and (20);
    obtain the embedding $z_1, z_2$ of the contrastive pairs after
      the encoder of denoising network;
    calculate the intra contrastive loss $\mathcal{L}_{CL}$ using (12);
    calculate the denoising loss $\mathcal{L}_\epsilon$ using (21);
    calculate the total loss $\mathcal{L}$ using (22);
    Update the gradient $\nabla_\theta\mathcal{L}$;
until converged;
```

4.5 Inference

During the inference process, all elements in the mixing coefficient matrix is set to 1. By repeatedly running the denoising step till k equals 1, we obtain the $\hat{\epsilon}$ as the final predicted noise. The inference procedure is shown in Algorithm 2.

5 EXPERIMENTS

In this section, we first introduce the experimental setups, including the datasets, baselines, evaluation metrics and implementation details. Then, we evaluate our model, MTSCI, with extensive experiments to answer the following research questions:

- **RQ1:** How does MTSCI perform against other baselines in the multivariate time series imputation task?

Algorithm 2: Inference process of MTSCI.

```
Input: A sample of incomplete time series $\mathbf{X}$, mask matrix
       $\mathbf{M}$, the number of diffusion step $K$, the optimized
       denoising network $\epsilon_\theta$.
Output: The imputed values $\hat{\mathbf{x}}_0^{ta}$ of the $\mathbf{X}$
sample the noised $\mathbf{x}_k^{ta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
for $k = K, \cdots, 1$ do
    $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, if $k > 1$, else $\epsilon = 0$;
    obtain the diffusion step embedding using (16) and (17);
    Set all the element of mixing coefficient matrix as 1 and
      then obtain $\mathbf{x}_{mix}^{co}$ by (13);
    use the $\mathbf{x}_{mix}^{co}$ to acquire conditional embedding $\mathbf{c}$;
    predict the $\hat{\mathbf{x}}_{k-1}$ by (19) and (20);
    $\mathbf{x}_{k-1}^{ta} = \hat{\mathbf{x}}_{k-1}^{ta}$;
end
return $\hat{\mathbf{x}}_0^{ta}$;
```

- **RQ2:** How does the imputation consistency of MTSCI contribute to its imputation performance?
- **RQ3:** How does the imputation performance for MTSCI perform about different missing scenarios, including different missing ratios and diverse mask patterns?
- **RQ4:** What is the impact of weighted coefficient λ and major hyperparameters of MTSCI on the imputation performance?

5.1 Experimental Setup

Datasets. We evaluate our proprosed model on three commonly used public datasets: an electricity dataset *ETT* [9], a climate dataset *Weather* [43], and a traffic speed dataset *METR-LA* [23]. The statistical details of these datasets are listed in Table 1.

Table 1: Statistical details of datasets.

Datasets	ETT	Weather	METR-LA
Time span	69,680	52,696	34,272
Interval	15 min	10 min	5 min
Features	7	21	207
Sequence length	24	24	24
Original missing rate	0%	0.017%	8.6%

Baselines. We select 13 baselines to evaluate the performance of our proposed method on multivariate time series imputation task. These baselines include statistical methods (Mean, KNN), typical machine learning methods (MICE, TRMF), deterministic imputation methods (BRITS, mTAN, SAITS, TimesNet, Non-stationary Transformer) and deep generative imputation models (GP-VAE, rGAIN, CSBI, CSDI). We briefly introduce the baseline methods as follows: (1)**Mean**: directly use the average value of observed values to impute. (2)**KNN**: use the average value of similar samples to missing sample, as implemented by *fancyimpute*. (3)**MICE** [40]: multiple imputation method by chained equations. (4)**TRMF** [46]: a temporal regularized matrix factorization method. (5)**GP-VAE** [12]: combine VAE [19] with Guassion process for time series probabilistic imputation. (6)**rGAIN** [45]: a GAN-based method with a bidirectional recurrent encoder-decoder. (7)**BRITS** [5]: use bidirectional RNN for multivariate time series imputation. (8)**mTAN** [33]: use multi-time attention network to impute missing values, which is a transformer-based method. (9)**SAITS** [9]: use joint-optimization training strategy (masked imputation task and observed reconstruction task) to impute missing values, which is also a tranformer-based method. (10)**Non-stationary Transformer** [27]: a tranformer-based method to attenuate time series non-staionary. (11)TimesNet [42]: transform the 1D time series into 2D space and capture the temporal 2D-variations dependencies. Both Non-stationary Transformer and TimesNet are the state-of-the-art multivariate time series imputation methods implemented in TSlib. (12)**CSBI** [8]: use the Schrödinger bridge algorithm to probabilistic time series imputation. (13)**CSDI** [35]: a score-based conditional diffusion model for probabilistic time series imputation method.

Evaluation metrics. Three commonly used metrics in multivariate time series imputation task are used to evaluate the performance of all methods, including the Mean Absolute Error(MAE), Root Mean Squared Error(RMSE), Mean Absolute Percentage Error(MAPE).

Implementation. We divide the training/validation/testing set following the settings of previous works [9, 24, 43]. For ETT, we select the first four-month data(2016/07-2016/10) as the testing set, the following four-month data(2016/11-2017/02) as the validation set, and the left sixteen months(2017/03-2018/06) as the training set. For Weather and METR-LA, we split the training/validation/testing set by 70%/10%/20%. We divide the samples by a window size of 24 steps without overlapping. We consider two different missing patterns to artificially simulate the missing values for evaluation: (1)**Point missing:** we randomly mask 20% data points in the datasets. (2)**Block missing:** based on randomly masking 5% of the observed data, mask observations ranging from [L/2, 2L] (L is the window size) with 0.15% probability. For training strategies, we utilize point and block strategies for self-supervised learning. Specifically, for point strategy, we randomly choose r ($r \in [0\%, 100\%]$) of observed values as imputation targets. For block strategy, we randomly choose a sequence with a length in the range [L/2,L] with a probability r ($r \in [0\%, 15\%]$) as imputation targets.

5.2 Overall Performance(RQ1)

The overall performance is shown in Table 2 and Table 3. We make the following observations: (1) MTSCI achieves the state-of-the-art performance across multiple datasets, both in point missing and block missing patterns, with an average improvement of 17.88% in MAE, 15.09% in RMSE and 13.64% in MAPE, respectively. Notably, in the case of block missing pattern with continuous missing scenarios, our method demonstrates greater superiority, highlighting the effectiveness of the imputation consistency strategy adopted in MTSCI. (2) The statistical methods and classical machine learning methods perform poor on all datasets due to the strong non-linearity of incomplete multivariate time series. These methods impute missing values based on assumptions such as stability or linear dynamics, which fail to capture the complex temporal correlations in real-world datasets. (3) Compared to deterministic imputation models, MTSCI achieves performance improvements of 42.07% in MAE, 24.15% in RMSE and 39.76% in MAPE respectively on average. The difference between these methods and our model is that we employ the conditional diffusion process to model the incomplete time

Table 2: Overall performance on three datasets with point missing pattern. Performance averaged over 5 runs.

Methods	ETT			Weather			METR-LA		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Mean	3.676 ± 0.000	6.557 ± 0.000	46.451 ± 0.000	38.331 ± 0.000	89.201 ± 0.000	22.726 ± 0.000	7.136 ± 0.000	11.272 ± 0.000	12.458 ± 0.000
KNN	3.561 ± 0.000	6.708 ± 0.000	44.994 ± 0.000	5.002 ± 0.000	25.454 ± 0.000	2.965 ± 0.000	8.365 ± 0.000	14.473 ± 0.000	14.592 ± 0.000
MICE	2.324 ± 0.000	5.586 ± 0.000	29.371 ± 0.000	12.834 ± 0.000	46.177 ± 0.000	7.609 ± 0.000	3.301 ± 0.000	5.125 ± 0.000	5.763 ± 0.000
TRMF	2.023 ± 0.000	3.213 ± 0.000	25.646 ± 0.004	30.106 ± 0.176	75.557 ± 0.412	17.871 ± 0.105	5.908 ± 0.002	8.416 ± 0.003	10.228 ± 0.003
GP-VAE	1.446 ± 0.084	2.474 ± 0.113	17.376 ± 0.982	9.216 ± 0.602	27.112 ± 1.032	5.470 ± 0.357	3.305 ± 0.040	5.312 ± 0.020	5.788 ± 0.071
rGAIN	0.804 ± 0.032	1.743 ± 0.071	10.167 ± 0.399	7.597 ± 0.129	25.139 ± 0.230	4.505 ± 0.077	3.115 ± 0.010	4.810 ± 0.014	5.439 ± 0.017
BRITS	0.634 ± 0.041	1.491 ± 0.115	8.006 ± 0.516	8.487 ± 0.317	25.844 ± 0.842	5.032 ± 0.188	2.981 ± 0.009	4.672 ± 0.011	5.205 ± 0.015
mTAN	0.525 ± 0.114	0.984 ± 0.185	6.654 ± 1.444	5.872 ± 0.215	21.498 ± 0.271	3.483 ± 0.128	3.611 ± 0.091	5.681 ± 0.203	6.306 ± 0.160
SAITS	0.460 ± 0.049	0.832 ± 0.121	5.812 ± 0.615	4.113 ± 0.108	19.207 ± 0.171	2.438 ± 0.064	2.543 ± 0.012	4.381 ± 0.008	4.441 ± 0.021
TimesNet	0.317 ± 0.002	0.511 ± 0.005	4.007 ± 0.027	4.114 ± 0.121	20.916 ± 0.242	2.439 ± 0.071	2.474 ± 0.004	4.346 ± 0.006	4.321 ± 0.006
Stationary	0.297 ± 0.003	0.481 ± 0.006	3.754 ± 0.035	4.415 ± 0.748	21.733 ± 2.763	2.618 ± 0.443	2.502 ± 0.006	4.402 ± 0.009	4.369 ± 0.010
CSBI	0.270 ± 0.014	0.467 ± 0.035	3.262 ± 0.164	3.506 ± 0.277	23.108 ± 2.354	1.950 ± 0.154	2.683 ± 0.010	4.702 ± 0.052	4.751 ± 0.017
CSDI	0.225 ± 0.000	0.383 ± 0.001	2.838 ± 0.004	2.084 ± 0.007	17.003 ± 0.135	1.236 ± 0.004	1.733 ± 0.000	3.248 ± 0.001	3.026 ± 0.000
MTSCI(ours)	**0.214 ± 0.001**	**0.358 ± 0.003**	**2.711 ± 0.020**	**1.955 ± 0.011**	**16.162 ± 0.186**	**1.160 ± 0.006**	**1.655 ± 0.011**	**3.076 ± 0.026**	**2.889 ± 0.019**

Table 3: Overall performance on three datasets with block missing pattern. Performance averaged over 5 runs.

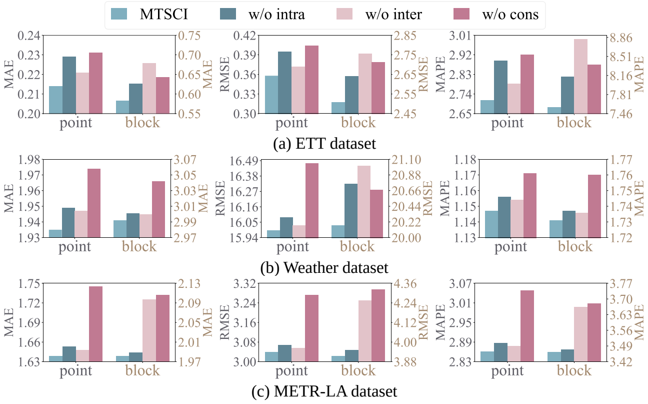
Methods	ETT			Weather			METR-LA		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Mean	3.407 ± 0.000	6.248 ± 0.000	46.235 ± 0.000	41.383 ± 0.000	94.105 ± 0.000	23.057 ± 0.000	7.120 ± 0.000	11.229 ± 0.000	12.437 ± 0.000
KNN	2.701 ± 0.000	6.130 ± 0.000	36.652 ± 0.000	21.245 ± 0.000	71.288 ± 0.000	11.837 ± 0.000	9.012 ± 0.000	14.912 ± 0.000	15.721 ± 0.000
MICE	1.973 ± 0.000	5.212 ± 0.000	26.783 ± 0.000	10.023 ± 0.000	34.550 ± 0.000	5.584 ± 0.000	8.668 ± 0.000	11.697 ± 0.000	15.143 ± 0.000
TRMF	1.863 ± 0.012	3.156 ± 0.018	25.329 ± 0.157	31.619 ± 0.136	77.851 ± 0.350	17.620 ± 0.076	5.781 ± 0.001	8.299 ± 0.000	10.009 ± 0.001
GP-VAE	2.365 ± 0.100	4.344 ± 0.100	29.726 ± 1.124	14.579 ± 0.034	38.357 ± 0.344	8.493 ± 0.084	3.393 ± 0.008	5.478 ± 0.013	5.942 ± 0.013
rGAIN	1.964 ± 0.099	4.837 ± 0.159	26.652 ± 1.347	11.359 ± 0.118	33.929 ± 0.301	6.329 ± 0.065	3.604 ± 0.014	5.592 ± 0.020	6.297 ± 0.024
BRITS	1.676 ± 0.055	4.790 ± 0.124	22.744 ± 0.745	13.239 ± 0.949	37.372 ± 2.234	7.376 ± 0.529	3.352 ± 0.014	5.372 ± 0.025	5.856 ± 0.025
mTAN	1.284 ± 0.070	3.558 ± 0.035	17.478 ± 0.949	11.411 ± 0.528	38.816 ± 1.621	6.351 ± 0.294	3.809 ± 0.060	6.605 ± 0.141	6.656 ± 0.106
SAITS	1.475 ± 0.060	4.232 ± 0.117	20.011 ± 0.807	7.286 ± 0.169	28.784 ± 0.157	4.059 ± 0.094	2.928 ± 0.012	5.163 ± 0.018	5.115 ± 0.021
TimesNet	1.553 ± 0.005	3.940 ± 0.004	21.072 ± 0.065	21.915 ± 0.311	71.433 ± 0.619	12.210 ± 0.173	5.140 ± 0.007	8.629 ± 0.011	8.980 ± 0.012
Stationary	1.525 ± 0.005	3.926 ± 0.004	20.692 ± 0.064	21.808 ± 0.083	71.322 ± 0.207	12.151 ± 0.046	5.139 ± 0.004	8.626 ± 0.010	8.978 ± 0.007
CSBI	1.261 ± 0.064	4.365 ± 0.230	14.604 ± 0.703	18.653 ± 1.082	63.743 ± 4.381	13.440 ± 0.780	5.574 ± 0.248	9.665 ± 0.240	9.904 ± 0.440
CSDI	0.967 ± 0.004	3.580 ± 0.028	13.120 ± 0.058	4.648 ± 0.021	26.598 ± 0.053	2.590 ± 0.011	2.582 ± 0.002	5.534 ± 0.006	4.511 ± 0.003
MTSCI(ours)	**0.642 ± 0.032**	**2.706 ± 0.101**	**8.350 ± 0.418**	**3.092 ± 0.035**	**21.267 ± 0.159**	**2.407 ± 0.063**	**1.982 ± 0.016**	**3.914 ± 0.027**	**3.462 ± 0.028**

series imputation task, which refines the imputed values through multiple steps instead of non-autoregressive single step imputation as deterministic methods do. (4) Compared with deep generative imputation methods, MTSCI consistently outperforms on several datasets. This indicates that our imputation consistency strategy effectively enhances imputation performance. Previous generative models, including the conditional diffusion models like CSBI and CSDI, still exhibit significant errors because they rely solely on self-supervised masking strategy to generate imputation targets and directly guide the denoising network through the inductive bias at the imputation targets. In contrast, MTSCI utilizes a complementary mask strategy to generate dual views for intra contrastive loss and a mixup mechanism to combine conditional information from adjacent windows, facilitating more accurate and consistent imputation performance.

5.3 Ablation Study(RQ2)

We conduct an ablation study to evaluate the effectiveness of the complementary mask strategy to generate intra contrastive loss and the inter-consistency condition network with mixup mechanism for utilizing conditional information from adjacent windows. We compare three variants of MTSCI with and without these components: (1) We remove the complementary mask strategy on the forward noising process along with the intra-consistency loss. This variant is denoted as *w/o intra.* (2) We remove the mixup mechanism in inter-consistency condition network, which is denoted as *w/o inter.* (3) We use only the conditional information of observed values from single window and the denoising network of MTSCI, without the complementary mask strategy and mixup mechanism. This variant is denoted as *w/o cons.*

Figure 3 shows the performance comparison of these three variants and MTSCI. First, we observe that without the complementary mask strategy, the performance of this variant deteriorates. This indicates that generating contrastive views to facilitate the mutual reconstruction of observed and imputed values, improves the imputation performance. Second, using adjacent windows to provide supplemental conditional information also enhances the imputation performance. This suggests that the adjacent windows can bring contextual consistency constraints to the missing values of the current window, alleviating the estimation error between the imputation results and the ground-truth. Finally, the collaboration



(a) ETT dataset

(b) Weather dataset

(c) METR-LA dataset

Figure 3: Ablation study of MTSCI.

Table 4: RMSEs of two different denoising objectives: Predicting noise ϵ_θ vs Predicting target x_θ.

Methods	ETT		Weather		METR-LA	
	Point	Block	Point	Block	Point	Block
Predicting ϵ_θ	**0.358**	**2.706**	**16.162**	**21.267**	**3.076**	**3.914**
Predicting x_θ	0.775	2.836	22.739	30.847	4.111	5.523

of these modules jointly improves the imputation performance, further confirming the necessity of utilizing both simultaneously.

We also conduct experiments to compare the performance using the same denoising network architecture but with different objectives: predict noise ϵ or predict missing values $\mathbf{x}_0^{ta}$. As shown in Table 4, performance is better when the denoising network's objective is to predict noise. This is likely because noise follows a Gaussian distribution, which aids the complementary mask views in predicting noise and reconstructing each other, thereby maintaining the intra-consistency. However, the distributions of observed values and missing values may not be the same.

5.4 Case Study(RQ2)

In order to intuitively understand how MTSCI imputes the incomplete time series, we visualize the imputation results of MTSCI and the sub-optimal method CSDI. Specifically, we randomly select three snapshots of incomplete time series with block missing pattern from three datasets. As shown in Figure 4, MTSCI demonstrates better imputation performance compared to the CSDI, which is attributed to intra contrastive consistency and inter-consistency condition network. In addition, compared to three variants of MTSCI, our model exhibits consistent trend between imputed and observed values, alleviating the imputation error. This improvement is due to MTSCI's consideration of intra-consistency within a single window that observed and imputed values can reconstruct each other, as well as the inter-consistency between adjacent windows that can provide supplemental condition information to guide imputation. Additionally, we use the CRPS metric [28, 35] to measure the imputation consistency between the imputed results and the observed values at the whole dataset. As shown in Table 5, our method outperforms the sub-optimal method CSDI.

Table 5: The imputation consistency, evaluated by CRPS [28], are presented for both MTSCI and CSDI. Lower values indicate better consistency performance.

Methods	ETT		Weather		METR-LA	
	Point	Block	Point	Block	Point	Block
MTSCI	**0.0206**	**0.0626**	**0.0085**	**0.0137**	**0.0220**	**0.0265**
CSDI	0.0220	0.0652	0.0095	0.0164	0.0233	0.0383

Table 6: Imputation performance comparison w.r.t different missing ratios on Weather dataset.

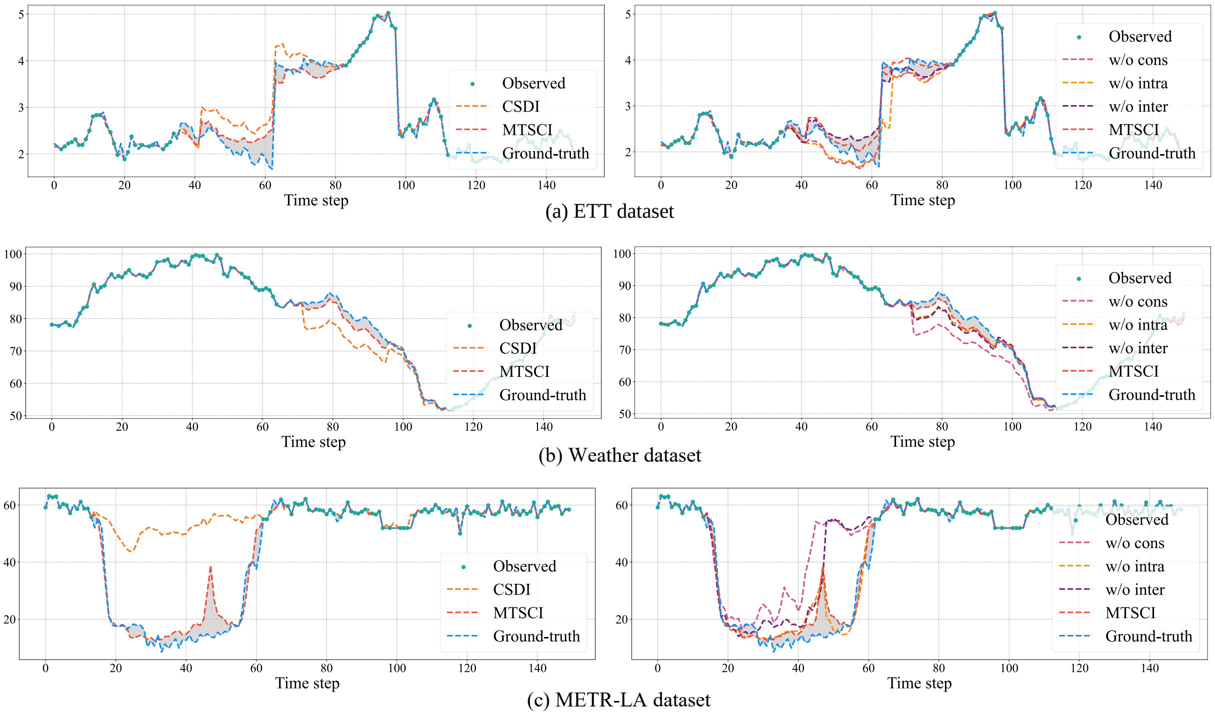
Methods		**MTSCI**			CSDI			CSBI		
Miss Ratio		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Point	10%	**1.995**	16.831	**1.186**	2.123	**15.524**	1.265	4.253	30.421	2.423
	30%	**1.962**	15.711	**1.157**	2.077	**14.952**	1.225	3.875	26.488	2.185
	50%	**2.474**	18.417	**1.464**	2.569	**16.923**	1.520	4.714	31.090	2.710
	70%	**3.546**	**21.286**	**2.109**	4.460	24.214	2.647	5.873	35.816	3.375
Block	10%	**3.580**	**23.496**	**2.041**	4.314	24.513	2.459	17.875	63.623	11.649
	30%	**2.714**	**19.280**	**1.579**	3.021	20.064	1.757	10.783	47.616	6.394
	50%	**2.877**	**20.304**	**1.694**	2.951	20.458	1.738	9.377	43.415	5.518
	70%	**3.495**	**22.287**	**2.073**	3.579	23.260	2.232	9.598	43.918	5.574

Table 7: Performance comparison of different missing patterns during training and testing.

Settings	Methods	ETT			Weather			METR-LA		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Point	MTSCI	0.707	**2.857**	9.191	**6.422**	**26.739**	**3.716**	**3.478**	6.485	**6.075**
↓	CSDI	0.936	3.573	12.700	14.340	67.415	8.297	3.485	**6.457**	6.087
Block	CSBI	**0.705**	2.881	**8.385**	11.566	53.313	6.968	3.928	7.962	6.967
Block	MTSCI	0.345	0.679	4.360	**1.899**	**15.885**	**1.126**	**1.783**	**3.363**	**3.113**
↓	CSDI	**0.229**	**0.385**	**2.893**	2.209	17.916	1.310	1.988	3.736	3.472
Point	CSBI	0.401	0.735	4.850	4.616	26.578	2.567	2.790	4.766	4.941

5.5 Sensitivity Analysis(RQ3)

To evaluate the generalization ability of MTSCI, we carry out an assessment of performance w.r.t. different missing ratios on Weather dataset (due to space limitations). To comprehensively account for the sufficiency and sparsity of missing data, we use missing ratios of 10%,30%,50%,70% in testing set. We compare MTSCI with two other conditional diffusion models: CSBI and CSDI. The results are shown in Table 6. MTSCI outperforms the baselines across different missing ratios both on two different mask patterns. However, we observe that the imputation performance does not consistently decrease with increasing missing ratios, indicating the presence of distribution shift in incomplete time series. Notably, our consistency strategy helps alleviate this issue to some extent, maintaining good generalization ability. To further verify the generalization of our model, we also evaluate it under different missing patterns for the training and testing sets. Specifically, we use two settings: **Point->Block** (Point missing pattern in training set, Block missing pattern in testing set.) and **Block->Point** (Block missing pattern in training set, Point missing pattern in testing set.) As shown in Table 7, our method achieves relatively better performance even when the missing patterns in the training and testing sets differ. This indicates that our trained model can handle imputation tasks in testing environments with missing patterns that are different from those in the training environment.

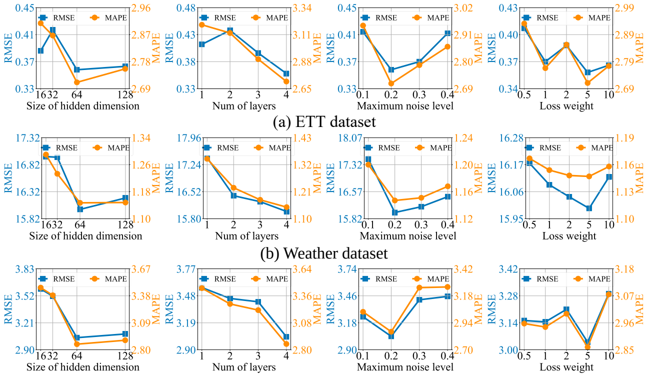


(a) ETT dataset

(b) Weather dataset

(c) METR-LA dataset

Figure 4: The visualization of the imputation results of snapshots on three datasets: MTSCI vs CSDI and MTSCI vs three variants.



(a) ETT dataset

(b) Weather dataset

(c) METR-LA dataset

Figure 5: Hyperparameter study on four key parameters of MTSCI: the size of hidden dimension d_h, the number of encoder-layer l in denoising network, the maximum noise level β_K and the weighted coefficient λ of intra-consistency loss. Both RMSE and MAPE are reported.

5.6 Hyperparameter Study(RQ4)

We conduct a hyperparameter study on key parameters in MTSCI to select the optimal settings across three datasets: the size of hidden dimension d, the number of encoder layers l in the denoising network, the maximum noise level β_T and the weighted coefficient λ of the intra contrastive loss. The results are shown in Figure 5. d affects the representation ability of the model, leading to poor performance if it is too large or too small. While a large l enhances the ability to capture temporal and variable dependencies, thereby improving imputation performance, it also increases the amount of model parameters and computation complexity. For the level of sampled noise, a moderate value is more conducive to the noise prediction in the denoising network. To balance the learning objectives of MTSCI during training, we need to adjust the weight coefficient of the contrastive loss on a small scale for better performance.

6 CONCLUSION

In this paper, we systematically summarize the imputation consistency for improving imputation performance, including the *intra-consistency* and *inter-consistency*. We propose a conditional diffusion model for multivariate time series consistent imputation, **MTSCI**. Specifically, we adopt a complementary mask strategy to introduce intra contrastive loss, ensuring the mutual consistency between the imputed and observed values. Moreover, we utilize an inter-consistency condition network with a mixup mechanism to incorporate the conditional information from adjacent windows to facilitate imputation. Extensive experiments demonstrate the effectiveness of our method, achieving state-of-the-art performance on multiple real-world datasets under various experimental settings. In the future, we will extend our method to apply to more complex missing data scenarios.

ACKNOWLEDGMENTS

We thank the kind help from Pan Liu (SJTU). This work was sponsored by National Key Research and Development Program of China under Grant No.2022YFB3904204, National Natural Science Foundation of China under Grant No. 62102246, 62272301, and Provincial Key Research and Development Program of Zhejiang under Grant No. 2021C01034.

REFERENCES

[1] Kumar Abhishek, Maheshwari Prasad Singh, Saswata Ghosh, and Abhishek Anand. 2012. Weather forecasting model using artificial neural network. *Procedia Technology* 4 (2012), 311–318.

[2] Diyar Akay and Mehmet Atak. 2007. Grey prediction with rolling mechanism for electricity demand forecasting of Turkey. *energy* 32, 9 (2007), 1670–1675.

[3] Juan Lopez Alcaraz and Nils Strodthoff. 2022. Diffusion-based Time Series Imputation and Forecasting with Structured State Space Models. *Transactions on Machine Learning Research* (2022).

[4] Gustavo EAPA Batista, Maria Carolina Monard, et al. 2002. A study of K-nearest neighbour as an imputation method. *His* 87, 251-260 (2002), 48.

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

[6] 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.

[7] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. 2020. Wavegrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713* (2020).

[8] Yu Chen, Wei Deng, Shikai Fang, Fengpei Li, Nicole Tianjiao Yang, Yikai Zhang, Kashif Rasul, Shandian Zhe, Anderson Schneider, and Yuriy Nevmyvaka. 2023. Provably convergent Schrödinger bridge with applications to probabilistic time series imputation. In *International Conference on Machine Learning*. PMLR, 4485–4513.

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

[10] James Durbin and Siem Jan Koopman. 2012. *Time series analysis by state space methods*. Vol. 38. OUP Oxford.

[11] Emadeldeen Eldele, Mohamed Ragab, Zhenghua Chen, Min Wu, Chee Keong Kwoh, Xiaoli Li, and Cuntai Guan. 2022. Self-supervised Contrastive Representation Learning for Semi-supervised Time-Series Classification. *arXiv preprint arXiv:2208.06616* (2022).

[12] Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. 2020. Gp-vae: Deep probabilistic time series imputation. In *International conference on artificial intelligence and statistics*. PMLR, 1651–1661.

[13] David S Fung. 2006. Methods for the estimation of missing values in time series. (2006).

[14] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. 2022. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933* (2022).

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

[16] Sergio Greco, Cristian Molinaro, and Francesca Spezzano. 2012. *Incomplete data and data dependencies in relational databases*. Vol. 29. Morgan & Claypool Publishers.

[17] Jindong Han, Hao Liu, Hengshu Zhu, Hui Xiong, and Dejing Dou. 2021. Joint air quality and weather prediction based on multi-adversarial spatiotemporal networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4081–4089.

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

[19] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[20] Daniel Lee and H Sebastian Seung. 2000. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems* 13 (2000).

[21] Seunghan Lee, Taeyoung Park, and Kibok Lee. 2023. Learning to Embed Time Series Patches Independently. In *The Twelfth International Conference on Learning Representations*.

[22] Guozhong Li, Byron Choi, Jianliang Xu, Sourav S Bhowmick, Daphne Ngar-yin Mah, and Grace LH Wong. 2022. IPS: instance profile for shapelet discovery for time series classification. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 1781–1793.

[23] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *International Conference on Learning Representations*.

[24] Mingzhe Liu, Han Huang, Hao Feng, Leilei Sun, Bowen Du, and Yanjie Fu. 2023. Pristi: A conditional diffusion framework for spatiotemporal imputation. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 1927–1939.

[25] Shuai Liu, Xiucheng Li, Gao Cong, Yile Chen, and Yue Jiang. 2022. Multivariate time-series imputation with disentangled temporal representations. In *The Eleventh International Conference on Learning Representations*.

[26] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. 2023. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625* (2023).

[27] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2022. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems* 35 (2022), 9881–9893.

[28] James E Matheson and Robert L Winkler. 1976. Scoring rules for continuous probability distributions. *Management science* 22, 10 (1976), 1087–1096.

[29] Jingci Ming, Le Zhang, Wei Fan, Weijia Zhang, Yu Mei, Weicen Ling, and Hui Xiong. 2022. Multi-graph convolutional recurrent network for fine-grained lane-level traffic flow imputation. In *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE, 348–357.

[30] Trevor W Richardson, Wencheng Wu, Lei Lin, Beilei Xu, and Edgar A Bernal. 2020. Mcflow: Monte carlo flow models for data imputation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14205–14214.

[31] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.

[32] Lifeng Shen and James Kwok. 2023. Non-autoregressive conditional diffusion models for time series prediction. In *International Conference on Machine Learning*. PMLR, 31016–31029.

[33] Satya Narayan Shukla and Benjamin Marlin. 2020. Multi-Time Attention Networks for Irregularly Sampled Time Series. In *International Conference on Learning Representations*.

[34] Qiaoyu Tan, Guoxian Yu, Carlotta Domeniconi, Jun Wang, and Zili Zhang. 2018. Incomplete multi-view weak-label learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2703–2709.

[35] 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.

[36] David Alexander Tedjopurnomo, Zhifeng Bao, Baihua Zheng, and Farhana Murtaza Choudhury. 2020. A survey on modern deep neural network for traffic prediction: Trends, methods and challenges. *IEEE Transactions on Knowledge and Data Engineering* 34, 4 (2020), 1544–1561.

[37] Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. 2020. Unsupervised Representation Learning for Time Series with Temporal Neighborhood Coding. In *International Conference on Learning Representations*.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[39] Xu Wang, Hongbo Zhang, Pengkun Wang, Yudong Zhang, Binwu Wang, Zhengyang Zhou, and Yang Wang. 2023. An observed value consistent diffusion model for imputing missing values in multivariate time series. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2409–2418.

[40] Ian R White, Patrick Royston, and Angela M Wood. 2011. Multiple imputation using chained equations: issues and guidance for practice. *Statistics in medicine* 30, 4 (2011), 377–399.

[41] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. 2021. CoST: Contrastive Learning of Disentangled Seasonal-Trend Representations for Time Series Forecasting. In *International Conference on Learning Representations*.

[42] 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 *International Conference on Learning Representations*.

[43] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems* 34 (2021), 22419–22430.

[44] Hugo Yèche, Gideon Dresdner, Francesco Locatello, Matthias Hüser, and Gunnar Rätsch. 2021. Neighborhood contrastive learning applied to online patient monitoring. In *International Conference on Machine Learning*. PMLR, 11964–11974.

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

[46] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. 2016. Temporal regularized matrix factorization for high-dimensional time series prediction. *Advances in neural information processing systems* 29 (2016).

[47] 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.

[48] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting?. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 11121–11128.

[49] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2018. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*.

[50] Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. 2022. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in Neural Information Processing Systems* 35 (2022), 3988–4003.