International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 14 Issue 1 ǁ January 2026 ǁ PP. 115-119

Diffusion-Based time series generation with Transformer and gated temporal convolutions

Wei Liu
[*1]*School of Computer Science and Technology, Tongji University, Shanghai, China*
Corresponding Author: Wei Liu

Abstract
Time series data in real-world systems are often affected by noise, missing observations, and complex temporal dependencies, which pose significant challenges to accurate modeling and generation. To address these issues, this paper proposes a diffusion-based framework for time series generation from a probabilistic generative modeling perspective. The proposed framework integrates a Transformer encoder and a gated temporal convolutional network to jointly capture global and local temporal dependencies. Specifically, the input time series is first embedded with positional encoding and processed by a Transformer architecture to model long-range temporal correlations through multi-head self-attention, after which the encoded representations are decoded by a gated temporal convolutional network that employs dilated convolutions and gating mechanisms to capture multi-scale temporal patterns. Based on the decoded representations, a diffusion probabilistic model is constructed by learning a reverse denoising process from progressively noised data. During training, the model learns to recover clean time series from noisy inputs at different diffusion steps, while during inference, realistic time series samples are generated by iteratively denoising from pure noise. Experimental results demonstrate that the proposed method effectively models complex temporal dynamics and exhibits strong robustness under noisy and missing data scenarios.

Keywords: *Time series generation, Diffusion model.*

Date of Submission: 05-01-2026 Date of acceptance: 15-01-2026

I. INTRODUCTION

Time series data are widely encountered in real-world systems such as communication networks, industrial monitoring, finance, and energy management [1]. Accurate modeling and generation of time series are critical for tasks including forecasting, anomaly detection, data completion, and system simulation [2]. However, real-world time series often exhibit strong non-stationarity, complex temporal dependencies, and are frequently corrupted by noise or missing observations due to sensor failures, transmission errors, or environmental disturbances [3]. These characteristics significantly challenge conventional time series modeling methods.

Traditional statistical models, such as autoregressive and state-space approaches, rely on strong assumptions about linearity and stationarity, which limit their effectiveness in complex real-world scenarios [4]. With the development of deep learning, neural network based methods, including recurrent neural networks and temporal convolutional networks, have demonstrated improved representation capacity for temporal dynamics [5, 6]. Nevertheless, most existing deep learning approaches focus on deterministic prediction or point estimation and lack the ability to explicitly model the underlying data distribution and uncertainty, which is crucial for robust generation and completion tasks.

Recently, generative modeling has emerged as a promising paradigm for time series analysis. By learning the data distribution rather than producing single-point predictions, generative models can naturally capture uncertainty and generate diverse samples. Generative adversarial networks have been applied to time series generation, but their training instability and mode collapse issues limit their practical applicability [7]. In contrast, diffusion probabilistic models have shown remarkable performance and training stability in image generation by learning a gradual denoising process, making them an attractive alternative for generative modeling [8].

Motivated by these advances, this paper investigates diffusion-based generative modeling for time series data. However, directly applying diffusion models to time series remains challenging due to the need to simultaneously capture long-range temporal dependencies and fine-grained local temporal patterns. To address this challenge, we propose a diffusion-based framework that integrates a Transformer encoder and a gated temporal convolutional network. The Transformer module leverages multi-head self-attention to model global temporal correlations, while the gated temporal convolutional network captures local and multi-scale temporal structures through dilated convolutions and gating mechanisms. By combining these components within a

diffusion probabilistic framework, the proposed method provides a unified solution for time series generation and completion under noisy and missing data scenarios.

II. RELATED WORK

Time series modeling has been widely studied, with existing methods generally categorized into traditional statistical models, deep learning based deterministic approaches, and generative models. Traditional methods, such as autoregressive and state-space models, rely on assumptions of linearity and stationarity, which limit their effectiveness in complex and non-stationary real-world scenarios. Deep learning approaches, including recurrent neural networks, temporal convolutional networks, and Transformer-based models, have demonstrated strong capability in modeling temporal dependencies. However, most of these methods focus on deterministic prediction and do not explicitly model data uncertainty or generation mechanisms [9].

Generative models provide an alternative perspective by learning the underlying data distribution. While generative adversarial networks and variational autoencoders have been applied to time series generation, they often suffer from training instability or limited generation quality. Recently, diffusion probabilistic models have emerged as a promising generative framework due to their stable training and iterative denoising formulation [10]. Nevertheless, effectively capturing both global temporal dependencies and local multi-scale patterns in time series remains challenging. To address this issue, this work integrates diffusion modeling with a Transformer encoder and a gated temporal convolutional network to jointly model global and local temporal structures.

III. METHODOLOGIES

3.1 Problem definition

Let $X \in \mathbb{R}^{T \times F}$ denote a multivariate time series of length T with F feature dimensions, where $x_t \in \mathbb{R}^F$ represents the observation at time step t. In real-world scenarios, the observed time series is often corrupted by noise or contains missing values due to sensor failures, transmission errors, or external disturbances.

The objective of this work is to learn a probabilistic generative model that captures the underlying data distribution $p(X)$ of time series data, rather than producing deterministic point estimates. Such a model should be capable of generating realistic time series samples and reconstructing corrupted or incomplete observations in a unified framework.

Formally, given a partially observed or noisy time series $\tilde{X}$, the goal is to model the conditional distribution $p(X|\tilde{X})$ and recover the clean time series representation. By learning the data generation mechanism from a probabilistic perspective, the proposed approach enables both time series generation from noise and effective completion under missing or corrupted data conditions.

3.2 Input

Given a multivariate time series $X \in \mathbb{R}^{T \times F}$, each observation x_t is first mapped into a latent embedding space through a linear projection, producing an embedded sequence $E \in \mathbb{R}^{T \times d}$, where d denotes the embedding dimension. This transformation enables the model to represent heterogeneous input features in a unified latent space.

To retain temporal order information, positional encoding is incorporated into the embedded representations. By explicitly encoding time-step information, the positional encoding allows the model to distinguish different temporal positions and facilitates effective modeling of sequential dependencies. The resulting representations, obtained by combining value embeddings and positional encodings, are used as the input to the Transformer encoder. Figure 1 illustrates the overall architecture of the proposed framework.

3.3 Transformer encoder

Based on the embedded input representations obtained in Section 3.2, the Transformer encoder is employed to model global temporal dependencies in the time series. Let

$$H^{(0)} = E + P \in \mathbb{R}^{T \times d}, \tag{1}$$

denote the initial input to the Transformer encoder, where E is the value embedding and P represents the positional encoding.

The Transformer encoder updates the representations through stacked self-attention layers, allowing each time step to attend to all other time steps in the sequence. The output of the l-th Transformer layer is given by

$$H^{(l)} = \text{TransformerLayer}\big(H^{(l-1)}\big), \tag{2}$$

where the self-attention mechanism enables the model to capture long-range temporal correlations across the entire sequence.

Through this process, the encoder produces context-aware representations $H = H^{(L)}$ that integrate global temporal information, which are subsequently used for local modeling and generative learning.

3.4 Gated temporal convolutional network

Although the Transformer encoder effectively captures global temporal dependencies, its self-attention mechanism does not explicitly model local temporal continuity and multi-scale patterns. To complement global modeling, a gated temporal convolutional network (GTCN) is employed to capture local and multi-scale temporal dynamics.

Let $H \in \mathbb{R}^{T\times d}$ denote the context-aware representations produced by the Transformer encoder. The GTCN applies one-dimensional dilated convolutions along the temporal dimension to model local dependencies with different receptive fields. Specifically, two parallel convolutional branches are used to construct a gating mechanism:

$$\tilde{H} = \tanh\left(\mathrm{Conv}_f(H)\right), G = \sigma(\mathrm{Conv}_g(H)), \tag{3}$$

where $\mathrm{Conv}_f(\cdot)$ and $\mathrm{Conv}_g(\cdot)$ denote temporal convolution operations with dilation, $\tanh(\cdot)$ generates candidate features, and $\sigma(\cdot)$ is the sigmoid activation producing gating weights.

The final output of the gated temporal convolutional network is obtained by element-wise modulation:

$$Z = \tilde{H} \odot G, \tag{4}$$

where $\odot$ denotes element-wise multiplication. Through the gating mechanism, the network adaptively controls information flow across different temporal scales, enabling effective modeling of local temporal structures. The resulting representations Z serve as the input for subsequent diffusion-based generative modeling.

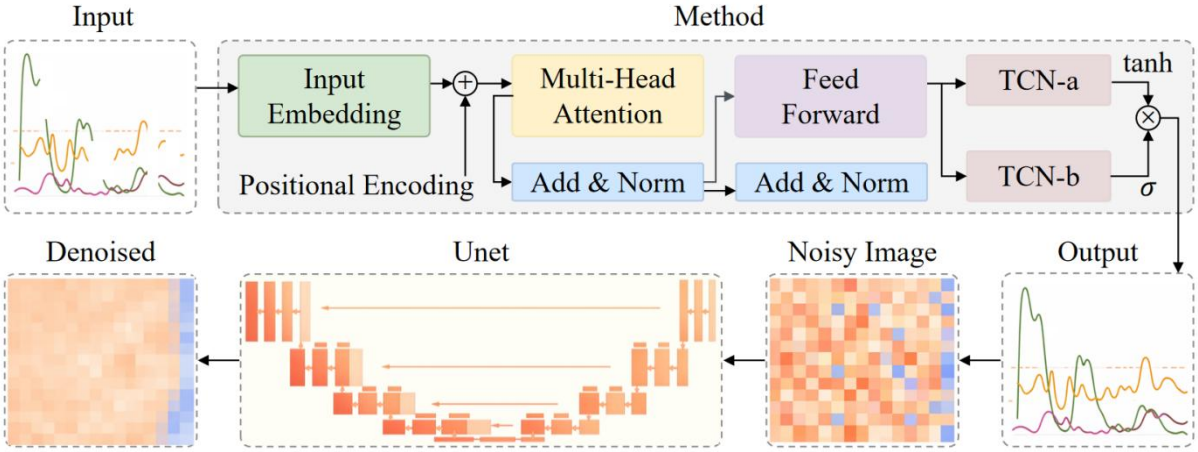

Figure1: Overall architecture of the proposed diffusion-based time series generation framework

3.5 Diffusion based generative modeling

Based on the locally enhanced representations produced by the gated temporal convolutional network, a diffusion probabilistic model is constructed to learn the generation mechanism of time series data. Let $Z \in \mathbb{R}^{T\times d}$ denote the output representation of the GTCN. To facilitate diffusion modeling, the sequential representation Z is first transformed into a two-dimensional structured representation through a sequence-to-image mapping operation, which is motivated by classical time-delay embedding techniques originating from the work of Takens [11].

$$I = \mathcal{T}(Z), \tag{5}$$

where $I \in \mathbb{R}^{H\times W}$ and $\mathcal{T}(\cdot)$ denotes a deterministic reshaping operation that preserves temporal structure.

The diffusion model defines a forward noising process on the image-like representation I, where Gaussian noise is progressively added according to a predefined noise schedule:

$$I_t = \sqrt{\alpha_t} I + \sqrt{1-\alpha_t}\epsilon, \epsilon \sim \mathcal{N}(0, I), \tag{6}$$

with $t \in \{1, \dots, T_d\}$ denoting the diffusion step and $\alpha_t \in (0,1)$ controlling the noise level.

The reverse process aims to recover the clean representation by learning a denoising function parameterized by a neural network:

$$\hat{\epsilon}_t = f_\theta(I_t, t), \tag{7}$$

where $f_\theta(\cdot)$ predicts the injected noise at each diffusion step. During training, the model is optimized by minimizing the denoising loss:

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{t,\epsilon}[|\epsilon - f_\theta(I_t, t)|_2^2], \tag{8}$$

which encourages accurate noise estimation across different diffusion steps.

During inference, the generation process starts from pure Gaussian noise and iteratively applies the learned denoising function in reverse order, producing a clean image-like representation $\hat{I}$. The generated

representation is then transformed back to the sequential domain via the inverse mapping $\mathcal{T}^{-1}(\cdot)$, yielding the final time series output. Through this diffusion-based generative modeling, the proposed framework enables both realistic time series generation and effective reconstruction under noisy or missing data scenarios.

IV. EXPERIMENTS

4.1 Experimental setup

This section conducts a systematic evaluation of the proposed generative method under irregularly observed time series settings. The experiments cover multiple real-world and synthetic time series datasets, aiming to examine the ability of generative models to capture data distributions and dynamic structures in the presence of missing observations. The experiments are performed on four representative multivariate time series datasets, including Weather, Electricity, Energy, and Stocks. These datasets exhibit diverse characteristics in terms of temporal dynamics, noise levels, and inter-variable correlations, covering a wide range of typical patterns such as dominant periodicity, trend variations and stochastic fluctuations. To characterize modeling difficulty under incomplete observations, the experiments are conducted in irregular time series settings by randomly discarding a fixed proportion of observations from each sequence. Specifically, observation drop rates of 30%, 50%, and 70% are considered, corresponding to different levels of data incompleteness. All methods are evaluated under identical data splits, missing-rate settings, and training budgets to ensure fair comparisons.

4.2 Evaluation metric

To quantitatively assess the quality of generated time series under irregular observation settings, this work adopts a discriminative evaluation protocol, which measures the distinguishability between real and generated samples. This metric evaluates whether the generated sequences match the statistical characteristics of real data, and has been widely used as an indirect but effective criterion for generative time series modeling.

Specifically, a binary discriminator $D(\cdot)$ is constructed to classify input time series samples as either real or generated. Real and generated samples are mixed and then split into training and test sets. The discriminator is trained on the training set and evaluated on the test set, yielding a classification accuracy denoted as Acc.

To provide a meaningful reference point corresponding to the case where real and generated samples are indistinguishable, the discriminative score is defined as

$$\mathrm{Dis} = |\mathrm{Acc} - 0.5|. \tag{9}$$

When the generated samples closely match the real data distribution, the discriminator performs no better than random guessing, resulting in $\mathrm{Acc} \approx 0.5$ and thus $\mathrm{Dis} \approx 0$. Conversely, a larger discriminative score indicates that the generated samples are easier to distinguish from real data, implying greater distributional discrepancy. Therefore, a lower discriminative score corresponds to higher generation quality.

4.3 Results analysis

In this subsection, the proposed method is compared with representative baseline models using the discriminative evaluation protocol described in Section 4.2. All methods are evaluated under identical data splits, missing-rate settings, and discriminator configurations to ensure fair comparison. The discriminative scores under 30%, 50%, and 70% observation drop rates are jointly reported in Table 1.

Table 1: Discriminative scores under different drop rates (best in bold).

Drop Rate	Method	Weather	Electricity	Energy	Stocks
30%	GT-GAN	0.472	0.422	0.332	0.252
	TimeGAN	0.495	0.497	0.454	0.466
	RCGAN	0.493	0.498	0.500	0.441
	OURS	**0.139**	**0.420**	**0.184**	**0.107**
50%	GT-GAN	0.497	0.396	0.314	0.263
	TimeGAN	0.500	0.498	0.483	0.487
	RCGAN	0.500	0.498	0.500	0.475
	OURS	**0.157**	**0.360**	**0.164**	**0.077**
70%	GT-GAN	0.479	0.481	0.330	0.230
	TimeGAN	0.500	0.500	0.496	0.491
	RCGAN	0.498	0.500	0.500	0.380
	OURS	**0.216**	**0.401**	**0.225**	**0.080**

As discussed earlier, a lower discriminative score indicates that the generated samples are statistically closer to real data and thus more difficult to distinguish. When the observation drop rate is relatively low (30%), most methods are able to preserve a certain level of distributional consistency, although noticeable performance differences already emerge across different generative paradigms. As the drop rate increases to 50% and further to 70%, the discriminative scores of many baseline methods increase substantially, indicating that their generated samples deviate more clearly from the real data distribution under more severe incompleteness.

In contrast, the proposed method consistently achieves lower discriminative scores across all datasets and missing-rate settings. Moreover, its performance exhibits a more gradual degradation trend as the observation drop rate increases, suggesting improved stability under increasingly incomplete observation conditions. These results indicate that the proposed framework more effectively captures the statistical characteristics of time series data when observations are sparse.

Further dataset-level analysis reveals that different temporal dynamics pose distinct challenges for generative modeling. Datasets dominated by smooth or regular patterns tend to expose subtle artifacts such as phase shifts or amplitude distortions, while datasets characterized by strong stochastic fluctuations are more sensitive to over-smoothing or unrealistic variability. Despite these challenges, the proposed method maintains competitive or superior discriminative performance across diverse data characteristics, demonstrating its effectiveness in modeling time series distributions under irregular observation scenarios.

V. CONCLUSIOSN

This paper presents a diffusion-based generative framework for modeling multivariate time series under irregular observation settings. By integrating a Transformer encoder for global dependency modeling, a gated temporal convolutional network for local and multi-scale structure extraction, and a diffusion probabilistic process operating on structured representations, the proposed method provides a unified approach to time series generation in the presence of missing observations. Experimental results based on discriminative evaluation across multiple real-world datasets and varying observation drop rates demonstrate that the proposed framework consistently generates samples that are harder to distinguish from real data, and exhibits more stable performance as the degree of incompleteness increases. These results indicate that the proposed method effectively captures the underlying statistical characteristics of incomplete time series data, offering a promising direction for generative modeling under irregular observation conditions.

REFERENCES

[1]. Box, G. E. P., Jenkins, G. M., Reinsel, G. C. and Ljung, G. M. “Time Series Analysis: Forecasting and Control” Wiley, Hoboken, NJ, pp. 1–712 (2015).
[2]. Hyndman, R. J. and Athanasopoulos, G. (2018) “Forecasting: Principles and Practice” OTexts, Melbourne, pp. 1–352.
[3]. Kalman, R. E. (1960) “A new approach to linear filtering and prediction problems” Journal of Basic Engineering, Vol. 82, No. 1, pp. 35–45.
[4]. Hochreiter, S. and Schmidhuber, J. (1997) “Long short-term memory” Neural Computation, Vol. 9, No. 8, pp. 1735–1780.
[5]. Bai, S., Kolter, J. Z. and Koltun, V. (2018) “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling” arXiv preprint arXiv:1803.01271.
[6]. Yoon, J., Jordon, J. and van der Schaar, M. (2019) “TimeGAN: Generating realistic time series with generative adversarial networks” Advances in Neural Information Processing Systems, Vol. 32, pp. 5508–5518.
[7]. Ho, J., Jain, A. and Abbeel, P. (2020) “Denoising diffusion probabilistic models” Advances in Neural Information Processing Systems, Vol. 33, pp. 6840–6851.
[8]. Papoulis, A. and Pillai, S. U. (2002) “Probability, Random Variables, and Stochastic Processes” McGraw-Hill, New York, pp. 1–852.
[9]. Vaswani, A., Shazeer, N., Parmar, N. *et al.* (2017) “Attention is all you need” Advances in Neural Information Processing Systems, Vol. 30, pp. 5998–6008.
[10]. Song, Y. and Ermon, S. (2019) “Generative modeling by estimating gradients of the data distribution” Advances in Neural Information Processing Systems, Vol. 32, pp. 11895–11907.
[11]. Takens, F. (2006) “Detecting strange attractors in turbulence” *Dynamical Systems and Turbulence, Warwick 1980: Proceedings of a Symposium Held at the University of Warwick 1979/80*, Springer, Berlin, pp. 366–38