MetaIndux-TS: Frequency-Aware AIGC Foundation Model for Industrial Time Series

Haiteng Wang, *Graduate Student Member, IEEE*, Lei Ren, *Senior Member, IEEE*, Yikang Li, *Member, IEEE*, and Yuqing Wang, *Graduate Student Member, IEEE*

***Abstract*—Implementing advanced AI techniques in industrial manufacturing requires large volumes of annotated sensor data. Unfortunately, collecting such data is often impractical due to extreme environments and the manual burden of expert annotation. Recent advancements in artificial intelligence generated content (AIGC) have inspired the exploration of industrial time-series generation to mitigate data shortages. However, existing AIGC models encounter difficulties in generating industrial time series due to their complex temporal dynamics, multichannel intercolumn correlations, and diverse frequency characteristics. To address these challenges, we propose MetaIndux-TS, a frequency-informed AIGC foundation model based on diffusion model frameworks. This model is designed to generate industrial time-series data under a variety of working conditions, across different types of equipment, and with variable lengths. Specifically, MetaIndux-TS integrates dual-frequency cross-attention networks, transforming time series into the frequency domain to model multivariate dependencies and capture intricate temporal details. In addition, the contrastive synthesis layer is constructed to generate high-fidelity time series by comparing periodic and long-term trends with initial noisy sequences. Comprehensive experiments show that MetaIndux-TS outperforms state-of-the-art models (SSSD, Dit, and TabDDPM), achieving a 57.5% improvement in fidelity and 20.4% in predictive score. MetaIndux-TS exhibits zero-shot generation capabilities for samples under unseen conditions, offering the potential to address data collection challenges in extreme environments. Codes are available at: https://github.com/Dolphin-wang/MetaIndux**

***Index Terms*—Artificial intelligence generated content (AIGC), diffusion model, foundation model, generative model, industrial time series.**

I. Introduction

In RECENT years, artificial intelligence generated content (AIGC) technologies have made remarkable progress in various domains [1], [2], including computer vision (CV) [3], natural language processing (NLP) [4], [5], and audio [6], [7]. These models leverage generative AI techniques to simulate the distribution of original data, achieving high-quality generation results. They have been widely applied, such as image enhancement training [8], art design [9], and the creation of generative world models [10].

In industrial manufacturing, the implementation of advanced AI techniques [11] requires substantial volumes of annotated sensor data, but this is often impractical due to challenges, such as extreme environments, sensor noise, high-frequency data processing, and the manual burden of expert annotation. The significant achievements of AIGC foundation models inspire further exploration of their potential in industrial manufacturing. Time-series AIGC foundation models have the potential to generate time-series data, addressing data shortages and enriching the diversity of industrial signals [12]. This, in turn, enhances the training of industrial deep learning models.

Despite the significance of foundational generative models, industrial time series exhibit complex temporal dynamics and diverse frequency variables that current generative models struggle to synthesize with high fidelity. Time-series generative models can be categorized into variational autoencoders (VAEs) [13], [14], generative adversarial networks (GANs) [15], [16], and diffusion models [17], [18]. VAEs utilize an encoder-decoder architecture to encode the data distribution into a latent space representation and reconstruct the original data via the decoder. However, VAEs often struggle to capture the intricate structure of high-dimensional data and exhibit limitations in generating data with rich diversity. GANs employ adversarial training between a generator and a discriminator, enabling the generator to produce sequences increasingly akin to real data distribution. Despite this, GANs suffer from training instability and are prone to mode collapse. Recently, diffusion models have achieved significant success in image and speech generation by progressively adding and removing noise to generate high-quality data samples. Although diffusion models have been initially applied in time series [6], [19], they are constrained to a single frequency and simple industrial signals, resulting in a limited ability to capture temporal dynamic patterns. The construction of the industrial time-series AIGC foundation model faces the following challenges.

1) *Multichannel Intercolumn Correlations:* Unlike continuous pixel values in images, the dimensions in industrial time-series data represent different variables, each with unique meanings. The data involve multiple variables with complex interdependencies, making it difficult to learn the joint probabilities across these variables.

Received 14 December 2024; revised 27 March 2025 and 5 May 2025; accepted 2 June 2025. Date of publication 23 June 2025; date of current version 9 October 2025. This work was supported by the National Natural Science Foundation of China (NSFC) under Project 62225302, Project 623B2014, and Project 62173023. *(Corresponding author: Lei Ren.)*

Haiteng Wang, Yikang Li, and Yuqing Wang are with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China (e-mail: wanghaiteng@buaa.edu.cn; liyikang@buaa.edu.cn).

Lei Ren is with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China, also with Hangzhou International Innovation Institute, Beihang University, Hangzhou, Zhejiang 311115, China, and also with the State Key Laboratory of Intelligent Manufacturing System Technology, Beijing 100854, China (e-mail: renlei@buaa.edu.cn).

Digital Object Identifier 10.1109/TNNLS.2025.3577203

2) *Complex Temporal Dynamics Patterns:* Industrial time-series data are collected from physical devices and reflect the dynamic details of equipment influenced by real-world environments and performance over time. These data include long-term degradation processes and short-term dynamic changes, such as temperature and pressure fluctuations. The temporal dynamics make it challenging for diffusion models designed for static images to synthesize industrial time-series effectively.
3) *Diverse Frequency Variables:* Fundamental industrial time-series information is embedded in the frequency domain, such as the frequencies of vibration signals. Although recent diffusion models can generate time series, they primarily focus on temporal domains, neglecting the rich information in the frequency domain.

To address these challenges, we have constructed a frequency-informed AIGC foundation model named MetaIndux-TS. This model is specifically designed for generating industrial time-series data across diverse operating conditions, with various types of equipment, and at variable lengths. The core idea is to integrate frequency domain information to enhance the time-series generation process, thereby facilitating the learning of complex and diverse patterns. Specially, MetaIndux-TS integrates dual frequency cross-attention learners into the diffusion framework, which transforms time series into the frequency domain enabling the modeling of multivariate dependencies and the capture of intricate temporal details. Then, considering the periodicity and long-term trends of time series and comparing them with initial noisy sequences, a contrastive synthesis layer is proposed to generate high-fidelity time-series data. Our contribution can be summarized as follows.

1) A novel AIGC foundation model for industrial time series is proposed, which integrates dual-frequency domain information to enhance the generation process. Through comprehensive experiments, MetaIndux-TS achieves state-of-the-art performance in generating high-fidelity time series. Furthermore, it demonstrates strong few-shot and zero-shot generation capabilities, capable of synthesizing unseen data.
2) Dual-frequency cross learners are proposed to leverage the strengths of both frequency domain analysis and attention mechanisms. Specifically, the frequency cross-channel learner models multivariate relationships between channels, while the frequency cross-temporal learner captures dynamic patterns of time series.
3) The contrastive synthesis layer employs a deep decomposition model architecture, allowing the generated time series to synthesize both long-term degradation information and short-term detail information, promoting the precise reproduction of the intrinsic temporal details of the data during the generation process.

II. Related Works

A. Generative Foundation Models

The generative foundation models relevant to our work are mainly based on diffusion probabilistic models, such as stable diffusion [20] and Sora [21]. Diffusion probabilistic models have been widely applied across various domains, achieving remarkable success. Initially, these models gained prominence in image synthesis, where methods like denoising diffusion probabilistic models (DDPMs) [22] progressively added Gaussian noise and then learned to reverse this process for high-fidelity sample generation. Building on this, advanced models such as DDIM [23] and latent diffusion models (LDMs) [20] improved efficiency by accelerating reverse sampling and reducing computational overhead through latent space diffusion. These advancements extended to video generation [24], introducing both text-to-video and image-to-video models, which have significantly advanced video synthesis technology in academia and industry. Furthermore, diffusion models have contributed to the design of protein structure [25] and NLP [26] by generating coherent and contextually accurate content.

Unlike diffusion models primarily designed for images or videos with continuous pixel values, MetaIndux-TS focuses on time-series data characterized by complex frequency components and intervariable relationships. MetaIndux-TS introduces a frequency-enhanced framework specifically tailored for time-series generation, enhancing the granularity and detail of generated sequences.

B. Generative Models for Industrial Time Series

In the realm of industrial time-series generation, several deep generative models have been explored. VAEs [13], [14] use an encoder-decoder structure to encode data distributions into a latent space and reconstruct the original data. However, VAEs often struggle with capturing the intricate structures of high-dimensional data and generating diverse samples. GANs [15], [16] leverage adversarial training between a generator and a discriminator to produce sequences that closely resemble the real data distribution. Despite their capabilities, GANs face issues with training instability and mode collapse. Diffusion models, with their success in other domains, also show potential in industrial applications. A diffusion model combined with a U-Net is proposed for extracting features to enhance fault diagnosis [17]. A novel adaptive dynamic neighbor mask (ADNM) mechanism, the Transformer and denoising diffusion model are combined to effectively identify anomalies by mitigating reconstruction challenges, achieving state-of-the-art performance [18]. However, they are constrained to single frequency and simple industrial signals, resulting in limited frequency perception and temporal dynamics patterns capturing capabilities.

Our method considers the temporal dynamics, multivariate dependencies, and frequency characteristics inherent in industrial time series to address these limitations. We propose a channel-frequency attention mechanism to enhance feature representation for frequency-rich data and a temporal-frequency attention mechanism to capture both long-term degradation patterns and short-term variations.

III. Methodology

A. Problem Description and Model Advantages

This section briefly revisits the challenges introduced in Section I and presents the advantages of MetaIndux-TS in

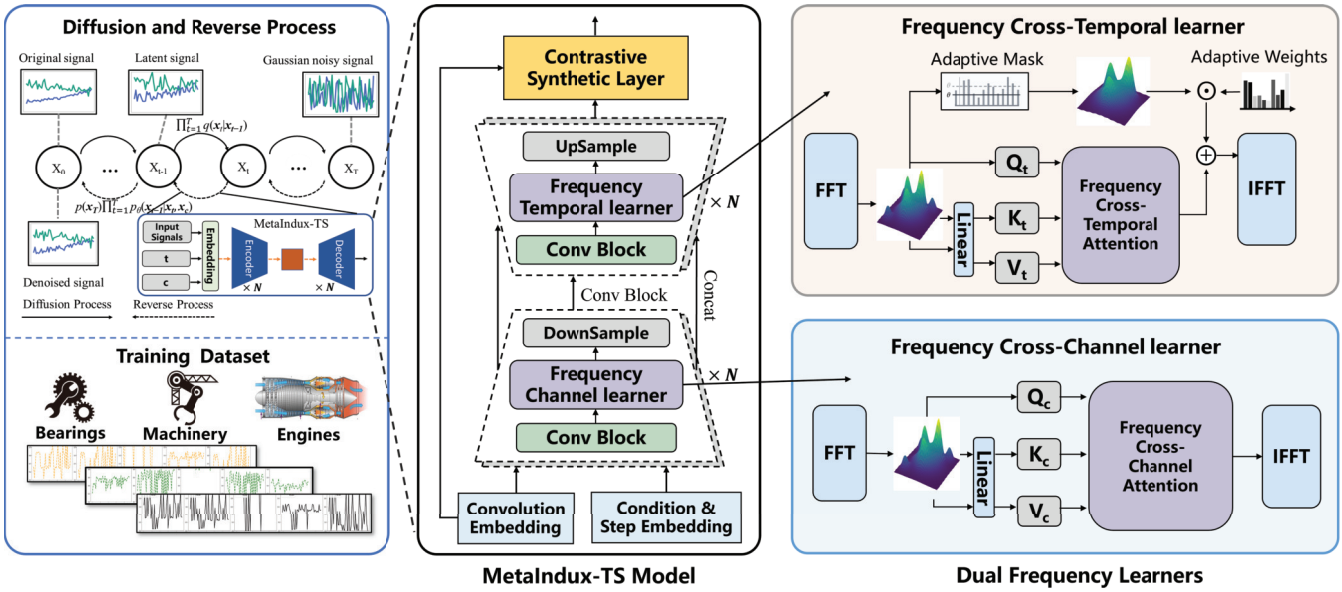

Fig. 1. MetaIndux-TS framework for industrial time-series generation. The model utilizes both a frequency temporal learner and a frequency channel learner to model the signals in the frequency domain. The right side describes potential industrial applications of the model, such as predictive maintenance (limited sample augmentation of complex equipment), or industrial metaverse and digital twin systems.

addressing these challenges. Currently, industrial time-series generation faces the following three major challenges.

1) *Multichannel Intercolumn Correlations:* Industrial time-series data exhibit complex intervariable correlations, making it challenging to model their joint distributions.
2) *Complex Temporal Dynamics Patterns:* The data contain intricate temporal dynamics, including both long-term degradation and short-term fluctuations, which complicate effective sequence generation.
3) *Diverse Frequency Variables:* Essential information reside in the frequency domain, but existing diffusion models mainly focus on the temporal domain and overlook frequency features.

To address these challenges, we propose MetaIndux-TS, which offers the following key advantages.

1) *Modeling Multichannel Dependencies:* For time-series generation, it is essential to account for multichannel dependencies, as they enable the model to capture interactions and correlations between different variables. MetaIndux-TS addresses this requirement through its frequency cross-channel learner, which performs cross-channel attention after transforming data into the frequency domain at each timestamp, thereby modeling the multichannel intercolumn correlations in industrial time-series data.
2) *Capturing Temporal Dynamic Details:* Industrial time-series data are inherently rich in the frequency-domain information. The frequency cross-temporal learner in MetaIndux-TS is designed to capture both dynamic details and long-term temporal patterns. In addition, the model adaptively attenuates high frequencies to reduce noise and enhance signal clarity, thus enhancing the high fidelity of the generated data.
3) *Synthetic Layer for Trend and Seasonality:* After learning the temporal patterns and cross-channel dependencies, the contrastive synthesis layer reconstructs the generated time series by explicitly modeling trend and seasonality components in the frequency domain. This results in improved fidelity and a more accurate representation of the industrial time series.

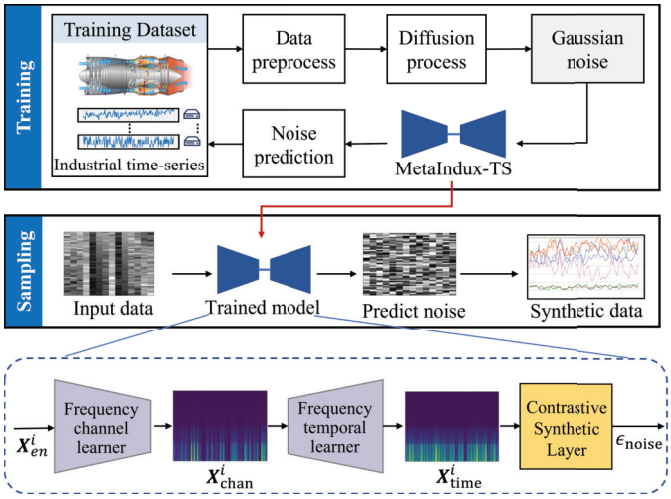

Fig. 2. Workflow of the MetaIndux-TS.

B. Framework of MetaIndux-TS

MetaIndux-TS is an AIGC foundation model based on the diffusion probabilistic architecture, as shown in Fig. 1, specifically designed for industrial time-series generation.

1) Workflow: The workflow is depicted in Fig. 2, MetaIndux-TS generates realistic synthetic time series by iteratively denoising random Gaussian noise, effectively capturing the temporal dependencies of the original data. The workflow steps involved in each phase are detailed below.

In the training phase, real-world aircraft engine sensor data are first collected and preprocessed through normalization,

cleaning, and handling of missing values to ensure consistency. The preprocessed data then undergo a forward diffusion process, where Gaussian noise is progressively added over multiple steps, gradually transforming the data into a standard Gaussian distribution, with the noise intensity controlled by α_t. Finally, MetaIndux-TS model is trained to reverse the noise effects by learning to predict and remove the added noise, thereby recovering the original time-series data.

In the sampling phase, the process begins with a random Gaussian noise sample, which serves as the starting point for generating synthetic data. Using the pretrained MetaIndux-TS model, the reverse diffusion process is performed iteratively to gradually remove noise and reconstruct synthetic time-series data that closely resembles the original sensor data. At each reverse step t, the denoised data sample $\boldsymbol{x}_{t-1}$ is computed based on the distribution

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}; \mu_\theta(\boldsymbol{x}_t, t), \Sigma_\theta(\boldsymbol{x}_t, t))$$

where μ_θ and Σ_θ denote the mean and variance estimates predicted by the MetaIndux-TS model for step t, respectivley.

2) Overall Network Architecture: Following the work [22], our proposed model also employs a U-Net-like structure to generate time-series data. We modify the convolutional block to fit for time-series signals instead of images. The key innovation lies in proposing the frequency temporal learner, frequency channel learner, and synthetic block to capture characteristics of industrial time series. The overall architecture is described below, as shown in Fig. 1.

a) Input/Step/Condition Embedding: The embedding module processes the latent variable, condition, and timestep t to preserve temporal relationships and condition information. The latent variable is encoded using 1-D convolution, while the timestep information are represented through sinusoidal positional embeddings, which provide a structured way to encode temporal dependencies. By leveraging sinusoidal functions, this embedding method captures relative positional information across different timesteps, making it particularly suitable for modeling periodic patterns in time series. The embedded timestep representations are then processed through fully connected (FC) layers to enhance their expressiveness before being integrated into the model. The condition information is encoded using an FC network with masking to ensure selective conditioning.

b) U-Net-Based Architecture: The primary architecture consists of encoders, decoders, and a contrastive synthesis layer, aimed at capturing and reconstructing time-series data. Specifically, each encoder includes two convolutional layers and a 1/2 downsampling block, integrating a frequency channel learner to encode the entire time series and model multivariate correlations. For the convolutional layers, embedded time steps and condition vectors are inputted into each layer to retain diffusion steps and condition information. The middle Conv block serves as a bridge, refining abstract features between the encoder and decoder stages. Each decoder corresponds to an encoder, containing convolutional blocks, frequency learners, and a $2\times$ upsampling block to enhance details and restore signals to their original dimensions. Finally, the contrastive synthesis layer reconstructs the time series by synthesizing trend and seasonal information, denoising against the initial input, and ultimately generating high-fidelity time-series data.

C. Frequency Cross-Channel Learner

Considering channel dependencies for time-series generation is crucial as it allows the model to capture interactions and correlations between different variables. The frequency cross-channel learner is proposed to facilitate communication between different channels by employing frequency cross-channel attention at each timestamp, thus enabling the learning of multichannel intercolumn correlation. This approach aims to leverage the strengths of both frequency-domain analysis and attention mechanisms.

Let $\boldsymbol{X}^i_{\text{en}} \in \mathbb{R}^{T\times C}$ be the input of the ith encoder layer, where T is the number of timesteps for each time series and C is the channel numbers. The channel dependencies learner takes $\boldsymbol{X}^i_{\text{en}}$ as input and consists of three stages.

1) *Frequency Conversion:* This stage converts the input time series $\boldsymbol{X}^i_{\text{en}}$ along the timestep dimension to the frequency domain to capture global frequency information. This converts the input from the time domain to the frequency domain

$$\boldsymbol{X}_{\text{FFT}} = \text{FFT}\left(\boldsymbol{X}^i_{\text{en}}\right) \tag{1}$$

where $\boldsymbol{X}_{\text{FFT}} \in \mathbb{C}^{C\times(T/2+1)}$. The FFT allows the model to analyze periodic patterns and frequency components.

2) *Frequency Cross-Channel Attention:* $\boldsymbol{X}_{\text{FFT}}$ serves as the initial query $\boldsymbol{Q}$, while $\boldsymbol{K} = \text{FFT}(\boldsymbol{W}_K\boldsymbol{X}^i_{\text{en}})$ and $\boldsymbol{V} = \text{FFT}(\boldsymbol{W}_V\boldsymbol{X}^i_{\text{en}})$, where $\boldsymbol{W}_K$ and $\boldsymbol{W}_V$ are learnable weight matrices. The direct conversion of $\boldsymbol{Q}$ to frequency is for obtaining the original frequency features, while the linear transformation of $\boldsymbol{K}, \boldsymbol{V}$ is for more flexible feature representations. After converting to the frequency domain, the $\boldsymbol{Q}$, $\boldsymbol{K}$, and $\boldsymbol{V}$ matrices are transposed to focus on the channel dimension. Then, frequency channel attention mechanisms are employed to capture dependencies across different channels. This process can be formulated as follows:

$$\begin{aligned} \boldsymbol{Q}_c &= \boldsymbol{Q}^T, \quad \boldsymbol{K}_c = \boldsymbol{K}^T \\ \boldsymbol{V}_c &= \boldsymbol{V}^T \\ \text{FreqAtten}_{\text{chan}}\left(\boldsymbol{Q}_c, \boldsymbol{K}_c, \boldsymbol{V}_c\right) &= \text{Softmax}_{\text{chan}}\left(\frac{\boldsymbol{Q}_c\boldsymbol{K}_c^T}{\sqrt{d_k}}\right)\boldsymbol{V}_c \end{aligned} \tag{2}$$

where d_k is the dimensionality of the $\boldsymbol{K}_c$ vectors. $\text{Softmax}_{\text{chan}}(\cdot)$ means applying the softmax function along the channel dimension to normalize the attention weights, allowing the network to emphasize the influence of different channels and learn multichannel correlations in the frequency feature domain.

3) *Frequency Inversion:* The frequency attention vector $\boldsymbol{X}^i_{\text{att}}$ obtained from the attention mechanism is mapped back to the original time-series domain for further downsample processing. This process can be expressed as follows:

$$\begin{aligned} \boldsymbol{X}^i_{\text{chan}} &= \text{IFFT}\left(\boldsymbol{X}^i_{\text{att}}\right) \\ \boldsymbol{X}^{i+1}_{\text{en}} &= \text{DownSample}\left(\boldsymbol{X}^i_{\text{chan}}\right). \end{aligned} \tag{3}$$

D. Frequency Cross-Temporal Learner

The frequency temporal learner utilizes the Fourier domain and cross-temporal attention mechanisms to capture long-range dependencies and intricate patterns that are challenging to identify in the time domain. In addition, high-frequency components often represent rapid fluctuations that deviate from the underlying trend, making them appear more random. Therefore, we propose an adaptive high frequency mask that allows the model to dynamically adjust the level of filtering and remove these high-frequency noisy components.

1) Frequency Conversion and Frequency Cross-Temporal Attention: To adaptively select important information across the temporal dimension, we propose the frequency cross-temporal attention mechanism. This approach, similar to frequency cross-channel attention, employs FFT and linear transformations for query, key, and value matrices. The difference is that the frequency cross-temporal attention focuses on the temporal dimension without transposing the attention matrix. The process is described as follows:

$$\begin{aligned} \boldsymbol{Q}_t &= \mathrm{FFT}\left(\boldsymbol{X}_{\mathrm{de}}^{i}\right) \\ \boldsymbol{K}_t &= \mathrm{FFT}\left(\boldsymbol{W}_K \boldsymbol{X}_{\mathrm{de}}^{i}\right) \\ \boldsymbol{V}_t &= \mathrm{FFT}\left(\boldsymbol{W}_V \boldsymbol{X}_{\mathrm{de}}^{i}\right) \\ \mathrm{FreqAtten}_{\mathrm{time}}\left(\boldsymbol{Q}_t, \boldsymbol{K}_t, \boldsymbol{V}_t\right) &= \mathrm{Softmax}_{\mathrm{time}}\left(\frac{\boldsymbol{Q}_t \boldsymbol{K}_t^T}{\sqrt{d_k}}\right) \boldsymbol{V}_t \end{aligned} \tag{4}$$

where $\boldsymbol{X}_{\mathrm{de}}^{i} \in \mathbb{R}^{T \times C}$ denotes the input of the ith decoder layer. $\boldsymbol{Q}_t$, $\boldsymbol{K}_t$, and $\boldsymbol{V}_t$ represent the attention matrices in the frequency domain across the temporal dimension. Unlike frequency cross-channel attention emphasizing dependencies across channels, the frequency cross-temporal attention mechanism applies softmax along the temporal dimension using $\mathrm{Softmax}_{\mathrm{time}}(\cdot)$. This attention mechanism operates over different timesteps, enabling the model to capture long-range dependencies and dynamic patterns across various time steps.

2) Adaptive High-Frequency Mask: To further enhance the model's ability to focus on significant frequency components while reducing noise, we propose an adaptive filtering mechanism inspired by the work [27]. First, we compute the power spectrum of $\boldsymbol{X}_{\mathrm{FFT}}$, representing the strength of each time-series frequency

$$\boldsymbol{X}_{\mathrm{SP}} = \left|\boldsymbol{X}_{\mathrm{FFT}}\right|^2. \tag{5}$$

The spectrum is then normalized and compared to a learnable threshold θ, which is automatically optimized during training by backpropagation ($\partial \mathrm{Loss}/\partial\theta$). This comparison generates a binary mask, where each element is either 0 or 1. Frequencies with a spectrum below the threshold θ are filtered out to reduce noise interference. The filtered frequency components are then enhanced using an adaptive learnable weight. This process is formulated as follows:

$$\begin{aligned} \boldsymbol{X}_{\mathrm{filtered}} &= \boldsymbol{X}_{\mathrm{FFT}} \odot \mathrm{Mask}\left(\boldsymbol{X}_{\mathrm{SP}} < \theta\right) \\ \boldsymbol{X}_{\mathrm{enhanced}} &= \boldsymbol{X}_{\mathrm{filtered}} \odot \boldsymbol{W}_{\mathrm{ada}} \end{aligned} \tag{6}$$

where $\odot$ denotes elementwise multiplication. $\boldsymbol{W}_{\mathrm{ada}}$ denotes an adaptive learnable weight enhancing the significant frequency components selectively.

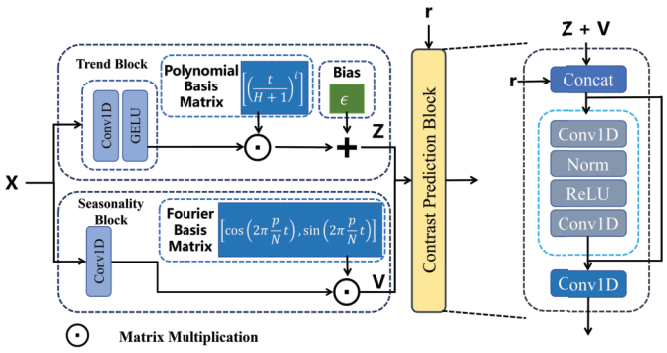

Fig. 3. Illustration of contrastive synthesis layer.

3) Frequency Inversion: The enhanced frequency components $\boldsymbol{X}_{\mathrm{enhanced}}$ along with the frequency attention vector $\boldsymbol{X}_{\mathrm{att}}^{i}$ obtained from the frequency cross-temporal attention mechanism, are then transformed back to the time domain using IFFT

$$\begin{aligned} \boldsymbol{X}_{\mathrm{time}}^{i} &= \mathrm{IFFT}\left(\boldsymbol{X}_{\mathrm{att}}^{i}\right) \\ \boldsymbol{X}_{\mathrm{de}}^{i+1} &= \mathrm{UpSample}\left(\boldsymbol{X}_{\mathrm{time}}^{i}\right). \end{aligned} \tag{7}$$

E. Contrastive Synthesis Layer

After learning from the previous frequency learners, the temporal dynamic details and cross-channel dependencies of the time series have been well modeled. To generate diverse industrial time series, we proposed a contrastive synthesis layer, consisting of three components: the trend synthesis block, the seasonality synthesis block, and the contrast prediction block, as shown in Fig. 3. The trend block utilized polynomial curve fitting to synthesize slow-changing long-term trend information, such as performance degradation over the device's life cycle. The seasonality block uses Fourier series expansion to synthesize short-term seasonal trends, such as changes in environmental factors such as pressure, temperature, and altitude. Finally, we develop a contrast prediction block, which connects with the residuals of the original noise signal. By modeling the differences between the synthesized and original sequences, this block generates a more diverse time series.

1) Trend Synthesis Block: In industrial time-series modeling, the trend synthesis is essential for capturing long-term, smooth variations that occur due to gradual changes in system behavior, such as equipment aging or evolving operational conditions. If these trends are not effectively modeled, the generated time series may lack structural coherence, leading to unrealistic fluctuations or inaccurate long-term predictions. To address this, we adopt a polynomial regression-based approach, which provides a structured yet flexible representation of long-term trends. Unlike linear models that assume a constant rate of change, polynomial regression can capture nonlinear but smooth variations without overfitting to short-term fluctuations. This makes it particularly effective for trend modeling, where the objective is to extract gradual and persistent patterns rather than transient variations. Mathematically, the trend synthesis block follows a polynomial regression

framework, where the input feature x is expanded into a polynomial basis matrix P, and a learned feature representation Z is used instead of fixed regression coefficients to weight the polynomial bases

$$y = Z \cdot P + \epsilon \tag{8}$$

where

$$P_i = \left[\frac{1}{H+1} i, \frac{2}{H+1} i, \ldots, \frac{H}{H+1} i \right] \tag{9}$$

$$Z = \mathrm{GELU}\left(\mathrm{Conv1d}\left(X\right)\right). \tag{10}$$

Here, the convolutional operation extracts localized features from the input sequence, while the polynomial basis matrix ensures that the learned representation maintains a smooth and structured temporal progression. The normalization of time steps prevents numerical instability and ensures consistent trend modeling across different scales. By explicitly modeling trends in this manner, the trend synthesis block enhances the overall time-series generation process. It ensures that long-term variations are effectively separated from short-term fluctuations, allowing the model to better capture seasonal patterns and noise components in subsequent layers. Moreover, by incorporating a learned feature representation Z, the model dynamically adjusts to different trend structures, improving its ability to generalize across diverse industrial time-series datasets.

2) Seasonality Synthesis Block: In time-series analysis, seasonality refers to recurring periodic patterns in the data. To model these seasonal patterns, we use a Fourier series modeling approach to decompose a complex time series into a series of simple sine and cosine functions. The seasonality synthesis block models in time-series data through Fourier series. Its basic structure consists of a convolutional layer and a Fourier transform layer.

The convolutional layer extracts the feature matrix $\boldsymbol{S}$ from the input time-series data by performing a 1-D convolutional operation on the input data $\boldsymbol{X}$. In order to transform the output $\boldsymbol{S}$ of the convolutional layer as weighting coefficients into the Fourier space, we define the Fourier basis functions as follows:

$$\boldsymbol{S} = \mathrm{Conv1d}(\boldsymbol{X}) \tag{11}$$

$$\boldsymbol{F}_{\mathbf{cos}} = \left[\cos\left(\frac{2\pi p}{\mathrm{out\_dim}} t \right) \middle| \, p \in [1, P_1] \right] \tag{12}$$

$$\boldsymbol{F}_{\mathbf{sin}} = \left[\sin\left(\frac{2\pi p}{\mathrm{out\_dim}} t \right) \middle| \, p \in [1, P_2] \right] \tag{13}$$

$$\boldsymbol{F} = [\boldsymbol{F}_{\cos}, \boldsymbol{F}_{\sin}] \tag{14}$$

$$\boldsymbol{V}_{\mathrm{seas}} = \boldsymbol{S} \cdot \boldsymbol{F} \tag{15}$$

where P_1 and P_2 denote the number of sine and cosine basis functions. The Fourier basis matrix F is obtained by combining the sine basis function $F_{\sin}$ and cosine basis function $F_{\cos}$. Here, $F_{\cos}$ and $F_{\sin}$ serve as the fundamental components for constructing periodic functions in Fourier space. They provide a complete basis for representing periodic variations in the time series, allowing the model to capture both symmetric and asymmetric seasonal patterns. By projecting the feature matrix S onto these basis functions, the model learns how different frequency components contribute to the overall seasonal structure of the data. The feature matrix $\boldsymbol{S}$ output from the convolutional layer is multiplied with the Fourier basis matrix $\boldsymbol{F}$ as the weighting coefficients of different sine and cosine basis functions to obtain the periodic function $\boldsymbol{V}_{\mathbf{seas}}$ which is similar to $\sum_{n=1}^{\infty}(a_n \cos(2\pi nt/T) + b_n \sin(2\pi nt/T))$. This achieves the following modeling of the seasonal component in the time-series data. The model calculates the loss and propagates it backward by comparing the predicted seasonal component with the true seasonal component, and the feature matrix $\boldsymbol{S}$ gradually refines the periodicity of the input data by continuously adjusting the convolutional kernel weights and bias terms, optimizing the modeling of seasonal components.

3) Contrast Prediction Block: In the original diffusion model, the primary task is to reconstruct the original time series by predicting the added noise. However, the trend synthesis block and seasonality synthesis block do not explicitly capture the noise characteristics inherent in the original signal. To address this limitation, we introduce a contrast prediction block to enhance the model's ability to predict noise more effectively. Our approach begins by aggregating the trend block features Z and the seasonality synthesis block features V_{seas}. These synthesized features are then concatenated with the original input signal r, forming a composite feature representation that allows the model to directly compare the synthesized time series with the original series

$$x_{\mathrm{cat}} = \mathrm{concat}\left(Z + V_{\mathrm{seas}}, r\right) \tag{16}$$

where concat$(\cdot)$ represents the concatenation of tensors along a specific dimension. This spliced feature representation is subsequently processed through a residual block that consists of multilayer convolutional operations and nonlinear activation functions, facilitating deep feature extraction and transformation. Specifically, the noise component is extracted as follows:

$$\epsilon_{\mathrm{noise}} = \mathrm{ReLU}\left(W \cdot x_{\mathrm{cat}}\right) \tag{17}$$

where W represents the learnable parameters of the convolutional layers and rectified linear unit (ReLU) is the activation function that introduces nonlinearity into the transformation. By zeroing out negative values, ReLU prevents the model from interpreting weak or canceling-out fluctuations as significant noise, allowing it to focus on dominant positive deviations that contribute to overall noise patterns. This ensures that only the most relevant noise features are retained while filtering out minor perturbations. The convolutional processing ensures that the extracted features are effectively transformed into accurate noise predictions while maintaining the dimensional consistency between the input and output data. This contrast-based mechanism enables the model to discern fine-grained differences between the synthesized and original time series, thereby improving its noise estimation capability and overall reconstruction performance.

IV. Experiments

A. Experimental Setup

1) Dataset: In order to generate industrial time series for different operating conditions and components, we collected a large dataset of over ten million sampling points, including

TABLE I

COMPARISON RESULTS OF THE PROPOSED METHOD WITH OTHER METHODS IN TERMS OF DISCRIMINATION SCORES AND PREDICTION SCORES ON THE CMAPSS DATASET. THE BEST RESULTS ARE INDICATED IN BOLD, WHILE THE SECOND BEST OUTCOMES ARE DENOTED BY UNDERLINING

Methods	TTS-GAN [30]	TimeGAN [31]	Tabddpm [32]	SSSD [19]	DiffWave [6]	Dit [33]	MetaIndux-TS
Discriminative Score (Fidelity)							
FD001-24	0.498	0.464	0.305	0.368	0.328	0.188	**0.137**
FD001-48	0.495	0.419	0.389	0.411	0.414	0.358	**0.102**
FD001-96	0.487	0.435	0.311	0.450	0.404	0.402	**0.140**
FD002-24	0.494	0.457	0.302	0.343	0.357	**0.065**	0.280
FD002-48	0.491	0.436	0.296	0.231	0.176	0.268	**0.109**
FD002-96	0.498	0.441	0.313	0.285	0.301	0.449	**0.008**
FD003-24	0.498	0.420	0.292	0.305	0.341	0.168	**0.112**
FD003-48	0.498	0.483	0.398	0.417	0.431	0.333	**0.114**
FD003-96	0.483	0.451	0.305	0.470	0.425	0.379	**0.094**
FD004-24	0.489	0.417	0.307	0.362	0.343	**0.060**	0.085
FD004-48	0.491	0.465	0.309	**0.219**	0.284	0.358	0.239
FD004-96	0.498	0.446	0.300	0.318	0.329	0.345	**0.112**
Average	0.494	0.444	0.319	0.358	0.374	0.301	**0.128**(57.5%↓)
Predictive Score							
FD001-24	85.581	70.536	28.539	19.864	20.431	21.062	**14.697**
FD001-48	94.107	41.497	30.636	23.100	18.085	23.777	**13.949**
FD001-96	85.291	37.933	36.088	17.364	21.774	27.323	**15.191**
FD002-24	87.154	41.230	30.922	27.724	26.950	**23.360**	25.822
FD002-48	95.273	46.780	29.837	25.431	27.184	26.740	**24.565**
FD002-96	87.211	41.132	32.956	27.628	30.122	27.194	**22.860**
FD003-24	72.454	85.800	25.165	**21.208**	21.552	23.511	21.420
FD003-48	86.811	53.928	26.274	25.959	23.749	40.200	**16.206**
FD003-96	72.354	85.170	28.994	22.946	28.433	24.719	**16.138**
FD004-24	86.216	51.086	34.557	43.531	34.382	29.665	**27.333**
FD004-48	94.756	86.609	33.604	39.764	32.793	30.654	**26.577**
FD004-96	84.803	50.805	42.764	41.186	32.515	30.106	**23.776**
Average	85.732	53.439	32.907	29.464	27.736	26.029	**20.711**(20.4%↓)

turbofan engines and bearings in different degradation states. The turbofan engine dataset (CMAPSS) [28] is widely used for aeroengine remaining useful life (RUL) prediction and health management studies. The dataset consists of four subsets: FD001–FD004, each of which covers different failure modes and operating conditions. The dataset records operational settings, sensor readings, and unit settings, where operational settings relate to flight altitude and speed. Sensor readings provide time-series data from 21 sensors, reflecting engine performance and health. In addition, the FEMTO dataset [29] uses multiple sensors to collect bearing wear data with different sampling frequencies. The sampling frequency is 25.6 kHz. The dataset reflects the actual condition of the bearings that degrade at an accelerated rate under three different conditions.

2) Implementation Details: In this study, we set the number of time steps for sampling during training to 1000. The noise schedule follows a linear design. Specifically, the noise intensity increases linearly from a minimum value of 0.0001 to a maximum value of 0.02. This is achieved by scaling the base range (0.0001 and 0.02) according to the ratio of the actual number of time steps to the default 1000 steps. During sampling, the model employs a reverse sampling strategy based on DDPM, and the number of sampling steps matches that used during training.

We implemented the proposed model using the PyTorch deep learning framework and trained it with the Adam optimizer, setting the initial learning rate to $2e^{-3}$. All experiments were repeated five times, with the average values computed for the final results. All the models were trained for a total of 70 epochs.

Following the original DDPM [22], we designed the encoder and decoder architecture similar to the 1-D U-Net model. This multiscale convolutional network architecture captures various frequency components in time-series data and leverages conditional information to enhance the generated quality. Both the encoder and decoder consist of two submodules for feature extraction, a residual connection module, and a sampling operation module. Each submodule contains a convolutional layer with the SiLU activation function. The downsampling operation in the encoder utilizes a 4×4 convolutional kernel with a stride of 2 to extract high-level feature information, while the upsampling operation in the decoder employs transposed convolution to restore the input data shape.

3) Evaluation Metrics:

1) *Discriminative Score (Fidelity):* This score is based on the accuracy of an RNN classifier distinguishing real from generated data. The classifier's accuracy reflects its ability to differentiate the two, with an accuracy near 0.5 suggesting poor distinction. The discriminative score is the absolute difference between the accuracy and 0.5. The lower the discriminative score, the more similar the generated data is to the real data, with a score closer to 0 indicating higher similarity.
2) *Predictive Score:* To evaluate the usability of the generated data, we train an LSTM model using 70% of the synthetic data for training and 30% for validation, with real data as the test set. The performance is assessed

TABLE II
EVALUATION OF METAINDUX-TS AGAINST OTHER METHODS ON FEMTO DATASETS

Dataset–Metric	FEMTO–Predictive.			FEMTO–Discriminative.		
Seq length	80	160	320	80	160	320
TTS-GAN [30]	0.862	0.975	0.442	0.500	0.489	0.491
TimeGAN [31]	0.802	0.882	0.389	0.490	0.479	0.482
GAN-LSTM [15]	0.728	0.829	0.427	0.486	0.498	0.483
DiffWave [6]	0.213	0.204	0.239	0.364	0.241	0.241
MetaIndux-TS	**0.115**	**0.201**	**0.094**	**0.115**	**0.183**	**0.155**

using the root mean square error (RMSE) between predicted and true values. A lower RMSE indicates better data quality and usability for downstream tasks.

B. Comparison With the State-of-the-Art Methods

To demonstrate the significant advantages of our proposed method, we conduct experiments on industrial datasets and compare it with several state-of-the-art models. These compared models include several GAN-based methods (GAN-LSTM [15], TTS-GAN [30], and TimeGAN [31]) as well as several diffusion model-based methods (Dit [33], SSSD [19], DiffWave [6], and Tabddpm [32]). During the experiments, we evaluate the performance of each model under different generation conditions by adjusting the length of the input data and the corresponding length of the generated samples. We set three length combinations of 24, 48, and 96. From the results, it can be seen that the MetaIndux-TS model outperforms several current state-of-the-art models in most of the evaluated metrics in datasets of specific lengths.

As shown in Table I, MetaIndux-TS's discriminative scores outperform other diffusion and GAN models on most datasets. In FD001, MetaIndux-TS's discriminative score is 0.102, which is a 71.5% improvement over the best discriminative score of 0.358 of the remaining models. In FD003, the discriminative score of MetaIndux-TS is 0.094, which is 69.2% higher than the optimal result of 0.305 in the rest of the models. These results indicate that MetaIndux-TS is able to more accurately model the real data distribution and generate samples that are more similar. It is worth noting that most of the discriminative scores of the GAN-based generative models in the experiments of generating time-series data are higher than 0.4. This is due to the fact that GAN models are subject to pattern crashes and training instability, which lead to large differences between the generated samples and the real data. In contrast, the MetaIndux-TS model stably captures the complex dynamic features of the time series through a step-by-step denoising process. It overcomes the shortcomings of the GAN model and generates time series that are more accurate and more consistent with the real data.

From the perspective of the predictive score, MetaIndux-TS also shows significant advantages in Table I. Compared with the optimal results of the rest of the models under the same conditions, MetaIndux-TS's predictive score in the FD001 dataset decreases from 17.364 to 15.191, a decrease of 12.5%; in FD002, the predictive score decreases from 27.194 to 22.860, a decrease of 15.9%; in FD003, it decreases from 22.946 to 16.138, a decrease of 29.0%; and in FD004, the predictive score decreases from 30.106 to 23.776, a decrease of 21.0%. This indicates that MetaIndux-TS is not only able to generate samples that are more similar to real data, but also provides more reliable and practical synthetic data in the prediction task.

TABLE III
ABLATION STUDY OF CONTRASTIVE SYNTHESIS LAYER

Dataset	Length	MetaIndux-TS (w/o contrastive synthesis layer)	MetaIndux-TS
		Predictive Score	
FD001	24	26.714	**14.697**
	48	29.333	**13.949**
	96	22.751	**15.191**
FD002	24	27.526	**25.822**
	48	**23.155**	24.565
	96	37.044	**22.860**
FD003	24	25.214	**21.420**
	48	27.861	**16.206**
	96	32.186	**16.138**
FD004	24	36.867	**27.333**
	48	34.144	**26.577**
	96	32.231	**23.776**

In addition, we conduct experiments on the FEMTO dataset in Table II. This dataset is divided into different lengths to test the quality of data generated by different generative models. From the experimental results, it is concluded that the quality of data generated by MetaIndux-TS under any length of the dataset is better than other models. This further proves the superiority of MetaIndux-TS in data generation.

In summary, MetaIndux-TS demonstrates more desirable results in both classification scores and prediction scores, proving its higher similarity and utility in generating time series that can better meet the needs of industrial data generation and analysis.

C. Ablation Study

1) Ablation Study of the Contrastive Synthesis Layer: The key module of the MetaIndux-TS model is the contrastive synthesis layer for predictive noise. In order to verify its validity and contribution to the overall model, we modify the MetaIndux-TS model by using MetaIndux-TS (w/o contrastive synthesis layer) to denote the model with the contrastive synthesis layer removed to perform ablation experiments and derive the discriminative and predictive scores. From the histograms of discriminative scores in Fig. 4, the quality of the data generated by the original model is significantly higher than that of the model with the contrastive synthesis layer removed for any length of any dataset. In addition, predictive scores are listed in Table III.

For the same dataset and sequence length, the discriminative score of the MetaIndux-TS model improves by 66.1% (0.301 → 0.102) relative to the MetaIndux-TS without contrastive synthesis layer. The predictive score improves 52.4% (29.333 → 13.949) relative to MetaIndux-TS without contrastive synthesis layer. This indicates that the contrastive synthesis layer contributes significantly to the MetaIndux-TS model. The data generated with the inclusion of the contrastive synthesis layer are more similar and usable.

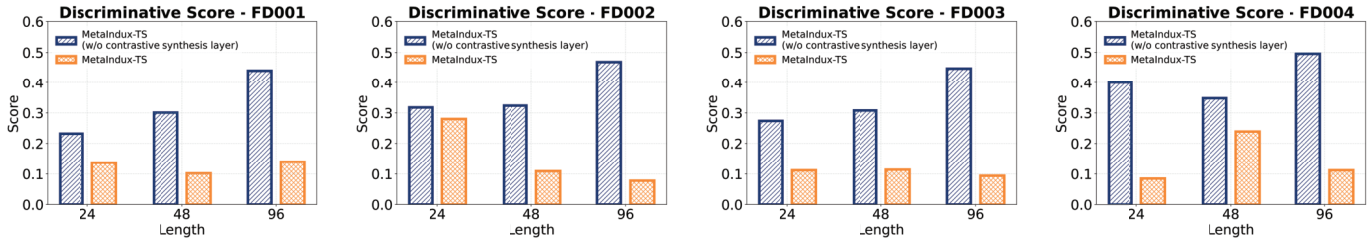

Fig. 4. Impact of the contrastive synthesis layer on turbofan engine datasets from FD001 to FD004.

TABLE IV

ABLATION STUDY OF FREQUENCY CROSS-CHANNEL LEARNER AND FREQUENCY CROSS-TEMPORAL LEARNER

Dataset	FD001			FD002			FD003			FD004			Average
Length	24	48	96	24	48	96	24	48	96	24	48	96	
Discriminative Score (Fidelity)													
MetaIndux-TS (w/o frequency learners)	0.111	**0.086**	0.086	**0.099**	0.216	0.078	**0.016**	0.367	0.374	0.073	0.348	0.396	0.188
MetaIndux-TS$_{time}$	0.103	0.091	**0.077**	0.196	0.116	0.382	0.089	0.115	0.115	0.069	0.275	0.291	0.160
MetaIndux-TS$_{chan}$	**0.100**	0.112	0.410	0.161	0.147	0.108	0.101	0.082	**0.075**	**0.055**	0.347	0.462	0.180
MetaIndux-TS	0.137	0.102	0.140	0.280	**0.109**	**0.008**	0.112	0.114	0.094	0.085	**0.239**	**0.112**	**0.128**
Predictive Score													
MetaIndux-TS (w/o frequency learners)	16.072	**13.470**	16.191	**21.885**	28.629	23.604	21.759	37.721	32.260	27.589	33.328	43.686	26.350
MetaIndux-TS$_{time}$	15.594	14.079	15.386	28.553	**24.224**	24.705	21.783	16.616	17.094	27.338	28.191	28.005	21.797
MetaIndux-TS$_{chan}$	16.148	13.640	26.556	25.794	28.919	24.231	22.470	**15.912**	**15.878**	27.801	33.956	38.389	24.141
MetaIndux-TS	**14.697**	13.949	**15.191**	25.822	24.565	**22.860**	**21.420**	16.206	16.138	**27.333**	**26.577**	**23.776**	**20.711**

TABLE V

PERFORMANCE ON ZERO-SHOT LEARNING TASKS OF DIFFERENT DIFFUSION MODELS

Methods	DiffWave	DiT	Tabddpm	MetaIndux-TS
RMSE				
FD001	20.545	58.284	33.943	**6.353**
FD002	30.683	32.555	23.200	**22.574**
FD003	70.927	67.121	14.901	**11.289**
FD004	47.995	57.097	38.562	**35.062**
Average	42.538	53.764	27.652	**18.820**

TABLE VI

PERFORMANCE ON FEW-SHOT LEARNING TASKS OF DIFFERENT DIFFUSION MODELS

Methods	DiffWave	DiT	Tabddpm	MetaIndux-TS
RMSE				
FD001	15.960	53.224	22.533	**5.219**
FD002	40.891	29.938	27.095	**24.758**
FD003	69.854	28.766	13.568	**9.244**
FD004	49.339	44.747	42.809	**24.816**
Average	44.011	39.169	26.501	**16.009**

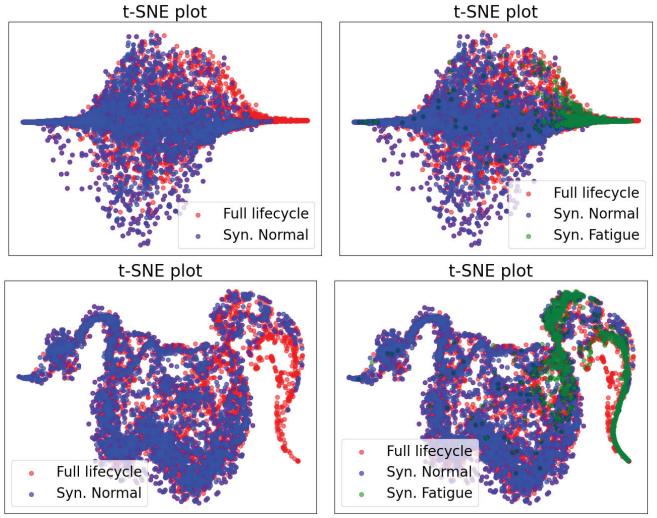

Fig. 5. t-SNE visualization of zero-shot generation results of MetaIndux-TS. The red points represent the original full lifecycle samples, while the blue and green points correspond to the generated normal and fatigue-condition samples, respectively.

2) Ablation Study of the Dual Frequency Learners: Another feature of the MetaIndux-TS model is the introduction of the cross-channel attention mechanism and the cross-temporal attention mechanism, for which we conduct a second type of ablation experiments, respectively, for the model MetaIndux-TS (w/o contrastive synthesis layer) without all frequency learners, the model MetaIndux-TS$_{chan}$ with only the frequency cross-temporal attention mechanism removed, the model MetaIndux-TS$_{time}$ with only the frequency cross-channel attention mechanism removed, and the proposed model MetaIndux-TS are evaluated on the generated data of the four models. The discriminative scores and predictive scores of the four models are reported. From Table IV, it is seen that the introduction of both types of attention mechanisms improves the quality of synthetic data generated by the models. In datasets with complex temporal logic such as FD004, the discriminative scores of MetaIndux-TS$_{time}$ and MetaIndux-TS$_{chan}$ are improved by 20.9% (0.348 → 0.275) and 0.3% (0.348 → 0.347), respectively, in comparison with MetaIndux-TS without all frequency learners. The predictive scores improved by 35.9% (43.686 → 28.005) and 12.1% (43.686 → 38.389). The improvement in the discriminative and predictive scores is due to the fact that the frequency cross-channel attention mechanism is able to effectively identify and capture the correlation between high-frequency and low-frequency components. Also, the frequency cross-temporal

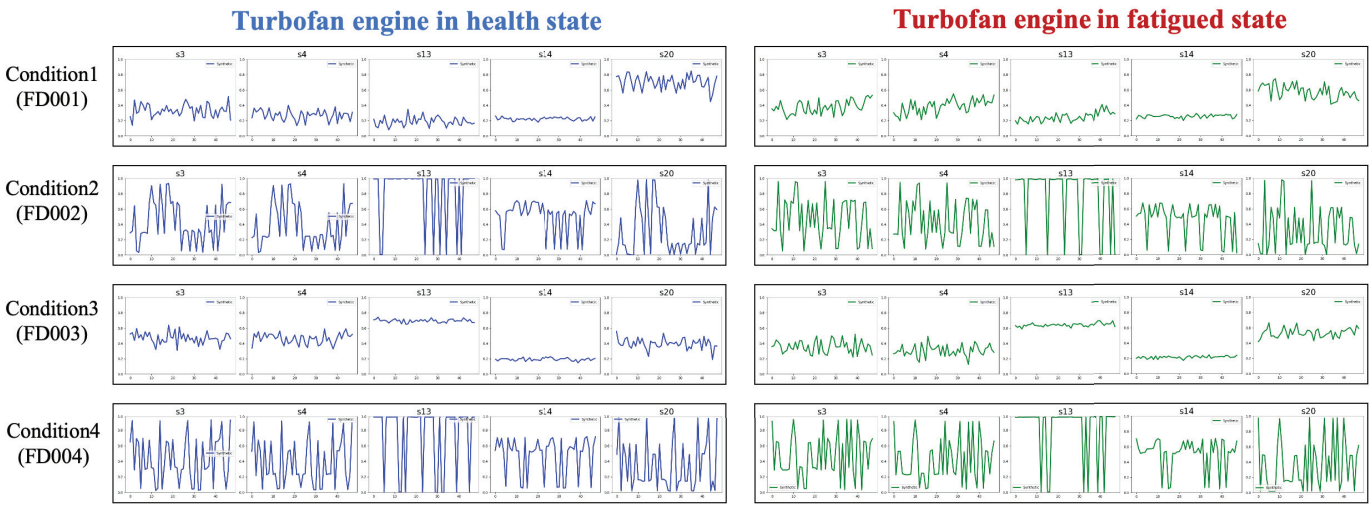

Fig. 6. MetaIndux-TS generates some turbofan engine signals under different conditions and states. The left column represents the engine signals in a healthy state (blue), while the right column represents the engine signals in a fatigued state (green).

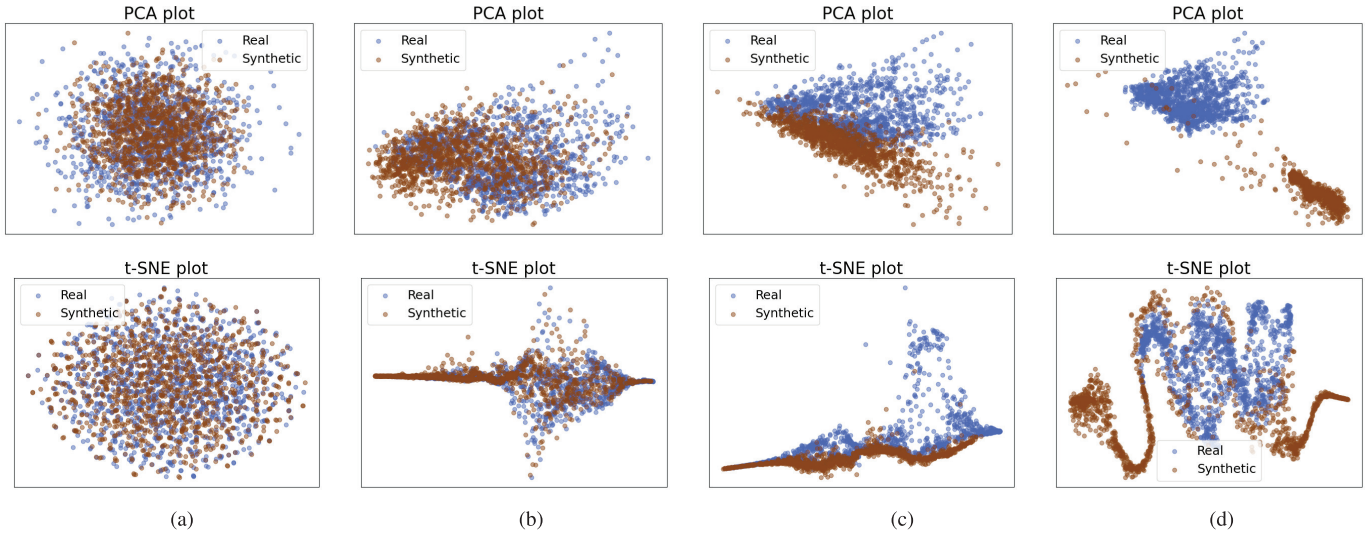

Fig. 7. Visualizations of the generated samples are presented using PCA (first row) and t-SNE (second row) techniques. Each column displays the comparative visualizations for the four methods. Original samples are denoted in blue, while synthetic samples are indicated in brown. The degree of overlap between the two types of points is indicative of their similarity, with greater overlap signifying a higher degree of resemblance. (a) Diff-MTS. (b) DiffWave. (c) DiT. (d) TimeGAN.

attention mechanism can capture the dynamic changes of frequency components at different time points. Both of them can make the generated data closer to the frequency characteristics of the real data.

D. Zero-Shot and Few-Shot Generation

To assess the adaptability of MetaIndux-TS with unseen samples, we conduct zero-shot and few-shot experiments. In the C-MAPSS dataset, samples with an RUL greater than 30 are categorized as normal conditions, while those less than 30 fall under fatigue conditions. Generative models are trained in two settings: zero-shot, using only normal condition data, and few-shot, incorporating 10% fatigue condition data alongside normal data. For evaluation, the generated datasets from these models are used to train a two-layer LSTM model, which is then tested exclusively on real fatigue condition data.

The results are shown in Tables V and VI, which show that MetaIndux-TS consistently outperforms other models in both zero-shot and few-shot settings. When using MetaIndux-TS-generated data under the zero-shot setting, the predicted RMSE is reduced by an average of 31.9% (27.652 → 18.820) compared to the best-performing alternative models across the FD001–FD004 datasets. In the few-shot setting, the reduction reaches 39.6% (26.501 → 16.009). In Fig. 5, we present the t-SNE visualization of the zero-shot data generation results. Notably, although the model is never exposed to fatigue condition samples during training, MetaIndux-TS successfully generates realistic fatigue condition data that aligns with the overall data distribution. This demonstrates MetaIndux-TS's ability to generalize the underlying distribution of unseen data, making it a promising solution for enhancing predictive accuracy in real-world industrial scenarios where data collection

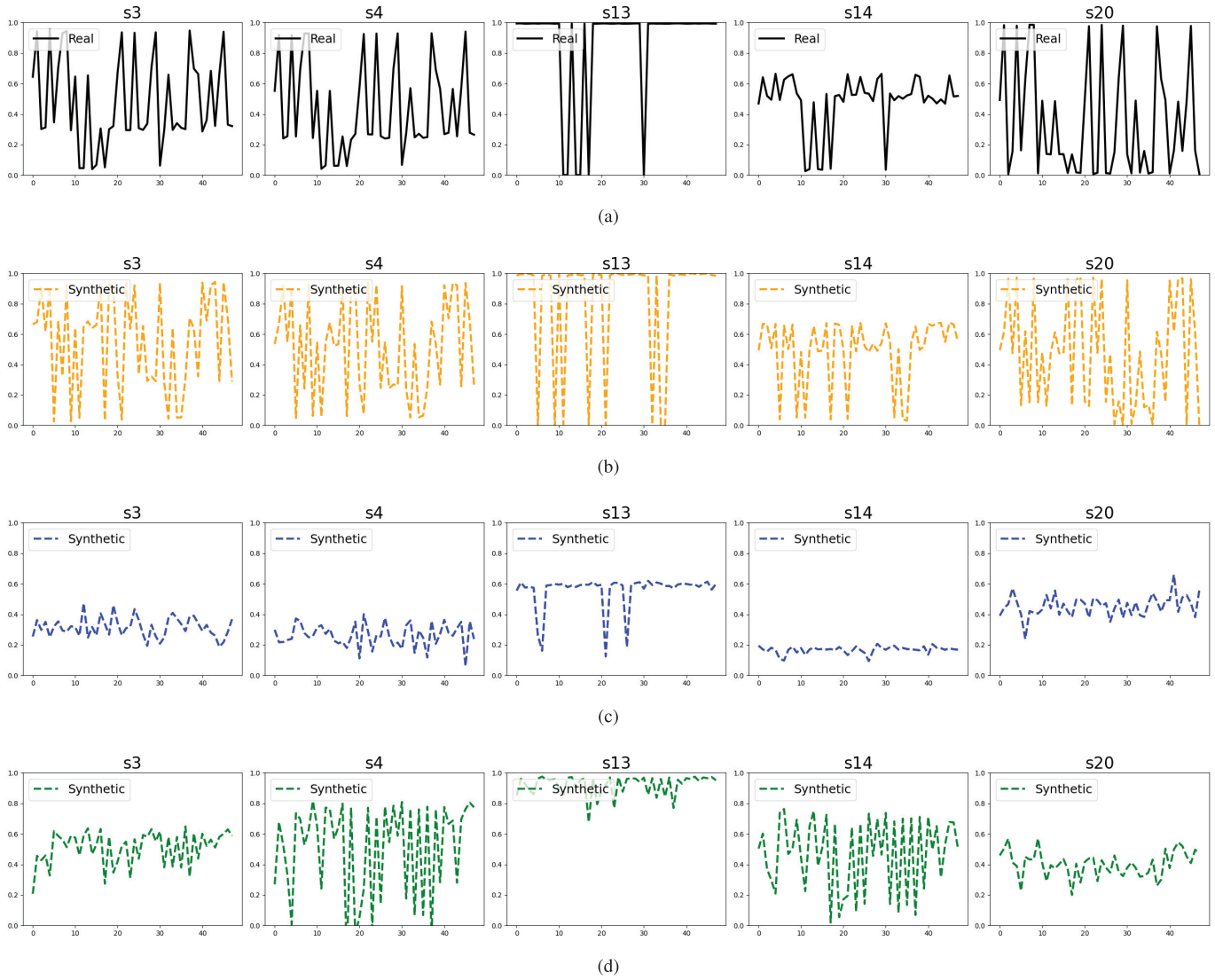

Fig. 8. Multiple visualizations of prediction results. The original time-series data are depicted by black curves. The synthetic time series are illustrated by distinct colors: gold for Diff-MTS, blue for DiffWave, and green for TimeGAN. (a) Original data. (b) MetaIndux-TS. (c) DiffWave. (d) TimeGAN.

under extreme conditions (high pressure, low temperature, or elevated heat) is challenging.

E. Visualization Analysis

MetaIndux-TS is capable of generating a variety of industrial time series for different operating conditions and health states of industrial equipment. Fig. 6 shows the synthesized data for various operating conditions and different health states. In Fig. 6, the blue plots represent signals from turbofan engines in a healthy state, while the green plots depict signals from fatigued engines. Each row corresponds to a different condition (FD001–FD004), capturing diverse operational scenarios. Furthermore, this figure demonstrates the capability of the MetaIndux-TS model to generate realistic industrial time series.

To further qualitatively analyze the degree of similarity between the generated data and the original data, we plot PCA and t-SNE plots of the generated data and the original data under different models. The blue dots on each plot represent the real data, while the brown dots represent the synthetic data. By looking at the distribution of the dots in the two colors, we can intuitively determine the degree of similarity between the synthetic data and the real data. As shown in Fig. 7, the data generated by DiT and TimeGAN models have less overlap with the real data distribution. The similarity between the generated data and the real data is not high. The generated data of the DiffWave model are slightly different from the real data on the right side of the PCA plot and the lower side of the t-SNE plot, where the distribution of the real data does not account for a high proportion. In contrast, the distribution of the generated data of the MetaIndux-TS model has a higher overlap with the real data in both the PCA plot and the t-SNE plot, indicating that the generated data are more similar. In addition, in order to observe the generated data and the original real data more intuitively, we select all the data in one of the time steps. The generated data and original data are plotted for each dimension. From Fig. 8, it is seen that in the signal of sensor s13, the data generated by the TimeGAN model does not generate the data in the lower part

TABLE VII
SAMPLING TIME COMPARISON ACROSS DIFFERENT METHODS

Architecture	Sampling	FD001	FD002	FD003	FD004
Original UNet +attention	DDPM	330.372s	1094.981s	557.183s	1465.010s
MetaIndux-TS	DDPM	192.523s	808.291s	514.192s	1416.325s
MetaIndux-TS (proposed)	DDIM	20.119s	49.271s	25.141s	59.292s
Speedup	—	16.42×	22.22×	22.16×	24.71×

of the image. In the signal of sensor s14, the data generated by the DiffWave model oscillates much less than the real data. Also, the generated data of the MetaIndux-TS model all well reproduces the timing pattern of the real data with much higher similarity.

F. Efficiency Analysis

To evaluate the sampling efficiency of different architectures, we conducted comprehensive experiments under multiple settings. Specifically, we compared three structures as shown in Table VII: 1) "Original U-Net + attention," which represents a baseline diffusion model using DDPM sampling; 2) "MetaIndux-TS," which also employs DDPM sampling but with the proposed frequency architecture for time series; and 3) "MetaIndux-TS (proposed)," which utilizes DDIM sampling [23] with $t = 100$ steps for acceleration. Here, DDPM and DDIM refer to different diffusion sampling strategies, where DDIM typically enables faster generation with fewer steps. The experiments are performed across four different datasets to validate the generality of the proposed improvements.

As shown in Table VII, the proposed MetaIndux-TS with DDIM achieves significant acceleration compared to both the Original U-Net + attention and the MetaIndux-TS with DDPM sampling. Specifically, on the FD001 dataset, our method reduces the sampling time from 330.372 to 20.119 s, achieving a 16.42× speedup. Similar improvements are observed across other datasets, with speedups of 22.22×, 22.16×, and 24.71× on FD002–FD004, respectively. Moreover, even under the same DDPM sampling strategy, MetaIndux-TS outperforms the Original U-Net due to its compression of redundant high- and low-frequency components, effectively concentrating useful information and reducing redundancy in time-series features. These results demonstrate that our method substantially improves sampling efficiency while maintaining strong generative performance.

V. CONCLUSION

This article introduces MetaIndux-TS, a frequency-informed AIGC foundation model specifically designed for industrial time-series generation. Addressing the challenges posed by complex temporal dynamics, multichannel intercolumn correlations, and diverse frequency variables in industrial time series, MetaIndux-TS effectively incorporates a frequency cross-channel learner to model multichannel dependencies and a frequency cross-temporal learner to capture detailed temporal dynamics.

However, MetaIndux-TS has limitations in generating time series that strictly adhere to the physical laws governing industrial equipment. In the future, we plan to explore methods for integrating physical constraints into generative models. This approach aims to ensure that the generated time series align with the physical laws of industrial equipment, making them more applicable to real-world physical scenarios.

REFERENCES

[1] L. Ren et al., "Industrial foundation model," *IEEE Trans. Cybern.*, vol. 55, no. 6, pp. 1234–1245, Jun. 2025.

[2] L. Ren, H. Wang, Y. Wang, K. Huang, L. Wang, and B. Li, "Foundation models for the process industry: Challenges and opportunities," *Engineering*, vol. 11, pp. 423–432, Mar. 2025.

[3] S. Khanna et al., "DiffusionSat: A generative foundation model for satellite imagery," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2023.

[4] Z. Qiu, Y. Tao, S. Pan, and A. W.-C. Liew, "Knowledge graphs and pretrained language models enhanced representation learning for conversational recommender systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 4, pp. 6107–6121, Apr. 2025.

[5] Y. Ling, F. Cai, J. Liu, H. Chen, and M. de Rijke, "Keep and select: Improving hierarchical context modeling for multi-turn response generation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 7, pp. 3636–3649, Jul. 2023.

[6] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "DiffWave: A versatile diffusion model for audio synthesis," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2020.

[7] C. Li et al., "Multimodal foundation models: From specialists to general-purpose assistants," *Found. Trends Comput. Graph. Vis.*, vol. 16, nos. 1–2, pp. 1–214, 2024.

[8] Y. Pi, Y. Wu, Y. Huang, Y. Shi, and S. Wang, "Inhomogeneous diffusion-induced network for multiview semi-supervised classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 5, pp. 8606–8618, May 2025.

[9] Z. Zhang, H. Weng, T. Zhang, and C. Chen, "A broad generative network for two-stage image outpainting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 9, pp. 12731–12745, Sep. 2024.

[10] J. Bruce et al., "Genie: Generative interactive environments," in *Proc. 41st Int. Conf. Mach. Learn.*, Feb. 2024, pp. 4603–4623.

[11] L. Ren, H. Wang, and G. Huang, "DLformer: A dynamic length transformer-based network for efficient feature representation in remaining useful life prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 5, pp. 5942–5952, May 2024.

[12] L. Ren, H. Wang, J. Li, Y. Tang, and C. Yang, "AIGC for industrial time series: From deep generative models to large generative models," 2024, *arXiv:2407.11480*.

[13] Z. Pan, Y. Wang, Y. Cao, and W. Gui, "VAE-based interpretable latent variable model for process monitoring," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 5, pp. 6075–6088, May 2024.

[14] S. Lin, R. Clark, R. Birke, S. Schönborn, N. Trigoni, and S. Roberts, "Anomaly detection for time series using VAE-LSTM hybrid model," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 4322–4326.

[15] B.-L. Lu, Z.-H. Liu, H.-L. Wei, L. Chen, H. Zhang, and X.-H. Li, "A deep adversarial learning prognostics model for remaining useful life prediction of rolling bearing," *IEEE Trans. Artif. Intell.*, vol. 2, no. 4, pp. 329–340, Aug. 2021.

[16] X. Zhang, Y. Qin, C. Yuen, L. Jayasinghe, and X. Liu, "Time-series regeneration with convolutional recurrent generative adversarial network for remaining useful life estimation," *IEEE Trans. Ind. Informat.*, vol. 17, no. 10, pp. 6820–6831, Oct. 2021.

[17] L. Yan, Z. Pu, Z. Yang, and C. Li, "Bearing fault diagnosis based on diffusion model and one-class support vector machine," in *Proc. Prognostics Health Manage. Conf. (PHM)*, May 2023, pp. 307–311.

[18] C. Yang, T. Wang, and X. Yan, "DDMT: Denoising diffusion mask transformer models for multivariate time series anomaly detection," 2023, *arXiv:2310.08800*.

[19] J. M. L. Alcaraz and N. Strodthoff, "Diffusion-based time series imputation and forecasting with structured state space models," *Trans. Mach. Learn. Res.*, vol. 4, pp. 1–33, Jan. 2022.

[20] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 10684–10695.

[21] Y. Liu et al., "Sora: A review on background, technology, limitations, and opportunities of large vision models," 2024, *arXiv:2402.17177*.
[22] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. NIPS*, 2020, pp. 6840–6851.
[23] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," 2020, *arXiv:2010.02502*.
[24] H. Chen et al., "VideoCrafter1: Open diffusion models for high-quality video generation," 2023, *arXiv:2310.19512*.
[25] K. E. Wu et al., "Protein structure generation via folding diffusion," *Nature Commun.*, vol. 15, no. 1, p. 1059, Feb. 2024.
[26] J. Lovelace, V. Kishore, C. Wan, E. Shekhtman, and K. Q. Weinberger, "Latent diffusion for language generation," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2022, pp. 1–10.
[27] E. Eldele, M. Ragab, Z. Chen, M. Wu, and X. Li, "TSLANet: Rethinking transformers for time series representation learning," in *Proc. Int. Conf. Mach. Learn.*, Apr. 2024, pp. 1234–1245.
[28] A. Saxena, K. Goebel, D. Simon, and N. Eklund, "Damage propagation modeling for aircraft engine run-to-failure simulation," in *Proc. Int. Conf. Prognostics Health Manage.*, Oct. 2008, pp. 1–9.
[29] L. Ren, H. Wang, and Y. Laili, "Diff-MTS: Temporal-augmented conditional diffusion-based AIGC for industrial time series towards the large model era," 2024, *arXiv:2407.11501*.
[30] X. Li et al., "TTS-GAN: A transformer-based time-series generative adversarial network," in *Proc. Int. Conf. Artif. Intell. Med.* Cham, Switzerland: Springer, 2022, pp. 133–143.
[31] J. Yoon, D. Jarrett, and M. V. D. Schaar, "Time-series generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Sep. 2019, pp. 5508–5518.
[32] A. Kotelnikov, D. Baranchuk, I. Rubachev, and A. Babenko, "TabDDPM: Modelling tabular data with diffusion models," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2022, pp. 17564–17579.
[33] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 4195–4205.

Haiteng Wang (Graduate Student Member, IEEE) received the B.Eng. degree in automation engineering from Beihang University, Beijing, China, in 2021, where he is currently pursuing the Ph.D. degree with the School of Automation Science and Electrical Engineering.

His current research interests include time-series prediction, generative AI, and industrial AI applications.

Lei Ren (Senior Member, IEEE) received the Ph.D. degree in computer science from the Institute of Software, Chinese Academy of Sciences, Beijing, China, in 2009.

He is currently a Professor with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, also with Hangzhou International Innovation Institute, Beihang University, Hangzhou, China, and also with the State Key Laboratory of Intelligent Manufacturing System Technology, Beijing. His research interests include neural networks and deep learning, time-series analysis, and industrial AI applications.

Dr. Ren serves as an Associate Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE/ASME TRANSACTIONS ON MECHATRONICS, and other international journals.

Yikang Li (Member, IEEE) received the B.Eng. degree in automation engineering from Beihang University, Beijing, China, in 2024, where he is currently pursuing the M.S. degree with the School of Automation Science and Electrical Engineering.

His current research interests include time-series analysis and generative models.

Yuqing Wang (Graduate Student Member, IEEE) received the B.S. degree from the School of Mechanical Engineering and Automation, Beihang University, Beijing, China, in 2022, where is currently pursuing the Ph.D. degree in pattern recognition and intelligent systems with the School of Automation Science and Electrical Engineering.

His main research interests are industrial big data, robotic vision, and computer graphics.