TimeDiT: General-purpose Diffusion Transformers for Time Series Foundation Model

A Preprint

Defu Cao,[*] Wen Ye,[*] Yizhou Zhang, Yan Liu
Department of Computer Science
University of Southern California
Los Angeles, CA 90007
{defucao, yewen, zhangyiz, yanliu.cs}@usc.edu

September 5, 2024

Abstract

With recent advances in building foundation models for texts and video data, there is a surge of interest in foundation models for time series. A family of models have been developed, utilizing a temporal auto-regressive generative Transformer architecture, whose effectiveness has been proven in Large Language Models. While the empirical results are promising, almost all existing time series foundation models have only been tested on well-curated "benchmark" datasets very similar to texts. However, real-world time series exhibit unique challenges, such as variable channel sizes across domains, missing values, and varying signal sampling intervals due to the multi-resolution nature of real-world data. Additionally, the uni-directional nature of temporally auto-regressive decoding limits the incorporation of domain knowledge, such as physical laws expressed as partial differential equations (PDEs). To address these challenges, we introduce the Time Diffusion Transformer (TimeDiT), a general foundation model for time series that employs a denoising diffusion paradigm instead of temporal auto-regressive generation. TimeDiT leverages the Transformer architecture to capture temporal dependencies and employs diffusion processes to generate high-quality candidate samples without imposing stringent assumptions on the target distribution via novel masking schemes and a channel alignment strategy. Furthermore, we propose a finetuning-free model editing strategy that allows the seamless integration of external knowledge during the sampling process without updating any model parameters. Extensive experiments conducted on a varity of tasks such as forecasting, imputation, and anomaly detection, demonstrate the effectiveness of TimeDiT.

1 Introduction

Time series analysis is pivotal in a diverse set of AI applications, such as natural science [Cuomo et al., 2022, Wang et al., 2020], social science [Zhang et al., 2022, Sharma et al., 2021],sustainability [Krenn and Buffoni, 2023],health [Kaushik et al., 2020, Kamra et al., 2021], etc. These applications root in diverse domains [Li et al., 2018, Bi et al., 2023, Cao et al., 2023c, Zhang et al., 2021, Ye and Gao, 2022], leading to time series with various distributions [Wang et al., 2023] and a divers set of analysis tasks, such as forecasting [Nie et al., 2022, Jia et al., 2024], imputation [Tashiro et al., 2021], anomaly detection [Tuli et al., 2022], etc. Even though considerable progress has been made in developing specialized models optimized for specific scenarios and individual tasks, an open question remains: *Can a single time series foundation model excel across domains?* Recent initiatives have explored the possibility of universal time series models on zero-shot setting [Ansari et al., 2024, Liu et al., 2024, Gruver et al., 2024, Cao et al., 2023b], drawing inspiration from large pre-trained language models in natural language processing(NLP) and computer vision(CV), such as GPT[Radford et al., 2018], CLIP [Radford et al., 2021], which are known for their robust transfer learning capabilities. However, due to the fundamentally different semantics between text/images and time series data, the unique challenges of achieving a truly flexible and general-purpose time series model remain an open problem.

[*] Equal Contribution with Alphabetical Order.

Recently, the emergence of LLMs like GPT-4 [OpenAI, 2023] and LLaMA [Touvron et al., 2023] suggests the potential for building time series foundation models enabling a general solution to handle multiple time series distributions. Previous attempts typically build upon the transformer backbone, which has achieved state-of-the-art performance on various time series tasks, particularly in modeling long-term dependencies. However, the tokenization of time series data for transformers is especially sensitive to variations in data sources and sampling rates. Previous tokenization approaches with different schemes including token patching [Das et al., 2023, Woo et al., 2024a]; discretization tokens [Talukder et al., 2024] and tokens based on time series features [Yue et al., 2022, Ansari et al., 2024, Rasul et al., 2023] have either fragmented the global information or have been constructed in a manner that inherently loses important information. Most, if not all, of them employ a channel independence strategy [Nie et al., 2023] or focus solely on univariate time series. Channel independence strategy, though beneficial in certain contexts, often overlooks the complex inter-temporal and cross-feature dependencies in practical applications and thus presents an opportunity for optimization [Zeng et al., 2023].

Moreover, compared with texts and images, time series exhibit unique characteristics such as *missing values* [Kollovieh et al., 2023], *irregular sampling* [Kidger et al., 2020, Cao et al., 2023a, Zhang et al., 2022], *multi-resolution* [Niu et al., 2023, 2022], etc. To address these challenges, a foundation model for time series must be capable of demonstrating flexibility across different scales to handle diverse inputs with varying distributions. However, these unique natures and challenges are not covered by the popular well-curated benchmark datasets[Li et al., 2018, Zhou et al., 2021, Alexandrov et al., 2020]. As a result, most existing works, which are developed and evaluated primarily on these datasets, may not fully address the complexities encountered in real-world time series applications. In addition, time series processes are often governed by underlying physical principles [Li et al., 2021, Meng et al., 2022, Xiao et al., 2023] and can be guided by domain-specific textual information [Jin et al., 2023, Sun et al., 2023]. However, integrating these diverse sources of information into a unified model poses further challenges, as the model must effectively leverage the relevant physics context while adapting to the unique characteristics and distributions of each domain. Addressing each of these issues requires innovative approaches in data preprocessing, model architecture, and training strategies to create models that can seamlessly handle the diverse and complex nature of time series data.

Current approaches to time series modeling lack a unified framework for handling the aforementioned diverse and imperfect data inputs, frequently prioritizing performance on well-curated benchmarks over addressing real-world challenges. Diffusion models, such as DDPM [Ho et al., 2020], offer a promising solution by framing data generation as a series of conditional transformations, effectively recasting density estimation as sequential reconstruction. Unlike autoregressive methods that generate future tokens sequentially, diffusion models can directly produce high-quality samples through a reverse denoising process. This process can be analogized to solving partial differential equations (PDEs), allowing for the natural incorporation of physics-based knowledge. This capability, combined with transformers' strength in capturing temporal dependencies, presents an opportunity to develop a more versatile and robust time series foundation model. Such a hybrid approach could effectively address the complexities of real-world data while maintaining the flexibility to adapt to various forecasting tasks and data conditions.

In this work, we introduce TimeDiT—a diffusion transformer-based foundation model equipped with a standardized training pipeline for different shapes of input time series and tailored for diverse distributions and downstream tasks. TimeDiT leverages the Transformer architecture's inherent ability to capture temporal dependencies through its attention mechanisms, while also benefiting from the scalability that allows for increased model capacity crucial for complex time series tasks. By adopting a diffusion model approach, TimeDiT treats time series holistically, avoiding the error accumulation issues common in autoregressive solutions. The model incorporates a novel comprehensive mask mechanism that enables a single, unified foundation model to handle multiple tasks without additional modules or parameters. This design naturally addresses real-world challenges such as multi-resolution data and missing values. During the sampling stage, TimeDiT introduces an innovative strategy to incorporate physics knowledge as an energy-based prior, supported by theoretical guarantees. This approach guides the reverse diffusion process using physics-based constraints, including partial differential equations, resulting in generated samples that adhere to known physical laws and domain-specific requirements, thereby enhancing sample quality and model applicability across various scientific and engineering contexts.

TimeDiT's performance is rigorously evaluated through an extensive experimental setup encompassing over 20 diverse datasets from domains including traffic, weather, finance, etc. The model is benchmarked against more than 25 open-source baselines, ranging from linear-based models to diffusion-based models, transformer-based models, and other forecasting foundation models. These comprehensive experiments cover multiple challenging time series tasks, including in-domain and zero-shot probabilistic forecasting, imputation, anomaly detection, and synthetic data generation. TimeDiT demonstrated state-of-the-art or highly competitive results across these tasks, showcasing its effectiveness and efficiency as a foundation model for various time series applications. Notably, TimeDiT achieved new state-of-the-art CRPS$_{sum}$ scores on the Electricity and Traffic datasets for probabilistic forecasting. In addition, the results on zero-shot experiments show that our model can be used as a foundation model even without fine-tuning, although fine-tuning may be necessary in some cases. Furthermore, TimeDiT's scalability and adaptability are evident in its ability to incorporate external knowledge, such as physical constraints, during the sampling stage. This feature allows for the generation of samples that better conform to known physical laws and domain-specific requirements. This combination of state-of-the-art performance, adaptability across diverse tasks, scalability, and the ability to incorporate domain-specific knowledge positions TimeDiT as a powerful and versatile foundation model, capable of addressing a wide spectrum of time series challenges and opening new avenues for advanced time series analysis across various fields.

In summary, our contributions can be summarized as three unfolds:

- We introduce TimeDiT, a novel diffusion transformer-based foundation model for time series analysis. By combining the strengths of diffusion models and transformers, our approach offers a flexible architecture adaptable to various downstream tasks. The model incorporates a comprehensive mask mechanism for reconstruction pretraining and task-specific fine-tuning, ensuring a standardized training pipeline capable of handling diverse input shapes and distributions.

- Unlike autoregressive approaches, TimeDiT addresses real-world challenges in time series data by directly processing multivariate inputs and employing a denoising process to generate cohesive target time series. This method effectively handles issues such as missing values and multi-resolution data. In addition, TimeDiT can generate time series that adhere to known physical laws and domain-specific requirements, enhancing its applicability in scientific and engineering contexts.

- Evaluated on over multiple datasets across different domains and tasks, TimeDiT achieves state-of-the-art or competitive results. It excels in probabilistic forecasting, imputation, anomaly detection, and data generation, showcasing its versatility as a foundation model in both in-domain and zero-shot settings.

2 Related Work

General Purpose Time Series Model In the past decades, researchers have excelled in designing sophisticated models for specific time series analysis tasks [Zhang et al., 2024b, Fan et al., 2024, Cao et al., 2020, 2022]. However, in recent years, the emergence of large language models has inspired the development of general-purpose time series models [Zerveas et al., 2021, Zhang et al., 2024a, Garza and Mergenthaler-Canseco, 2023] and the field of time series has seen tremendous exploration efforts towards foundation models. [Gruver et al., 2024] simply encoded time series as strings while [Jin et al., 2023] converted time series into language representations by alignment. [Cao et al., 2023b] and [Pan et al., 2024] further incorporated decomposition technique and prompt design and generalizes to unseen data and multimodal scenarios. [Rasul et al., 2023] worked towards foundation model from a probabilistic perspective but only considered univariate time series only which rarely appears in real-life. Additionally, many studies started to follow a two-stage training paradigm of pretraining and finetuning [Chang et al., 2023, Dong et al., 2024, Nie et al., 2022]. However, these works mainly focused on the forecasting task only [Woo et al., 2024a, Das et al., 2023]. [Zhou et al., 2023] first adapted GPT2 as a general-purpose time series analysis model and extended it to various time series tasks. [Talukder et al., 2024] leveraged VQVAE as a tokenizer for transformer to handle time series tasks and [Ansari et al., 2024] employed a scaling and quantization technique to embed time series. For more detailed literatures of the general-purpose time series model, please refer to recent surveys and position paper [Liang et al., 2024, Jin et al., 2024, Jiang et al., 2024]

Diffusion models for Time Series Despite the growing interest of diffusion models in various scenarios [Peebles and Xie, 2022, Li et al., 2022a, Lu et al., 2024, Sui et al., 2024a,b], the use of diffusions in time series analysis is less explored compared to pre-trained language models and transformers. Most existing studies also focused solely on forecasting and the choice of backbone model also varies among VAE[Li et al., 2022b], RNN[Rasul et al., 2021], and transformer. CSDI [Tashiro et al., 2021] utilized diffusion model for time series imputation. [Yuan and Qiao, 2024] incorporated decomposition into diffusion model to improve interoperability. Although [Kollovieh et al., 2023] build a diffusion pipeline for multiple tasks with refinement, they still train different models for each task. To the best of our knowledge, there has been no exploration of leveraging unified diffusion models for a comprehensive set of time series tasks yet. Please refer to [Yang et al., 2024] for a comprehensive literature review on diffusion models for time series analysis.

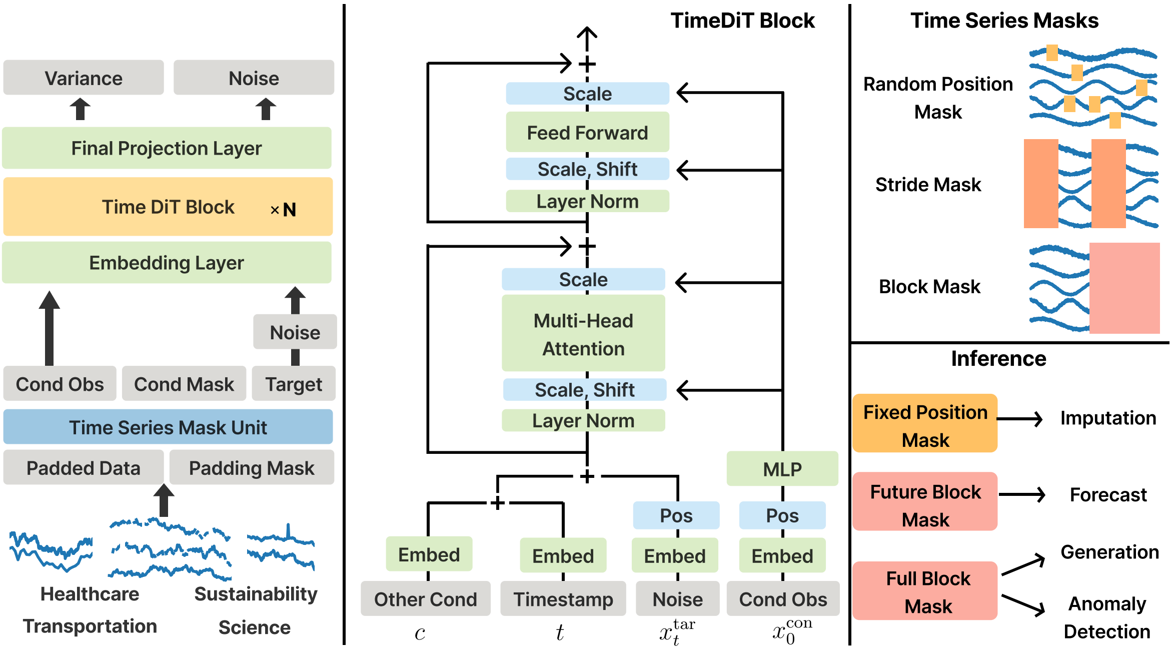


Figure 1: TimeDiT Architecture. Left: TimeDiT framework with diverse multivariate time series from different domains with multi-resolution, missing values; Middle: Structure of TimeDiT block; Right top: Illustration of masks generated by Time Series Mask Unit, reconstruction mask is neglected as it's an all-zero mask; Right bottom: Masks for downstream tasks that TimeDiT handles during inference.

3 Preliminaries

3.1 Diffusion Model

In recent years, diffusion models have emerged as a promising approach in generative modeling. A diffusion process is a Markov chain that incrementally adds Gaussian noise to data over a sequence of steps, effectively destroying the data structure in forward process and destroying the data structure in backward structure.

The forward process adds noise to the data $\mathbf{x}_0$ over a series of timesteps t according to a variance schedule β_t, resulting in a set of noisy intermediate variables $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$. Each subsequent $\mathbf{x}_t$ is derived from the previous step by applying Gaussian noise:

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

The reverse process aims to denoise the noisy variables step by step, sampling each $\mathbf{x}_{t-1}$ from the learned distribution $p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$. This distribution, modeled by a neural network parameterized by θ, approximates the Gaussian distribution:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_\theta(\mathbf{x}_t, t)) \tag{2}$$

By iterating this reverse process from $t = T$ down to $t = 0$, the model gradually reconstructs the original data from noise. The reverse process learns to predict the mean and covariance of each intermediate distribution, effectively approximating the original data distribution.

4 Methodology

In this section, we present our main contributions: the proposed foundation model, TimeDiT , a diffusion model with transformer backbone designed for multiple time series tasks, along with uniform masking strategies and incorporation of physics knowledge and textual information as an extension. We first outline the uniformed problem setting for multiple down-stream tasks and offer an in-depth examination of the model architecture. Subsequently, we delve into the training pipeline with mask strategies, which help to build the training scheme in self-supervised learning for time series. Next, we present how to incorporate external information to improve the model's performance during both the training and inference stages. By doing so, TimeDiT can generate samples that better conform to real-world requirements and enhance its performance on various downstream tasks. These extensions showcase the flexibility and adaptability of our proposed model, making it a powerful tool for a wide range of time series applications.

4.1 Problem Definition

We denote a multivariate time series as $\mathbf{X} = \{x_{i,j}\} \in \mathbb{R}^{K\times L}$, where K is the number of features and L is the length of the time series. Each individual entry $x_{i,j}$ represents the j-th feature at time step l, for $i \in \{1, \ldots, K\}$ and $j \in \{1, \ldots, L\}$. We define an observation mask $\mathbf{M_{obs}} = \{m_{i,j}\} \in \{0, 1\}^{K\times L}$, where $m_{i,j} = 0$ if $x_{i,j}$ is missing, and $m_{i,j} = 1$ if $x_{i,j}$ is observed. Let $\mathbf{x}_0^{\text{obs}} \in X^{\text{obs}}$ denote the observed subsequence; $\mathbf{x}_0^{\text{tar}}$ denote the target subsequence of $\mathbf{x}_0^{obs}$ which could be forecast target or imputation target or the whole sequence depending on the task. Let $\mathbf{x}_0^{\text{con}}$ denote the unmasked partial observations in $\mathbf{x}_0^{obs}$ which acts like conditions for the masked area $\mathbf{x}_0^{tar}$. Let us use all subscripts of x to denote diffusion timestamp, and a subscript of 0 means no noise has been applied to the original data. Formally, the goal of our task is to approximate the true conditional data distribution given the conditional information $q_{\mathbf{X}}\left(\mathbf{x}_0^{\text{ta}} \mid \mathbf{x}_0^{\text{con}}\right)$ with a model distribution $p_\theta(\mathbf{x}_0^{\text{tar}} \mid \mathbf{x}_0^{\text{con}})$, which can be calculated by a diffusion model with conditional information:

$$
\begin{aligned}
&p_\theta\left(\mathbf{x}_{0:T}^{\text{tar}} \mid \mathbf{x}_0^{\text{con}}\right) := p\left(\mathbf{x}_T^{\text{tar}}\right) \prod_{t=1}^{T} p_\theta\left(\mathbf{x}_{t-1}^{\text{tar}} \mid \mathbf{x}_t^{\text{tar}}, \mathbf{x}_0^{\text{con}}\right), \mathbf{x}_T^{\text{tar}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \text{where} \\
&p_\theta\left(\mathbf{x}_{t-1}^{\text{tar}} \mid \mathbf{x}_t^{\text{tar}}, \mathbf{x}_0^{\text{con}}\right) := \mathcal{N}\left(\mathbf{x}_{t-1}^{\text{tar}}; \boldsymbol{\mu}_\theta\left(\mathbf{x}_t^{\text{tar}}, t \mid \mathbf{x}_0^{\text{con}}\right), \sigma_\theta\left(\mathbf{x}_t^{\text{tar}}, t \mid \mathbf{x}_0^{\text{con}}\right)\mathbf{I}\right).
\end{aligned} \tag{3}
$$

The mask mechanism $\mathbf{M}$ plays a critical role in identifying the positions of $\mathbf{x}_0^{\text{con}}$ and $\mathbf{x}_0^{\text{tar}}$. By leveraging these positional differences, our model adeptly adapts to various downstream tasks, including forecasting, imputation, anomaly detection, etc, within a unified framework.

4.2 Time Series Diffusion Transformer

Figure 1 shows the overall framework of TimeDiT. We first establish the $\mathbf{M_{obs}}$ and $\mathbf{x}_0^{\text{obs}}$ based on the given input from different distributions with multivariate sequences, missing value and multi-resolution by injecting placeholders to standardize the input shape across different time series, facilitating more efficient and consistent processing. Then, the unified time series mask unit adapts to diverse time series scenarios and builds the $\mathbf{x}_0^{\text{con}}$, $\mathbf{M}$ and $\mathbf{x}_0^{\text{tar}}$, with shape $\mathbb{R}^{B\times L\times K}$, to help TimeDiT learn robust representations in a self-supervised manner by reconstructing the original sequence through denoising $\mathbf{x}_T^{\text{tar}}$. After that, the embedding layer directly treats $\mathbf{x}_0^{\text{con}}$ and $\mathbf{x}_0^{\text{tar}}$ as without any patching, as the diffusion process is designed to handle multivariate input and operate in a continuous token space. By preserving the integrity of the input time series, TimeDiT ensures that the model can effectively capture and utilize the rich information contained within the data. The TimeDiT block's attention mechanism is designed to autonomously learn cross-channel and temporal correlations through end-to-end training.

Standardize Pipeline We introduce placeholders within the input sequences to standardize the input shape across different time series, accounting for varying channel numbers K and sequence length L. Specifically, we define the maximum channel number K_{max} such that any input with channel $k < K_{max}$ is padded to have K^{max} channels while any input with more than K_{max} channels are segmented into $\lceil \frac{k}{K_{max}} \rceil$ blocks of inputs where each block has K_{max} channels and undergoes independent processing. This segmentation allows our model to manage high-dimensional data efficiently, reducing computational overhead and maintaining relative positional integrity of the data and consistency across inputs. Additionally, for any input with sequence length less than the designated maximum length L_{max}, we pad the sequence in the front to achieve the desired length. This standardization is essential for establishing a uniform input structure that enhances processing efficiency and consistency.

Time Series Mask Unit We propose a unified time series mask mechanism that includes a variety of masks that seamlessly integrates with the model during self-supervised task agnostic pre-training and task specific fine-tuning to cater to diverse time series scenarios. The time series mask unit generate four types of masks: reconstruction mask, stride mask, block mask, and random mask. Firstly, the task-agnostic pre-training aims to improve the overall time series representation by encouraging the model to learn robust and generalizable features from the input data. Secondly, the task-specific training is designed specifically for the most common downstream tasks, including forecasting and imputation, enabling the model to adapt to the unique requirements of each task.

As shown in Figure 1 right top, given $\mathbf{x} \in \mathbb{R}^{K\times L}$, the random mask M^{R} can be generated by:

$$
\mathbf{M}^{\mathrm{R}}(x, r) = \begin{cases} 1 & z_{i,j} > r, z \in \mathbb{R}^{K\times L}, z \sim Uniform(0, 1) \\ 0, & otherwise, \end{cases} \tag{4}
$$

where r is the mask ratio. In addition, for task-specific training and inference, we allow the user to supply customized imputation masks that could simulate the naturally missing data and multi-resolution cases.

Block mask $\mathbf{M}^{\mathrm{B}}$ can be generated via:

$$\mathbf{M}^{\mathrm{B}}(x, l) = \begin{cases} 1 & j < L - l, \\ 0, & otherwise, \end{cases} \tag{5}$$

where l is the predicted length. We can randomly select l during pretraining and use the designated prediction length during the finetuning and inference stage for specific experiment settings.

Stride mask $\mathbf{M}^{\mathrm{S}}$, a variant of $\mathbf{M}^{\mathrm{B}}$, is placed intermittently within the series and is defined as follows:

$$\mathbf{M}^{\mathrm{S}}(x, n_{\text{blocks}}) = \begin{cases} 1 & \lfloor \frac{j}{b} \rfloor \bmod 2 = 0 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where n_{block} is the number of blocks into which the sequence is divided; $b = \left\lceil \frac{L}{n_{\text{blocks}}} \right\rceil$ is the length of each block; j is the index of the sequence. $\mathbf{M}^{\mathrm{S}}$is designed for task-agnostic pretraining further to enhance the model's representation ability by integrating information across non-contiguous parts of the series.

In addition, reconstruction mask $\mathbf{M}^{\mathrm{Rec}} = 0$ is used for tasks including synthetic data generation and anomaly detection, where we can directly generate synthetic data or obtain an anomaly score for each temporal position based on the difference between original and reconstructed series.

For the pretraining stage, we random select one conditional mask type from $\mathbf{M} = \{\mathbf{M}^{\mathrm{R}}, \mathbf{M}^{\mathrm{S}}, \mathbf{M}^{\mathrm{B}}, \mathbf{M}^{\mathrm{Rec}}\}$ for each instance. TimeDiT's goal is to reconstruct the $\mathbf{x}^{\text{tar}}$, defined as $\mathbf{x}_0 \times (J - \mathbf{M})$ where J is unit matrix. The target sequence is the masked portion of the original sequence. In the finetuning and inference stage, the choice of mask is tailored to align with the specific requirements of the user. This flexibility allows TimeDiT to apply the most appropriate masking strategy based on the context of the task and application.

Condition Injection Instead of following [Peebles and Xie, 2022] to integrate diffusion timestep and label information (texts in out case) through the layer normalization preceeding the attention block, we add the diffusion timestep and texts information to the target noise as these are universal information to the series.

Given that TimeDiT utilizes a transformer-based architecture, a straightforward and intuitive approach is to include conditional information directly as part of the input sequence by concatenation as done in latent diffusion [Rombach et al., 2022]. However, we empirically found out that controlling mean and variance through layer normalization is a stronger form of conditional information injection. We incorporate the partial observations x_0^con through adaptive layer normalization to control the scale and shift of the x_0^{tar}. This design choice is motivated by the face that the scale and shift of the partial observations are more relevant to the target observations. This integration can be expressed as

$$\text{AdaLN}(h, c) = c_{scale}\text{LayerNorm}(h) + c_{shift} \tag{7}$$

where h is the hidden state and c_{scale} and c_{shift} are the scale and shift parameters derived from the partial observations. We perform temporal attention in the self-attention block to capture the temporal dependency within the input.

Inference We perform pre-training and finetuning for forecasting and imputation. We exclude the pre-training process from anomaly detection and synthetic generation because the two tasks are very dataset specific and do not necessarily benefit from learning distributions beyond the target dataset. Let n represent the number of samples generated for each prediction, which we set to $n = 10$ ($n = 30$ for forecasting tasks) in our experimental setup at inference time. We use the median of these n predictions as the final prediction, providing the added benefit of obtaining a confidence interval for TimeDiT 's predictions. To prevent channel padding from affecting the generated samples, we mask out the invalid channels during sampling at each diffusion timestep so that TimeDiT does not falsely treat information in the non-valid channels as meaningful information. Padding is applied at the beginning of the temporal dimension to ensure that the most relevant information remains at the end, thereby mitigating the effect of padding.

4.3 Physics-Informed TimeDiT

Physics principles are fundamental in shaping the evolution of temporal signals observed in real-world phenomena, such as climate patterns and oceanographic data. Therefore, it is essential to integrate physical knowledge into foundational time series models. In this section, we propose a strategy to incorporate physics knowledge as an energy-based prior for

Algorithm 1 Physics Informed TimeDiT through Energy-based Sampling

```
$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t = T, \dots, 1$ do
    $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
    $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
end for
for $j = 0, 1, .., k-1$ do
    $\mathbf{x}_{j+1}^{tar} = \mathbf{x}_j^{tar} + \epsilon \nabla K(\mathbf{x}_j^{tar}; \mathbf{x}^{obs}) + \alpha \epsilon \nabla \log p(\mathbf{x}_j^{tar} | \mathbf{x}^{obs}) + \sqrt{2\epsilon}\sigma, \sigma \sim \mathcal{N}(0, 1)$
end for
return $\mathbf{x}_k^{tar}$
```

TimeDiT during inference, which iteratively refines the reverse diffusion process. By guiding the denoising process during inference with gradients derived from physical laws represented by partial differential equations (**PDEs**), the integration of this knowledge can significantly enhance the quality of the generated samples.

A generic form of a physical law represented as a PDE that describes the evolution of a continuous temporal signal $\mathbf{x}(\mathbf{u}, t)$ over a spatial coordinate $\mathbf{u}$ is given by:

$$\frac{\partial \mathbf{x}}{\partial t} = F(t, \mathbf{x}, \mathbf{u}, \frac{\partial \mathbf{x}}{\partial \mathbf{u}_i}, \frac{\partial^2 \mathbf{x}}{\partial \mathbf{u}_i \partial \mathbf{u}_j}, \dots) \tag{8}$$

Based on this PDE representation of physical knowledge, the consistency between the predicted time series $\mathbf{x}^{\text{tar}}$ and the physics knowledge can be quantified using the following squared residual function:

$$K(\mathbf{x}^{\text{tar}}; F) = -|| \frac{\partial \mathbf{x}^{\text{tar}}}{\partial t} - F(t, \mathbf{x}^{\text{tar}}, \mathbf{u}, \frac{\partial \mathbf{x}^{\text{tar}}}{\partial \mathbf{u}_i}, \frac{\partial^2 \mathbf{x}^{\text{tar}}}{\partial \mathbf{u}_i \partial \mathbf{u}_j}, \dots)||_2^2 \tag{9}$$

This function reaches its maximum when the predicted time series is perfectly consistent with the physical model, resulting in a residual of 0. Using this metric K, physics knowledge can be integrated into a probabilistic time series foundation model $p(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}})$ by solving the following optimization problem to obtain a refined model $q(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}})$:

$$q(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}}) = \arg\max_q \mathbb{E}_{\mathbf{x}^{\text{tar}} \sim q} K(\mathbf{x}^{\text{tar}}; F) - \alpha D_{KL}(q(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}})||p(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}})) \tag{10}$$

where the first term represents the aforementioned physics knowledge metric, and the second term controls the divergence between $q(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}})$ and $p(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}})$. The following theorem provides a closed-form solution to the above optimization problem:

Theorem 4.1. *The optimal $q(\mathbf{x}^{tar}|\mathbf{x}^{con})$ in Eq.10 is the Boltzmann distribution defined on the following energy function:*

$$E(\mathbf{x}^{tar}; \mathbf{x}^{con}) = K(\mathbf{x}^{tar}; F) + \alpha \log p(\mathbf{x}^{tar}|\mathbf{x}^{con}) \tag{11}$$

in other words, the optimal $q(\mathbf{x}^{tar}|\mathbf{x}^{con})$ is:

$$q(\mathbf{x}^{tar}|\mathbf{x}^{con}) = \frac{1}{Z} \exp(K(\mathbf{x}^{tar}; F) + \alpha \log p(\mathbf{x}^{tar}|\mathbf{x}^{con})), \tag{12}$$

where $Z = \int \exp(K(\mathbf{x}^{tar}; F) + \alpha \log p(\mathbf{x}^{tar}|\mathbf{x}^{con})) d\mathbf{x}^{tar}$ is the partition function.

The theorem illustrates that sampling from the Boltzmann distribution defined in Eq. 11, is analogous to incorporating physics knowledge into model edition. In the context of diffusion models, this distribution can be effectively sampled using Langevin dynamics [Stoltz et al., 2010]:

$$\begin{aligned} \mathbf{x}_{j+1}^{\text{tar}} &= \mathbf{x}_j^{\text{tar}} + \epsilon \nabla \log q(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}}) + \sqrt{2\epsilon}\sigma, \sigma \sim \mathcal{N}(0, 1) \\ &= \mathbf{x}_j^{\text{tar}} + \epsilon \nabla K(\mathbf{x}_j^{\text{tar}}; \mathbf{x}^{\text{con}}) + \alpha \epsilon \nabla \log p(\mathbf{x}_j^{\text{tar}}|\mathbf{x}^{\text{con}}) + \sqrt{2\epsilon}\sigma, \sigma \sim \mathcal{N}(0, 1) \end{aligned} \tag{13}$$

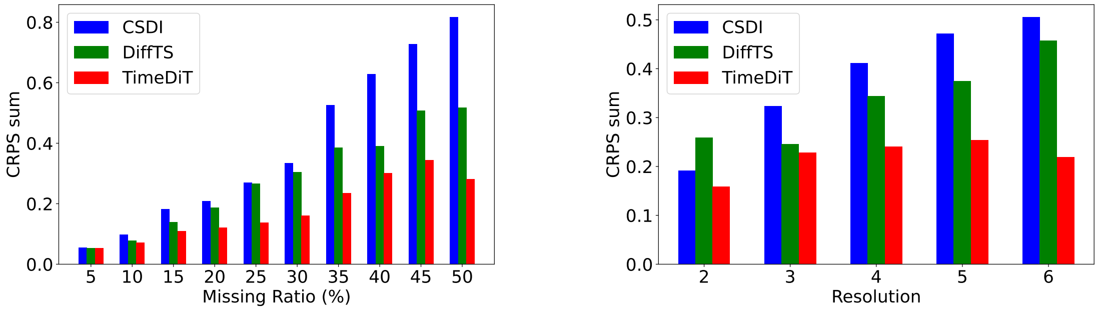
In diffusion model, precisely calculate the likelihood $\log p(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}})$ is intractable. To tackle this issue, following previous works [Kollovieh et al., 2023], we approximate likelihood with the objective to edit the pre-trained diffusion model:

$$\log p(\mathbf{x}^{\text{tar}}|\mathbf{x}^{\text{con}}) = -\mathbb{E}_{\epsilon, t}[||\epsilon_\theta(\mathbf{x}^{\text{tar}}, t; \mathbf{x}^{\text{con}}) - \epsilon||^2] \tag{14}$$

The approximation presented above constitutes the optimizable component of the evidence lower bound(ELBO). Algorithm 1 summarizes the comprehensive model editing process.

5 Experiments

To comprehensively assess our time series foundation model, we evaluate on a diverse set of tasks that reflect real-world challenges and applications. We begin by testing the model's performance in practical scenarios and its ability to integrate domain knowledge. This includes handling missing data and multi-resolution forecasting on customized datasets, which allows us to evaluate the model's robustness in situations that frequently occur in real-world applications, as well as physics-informed modeling crucial for scientific and engineering domains[Yuan and Qiao, 2024], which uses 6 practical partial differential equations (PDEs). We then assess the model's capabilities in well-established benchmarking tasks across various fields such as finance, healthcare, and industrial monitoring. These tasks include forecasting on Solar, Electricity, Traffic, Taxi, and Exchange datasets[Tashiro et al., 2021, Rasul et al., 2021] to evaluate temporal dependency modeling, imputation on ETTh, ETTm, Weather and Electricity datasets[Zhou et al., 2021] to assess the handling of missing data, anomaly detection on MSL, SMAP, SWaT, SMD, and PSM datasets[Xu et al., 2021, Zhao et al., 2020] to gauge sensitivity to unusual patterns, and synthetic data generation on Stock, Air Quality, and Energy datasets[Yoon et al., 2019, Desai et al., 2021] to test understanding of underlying distributions. By evaluating these diverse tasks, we can demonstrate that our model truly serves as a foundation for various time series applications, potentially reducing the need for task-specific models.



(a)

(b)

Figure 2: Visualization of miss value (a) and multi resolution (b) forecasting results on the Exchange dataset. Compared between our model TimeDiT and state-of-the-art diffusion-based methods. The x-axis number in (b) is the number of different sampling resolutions in the multivariate input.

5.1 Practical Scenarios: Missing Data and Multi-Resolution Forecasting

To simulate more realistic scenarios in time series tasks, we introduced two additional challenges: missing values and multi-resolution data. These conditions are common in real-world applications and test a model's robustness and adaptability. For the missing value scenario, we created datasets with various missing ratios, simulating incomplete data often encountered in practice. In the multi-resolution setting, we sampled each individual time series within the multivariate dataset at different resolutions, reflecting the diverse sampling frequencies often present in real-world data collection.

Figure 2 illustrates TimeDiT's performance in realistic scenarios, showcasing its effectiveness across different sampling frequencies on the Exchange dataset. In Figure 2 (a), we observe TimeDiT's superior performance in handling missing data. As the missing ratio increases from 5% to 50%, TimeDiT maintains the lowest CRPS$_{sum}$ across all scenarios, indicating its robustness to data gaps. The performance gap between TimeDiT and other models widens as the missing ratio increases, highlighting its effectiveness in more challenging conditions. Figure 2 (b) demonstrates TimeDiT's ability to manage multi-resolution data, where it maintains a clear performance advantage as the number of different sampling resolutions increases from 2 to 6. This demonstrates its ability to effectively integrate and forecast time series data sampled at varying frequencies. These findings underscore TimeDiT's potential as a practical and versatile tool for time series forecasting in diverse, challenging scenarios that more closely resemble real-world applications.

5.2 Domain Knowledge Integration: Physics-Informed TimeDiT

A key advantage of our approach is the ability to directly incorporate physics knowledge into the pretrained foundation model without additional fine-tuning. This is possible because the diffusion model reconstructs the entire process, allowing for seamless integration of PDE-based constraints during inference. By encoding the known physical laws governing the system into the sampling process, we can guide the model towards more physically consistent and

	MSE	**RMSE**	**MAE**	**CRPS**	**MSE**	**RMSE**	**MAE**	**CRPS**
	Advection				Navier-Stokes			
DDPM	0.011(0.000)	0.106(0.001)	0.084(0.001)	0.472(0.007)	0.309(0.004)	0.556(0.004)	0.332(0.005)	0.415(0.006)
DDIM	0.015(0.000)	0.122(0.002)	0.096(0.001)	0.559(0.009)	0.350(0.014)	0.591(0.011)	0.377(0.009)	0.470(0.013)
TSDiff	0.011(0.000)	0.106(0.022)	0.085(0.001)	0.472(0.011)	0.399(0.008)	0.556(0.007)	0.331(0.006)	0.414(0.007)
TimeDiT	**0.010(0.000)**	**0.103(0.002)**	**0.082(0.001)**	**0.464(0.008)**	**0.299(0.006)**	**0.546(0.006)**	**0.322(0.06)**	**0.403(0.007)**
	Burgers				Vorticity			
DDPM	0.016(0.001)	0.128(0.004)	0.101(0.003)	1.787(0.040)	1.917(0.020)	1.385(0.007)	0.851(0.009)	0.476(0.005)
DDIM	0.018(0.000)	0.136(0.001)	0.116(0.001)	1.858(0.015)	1.567(0.031)	1.252(0.012)	**0.754(0.012)**	**0.401(0.006)**
TSDiff	0.017(0.001)	0.129(0.005)	0.102(0.004)	1.800(0.055)	1.966(0.073)	1.402(0.026)	0.866(0.010)	0.485(0.005)
TimeDiT	**0.011(0.001)**	**0.104(0.005)**	**0.083(0.003)**	**1.395(0.053)**	**1.524(0.523)**	**1.234(0.021)**	0.772(0.009)	0.445(0.006)
	Diffusion Sorption				CFD			
DDPM	0.309(0.004)	0.556(0.004)	0.332(0.005)	0.415(0.006)	0.004(0.000)	0.065(0.001)	0.054(0.000)	0.082(0.000)
DDIM	0.349(0.013)	0.591(0.011)	0.377(0.009)	0.470(0.013)	0.039(0.002)	0.194(0.006)	0.188(0.006)	0.313(0.012)
TSDiff	0.309(0.008)	0.556(0.007)	0.331(0.006)	**0.414(0.007)**	-	-	-	-
TimeDiT	**0.284(0.005)**	**0.533(0.005)**	**0.327(0.005)**	0.423(0.007)	**0.004(0.000)**	**0.062(0.001)**	**0.051(0.001)**	**0.080(0.001)**

Table 1: Physics informed results: This table presents results of PDE forecasting for various sampling strategies, demonstrating the model's ability to incorporate physics constraints without updating the foundation model. Metrics shown include point estimates metrics, which is MSE, RMSE and MAE; probabilistic metrics, CRPS, for uncertainty quantification. Lower values indicate better performance and closer adherence to physical laws.

accurate predictions. In this section, we evaluate how effectively our pre-trained foundation model can integrate physics-informed knowledge into time series forecasting without the need for fine-tuning. We study four 1D partial differential equations (PDEs) from [Takamoto et al., 2022]: general Navier-Stokes Equations, Kolmogorov Flow (a specific case of Navier-Stokes Equations), Advection Equations, Burgers Equations, Diffusion Soeption and Computational Fluid Dynamics (CFD). These equations are used to generate synthetic data with random initial conditions, and we apply diffusion models to forecast time series based on data from a historical window.

Table 1 presents the results, including both mean error and error bars. The table clearly demonstrates that our proposed model editing solution, which incorporates physics knowledge, significantly outperforms previous sampling strategies such as DDPM [Ho et al., 2020], DDIM [Song et al.], and TS Diffusion's Self-Guidance [Kollovieh et al., 2023]. By leveraging domain-specific physical information, our approach achieves substantial performance improvements over these baselines, highlighting the effectiveness of integrating physics-informed priors into the diffusion model sampling process. This performance gain underscores the potential of combining pretrained foundation models with domain-specific knowledge. Our method offers a flexible framework for enhancing time series forecasting in scientific and engineering applications where the underlying physical laws are partially known. It demonstrates that by bridging the gap between data-driven approaches and physics-based modeling, we can achieve more accurate and physically consistent predictions without the computational overhead of retraining or fine-tuning the entire model. The ability to easily incorporate physics knowledge into a pretrained foundation model represents a significant advance in the field of scientific machine learning. It opens up new possibilities for rapid adaptation to specific physical systems and phenomena, potentially accelerating research and discovery in various scientific domains.

5.3 Forecasting on Full-shot Setting and Zero-shot Setting

In the forecasting task, we conduct two types of experiments. First, we compare our proposed TimeDiT with baselines in a full-shot setting, where models are trained and tested on separate datasets. This approach evaluates their performance on conventional time series forecasting tasks, ensuring that models can effectively learn and generalize from complete data. Second, we assess TimeDiT as a foundation model in a zero-shot setting, comparing it to previous transformer-based time series models. This setting is crucial as it tests the model's ability to generalize and adapt to entirely new datasets without prior exposure, highlighting its robustness and versatility. Together, these experiments provide a holistic view of TimeDiT's capabilities, addressing both specialized performance and broad applicability in time series forecasting.

Table 2 presents the full-shot forecasting results, comparing TimeDiT with state-of-the-art models in two categories: deterministic forecasting models, which are trained with the Student's t-distribution head to support probabilistic results, and inherently probabilistic time series forecasting models, including diffusion-based models, such as CSDI and non-diffusion-based, such as GP-copula. Our model achieves the lowest CRPS$_{sum}$ on four datasets and the

		Solar	Electricity	Traffic	Taxi	Exchange
Deterministic	DLinear	0.432(0.002)	0.033(0.000)	0.070(0.001)	0.177(0.000)	0.011(0.001)
	PatchTST	0.457(0.019)	0.037(0.002)	0.405(0.001)	0.190(0.005)	0.026(0.001)
	Latent ODE	0.445(0.002)	0.140(0.017)	0.095(0.004)	0.181(0.006)	0.013(0.001)
	GPT2(6)	0.467(0.002)	0.033(0.001)	0.069(0.001)	0.187(0.001)	0.013(0.001)
Probabilistic	GP-copula	0.337(0.024)	0.024(0.002)	0.078(0.002)	0.208(0.183)	0.007(0.000)
	TransMAF	0.301(0.014)	0.021(0.011)	0.056(0.001)	0.179(0.002)	0.005(0.003)
	TimeGrad	0.287(0.020)	0.021(0.001)	0.044(0.006)	**0.114(0.020)**	0.006(0.001)
	CSDI	0.298(0.004)	0.017(0.000)	0.020(0.001)	0.123(0.003)	0.007(0.001)
	Diffusion-TS	0.286(0.003)	0.019(0.002)	0.097(0.001)	0.303(0.004)	0.009(0.001)
	TimeDiT	**0.278(0.001)**	**0.017(0.000)**	**0.019(0.000)**	0.123(0.001)	**0.005(0.001)**

Table 2: Forecasting results on CRPS$_{sum}$ with full shot setting.

		Solar	Electricity	Traffic	Taxi	Exchange
Zero-shot	TEMPO	0.581(0.002)	0.081(0.003)	0.147(0.000)	0.400(0.001)	0.030(0.001)
	Moirai small	0.884(0.005)	0.079(0.002)	0.215(0.000)	0.463(0.001)	**0.007(0.000)**
	Moirai base	0.948(0.002)	0.072(0.002)	0.191(0.001)	0.428(0.000)	0.012(0.000)
	Moirai large	1.042(0.002)	0.039(0.001)	**0.111(0.000)**	0.597(0.000)	0.011(0.000)
	LagLLama	0.690(0.005)	0.065(0.005)	0.275(0.001)	0.620(0.003)	0.024(0.001)
	TimeDiT	**0.457(0.002)**	**0.026(0.001)**	0.185(0.010)	**0.398(0.001)**	0.021(0.002)

Table 3: Forecasting results on CRPS$_{sum}$. Zero-shot implies that the model has not encountered any samples from the above datasets during training.

second-best performance on the Taxi dataset. In the zero-shot setting, TimeDiT is compared with the open-sourced foundation models including TEMPO [Cao et al., 2023b], which is pre-trained with Student's t-distribution head to support probabilistic results, Moirai [Woo et al., 2024b] and LagLLama [Rasul et al., 2023] in Table 3. TimeDiT's ability to outperform other open-source foundation models in most cases is particularly noteworthy, as it suggests that TimeDiT can be effectively applied to a wide range of time series forecasting tasks across different domains with minimal adaptation.

Datasets	**ETTh1**		**ETTh2**		**ETTm1**		**ETTm2**		**Weather**		**Electricity**		**1st Pl**
	MSE	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**Count**
DLinear	0.201	0.306	0.142	0.259	0.093	0.206	0.096	0.208	0.052	0.110	0.132	0.260	0
LightTS	0.284	0.373	0.119	0.250	0.104	0.218	0.046	0.151	0.055	0.117	0.131	0.262	0
ETSformer	0.202	0.329	0.367	0.436	0.120	0.253	0.208	0.327	0.076	0.171	0.214	0.339	0
FEDformer	0.117	0.246	0.163	0.279	0.062	0.177	0.101	0.215	0.099	0.203	0.130	0.259	0
Autoformer	0.103	0.214	0.055	0.156	0.051	0.150	0.029	0.105	0.031	0.057	0.101	0.225	0
PatchTST	0.115	0.224	0.065	0.163	0.047	0.140	0.029	0.102	0.034	0.055	0.072	0.183	0
TimesNet	0.078	0.187	0.049	0.146	0.027	0.107	0.022	0.088	**0.030**	0.054	0.092	0.210	1
GPT2(3)	0.069	0.173	0.048	0.141	0.028	0.105	**0.021**	0.084	0.031	0.056	0.090	0.207	1
TimeDiT	**0.042**	**0.135**	**0.042**	**0.139**	**0.023**	**0.098**	0.024	**0.083**	0.031	**0.036**	**0.069**	**0.174**	10

Table 4: Imputation result on 96-length multivariate time series averaged over the four mask ratios. We calculate MSE and MAE for each dataset. **Bold** indicates best result, Underline indicates the second best result.

5.4 Imputation Task

We conduct experiments on six benchmark time-series datasets: ETTh1, ETTh2, ETTm1, ETTm2, Electtricity, and Weather. We use random mask ratios $\{12.5\%, 25\%, 37.5\%, 50\%\}$ following previous studies' settings with sequence length set to 96. We finetune on model checkpoints pretrained on solar, traffic, exchange, taxi, Huawei cloud, air quality, and weather (different from the evaluation weather data). Table 4 shows the imputation result averaged over the four mask ratios. TimeDiT achieves the best performance on most dataset. obtaining 10 first places out of the 12 evaluations while the remaining baselines obtained 2 first place count in total. In particular, TimeDiT achived a 39% reduction in MSE and 22% reduction in MAE compared to the strongest baseline on ETTh1 dataset. For full result on each mask ratio, please refer to section F.

5.5 Anomaly Detection Task

We conduct experiments on five real-world datasets from industrial applications: MSL, SMAP, SWaT, SMD, and PSM. The diffusion model, renowned for its proficiency in distribution learning, may inadvertently overfit by reconstructing anomalies alongside normal data points. To counteract this, we opted to bypass pretraining and introduced spectral residue (SR) transformation at the preprocessing stage of TimeDiT . This transformation helps to conceal points most likely to be anomalies and their immediate neighbors. The number of neighbors affected is controlled by the hyperparameter $n_{neighbor}$. The SR method utilizes Fourier Transformation to convert the original time series into a saliency map, thereby amplifying abnormal points, as detailed in [Ren et al., 2019, Zhao et al., 2020]. For additional information about this transformation, please see section G. Consistent with prior methodologies, we set the sequence length to be 100 identify anomalies using the 99th percentile of reconstruction errors. During evaluations, we apply standard anomaly adjustments as suggested by [Xu et al., 2018]. As demonstrated in table G, TimeDiT outperforms baseline models on four of the five datasets. In particular, TimeDiT 23.03 points of improvement in terms of F1 score on the SMAP dataset compared to the strongest baseline.

Methods	TimeDiT	GPT2(6)	TimesNet	PatchTS.	ETS.	FED.	LightTS	DLinear	Auto.	Anomaly.*
MSL	**89.33**	82.45	81.84	78.70	85.03	78.57	78.95	84.88	79.05	83.31
SMAP	**95.91**	72.88	69.39	68.82	69.50	70.76	69.21	69.26	71.12	71.18
SWaT	**96.46**	94.23	93.02	85.72	84.91	93.19	93.33	87.52	92.74	83.10
SMD	83.28	**86.89**	84.61	84.62	83.13	85.08	82.53	77.10	85.11	85.49
PSM	**97.57**	97.13	97.34	96.08	91.76	97.23	97.15	93.55	93.29	79.40
1st Pl Count	4	1	0	0	0	0	0	0	0	0

* We replace the joint criterion in Anomaly Transformer with reconstruction error for fair comparison.

Table 5: Anomaly Detection result on 100-length multivariate time series. We calculate F1 score as % for each dataset. '.' notation in model name stands for transformer. **Bold** indicates best result, Underline indicates the second best result.

Metric	Methods	Sine	Stocks	Air Quality	Energy
Discriminative Score	TimeGAN	0.1217(0.039)	0.2038(0.057)	0.3913(0.039)	0.4969 (0.000)
	TimeVAE	0.0489(0.0562)	0.1987(0.037)	0.2869(0.053)	0.4993(0.001)
	Diffusion-TS	0.0099(0.003)	0.1869(0.0159)	**0.1227(0.006)**	0.2301(0.006)
	TimeDiT	**0.0086(0.004)**	**0.0087(0.006)**	0.2109(0.003)	**0.0053(0.002)**
Predictive Score	TimeGAN	0.2797(0.015)	0.0481(0.002)	0.035(0.002)	0.3305(0.003)
	TimeVAE	0.2285(0.000)	0.0485(0.000)	0.0269(0.001)	0.2878(0.001)
	Diffusion-TS	0.2262(0.000)	**0.042(0.000)**	0.022(0.002)	0.2506(0.000)
	TimeDiT	**0.1915(0.000)**	0.0445(0.000)	**0.0217(0.000)**	**0.2489(0.000)**

Table 6: Synthetic Generation result on 24-length multivariate time series. We calculate discriminative and predictive score according to Yoon et al. [2019] and results are averaged over five runs. **Bold** indicates the best performance.

5.6 Synthetic Generation Task

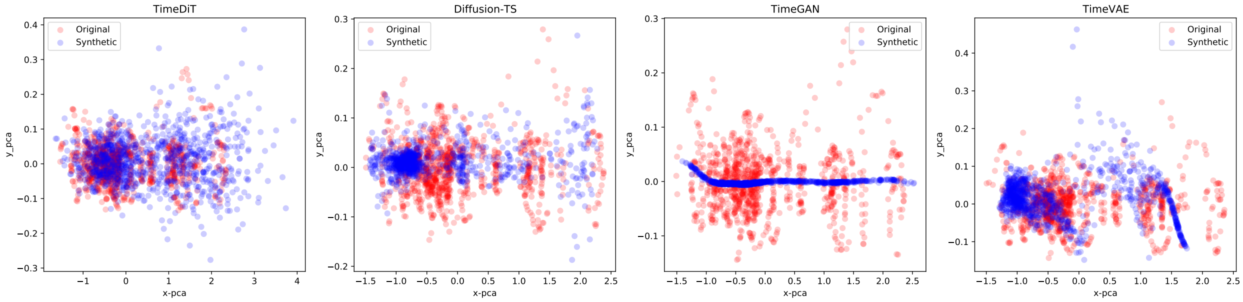


Figure 3: PCA Evaluation of Synthetic Time Series from TimeDiT and other baselines on stock dataset.

We conduct experiments to synthesize multivariate time series and evaluate performance using the discriminative score and predictive score metrics under a "train on synthetic test on real" experimental setup with sequence length set to 24 [Yuan and Qiao, 2024]. We finetune on model checkpoints pretrained on etth1, etth2, ettm1, ettm2, electricity, and weather. Table 5.5 shows the result on synthetic generation where TimeDiT, in general, consistently generates more realistic synthetic samples compared to baselines, even on high-dimensional energy datasets. This demonstrates TimeDiT's strength in complex time series synthesis. We visualize synthesis performance using PCA and t-SNE in Appendix E. As shown in Figure 3, TimeDiT's samples markedly overlap the original data distribution better than other methods. Qualitative and quantitative results confirm TimeDiT's superior ability to model intricate characteristics for realistic time series synthesis, even on multidimensional, complex datasets.

6 Conclusion

In this paper, we introduced TimeDiT, a pioneering approach to creating a versatile and robust foundation model for various time series tasks under practical scenarios. By integrating transformer architecture with diffusion model, TimeDiT effectively captures temporal dependencies and addresses real world challenges unique to time series regarding multi-resolution and missing values as well as incorporating external knowledge. Our innovative masking strategies allow for a consistent training framework adaptable to diverse tasks such as forecasting, imputation, and anomaly detection and synthetic data generation. Extensive experiments demonstrated the strong performance of TimeDiT on both practical scenarios and standard benchmarks. However, we recognize some limitations. We primarily explored common sequence lengths and did not assess TimeDiT's performance on very long sequences. While we have introduced randomness in prediction length and feature numbers up to a maximum, we aim to develop more scalable solutions for highly variable multivariate time series. Additionally, our understanding of how different types of external information contribute to performance is still developing. For future work, we envision several key directions: enhancing scalability to improve TimeDiT's ability to handle practical time series with varying multivariate numbers; developing techniques for seamless multi-modal integration, allowing TimeDiT to leverage diverse data sources for improved performance across different tasks; and extending TimeDiT's capabilities to effectively process and analyze very long time series sequences, addressing a critical need in many real-world applications.

References

A. Abdulaal, Z. Liu, and T. Lancewicki. Practical approach to asynchronous multivariate time series anomaly detection and localization. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2485–2494, 2021.

A. Alexandrov, K. Benidis, M. Bohlke-Schneider, V. Flunkert, J. Gasthaus, T. Januschowski, D. C. Maddix, S. Rangapuram, D. Salinas, J. Schulz, L. Stella, A. C. Trkmen, and Y. Wang. Gluonts: Probabilistic and neural time series modeling in python. *Journal of Machine Learning Research*, 21(116):1–6, 2020. URL `http://jmlr.org/papers/v21/19-820.html`.

A. F. Ansari, L. Stella, C. Turkmen, X. Zhang, P. Mercado, H. Shen, O. Shchur, S. S. Rangapuram, S. P. Arango, S. Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

A. Asuncion and D. Newman. Uci machine learning repository, 2007.

K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619(7970):533–538, 2023.

D. Cao, Y. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, Y. Tong, B. Xu, J. Bai, J. Tong, et al. Spectral temporal graph neural network for multivariate time-series forecasting. *Advances in neural information processing systems*, 33: 17766–17778, 2020.

D. Cao, Y. El-Laham, L. Trinh, S. Vyetrenko, and Y. Liu. A synthetic limit order book dataset for benchmarking forecasting algorithms under distributional shift. In *NeurIPS 2022 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2022.

D. Cao, J. Enouen, Y. Wang, X. Song, C. Meng, H. Niu, and Y. Liu. Estimating treatment effects from irregular time series observations with hidden confounders. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 6897–6905, 2023a.

D. Cao, F. Jia, S. O. Arik, T. Pfister, Y. Zheng, W. Ye, and Y. Liu. Tempo: Prompt-based generative pre-trained transformer for time series forecasting. *arXiv preprint arXiv:2310.04948*, 2023b.

D. Cao, Y. Zheng, P. Hassanzadeh, S. Lamba, X. Liu, and Y. Liu. Large scale financial time series forecasting with multi-faceted model. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pages 472–480, 2023c.

C. Chang, W.-C. Peng, and T.-F. Chen. Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms. *arXiv preprint arXiv:2308.08469*, 2023.

S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88, 2022.

A. Das, W. Kong, R. Sen, and Y. Zhou. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*, 2023.

A. Desai, C. Freeman, Z. Wang, and I. Beaver. Timevae: A variational auto-encoder for multivariate time series generation. *arXiv preprint arXiv:2111.08095*, 2021.

J. Dong, H. Wu, H. Zhang, L. Zhang, J. Wang, and M. Long. Simmtm: A simple pre-training framework for masked time-series modeling. *Advances in Neural Information Processing Systems*, 36, 2024.

W. Fan, S. Zheng, P. Wang, R. Xie, J. Bian, and Y. Fu. Addressing distribution shift in time series forecasting with instance normalization flows. *arXiv e-prints*, pages arXiv–2401, 2024.

A. Garza and M. Mergenthaler-Canseco. Timegpt-1. *arXiv preprint arXiv:2310.03589*, 2023.

R. Godahewa, C. Bergmeir, G. I. Webb, R. J. Hyndman, and P. Montero-Manso. Monash time series forecasting archive. In *Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

N. Gruver, M. Finzi, S. Qiu, and A. G. Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36, 2024.

J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 387–395, 2018.

F. Jia, K. Wang, Y. Zheng, D. Cao, and Y. Liu. Gpt4mts: Prompt-based large language model for multimodal time-series forecasting. In *The 14th Symposium on Educational Advances in Artificial Intelligence (EAAI-24)*, 2024.

Y. Jiang, Z. Pan, X. Zhang, S. Garg, A. Schneider, Y. Nevmyvaka, and D. Song. Empowering time series analysis with large language models: A survey. *arXiv preprint arXiv:2402.03182*, 2024.

M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.

M. Jin, Y. Zhang, W. Chen, K. Zhang, Y. Liang, B. Yang, J. Wang, S. Pan, and Q. Wen. Position paper: What can large language models tell us about time series analysis. *arXiv preprint arXiv:2402.02713*, 2024.

A. Joosen, A. Hassan, M. Asenov, R. Singh, L. Darlow, J. Wang, and A. Barker. How does it function? characterizing long-term trends in production serverless workloads. In *Proceedings of the 2023 ACM Symposium on Cloud Computing*, pages 443–458, 2023.

N. Kamra, Y. Zhang, S. Rambhatla, C. Meng, and Y. Liu. Polsird: modeling epidemic spread under intervention policies: analyzing the first wave of covid-19 in the usa. *Journal of Healthcare Informatics Research*, 5(3):231–248, 2021.

S. Kaushik, A. Choudhury, P. K. Sheron, N. Dasgupta, S. Natarajan, L. A. Pickett, and V. Dutt. Ai in healthcare: time-series forecasting using statistical, neural, and ensemble architectures. *Frontiers in big data*, 3:4, 2020.

P. Kidger, J. Morrill, J. Foster, and T. Lyons. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems*, 33:6696–6707, 2020.

M. Kollovieh, A. F. Ansari, M. Bohlke-Schneider, J. Zschiegner, H. Wang, and B. Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=q6X038vKgU`.

M. Krenn and L. Buffoni. Predicting the future of ai with ai: High-quality link prediction in an exponentially growing knowledge network. *Nature machine intelligence*, 2023.

G. Lai, W.-C. Chang, Y. Yang, and H. Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 95–104, 2018.

X. Li, J. Thickstun, I. Gulrajani, P. S. Liang, and T. B. Hashimoto. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022a.

Y. Li, R. Yu, C. Shahabi, and Y. Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR '18)*, 2018.

Y. Li, X. Lu, Y. Wang, and D. Dou. Generative time series forecasting with diffusion, denoise, and disentanglement. *Advances in Neural Information Processing Systems*, 35:23009–23022, 2022b.

Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=c8P9NQVtmnO`.

Y. Liang, H. Wen, Y. Nie, Y. Jiang, M. Jin, D. Song, S. Pan, and Q. Wen. Foundation models for time series analysis: A tutorial and survey. *arXiv preprint arXiv:2403.14735*, 2024.

Y. Liu, H. Zhang, C. Li, X. Huang, J. Wang, and M. Long. Timer: Transformers for time series analysis at scale. *arXiv preprint arXiv:2402.02368*, 2024.

H. Lu, G. Yang, N. Fei, Y. Huo, Z. Lu, P. Luo, and M. Ding. VDT: General-purpose video diffusion transformers via mask modeling. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=Un0rgm9f04`.

A. P. Mathur and N. O. Tippenhauer. Swat: A water treatment testbed for research and training on ics security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*, pages 31–36. IEEE, 2016.

C. Meng, H. Niu, G. Habault, R. Legaspi, S. Wada, C. Ono, and Y. Liu. Physics-informed long-sequence forecasting from multi-resolution spatiotemporal data. In *IJCAI*, pages 2189–2195, 2022.

Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations (ICLR '23)*, 2023.

H. Niu, C. Meng, D. Cao, G. Habault, R. Legaspi, S. Wada, C. Ono, and Y. Liu. Mu2rest: Multi-resolution recursive spatio-temporal transformer for long-term prediction. In *Advances in Knowledge Discovery and Data Mining: 26th Pacific-Asia Conference, PAKDD 2022, Chengdu, China, May 16–19, 2022, Proceedings, Part I*, page 68–80, Berlin, Heidelberg, 2022. Springer-Verlag. ISBN 978-3-031-05932-2. doi: 10.1007/978-3-031-05933-9_6. URL `https://doi.org/10.1007/978-3-031-05933-9_6`.

H. Niu, G. Habault, R. Legaspi, C. Meng, D. Cao, S. Wada, C. Ono, and Y. Liu. Time-delayed multivariate time series predictions. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pages 325–333. SIAM, 2023.

OpenAI. Gpt-4 technical report, 2023.

Z. Pan, Y. Jiang, S. Garg, A. Schneider, Y. Nevmyvaka, and D. Song. $\mathbf{S}^2$ ip-llm: Semantic space informed prompt learning with llm for time series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.

W. Peebles and S. Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.

A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

K. Rasul, C. Seward, I. Schuster, and R. Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, pages 8857–8868. PMLR, 2021.

K. Rasul, A. Ashok, A. R. Williams, A. Khorasani, G. Adamopoulos, R. Bhagwatkar, M. Biloš, H. Ghonia, N. V. Hassen, A. Schneider, et al. Lag-llama: Towards foundation models for time series forecasting. *arXiv preprint arXiv:2310.08278*, 2023.

H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, and Q. Zhang. Time-series anomaly detection service at microsoft. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3009–3017, 2019.

R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

D. Salinas, M. Bohlke-Schneider, L. Callot, R. Medico, and J. Gasthaus. High-dimensional multivariate forecasting with low-rank gaussian copula processes. *Advances in neural information processing systems*, 32, 2019.

K. Sharma, Y. Zhang, E. Ferrara, and Y. Liu. Identifying coordinated accounts on social media through hidden influence and group behaviours. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1441–1451, 2021.

J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*.

G. Stoltz, M. Rousset, et al. *Free energy computations: A mathematical perspective*. World Scientific, 2010.

Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2828–2837, 2019.

Y. Sui, Y. Li, A. Kag, Y. Idelbayev, J. Cao, J. Hu, D. Sagar, B. Yuan, S. Tulyakov, and J. Ren. Bitsfusion: 1.99 bits weight quantization of diffusion model. *arXiv preprint arXiv:2406.04333*, 2024a.

Y. Sui, H. Phan, J. Xiao, T. Zhang, Z. Tang, C. Shi, Y. Wang, Y. Chen, and B. Yuan. Disdet: Exploring detectability of backdoor attack on diffusion models. *arXiv preprint arXiv:2402.02739*, 2024b.

C. Sun, Y. Li, H. Li, and S. Hong. Test: Text prototype aligned embedding to activate llm's ability for time series. In *The Twelfth International Conference on Learning Representations*, 2023.

M. Takamoto, T. Praditia, R. Leiteritz, D. MacKinlay, F. Alesiani, D. Pflüger, and M. Niepert. Pdebench: An extensive benchmark for scientific machine learning. *Advances in Neural Information Processing Systems*, 35:1596–1611, 2022.

S. Talukder, Y. Yue, and G. Gkioxari. Totem: Tokenized time series embeddings for general time series analysis, 2024.

Y. Tashiro, J. Song, Y. Song, and S. Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021.

N. Tlc. Nyc taxi and limousine commission (tlc) trip record data. *URL http://www. nyc. gov/html/tlc/html/about/trip record data. shtml*, 2017.

H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971, 2023. URL `https://api.semanticscholar.org/CorpusID:257219404`.

S. Tuli, G. Casale, and N. R. Jennings. Tranad: deep transformer networks for anomaly detection in multivariate time series data. *Proceedings of the VLDB Endowment*, 15(6):1201–1214, 2022.

R. Wang, K. Kashinath, M. Mustafa, A. Albert, and R. Yu. Towards physics-informed deep learning for turbulent flow prediction. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1457–1466, 2020.

R. Wang, Y. Dong, S. O. Arik, and R. Yu. Koopman neural operator forecaster for time-series with temporal distributional shifts. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=kUmdmHxK5N`.

G. Woo, C. Liu, A. Kumar, C. Xiong, S. Savarese, and D. Sahoo. Unified training of universal time series forecasting transformers. *arXiv preprint arXiv:2402.02592*, 2024a.

G. Woo, C. Liu, A. Kumar, C. Xiong, S. Savarese, and D. Sahoo. Unified training of universal time series forecasting transformers. In *Forty-first International Conference on Machine Learning*, 2024b.

X. Xiao, D. Cao, R. Yang, G. Gupta, G. Liu, C. Yin, R. Balan, and P. Bogdan. Coupled multiwavelet neural operator learning for coupled partial differential equations. *ICLR 2023*, 2023.

H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 world wide web conference*, pages 187–196, 2018.

J. Xu, H. Wu, J. Wang, and M. Long. Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint arXiv:2110.02642*, 2021.

Y. Yang, M. Jin, H. Wen, C. Zhang, Y. Liang, L. Ma, Y. Wang, C. Liu, B. Yang, Z. Xu, et al. A survey on diffusion models for time series and spatio-temporal data. *arXiv preprint arXiv:2404.18886*, 2024.

W. Ye and S. Gao. Spatiotemporal heterogeneities of the associations between human mobility and close contacts with covid-19 infections in the united states. In *Proceedings of the 30th International Conference on Advances in Geographic Information Systems*, pages 1–2, 2022.

J. Yoon, D. Jarrett, and M. Van der Schaar. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32, 2019.

X. Yuan and Y. Qiao. Diffusion-ts: Interpretable diffusion for general time series generation. *arXiv preprint arXiv:2403.01742*, 2024.

Z. Yue, Y. Wang, J. Duan, T. Yang, C. Huang, Y. Tong, and B. Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8980–8987, 2022.

A. Zeng, M. Chen, L. Zhang, and Q. Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2114–2124, 2021.

K. Zhang, Q. Wen, C. Zhang, R. Cai, M. Jin, Y. Liu, J. Y. Zhang, Y. Liang, G. Pang, D. Song, et al. Self-supervised learning for time series analysis: Taxonomy, progress, and prospects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024a.

Y. Zhang, K. Sharma, and Y. Liu. Vigdet: Knowledge informed neural temporal point process for coordination detection on social media. *Advances in Neural Information Processing Systems*, 34:3218–3231, 2021.

Y. Zhang, D. Cao, and Y. Liu. Counterfactual neural temporal point process for estimating causal influence of misinformation on social media. *Advances in Neural Information Processing Systems*, 35:10643–10655, 2022.

Y. Zhang, R. Wu, S. M. Dascalu, and F. C. Harris. Multi-scale transformer pyramid networks for multivariate time series forecasting. *IEEE Access*, 2024b.

H. Zhao, Y. Wang, J. Duan, C. Huang, D. Cao, Y. Tong, B. Xu, J. Bai, J. Tong, and Q. Zhang. Multivariate time-series anomaly detection via graph attention network. In *2020 IEEE international conference on data mining (ICDM)*, pages 841–850. IEEE, 2020.

H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of AAAI*, 2021.

T. Zhou, P. Niu, L. Sun, R. Jin, et al. One fits all: Power general time series analysis by pretrained lm. *Advances in neural information processing systems*, 36:43322–43355, 2023.

A Datasets

1. The ETT datasets Zhou et al. [2021][2] include electricity load data at various resolutions (ETTh & ETTm) from two different electricity stations.
2. The Weather dataset comprises 21 meteorological indicators collected in Germany over the span of one year.
3. The Electricity dataset provides information on electricity consumption.
4. The SMD dataset Su et al. [2019] includes multivariate time-series data collected from server machines in a data center. It typically contains metrics such as CPU usage, memory usage, and disk activity.
5. The PSM dataset Abdulaal et al. [2021]is used for predictive maintenance and includes sensor data from industrial machines. It often contains readings such as temperature, pressure, and vibration over time.
6. The MSL dataset Hundman et al. [2018] comes from the Mars Science Laboratory mission, specifically the Curiosity rover. It includes telemetry data from the rover's sensors and systems.
7. The SWaT dataset Mathur and Tippenhauer [2016] originates from a scaled-down water treatment testbed designed to reflect a real-world water treatment process. It includes sensor and actuator data collected over time.
8. The SMAP dataset Hundman et al. [2018]comes from NASA's Soil Moisture Active Passive (SMAP) mission, which measures soil moisture and freeze/thaw state. It includes time-series data from multiple sensors aboard the SMAP satellite.
9. The Sine dataset Yoon et al. [2019] is synthetically generated by sinusoidal waves.
10. The Air Quality dataset [3]contains hourly averaged readings from five metal oxide chemical sensors integrated into an Air Quality Chemical Multisensor Device. This device was positioned at road level in a highly polluted area of an Italian city. Data were collected from March 2004 to February 2005, making it the longest freely available record of on-field air quality chemical sensor responses.
11. The Stock dataset [4] contains daily historical Google stocks data from 2004 to 2019.
12. The UCI Appliances Energy prediction dataset [5]consists of multivariate, continuous-valued measurements including numerous temporal features measured at close intervals.
13. The Cloud dataset: The Huawei cloud datasets contain serverless traces [Joosen et al., 2023]. Following [Rasul et al., 2023], we selected 8 time series containing metrics based on the minute-frequency occurrences of the top 10 functions over a period of 141 days. The metrics included in these series are: Function delay; Platform delay; CPU usage; Memory usage; CPU limit; Memory limit; Instances; Requests. The functions were chosen based on their median occurrences throughout the dataset.
14. The Weather_2 dataset: The Weather_2 dataset comprises hourly climate time series data collected near Monash University, Clayton, Victoria, Australia, from January 2010 to May 2021. It includes series for temperature, dewpoint temperature, wind speed, mean sea level pressure, relative humidity, surface solar radiation, surface thermal radiation, and total cloud cover [Godahewa et al., 2021].

B Forecasting Experiment Setting

For the forecasting task, we utilized five widely-used open datasets to evaluate probabilistic time series forecasting performance. These datasets were collected in GluonTS [Alexandrov et al., 2020] and have been previously employed in [Tashiro et al., 2021, Salinas et al., 2019]:

- Solar[6]: Hourly solar power production records from 137 stations in Alabama State, as used in [Lai et al., 2018].
- Electricity[7]: Hourly time series of electricity consumption for 370 customers, as used in [Asuncion and Newman, 2007].

[2] ETT: `https://github.com/zhouhaoyi/ETDataset`

[3] Air Quality: `https://archive.ics.uci.edu/dataset/360/air+quality`

[4] Stock: `https://finance.yahoo.com/quote/GOOG`

[5] Energy: `https://archive.ics.uci.edu/ml/datasets`

[6] Solar: `https://www.nrel.gov/grid/solar-power-data.html`

[7] Electricity: `https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014`

Table 7: Dataset details

Dataset	Domain	Length	Dimension	Frequency
ETTh	Energy	17420	7	1 hour
ETTm	Energy	69680	7	15 min
Weather	Nature	52696	21	10 min
Electricity	Energy	26304	321	1 hour
Air Quality	Nature	9357	13	1 hour
Sine	Synthetic	10000	5	N/A
Stock	Finance	3685	6	1 day
Energy	Energy	19745	28	10 min
MSL	Space	132046	55	1 min
PSM	Cloud	220322	25	1 min
SMAP	Space	562800	25	1 min
SMD	Cloud	1416825	38	1 min
SWaT	Energy	944920	51	1 second
Requests Minute	Cloud	64800	10	1 min
Function Delay Minute	Cloud	64800	10	1 min
Platform Delay Minute	Cloud	64800	10	1 min
Memory Usage Minute	Cloud	64800	10	1 min
CPU Limit Minute	Cloud	64800	10	1 min
Memory Limit Minute	Cloud	64800	10	1 min
Instances Minute	Cloud	64800	10	1 min
Weather_2	Climate	3001	695	1 day

- Traffic[8]: Hourly occupancy rates of 963 San Francisco freeway car lanes, with values between 0 and 1 [Asuncion and Newman, 2007].
- Taxi[9]: Half-hourly spatio-temporal time series of New York taxi rides taken at 1,214 locations, using data from January 2015 for training and January 2016 for testing, as proposed in [Tlc, 2017].
- Exchange rate[10]: Daily exchange rates between 8 currencies, namely Australia, the United Kingdom, Canada, Switzerland, China, Japan, New Zealand, and Singapore, as used in [Lai et al., 2018].

Table 8 summarizes the characteristics of each dataset. The task for these datasets is to predict the future L_2 steps given the observed L_1 steps. We set L_1 and L_2 values based on previous studies [Tashiro et al., 2021, Salinas et al., 2019]. For training, we randomly selected $L_1 + L_2$ consecutive time steps as a single time series and designated the last L_2 steps as forecasting targets. We adhered to the train/test splits used in previous studies and utilized the last five samples of the training data as validation data.

For the full-shot setting, we trained separate models on different datasets. Due to the large number of features in multivariate time series, we adopted subset sampling of features for training. For each input, we split them into subsets based on their order. If the last subset was smaller than the fixed shape, we applied padding to ensure equal input sizes across all subsets. In the multi-resolution setting, we used different resolutions identified by the resolution number, which corresponded to different sampling rates for the exchange features. It is worth to note that the aforementioned strategy was also employed for zero-shot training, as the input feature length varied across datasets.

C Training details

The codebase for TimeDiT is modified from `https://github.com/facebookresearch/DiT`, where they provide different model sizes, including Small (S), Big (B), Large (L), Extra Large (XL) [Peebles and Xie, 2022]. In our training, we used Adam optimizer with a training rate of 0.0001 without weight decay. Batch size is set to 512. The maximum channel number K_{max} is set to 40. All experiments are run on NVIDIA A100 GPUs. The zero-shot foundation model was trained on the ETT, weather, illness, air quality, and cloud datasets and used for different downstream tasks. We will include more available time series datasets to develop a more robust time series foundation model as the future work.

[8] Traffic_nips: `https://archive.ics.uci.edu/dataset/204/pems_sf`

[9] Taxi: `https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data`

[10] Exchange: `https://github.com/laiguokun/multivariate-time-series-data`

Table 8: Evaluate time series dataset for forecasting tasks.

	Features	Time Steps	History Length (L_1)	Prediction Horizon (L_2)	Rolling Windows	Frequency	Domain
Solar	137	7009	168	24	7	1 hour	$\mathbb{R}^+$
Electricity	370	5833	168	24	7	1 hour	$\mathbb{R}^+$
Traffic	963	4001	168	24	7	1 hour	(0, 1)
Taxi	1214	1488	48	24	56	30 mins	$\mathbb{N}$
Exchange	8	6071	60	30	5	1 day	$\mathbb{R}^+$

Table 9: Training Details. Imp stands for Imputation. SG stands for Syntheric Generation. AD stands for Anomaly Detection. FC stands for Forecasting

Dataset	Task	Loss	model size	hidden size	attention head	depth
ETTh	Imp	L_{simple}	S	384	6	12
ETTm	Imp	L_{simple}	S	384	6	12
Weather	Imp	L_{simple}	S	384	6	12
Electricity	Imp	L_{simple}	S	384	6	12
Air Quality	Imp	L	S	384	6	12
Sine	SG	L	S	384	6	12
Stock	SG	L	S	384	6	12
Energy	SG	L	S	384	6	12
MSL	AD	L_{simple}	S	384	6	12
PSM	AD	L_{simple}	S	384	6	12
SMAP	AD	L_{simple}	S	384	6	12
SMD	AD	L_{simple}	S	384	6	12
SWaT	AD	L_{simple}	S	384	6	12
Solar	FC	L_{simple}	B	768	12	12
Taxi	FC	L_{simple}	B	768	12	12
Traffic	FC	L_{simple}	B	768	12	12
Exchange	FC	L_{simple}	B	768	12	12
Electricity	FC	L_{simple}	B	768	12	12

D Metrics

MAE describes the mean absolute error that measures the absolute difference between ground truth and prediction.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{15}$$

MSE describes the mean squared difference between ground truth and prediction.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{16}$$

RMSE is the sqaure root of MSE.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{17}$$

Discriminative score Following TimeGAN, we train a post-hoc time-series classification model (by optimizing a 2-layer LSTM) to distinguish between sequences from the original and generated datasets. First, each original sequence is labeled real, and each generated sequence is labeled not real. Then, an off-the-shelf (RNN) classifier is trained to distinguish between the two classes as a standard supervised task. We then report the classification error on the held-out test set.

Predictive Score Following TimeGAN, we train a post-hoc sequence-prediction model (by optimizing a 2-layer LSTM) to predict next-step temporal vectors over each input sequence. Then, we evaluate the trained model on the

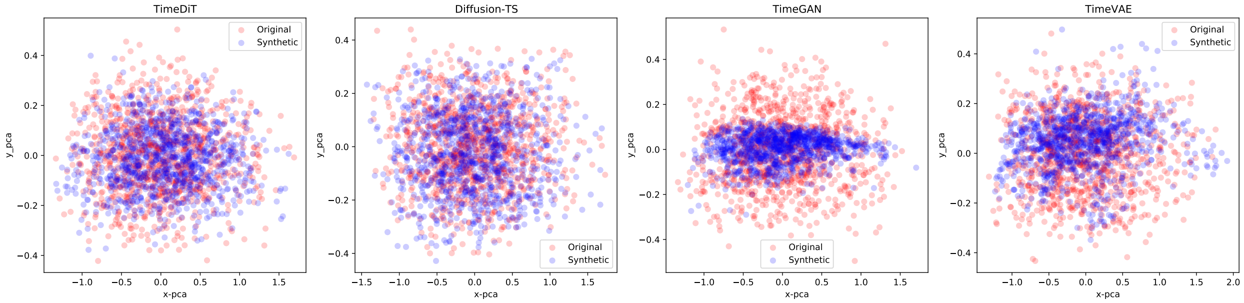


Figure 4: PCA plot for sine dataset.

original dataset. Performance is measured in terms of the mean absolute error (MAE); for event-based data, the MAE is computed as the absolute value of 1 - estimated probability that the event occured.

Computations of CRPS We explain the definition and calculation of the CRPS metric. The continuous ranked probability score (CRPS) assesses how well an estimated probability distribution F aligns with an observation x. It is defined as the integral of the quantile loss $\Lambda_\alpha(q,z) := (\alpha - \mathbf{1}_{z<q})(z-q)$ over all quantile levels $\alpha \in [0,1]$:

$$\text{CRPS}(F^{-1}, x) = \int_0^1 2\Lambda_\alpha(F^{-1}(\alpha), x)\, d\alpha$$

where $\mathbf{1}$ represents the indicator function. We then calculated quantile losses for quantile levels discretized in 0.05 increments. Thus, we approximated CRPS as follows:

$$\text{CRPS}(F^{-1}, x) \approx \frac{1}{19}\sum_{i=1}^{19} 2\Lambda_{i\cdot 0.05}(F^{-1}(i \cdot 0.05), x).$$

Next, we computed the normalized average CRPS for all features and time steps:

$$\frac{\sum_{k,l} \text{CRPS}(F_{k,l}^{-1}, x_{k,l})}{\sum_{k,l} |x_{k,l}|}$$

where k and l denote the features and time steps of the imputation targets, respectively.

CRPS$_{sum}$ measures CRPS for the distribution F of the sum of all K features, calculated by:

$$\frac{\sum_{l} \text{CRPS}(F^{-1}, \sum_k x_{k,l})}{\sum_{k,l} |x_{k,l}|}$$

where $\sum_k x_{k,l}$ is the total of the forecasting targets for all features at time point l.

E Synthetic Generation

We use 80% of all data for training and evaluate on the same data. For the air quality dataset, previous methods did not carefully the -200 values as placeholder for missing values. In our experiment, we masked all the -200 values for TimeDiT and baselines that support masks. For baselines that do not support mask, we replace -200 with the mean value. Minmax scaler is used for all models. Figure 4, 3,5,6 shows the PCA plots for all datasets and baselines. The visual comparison also validates the superiority of TimeDiT.

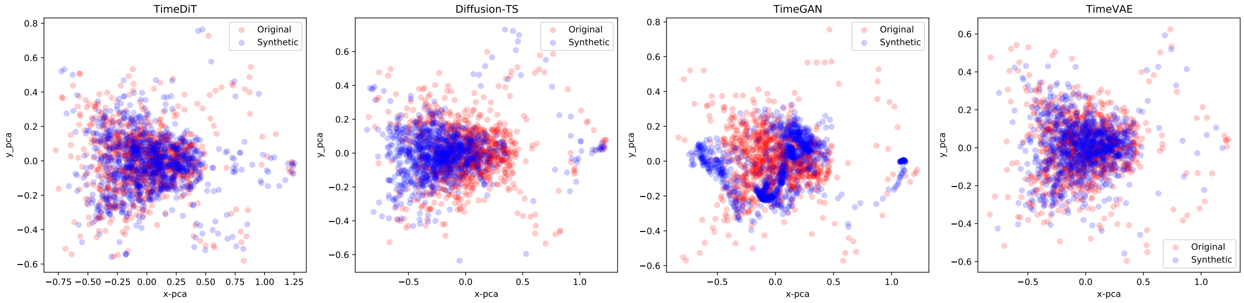


Figure 5: PCA plot for air quality dataset.

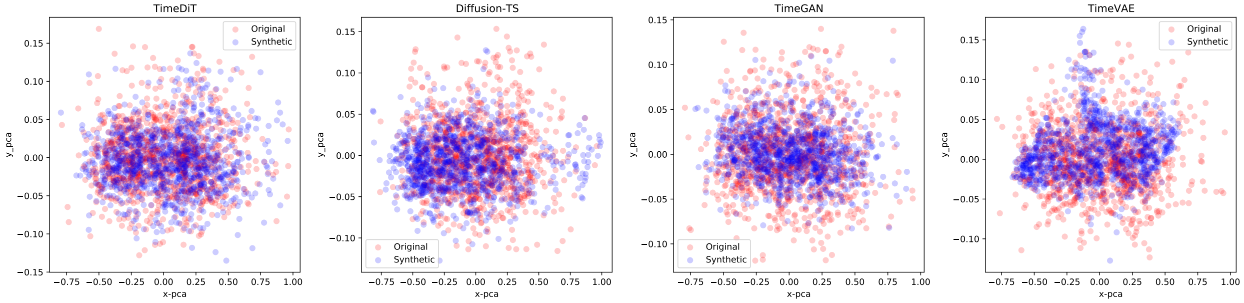


Figure 6: PCA plot for energy dataset.

Methods		TimeDiT		GPT2(3)		TimesNet		PatchTST		ETSformer		LightTS		DLinear		FEDformer		Stationary		Autoformer		Informer		Reformer	
Mask	Ratio	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
$ETTh1$	12.5%	**0.025**	**0.101**	0.043	0.140	0.057	0.159	0.093	0.201	0.126	0.263	0.240	0.345	0.151	0.267	0.070	0.190	0.060	0.165	0.074	0.182	0.114	0.234	0.074	0.194
	25%	**0.034**	**0.122**	0.054	0.156	0.069	0.178	0.107	0.217	0.169	0.304	0.265	0.364	0.180	0.292	0.106	0.236	0.080	0.189	0.090	0.203	0.140	0.262	0.102	0.227
	37.5%	**0.047**	**0.143**	0.072	0.180	0.084	0.196	0.120	0.230	0.220	0.347	0.296	0.382	0.215	0.318	0.124	0.258	0.102	0.212	0.109	0.222	0.174	0.293	0.135	0.261
	50%	**0.063**	**0.166**	0.107	0.216	0.102	0.215	0.141	0.248	0.293	0.402	0.334	0.404	0.257	0.347	0.165	0.299	0.133	0.240	0.137	0.248	0.215	0.325	0.179	0.298
	Avg	**0.042**	**0.135**	0.069	0.173	0.078	0.187	0.115	0.224	0.202	0.329	0.284	0.373	0.201	0.306	0.117	0.246	0.094	0.201	0.103	0.214	0.161	0.279	0.122	0.245
$ETTh2$	12.5%	**0.025**	**0.107**	0.039	0.125	0.040	0.130	0.057	0.152	0.187	0.319	0.101	0.231	0.100	0.216	0.095	0.212	0.042	0.133	0.044	0.138	0.305	0.431	0.163	0.289
	25%	**0.037**	**0.129**	0.044	0.135	0.046	0.141	0.061	0.158	0.279	0.390	0.115	0.246	0.127	0.247	0.137	0.258	0.049	0.147	0.050	0.149	0.322	0.444	0.206	0.331
	37.5%	**0.046**	0.149	0.051	**0.147**	0.052	0.151	0.067	0.166	0.400	0.465	0.126	0.257	0.158	0.276	0.187	0.304	0.056	0.158	0.060	0.163	0.353	0.462	0.252	0.370
	50%	0.062	0.173	**0.059**	**0.158**	0.060	0.162	0.073	0.174	0.602	0.572	0.136	0.268	0.183	0.299	0.232	0.341	0.065	0.170	0.068	0.173	0.369	0.472	0.316	0.419
	Avg	**0.042**	**0.139**	0.048	0.141	0.049	0.146	0.065	0.163	0.367	0.436	0.119	0.250	0.142	0.259	0.163	0.279	0.053	0.152	0.055	0.156	0.337	0.452	0.234	0.352
$ETTm1$	12.5%	**0.016**	**0.083**	0.017	0.085	0.023	0.101	0.041	0.130	0.096	0.229	0.093	0.206	0.080	0.193	0.052	0.166	0.032	0.119	0.046	0.144	0.063	0.180	0.042	0.146
	25%	**0.019**	**0.091**	0.022	0.096	0.023	0.101	0.044	0.135	0.096	0.229	0.093	0.206	0.080	0.193	0.052	0.166	0.032	0.119	0.046	0.144	0.063	0.180	0.042	0.146
	37.5%	**0.025**	**0.102**	0.029	0.111	0.029	0.111	0.049	0.143	0.133	0.271	0.113	0.231	0.103	0.219	0.069	0.191	0.039	0.131	0.057	0.161	0.079	0.200	0.063	0.182
	50%	**0.032**	**0.115**	0.040	0.128	0.036	0.124	0.055	0.151	0.186	0.323	0.134	0.255	0.132	0.248	0.089	0.218	0.047	0.145	0.067	0.174	0.093	0.218	0.082	0.208
	Avg	**0.023**	**0.098**	0.028	0.105	0.027	0.107	0.047	0.140	0.120	0.253	0.104	0.218	0.093	0.206	0.062	0.177	0.036	0.126	0.051	0.150	0.071	0.188	0.055	0.166
$ETTm2$	12.5%	**0.016**	**0.065**	0.017	0.076	0.018	0.080	0.026	0.094	0.108	0.239	0.034	0.127	0.062	0.166	0.056	0.159	0.021	0.088	0.023	0.092	0.133	0.270	0.108	0.228
	25%	0.022	**0.078**	**0.020**	0.080	**0.020**	0.085	0.028	0.099	0.164	0.294	0.042	0.143	0.085	0.196	0.080	0.195	0.024	0.096	0.026	0.101	0.135	0.272	0.136	0.262
	37.5%	0.027	0.089	**0.022**	**0.087**	0.023	0.091	0.030	0.104	0.237	0.356	0.051	0.159	0.106	0.222	0.110	0.231	0.027	0.103	0.030	0.108	0.155	0.293	0.175	0.300
	50%	0.031	0.099	**0.025**	**0.095**	0.026	0.098	0.034	0.110	0.323	0.421	0.059	0.174	0.131	0.247	0.156	0.276	0.030	0.108	0.035	0.119	0.200	0.333	0.211	0.329
	Avg	0.024	**0.083**	**0.021**	0.084	0.022	0.088	0.029	0.102	0.208	0.327	0.046	0.151	0.096	0.208	0.101	0.215	0.026	0.099	0.029	0.105	0.156	0.292	0.157	0.280
ECL	12.5%	**0.051**	**0.148**	0.080	0.194	0.085	0.202	0.055	0.160	0.196	0.321	0.102	0.229	0.092	0.214	0.107	0.237	0.093	0.210	0.089	0.210	0.218	0.326	0.190	0.308
	25%	**0.061**	**0.163**	0.087	0.203	0.089	0.206	0.065	0.175	0.207	0.332	0.121	0.252	0.118	0.247	0.120	0.251	0.097	0.214	0.096	0.220	0.219	0.326	0.197	0.312
	37.5%	**0.074**	**0.181**	0.094	0.211	0.094	0.213	0.076	0.189	0.219	0.344	0.141	0.273	0.144	0.276	0.136	0.266	0.102	0.220	0.104	0.229	0.222	0.328	0.203	0.315
	50%	**0.090**	**0.202**	0.101	0.220	0.100	0.221	0.091	0.208	0.235	0.357	0.160	0.293	0.175	0.305	0.158	0.284	0.108	0.228	0.113	0.239	0.228	0.331	0.210	0.319
	Avg	**0.069**	**0.174**	0.090	0.207	0.092	0.210	0.072	0.183	0.214	0.339	0.131	0.262	0.132	0.260	0.130	0.259	0.100	0.218	0.101	0.225	0.222	0.328	0.200	0.313
$Weather$	12.5%	0.029	**0.033**	0.026	0.049	**0.025**	0.045	0.029	0.049	0.057	0.141	0.047	0.101	0.039	0.084	0.041	0.107	0.027	0.051	0.026	0.047	0.037	0.093	0.031	0.076
	25%	0.031	**0.033**	**0.028**	0.052	0.029	0.052	0.031	0.053	0.065	0.155	0.052	0.111	0.048	0.103	0.064	0.163	0.029	0.056	0.030	0.054	0.042	0.100	0.035	0.082
	37.5%	0.034	**0.037**	0.033	0.060	**0.031**	0.057	0.035	0.058	0.081	0.180	0.058	0.121	0.057	0.117	0.107	0.229	0.033	0.062	0.032	0.060	0.049	0.111	0.040	0.091
	50%	**0.031**	**0.041**	0.037	0.065	0.034	0.062	0.038	0.063	0.102	0.207	0.065	0.133	0.066	0.134	0.183	0.312	0.0377	0.068	0.037	0.067	0.053	0.114	0.046	0.099
	Avg	0.031	**0.036**	0.031	0.056	**0.030**	0.054	0.060	0.144	0.076	0.171	0.055	0.117	0.052	0.110	0.099	0.203	0.032	0.059	0.031	0.057	0.045	0.104	0.038	0.087

Table 10: Full imputation result.

F Imputation

G Anomaly Detection

G.1 SR processing for Anomaly Detection

Spectral Residue The SR Transformation involves the following equations. Table G shows the full anomaly detection results.

$$A(f) = \text{Amplitude}(F(x)) \tag{18}$$

$$P(f) = \text{Phase}(F(x)) \tag{19}$$

$$L(f) = \log(A(f)) \tag{20}$$

$$AL(f) = h_q(f) \cdot L(f) \tag{21}$$

$$R(f) = L(f) - AL(f) \tag{22}$$

$$S(x) = F^{-1}(\exp(R(f) + iP(f))) \tag{23}$$

H Physics Equations

The Burgers Equation is:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} - v\frac{\partial^2 u}{\partial x^2} = 0 \tag{24}$$

where v is the diffusion term. We set the v (diffusion term) as 0.1 and randomly sample a combination of sine waves as initial status

The Advection Equation is:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \tag{25}$$

where c is the advection speed. We set the c as 1.0 and randomly placed Gaussian peaks as initial status

The diffusion-reaction Equation is:

$$\frac{\partial u}{\partial t} - D\frac{\partial^2 u}{\partial x^2} - R(u) = 0 \tag{26}$$

where D is the diffusion coefficient and $R(u)$ is the reaction term. Here, we apply a linear reaction term $R(u) = -k \cdot u$, where k is the reaction speed. We set the D as 1.0, k as 0.1, and a Gaussian distribution with random parameters as initial status.

The Kolmogrov Flow is a specific case of NS equation. More specifically, it is described by:

$$\mathbf{u}(x, y, z, t) = \left(-\frac{\partial \psi}{\partial y}, \frac{\partial \psi}{\partial x}, 0\right) \tag{27}$$

where the psi is the flow function. It is usually set as:

Methods		**MSL**			**SMAP**			**SWaT**			**SMD**			**PSM**		**1st Pl**
Metrics	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	**Count**
TimeDiT	**91.54**	87.23	**89.33**	**93.35**	**98.61**	**95.91**	**93.64**	**99.46**	**96.46**	78.83	**88.26**	83.28	97.36	**97.79**	**97.57**	11
GPT(6)	82.00	82.91	82.45	90.60	60.95	72.88	92.20	96.34	94.23	88.89	84.98	**86.89**	98.62	95.68	97.13	1
TimesNet	89.54	75.36	81.84	90.14	56.40	69.39	90.75	95.40	93.02	87.91	81.54	84.61	98.51	96.20	97.34	0
PatchTST	88.34	70.96	78.70	90.64	55.46	68.82	91.10	80.94	85.72	87.26	82.14	84.62	98.84	93.47	96.08	0
ETSformer	85.13	84.93	85.03	92.25	55.75	69.50	90.02	80.36	84.91	87.44	79.23	83.13	**99.31**	85.28	91.76	1
FEDformer	77.14	80.07	78.57	90.47	58.10	70.76	90.17	96.42	93.19	87.95	82.39	85.08	97.31	97.16	97.23	0
LightTS	82.40	75.78	78.95	92.58	55.27	69.21	91.98	94.72	93.33	87.10	78.42	82.53	98.37	95.97	97.15	0
DLinear	84.34	85.42	84.88	92.32	55.41	69.26	80.91	95.30	87.52	83.62	71.52	77.10	98.28	89.26	93.55	0
Autoformer	77.27	80.92	79.05	90.40	58.62	71.12	89.85	95.81	92.74	88.06	82.35	85.11	99.08	88.15	93.29	0
AnoTransformer	79.61	**87.37**	83.31	91.85	58.11	71.18	72.51	97.32	83.10	**88.91**	82.23	85.49	68.35	94.72	79.40	2

Table 11: Anomaly Detection result on 100-length multivariate time series. We calculate Precision, Recall, and F1 score as % for each dataset. '.' notation in model name stands for transformer. **Bold** indicates best result, Underline indicates the second best result.

$$\psi(x,y,z,t) = A\sin(kx)\cos(zy+\omega t) \tag{28}$$

where A, k, w are hyperparameters.

I Proof of the Theorem

Proof. Let us consider the objective function:

$$\begin{aligned} O(q(y|x)) &= \mathbb{E}_{y\sim q(y|x)}K(y) - \alpha D_{KL}(q(y|x)||p(y|x)) \\ &= \mathbb{E}_{y\sim q(y|x)}K(y) - \alpha\int_y q(y|x)\log(\frac{q(y|x)}{p(y|x)})dy \\ &= \int_y q(y|x)[K(y) + \alpha\log p(y|x) - \alpha\log q(y|x)]dy \end{aligned} \tag{29}$$

We try to find the optimal $q(y|x)$ through Lagrange multipliers. The constraint of the above objective function is that $q(y|x)$ is a valid $\int_y q(y|x)dy = 1$. Thus, the Lagrangian is:

$$\begin{aligned} L(q(y|x),\lambda) &= \int_y q(y|x)[K(y) + \alpha\log p(y|x) - \alpha\log q(y|x)]dy - \lambda(\int_y q(y|x)dy - 1) \\ &= \int_y q(y|x)[K(y) + \alpha\log p(y|x) - \alpha\log q(y|x) - \lambda q(y|x)]dy + \lambda \end{aligned} \tag{30}$$

We define $f(q(y|x), y, \lambda) = q(y|x)[K(y) + \alpha\log p(y|x) - \alpha\log q(y|x) - \lambda] + \lambda h(y)]$, where $h(y)$ can be the density function of any fixed distribution defined on the support set of y. Therefore, $L(q(y|x),\lambda) = \int_y f(q(y|x), y, \lambda)dy$. According to Euler-Lagrange equation, when the above Lagrangian achieve extreme point, we have:

$$\frac{\partial f}{\partial q} = K(y) + \alpha\log p(y|x) - \alpha\log q(y|x) - \lambda - \alpha = 0 \tag{31}$$

Thus, we have:

$$\begin{aligned} \alpha\log q(y|x) &= K(y) + \alpha\log p(y|x) - \log q(y|x) - \lambda - \alpha \\ q(y|x) &= \exp(\frac{1}{\alpha}K(y) + \log p(y|x) - \frac{\lambda}{\alpha} - 1) \\ &= \frac{1}{\exp(\frac{\lambda}{\alpha}+1)}\exp(\frac{1}{\alpha}K(y) + \log p(y|x)) \end{aligned} \tag{32}$$

Meanwhile, since $\int_y q(y|x)dy = 1$, we have:

$$\begin{aligned} \int_y \exp(\frac{1}{\alpha}K(y) + \log p(y|x) - \frac{\lambda}{\alpha} - 1)dy = 1 \\ \frac{1}{\exp(\frac{\lambda}{\alpha}+1)}\int_y \exp(\frac{1}{\alpha}K(y) + \log p(y|x))dy = 1 \end{aligned} \tag{33}$$

Thus, we have $\exp(\frac{\lambda}{\alpha}+1) = \int_y \exp(\frac{1}{\alpha}K(y) + \log p(y|x))dy = Z$, leading to:

$$q(y|x) = \frac{1}{Z}\exp(K(y) + \alpha\log p(y|x)), Z = \int \exp(K(y) + \alpha\log p(y|x))dy \tag{34}$$

□