SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

December 2026, Vol. 69, Iss. 12, 222203:1–222203:16
https://doi.org/10.1007/s11432-024-5009-0

Adapting to industrial demands: state-guided self-calibration conditional diffusion transformer for time series forecasting

Diju LIU[1,2], Yalin WANG[1,2], Chenliang LIU[1,2*], Chunhua YANG[1,2] & Weihua GUI[1,2]

[1]*School of Automation, Central South University, Changsha 410083, China*
[2]*National Engineering Research Centre of Advanced Energy Storage Materials, Central South University, Changsha 410083, China*

Received 27 October 2024/Revised 4 September 2025/Accepted 28 October 2025/Published online 23 November 2026

Abstract Time series forecasting is an essential cornerstone for optimizing and regulating industrial processes. However, the dynamic environment of industrial processes poses challenges in extracting multiscale temporal features and calibrating models, which limits the ability of existing methods to meet the adaptive demands of the industry. To tackle these challenges, this paper presents a novel dynamic adaptive forecasting framework known as the state-guided adaptive progressive self-calibration conditional diffusion transformer (SGSC-DiT). Specifically, it designs a multiscale transformer architecture for noise prediction, enabling comprehensive feature extraction and assisting in modeling conditional probabilities between historical samples and forecasting targets. Moreover, a state-guided progressive adaptive self-calibration strategy is proposed to calibrate online forecasting results without model retraining or fine-tuning, enhancing the adaptability to dynamic environmental changes at a minimal cost. Finally, the effectiveness and superiority of the proposed method are demonstrated through extensive simulation experiments on two real-world datasets.

Keywords industrial time series forecasting, state-guided adaptive progressive self-calibration, online adaptive self-calibration, multi-scale temporal feature extraction

Citation Liu D J, Wang Y L, Liu C L, et al. Adapting to industrial demands: state-guided self-calibration conditional diffusion transformer for time series forecasting. Sci China Inf Sci, 2026, 69(12): 222203, https://doi.org/10.1007/s11432-024-5009-0

1 Introduction

Forecasting industrial time series of key quality variables involves constructing a time-series forecasting model to infer the difficult-to-measure indicators from easily measurable process variables, such as temperature, pressure, and liquid level [1–5]. Such forecasting yields invaluable guidance for optimizing operational parameters, occupying a pivotal position in process control within process industries [6, 7]. There has been rapid advancement, notably in the realm of deep learning-based time series forecasting technology, in recent years [8, 9].

However, two key challenges remain, impeding the practical application of data-driven methods, as depicted in Figure 1. These include multiscale temporal feature extraction for industrial time series data and the absence of online self-calibration capability for the model in dynamic environments [10–12]. The first challenge refers to the fact that industrial time series modeling necessitates considering both long-range evolutionary patterns of historical samples and short-term fluctuations from nearby samples [13]. This arises from the continuous operation of the process industry, where historical samples exert a constant influence, while nearby samples reflect the current process dynamics, collectively portraying the law of process evolution [14]. The other challenge involves the lack of online adaptability to dynamic environments [15, 16]. When the model is applied online, it cannot adaptively calibrate results as the environment changes, leading to rapid degradation of the online application performance [17].

As for multiscale temporal feature extraction, existing dynamic modeling methods face fundamental limitations despite recent advances. While sophisticated approaches such as faster dynamic feature extractors have shown promise in capturing rapid process dynamics for industrial quality prediction [18], the core challenge of achieving comprehensive multiscale temporal modeling persists. Convolutional neural networks (CNNs) [19], with their limited receptive field size, tend to emphasize extracting local short-term dynamics [20] but struggle with long-range temporal dependencies. Even with receptive field superposition techniques, capturing comprehensive long-range

* Corresponding author (email: lcliang@csu.edu.cn)

© Science China Press 2026

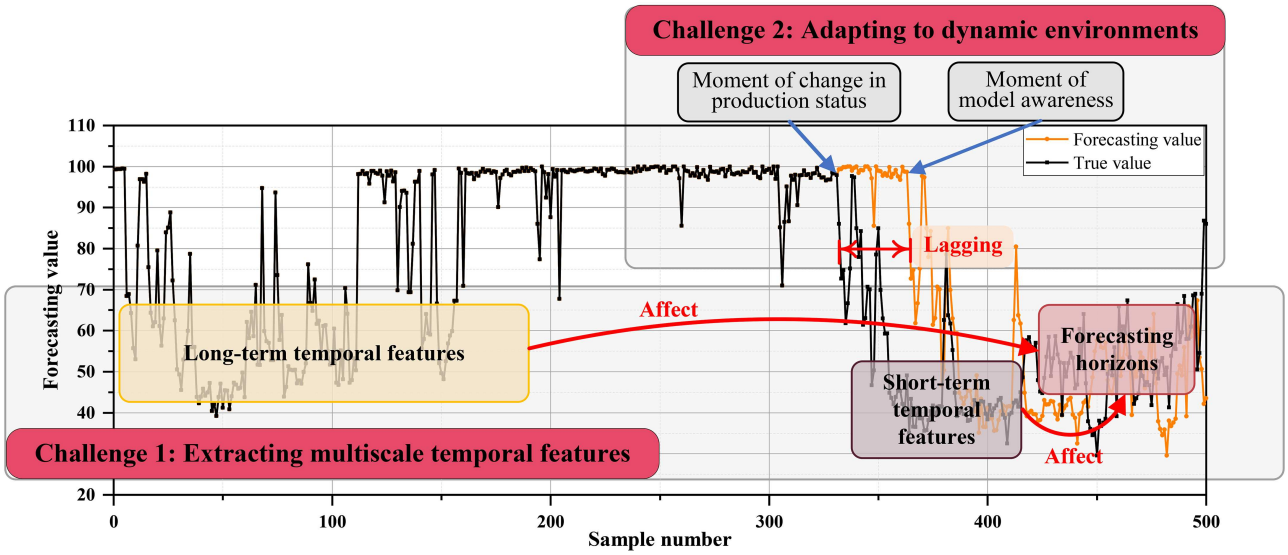

Figure 1 (Color online) The schematic illustration of the challenges in industrial time series forecasting.

time-series features remains challenging [21]. Long and short-term memory (LSTM) networks [22] and transformers [23] represent different approaches to this temporal modeling challenge. LSTM-based methods, with their recursive design, excel in time series modeling but struggle to capture ultra-long-term temporal features [24]. Structures like highway channels, such as LSTNet [25] and HLSTMs [26], have been proposed to address this limitation, yet these approaches only partially tackle ultra-long-term temporal features, falling short of a comprehensive solution. Conversely, the parallel structure of transformers resolves long-distance forgetting and facilitates capturing ultra-long-range temporal features [27]. Methods like LogTrans [28] and Informer [29] have shown strong performance in various temporal modeling tasks. However, this strength comes at a cost, relying on attentional mechanisms for temporal feature extraction results in suboptimal performance in short-term localized feature extraction, as a large number of historical samples may overshadow the characteristics of localized samples [30]. This fundamental trade-off between long-range and short-term feature extraction capabilities highlights the need for a unified approach that can effectively leverage the strengths of both paradigms.

Regarding the online self-calibration challenge, the industrial community has explored various adaptive strategies, with retraining and just-in-time learning emerging as the most commonly adopted methods [4]. Retraining is typically a final option due to high computational resource consumption and the absence of clear criteria for determining when and to what extent retraining is necessary. Just-in-time learning represents a more sophisticated approach, involving fine-tuning the model during online application by selecting similar historical samples based on known auxiliary variables [31]. Recent developments in this area have yielded promising results, including optimal weighting distance-based similarity measures that enhance locally weighted PLS modeling [32] and fast locally weighted methods specifically designed for large-scale industrial processes [33]. These advances demonstrate the potential of adaptive learning in addressing industrial scalability concerns. Despite these improvements, fundamental challenges persist in JITL-based approaches. Firstly, fine-tuning reduces single-step computation but still maintains high overall resource consumption. Secondly, accurate sample selection is challenging, as common metrics inadequately capture complex correlations among industrial samples and utilizing complex similarity measures increases computational burden. Additionally, when few similar samples with labels exist in the historical pool, precise calibration of model results is unattainable [34]; and both retraining and just-in-time learning methods struggle to determine optimal training epochs to adjust the model, significantly affecting calibration performance. These challenges are further compounded by the growing complexity of modern industrial processes, where traditional calibration methods face increasing demands for real-time adaptability while maintaining computational efficiency. This situation mirrors broader trends in adaptive machine learning, where recent advances in pattern recognition [35], hyperspectral imaging [36, 37], and multimodal representation learning [38] have highlighted the critical importance of developing methods that can dynamically adjust to changing operational conditions without sacrificing performance or efficiency.

In summary, while solutions to both challenges are currently absent, continued exploration is invaluable. Integrating parallel transformer and serial LSTM architectures may address multiscale temporal feature extraction, while investigating strategies to circumvent retraining or fine-tuning is ideal for resolving the dynamic adaptive calibration

problem. Fortunately, the recently proposed diffusion model shows promise in addressing both challenges. This method models probability distributions without being hindered by normalization constants in maximum likelihood estimation or ambiguity in generated results from target agent usage [39]. It also avoids training challenges in adversarial methods. As a result, it offers flexibility in designing model structures, making it suitable for incorporation into the diffusion model for network design, including transformer and LSTM structures, without training concerns [40]. The recent exceptional performance of diffusion model-based time series modeling, like CSDI [41] and TimeGrad [42], further underscores their potential. Significantly, the proposed classifier guidance strategy, based on the diffusion modeling framework, offers a posteriori guidance generation method without retraining [43]. This provides a fresh perspective for the online self-calibration problem.

Therefore, this paper introduces a progressive state-guided self-calibration conditional diffusion transformer framework (SGSC-DiT). The proposed approach systematically addresses both identified challenges through carefully designed components. For the multiscale temporal feature extraction challenge, we introduced a multiscale transformer network comprising both transformer and LSTM structures. This hybrid architecture strategically combines the parallel processing capabilities of transformers for long-range dependency capture with the sequential processing strengths of LSTM for local temporal pattern preservation, thereby transcending the limitations of individual approaches. For the online self-calibration challenge, we developed a progressive state-guided self-calibration strategy that fundamentally differs from existing JITL approaches. Rather than relying on computationally expensive sample selection and model fine-tuning, our method utilizes real-time feedback from auxiliary variables to guide the calibration of forecasting results for key quality variables, achieving adaptation through gradient-based guidance within the diffusion process. This mechanism enables the model to adapt to dynamic environmental changes during online applications without retraining, significantly reducing computational overhead while maintaining effective adaptation capabilities.

(1) A state-guided progressive adaptive self-calibration strategy for enhancing model adaptation to dynamic environments without retraining.

(2) A multiscale transformer model for improving the perception of multiscale temporal features.

(3) A conditional diffusion transformer (SGSC-DiT) that addresses both multiscale temporal feature extraction and dynamic calibration challenges simultaneously.

(4) A comprehensive experimental demonstration of the validity and superiority of the proposed method compared to the existing methods.

The rest of the manuscript unfolds as follows. Section 2 offers a brief review of diffusion and transformer models. Section 3 presents the problem definition along with the proposed method. Section 4 provides detailed experimental demonstration and analysis, while Section 5 concludes and outlines future prospects.

2 Preliminaries

Important symbols used throughout this paper are defined in Table 1 for reference. Notably, for easy distinction, the superscript t denotes the sampling time and the subscript t denotes the diffusion step.

2.1 Diffusion model

Diffusion models, a subset of generative models, adjust data distributions to synthesize data that accurately reflects real characteristics with improved quality and diversity. Figure 2 illustrates its basic structure, containing a forward and a reverse process. The forward process is a Markov process wherein noise is incrementally added T times to the original data $\mathbf{Z}_0 \sim q(\mathbf{Z}_0)$, converting it to $\mathbf{Z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

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

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

where $\{\beta_t \in (0,1)\}_{t=1}^{T}$ are the monotonically increasing hyperparameters to control the noise variance.

Instead, the inverse process progressively removes the added noise from the $\mathbf{Z}_T$, where the posterior probability $q\left(\mathbf{Z}_{t-1} \left| \mathbf{Z}_t \right.\right)$ is crucial. Ho et al. [39] proposed to fit it using a deep learning network

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

Table 1 Description of important symbols.

Symbol	Description	Shape	Symbol	Description	Shape
		Diffusion model symbols			
$\mathbf{Z}_0$	Original data	$\mathbb{R}^{N\times d}$	$\mathbf{Z}_t$	Noisy data at step t	$\mathbb{R}^{N\times d}$
$\mathbf{Z}_T$	Pure Gaussian noise	$\mathbb{R}^{N\times d}$	β_t	Noise variance schedule	–
α_t	Noise coefficient	–	$\bar{\alpha}_t$	Cumulative coefficient	–
$\varepsilon_\theta(\mathbf{Z}_t,t)$	Predicted noise	$\mathbb{R}^{N\times d}$	μ_θ	Predicted mean	$\mathbb{R}^{N\times d}$
		Transformer symbols			
$\mathbf{X}$	Input data	$\mathbb{R}^{N\times d_x}$	$\mathbf{X}_e$	Embedded features	$\mathbb{R}^{N\times d_{\text{model}}}$
$\mathbf{X}_A$	Attention output	$\mathbb{R}^{N\times d_{\text{model}}}$	$\mathbf{X}_F$	Feed-forward output	$\mathbb{R}^{N\times d_{\text{model}}}$
d_{model}	Embedding dimension	–	d_k	Attention dimension	–
$\mathbf{W}_{Qi}$	Query parameters	$\mathbb{R}^{d_{\text{model}}\times d_k}$	$\mathbf{W}_{Ki}$	Key parameters	$\mathbb{R}^{d_{\text{model}}\times d_k}$
		Proposed method symbols			
$\mathbf{X}$	Auxiliary variables	$\mathbb{R}^{N\times d_x}$	$\mathbf{Y}$	Quality variables	$\mathbb{R}^{N\times 1}$
$\mathbf{Z}_0$	Implicit variables	$\mathbb{R}^{N\times d_h+1}$	$\mathbf{H}$	Implicit features	$\mathbb{R}^{N\times d_h}$
$\mathbf{C}$	Conditional information	$\mathbb{R}^{N\times d_h}$	$\mathbf{F}_t^l$	Long-term patterns	$\mathbb{R}^{N\times d_f}$
$\mathbf{M}_t^l$	Short-term patterns	$\mathbb{R}^{N\times d_m}$	$\mathbf{O}_t^l$	Output features	$\mathbb{R}^{N\times d_o}$
t_0	Forecasting start	–	H	Recall window length	–
P	Forecasting horizon	–			

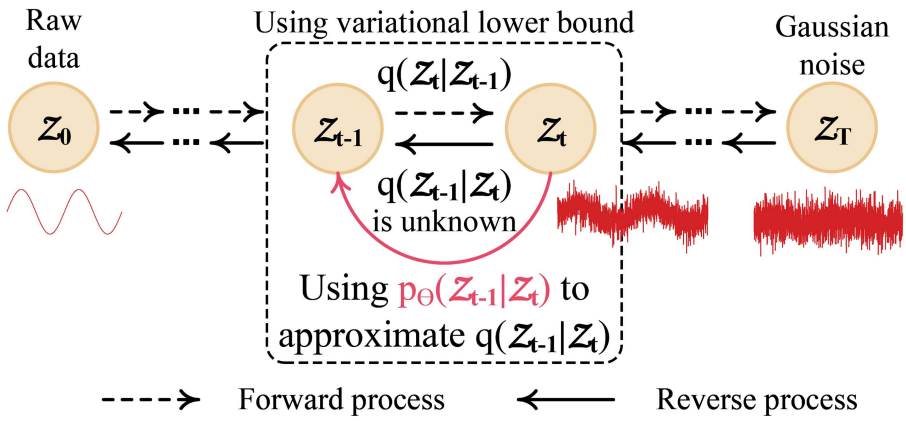

Figure 2 (Color online) The illustration of the diffusion model.

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

where $\alpha_t=1-\beta_t,\bar{\alpha}_t=\prod_{i=1}^{t}\alpha_i$, and $p_\theta\left(\mathbf{Z}_{t-1}\left|\mathbf{Z}_t\right.\right)\approx q\left(\mathbf{Z}_{t-1}\left|\mathbf{Z}_t\right.\right)$ denotes the fitting posterior probability, $\varepsilon_\theta\left(\mathbf{Z}_t,t\right)$ denotes the noise prediction by the deep learning network at step t. Accordingly, the training loss function of the deep learning network is the noise prediction error

$$\mathcal{L}_t=\mathbb{E}_{\mathbf{Z}_0,\bar{\varepsilon}_t}\left[\left\|\bar{\varepsilon}_t-\varepsilon_\theta\left(\mathbf{Z}_t,t\right)\right\|^2\right], \tag{5}$$

where $\bar{\varepsilon}_t$ denotes the real added noise in the forward process at step t.

Upon obtaining a trained diffusion model, data sampled from the standard Gaussian distribution can be converted to the target distribution, thereby enabling target generation.

2.2 Transformer encoder

The Transformer encoder, depicted in Figure 3, is widely adopted in time series modeling. It is composed of N_e identical modules revolving around the self-attention mechanism with ultra-long-range feature extraction capability.

First, the input data $\mathbf{X}\in\mathbb{R}^{N\times d_x}$ needs to traverse the linear and positional embedding layers to augment its representativeness and to scale the sample timings

$$\mathbf{X}_e=\mathbf{X}\mathbf{W}_e{}^{\mathrm{T}}+\mathrm{PE}\left(\mathbf{X}\mathbf{W}_e{}^{\mathrm{T}}\right), \tag{6}$$

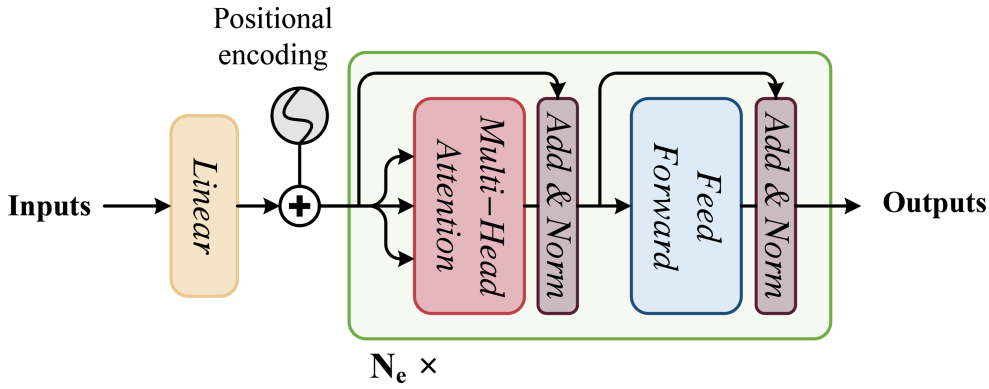

Figure 3 (Color online) The illustration of the Transformer encoder.

$$\begin{cases} \mathrm{PE}\left(pos, 2i\right) = \sin\left(pos/10000^{2i/d_{\mathrm{model}}}\right), \\ \mathrm{PE}\left(pos, 2i+1\right) = \cos\left(pos/10000^{2i/d_{\mathrm{model}}}\right), \end{cases} \tag{7}$$

where $\mathbf{X}_e \in \mathbb{R}^{N\times d_{\mathrm{model}}}$ denotes the embedded feature, $\mathbf{W}_e \in \mathbb{R}^{d_{\mathrm{model}}\times d_x}$ denotes the parameters. $pos \in [1, N]$ and $i \in [1, d_{\mathrm{model}}/2]$ denote the sample time position and dimension, d_x and N denote the input data dimension and sample number, d_{model} denotes the embedding dimension.

Next, the embedding feature temporal features are extracted through N_e layers of the encoder. In each layer, the long-range inter-sample similarities are first taken through the multi-head attention module

$$\begin{aligned} \mathbf{X}_A &= \mathrm{LN}\left(\mathbf{X}_e + \mathrm{MHA}\left(\mathbf{X}_e\right)\right) \\ &= \mathrm{LN}\left(\mathbf{X}_e + \mathrm{Concat}\left(\mathbf{X}_{A_1}, \ldots, \mathbf{X}_{A_h}\right)\mathbf{W}_A{}^{\mathrm{T}}\right), \end{aligned} \tag{8}$$

$$\begin{aligned} \mathbf{X}_{A_i} =& \mathrm{softmax}\left(\frac{\mathbf{Q}_i\mathbf{K}_i{}^{\mathrm{T}}}{\sqrt{d_k}}\right)\mathbf{V}_i \\ =& \mathrm{softmax}\left(\frac{\left(\mathbf{X}_e\mathbf{W}_{Q_i}{}^{\mathrm{T}}\right)\left(\mathbf{X}_e\mathbf{W}_{K_i}{}^{\mathrm{T}}\right)^{\mathrm{T}}}{\sqrt{d_k}}\right)\left(\mathbf{X}_e\mathbf{W}_{V_i}{}^{\mathrm{T}}\right), \end{aligned} \tag{9}$$

where $\mathbf{X}_A, \mathbf{X}_{A_i} \in \mathbb{R}^{N\times d_{\mathrm{model}}}$ denote the extracted final features and sub-features, respectively. $\mathbf{W}_A, \mathbf{W}_{Q_i}, \mathbf{W}_{K_i}, \mathbf{W}_{V_i}$ are the parameters, d_k is the attention space dimension, and $\mathrm{LN}(\cdot)$ is the layer normalization.

Then, enhancing the nonlinear representativeness of the features via Feed Forward layer

$$\begin{aligned} \mathbf{X}_F &= \mathrm{LN}\left(\mathbf{X}_A + \mathrm{FeedForward}\left(\mathbf{X}_A\right)\right) \\ &= \mathrm{LN}\left(\mathbf{X}_A + \max\left(0, \mathbf{X}_A\mathbf{W}_1{}^{\mathrm{T}} + \mathbf{b}_1\right)\mathbf{W}_2{}^{\mathrm{T}} + \mathbf{b}_2\right), \end{aligned} \tag{10}$$

where $\mathbf{X}_F \in \mathbb{R}^{N\times d_{\mathrm{model}}}$ denotes the final features, $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{b}_1$, and $\mathbf{b}_2$ are the parameters.

By iteratively extracting features across multiple layers, each sample will eventually encapsulate the relevant temporal information of the entire sequence, applicable to various downstream tasks.

3 Methodology

3.1 Problem definition

The task of key quality variables forecasting involves predicting future variables $[\mathbf{X}; \mathbf{Y}]^{t_0:t_0+P}$ using historical auxiliary variables $\mathbf{X}^{t_0:t_0-H}$, where the key quality variables $\mathbf{Y}^{t_0:t_0+P}$ are the primary objectives and $\mathbf{X}^{t_0:t_0+P}$ is the secondary objective. In this paper, it is considered a conditional diffusion process.

The forward process is essentially the same as described by (1) except that $\mathbf{Z}_0$ is replaced by the forecasting target $[\mathbf{X}; \mathbf{Y}]^{t_0:t_0+P}$. While the reverse process is expanded into a conditional probability based on (3) as

$$\begin{aligned} & p_{\theta,t}\left([\mathbf{X}; \mathbf{Y}]_{t-1}^{t_0:t_0+P} \middle| [\mathbf{X}; \mathbf{Y}]_t^{t_0:t_0+P}, \mathbf{X}^{t_0:t_0-H}\right) \\ &= \mathcal{N}\left([\mathbf{X}; \mathbf{Y}]_{t-1}^{t_0:t_0+P}; \mu_\theta\left([\mathbf{X}; \mathbf{Y}]_t^{t_0:t_0+P}, t \middle| \mathbf{X}^{t_0:t_0-H}\right), \sigma_\theta^2\right). \end{aligned} \tag{11}$$

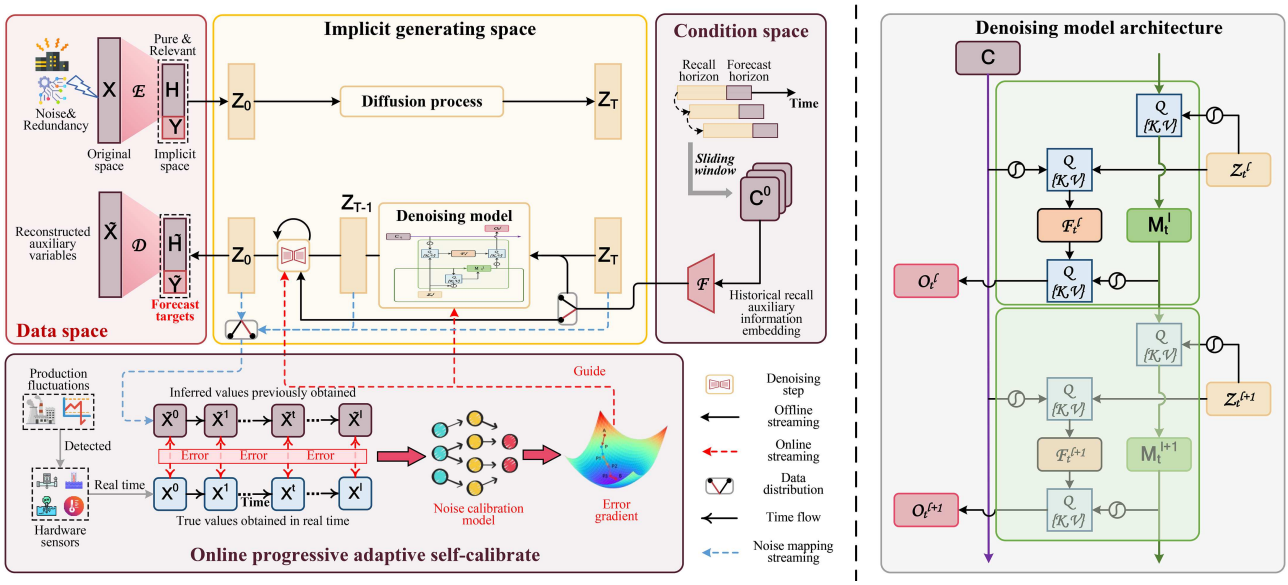

Figure 4 (Color online) The illustration of the SGSC-DiT framework. The framework processes raw industrial data through SQAE embedding, applies diffusion-based forecasting with a novel MultiscaleFormer architecture that combines transformer ($\mathbf{F}_t^l$) and LSTM ($\mathbf{M}_t^l$) structures for comprehensive temporal modeling, and implements progressive self-calibration using real-time sensor feedback ($\mathbf{X}_{\text{online}}$) through gradient guidance $\nabla_{\mathbf{Z}_t} \log p_\phi(\mathbf{X}_{\text{online}}|\mathbf{Z}_t)$, enabling online adaptation without retraining.

Once a trained deep neural network adequately fits the distribution, the forecasting task can be executed through a full reverse inference process as

$$p_{\theta,0:T} = [\mathbf{X};\mathbf{Y}]_T^{t_0:t_0+P} \prod_{t=1}^{T} p_{\theta,t}; \quad [\mathbf{X};\mathbf{Y}]_T^{t_0:t_0+P} \sim \mathcal{N}(\mathbf{0},\mathbf{I}). \tag{12}$$

Next, this paper will focus on efficiently solving the task, including refining the diffusion model, constructing the noise prediction deep network, and calibrating results online.

3.2 State-guided progressive adaptive self-calibration conditional diffusion framework

As shown in Figure 4, this subsection presents an SGSC-DiT to tackle multiscale industrial process time series forecasting, involving four main procedures.

3.2.1 *Embedding and forward*

Auxiliary variables $\mathbf{X}$ are obtained in real time by multiple sets of sensors; they inherently exhibit strong redundancy and high noise. Thus, this paper first constructs a stacked quality-related auto-encoder (SQAE) [44] with N_s layers to convert $\mathbf{X}$ into the highly expressive implicit space, where each layer can be expressed as

$$\begin{aligned}
\mathbf{H}_i &= \text{EncoderLayer}_i\left(\mathbf{H}_{i-1}\right) = \tanh\left(\text{Linear}\left(\mathbf{H}_{i-1}\right)\right), \\
\tilde{\mathbf{H}}_{i-1} &= \text{DecoderLayer}_i\left(\mathbf{H}_i\right) = \tanh\left(\text{Linear}\left(\mathbf{H}_i\right)\right), \\
\tilde{\mathbf{Y}}_i &= \text{QualityDecoder}_i\left(\mathbf{H}_i\right) = \tanh\left(\text{Linear}\left(\mathbf{H}_i\right)\right),
\end{aligned} \tag{13}$$

where $\mathbf{H}_i$ and $\tilde{\mathbf{H}}_{i-1}$ denote the implicit variables and reconstructed inputs of the ith encoder layer ($1 \leqslant i \leqslant N_s$, $\mathbf{H}_0 = \mathbf{X}$), $\tilde{\mathbf{Y}}_i$ denotes the predicted value of $\mathbf{Y}$ obtained using $\mathbf{H}_i$. Then, each encoder is trained layer by layer using the sum of reconstruction error and prediction error as the loss function

$$\mathcal{L}_i^{\text{SQAE}} = \left\|\mathbf{H}_{i-1} - \tilde{\mathbf{H}}_{i-1}\right\|_2^2 + \left\|\mathbf{Y} - \tilde{\mathbf{Y}}_i\right\|_2^2, \tag{14}$$

where $\mathcal{L}_i^{\text{SQAE}}$ denotes the training loss of the ith layer. In this way, the final deep implicit features $\mathbf{H}_{N_s}$ will be highly expressive features with low redundancy and closely related to $\mathbf{Y}$.

Notably, while SQAE is not the only embedding method and alternatives such as MLP, VAE, and other advanced feature extraction methods could be employed, SQAE is specifically selected for its dual-objective design

that simultaneously performs reconstruction and quality-related feature extraction. This quality-aware mechanism ensures that the extracted implicit features are not only representative of the input auxiliary variables but also highly correlated with the target quality variables, making it particularly suitable for industrial predictive modeling tasks where the relationship between process variables and quality indicators is crucial.

Afterward, the hybrid high-expression implicit features of the targets are utilized as inputs for the diffusion model

$$\mathbf{Z}_0 = [\mathbf{H}_{N_s}; \mathbf{Y}]^{t_0:t_o+P} = [\text{Embedding}\,(\mathbf{X})\,; \mathbf{Y}]^{t_0:t_o+P}, \tag{15}$$

where the Embedding$(\cdot)$ denotes the embedding model. This implicit embedding method removes invalid redundant features, simplifying probabilistic modeling. Then, the inputs $\mathbf{Z}_0$ are noised according to (1) and (2) for the forward process.

3.2.2 *MultiscaleFormer-based denoising and reverse*

To accurately predict the added noise in the forward process, this paper proposes a multiscale transformer (MultiscaleFormer) noise prediction model, illustrated in Figure 4. It can be divided into three steps: truncating data, capturing long-term evolutionary patterns, and extracting short-term temporal fluctuation features. The following will detail the implementation of noise prediction, using the t diffusion step as an example.

Step 1: Truncating data. Firstly, to address the issue of accumulating feature extraction bias caused by recursive structures, this paper proposes a truncated recursive strategy

$$\mathbf{Z}_t^l = \text{Truncate}\,(\mathbf{Z}_t) = [\mathbf{H}_{N_s}; \mathbf{Y}]^{t_0+(l-1)k:t_0+lk}, \tag{16}$$

where k and $\mathbf{Z}_t^l$ denote the truncation length and the lth $(1 \leqslant l \leqslant P/k)$ block of $\mathbf{Z}_t$, respectively. This shortens the feature propagation distance from the original length P to P/k, thereby reducing the cumulative impact of feature errors in long-distance forecasting. Notably, to accurately perceive the current denoising step t, it is essential to append its information to $\mathbf{Z}_t$ using (6) and (7).

Step 2: Capturing long-term evolutionary patterns. Before predicting noise, it is crucial to convert the known long-range historical samples into processable conditional information

$$\mathbf{C} = \text{Embedding}\left(\mathbf{X}^{t_0-H:t_0}\right). \tag{17}$$

Subsequently, the acquired conditional information $\mathbf{C}$ can be utilized to extract the historical long-term evolutionary patterns

$$\mathbf{F}_t^l = \text{Encoder}_{\text{QKV}}\left(\mathbf{Q} = f\left(\mathbf{Z}_t^l\right), \{\mathbf{K}, \mathbf{V}\} = g\left(\mathbf{C}\right)\right), \tag{18}$$

where $\text{Encoder}_{\text{QKV}}(\cdot)$ denotes the Transformer encoder described in Section 2, $f(\cdot)$ and $g(\cdot)$ denote functions that generate queries and key-value pairs, respectively. Applying the attention mechanism enables the features $\mathbf{F}_t^l$ to capture long-term historical patterns in $\mathbf{C}$.

Step 3: Extracting short-term temporal fluctuation features. After identifying the long-term evolution pattern, a recursive structure is employed to extract short-term temporal fluctuation features, addressing the reduced sensitivity of transformer to timing in parallel structures

$$\mathbf{M}_t^l = \text{Encoder}_{\text{QKV}}\left(\mathbf{Q} = f\left(\mathbf{M}_t^{l-1}\right), \{\mathbf{K}, \mathbf{V}\} = g\left(\mathbf{Z}_t^l\right)\right), \tag{19}$$

$$\mathbf{O}_t^l = \text{Encoder}_{\text{QKV}}\left(\mathbf{Q} = f\left(\mathbf{F}_t^l\right), \{\mathbf{K}, \mathbf{V}\} = g\left(\mathbf{M}_t^l\right)\right), \tag{20}$$

where $\mathbf{M}_t^0 = \mathbf{X}^{t_0-H:t_0+H-M}$ denotes the initial short-term temporal fluctuation features. The resultant output $\mathbf{O}_t^l$ will encapsulate the multiscale temporal characteristics of the entire time series. Hence, it can be used to predict the forward noise as

$$\varepsilon_\theta\left(\mathbf{Z}_t, t\right) = \text{Linear}\left(\left[\mathbf{O}_t^1, \ldots, \mathbf{O}_t^l, \ldots, \mathbf{O}_t^{P/k}\right]\right). \tag{21}$$

Step 4: Multistep forecasting. With accurate noise prediction, the inverse diffusion process is completed by substituting (21) into (11) and (12), ultimately inferring the implicit variables $\tilde{\mathbf{Z}}_0 = [\tilde{\mathbf{H}}_{N_s}; \tilde{\mathbf{Y}}]^{t_0:t_o+P}$. $\tilde{\mathbf{Y}}$ denotes the forecasting results of the key quality variables $\mathbf{Y}$, while the forecasting results of the auxiliary variables $\tilde{\mathbf{X}}$ can be obtained by converting $\tilde{\mathbf{H}}_{N_s}$ back to the original space using the decoder in (13).

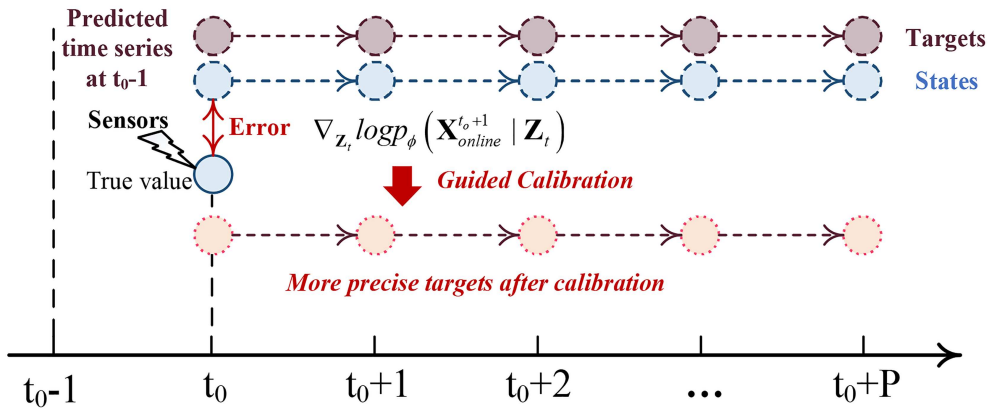

Figure 5 (Color online) Schematic diagram of the state-guided progressive adaptive self-calibration mechanism. The diagram shows how real-time auxiliary variable measurements from sensors guide the calibration of quality variable predictions through gradient-based guidance, enabling progressive adaptation without model retraining.

3.2.3 *State-guided self-calibration*

Environmental fluctuations often cause performance degradation or model mismatch when trained models are applied online. To address this, inspired by the classifier-guidance diffusion model [43], this paper introduces a state-guided self-calibration mechanism, which can enable self-calibration of forecasting results without retraining the diffusion model, simply by adding corrections to the denoising process. This mechanism holds considerable interest for real industrial applications and helps the model to adapt to dynamic environments. Figure 5 illustrates the progressive self-calibration mechanism. As shown in the diagram, when real-time auxiliary variable measurements become available from sensors, the system computes the error between predicted states and true values, then uses this error to generate gradient guidance $\nabla_{Z_t} \log p_\phi(X_{\text{online}}^{t_c+1}|Z_t)$ for calibrating the target predictions. This process results in more precise target forecasts that adapt to the current process conditions. The details are as follows.

Remark 1. The online self-calibration process is progressive, calibrating forecasting results $\mathbf{Y}^{t_0:t_0+P}$ individually using the true values of auxiliary variables at the corresponding moments. This calibration is essential for industrial processes, where multi-step quality variables forecasting results help anticipate adverse conditions, and precise current calibrations guide refined operations.

Remark 2. Before online calibration, an implicit variable decoding model $p_\phi = (\mathbf{X}|\mathbf{Z}_t)$ should be trained with noisy samples $\mathbf{Z}_t$. This paper employs the basic MLP model, and other more complex models are optional.

Assumption 1. During the online application, auxiliary state variables detected by hardware sensors in the forecasting horizon $t_0 : t_o + P$ can be precisely obtained progressively in real time and denoted as $\mathbf{X}_{\text{online}}^{t_0:t_o+P}$.

Theorem 1. If the true value $\mathbf{X}_{\text{online}}^{t_o+t_c}$ of the auxiliary state variables at time $0 < t_c \leqslant P$ is known, the calibrating value can be obtained by adjusting the denoising step of the diffusion model from (11) to the following equation at that moment

$$p_{\theta,t}^{\text{calibrate}} = \mathcal{N}\left(\mu_\theta + s\sigma_\theta^2 \nabla_{\mathbf{Z}_t} \log p_\phi\left(\mathbf{X}_{\text{online}}^{t_o+t_c}|\mathbf{Z}_t\right), \sigma_\theta^2\right), \tag{22}$$

where $p_{\theta,t}^{\text{calibrate}}$ denotes the noise prediction process during online self-calibration and s denotes a scalar of the magnitude of the calibration.

Proof. The proof can be completed analogously to the proof of the classifier-guidance diffusion model [43] (H conditional diffusion process).

3.3 SGSC-DiT-based industrial time series forecasting

The SGSC-DiT training process parallels traditional diffusion model training, utilizing (5) as the loss function and thus requires no further elaboration. After training, B samples $\mathbf{Z}_T^b \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ $(0 < b \leqslant B)$ are extracted from the Gaussian distribution, and the corresponding implicit variables are $\tilde{\mathbf{Z}}_0^b = [\tilde{\mathbf{H}}_{N_s}^b; \tilde{\mathbf{Y}}^b]^{t_0:t_o+P}$ derived using (12). Then, the mean values of these variables yield the predicted target values $[\tilde{\mathbf{X}}; \tilde{\mathbf{Y}}]^{t_0:t_0+P}$

$$\tilde{\mathbf{X}}^{t_0:t_0+P} = \frac{1}{B}\sum_{b=1}^{B} \text{DecoderLayer}\left(\left[\tilde{\mathbf{H}}_{N_s}^b\right]^{t_0:t_o+P}\right), \tag{23}$$

Table 2 Dataset statistical information.

Dataset characteristics	Salt lake chemical	Hydrocracking
Total samples	4000	2600
Training/testing split	First 3000/last 800	First 2000/last 500
Time span	6 months	36 months
Sampling interval	1 h	4 h
Auxiliary variables	30 (current, pressure, flow rate, etc.)	43 (temperature, pressure, flow rate, etc.)
Quality variables	1	1
Matrix dimensions	$\mathbb{R}^{4000\times 31}$	$\mathbb{R}^{2600\times 44}$

$$\tilde{\mathbf{Y}}^{t_0:t_0+P}=\frac{1}{B}\sum_{b=1}^{B}\left[\tilde{\mathbf{Y}}^{b}\right]^{t_0:t_o+P}. \tag{24}$$

As for online application, real-time auxiliary variable values refine the inverse process via Theorem 1 for enhanced accuracy. This process can be described by Algorithm 1.

Algorithm 1 : The state-guided online self-calibration processes of SGSC-DiT.

Input: A trained SGSC-DiT model $(\mu_\theta(\mathbf{Z}_t),\sigma_\theta^2(\mathbf{Z}_t))$, a trained implicit variable decoding model $p_\phi=(\mathbf{X}|\mathbf{Z}_t)$, calibration magnitude s.
for $t_c\leftarrow 1$ **to** P **do**
  if $\mathbf{X}_{\text{online}}^{t_o+t_c}$ is known **then**
    for $b\leftarrow 1$ **to** B **do**
      $\mathbf{Z}_T^b\sim\mathcal{N}(\mathbf{0},\mathbf{I})$;
      for All $t\leftarrow T$ **to** 1 **do**
        $\mathbf{Z}_{t-1}^b\leftarrow$ sample from $\mathcal{N}\left(\mu_\theta+s\sigma_\theta^2\nabla_{\mathbf{Z}_t}\log p_\phi\left(\mathbf{X}_{\text{online}}^{t_o+t_c}|\mathbf{Z}_t\right),\sigma_\theta^2\right)$;
      end for
    end for
    Return: $[\tilde{\mathbf{X}};\tilde{\mathbf{Y}}]^{t_0:t_0+P}$ using (23) and (24);
  end if
end for

4 Experiments

In this section, the effectiveness of the proposed SGSC-DiT method is validated by conducting simulation experiments using two real industrial datasets.

4.1 Datasets

To validate the effectiveness of the proposed method, this paper conducts abundant experiments on two real industrial datasets.

4.1.1 *Process descriptions*

• Salt lake chemical process. Figure 6 depicts the salt lake chemical process crystallization, where a potassium mixture reacts in the crystallizer to yield potassium sulfate granules. Realizing the underflow grade forecasting enables finer control of the reaction state. It is a typical multiscale temporal industrial process with a long reaction period and highly susceptible to feed fluctuations, which is suitable for validating the proposed method.

• Hydrocracking process. Figure 7 illustrates the hydrocracking process, which refers to the catalytic cracking of heavy oil into light oil using hydrogen in a high-temperature and high-pressure environment. It has multiple reactors leading to its multiscale characterization and is similarly suitable for verifying the proposed method. Therefore, this paper is verified with the example of the light naphtha C5 content forecasting task.

4.1.2 *Data characteristics*

Both datasets are collected from real industrial processes. All auxiliary variables represent real-time measurable process parameters, while quality variables require offline analysis with measurement delays, reflecting realistic industrial forecasting scenarios. Table 2 provides comprehensive statistical information about data organization, variable specifications and preprocessing procedures for both datasets. The separation between training and testing samples is the margin reserved for the sliding window.

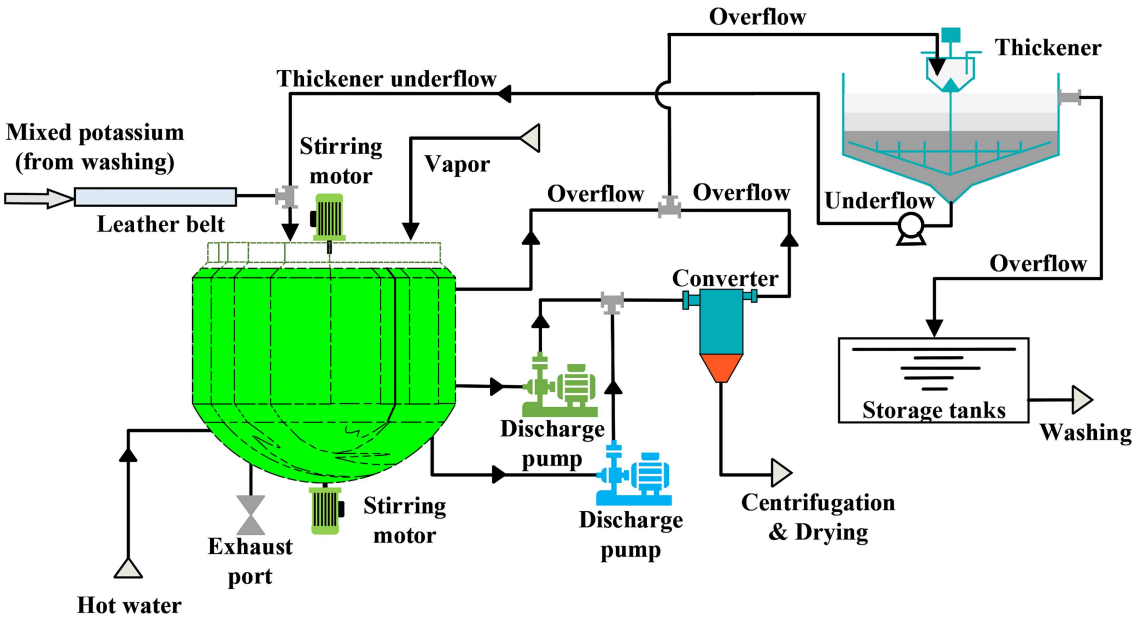

Figure 6 (Color online) The schematic diagram of the salt lake chemical process.

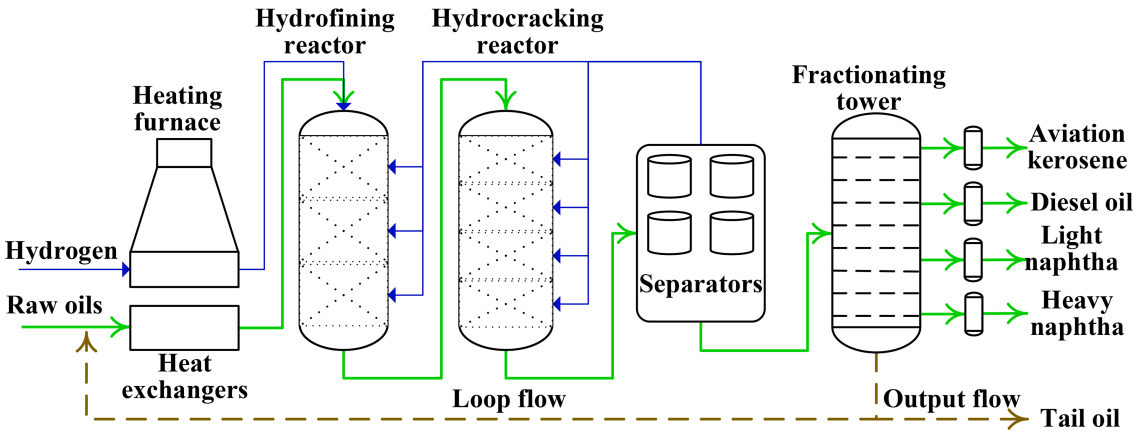

Figure 7 (Color online) The schematic diagram of the hydrocracking process.

4.2 Evaluation

4.2.1 *Baselines*

To demonstrate the superiority of the proposed method, representative methods from each dynamic modeling category are selected for comparison.

- LSTNet [25]. Utilizing LSTM and CNN to extract long- and short-term temporal features.
- Informer [29]. Using transformer with a sparse attention mechanism to extract longer range temporal features.
- LogTrans [28]. Extracting multiscale temporal features using transformer and convolutional neural networks.
- PatchTST [45]. A recent transformer-based method that employs channel-independent patching and self-supervised pre-training for improved long-term forecasting performance.
- iTransformer [46]. A transformer that applies attention mechanisms on the variate dimension rather than the temporal dimension, achieving good performance in multivariate time series forecasting.
- CSDI [41]. Realizing forecasting using a diffusion model and transformer network based on the idea of data imputation.
- TimeGrad [42]. Combining LSTM with a conditional diffusion model to model timing patterns.

Table 3 Hyperparameter settings of SGSC-DiT.

Hyperparameters	Salt lake chemical	Hydrocracking
Hidden space size (d_h)	{4, 8, 16, 32}	{4, 8, 16, 32}
Diffusion step (T)	{50, 100, 150, 200}	{50, 100, 150, 200}
Recall window length (H)	{5, 10, 15, 20, 30, 50, 100}	{5, 10, 15, 20, 30, 50, 100}

Table 4 The performance comparison on different datasets. Underlining denotes sub-optimal results, while bolding signifies optimal results. SGSC-DiT* indicates the results of SGSC-DiT after self-calibration.

Method		Salt lake chemical				Hydrocracking			
		Step = 1	Step = 4	Step = 8	Step = 16	Step = 1	Step = 4	Step = 8	Step = 16
LSTM-based									
LSTNet	RMSE	0.3044	0.7659	0.9187	1.0201	0.3930	0.5191	0.5664	0.6683
	MAE	0.2327	0.5768	0.7346	0.7808	0.2538	0.3523	0.3862	0.5044
Transformer-based									
Informer	RMSE	0.3762	0.7910	0.9219	0.9372	0.4065	0.5439	0.6181	0.7052
	MAE	0.2945	0.6178	0.7530	0.7569	0.2896	0.3954	0.4494	0.5199
LogTrans	RMSE	0.3170	0.7370	0.9197	0.9208	0.3727	0.5112	0.5789	0.6520
	MAE	0.2513	0.5747	0.7405	0.7282	0.2516	0.3527	0.4168	0.4649
PatchTST	RMSE	0.3938	0.8386	0.9037	0.9181	0.4300	0.5725	0.6084	0.6827
	MAE	0.3020	0.6884	0.7322	0.7459	0.2710	0.3500	0.3915	0.4816
iTransformer	RMSE	0.3884	0.7849	0.9017	0.9335	0.3918	0.5477	0.5513	0.6598
	MAE	0.3086	0.6031	0.7304	0.7396	0.2339	0.3528	0.3807	0.4614
Diffusion-based									
CSDI	RMSE	0.3530	0.8178	0.9137	0.9308	0.4682	0.5371	0.5888	0.6593
	MAE	0.2763	0.6372	0.7395	0.7502	0.2890	0.3943	0.4453	0.4862
TimeGrad	RMSE	0.3433	0.7933	0.9058	0.9250	0.4202	0.5134	0.5678	0.6465
	MAE	0.2820	0.6209	0.7306	0.7436	0.2860	0.3588	0.3801	0.4648
Proposed									
SGSC-DiT	RMSE	0.2695	0.7127	0.8837	0.9073	0.3674	0.4928	0.5282	0.6159
	MAE	0.2109	0.5462	0.7083	0.7106	0.2331	0.3077	0.3591	0.4483
SGSC-DiT*	RMSE	**0.2516**	**0.7024**	**0.8549**	**0.8806**	**0.3455**	**0.4801**	**0.5092**	**0.5754**
	MAE	**0.1976**	**0.5275**	**0.6759**	**0.6970**	**0.2091**	**0.2864**	**0.3285**	**0.4329**

4.2.2 *Metrics*

In this paper, root mean square error (RMSE) and mean absolute error (MAE) are chosen as quantitative metrics to measure the performance of each method

$$\mathrm{RMSE} = \sqrt{\sum_{j=1}^{P}\left(\mathbf{Y}^{t_0:t_0+i} - \tilde{\mathbf{Y}}^{t_0:t_0+i}\right)^2 / P}, \tag{25}$$

$$\mathrm{MAE} = \sum_{i=1}^{P}\left|\mathbf{Y}^{t_0:t_0+i} - \tilde{\mathbf{Y}}^{t_0:t_0+i}\right| / P. \tag{26}$$

Generally, the better the performance of the model, the smaller their values will be.

4.3 Experimental setting

To thoroughly validate the effectiveness of the proposed method, this paper conducts experiments with various forecasting steps, including 1, 4, 8, and 16. In this paper, the grid search method is used to find the optimal hyperparameters. The hyperparameter search space settings are shown in Table 3.

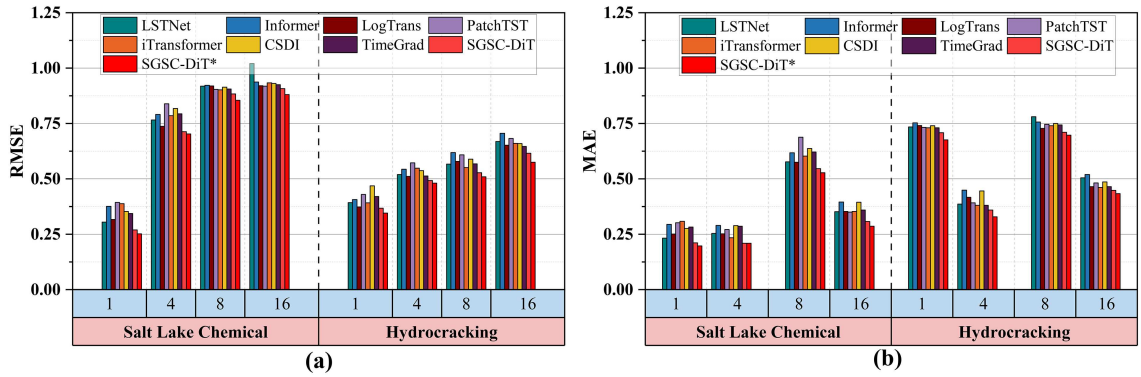

Figure 8 (Color online) Visualization results of quantitative evaluation of all methods on two datasets. (a) RMSE; (b) MAE.

4.4 Performance results and analysis

The optimal results, determined through grid search for hyperparameters, are shown in Table 4 and are visualized in Figure 8. Firstly, it focuses on the performance comparison of the proposed SGSC-DiT method with other commonly used methods. The experimental results demonstrate that the LSTNet method, utilizing the highway design, partially mitigates long-term information loss of LSTM-based methods. Its performance is comparable to the Informer method but remains inferior to LogTrans and other advanced Transformer methods. While traditional Transformer methods such as Informer and LogTrans can extract longer-range information, they do not decisively surpass LSTM-based methods. This is partly due to the compression in attention calculation of Informer and, more critically, the inadequate emphasis on proximate information, leading to the overshadowing of crucial near-neighbor data by extensive historical information. The recently proposed Transformer-based methods, PatchTST and iTransformer, demonstrate improved performance over traditional transformers in some long-term forecasting scenario by addressing some of these limitations. However, both methods still exhibit certain limitations in the context of industrial time series forecasting. PatchTST's channel-independent approach may overlook important cross-variable interactions that are crucial in industrial processes, while iTransformer's focus on variate-wise attention, though effective for general multivariate forecasting, does not specifically address the multiscale temporal dynamics inherent in industrial systems. Diffusion methods like CSDI and TimeGrad perform poorly in short-term forecasting but yield acceptable long-term results. This is because these methods derive their forecasts through multiple sampling, making short-term predictions prone to uncertainty. As the prediction horizon extends, the inherent uncertainty becomes more pronounced, and the diffusion-based methods, thus, outperform the deterministic approaches, enhancing their overall performance. The SGSC-DiT method, proposed in this study, integrates Transformer and LSTM advantages to enhance the ability of diffusion models to capture both short- and long-term sequence features comprehensively. Moreover, unlike existing methods including the recent PatchTST and iTransformer, SGSC-DiT incorporates an online adaptive calibration mechanism that enables real-time adaptation to dynamic industrial environments without model retraining. This improvement addresses inaccurate short-term predictions and strengthens long-term forecasting, demonstrating excellent performance across datasets.

Besides, to visually illustrate the performance disparities among various methods, this paper uses single-step prediction as an exemplar and presents the prediction curves of all methods, depicted in Figures 9 and 10. It is evident that the SGSC-DiT method proposed herein exhibits superior conformity to the actual curve at peaks and troughs. This enhancement stems from the extraction of multiscale features, rendering the model more responsive to abrupt sequence changes. Additionally, the curves of SGSC-DiT exhibit smoother transitions during steady-state operational phases, attributed to the richer descriptive capability of multi-scale features in probabilistic modeling of the diffusion model. This richness enhances the efficacy of SGSC-DiT in mitigating interference from factors like noise inherent in the original data.

Secondly, the self-calibrated SGSC-DiT* model exhibits higher accuracy than the uncalibrated SGSC-DiT model, validating the effectiveness of the proposed self-calibration mechanism. This enhancement becomes more pronounced with increased forecasting steps, further demonstrating its potential in practical industrial applications. It can enhance model perception accuracy and incrementally refine predictions after acquiring actual auxiliary variable values.

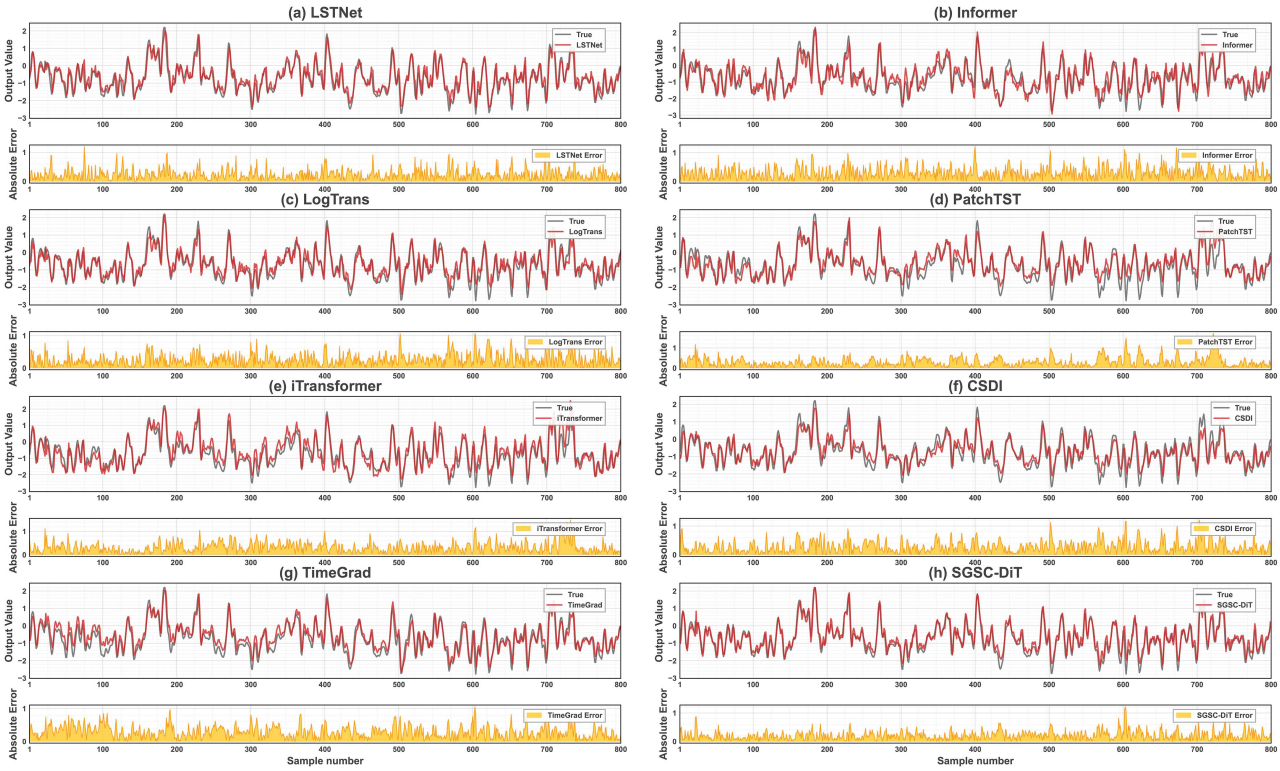

Figure 9 (Color online) Performance comparison of various methods for single-step forecasting on salt lake chemical dataset. (a) LSTNet; (b) Informer; (c) LogTrans; (d) PatchTST; (e) iTransformer; (f) CSDI; (g) TimeGrad; (h) SGSC-DiT.

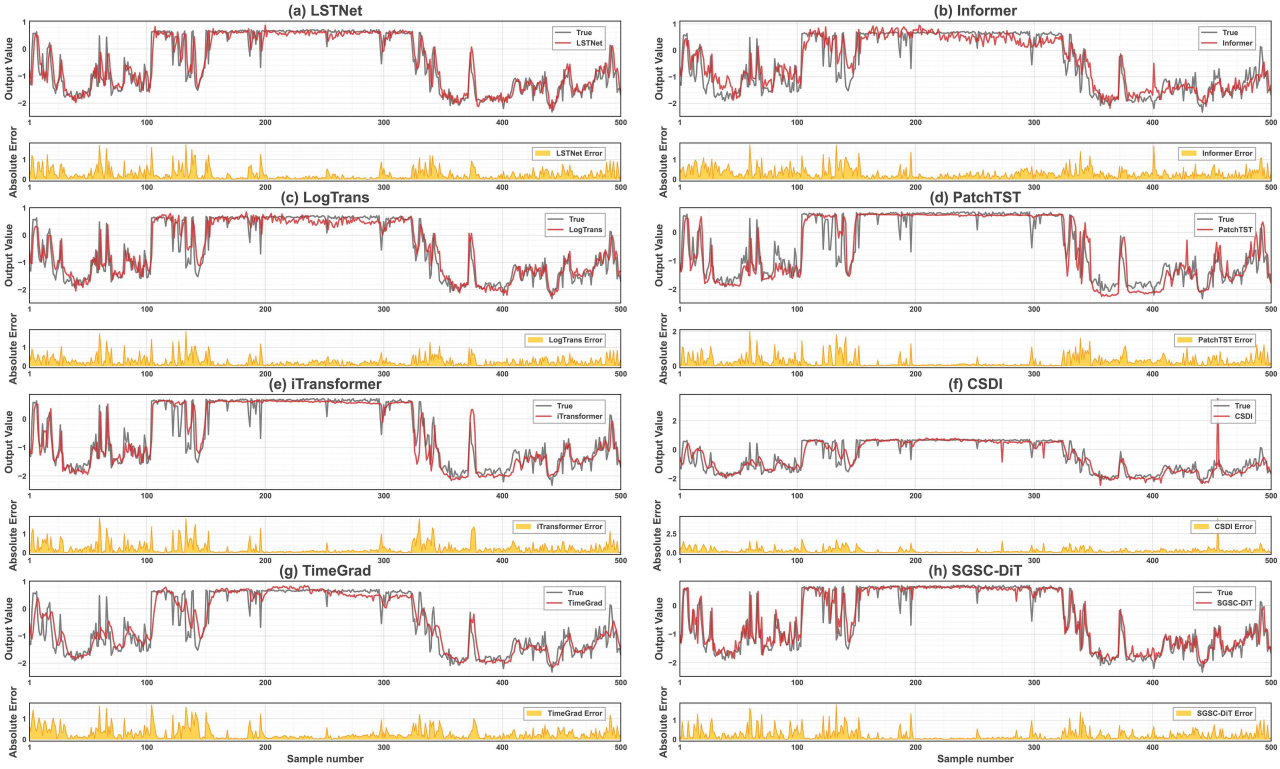

Figure 10 (Color online) Performance comparison of various methods for single-step forecasting on hydrocracking dataset. (a) LSTNet; (b) Informer; (c) LogTrans; (d) PatchTST; (e) iTransformer; (f) CSDI; (g) TimeGrad; (h) SGSC-DiT.

Table 5 Experimental configurations for calibration methods. Running time represents the average calibration time per sample.

Method	Learning rate	Training epochs	Sample selection	Key parameters
Retraining	1e−4 (Adam)	Max 50, early stop (patience = 5)	All historical samples (3000/2000 samples)	Patience = 5, validation-based stopping criterion
Just-in-time learning	1e−4 (Adam)	Max 20, early stop (patience = 3)	Euclidean distance-based similarity selection	Max 100 similar samples, 10th percentile threshold
SGSC-DiT*	–	–	Realtime auxiliary variable guidance	Calibration magnitude $s = 0.1$, sampling trajectories $B = 50$

Table 6 Performance comparison of different calibration strategies on the pre-trained SGSC-DiT, where * denotes results after applying the proposed state-guided self-calibration. Running time represents the average calibration time per sample.

Dataset	Method	RMSE	MAE	Running time (s)
Salt lake chemical	SGSC-DiT*	**0.2516**	**0.1976**	**61**
	Just-in-time learning	0.2620	0.2054	197
	Retraining	0.2794	0.2207	256
Hydrocracking	SGSC-DiT*	**0.3455**	**0.2091**	**29**
	Just-in-time learning	0.3517	0.2201	108
	Retraining	0.3593	0.2266	194

4.5 Validation of self-calibration mechanism

To further validate the effectiveness of the proposed self-calibration mechanism demonstrated in Subsection 4.4, this section provides a comprehensive comparison with existing calibration strategies. We compare our SGSC-DiT* method against two commonly used approaches: retraining and just-in-time learning, using single-step prediction as a representative case study.

4.5.1 *Calibration methods and experimental setup*

Three calibration approaches are evaluated based on the pre-trained SGSC-DiT model from Table 4.

• Retraining. Fine-tunes the entire pre-trained model using all available historical samples before each prediction.

• Just-in-time learning. Selectively fine-tunes the model using similar historical samples identified through distance-based similarity measures.

• SGSC-DiT*. Applies the proposed state-guided self-calibration mechanism (Algorithm 1) without any model retraining.

To ensure fair and reproducible comparisons, the detailed experimental configurations for each method are summarized in Table 5. All methods are evaluated under identical conditions using the same test samples and hardware platform.

4.5.2 *Comparative results and analysis*

The comparative results are presented in Table 6, which reveals several important findings. First, retraining demonstrates the highest computational cost while achieving suboptimal performance. This inefficiency arises because retraining lacks specificity in model fine-tuning and is heavily influenced by sample quantity in different states, becoming effective only after accumulating a significant number of changed state samples. Second, the just-in-time learning method shows improved adaptability compared to retraining by focusing on relevant historical samples. Nonetheless, the effective selection of similar samples is challenging, especially with limited labeled data in practical applications. Moreover, determining the optimal number of training iterations for re-fine-tuning is challenging, as both excessive and insufficient iterations yield suboptimal results, with the optimal number varying per sample. Consequently, both methods fail to achieve significant improvements and may even degrade performance post-calibration. In contrast, the proposed SGSC-DiT* method achieves superior performance while requiring minimal computational resources. This efficiency stems from its use of actual values of auxiliary variables pertinent to current key quality indicators to steer the calibration process, which is both highly relevant and real-time, and can be executed with just a single sample. Besides, it operates unsupervised, eliminating the need for labeled data. Thus, the state-guided self-calibration method offers considerable advantages and significant practical potential.

5 Conclusion

Time series modeling is pivotal in industrial process control. However, multiscale feature extraction and results self-calibration in dynamic environments limit its applications. Hence, this paper presents a state-guided adaptive progressive self-calibration conditional diffusion based on an implicit diffusion transformer network. It integrates recursive and parallel structures for multiscale feature extraction and employs auxiliary variable error gradients for self-calibration. This approach ensures effective multiscale feature extraction and progressive self-calibration in dynamic environments. Moreover, it outperforms mainstream techniques on two real industrial datasets, demonstrating significant application value.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. U25A20466, 62503507) and Natural Science Foundation of Hunan Province (Grant Nos. 2025JJ60423, 2025JJ10007).

References

1 Hu W X, Du C L, Li F B, et al. Adaptive observer-based integral event-triggered antivibration control of a full aircraft active landing gear system with irregular runway excitations. Sci China Inf Sci, 2024, 67: 162201
2 Dogru O, Xie J, Prakash O, et al. Reinforcement learning in process industries: review and perspective. IEEE CAA J Autom Sin, 2024, 11: 283–300
3 Liu D, Wang Y, Liu C, et al. EKG-AC: a new paradigm for process industrial optimization based on offline reinforcement learning with expert knowledge guidance. IEEE Trans Cybern, 2025, 1–11
4 Fujiwara K, Kano M, Hasebe S, et al. Soft-sensor development using correlation-based just-in-time modeling. AIChE J, 2009, 55: 1754–1765
5 Huang K K, Tao S J, Liu Y S, et al. Label propagation dictionary learning based process monitoring method for industrial process with between-mode similarity. Sci China Inf Sci, 2022, 65: 110203
6 Liu C, Wang Y, Fang Y, et al. Operating condition recognition of industrial flotation processes using visual and acoustic bimodal autoencoder with manifold learning. IEEE Trans Ind Inf, 2024, 20: 7428–7439
7 Liu C, Wang K, Wang Y, et al. Learning deep multimanifold structure feature representation for quality prediction with an industrial application. IEEE Trans Ind Inf, 2022, 18: 5849–5858
8 Chen H, Luo H, Huang B, et al. Transfer learning-motivated intelligent fault diagnosis designs: a survey, insights, and perspectives. IEEE Trans Neural Netw Learn Syst, 2024, 35: 2969–2983
9 Liu D, Wang Y, Liu C, et al. Scope-free global multi-condition-aware industrial missing data imputation framework via diffusion transformer. IEEE Trans Knowl Data Eng, 2024, 36: 6977–6988
10 Liu K, Lu N, Wu F, et al. Model fusion and multiscale feature learning for fault diagnosis of industrial processes. IEEE Trans Cybern, 2023, 53: 6465–6478
11 Liu C, Wang Y, Yang C, et al. Adaptive attention-driven manifold regularization for deep learning networks: industrial predictive modeling applications and beyond. IEEE Trans Ind Electron, 2024, 71: 13439–13449
12 Liu D, Wang Y, Liu C, et al. Blackout missing data recovery in industrial time series based on masked-former hierarchical imputation framework. IEEE Trans Automat Sci Eng, 2024, 21: 1138–1150
13 Chen D, Chen L, Zhang Y, et al. A multiscale interactive recurrent network for time-series forecasting. IEEE Trans Cybern, 2022, 52: 8793–8803
14 Aheleroff S, Xu X, Zhong R Y, et al. Digital twin as a service (DTaaS) in industry 4.0: an architecture reference model. Adv Eng Inf, 2021, 47: 101225
15 Jiang X, Ge Z. Improving the performance of just-in-time learning-based soft sensor through data augmentation. IEEE Trans Ind Electron, 2022, 69: 13716–13726
16 Lee M, Bae J, Kim S B. Uncertainty-aware soft sensor using Bayesian recurrent neural networks. Adv Eng Inf, 2021, 50: 101434
17 Urhan A, Alakent B. Integrating adaptive moving window and just-in-time learning paradigms for soft-sensor design. Neurocomputing, 2020, 392: 23–37
18 He B, Zhang Q, Zhang X. A faster dynamic feature extractor and its application to industrial quality prediction. IEEE Trans Ind Inf, 2023, 19: 6773–6784
19 Wang K, Shang C, Liu L, et al. Dynamic soft sensor development based on convolutional neural networks. Ind Eng Chem Res, 2019, 58: 11521–11531
20 Liu S, Ji H, Wang M C. Nonpooling convolutional neural network forecasting for seasonal time series with trends. IEEE Trans Neural Netw Learn Syst, 2020, 31: 2879–2888
21 Dudukcu H V, Taskiran M, Cam Taskiran Z G, et al. Temporal convolutional networks with RNN approach for chaotic time series prediction. Appl Soft Comput, 2023, 133: 109945
22 Ren L, Wang T, Laili Y, et al. A data-driven self-supervised LSTM-DeepFM model for industrial soft sensor. IEEE Trans Ind Inf, 2022, 18: 5859–5869
23 Liu D, Wang Y, Liu C, et al. Data mode related interpretable transformer network for predictive modeling and key sample analysis in industrial processes. IEEE Trans Ind Inf, 2023, 19: 9325–9336
24 Yamak P T, Yujian L, Gadosey P K. A comparison between ARIMA, LSTM, and GRU for time series forecasting. In: Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence, 2020. 49–55
25 Lai G, Chang W C, Yang Y, et al. Modeling long- and short-term temporal patterns with deep neural networks. In: Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018. 95–104
26 Zhang Y, Chen G, Yu D, et al. Highway long short-term memory RNNs for distant speech recognition. In: Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016. 5755–5759
27 Liu Y, Wu H, Wang J, et al. Non-stationary transformers: exploring the stationarity in time series forecasting. In: Proceedings of the Advances in Neural Information Processing Systems, 2022. 9881–9893
28 Li S, Jin X, Xuan Y, et al. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In: Proceedings of the Advances in Neural Information Processing Systems, 2019
29 Zhou H, Zhang S, Peng J, et al. Informer: beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2021. 11106–11115
30 Zeng A, Chen M, Zhang L, et al. Are transformers effective for time series forecasting? In: Proceedings of the AAAI Conference on Artificial Intelligence, 2023. 11121–11128
31 Kleiman E M, Glenn C R, Liu R T. The use of advanced technology and statistical methods to predict and prevent suicide. Nat Rev Psychol, 2023, 2: 347–359
32 Zhang X, Kano M, Song Z. Optimal weighting distance-based similarity for locally weighted PLS modeling. Ind Eng Chem Res, 2020, 59: 11552–11558

33 Zhang X, Wei C, Song Z. Fast locally weighted PLS modeling for large-scale industrial processes. Ind Eng Chem Res, 2020, 59: 20779–20786
34 S. V V, Mohanta H K, Pani A K. Adaptive non-linear soft sensor for quality monitoring in refineries using just-in-time learning-generalized regression neural network approach. Appl Soft Comput, 2022, 119: 108546
35 Fu Y, Liu Z, Lyu J. Reason and discovery: a new paradigm for open set recognition. IEEE Trans Pattern Anal Mach Intell, 2025, 47: 5586–5599
36 Tao X, Koirala B, Rasti B, et al. A spectral-spatial attention network for hyperspectral unmixing. IEEE Trans Geosci Remote Sens, 2025, 63: 1–15
37 Xu Y, Wang H, Zhou F, et al. MambaHSISR: Mamba hyperspectral image super-resolution. IEEE Trans Geosci Remote Sens, 2025, 63: 1–16
38 Fu Y M, Liu Z G, Wang Z C. Logit prototype learning with active multimodal representation for robust open-set recognition. Sci China Inf Sci, 2024, 67: 162204
39 Ho J, Jain A, Abbeel P. Denoising diffusion probabilistic models. In: Proceedings of the Advances in Neural Information Processing Systems, 2020. 6840–6851
40 Zonta T, da Costa C A, da Rosa Righi R, et al. Predictive maintenance in the Industry 4.0: A systematic literature review. Comput Industrial Eng, 2020, 150: 106889
41 Tashiro Y, Song J, Song Y, et al. CSDI: conditional score-based diffusion models for probabilistic time series imputation. In: Proceedings of the Advances in Neural Information Processing Systems, 2021. 24804–24816
42 Rasul K, Seward C, Schuster I, et al. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In: Proceedings of the 38th International Conference on Machine Learning, 2021. 8857–8868
43 Dhariwal P, Nichol A. Diffusion models beat GANs on image synthesis. In: Proceedings of the Advances in Neural Information Processing Systems, 2021. 34: 8780–8794
44 Yuan X, Zhou J, Huang B, et al. Hierarchical quality-relevant feature representation for soft sensor modeling: a novel deep learning strategy. IEEE Trans Ind Inf, 2020, 16: 3721–3730
45 Nie Y, Nguyen N H, Sinthong P, et al. A time series is worth 64 words: long-term forecasting with transformers. In: Proceedings of the Eleventh International Conference on Learning Representations, 2023
46 Liu Y, Hu T, Zhang H, et al. iTransformer: inverted transformers are effective for time series forecasting. In: Proceedings of the International Conference on Learning Representations, 2024