Scope-Free Global Multi-Condition-Aware Industrial Missing Data Imputation Framework via Diffusion Transformer

Diju Liu, Yalin Wang, *Senior Member, IEEE*, Chenliang Liu, Xiaofeng Yuan, *Member, IEEE*, Kai Wang, *Member, IEEE*, and Chunhua Yang, *Fellow, IEEE*

***Abstract*—Missing data is a common phenomenon in the industrial field. The recovery of missing data is crucial to enhance the reliability of subsequent data-driven monitoring and control of industrial processes. Most existing methods are limited by the confined scope of feature extraction, which makes it impossible to rely on global information to impute missing data. In addition, they usually assume that industrial data is a uniform distribution across all working conditions, ignoring the differences in data evolution patterns across different conditions. To address these issues, this paper proposes an innovative scope-free global multi-condition-aware imputation framework based on diffusion transformer (SGMCAI-DiT). First, it extends the diffusion model by introducing conditional probability to capture the condition distribution of the entire data. Then, a noise prediction model is designed based on a novel double-weighted attention mechanism (DW-SA) to broaden the horizons of feature extraction. By discerning the inter-conditional interactions and the intra-conditional local information, the missing data imputation performance can be improved. Finally, the effectiveness and suitability of the proposed SGMCAI-DiT are verified on four real datasets sourced from industrial processes and two public non-industrial datasets. Extensive experimental results demonstrate that the proposed method outperforms several state-of-the-art methods in different missing data scenarios.**

***Index Terms*—Diffusion model, missing data imputation, scope-free global multi-condition-aware imputation framework, transformer.**

I. Introduction

THE occurrence of missing data is a fundamental problem in the industrial manufacturing field that cannot be ignored [1], [2]. The existence of these low-quality datasets severely hampers the successful development and application of subsequent data-driven process monitoring and decision-making [3], [4]. Therefore, the imputation of missing data has become a fundamental issue that needs to be urgently solved in the industrial field. Usually, the foremost challenge in industrial time series missing data imputation tasks arises from its four noteworthy attributes: high-dimensional nonlinear coupling characteristics, low-quality and noisy characteristics, multi-condition characteristics, and non-uniform sampling characteristics. The first two characteristics are caused by the complexity of industrial systems and the harshness of the detection environment, and they are routine issues and there are already many solutions [5]. Hence, this paper is concentrating more on the latter two characteristics.

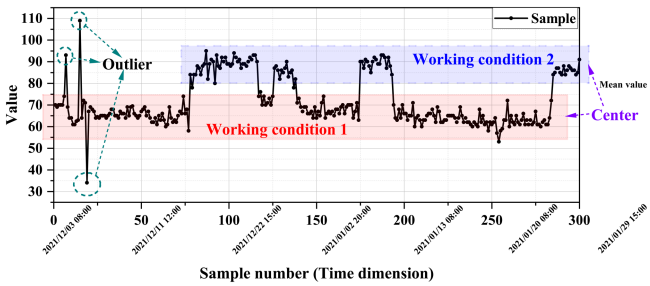

Fig. 1. Illustration of industrial data characteristics from the 2D perspective. The curve sampled at irregular intervals exhibits non-smooth nonlinear fluctuations, marked by two alternating phases of progression.

The multi-condition characteristic, alternatively denoted as the multi-phase characteristic, is a unique property of industrial manufacturing datasets as shown in Fig. 1. This periodic working condition transition reflects alterations in raw material composition or operator changes, subsequently influencing the configuration preferences of key equipment and altering the system's operational laws. Consequently, industrial time series commonly manifest similar evolutionary patterns between different periods. Notably, each working condition corresponds to a working condition center, which reflects the evolutionary pattern of the system within it. This can be briefly visualized and understood by the mean values of the varying zones of working condition 1 and 2 in Fig. 1. Evidently, the multi-condition property assumes significance in industrial manufacturing data processing due to the strong mutual referencing of time series within identical condition segments, providing meaningful cues for imputation tasks. Moreover, non-uniform sampling characteristics cannot be neglected in industrial imputation. This is because irregular

Manuscript received 6 September 2023; revised 27 February 2024; accepted 16 April 2024. Date of publication 24 April 2024; date of current version 27 September 2024. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 92267205, and Grant U1911401, and in part by the Science and Technology Innovation Program of Hunan Province in China under Grant 2021RC4054. Recommended for acceptance by L. Nie. *(Corresponding author: Chenliang Liu.)*

The authors are with the School of Automation and the National Engineering Research Centre of Advanced Energy Storage Materials, Central South University, Changsha 410083, China (e-mail: djliu@csu.edu.cn; ylwang@csu.edu.cn; lcliang@csu.edu.cn; yuanxf@csu.edu.cn; kaiwang@csu.edu.cn; ychh@csu.edu.cn).

Digital Object Identifier 10.1109/TKDE.2024.3392897

intervals between samples result in disparate reference values before and after, posing a crucial consideration [6].

The above industrial data characteristics pose formidable challenges to the imputation methods, including statistical imputation, hidden space reconstruction, and generation methods [7], [8]. Statistical methods represented by linear interpolation and multiple imputation fail to handle even basic high-dimensional nonlinearities, rendering them difficult to be effective in industrial imputation [9], [10]. Hidden space reconstruction methods navigate the challenge of high-dimensional nonlinearity by using available information to reconstruct the missing values with deep learning networks in a nonlinear space [11], [12]. The TRAE [13], AM-DAE [14], and mvts-transformer [15] are some typical methods in this category. Their performance on industrial imputation falls short of satisfaction, although having been remarkable on electrical and traffic datasets. The deleterious impact arises from the susceptibility of deterministic hidden-space reconstruction methods to low-quality and noisy data. Instances of outliers or highly noisy data points have the potential to significantly misalign the mapping of data within the hidden space. As for generative imputation methods, capturing the holistic data distribution globally, exhibits robustness against noise and outliers and efficacy in handling high-dimensional nonlinear properties [16], [17]. And they get more attention as they can easily handle the first two characteristics of industrial data. Generally, they can be divided into two types: adversarial generation and probabilistic generation [18], [19]. The adversarial generation methods such as conditional generative adversarial networks (CGANs) [20], [21] and generative adversarial imputation network (GAIN) [22], although they can achieve satisfactory results, are difficult to guarantee their performance in real industrial imputation tasks carried out by non-experts because of their intractability to training [19]. Thus, the more robust probabilistic generation methods may be a better choice for industrial imputation.

The probabilistic generative method aims to model the distribution over the entire dataset via maximum likelihood estimation and subsequent inference through sampling, ultimately imputing missing values [23], [24]. It demonstrates heightened stability and robustness in contrast to GANs, primarily through two classifications: Variable Autoencoders (VAEs) and Deep Probabilistic Diffusion Models (DDPMs). Yet, empirical findings indicate that the results of VAEs frequently exhibit ambiguity and inadequacy for imputation [7], [25], [26]. Instead, DDPMs overcome this problem by integrating deep learning and probabilistic modeling, with the goal of modeling and predicting the spread and changes of complex data [27], [28]. By introducing the probability diffusion process, DDPMs can effectively model the high-dimensional dynamic data distribution, thereby manifesting robust expression ability and probability inference ability. Some typical methods such as PriSTI [29] and CSDI [6] have also achieved excellent performance in routine imputation tasks. It is discerned that DDPMs hold considerable promise as forthcoming imputation methods for the enhancement of industrial manufacturing datasets. Nevertheless, both the nascent probabilistic generation methods epitomized by DDPMs and the adversarial generation methods exemplified by GANs overlook the multiphase attributes and non-uniform sampling characteristics inherent in industrial manufacturing datasets. Hence, the efficacy of these imputation methods in addressing industrial manufacturing datasets falls significantly short of their theoretical upper limits.

To address the above issues, this paper proposes a novel scope-free global multi-condition-aware industrial imputation framework based on the diffusion transformer (SGMCAI-DiT). The proposed framework extends DDPM into conditional probabilistic form, providing robust generation for the imputation process while leveraging a transformer network to extract nonlinear features. In this way, it adeptly addresses challenges arising from high-dimensional nonlinear coupling and low-quality noise characteristics. On this basis, further considering the multi-condition and non-uniform sampling characteristics in industrial datasets, a novel dual-weighted attention mechanism is proposed in the noise prediction model of SGMCAI-DiT, which can autonomously perceive the preferability among samples of different working conditions and take into account the nature of the preferability decaying with the increase of sampling interval. Notably, the proposed model is architected to extract the hidden features across the entire dataset, thereby enhancing the ability to model the intricate probability distribution of missing values. Specifically, 2D industrial datasets are initially chunked by working condition categories and stacked in new dimensions to form 3D tensors. Subsequently, the noise prediction model extracts short-range data features across multiple dimensions from the 2D slices within each condition and the 2D slices between different conditions with the help of the designed inter- and intra-condition feature extraction modules. This strategy enables comprehensive mining of comparable evolution patterns between the steady states of the same system across the entire dataset and captures the interplay between neighboring steady states. In this way, SGMCAI-DiT can more accurately describe the dynamic processes in industrial systems, providing more accurate modeling capabilities for industrial missing data imputation. In summary, the main contributions of this paper are as follows:

1) An industrial data imputation framework named SGMCAI-DiT is proposed to improve the imputation performance of missing data in complex industrial processes with multi-conditions and irregular sampling intervals.
2) A novel global feature extraction network is designed to augment the speed and effectiveness of extracting intricate global features throughout the entire dataset.
3) A refined double-weighted attention mechanism is devised to address the challenges of sample reference values in similar working conditions and their natural decrease over sampling intervals.
4) The experimental results on four industrial datasets and two non-industrial datasets exemplify the superior performance of the proposed SGMCAI-DiT methods over most representative state-of-the-art methods.

The remainder of this paper is organized as follows. First, the basics of the diffusion model and transformer model are briefly reviewed in Section II. Section III describes the proposed

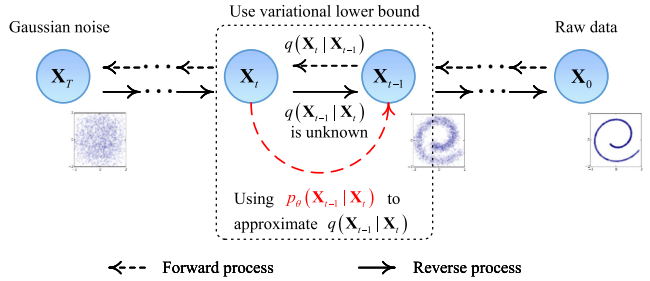

Fig. 2. The illustration of the diffusion probabilistic model.

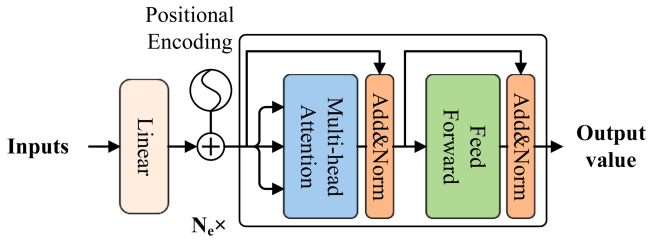

Fig. 3. The illustration of the transformer encoder.

SGMCAI-DiT framework and its training strategy in detail. Section IV presents the experimental comparison of the proposed framework with the state-of-the-art methods on two industrial process cases. Section V summarizes the paper and gives future research directions.

II. Preliminaries

A. Diffusion Probabilistic Model

The formation of the diffusion probabilistic model stems from integrating the diffusion process and probability modeling, enabling it to effectively capture complex data features and uncertainties in various applications. Fig. 2 simply illustrates a diffusion probability model consisting of a forward process and a reverse process. In the forward process modeled as a Markov process, Gaussian noise is incrementally added T times to transform the original data distribution $\mathbf{x}_0 \sim q(\mathbf{x})$ into $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

$$q\left(\mathbf{x}_{1:T}|\mathbf{x}_0\right) = \prod_{t=1}^{T} q\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right) \tag{1}$$

$$q\left(\mathbf{x}_t|\mathbf{x}_{t-1}\right) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}\right) \tag{2}$$

where $\{\beta_t \in (0,1)\}_{t=1}^{T}$ denote the incremental hyperparameters that control the variance of the Gaussian noise.

Instead, the reverse process gradually removes the introduced noise to recover $\mathbf{x}_0$. However, it is not feasible to directly calculate the posterior probability $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$. Ho et al. [27] proposed a solution by using the deep learning model to predict it as

$$p_\theta\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right) = \mathcal{N}\left(\mathbf{x}_{t-1}; \mu_\theta\left(\mathbf{x}_t, t\right), \sigma_\theta^2\left(\mathbf{x}_t, t\right)\right) \approx q\left(\mathbf{x}_{t-1}|\mathbf{x}_t\right) \tag{3}$$

$$\mu_\theta\left(\mathbf{x}_t, t\right) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\mathbf{z}_\theta\left(\mathbf{x}_t, t\right)\right), \quad \sigma_\theta^2\left(\mathbf{x}_t, t\right) = \beta_t \tag{4}$$

$$\alpha_t = 1 - \beta_t; \bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i \tag{5}$$

where $\mathbf{z}_\theta(\mathbf{x}_t, t)$ denotes the predicted noise, $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$ denotes the probability distribution obtained from the deep learning model. Hence, the objective function of t time is designed by using a variational method as

$$\mathcal{L}_t = \mathbb{E}_{\mathbf{x}_0, \bar{\mathbf{z}}_t}\left[\left\|\bar{\mathbf{z}}_t - \mathbf{z}_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\bar{\mathbf{z}}_\mathbf{t}, t\right)\right\|^2\right] \tag{6}$$

where $\bar{\mathbf{z}}_t$ and $\mathbf{z}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\bar{\mathbf{z}}_t, t)$ denote the real noise and the predicted noise at time step t, respectively.

After completing the training of the above objective function, the model can generate data samples that match the original distribution by sampling from a standard Gaussian distribution.

B. Transformer Encoder

The transformer encoder is a fundamental component of the transformer network to capture the long-range dependencies in data sequences [30]. Fig. 3 provides an illustration of transformer encoder, including multi-head attention, residual connection, batch normalization, and feedforward modules.

Suppose the input data is denoted as $\mathbf{X} \in \mathbb{R}^{N\times d_x}$. It first passes through the linear layer and positional encoding layer to convert the input data into an acceptable form as

$$\mathbf{X}_e = \mathbf{X}\mathbf{W}_e + \mathrm{PE}\left(\mathbf{X}\mathbf{W}_e\right) \tag{7}$$

$$\begin{cases} \mathrm{PE}(pos, 2i) = \sin\left(pos/10000^{2i/d_{\text{model}}}\right) \\ \mathrm{PE}\left(pos, 2i+1\right) = \cos\left(pos/10000^{2i/d_{\text{model}}}\right) \end{cases} \tag{8}$$

where $\mathbf{X}_e \in \mathbb{R}^{N\times d_{\text{model}}}$ denotes the embedded data, $\mathbf{W}_e \in \mathbb{R}^{d_x\times d_{\text{model}}}$ denotes the learned parameters.$pos \in [1, N]$ and $i \in [1, d_{\text{model}}/2\,]$ denote the sample position and dimension. N, d_x and d_{model} are the number of dataset samples, the original dimension, and the embedded dimension, respectively.

To extract the intrinsic correlations in the data, $\mathbf{X}_e$ is then passed through N_e identical layers. In each layer, the multi-head attention module is utilized to enhance the capacity to capture intricate relational nuances as

$$\begin{aligned} \mathbf{X}_\mathrm{A} &= \mathrm{LN}\left(\mathbf{X}_e + \mathrm{MultiHead}\left(\mathbf{X}_e\right)\right) \\ &= \mathrm{LN}\left(\mathbf{X}_e + \mathrm{Concat}\left(\mathbf{X}_{\mathrm{A}_1}, \ldots, \mathbf{X}_{\mathrm{A}_h}\right)\mathbf{W}_\mathrm{A}\right) \end{aligned} \tag{9}$$

$$\mathbf{X}_{\mathrm{A}_i} = \mathrm{Attention}\left(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i\right) = \mathrm{softmax}\left(\frac{\mathbf{Q}_i\mathbf{K}_i^T}{\sqrt{d_k}}\right)\mathbf{V}_i \tag{10}$$

where $\mathbf{X}_\mathrm{A} \in \mathbb{R}^{N\times d_{\text{model}}}$ and $\mathbf{X}_{\mathrm{A}_i} \in \mathbb{R}^{N\times d_{\text{model}}}$ denote the extracted features in the final and subspace, respectively. $\mathbf{Q}_i$ and $\{\mathbf{K}_i, \mathbf{V}_i\}$ denote the queries and key-value pairs obtained by $\mathbf{X}_e$ in the ith subspace. $LN(\cdot)$ denotes the layer normalization. On this basis, a feed-forward network is used to further extract

TABLE I
BASIC NOTATIONS OF SGMCAI-DIT

Symbol	Description	Shape
$\mathcal{X}$	Raw data space	$\mathbb{R}^{N\times d_x}$
$\mathbf{X}_0^O$	Observational parts of raw data	$\mathbb{R}^{N\times d_x}$
$\mathbf{X}_0^M$	Missing parts of raw data	$\mathbb{R}^{N\times d_x}$
$\mathbf{X}_t^M$	The inferred value of $\mathbf{X}_0^M$ at step t	$\mathbb{R}^{N\times d_x}$
$\mathbf{M}$	Marker matrix for missing values	$\mathbb{R}^{N\times d_x}$
$\mathbf{S}_0$	The embedded sampling timestamp	$\mathbb{R}^{N\times d_{\text{model}}}$
$\mathbf{L}_t$	The working condition type	$\mathbb{R}^{N\times 1}$
$\mathbf{C}_t$	The corresponding working condition center	$\mathbb{R}^{N\times d_x}$
$\mathbf{A}_t$	Available conditional information	/

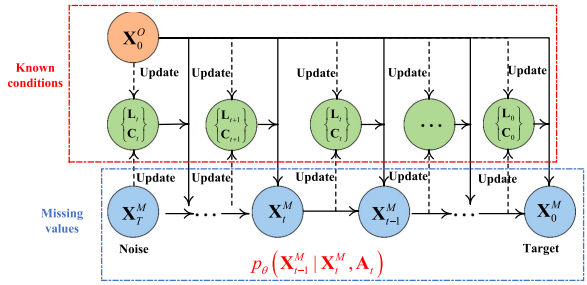

Fig. 4. The illustration of the conditional diffusion model for industrial data imputation tasks.

the nonlinear features $\mathbf{X}_{\text{F}} \in \mathbb{R}^{N\times d_{\text{model}}}$

$$\begin{aligned}\mathbf{X}_{\text{F}} &= \text{LN}\left(\mathbf{X}_{\text{A}} + \text{FeedForward}\left(\mathbf{X}_{\text{A}}\right)\right)\\ &= \text{LN}\left(\mathbf{X}_{\text{A}} + \max\left(0, \mathbf{W}_1\mathbf{X}_{\text{A}} + \mathbf{b}_1\right)\mathbf{W}_2 + \mathbf{b}_2\right)\end{aligned} \tag{11}$$

Then the learned features are input to the next layer to complete the subsequent feature transformation. After going through all layers, the final learned features are mapped as the output of the encoder.

III. METHODOLOGY

In order to provide the readers with a better understanding of the symbols used in this paper, some important symbols and descriptions are summarized in Table I.

A. Problem Definition

Suppose the raw data space is $\mathcal{X} = \{\mathbf{X}_0^O, \mathbf{X}_0^M, \mathbf{M}\}$, then the goal of imputation is to infer the missing parts $\mathbf{X}_0^M$ using the observable parts $\mathbf{X}_0^O$. In this paper, this inference process is considered as a conditional diffusion process, as shown in Fig. 4.

The forward process is similarly viewed as a progressive noise addition to the missing parts $\mathbf{X}_0^M$. It is just as described by (1)–(2), except that $\mathbf{x}_t$ is replaced by $\mathbf{X}_t^M$. While the reverse process is expanded into a conditional probability form based on (3)–(5) as

$$\begin{aligned}&p_\theta\left(\mathbf{X}_{t-1}^M \middle| \mathbf{X}_t^M, \mathbf{A}_t\right)\\ &= \mathcal{N}\left(\mathbf{X}_{t-1}^M; \mu_\theta\left(\mathbf{X}_t^M, t \middle| \mathbf{A}_t\right), \sigma_\theta^2\left(\mathbf{X}_t^M, t \middle| \mathbf{A}_t\right)\right)\end{aligned} \tag{12}$$

where $\mathbf{A}_t$ denotes the extra available auxiliary information at step t. In this paper it mainly contains the observable part $\mathbf{X}_t^M$, the working condition information $\{\mathbf{L}_t, \mathbf{C}_t\}$ and the sampling time information $\mathbf{S}_0$, and they will be described in detail in the next few sections.

Then, the true value $\mathbf{X}_0^M$ of the missing parts can be obtained after a complete inverse process as

$$\begin{aligned}&p_\theta\left(\mathbf{X}_{0:T}^M \middle| \mathbf{X}_t^M, \mathbf{A}_t\right)\\ &= p_\theta\left(\mathbf{X}_T^M\right)\prod_{t=1}^{T} p_\theta\left(\mathbf{X}_{t-1}^M \middle| \mathbf{X}_t^M, \mathbf{A}_t\right), \quad \mathbf{X}_T^M \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)\end{aligned} \tag{13}$$

If the posterior probability $p_\theta(\mathbf{X}_{t-1}^M|\mathbf{X}_t^M, \mathbf{A}_t)$ is obtained, then the data imputation task is solved. Therefore, subsequent sections of this paper will focus on how to utilize the extra available auxiliary information to help better accurately model the posteriori probability.

B. Multiple Working Condition Label-Splitting Based on Conditional Diffusion Model

In order to harness the multi-condition attributes inherent in the industrial manufacturing dataset for facilitating the imputation task, this paper introduces the condition information as additional auxiliary information into $\mathbf{A}_t$ within (12). Firstly, the working condition label and the corresponding working condition center of each sample are obtained by time series segmentation method (TSS) as

$$\mathbf{L}_t, \mathbf{C}_t = \text{TSS}\left(\mathbf{X}_t^M + \mathbf{X}_0^O, N_c\right) \tag{14}$$

where N_C denotes the number of working condition categories. Notably, the results are updated at each diffusion step using $\{\mathbf{X}_0^O, \mathbf{X}_t^M\}$. Thus, as $\mathbf{X}_{t\to 0}^M$ gradually converges to its true value, $\{\mathbf{L}_t, \mathbf{C}_t\}$ becomes progressively more accurate.

It is pertinent to highlight that, for the sake of computational simplicity, this paper employs the K-Means method within the TSS. The resultant category labels serve as condition labels, and the clustering centers as condition centers. Admittedly, more refined TSS methods, such as GMM [31] and advanced TICC approaches [32], remain viable alternatives. However, it is plausible that certain methods may not directly yield the working condition categories and centers, necessitating supplementary operations such as clustering.

Since this designed method is characterized by the inclusion of auxiliary information, the training process is similar to that of the original differential model, as described in Section II.

C. Scope-Free Global Multi-Condition-Aware Imputation Framework Via Diffusion Transformer

This section develops a novel scope-free global multi-condition-aware imputation framework via diffusion transformer (SGMCAI-DiT) to more comprehensively solve the imputation problem of industrial data. The training of SGMCAI-DiT is achieved through the cascade of multiple diffusion steps, while each individual step is clearly illustrated in Fig. 5. It mainly

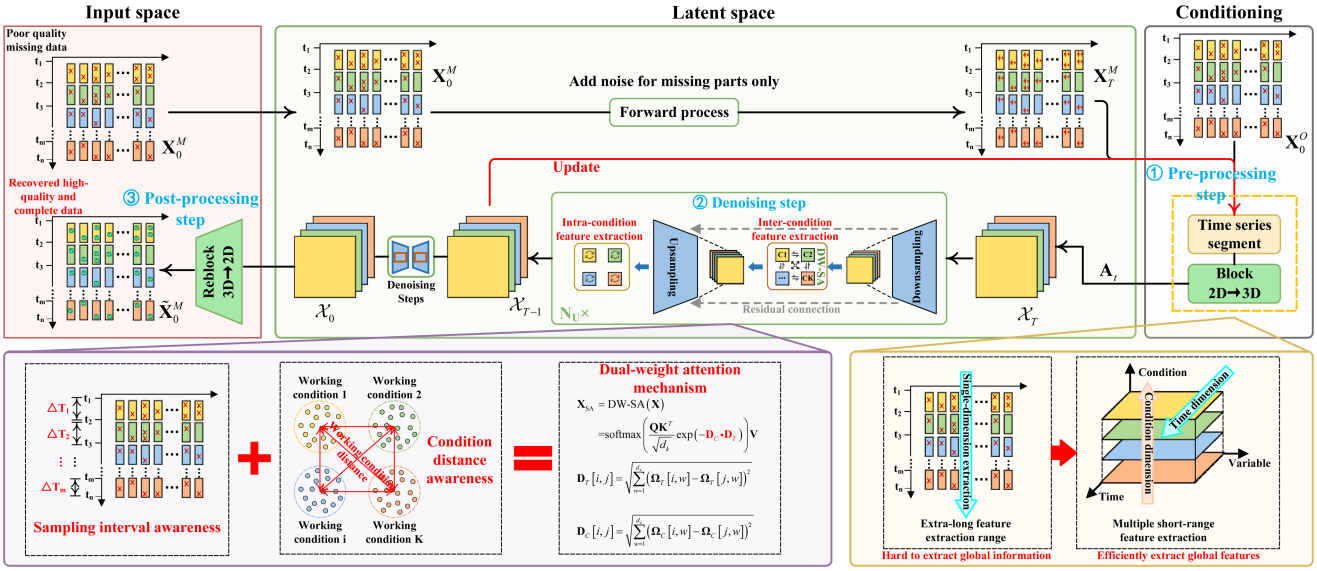

Fig. 5. The illustration of the unfolding of SGMCAI-DiT with the noise prediction model.

contains three key steps: 1) pre-processing step, which fuses auxiliary information and missing data in the condition space to construct high-quality inputs; 2) denoising step, which is innovatively designed to fully take into account the effects of different working conditions and multi-dimensional data features to guide the imputation of missing values; 3) post-processing step, which is mainly to ensure that the interpolated data conforms to the actual industrial scenario while being reliable.

1) Pre-Processing Step: At the beginning of each diffusion step t, the input data $\mathcal{X}_t = \{\mathbf{X}_0^O, \mathbf{X}_t^M, \mathbf{M}\}$ is segmented using (14) to obtain the corresponding $\{\mathbf{L}_t, \mathbf{C}_t\}$. Then, the data is partitioned into blocks of equal length M in the time dimension based on the working condition category. If the duration of a consecutive condition is shorter than the specified M length, the block is filled with the mean value. If the duration exceeds M, the block is filled with the mean value until a multiple of M is reached, at which the point is divided. This process can be expressed as

$$\mathbf{X}_{t,Block} = Block\left(Pad\left(Segment\left(\mathbf{X}_t^M + \mathbf{X}_0^O, \mathbf{L}_t\right), M\right), M\right) \quad (15)$$

where $\mathbf{X}_{t,Block}$ is a list containing all the K data blocks.

Then, all blocks are stacked in 3D space and projected into an M-dimensional space to obtain a 3D tensor with square feature mapping in the 2D slice. This process can be described mathematically as

$$\mathbf{X}_{t,3D} = Linear\left(Stack\left(\mathbf{X}_{t,Block}\right)\right) \quad (16)$$

where $\mathbf{X}_{t,3D} \in \mathbb{R}^{K\times M\times M}$ is the final 3D feature volume, and each of its 2D slices $\mathbf{X}_{t,3D} \in \mathbb{R}^{K\times M\times M}$ contains all information under the same working conditions.

The subsequent model would extract intra-condition and inter-condition features from the square 2D sliced feature maps. This approach effectively converts the ultra-long distance (N) feature extraction in the initial task into a shorter distance (K and $M \ll N$) operation in two different dimensions, thereby significantly reducing the complexity and computational burden associated with feature extraction. This also serves as the genesis of the "scope-free" concept in SGMCAI-DiT, suggesting that there exists no discernible restriction on the scope of feature extraction within the SGMCAI-DiT.

2) Denoising Step: Subsequently, the preprocessed data is fed into the designed noise prediction model, which aims to extract deep multi-condition features for estimating the noise added in the forward process. The proposed noise prediction model is underpinned by four key modules: downsampling, inter-condition transformer layer, upsampling, and intra-condition transform layer. Comprehensive explication of the functions and contributions in each module are meticulously presented in the following.

Downsampling: Extracting correlations between different working conditions is crucial when modeling the evolution of industrial processes. More specifically, it involves extracting correlations from different 2D slices. However, directly computing the correlation between two 2D feature maps is a challenging task. Hence, the following downsampling step is designed to convert 2D feature maps into 1D vectors with minimal information loss, thereby facilitating the computation of correlations.

$$\mathbf{X}_{t,3D}^{D,1} = \text{Conv}\left(\text{Reshape}_{4K\times\frac{M}{2}\times\frac{M}{2}}\left(\mathbf{X}_{t,3D}\right), 1\times 1, 4\right) \quad (17)$$

$$\mathbf{X}_{t,3D}^{D,2} = \text{Conv}\left(\text{Reshape}_{4K\times\frac{M}{4}\times\frac{M}{4}}\left(\mathbf{X}_{t,3D}^{D,1}\right), 1\times 1, 4\right) \quad (18)$$

where $\mathbf{X}_{t,3D}^{D,1} \in \mathbb{R}^{K\times M/2\times M/2}$ and $\mathbf{X}_{t,3D}^{D,2} \in \mathbb{R}^{K\times M/4\times M/4}$ denote the feature maps after downsampling once and twice, respectively. The use of the downsampling method requires an initial reshaping operation followed by convolution in a 4-channel configuration, which is an effective method to mitigate the information loss barrier. After that, $\mathbf{X}_{t,3D}^{D,2}$ is flattened to a

2D tensor as

$$\mathbf{X}_{t,2D}=\text{Flatten}\left(\mathbf{X}_{t,3D}^{D,2}\right) \tag{19}$$

where $\mathbf{X}_{t,2D}\in\mathbb{R}^{K\times M^2/16}$ denotes the flattened features, each row aggregates the main information of a continuous working condition. At this stage, the correlations between different working conditions can be extracted using common methods for correlation analysis of vectors. Since industrial systems exhibit relative stability under the same working condition, it is reasonable and feasible to use downsampling to transform the 2D working condition feature map into a 1D vector to reduce redundant information in the data.

Inter-condition transformer layer: To enhance a heightened degree of comprehension and accuracy in extracting correlations among working conditions, this paper proposes a novel dual-weight attention mechanism (DW-SA). This mechanism, distinct in its integration of temporal intervals and spatial proximity between working condition timestamps, supplants the multi-head attention mechanism in the traditional transformer encoder. Specifically, DW-SA incorporates the sampling interval between working conditions and the distance between working condition centers in the attention calculation, which provides enriched informational cues for the feature extraction process. The detailed calculation is given as follows:

$$\begin{aligned}\mathbf{X}_{t,\text{SA}}&=\text{DW-SA}\left(\mathbf{X}_{t,2D}\right)\\&=\text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\exp\left(-\mathbf{D}_C\,\text{.}\,\mathbf{D}_T\right)\right)\mathbf{V}\end{aligned} \tag{20}$$

$$\mathbf{D}_T\left[i,j\right]=\sqrt{\sum_{w=1}^{d_{\text{model}}}\left(\mathbf{S}_0\left[i,w\right]-\mathbf{S}_0\left[j,w\right]\right)^2} \tag{21}$$

$$\mathbf{D}_C\left[i,j\right]=\sqrt{\sum_{w=1}^{d_x}\left(\mathbf{C}_t\left[i,w\right]-\mathbf{C}_t\left[j,w\right]\right)^2} \tag{22}$$

where $\mathbf{D}_C\in\mathbb{R}^{K\times K}$ and $\mathbf{D}_T\in\mathbb{R}^{K\times K}$ denote the weights of the working condition center and sampling interval, respectively.

Since the proposed DW-SA highlights the importance of differences between sampling intervals and continuous working conditions, this design is well suited for inter-condition feature extraction and coincides well with real scenarios of industrial processes. Then, the complete inter-condition transformer layer can be described as follows:

$$\mathbf{X}_{t,2D}^{\text{inter}}=\text{TransformerEncoder}_{\text{DW-SA}}\left(\mathbf{X}_{t,2D}\right) \tag{23}$$

where $\mathbf{X}_{t,2D}^{\text{inter}}\in\mathbb{R}^{K\times M^2/16}$ denotes the feature map obtained by extracting correlations between conditions, and each row represents the feature information specific to a particular condition across the entire dataset.

Upsampling: After that, $\mathbf{X}_{t,2D}^{\text{inter}}$ is reconstructed by upsampling to generate 3D feature volumes, similar to the downsampling phase. The purpose of this step is to generate the feature volume using local correlations. This process is described as follows:

$$\begin{aligned}\mathbf{X}_{t,3D}^{U,1}&=\text{Conv}\left(\text{UpPad}_{\text{near}}\left(\text{Reshape}_{K\times\frac{M}{4}\times\frac{M}{4}}\left(\mathbf{X}_{t,2\text{D}}^{\text{inter}}\right)\right),3\times3,1\right)\end{aligned} \tag{24}$$

$$\mathbf{X}_{t,Block}^{U,2}=\text{Conv}\left(\text{UpPad}_{\text{near}}\left(\mathbf{X}_{t,3D}^{U,1}\right)+\mathbf{X}_{t,3D}^{D,1},3\times3,1\right) \tag{25}$$

where $\mathbf{X}_{t,3D}^{U,1}\in\mathbb{R}^{K\times M/2\times M/2}$ and $\mathbf{X}_{t,3D}^{U,2}\in\mathbb{R}^{K\times M\times M}$ denote the feature maps after upsampling once and twice, respectively. $\text{UpPad}_{\text{near}}(\text{.})$ means to double the length and width of the feature map using nearest neighbor padding. It is worth noting that residual connections are used to reduce information errors during the upsampling process.

Intra-condition transformer layer: So far, each slice in $\mathbf{X}_{t,3D}^{U,2}$ contains the evolved patterns under the same continuous working condition. Therefore, the intra-condition features can be extracted directly from each working condition using the transformer encoder. The goal of this critical step is to delve even deeper into information in each working condition. The specific process is described as follows:

$$\mathbf{X}_{t,3D}^{\text{intra}}=\text{TransformerEncoder}\left(\mathbf{X}_{t,3D}^{U,2}\right) \tag{26}$$

where $\mathbf{X}_{t,3D}^{\text{intra}}\in\mathbb{R}^{K\times M\times M}$ denotes the obtained feature map after extracting the features in the working condition. Subsequently, $\mathbf{X}_{t,3D}^{\text{intra}}$ is used as the input to the next layer of the noise prediction model to further extract deep features until all N_U layers are completed.

3) Post-Processing Step: The post-processing stage needs to fuse the features of two different dimensions to form a unified dimension to create a more consistent and comprehensive data expression for the subsequent analysis and application of the data. This process aims to reduce the dimensionality of features while retaining multidimensional information, thereby improving the interpretability of data and the stability of the model. This process is similar to preprocessing.

$$\mathbf{X}_{t,3\text{D}}^{\text{R}}=\text{ReBlocking}\left(\text{RePadding}\left(\text{Linear}\left(\mathbf{X}_{t,3D}^{\text{intra}}\right)\right)\right) \tag{27}$$

where $\mathbf{X}_{t,3D}^{\text{R}}\in\mathbb{R}^{N\times d_x}$ denote the final extracted features. $\text{RePadding}(\text{.})$ represents removing the mean value samples added for alignment length in pre-processing. $\text{ReBlocking}(\text{.})$ represents unifying the features distributed in two different dimensions to the same dimension.

At this stage, the reverse diffusion step of SGMCAI-DiT has been completed. The subsequent focus is to train the model to accurately predict the noise introduced during the forward process of the missing points in $\mathbf{X}_{t,3D}^{\text{R}}$. Hence, the loss function of the proposed SGMCAI-DiT is expressed as

$$\begin{aligned}\min_{\theta}\mathcal{L}\left(\theta\right)&=\min_{\theta}\mathbb{E}_{\mathbf{X}_0\sim q(\mathbf{X}_0),\bar{\mathbf{z}}_t\sim\mathcal{N}(0,\mathbf{I})}\left[\left\|\bar{\mathbf{z}}_t-\mathbf{z}_\theta\right\|^2\right]\\&=\min_{\theta}\mathbb{E}\left[\left\|\bar{\mathbf{z}}_t-\mathbf{z}_\theta\left(\mathbf{X}_t^{\text{M}},t\,|\mathbf{A}_t\right)\right\|^2\right]\\&=\min_{\theta}\mathbb{E}\left[\left\|\left(\mathbf{X}_t^{\text{M}}-\mathbf{X}_{t,3D}^{R}\right)\,\text{.}\,\left(\mathbf{M}-\mathbf{M}_{\text{random}}\right)\right\|^2\right]\end{aligned} \tag{28}$$

where $\mathbf{M}_{\text{random}}$ denotes the bool marker matrix obtained after randomly masking some data on the observable data points. The

Algorithm 1: The Training Process of SGMCAI-DiT.

repeat
  $t \sim \text{Uniform}(\{1, 2, \ldots, T\})$;
  Step 1: Randomly mask:
  $\mathbf{X}_0^M = \mathbf{X}_0^O \cdot (1 - \mathbf{M}_{\text{random}})$, $\mathbf{X}_0^O = \mathbf{X}_0^O \cdot \mathbf{M}_{\text{random}}$;
  Step 2: Forward process (add noise):
  $\mathbf{X}_t^M = \sqrt{\overline{\alpha}_t}\mathbf{X}_0^M + \sqrt{1 - \overline{\alpha}_t}\mathbf{z}_t$;
  Step 3: Inverse process (remove noise):
  $\mathbf{X}_{t,3D} = \text{Pre-processing}_{\text{Eqs. (15-16)}}\left(\mathbf{X}_t^M, \mathbf{X}_0^O, N_c\right)$;
  $\mathbf{X}_{t,3D}^{\text{intra}} = \text{Denoising}_{\text{Eqs. (17-26)}}\left(\mathbf{X}_{t,3D}\right)$;
  $\mathbf{X}_{t,3D}^{R} = \text{Post-processing}_{\text{Eq. (27)}}\left(\mathbf{X}_{t,3D}^{\text{intra}}\right)$;
  Step 4: Take gradient descent step using Eq. (28);
until *converged*;

Algorithm 2: The Imputation Process of SGMCAI-DiT.

for $i \leftarrow 1$ **to** Q **do**
  $\widetilde{\mathbf{X}}_t^M(i) \sim \mathcal{N}(0, \mathbf{I})$;
  for $t \leftarrow T$ **to** 1 **do**
    $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ if $t > 1$,else $\mathbf{z} = 0$;
    $\widetilde{\mathbf{X}}_{t-1}^M(i) = \frac{1}{\sqrt{\alpha_t}}\left(\widetilde{\mathbf{X}}_t^M(i) - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\mathbf{z}_\theta\left(\widetilde{\mathbf{X}}_t^M(i), t \mid \mathbf{A}_t\right)\right)$;
  end
  Return: $\widetilde{\mathbf{X}}_0^M(i)$
end
Return: $\frac{1}{Q}\sum_{i=1}^{Q}\widetilde{\mathbf{X}}_0^M(i)$
$\mathbf{X} = \mathbf{X}_0^O \cdot \mathbf{M} + \frac{1}{Q}\sum_{i=1}^{Q}\widetilde{\mathbf{X}}_0^M(i) \cdot (1 - \mathbf{M})$

original intention of this design is that since the true value of the missing part in the collected data set is difficult to obtain, this study adopts a self-supervised training strategy to address this challenge by randomly masking specific points in the observable dataset. Specifically, the forward process introduces noise to these masked points, while the inverse process focuses on accurately reconstructing these randomly masked points rather than only the true missing values. Through this strategy, the model can gain insights into the dynamics of the data from the masked points and their corresponding noise introduction during training. Importantly, this strategy enables the model to capture complex data patterns without complete data, making the training process more effective in handling real-world data with missing information.

In summary, the training process of the proposed SGMCAI-DiT is summarized as Algorithm 1.

D. SGMCAI-DiT-Based Missing Data Imputation Procedure

After completing the training process, we can reasonably assume that the deep learning model has accurately predicted $p_\theta(\mathbf{X}_{t-1}^M|\mathbf{X}_t^M, \mathbf{A}_t)$. Next, the missing data can be inferred using (13). At the operational level, Q samples of $\mathbf{X}_T^M(i) \sim \mathcal{N}(0, \mathbf{I})(0 < i < Q)$ are randomly drawn from a standard Gaussian distribution. These samples undergo a T step inverse diffusion process to generate $\tilde{\mathbf{X}}_0^M(i)$. Finally, the completed dataset $\mathbf{X} \in \mathbb{R}^{N \times d_x}$ is obtained by taking the average of these inferred values as

$$\mathbf{X} = \mathbf{X}_0^O \cdot \mathbf{M} + \frac{1}{Q}\sum_{i=1}^{Q}\tilde{\mathbf{X}}_0^M(i) \cdot (1 - \mathbf{M}) \tag{29}$$

In summary, the imputation process of the proposed SGMCAI-DiT is summarized as Algorithm 2.

IV. Experiments

In this section, the effectiveness of the proposed SGMCAI-DiT method is validated by conducting simulation experiments using extensive datasets and varied imputation scenarios.

A. Datasets

To verify the wide applicability of the proposed method, the following four different datasets are selected for simulation experiments in this paper:[1]

- *Salt Lake Chemical:* The dataset collects 4044 samples containing 29 key variables from a Salt Lake Chemical factory in China from January to June 2022. A detailed description of the variables can be found in [5].
- *Hydrocracking:* This dataset covers a total of 2607 samples from December 2015 to December 2018 in a Chinese oil refinery, and each sample contains 44 key variables. A detailed description of the process and variables can be found in [33].
- *Debutanizer:* This paper constructs an unequal interval sampling dataset, comprising 2000 samples extracted from the original dataset. Each sample encompasses 8 sensor pickups from the debutanizer tower unit. A detailed description of the dataset can be found in the [34].
- *Sulfur Recovery Unit (SRU):* Similarly, this paper constructs an unequal interval sampling dataset containing 4000 samples based on the original dataset, with each sample containing 7 sensors value in the SRU. A detailed description of the dataset can be found in the [35].

B. Baselines

1) Comparison Methods: In order to verify the effectiveness and superiority of the proposed method, this paper systematically selects representative methods from each class of imputation methods to construct comparative experiments. The detailed description of them are given as follows:

- *Linear interpolation* [9]: It utilizes local statistical properties of the data to infer missing values;
- *MICE* [10]: It synthesize the results of multiple interpolations to approximate missing values;
- *SAITS* [36]: It utilizes the self-attention mechanism to capture the spatio-temporal dependencies to accomplish the imputation task;

[1]The Debutanizer and SRU datasets are publicly available at: https://extras.springer.com/?query=978-1-84628-479-3

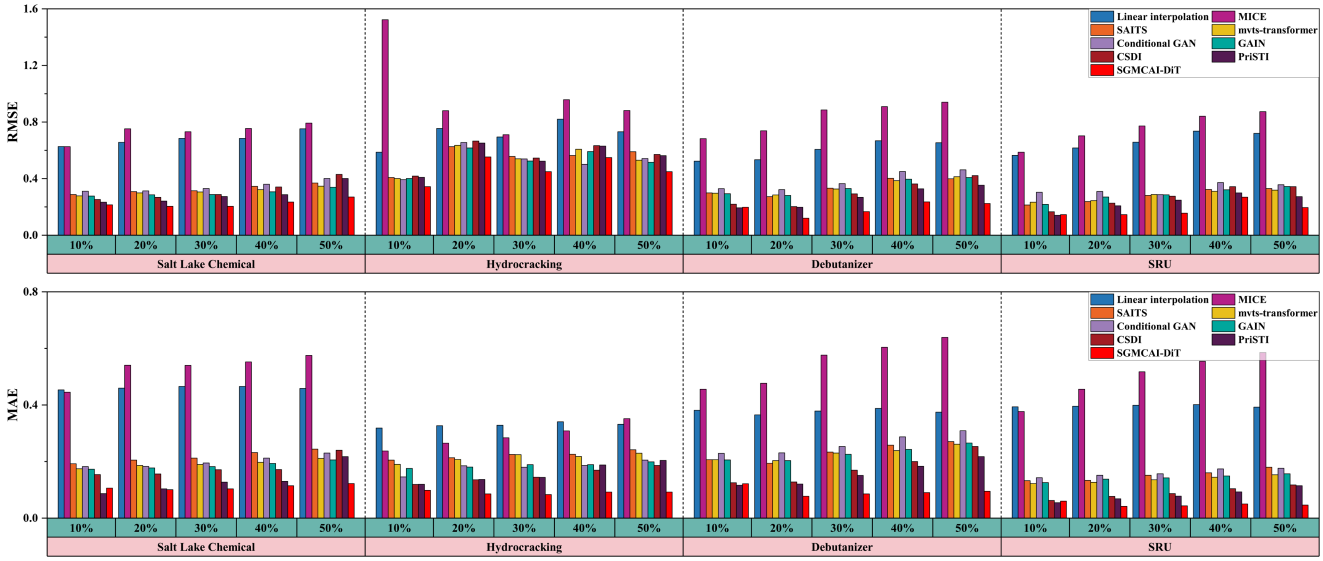

Fig. 6. The visual bar chart of the imputation results for all methods.

TABLE II
HYPERPARAMETER SETTINGS OF SGMCAI-DIT

Symbol	Description	Value
T	Diffusion step	{100, 200}
N_C	Number of working condition	{2, 3, 4}
M	Block length	{64, 128}
N_U	Noise prediction model layer	2

- *mvts-transformer* [15]: It leverages the strong nonlinear extraction ability of transformer to infer the high-dimensional representation of the missing data and then accomplish imputation;
- *Conditional GAN (CGAN)* [20]:GAN-based generation method with additional consideration of observables for task guidance;
- *GAIN* [22]:A generalized version of GAN, which basically adopts the architecture of GAN and can handle the filling of data well in case of incomplete datasets;
- *CSDI* [6]:A DDPM-based imputation method by introducing conditional probability and self-supervised training approach;
- *PriSTI* [29]: A spatio-temporal dependency extraction module is designed based on the conditional diffusion model for global context dependencies to accomplish the imputation.

2) Evaluation Criteria: In order to quantitatively evaluate the performance of the imputation methods, the root mean square error (RMSE) and the mean absolute error (MAE) are chosen as evaluation metrics in this paper. They are calculated as

$$RMSE = \sqrt{\frac{\sum_{i,j}\left(1-\mathbf{M}\left[i,j\right]\right)\left(\mathbf{X}\left[i,j\right]-\tilde{\mathbf{X}}\left[i,j\right]\right)^2}{\sum_{i,j}\left(1-\mathbf{M}\left[i,j\right]\right)}} \tag{30}$$

$$MAE = \frac{\sum_{i,j}\left(1-\mathbf{M}\left[i,j\right]\right)\left|\mathbf{X}\left[i,j\right]-\tilde{\mathbf{X}}\left[i,j\right]\right|}{\sum_{i,j}\left(1-\mathbf{M}\left[i,j\right]\right)} \tag{31}$$

where $\mathbf{X}$ and $\tilde{\mathbf{X}}$ denote the real data and the data obtained after imputation, respectively. The smaller the RMSE and MAE, the better the imputation performance of the method.

C. Experimental Setting

To comprehensively evaluate these methods, we create five scenarios with different random missing rates of 10%, 20%, 30%, 40%, and 50% and mark the real values as the ground truth for evaluation. Besides, in this study, the trial-and-error and grid search approach is utilized to select the hyperparameters of all methods. Among them, the search settings of several important hyperparameters of the proposed method are given in Table II.

D. Performance Results and Analysis

Table III delineates the exhaustive experimental findings across nine methods, and Fig. 6 offers a succinct visual comparison of their performance. These methods correspond to the different representative imputation techniques. As can be seen, the statistical approaches of linear interpolation and MICE demonstrate inferiority across all missing data scenarios and datasets compared to other methods. This reveals the inadequacy of statistical models in handling the nonlinear features of industrial data, rendering them unsuitable for industrial imputation. The SAITS and mvts-transformer, which are two concealed space reconstruction methods, and CGAN and GAIN, which are exemplified by generative adversarial techniques, demonstrate acceptable performance owing to the profound nonlinear feature extraction capabilities of deep neural networks. However, due to the lack of robustness, their performance is still some way from optimal. Moreover, it is noteworthy that these two

TABLE III
THE PERFORMANCE COMPARISON (THE AVERAGE RMSE AND MAE) OF MISSING DATA IMPUTATION ON DIFFERENT DATASETS

Datasets↓	Methods→	Statistical				Reconstruction				Adversarial				Diffusion				Proposed	
		Linear interpolation		MICE		SAITS		mvts-transformer		CGAN		GAIN		CSDI		PriSTI		SGMCAI-DiT	
	Scenarios↓	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Salt Lake Chemical	10%	0.6268	0.4533	0.6260	0.4450	0.2882	0.1920	0.2778	0.1742	0.3109	0.1825	0.2771	0.1729	0.2517	0.1541	0.2330	**0.0867**	**0.2139**	0.1063
	20%	0.6564	0.4595	0.7523	0.5403	0.3076	0.2048	0.2985	0.1859	0.3132	0.1826	0.2853	0.1772	0.2680	0.1563	0.2410	0.1034	**0.2049**	**0.1005**
	30%	0.6839	0.4653	0.7315	0.5398	0.3144	0.2124	0.3052	0.1896	0.3301	0.1948	0.2876	0.1821	0.2876	0.1709	0.2735	0.1270	**0.2043**	**0.1031**
	40%	0.6839	0.4657	0.7539	0.5522	0.3452	0.2316	0.3223	0.1975	0.3597	0.2125	0.3065	0.1935	0.3409	0.1715	0.2871	0.1298	**0.2343**	**0.1143**
	50%	0.7526	0.4584	0.7919	0.5749	0.3694	0.2441	0.3466	0.2113	0.4009	0.2302	0.3391	0.2056	0.4298	0.2398	0.4014	0.2172	**0.2695**	**0.1220**
Hydrocracking	10%	0.5867	0.3183	1.5238	0.2371	0.4073	0.2050	0.4007	0.1901	0.3925	0.1455	0.4007	0.1757	0.4181	0.1188	0.4090	0.1197	**0.3430**	**0.0981**
	20%	0.7546	0.3269	0.8803	0.2648	0.6257	0.2131	0.6351	0.2073	0.6556	0.1851	0.6162	0.1808	0.6659	0.1353	0.6508	0.1369	**0.5533**	**0.0858**
	30%	0.6948	0.3285	0.7098	0.2838	0.5571	0.2250	0.5395	0.2242	0.5393	0.1801	0.5243	0.1888	0.5453	0.1446	0.5231	0.1439	**0.4493**	**0.0831**
	40%	0.8203	0.3404	0.9580	0.3086	0.5629	0.2257	0.6083	0.2179	**0.5016**	0.1863	0.5907	0.1890	0.6337	0.1697	0.6295	0.1877	0.5484	**0.0921**
	50%	0.7306	0.3319	0.8808	0.3517	0.5904	0.2419	0.5298	0.2297	0.5422	0.2052	0.5156	0.1988	0.5713	0.1861	0.5623	0.2040	**0.4484**	**0.0924**
Debutanizer	10%	0.5237	0.3812	0.6817	0.4554	0.2989	0.2067	0.2962	0.2066	0.3289	0.2291	0.2935	0.2053	0.2186	0.1250	**0.1932**	**0.1163**	0.1980	0.1219
	20%	0.5337	0.3648	0.7380	0.4768	0.2724	0.1940	0.2845	0.2036	0.3221	0.2306	0.2806	0.2033	0.2023	0.1279	0.1975	0.1208	**0.1198**	**0.0772**
	30%	0.6063	0.3782	0.8859	0.5761	0.3317	0.2331	0.3268	0.2300	0.3646	0.2534	0.3299	0.2255	0.2926	0.1696	0.2679	0.1510	**0.1666**	**0.0853**
	40%	0.6682	0.3880	0.9084	0.6038	0.4024	0.2579	0.3859	0.2388	0.4502	0.2872	0.3951	0.2430	0.3626	0.2002	0.3274	0.1831	**0.2351**	**0.0898**
	50%	0.6531	0.3744	0.9396	0.6387	0.3989	0.2708	0.4138	0.2613	0.4623	0.3090	0.4081	0.2653	0.4212	0.2536	0.3533	0.2171	**0.2242**	**0.0948**
SRU	10%	0.5649	0.3934	0.5872	0.3764	0.2135	0.1321	0.2329	0.1221	0.3032	0.1430	0.2183	0.1254	0.1651	0.0625	**0.1412**	**0.0544**	0.1198	0.1198
	20%	0.6170	0.3957	0.7020	0.4556	0.2376	0.1332	0.2442	0.1269	0.3089	0.1518	0.2703	0.1380	0.2262	0.0768	0.2083	0.0685	**0.1451**	**0.0414**
	30%	0.6580	0.3988	0.7721	0.5171	0.2809	0.1516	0.2880	0.1353	0.2871	0.1567	0.2860	0.1423	0.2751	0.0865	0.2474	0.0778	**0.1559**	**0.0433**
	40%	0.7351	0.4011	0.8414	0.5545	0.3228	0.1601	0.3097	0.1443	0.3717	0.1737	0.3208	0.1489	0.3438	0.1038	0.2985	0.0929	**0.2692**	**0.0500**
	50%	0.7201	0.3923	0.8737	0.5855	0.3302	0.1798	0.3191	0.1534	0.3572	0.1760	0.3440	0.1569	0.3430	0.1172	0.2718	0.1145	**0.1958**	**0.0461**

The best results are highlighted in bold.

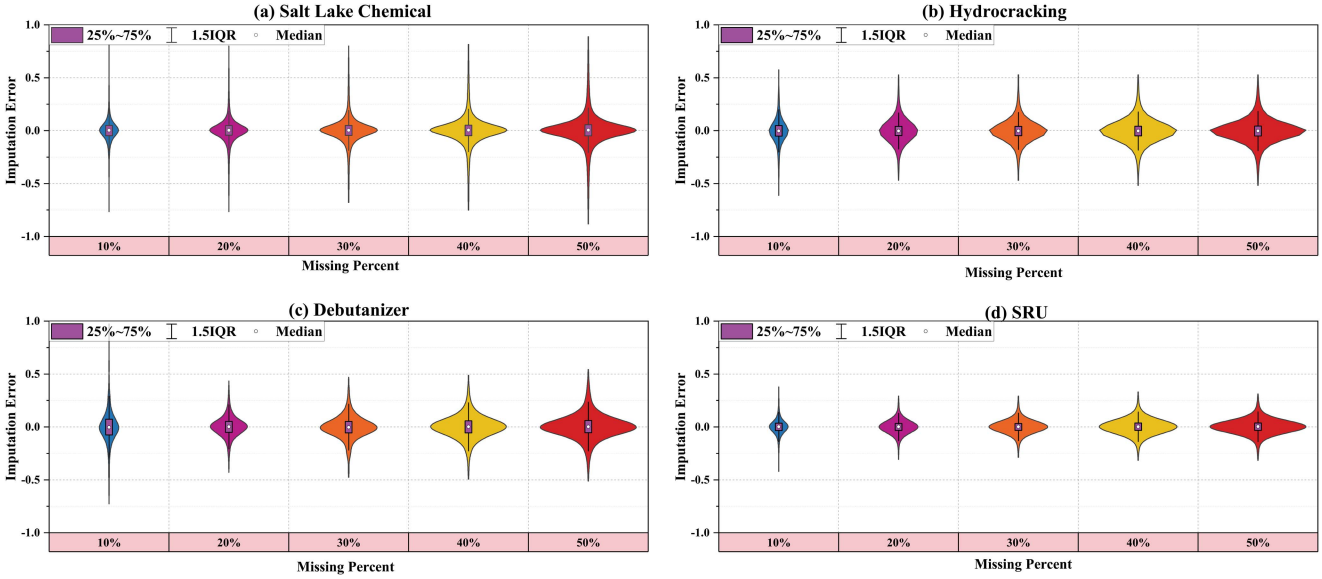

Fig. 7. The violin diagram of imputation error for SGMCAI-DiT.

types of methods typically underperform compared to DDPM-based models like CSDI and PriSTI at low missing rates. However, the performance of DDPMs declines further and becomes inferior to hidden-space reconstruction and adversarial generation methods as the missing rate increases. This is because the DDPMs, with their intricate task of modeling data distributions, require richer feature information for precise representation compared to other implicit modeling methods. As the missing rate increases, effective features diminish rapidly, leading to the performance degradation of CSDI and PriSTI. In contrast, the SGMCAI-DiT method tackles this issue twofold. Firstly, its specialized noise prediction model expands its perceptual range across the entire dataset, enhancing effective feature acquisition. Secondly, its dual-weighted attention mechanism discerns the significance of sample information across different conditions and moments, guiding and optimizing the feature extraction process efficiently. The combined superposition of these two advantages makes the proposed SGMCAI-DiT outperform other methods in most scenarios.

Furthermore, Fig. 7 displays violin plots depicting the imputation errors of SGMCAI-DiT across diverse datasets. These errors consistently exhibit an unimodal distribution centered at zero, showcasing the high accuracy and stability of the method across different missing rates.

E. Analysis of Time Complexity

This paper contrasts the time complexity of several high-performing methods including SAITS, GAIN, CSDI, and PriSTI. To ensure fairness, all experiments are performed on the 13th Gen Intel(R) Core(TM) i9-13900 K and NVIDIA GeForce RTX 4090 and torch 2.0.1 environments.

Fig. 8 depicts the average training and inference time across all methods, utilizing the evaluation metric of completing the

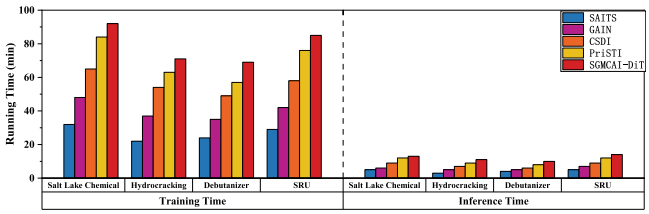

Fig. 8. The time costs of SGMCAI-DiT and other baselines.

TABLE IV
PERFORMANCE COMPARISON UNDER DIFFERENT TSS METHODS

Methods↓	Salt Lake Chemical RMSE	MAE	Training time	Hydrocracking RMSE	MAE	Training time
K-Means	0.2043	0.1031	**92 min**	0.4493	0.0831	**71 min**
GMM	0.2032	0.1014	95 min	0.4353	0.0812	74 min
TICC	**0.1903**	**0.0962**	108 min	**0.4284**	**0.0754**	87 min

The best results are highlighted in bold.

entire dataset imputation. It can be seen that since random noise prediction is more difficult than supervised learning, the training time of all DDPM-based algorithms is higher than the others. Besides, the inference time of the DDPM-based methods is higher than other methods. This is because their network structures require T times of inference for imputation. And the SGMCAI-DiT further increases time complexity due to expanded feature extraction. While the inference approach of DDIM mitigates this drawback, it remains incomparable to other methods [37]. Nevertheless, imputation tasks primarily serve as the preprocessing step for high-level tasks like process modeling and analysis, where algorithmic accuracy outweighs real-time demands, making minute-level inference times acceptable in industrial applications. In the future, reducing the inference time of DDPMs will also be one of our research priorities.

F. Sensitivity Experiments for TSS Methods

The working conditions segmentation serves as the cornerstone for the subsequent modules of SGMCAI-DiT, exerting a considerable influence on the performance. Consequently, this study assesses the impact of various time series segmentation methods, including K-Means, GMM, and TICC on the Salt Lake Chemical and Hydrocracking datasets. Table IV illustrates the imputation results with 30% missing rate, derived from diverse segmentation methods. The complex TSS method may enhance performance within limits but imposes a significant computational overhead, making it less cost-effective overall. This is because the feature extraction of SGMCAI-DiT across the entire dataset ensures robustness to the misclassification of samples. Thus, it is recommended that readers opt for simpler TSS methods, like K-Means, when using SGMCAI-DiT.

G. Ablation Experiments

This section validates the effectiveness of the proposed condition-aware (CAM) and sampling interval-aware mechanisms (SIAM) in SGMCAI-DiT through ablation experiments

TABLE V
EXPERIMENTAL RESULTS FOR FOUR ABLATION METHODS

Methods	Salt Lake Chemical RMSE	MAE	Hydrocracking RMSE	MAE
SGMCAI-DiT-N	0.2699	0.1484	0.4743	0.1188
SGMCAI-DiT-S	0.2428	0.1194	0.4679	0.1163
SGMCAI-DiT-C	0.2335	0.1163	0.4515	0.0878
SGMCAI-DiT	**0.2043**	**0.1031**	**0.4493**	**0.0831**

The best results are highlighted in bold.

with a 30% missing rate across two real industrial datasets. We systematically examine four scenarios:

- SGMCAI-DiT-N: neither CAM nor SIAM.
- SGMCAI-DiT-S: lack of CAM but with SIAM.
- SGMCAI-DiT-C: lack of SIAM but with CAM.
- SGMCAI-DiT: both CAM and SIAM.

The obtained final results are shown in Table V. From the experimental results, it can be seen that the performance of the proposed model decreases no matter missing CAM or SIAM, especially when CAM. This reveals that these two components can indeed play a role in performance enhancement in the proposed method by effectively extracting the multi-condition features and capturing the effects of uniform sampling. Notably these two components are plug-and-play, they can be ported to other models based on the self-attention mechanism for performance enhancement.

H. Generalizability Verification

It is noteworthy that SGMCAI-DiT exhibits prowess not only with process industrial datasets but also delivers competitive performance with time series datasets in other fields, even when their multiphase characteristics are less pronounced compared to those of process industry datasets. To substantiate this assertion, this paper validates its efficacy using the TrafficFlow dataset[2] and the AirQuality dataset.[3] Among these, the TrafficFlow dataset exhibits certain multiphase characteristics, whereas the AirQuality dataset lacks. Consequently, in the AirQuality dataset experiments, this paper omits the condition-aware attention mechanism. Additionally, the 2D samples are transformed into 3D samples with fixed lengths rather than using the working condition labels.

To validate the proposed method effectively, we select the superior results of two methods, CSDI and PriSTI, for comparison across the aforementioned public datasets. The final results are presented in Table VI. It is evident that the proposed SGMCAI-DiT attains satisfactory performance across various missing scenarios within the TrafficFlow and AirQuality datasets. Although the performance improvements are not as significant as on the process industrial datasets, it remains competitive with state-of-the-art methods. This stems from the structural enhancements delineated in this study, significantly broadening the scope of feature extraction. Simultaneously, this underscores

[2]https://archive.ics.uci.edu/dataset/608/traffic+flow+forecasting
[3]https://archive.ics.uci.edu/dataset/360/air+quality

TABLE VI
VERIFICATION RESULTS OF GENERALIZATION PERFORMANCE UNDER NON-PROCESS INDUSTRIAL DATASETS

Methods↓	Datasets→ Scenarios↓	TrafficFlow RMSE	TrafficFlow MAE	AirQuality RMSE	AirQuality MAE
CSDI	10%	**0.1616**	**0.1014**	0.1777	0.0582
	30%	0.1858	**0.1174**	0.2167	0.0794
	50%	0.2191	0.1416	0.2979	0.1226
PriSTI	10%	0.1656	0.1077	0.1718	0.0545
	30%	0.2238	0.1357	**0.2037**	**0.0794**
	50%	0.2963	0.2103	0.2811	0.1263
SGMCAI-DiT	10%	0.1750	0.1164	**0.1620**	**0.0514**
	30%	**0.1826**	0.1207	0.2107	0.0864
	50%	**0.2071**	**0.1411**	**0.2620**	**0.1161**

The best results are highlighted in bold.

the generalization prowess of SGMCAI-DiT in addressing time series imputation tasks.

V. CONCLUSION

Data analysis plays a key role in process monitoring tasks in the industrial field. The most urgent problem to be solved before data analysis is missing data imputation. Although efforts have been made to recover missing values in industrial process data, most advanced methods are limited by the scope of feature extraction and the multi-conditional distribution of the data. To this end, we propose an SGMCAI-DiT missing data imputation framework to enhance the imputation performance for industrial missing data. The effectiveness and advancements of SGMCAI-DiT are validated with four process industrial datasets and its generalizability is verified on two routine domain datasets. Experimental results demonstrate that the proposed SGMCAI-DiT outperforms existing methods, particularly in dealing with large-scale time series data by providing more accurate surrogate values for imputing missing data. In addition, we have explored the impact of different TSS methods on the results and the roles played by different components in the model, providing a clearer explanation of the design and ensuring the portability of the work.

REFERENCES

[1] X. Miao, Y. Wu, L. Chen, Y. Gao, and J. Yin, "An experimental survey of missing data imputation algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 6630–6650, Jul. 2023.

[2] Y. Gong, Z. Li, J. Zhang, W. Liu, Y. Yin, and Y. Zheng, "Missing value imputation for multi-view urban statistical data via spatial correlation learning," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 686–698, Jan. 2023.

[3] Y. Liu, T. Dillon, W. Yu, W. Rahayu, and F. Mostafa, "Missing value imputation for industrial IoT sensor data with large gaps," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6855–6867, Aug. 2020.

[4] C. Liu, Y. Wang, Y. Fang, C. Yang, and W. Gui, "Operating condition recognition of industrial flotation processes using visual and acoustic bimodal autoencoder with manifold learning," *IEEE Trans. Ind. Inform.*, early access, 2024, doi: 10.1109/TII.2024.3359416.

[5] D. Liu, Y. Wang, C. Liu, K. Wang, X. Yuan, and C. Yang, "Blackout missing data recovery in industrial time series based on masked-former hierarchical imputation framework," *IEEE Trans. Automat. Sci. Eng.*, vol. 21, no. 2, pp. 1138–1150, Apr. 2024.

[6] Y. Tashiro, J. Song, Y. Song, and S. Ermon, "CSDI: Conditional score-based diffusion models for probabilistic time series imputation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 24804–24816.

[7] Y. Sun, J. Li, Y. Xu, T. Zhang, and X. Wang, "Deep learning versus conventional methods for missing data imputation: A review and comparative study," *Expert Syst. Appl.*, vol. 227, 2023, Art. no. 120201.

[8] Q. Wen et al., "Time series data augmentation for deep learning: A survey," in *Proc. 30th Int. Joint Conf. Artif. Intell.*, 2021, pp. 4653–4660.

[9] P. Saeipourdizaj, P. Sarbakhsh, and A. A. Gholampour, "Application of imputation methods for missing values of pm10 and o3 data: Interpolation, moving average and k-nearest neighbor methods," *Environ. Health Eng. Manage. J.*, vol. 8, no. 3, pp. 215–226, 2021.

[10] M. D. Samad, S. Abrar, and N. Diawara, "Missing value estimation using clustering and deep learning within multiple imputation framework," *Knowl.-Based Syst.*, vol. 249, 2022, Art. no. 108968.

[11] W. Xinfeng and H. Wei, "Data imputation for methylation by variational auto-encoder," *J. Comput. Eng. Appl.*, vol. 58, no. 12, pp. 149–154, Jun. 2022.

[12] S. Phung, A. Kumar, and J. Kim, "A deep learning technique for imputing missing healthcare data," in *Proc. 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, 2019, pp. 6513–6516.

[13] X. Lai, X. Wu, L. Zhang, W. Lu, and C. Zhong, "Imputations of missing values using a tracking-removed autoencoder trained with incomplete data," *Neurocomputing*, vol. 366, pp. 54–65, 2019.

[14] Z. Pan, Y. Wang, K. Wang, H. Chen, C. Yang, and W. Gui, "Imputation of missing values in time series using an adaptive-learned median-filled deep autoencoder," *IEEE Trans. Cybern.*, vol. 53, no. 2, pp. 695–706, Feb. 2023.

[15] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, New York, NY, USA: Association for Computing Machinery, 2021, pp. 2114–2124.

[16] M. Kachuee, K. Karkkainen, O. Goldstein, S. Darabi, and M. Sarrafzadeh, "Generative imputation and stochastic prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1278–1288, Mar. 2022.

[17] S. Yoon and S. Sull, "GAMIN: Generative adversarial multiple imputation network for highly missing data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8453–8461.

[18] Y. Luo, X. Cai, Y. Zhang, J. Xu, and Y. Xiaojie, "Multivariate time series imputation with generative adversarial networks," in *Advances in Neural Information Processing*, S. Systems, H. Bengio, H. Wallach, K. Larochelle, N. GraumanCesa-Bianchi, and R. Garnett Eds., Red Hook, NY, USA: Curran Associates, Inc., 2018.

[19] Y. Luo, Y. Zhang, X. Cai, and X. Yuan, "E2GAN: End-to-end generative adversarial network for multivariate time series imputation," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 3094–3100.

[20] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.

[21] Y. Xiang, Y. Fu, P. Ji, and H. Huang, "Incremental learning using conditional adversarial networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6618–6627.

[22] J. Yoon, J. Jordon, and M. van der Schaar, "GAIN: Missing data imputation using generative adversarial nets," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 5689–5698.

[23] L. Theis, A. van den Oord, and M. Bethge, "A note on the evaluation of generative models," in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 1–10.

[24] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model," *IEEE Trans. Audio Speech Lang. Process.*, vol. 16, no. 2, pp. 435–447, Feb. 2008.

[25] A. Nazábal, P. M. Olmos, Z. Ghahramani, and I. Valera, "Handling incomplete heterogeneous data using vaes," *Pattern Recognit.*, vol. 107, 2020, Art. no. 107501.

[26] V. Fortuin, D. Baranchuk, G. Raetsch, and S. Mandt, "GP-VAE: Deep probabilistic time series imputation," in *Proc. Twenty Third Int. Conf. Artif. Intell. Statist.*, 2020, pp. 1651–1661.

[27] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin Eds, Red Hook, NY, USA: Curran Associates, Inc., 2020, pp. 6840–6851.

[28] K. Rasul, C. Seward, I. Schuster, and R. Vollgraf, "Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting," in *Proc. 38th Int. Conf. Mach. Learn.*, 2021, pp. 8857–8868.

[29] M. Liu, H. Huang, H. Feng, L. Sun, B. Du, and Y. Fu, "PriSTI: A conditional diffusion framework for spatiotemporal imputation," in *Proc. IEEE 39th Int. Conf. Data Eng.*, 2023, pp. 1927–1939.
[30] A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon Eds., Red Hook, NY, USA: Curran Associates, Inc., 2017.
[31] J. D. Banfield and A. E. Raftery, "Model-based gaussian and non-gaussian clustering," *Biometrics*, vol. 49, pp. 803–821, 1993.
[32] D. Hallac, S. Vare, S. Boyd, and J. Leskovec, "Toeplitz inverse covariance-based clustering of multivariate time series data," in *Proc. 23 rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, New York, NY, USA: Association for Computing Machinery, 2017, pp. 215–223.
[33] D. Liu, Y. Wang, C. Liu, X. Yuan, C. Yang, and W. Gui, "Data mode related interpretable transformer network for predictive modeling and key sample analysis in industrial processes," *IEEE Trans. Ind. Inform.*, vol. 19, no. 9, pp. 9325–9336, Sep. 2023.
[34] D. Liu, Y. Wang, C. Liu, X. Yuan, and C. Yang, "Multirate-former: An efficient transformer-based hierarchical network for multistep prediction of multirate industrial processes," *IEEE Trans. Instrum. Meas.*, vol. 73, 2024, Art. no. 2502313.
[35] L. Fortuna et al., *Soft Sensors for Monitoring and Control of Industrial Processes*. Berlin, Germany: Springer, 2007, vol. 22.
[36] W. Du, D. Côté, and Y. Liu, "Saits: Self-attention-based imputation for time series," *Expert Syst. Appl.*, vol. 219, 2023, Art. no. 119619.
[37] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," 2021, *arXiv:2010.02502*.

Diju Liu received the BEng degree in automation from Central South University, Changsha, China, in 2021. He is currently working toward the PhD degree in control science and engineering. His research interests include deep learning and artificial intelligence, machine learning and pattern recognition, industrial Big Data, and process modeling and control.

Yalin Wang (Senior Member, IEEE) received the BEng degree in automation and PhD degree in control science and engineering from the Department of Control Science and Engineering, Central South University, Changsha, China, in 1995 and 2001, respectively. She is currently a professor with the School of Automation, Central South University. Her research interests include the modeling, optimization and control for complex industrial processes, intelligent control, and process simulation.

Chenliang Liu received the BEng degree in automation from the School of Automation, Harbin University of Science and Technology, Harbin, China, in 2019. He is currently working toward the PhD degree in control science and engineering with the School of Automation, Central South University, Changsha, China. From 2023 to 2024, he was a joint training Ph.D. student with the School of Chemistry, Chemical Engineering and Biotechnology, Nanyang Technological University, Singapore. His research interests include deep learning, soft sensor modeling, and control of industrial processes.

Xiaofeng Yuan (Member, IEEE) received the BEng degree in automation and PhD degree in control science and engineering from the Department of Control Science and Engineering, Zhejiang University, Hangzhou, China, in 2011 and 2016, respectively. From 2014 to 2015, he was a visiting scholar with the Department of Chemical and Materials Engineering, University of Alberta, Edmonton, AB, Canada. He is currently a professor with the School of Automation, Central South University. His research interests include deep learning and artificial intelligence, machine learning and pattern recognition, industrial process soft sensor modeling, and process data analysis, etc.

Kai Wang (Member, IEEE) received the BEng and PhD degrees from the College of Control Science and Engineering, Zhejiang University, Hangzhou, China, in 2014 and 2019, respectively. He was a visiting scholar with the Department of Chemical and Biological Engineering, University of British Columbia, Vancouver, BC, Canada. He is currently a professor with the School of Automation, Central South University, Changsha, China, where he is involved in industrial data analytics, process health management, and machine learning.

Chunhua Yang (Fellow, IEEE) received the MEng degree in automatic control engineering and the PhD degree in control science and engineering from Central South University, Changsha, China, in 1988 and 2002, respectively. She was with the Department of Electrical Engineering, Katholieke Universiteit Leuven, Leuven, Belgium, from 1999 to 2001. She is currently a full professor with Central South University. Her current research interests include modeling and optimal control of complex industrial process, intelligent control system, and fault-tolerant computing of real-time systems.