TCDM: A Temporal Correlation-Empowered Diffusion Model for Time Series Forecasting

Huibo Xu[1] , **Likang Wu**[2] , **Xianquan Wang**[1] , **Zhiding Liu**[1] , **Qi Liu**[1,*]
[1]University of Science and Technology of China
[2]Tianjin University
{xhbxhb, wxqcn, zhiding}@mail.ustc.edu.cn, wulk@tju.edu.cn, qiliuql@ustc.edu.cn

Abstract

Although previous studies have applied diffusion models to time series forecasting, these efforts have struggled to preserve the intrinsic temporal correlations within the series, leading to suboptimal predictive outcomes. This failure primarily results from the introduction of independent, identically distributed (i.i.d.) noise. In the forward process, the addition of i.i.d. noise to the time series gradually diminishes these temporal correlations. The reverse process starts with i.i.d. noise and lacks priors related to temporal correlations, which can result in directional biases during sampling. From a frequency-domain perspective, noise similarly disrupts the low-frequency-dominated structure of trend components, making it difficult for the model to learn long-term temporal dependencies. To address these limitations, we introduce a decomposition prediction framework to complement the novel Temporal Correlation-Empowered Diffusion Model. Overall, We decompose the time series into trend and residual components, predict them using a base model and a diffusion model, and then combine the results. Specifically, a frequency-domain MLP model was adopted as the base model due to its not distorting the original sequence, and better the capture of long-range temporal dependencies. The diffusion model incorporates two key modules to capture short- and mid-range temporal correlations: the Maintaining Temporal Correlation Module and the Redesigned Initial Module. Extensive experiments across multiple datasets demonstrate that the proposed method significantly outperforms related strong baselines.

1 Introduction

Time series forecasting is a fundamental task that is crucial in many areas, such as financial markets [Aczel and Josephy, 1991; Jiang *et al.*, 2023] and weather systems [Shen *et al.*, 2021]. Specifically, time series are sequences composed of points that are temporally correlated and numerically related. As an essential characteristic of time series, **temporal correlation**, also known as temporal dependence, refers to the relationship between different points over time [Liu *et al.*, 1997; Wu *et al.*, 2024] and is crucial for making accurate predictions [Zeng *et al.*, 2023; Li *et al.*, 2024]. Moreover, Time series forecasting can be viewed as a generative task that generates a prediction sequence conditioned on a historical sequence. As state-of-the-art generative models, diffusion models—known for their remarkable success in fields like computer vision [Dhariwal and Nichol, 2021; Kawar *et al.*, 2021] and computational chemistry [Anand and Achim, 2022]—have been naturally introduced to the task of time series forecasting.

Recent research on diffusion models for time series forecasting [Rasul *et al.*, 2021; Tashiro *et al.*, 2021; Shen and Kwok, 2023; Li *et al.*, 2023] has shown promising results. Nevertheless, as will be empirically demonstrated in our experimental section 4.3, diffusion methods often **fail to effectively preserve the intrinsic temporal dependencies** of time series [Ma *et al.*, 2024], naturally leading to a decline in modeling performance. This shortcoming can be attributed to the independent and identically distributed (i.i.d.) Gaussian noise. Due to stochasticity, in the forward process, adding Gaussian noises over too many diffusion steps makes each time point approach independent Gaussian distribution, which can intuitively result in the disappearance of temporal correlation. In the reverse processes, these models typically start from i.i.d. noise, which lacks prior knowledge of temporal dependencies during sampling, resulting in generated time series with insufficient temporal correlation.

Meanwhile, due to the aforementioned properties of the diffusion model, not all components of time series data are well-suited for prediction using diffusion model [Selesnick *et al.*, 2014]. In particular, the trend component, extracted using the classical Moving Average method [Chen *et al.*, 2021], exhibits smooth variations, with energy primarily concentrated in the low-frequency domain, thereby reflecting long-term temporal dependencies [Zeng *et al.*, 2023; Chen *et al.*, 2021]. However, **the addition of noise can disrupt its temporal correlation**. From a frequency domain perspective, adding noise with a uniform frequency distribution is equivalent to introducing significant high-frequency energy into a predominantly low-frequency signal [Walters and Heston, 1982]. This interference hinders the model's ability to learn low-frequency features, thereby affecting the output's **long-term temporal dependencies**.

To address these challenges, we introduce a decomposition prediction framework to complement the Temporal Correlation-Empowered Diffusion Model, together referred to as TCDM. The framework adopts the divide-and-conquer approach. It begins by decomposing the time series into two components: a trend component and a residual component. For the trend component, characterized by long-term temporal correlations and energy concentrated in the low-frequency domain, we utilize a frequency-domain MLP-based model. This model more easily learns compact low-frequency representations, thereby enhancing its ability to effectively capture long-term temporal dependencies. The residual component, containing significant noise and seasonal information, primarily reflecting short- and mid-range temporal dependencies, is processed using a diffusion model. The diffusion model is utilized for its strong generative and denoising capabilities, while the residual component's broad spectral distribution ensures stable resistance to noise during diffusion training.

Specifically, during the forward process of diffusion, we introduce the Maintaining Temporal Correlation Module. This module is designed to handle the noise added at each time step ensuring that the model can still learn temporal correlations even at larger diffusion step. Concurrently, during the reverse process sampling, we specifically design the variance to continuously steer towards preserving temporal correlations. Meanwhile, fewer diffusion steps in the reverse process help maintain the internal temporal correlations of the data more effectively. By redesigning the initial state of the reverse process, achieved by truncating the diffusion process, we enhance the preservation of internal temporal correlations. This strategy also reduces the number of steps required in both the forward and reverse process.

In this work,we introduce a decomposition-based predictive framework to complement the novel temporal correlation-empowered diffusion model for multivariate time series forecasting. Our experiments across various real-world datasets demonstrate that the model achieves state-of-the-art performance. Our contributions are summarized as follows:

- We propose an innovative decomposition framework for multivariate time series forecasting that integrates a base model with a diffusion model to capture long-term and short-term temporal dependencies separately.
- We focus on the overall design of the temporal correlation in the diffusion model, which is divided into three key aspects: noise addition, initial state, and the sampling process. Each element is strategically implemented to improve the preservation of temporal correlations throughout the model.
- We demonstrate the state-of-the-art performance of proposed method through experiments on various real-world multivariate time series forecasting datasets.

2 Background

2.1 Denoising diffusion probabilistic models

Diffusion models comprise a forward diffusion process and a backward denoising process. Beginning with the widely recognized denoising diffusion probabilistic model [Ho *et al.*, 2020]. The data distribution $x_0 \sim q(x_0)$ is defined along with a Markovian noising process q that gradually adds noise to the data x_0 to produce noised samples x_T. Specifically, each step of the noising process adds Gaussian noise according to a variance schedule given by β_t:

$$q(x_t \mid x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}). \tag{1}$$

Furthermore, $q(x_t \mid x_0)$ can be expressed as a Gaussian distribution. With $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=0}^{t} \alpha_s$,

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{2}$$

It is sufficient to train a neural network to predict a mean μ_θ:

$$p_\theta(x_{t-1} \mid x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \tag{3}$$

To train this model such that $p(x_0)$ learns the true data distribution $q(x_0)$, the model can be trained by optimizing the variational lower bound L_{vlb} for $p_\theta(x_0)$. Training $\mu_\theta(x_t, t)$ and training $\epsilon_\theta(x_t, t)$ are equivalent [Luo, 2022]. They demonstrated that predicting the noise added at each step is effectively the same as predicting x_0 at each step. Rather than directly parameterizing $\mu_\theta(x_t, t)$ as a neural network, the methodology involves training the model $\epsilon_\theta(x_t, t)$ to predict the noise ϵ. The corresponding simplified objective is given by:

$$L_{\text{simple}} := E_{t,x_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right], \tag{4}$$

The mean $\mu_\theta(x_t, t)$ can also be derived using a denoising network $\mathbf{x}_\theta$, which estimates the clean data $\mathbf{x}_0$ given $\mathbf{x}_t$. This estimate $\mathbf{x}_\theta(\mathbf{x}_t, t)$ allows the following expression for $\mu_\theta(x_t, t)$ to be set:

$$\mu_\theta(x_t, t) = \frac{\sqrt{\alpha_t}\,(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_\theta(x_t, t), \tag{5}$$

The parameter θ is then optimized by minimizing the loss function as below:

$$L_{\text{simple}} := E_{t,x_0,\epsilon}\left[\|x_0 - x_\theta(x_t, t)\|^2\right]. \tag{6}$$

3 Method

The key components of our model include: (i) Decomposition of Time Series, (ii) Maintaining Temporal Correlation Module, and (iii) Redesigned Initial Module.

3.1 Problem Formulation

When using a diffusion model for time series forecasting, our objective is to predict future values $\mathbf{y}^0_{1:F} \in R^{C\times F}$ based on the observed historical data $\mathbf{x}_{1:H} \in R^{C\times H}$, where $\mathbf{x}_{1:H}$ (denoted as $\mathbf{X}$) is defined as $\{x_1, x_2, \ldots, x_H \mid x_t \in R^C\}$. and $\mathbf{y}^0_{1:F}$ (denoted as $\mathbf{Y}^0$) is defined as $\{y^0_1, y^0_2, \ldots, y^0_F \mid y^0_t \in R^C\}$. The superscript t in $\mathbf{y}^t_{1:F}$ denotes that these values are at the t-th step of diffusion. Thus, $\mathbf{y}^0_{1:F}$, the superscript 0 indicates that they are at the 0-th step of the diffusion process, meaning these are the true values before any noise is added. The conditioning of diffusion models for time-series [Shen and Kwok, 2023] can be done by conditioning the generation

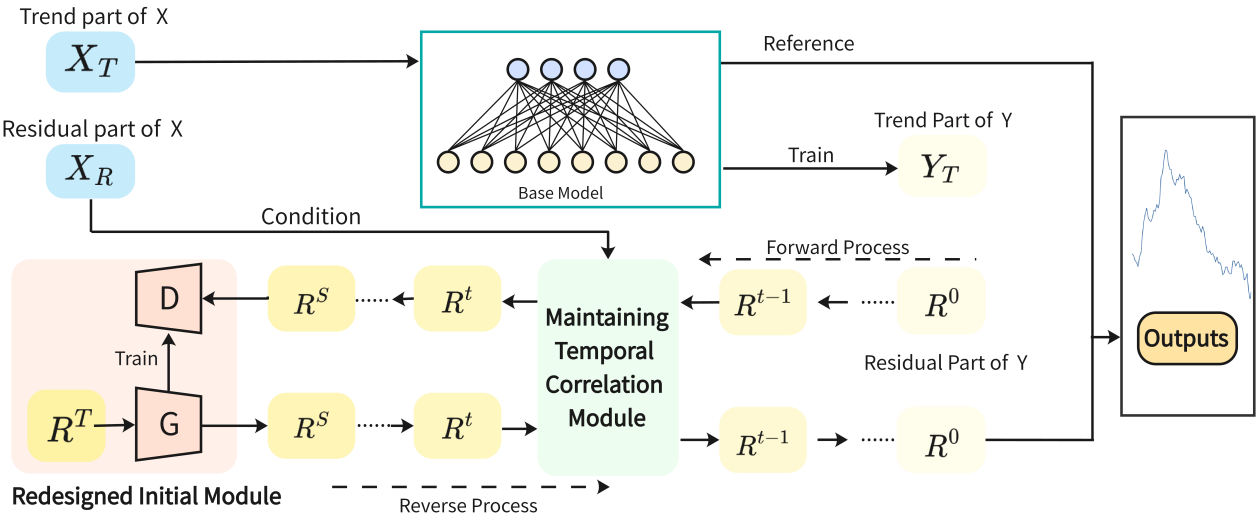


Figure 1: An illustration of the proposed TCDM: The lookback window $\mathbf{X}$ is decomposed into $\mathbf{X}_T$ and $\mathbf{X}_R$, while the target $\mathbf{Y}^0$ is decomposed into $\mathbf{Y}_T^0$ and $\mathbf{Y}_R^0$.

of the forecast through the reverse process on historical data, as follows:

$$p_\theta\left(\mathbf{y}_{1:F}^{0:T} \mid \mathbf{c}\right) = p\left(\mathbf{y}_{1:F}^{T} \mid \mathbf{c}\right) \prod_{t=1}^{T} p_\theta\left(\mathbf{y}_{1:F}^{t-1} \mid \mathbf{y}_{1:F}^{t}, \mathbf{c}\right), \quad (7)$$

where c is the condition, $\mathbf{c} = \mathcal{F}\left(\mathbf{x}_{1:H}\right)$ and $\mathcal{F}$ is a conditional network.

3.2 Overview of TCDM

As depicted in Figure 1, during the training process, the target $\mathbf{Y}^0$ is initially decomposed into the trend component $\mathbf{Y}_T^0$ and the residual component $\mathbf{Y}_R^0$, also denoted as $\mathbf{R}^0$. Similarly, $\mathbf{X}$ is decomposed into $\mathbf{X}_T$ and $\mathbf{X}_R$. The Base Model employs $\mathbf{X}_T$ to predict the trend component $\mathbf{Y}_T^0$. Concurrently, the residual component $\mathbf{R}^0$ is predicted using a diffusion model conditioned on $\mathbf{X}_R$. The diffusion model integrates the Maintaining Temporal Correlation Module and the Redesigned Initial Module. The predictive outputs from diffusion model are combined with those from the Base Model to generate the final results.

3.3 Decomposition of Time Series

Given that time series are generally modeled as a combination of trend, seasonal, and noise components, we refer to the seasonal and noise components as the residuals. The addition of noise can disrupt the intricate relationships within the data. To address this, we extract the more predictable trend component using the classical seasonal-trend decomposition technique described by wu2021autoformer.

$$y_i^0 = \tilde{y}_i^0 + \Delta y_i^0, \quad (8)$$

the data point at the ith time step is denoted by y_i^0. The trend component, represented as $\tilde{y}_i^0$, is computed as a moving average within a fixed-length time window, effectively capturing the underlying trend of the time series. The residual component, denoted by Δy_i^0, captures the cyclical variations and

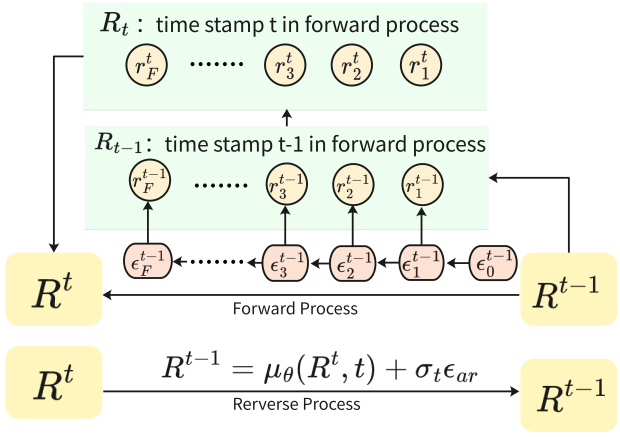


Figure 2: Maintaining Temporal Correlation Module: This module features an autoregressive noise design that introduces noise to various data points within the same diffusion step during the forward process. The sampling process aimed at temporally correlated directions is illustrated in the lower half of the figure.

the influence of noise after the trend has been removed. Similarly, the lookback window $\mathbf{X}$ is also decomposed. The model uses the trend part of $\mathbf{X}$ to predict the trend part of the target $\mathbf{Y}^0$, and the residual part of $\mathbf{X}$ to predict the residual part of $\mathbf{Y}^0$. This approach help to ensure consistency between the training and prediction data.

Base Model Selection

The moving average can be viewed as the convolution of the input signal Y^0 with a uniform filter $h[n]$. The uniform filter is defined as:

$$h[n] = \begin{cases} \frac{1}{M}, & -\frac{M-1}{2} \leq n \leq \frac{M-1}{2} \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

Therefore, the output of the moving average can be expressed as:

$$\tilde{y}_n^0 = (Y*h)[n] = \sum_{k=-\infty}^{\infty} y_k^0 \cdot h[n-k] = \frac{1}{M} \sum_{k=-(M-1)/2}^{(M-1)/2} y_{n-k}^0 \tag{10}$$

Proposition 1. *A symmetric moving average filter in the time domain is equivalent to a low-pass filter in the frequency domain.*

The proof is provided in the ***Appendix***.

In the frequency domain, trend components are characterized by energy concentrated in the low-frequency region of the spectrum. When training a frequency-domain-based MLP, the input features primarily capture this valuable low-frequency information, allowing the model to effectively learn these features. Therefore, theoretically, a frequency-domain-based MLP model is more suitable for trend component forecasting. Experimentally, as shown in the ***Appendix***, we compare the predictive performance of a frequency-domain-based MLP, a time-domain-based MLP, and a transformer-based model. The results demonstrate that the frequency-domain-based MLP outperforms the others. The detailed structure of the Base Model is also provided in the Appendix.

After obtaining the prediction model for the trend component, we proceeded to design a diffusion model for predicting the residual component. Due to the complexity of these cyclical patterns and the uncertainty of noise, the residual component is a key factor that affects the accuracy of time series predictions.

3.4 Maintaining Temporal Correlation Module

We use the Residual part of $\mathbf{Y}$: $\mathbf{R}^0$ as the initial state of the forward process in the diffusion model,thus it also serves as the output of the diffusion model. Then for the transition equation utilized during sampling, $p_\theta(R^{t-1} \mid R^t, R_0) = \mathcal{N}(R^{t-1}; \mu_\theta(R^t, t), \Sigma_\theta(R^t, t))$, where $\mu_\theta(R^t, t)$ is formulated as $\frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} R_t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t} R_\theta(R_t, t)$, with $R_\theta(R_t, t)$ representing the trained diffusion network. Our Temporally Correlated Noise is specifically designed for $\mu_\theta(R^t, t)$, and accordingly, the Sampling Towards Temporal Correlation is tailored for $\Sigma_\theta(R^t, t)$.And details on the initial state of sampling presented in the Redesigned Initial Module.

Temporal Correlated Noise in Forward Process

For the target data $\mathbf{R}^t = \mathbf{r}_{1:F}^t$, composed of time points $r_1^t, r_2^t, \ldots, r_F^t$, during the forward diffusion process, the noise added at diffusion step t to the i th data point r_i^{t-1} is denoted by ϵ_i^t:

$$r_i^t = \sqrt{1-\beta_t} r_i^{t-1} + \beta_t \epsilon_i^{t-1}. \tag{11}$$

We propose the introduction of temporally correlated noise during the forward process, enabling the model to learn more temporal correlations even for larger diffusion step t. That means $\mu_\theta(R^t, t)$ will exhibit temporal correlation during the sampling process.

Our noise design emulates the method of noise addition used in diffusion models as referenced in Equation 1. We have modeled the noise as a Markov process (first-order autoregressive process) where the noise for the next time point i is based on the noise at the current time point $i-1$, with new random variations introduced at each subsequent time point. The noise added at time t to the ith timestep data point r_i^{t-1} is denoted by ϵ_i^t.

$$\begin{aligned} &\epsilon_i^t := \sqrt{\lambda^i}\epsilon_{i-1}^t + \sqrt{1-\lambda^i} b_i, \\ &b_i \sim \mathcal{N}(0,1), \quad \epsilon_0^t \sim \mathcal{N}(0,1). \end{aligned} \tag{12}$$

The variance design mentioned above ensures that $\mathrm{Var}(\epsilon_i^t)$ remains constant at 1, ensuring that the noise level added at each time point within the target window is the same as i increases.

This design ensures that as the diffusion step t increases, a correlation is maintained within the same target window , even when t becomes large. Although this correlation is intentionally engineered, when λ^i is set appropriately, the Temporal Correlated Noise forces the model to preserve local correlations between consecutive data points, enabling the model to automatically adjust to the correct correlations.

Sampling Towards Temporal Correlation

Following the precedent set by [Ho *et al.*, 2020], $\Sigma_\theta(R^t, t)$ is typically configured as $\sigma_t^2 \mathbf{I}$, then

$$R^{t-1} = \mu_\theta(R^t, t) + \sigma_t z^t, \quad z^t \sim \mathcal{N}(0, I). \tag{13}$$

This formulation models the transition from R^t to R^{t-1}, with the noise variance governed by σ_t^2. Our objective in the reverse process is to impart temporal correlation priors to the model. Our optimization goal is to minimize the KL divergence $D_{KL}(q(R_{t-1}|R_t, R_0) \| p_\theta(R_{t-1}|R_t))$. To echo the noise in the forward process, during the sampling phase, we adjust the z^t values to facilitate sampling in a direction that respects the sequence correlations. Consequently, we define:

$$z^t = \epsilon_{ar} = [\epsilon_0^t, \epsilon_1^t, \ldots, \epsilon_F^t], \tag{14}$$

where ϵ_i^t is sampled by Equation 12, transforming the sampling equation as follows:

$$R^{t-1} = \mu_\theta(R^t, t) + \sigma_t \epsilon_{ar}, \tag{15}$$

this approach ensures that the sampling process is optimally aligned with the inherent temporal dynamics of the sequence.

As illustrated in Figure 3, the top panel presents the scenario 'without Maintaining Temporal Correlation Module', where the sampled $\mu_\theta(R^t, t)$ can exhibit significant fluctuations due to the absence of guidance on temporal correlations. In this scenario, z^t adheres to a standard normal distribution, leading to a spherical sampling range that signifies indiscriminate sampling across all directions. Conversely, the bottom panel, labeled 'with Maintaining Temporal Correlation Module', depicts $\mu_\theta(R^t, t)$ being influenced by noise specifically tailored for temporal correlations, which directs the sampling towards R^{target}. The design of ϵ_{ar} in this scenario makes the sampling range elliptical, enhancing sampling along the axes of temporal correlations. Thus, the final sampled outcome R^0 is more likely to closely approximate R^{target}, resulting in superior generative results.

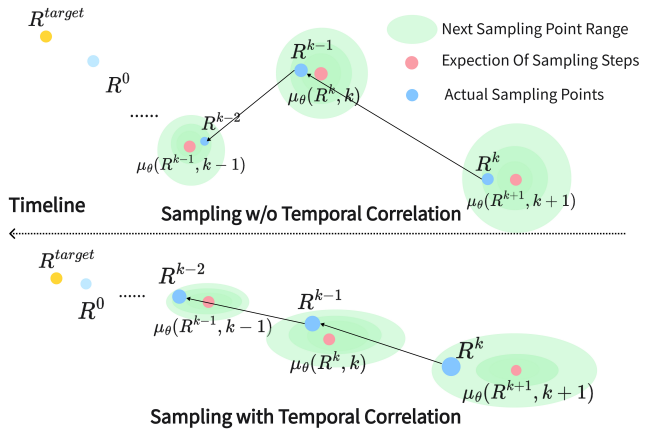


Figure 3: Diagram of Sampling Towards Temporal Correlation, where proximity to the upper-left corner indicates closer alignment with the temporal correlations of the target time series.

3.5 Redesigned Initial Module

Instead of relying on manually designed functions, we suggest halting the diffusion process before it reaches a state close to pure noise and then initiating the reverse process from that point. By doing so, the initial state of the reverse process retains the natural temporal correlations inherent in the time series, rather than those imposed by artificial design.

When truncating the diffusion process at time S, the forward process is halted at $\mathbf{R}^S$. Therefore, we maintain the original noise schedule $\{\beta_1, \beta_2, \ldots, \beta_T\}$ and select the subset $\{\beta_1, \beta_2, \ldots, \beta_{S+1}\}$ as the new noise schedule[Zheng *et al.*, 2022]. This approach retains the remaining original diffusion process. Since $\mathbf{R}^S$ has not fully diffused into a noise-dominated state, reverse sampling from Gaussian noise is infeasible. To address this issue, a generator is necessary to obtain the data distribution at S. This is accomplished using a Generative Adversarial Network (GAN) [Xiao *et al.*, 2021].

During the reverse process, a generator G_ϕ is required to produce $\mathbf{R}^S$, such that $\mathbf{R}^S = G_\phi(z)$, where $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. If we approximate $q(\mathbf{R}^{S+1} \mid \mathbf{R}^S) \approx q(\mathbf{R}^T)$ and $q(\mathbf{R}^T) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, then

$$\begin{aligned} G_\phi(\mathbf{R}^T) = \mathbf{R}^S = & \frac{\sqrt{\alpha_{S+1}}\,(1-\bar{\alpha}_S)}{1-\bar{\alpha}_{S+1}} R_T \\ & + \frac{\sqrt{\bar{\alpha}_s}\beta_{S+1}}{1-\bar{\alpha}_{S+1}} R_\theta(R_T, S+1). \end{aligned} \tag{16}$$

This can be interpreted as enabling the diffusion model to use $\mathbf{R}^T$ to reach $\mathbf{R}^S$ with one step of the reverse process[Wang *et al.*, 2024]. Therefore, the optimization objective is to minimize the KL divergence between the prior distribution of the forward process $q(\mathbf{R}^S)$ and the posterior distribution $p_\phi(\mathbf{R}^S)$, where $\mathbf{R}^S$ is generated in one step by the generator

$$L^S(\phi) := \mathcal{D}_{KL}\left(q(\mathbf{R}^S) \| p_\phi(\mathbf{R}^S)\right). \tag{17}$$

Referring to Equation 6, since the preserved parts of the diffusion process remain unchanged, our new loss function is defined as:

$$L_{\text{GD}} := \sum_{t=1}^{S-1} L^t(\theta) + L^S(\phi). \tag{18}$$

This represents the final version of the loss function for the diffusion model, where $L^t(\theta)$ is set as the version predicting the initial value R^0. The generator G_ϕ is adversarially optimized using a discriminator D_ψ, details of D_ψ are provided in the appendix. The optimization objective $\mathcal{L}^S(\phi)$is as follows:

$$\begin{aligned} & \min_\phi \max_\psi E_{\mathbf{R}^S}[\log D_\psi(\mathbf{R}^S)] \\ & + E_{\mathbf{R}^S}[\log(1 - D_\psi(G_\phi(\mathbf{R}^T)))]. \end{aligned} \tag{19}$$

Therefore, the denoising network of the diffusion model is trained using $\mathbf{R}^t$, with the diffusion step t as input and $\mathbf{X}_R$ as a condition, aiming to output $\mathbf{R}^{t-1}$. This denoising network also acts as the generator $\mathbf{G}$ for the reverse process, hence during training, we employ adversarial training of the denoising network and the discriminator $\mathbf{D}$. Once trained, during inference, Gaussian noise $\mathbf{R}^T$ is inputted into $\mathbf{G}$, which generates predictions $\mathbf{R}^S$. These are then iteratively sampled through the Maintaining Temporal Correlation Module to eventually produce the predicted $\mathbf{Y}_R$, which, when combined with the Base Model's predicted $\mathbf{Y}_T$, yields the final outputs.

4 Experiments

4.1 Experiment Setup

We conducted experiments on 9 public datasets [Meijer and Chen, 2024], including NorPool, Caiso, Traffic, Electricity, Weather, Exchange, ETTh1, ETTm1, and Wind. Each of these datasets comprises multivariate time series. Detailed descriptions and Implementation Details are provided in ***Appendix***. The history length, chosen from the set of (96, 192, 336, 720), and 1440 based on the validation set performance, informs the model's temporal context. A prediction length of 168 was selected for the ETTm1, Wind, Traffic, Electricity, ETTh1, and Exchange datasets. For the NorPool and Caiso datasets, a longer prediction length of 720 was opted to accommodate their unique temporal dynamics.

Evaluation Metrics

Following previous studies, the experiments utilized Mean Squared Error (MSE) to measure the predictive performance of the models. To ensure the results' convincing nature, we averaged the results over 10 runs for each experiment.

4.2 Overall Performance

Table 1 shows the results of multivariate predictions. Non-autoregressive diffusion models TimeDiff for time series have strong predictive abilities, suggesting that diffusion models are promising. Our method outperforms not only existing diffusion-based ones but also other baselines. It shows significant improvement on complex datasets like ETTm1 and ETTh1. Time series decomposition helps the model understand the data better. However, for Casio and Traffic, the obvious periodicity of the sequences may cause additional

	NorPool	Caiso	Weather	ETTm1	Wind	Traffic	Electricity	ETTh1	Exchange	Avg Rank
Our Model	**0.655 (1)**	0.132 (2)	**0.211 (1)**	**0.323 (1)**	0.881 (2)	0.445 (2)	**0.161 (1)**	**0.396 (1)**	**0.160 (1)**	**1.333 (1)**
TimeDiff	0.664 (2)	0.138 (4)	0.219 (4)	0.330 (3)	**0.875 (1)**	0.449 (3)	0.172 (5)	0.405 (2)	0.169 (4)	3.111 (2)
TimeGrad	1.105 (14)	0.242 (12)	0.308 (13)	0.479 (11)	1.052 (11)	0.674 (9)	0.253 (12)	0.612 (12)	0.290 (9)	11.444 (12)
CSDI	0.801 (10)	0.191 (7)	0.280 (9)	0.477 (10)	1.045 (10)	-	-	0.497 (8)	0.261 (7)	8.714 (8)
SSSD	0.748 (7)	0.208 (9)	0.275 (8)	0.430 (9)	1.016 (9)	0.721 (10)	0.225 (10)	0.561 (11)	0.301 (11)	9.333 (9)
D^3VAE	0.765 (9)	0.238 (11)	0.291 (10)	0.351 (7)	1.013 (8)	0.781 (13)	0.206 (8)	0.504 (10)	0.316 (13)	9.889 (10)
FreTS	0.669 (3)	0.135 (3)	0.225 (6)	0.328 (2)	0.895 (4)	0.470 (5)	0.170 (4)	0.425 (4)	0.162 (2)	3.667 (3)
TimesNet	0.682 (6)	**0.130 (1)**	0.215 (2)	0.341 (5)	0.905 (6)	0.601 (6)	0.172 (6)	0.433 (7)	0.221 (6)	5.000 (6)
PatchTST	0.671 (4)	0.139 (6)	0.224 (5)	0.333 (4)	0.891 (3)	0.464 (4)	0.165 (2)	0.430 (6)	0.165 (3)	4.111 (4)
iTransformer	0.675 (5)	0.138 (5)	0.217 (3)	0.344 (6)	0.899 (5)	**0.437 (1)**	0.168 (3)	0.428 (5)	0.170 (5)	4.222 (5)
FedFormer	0.752 (8)	0.205 (8)	0.272 (7)	0.389 (8)	1.012 (7)	0.609 (7)	0.205 (7)	0.417 (3)	0.267 (8)	7.000 (7)
Autoformer	0.836 (11)	0.226 (10)	0.302 (11)	0.513 (13)	1.083 (13)	0.615 (8)	0.212 (9)	0.498 (9)	0.302 (12)	10.667 (11)
Pyraformer	0.972 (12)	0.273 (14)	0.305 (12)	0.494 (12)	1.061 (12)	0.745 (11)	0.257 (13)	0.641 (14)	0.322 (14)	12.667 (13)
Informer	0.980 (13)	0.242 (13)	0.315 (14)	0.541 (14)	1.168 (14)	0.779 (12)	0.250 (11)	0.625 (13)	0.298 (10)	13.000 (14)

Table 1: Testing MSE in the multivariate setting. Number in brackets is the rank. The best is in **bold**. CSDI runs out of memory on *Traffic*, *Electricity*.

errors when time series decomposition is used. Our model achieved the best results across six datasets, demonstrating its advanced capabilities. Details of the model and additional result visualizations are provided in the ***Appendix***.

Additionally, we selected a portion of the prediction results for demonstration. As shown in Figure 4, our method aligns more closely with the actual temporal values on the *ETTm1* dataset compared to other methods. This chart illustrates the differences between non-autoregressive diffusion models, represented by TimeDiff, and autoregressive diffusion models, exemplified by CSDI. CSDI suffers from error accumulation, leading to subpar predictions, especially in capturing significant peaks. TimeDiff addresses the issue of error accumulation but fails to adequately model the internal correlations within the sequence. Our TCDM can predict values across different timestamps very accurately. This further demonstrates the effectiveness of our method in capturing seasonal and trend patterns.

4.3 Designing Criteria to Validate the Temporal Correlation of Predicted Series

For the predicted series Y_P and the actual series Y_T, we aim to quantitatively measure the difference in their temporal correlations. Initially, we compute the partial autocorrelation coefficients for each feature channel i within the range $\{1, \ldots, D\}$. Employing partial autocorrelation coefficients is advantageous as they mitigate the confounding influences of intermediate variables, thus more precisely delineating the direct relationships between two variables. Using the mean squared error helps to limit the disproportionate impact of any single channel. Given a specific lag value k, the partial autocorrelation coefficients for both series are calculated and denoted by $\phi_P^{i,k}$ and $\phi_T^{i,k}$, respectively. After setting a maximum lag value E, we can generate the arrays $[\phi_P^{i,1}, \phi_P^{i,2}, \ldots, \phi_P^{i,E}]$ and $[\phi_T^{i,1}, \phi_T^{i,2}, \ldots, \phi_T^{i,E}]$. The index, calculated as

$$\frac{1}{D}\frac{1}{E}\sum_{i=1}^{D}\sum_{j=1}^{E}(\phi_P^{i,j} - \phi_T^{i,j})^2.$$

We propose and refer to it as the Normalized Partial Autocorrelation Difference Index (N-PADI). A lower N-PADI value indicates that the predicted series effectively captures the temporal correlations of the actual series. From the data

	TCDM	TimeDiff	FreTS	TimeGrad	PatchTST
ETTh1	**0.559**	60.057	0.581	80.012	4.527
ETTm1	**4.972**	56.954	5.041	60.097	14.664

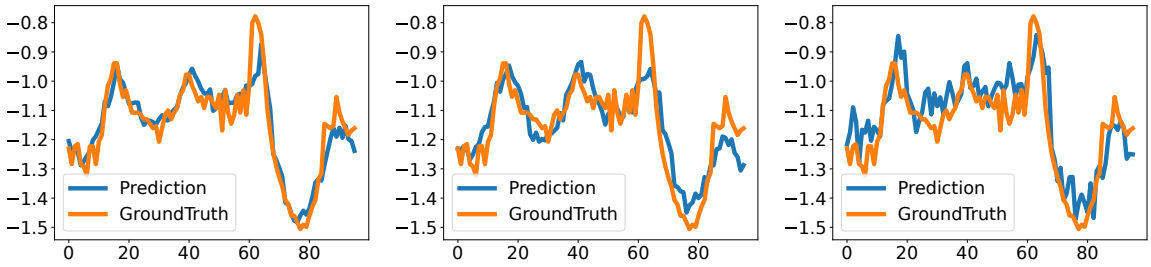
Table 2: N-PADI for various models on the ETTh1 and ETTm1 datasets. **Bold** font denotes best results.

presented in Table 2, it is evident that the performance of N-PADI is heavily influenced by its use of the L2 distance, which significantly affects the accuracy of temporal capture. TimeGrad faces challenges in grasping temporal correlations within time series. Another diffusion-based model, TimeDiff, achieves better MSE performance compared to FreTS. However, the noise-adding mechanism inherent to diffusion models, combined with the extended forward step t, limits its ability to capture temporal correlations, leading to higher N-PADI values than FreTS. PatchTST, a transformer-based model, also fails to learn additional temporal correlations compared to FreTS, underscoring the limitation of transformers in effectively retaining temporal correlations even with positional embeddings. Conversely, TCDM, our non-autoregressive diffusion model, incorporates the Maintaining Temporal Correlation Module and the Redesigned Initial Module, which significantly aid in preserving temporal correlations, thereby improving predictive performance.

4.4 Hyperparameter Analysis

Truncated Timestamp

The experiment was conducted on the ETTh1 dataset. As shown in Figure 5, when $S = 2$, the generator struggles to learn effectively due to the limited number of diffusion steps, resulting in poor performance. However, as S increases from 2 to 50, the additional diffusion steps enable the generator to acquire more precise knowledge, significantly improving the quality of the generated initial states. This suggests that at



(a) TCDM (b) TimeDiff (c) CSDI

Figure 4: Visualization Cases: Predictive Results of TCDM, TimeDiff, and CSDI on the ETTm1 Dataset.

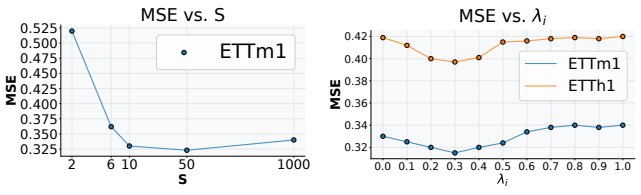
$S = 50$, the model is capable of producing fairly realistic R^S. Comparing $S = 50$ to $S = 1000$, $S = 50$ places the model in a better initial state. Furthermore, when the diffusion step exceeds 100, it becomes difficult to learn effective knowledge, including temporal correlations, highlighting the effectiveness of the Redesigned Initial Module.

Settings of Maintaining Temporal Correlation Module λ^i From Figure 5, it is observed that on both datasets, as λ^i increases, the MSE initially decreases and then increases, with the minimum point reached at 0.3. This pattern may be attributed to both ETTh1 and ETTm1 being excerpts from the ETT dataset. As λ^i ranges from 0 to 0.3, the Temporal Coherent Noise Module gradually aids the model in better capturing the relationships within the time series. Although the relationships are artificially defined, they can guide the model to autonomously learn the true event correlations. However, when λ^i increases beyond 0.3, overfitting occurs. This is due to the Temporal Coherent Noise Module excessively modeling the artificially designed temporal correlations, which impedes the model's ability to learn authentic correlations, resulting in a decline in predictive performance.

4.5 Ablation Study

We maintain consistency with the hyperparameters of the main experiment and conduct ablation studies to validate individual model effectiveness. We perform these module ablation experiments in a multivariate setting on the *ETTh1* and *Exchange* datasets.

As shown in Table 3, ablation studies on the TCDM model demonstrate that the full TCDM module consistently outperforms configurations with any single module removed across



(a) S varies (b) λ_i varies

Figure 5: Analysis of Hyperparameters: Noise Parameter λ_i and Stop Step S.

Module	*ETTh1*	*Exchange*
Full TCDM	**0.396**	**0.160**
w/o MTC Module	0.403	0.190
w/o Redesigned Initial Module	0.406	0.171
w/o Decomposition of Time Series	0.402	0.185

Table 3: MSE for ablation study. **Bold** font denotes best results. MTC means Maintaining Temporal Correlation.

datasets, indicating a synergistic enhancement of overall performance. Removing the Maintaining Temporal Correlation Module led to significant declines in performance on the Exchange, underscoring its crucial role in predictive accuracy. The Redesigned Initial Module proved especially effective on ETTh1, enhancing prediction precision for dynamic patterns. Although removing the Time Series Decomposition had a minor overall impact, it increased the error on the Exchange dataset (from 0.160 to 0.185), highlighting its importance for temporally coherent data. These findings emphasize the critical contributions of each module to achieving high accuracy in multivariate time series forecasting.

5 Conclusion

In this study, we propose a novel decomposition-based prediction framework to enhance the Temporal Correlation-Empowered Diffusion Model, collectively referred to as TCDM. The framework leverages time series decomposition to integrate a base model with a diffusion model, enabling it to effectively capture both long-term and short-term temporal dependencies. The diffusion model is specifically designed to preserve temporal correlations through two key components: the Maintaining Temporal Correlation Module and the Redesigned Initial Module. These components are developed by addressing three critical aspects of diffusion models: noise addition, initial state, and the sampling process. Experimental results demonstrate that each component contributes to improved model performance, and TCDM significantly surpasses existing models, establishing it as a powerful and reliable solution for accurate time series prediction.

References

[Aczel and Josephy, 1991] Amir D Aczel and Norman H Josephy. The chaotic behavior of foreign exchange rates. *The American Economist*, 35(2):16–24, 1991.

[Anand and Achim, 2022] Namrata Anand and Tudor Achim. Protein structure and sequence generation with equivariant denoising diffusion probabilistic models, 2022.

[Chen *et al.*, 2021] Minghao Chen, Houwen Peng, Jianlong Fu, and Haibin Ling. Autoformer: Searching transformers for visual recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12270–12280, 2021.

[Dhariwal and Nichol, 2021] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021.

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

[Jiang *et al.*, 2023] Junji Jiang, Likang Wu, Hongke Zhao, Hengshu Zhu, and Wei Zhang. Forecasting movements of stock time series based on hidden state guided deep learning approach. *Information Processing & Management*, 60(3):103328, 2023.

[Kawar *et al.*, 2021] Bahjat Kawar, Gregory Vaksman, and Michael Elad. Stochastic image denoising by sampling from the posterior distribution, 2021.

[Li *et al.*, 2023] Yuxin Li, Wenchao Chen, Xinyue Hu, Bo Chen, Mingyuan Zhou, et al. Transformer-modulated diffusion models for probabilistic multivariate time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2023.

[Li *et al.*, 2024] Kunxi Li, Tianyu Zhan, Kairui Fu, Shengyu Zhang, Kun Kuang, Jiwei Li, Zhou Zhao, Fan Wu, and Fei Wu. Mergenet: Knowledge migration across heterogeneous models, tasks, and modalities, 2024.

[Liu *et al.*, 1997] Yanhui Liu, Pierre Cizeau, Martin Meyer, C-K Peng, and H Eugene Stanley. Correlations in economic time series. *Physica A: Statistical Mechanics and its Applications*, 245(3-4):437–440, 1997.

[Luo, 2022] Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022.

[Ma *et al.*, 2024] Xiang Ma, Xuemei Li, Lexin Fang, Tianlong Zhao, and Caiming Zhang. U-mixer: An unet-mixer architecture with stationarity correction for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 14255–14262, 2024.

[Meijer and Chen, 2024] Caspar Meijer and Lydia Y Chen. The rise of diffusion models in time-series forecasting. *arXiv preprint arXiv:2401.03006*, 2024.

[Rasul *et al.*, 2021] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, pages 8857–8868. PMLR, 2021.

[Selesnick *et al.*, 2014] Ivan W Selesnick, Harry L Graber, Douglas S Pfeil, and Randall L Barbour. Simultaneous low-pass filtering and total variation denoising. *IEEE Transactions on Signal Processing*, 62(5):1109–1124, 2014.

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

[Shen *et al.*, 2021] Bo-Wen Shen, Roger A Pielke Sr, Xubin Zeng, Jong-Jin Baik, Sara Faghih-Naini, Jialin Cui, and Robert Atlas. Is weather chaotic?: Coexistence of chaos and order within a generalized lorenz model. *Bulletin of the American Meteorological Society*, 102(1):E148–E158, 2021.

[Tashiro *et al.*, 2021] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021.

[Walters and Heston, 1982] Roy A Walters and Cynthia Heston. Removing tidal-period variations from time-series data using low-pass digital filters. *Journal of physical oceanography*, 12(1):112–115, 1982.

[Wang *et al.*, 2024] Qinghe Wang, Baolu Li, Xiaomin Li, Bing Cao, Liqian Ma, Huchuan Lu, and Xu Jia. Characterfactory: Sampling consistent characters with gans for diffusion models. *arXiv preprint arXiv:2404.15677*, 2024.

[Wu *et al.*, 2024] Likang Wu, Zhaopeng Qiu, Zhi Zheng, Hengshu Zhu, and Enhong Chen. Exploring large language model for graph data understanding in online job recommendations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 9178–9186, 2024.

[Xiao *et al.*, 2021] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. *arXiv preprint arXiv:2112.07804*, 2021.

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

[Zheng *et al.*, 2022] Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Truncated diffusion probabilistic models and diffusion-based adversarial auto-encoders. *arXiv preprint arXiv:2202.09671*, 2022.