MIXTURE-OF-DIFFUSERS: DUAL-STAGE DIFFUSION MODEL FOR IMPROVED TIME SERIES GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Synthetic Time Series Generation (TSG) is a crucial task for data augmentation and various downstream applications. While TSG has advanced, its effectiveness often relies on the availability of extensive training datasets, posing challenges in data-scarce scenarios. Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) have shown promise, but they frequently struggle to capture the complex temporal dynamics and interdependencies inherent in time series data. To address these limitations, we propose a novel generative framework, Mixture-of-Diffusers (MoD). This approach decomposes the diffusion process into a collection of specialized diffusers, each designed to model specific patterns at distinct noise levels. Early-stage diffusers focus on capturing overarching global and coarse patterns, while late-stage diffusers specialize in capturing fine-grained details as the noise level diminishes. This decomposition empowers MoD to learn robust representations and generate realistic time series samples. The model is trained using a combination of multi-objective loss functions, ensuring both temporal consistency and alignment with the true data distribution. Extensive experiments on a diverse range of real-world and simulated time series datasets demonstrate the superior performance of MoD compared to state-of-the-art TSG generative models. Furthermore, rigorous evaluations incorporating both qualitative and quantitative metrics, coupled with assessments of downstream task performance on long-term generation and scarce time series data (see Figure 1), collectively validate the efficacy of our proposed approach.

1 INTRODUCTION

Synthetic time series generation (TSG) has become a focal point in recent research, driven by the growing demand for synthetic data in diverse applications, including data augmentation, anomaly detection, privacy preservation, and domain adaptation (Nikitin et al. (2024)). The ability to generate realistic time series data is crucial for augmenting machine learning models, especially when real-world data is limited, sensitive, or difficult to collect (Yoon et al. (2019b)). A primary objective of TSG is to create synthetic data that closely resemble real-world time series, preserving essential temporal dependencies and multidimensional correlations. This requires accurately capturing the intricate statistical properties and dynamics inherent in time series data, a challenging task due to their sequential and often stochastic nature (Qiu et al. (2018)).

Moreover, the scarcity of data, particularly in scenarios involving rare or unique events, hinders the training of generative models that rely on extensive datasets to capture the full nuances of the data distribution (Rubanova et al. (2019)). To address these challenges, various methodologies have been explored, leveraging various generative techniques such as GANs (Goodfellow et al. (2014)), VAEs (vae), and Diffusion Models. In early stage, GAN-based approaches have demonstrated proficiency in modeling complex, high-dimensional data distributions and capturing intricate time series characteristics. Models like TimeGAN (Yoon et al. (2019a)) and RCGAN (Esteban et al. (2017)) incorporate recurrent architectures within the GAN framework to effectively capture temporal dependencies. TimeGAN, for instance, combines an autoregressive model with adversarial training to generate realistic time series data that preserve temporal dynamics and feature correlations. However, GANs often suffer from training instability and issues such as mode collapse, limiting sample diversity and the ability to generate high-fidelity data, particularly for long-sequence time series (Ramponi et al. (2019)).

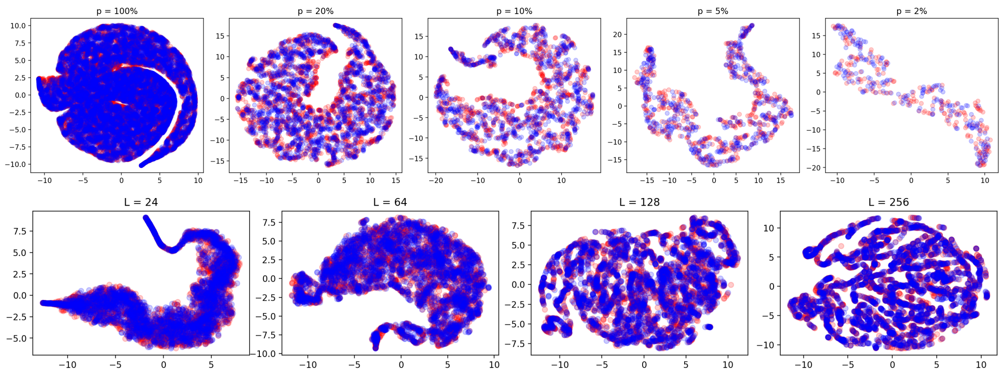


Figure 1: t-SNE visualization comparing synthetic data generated by our model (blue) to the original data (red). The upper panel demonstrates the model's performance across varying proportions (p) of the Sines dataset, ranging from 100% to 2%. The lower panel evaluates its effectiveness with different sequence lengths (L) on the Energy dataset, from 24 to 256.

Later, VAEs have emerged as a leading technique in TSG due to their capacity to balance data fidelity with latent space statistical consistency. They encode input data into a latent space and then decode it back to reconstruct the original data, enabling the generation of new samples by sampling from the latent space. The Variational Recurrent Autoencoder (VRAE) (Fabius & van Amersfoort (2015)) extends the VAE framework to sequential data by incorporating recurrent neural networks into the encoder and decoder. However, VAEs typically strive for independent mapping between latent features and external conditions (vae). Nevertheless, these conditions often exhibit inter-correlations; changing one condition might unintentionally influence others, complicating the capture of accurate relationships among external conditions (Li et al. (2023)). Furthermore, VAEs are often challenged by the need to model complex temporal dynamics and may not efficiently handle the complexities of high-dimensional, long-sequence data (Alaa et al. (2022)).

Recently, in light of these challenges, diffusion-based models have emerged as a promising alternative (Dhariwal & Nichol (2021)). Originating from advances in computer vision and natural language processing, diffusion models involve a forward diffusion process where noise is incrementally added to the data, followed by a reverse process where a neural network is trained to reconstruct the data from the noisy input (Ho et al. (2020)). This framework effectively addresses core challenges faced by GANs and VAEs, such as training instability and mode collapse, by providing a stable learning process and effectively capturing the underlying data distribution (Lee et al. (2023)). Diffusion models have garnered significant attention due to their stable training and ability to model complex distributions. In the context of TSG, they offer several advantages, including the capability to model complex temporal dynamics and handle high-dimensional data with long sequences and variable lengths (Zhou et al. (2023a)). Approaches such as DiffWave (Kong et al. (2021)), Diffusion-TS (Yuan & Qiao (2024)), and TimeDiff (Shen & Kwok (2023)) have adopted the diffusion framework for modeling the data generation process. However, their application to long-term and scarce time series generation, remains an area that warrants further investigation.

To address these limitations, we investigate the application of diffusion models for TSG. Specifically, we propose a Mixture of Diffuser approach designed to learn the underlying representations and data distribution of multivariate time series data through a diffusion process, utilizing the trained denoiser model to generate new data samples that closely resemble the original data. Importantly, we segment the diffusion process into dual stages and employ a Transformer-based model at each stage to capture the dependencies and patterns corresponding to each stage. This approach enables each diffuser to concentrate on different aspects of the data that vary throughout the diffusion process. By doing so, the early-stage diffuser becomes specialized in capturing coarse-grained patterns in high-noise regimes, while the late-stage diffuser focuses on learning fine-grained patterns as the noise level decreases. The combination of representations learned by these two specialized diffusers empowers the model to learn the intricate dependencies and temporal dynamics inherent in time series data, facilitating the generation of highly realistic samples that closely resemble the original time series data. Our key contributions can be summarized as follows:

- We introduce a diffusion-based generative framework that partitions the diffusion process into two expert diffusers, each specialized in handling distinct noise levels, enabling the model to capture a wide range of temporal dynamics present in time series data.
- Our model segments the diffusion into two stages, allowing it to effectively discern overarching, coarse-scale patterns in the initial stage and intricate, fine-grained details in late stage, while facilitating a smooth transition between both. This stratified approach empowers the model to learn robust representations.
- To ensure the generation of both coherent and representative samples, we employ a joint training regimen that integrates different loss functions, concurrently enforcing temporal consistency and data distribution similarity.
- To underscore the model's robustness and generalizability, we evaluate its performance on challenging time series data characterized by extended sequences and limited availability.

2 Problem Statement

This study introduces a diffusion-based generative approach designed to learn the underlying distribution of multivariate time series data, thereby producing synthetic samples that closely resemble real data. Let $\{S_k \in \mathbb{R}^C\}_{k=1}^K$ represent a multivariate time series with, C, variables over, K, time steps. The time series data is segmented into sequences of length L to form the input $x \in \mathbb{R}^{L \times C}$. Given the data x, our objective is to train a diffusion model capable of generating samples that mimic the patterns observed in the original data. To achieve this, in the forward diffusion process, Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is gradually added over T diffusion steps. A neural network $\epsilon_\theta(x)$ is then used to predict the noise at each diffusion step. By training $\epsilon_\theta(x)$, we aim to approximate the true data distribution $q(x)$, enabling the generation of high-quality synthetic time series data.

3 Method

As depicted in Figure 2, the proposed framework utilizes a Mixture-of-Diffusers (MoD) architecture to enhance the model's capacity to capture diverse data patterns across different noise levels. The MoD comprises two transformer-based expert diffusers specifically designed to handle high and low noise levels, respectively. To incorporate valuable temporal context, the model leverages advanced positional encodings in conjunction with a Transformer encoder and Conv1D. The model is trained to minimize a loss function that combines MSE for noise prediction and KL divergence for posterior matching. The following sections provide a comprehensive analysis of each component.

3.1 Diffusion Processes

Denoising Diffusion Probabilistic Models (DDPMs) are generative models that employ a two-stage process for data generation and reconstruction (Ho et al. (2020)). In the forward diffusion process, noise is gradually added to the input data, x_0, following a predefined noise schedule over T diffusion steps. At each step, $t \in [1, T]$, the diffused sample, x_t, is obtained by scaling the previous sample, x_{t-1}, with $\sqrt{1-\beta_t}$ and adding independent and identically distributed noise. This process can be mathematically represented as:

$$q(x_1, x_2, ..., x_T | x_0) = \prod_{t=1}^{T} q(x_t | x_{t-1}); \tag{1}$$

$$q(x_t | x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}\, x_{t-1}, \beta_t I) \tag{2}$$

where $\beta_t \in [0, 1]$ is the noise variance at step t. Based on Ho et al. (2020), we can leverage a forward diffusion process in Equation 2 to sample noisy data directly conditioned on the input x_0, where $\alpha_t := 1 - \beta_t$, $\bar{\alpha}_t := \prod_{i=1}^{t} \alpha_i$, and $\epsilon \sim \mathcal{N}(0, I)$:

$$q(x_t | x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}\, x_0, (1 - \bar{\alpha}_t) I) \tag{3}$$

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon \tag{4}$$

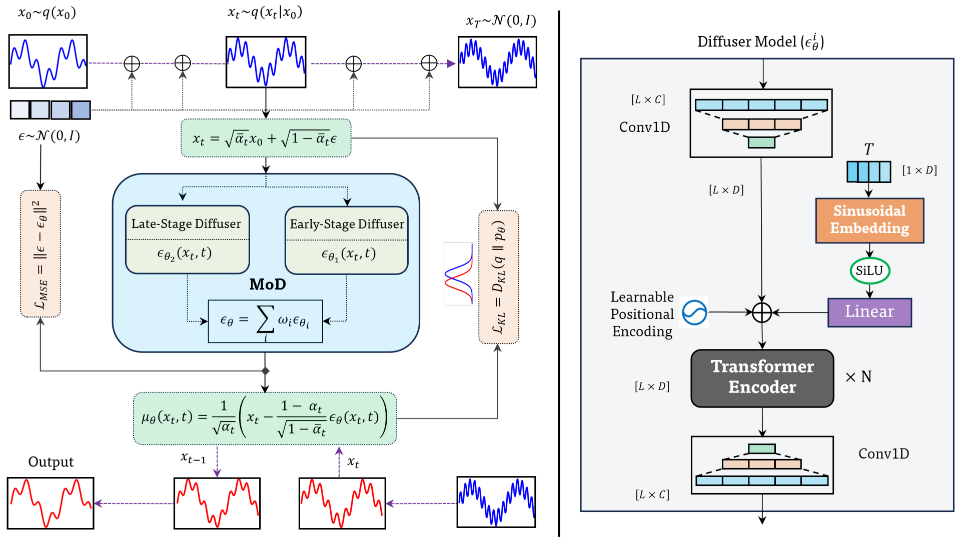


Figure 2: Schematic architecture of MoD. The left panel illustrates the core MoD component, along with the forward and reverse diffusion processes, where noise is progressively added and removed over (T) timesteps. The right panel provides a detailed view of the Diffuser model's architecture.

By applying Bayes' theorem, we can derive the posterior distribution $q(x_{t-1}|x_t, x_0)$ in terms of its mean $\tilde{\mu}_t(x_t, x_0)$ an variance $\tilde{\beta}_t$:

$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t; \quad \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \tag{5}$$

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \tilde{\beta}_t I) \tag{6}$$

For sufficiently large T and a well-designed β_t, x_t approaches an isotropic Gaussian distribution. While $q(x_{t-1}|x_t)$ depends on the entire data distribution, we approximate it in the reverse diffusion process using our proposed MoD as follows:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t); \sigma_t^2 I) \tag{7}$$

We select to parametrize $\mu_\theta(x_t, t)$ in the prior by directly predicting the noise component ϵ_θ in Equation 10 using the MoD, leveraging Equations 4 and 5 to derive:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) \tag{8}$$

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right) + \sigma_t z \tag{9}$$

At each timestep, the MoD model predicts the noise component, and x_{t-1} is computed using Equation 9, where $z \sim \mathcal{N}(0, I)$ when $t > 1$ and 0 otherwise. The combination of two experts, guided by the weighting functions, enables the model to adaptively address varying noise levels during the denoising process.

3.2 Mixture of Diffusers (MoD)

Modeling the complex patterns and dependencies within multivariate time series data presents a significant challenge. While a single model may struggle to capture this full spectrum, the Mixture-of-Experts (MoE) framework offers a solution by allowing specialized models to handle specific data aspect, thereby augmenting the overall modeling capacity.

Time series data can be conceptualized as a signal comprised of information at multiple frequency scales. Low-frequency components, such as long-term trends, represent coarse-grained features that

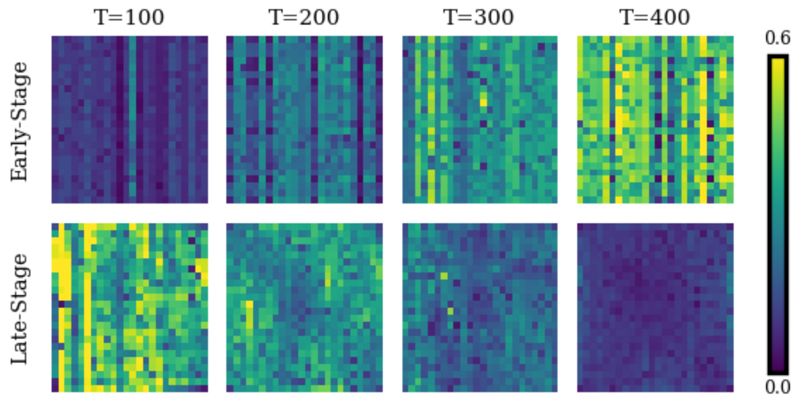


Figure 3: Attention weights heatmaps for encoder layer in both diffusers on ETTh dataset.

provide a foundational understanding of the data's overall behavior. High-frequency components, on the other hand, capture fine-grained features like short-term patterns and fluctuations. In the context of time series diffusion, the forward diffusion process involves adding noise to the data, gradually disrupting its structure until it becomes indistinguishable from pure noise. This process increases the randomness in the data and affects both high-frequency and low-frequency components, making it more challenging to discern the original patterns. The reverse diffusion process, or denoising, can be viewed as dual-stage reconstruction process. In the early stages of the reverse diffusion process, characterized by high noise levels, fine-grained details are obscured. To establish a robust foundation for reconstruction, the model focuses on reconstructing the most prominent features that stand out despite the noise—often the coarse-grained, global structures and long-term trends. As the noise level decreases, finer details of the original data emerge. The model gradually refines the reconstruction by incorporating high-frequency details, such as short-term fluctuations.

Building upon this concept, our proposed MoD framework utilizes two specialized expert Diffusers, each learns specific underlying patterns within its designated domain of expertise under specific noise regime. This segmentation of the diffusion process into two distinct stages aligns perfectly with the idea that different patterns and characteristics within time series data become more prominent at various stages of the diffusion process, particularly as the noise level fluctuates:

- **Early-Stage Diffuser** (ϵ_{θ_1}): Specializes in initial diffusion stages characterized by elevated noise levels. It concentrates on capturing the overarching global structures, long-term trends, and coarse patterns that are more conspicuous when noise dominates.
- **Late-Stage Diffuser** (ϵ_{θ_2}): Specializes in the latter diffusion stages characterized by diminished noise levels, capturing fine-grained details, short-term patterns, and subtle variations that become more apparent as the noise subsides.

Figures 3 and 5 illustrate heatmaps of the attention weights associated with last Transformer encoder layer for each diffuser at different steps (T=500). As depicted, the early-stage diffuser exhibits higher attention weights in the early stages (T=400), focusing on long-term patterns, while the late-stage diffuser dominates in the later stages (T=100), capturing short-term details. A smooth transition between the two diffusers is evident at intermediate steps (200 and 300). Furthermore, we conduct experiments to assess the potential benefits of employing more than two diffusers (Table 5). However, our findings indicate that increasing the number of diffusers to three or four doesn't not yield improvements in model performance. Instead, it results in increased complexity and longer training and inference times. A more detailed discussion on the rationale behind using dual-stage diffusers and their effectiveness is provided in Appendix C.

To seamlessly integrate the contributions of both diffusers, a time-dependent weighting scheme is employed. This scheme dynamically adjusts the influence of each expert based on the current diffusion timestep t. The weight for the early-stage Diffuser,w_1, is defined as a function of time: $w_1 = t/T$, where T is the total number of diffusion steps. Conversely, the weight for the late-stage

Diffuser is $w_2 = 1 - w_1$. This weighting mechanism ensures a smooth and intuitive transition between the two diffusers. During the initial diffusion steps ($t \approx T$) where noise levels are high, $w_1 \approx 1$, giving greater influence to the early-stage diffuser to effectively capture the coarse-grained features, allowing it to primarily guide the denoising process. As the diffusion process advances, t approaches 0 and the noise level decreases, the weight shifts the influence towards the late-stage diffuser with $w_2 \approx 1$. This transition allows the late-stage diffuser to take precedence in refining the generated samples by incorporating fine-grained details. The predicted noise is then calculated by combining the outputs of both expert diffusers, weighted according to the current timestep:

$$\epsilon_\theta(x_t, t) = w_1 \cdot \epsilon_{\theta_1}(x_t, t) + w_2 \cdot \epsilon_{\theta_2}(x_t, t) \tag{10}$$

3.3 Diffuser Model

Each Diffuser model employs a Transformer-based architecture tailored for multivariate time series modeling, as depicted in the right panel of Figure 2. This architecture integrates convolutional layers, transformer encoders, and advanced positional encodings to ensure efficient noise removal and high-fidelity reconstruction of temporal sequences. The noisy data x_t is initially processed by a 1D Convolutional Layer (Conv1D) that extracts localized features by learning temporal dependencies within a restricted receptive field. This local processing is essential for capturing short-range patterns while maintaining robustness to noise in the input sequence. A learnable positional encoding is then applied to preserve the temporal order and periodic characteristics of the input data. Concurrently, a sinusoidal embedding of the diffusion time step is activated using the Sigmoid Linear Unit (SiLU) and undergoes a linear transformation to provide the model with information about the prevailing noise level in the diffusion process. The transformed features from the convolutional layers, positional encodings, and diffusion time step embeddings are combined and fed into a Transformer encoder, composed of multiple stacked layers. By employing multi-head self-attention mechanisms, the Transformer can dynamically assign different weights to various parts of the sequence, capturing long-range temporal dependencies that are crucial for reconstructing coherent and realistic time series data from noisy inputs. After processing through the Transformer encoder, the features are passed through an additional Conv1D layer that maps the output back to the original feature dimension. This layer refines the output by focusing on local structures within the sequence, ensuring that the output not only aligns globally but also exhibits fine-grained detail at the local level.

3.4 Training and Sampling

Our proposed model's training objective, as formalized in Equation 11, incorporates two loss components to ensure precise noise prediction and alignment with the underlying data distribution.

$$\mathcal{L} = \mathbb{E}_{t,x_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2\right] + \lambda_{kl} D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)) \tag{11}$$

here, λ_{kl} is hyperparameter balancing the contribution of the KL divergence loss.

The first term, MSE loss, quantifies the difference between ϵ, the actual noise introduced during forward diffusion, and ϵ_θ, the noise predicted by the model. Based on (Kingma & Welling (2022)), the combination of q and p forms a variational autoencoder with a variational lower bound, which is a summation of KL divergences over $t \in [0, T]$. However, at $t = T$, the KL divergence $D_{KL}\left(q(x_T|x_0)||p(x_T)\right)$ becomes independent of θ and approaches zero if the forward diffusion process effectively destroys the data distribution, such that $q(x_T|x_0) \approx \mathcal{N}(0, I)$ (Nichol & Dhariwal (2021)). Additionally, at $t = 0$, the KL divergence reduces to the negative log-likelihood of $p_\theta(x_0|x_1)$, which we compute using the CDF. Consequently, the second term of our loss simplifies to $\Sigma_{T-1}^{1} D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))$. To compute each individual term, the Equation 4 provides an efficient method to sample from arbitrary steps of the forward diffusion process and estimate the KL divergence using the posterior distribution (Equation 6) and prior distribution (Equation 7).By randomly sampling t and calculating the expectation $\mathbb{E}_{t,x_0,\epsilon}\left[D_{KL}(q(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t))\right]$, we can approximate $D_{KL}\left(q||p_\theta\right)$. Incorporating KL loss encourages the model to generate samples whose distribution more closely approximates the true posterior distribution, thereby enhancing the accuracy of reconstructions. To empirically validate this, we present visualizations of t-SNE, KDE, and PCA plots for both the original and generated data in two experimental settings: one with both KL and MSE, and another without the KL term. The comparative results are illustrated in Figures 13 and 14 of the Appendix. Detailed training and inference procedures are outlined in Algorithms 1 and 2, respectively, in Appendix C.2.

4 Experiments

4.1 Experimental Setup

Datasets. We compiled a diverse collection of benchmark time series datasets, encompassing both real-world and simulated data. Real-world datasets included financial data (Stocks, Stockv), mechanical system data (ETTh), electricity consumption data (Energy), and air quality data. Simulated datasets included synthetic sine waves and MuJoCo physics simulations. Detailed statistics of these datasets are provided in Table 6 and Appendix D.1.

Baselines. We conduct comparative experiments against established time series generative methods, including: KoVAE (Naiman et al. (2024b)), Diffusion-TS (Yuan & Qiao (2024)), TimeGAN (Yoon et al. (2019a)), TimeVAE (Desai et al. (2021)), Diffwave (Kong et al. (2021)), DiffTime (Coletta et al. (2023)), Cot-GAN (Xu et al. (2020)), T-Forcing (Sutskever et al. (2011)), which is RNNs trained with teacher-forcing, and RCGAN (Esteban et al. (2017). Source code links for these methods are provided in Table 7.

Evaluation Metrics. To evaluate the quality of generated synthetic time series data, we employ a suite of metrics, including:: Discriminative Score (Yoon et al. (2019a)). Predictive Score(Yoon et al. (2019a)). Context-Frechet Inception Distance (Context-FID) (Jeha et al. (2022)). Correlational Score (Ni et al. (2021)). For a more detailed discussion of these metrics, please refer to Appendix D.3.

4.2 Experimental Results

In this section, we evaluate the efficacy of our proposed model across three primary dimensions: (1) *Representation Analysis*: We employ Kernel Density Estimation, Principal Component Analysis (PCA), and t-Distributed Stochastic Neighbor Embedding (t-SNE) van der Maaten & Hinton (2008) to visualize the learned representations of both the original and synthetic data. (2) *Sampling Quality and Quantity*: We compare the performance of MoD against several baselines across five diverse datasets using the four established metrics. (3) *Downstream Task Performance*: We evaluate the performance of our model and other baselines in the contexts of long-sequence and scarce data generation.

4.2.1 Comparison of Generated and Original Data Representations

To evaluate the effectiveness of our proposed MoD model in capturing underlying data patterns and generating realistic synthetic samples, we visualize the learned representations of both original and generated data distributions across the Energy and ETTh datasets using t-SNE. Additionally, KDE is employed to provide a more granular analysis of their distributional similarity.

Figure 4 demonstrates that our model exhibits a notable overlap in the KDE plot between the original and generated data distributions, suggesting its proficiency in learning the underlying distribution. Furthermore, the t-SNE plot reveals that the generated data from our model closely resembles the original data, indicating its effectiveness in capturing the underlying data patterns and generating high-quality samples. In contrast, the visualizations for Diffusion-TS and TimeGAN show a less pronounced overlap, suggesting potential limitations in their ability to accurately represent the original data distribution. For additional visualizations, please refer to Appendix E.

4.2.2 Time Series Generation Results

We evaluate the performance of MoD across a rang of data complexities with varying in dimensionality. We employ four metrics to comprehensively assess data quality, with the results summarized in Table 1. For each dataset, we utilize sequences of 24 time steps, aligning with common practices in existing research (Yoon et al. (2019a)). Notably, our model consistently outperforms baseline methods, achieving notable improvements across most metrics. On the Energy dataset, MoD achieves a discriminative score reduction of 23%, 34%, and 60% compared to Diffusion-TS, KoVAE, and TimeGAN, respectively. Additionally, MoD attains predictive scores close to *original* on all datasets, indicating the high efficacy of its generated data. Furthermore, MoD demonstrates superior preservation of temporal dependencies, with correlational score reduction of 36%, 40%,

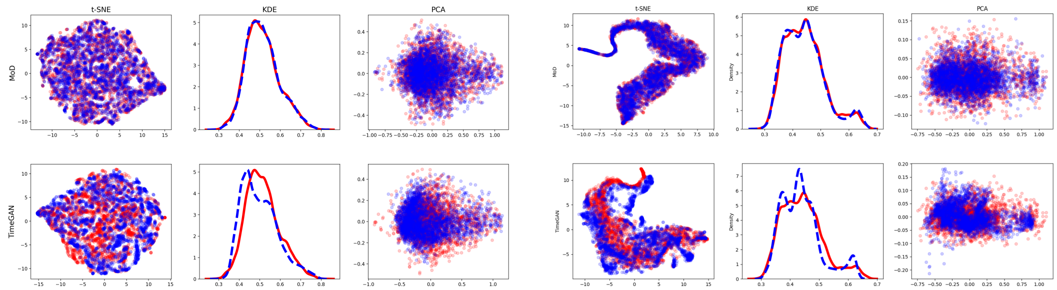


Figure 4: t-SNE, KDE, and PCA distributions visualization for the original (red) and synthetic (blue) data generated by MoD (up) and TimeGAN (down) on the ETTh (left) and Energy (right).

Table 1: Performance on multiple time series datasets. *Original* values means that predictor is trained on original data instead of synthetic data generated by various models. **Boldface** values indicate the best results, while underlined values denote the second-best performance.

Metric	Methods	Energy	ETTh	Stocks	Sines	MuJoCo
Context-FID Score (Lower the Better)	MoD	**0.035±.005**	**0.015±.001**	**0.027±.007**	0.017±.001	**0.013±.000**
	KoVAE	0.078±.010	0.120±.009	0.095±.013	0.015±.002	0.024±.009
	Diffusion-TS	0.089±.024	0.116±.010	0.147±.025	**0.006±.000**	0.013±.001
	TimeGAN	0.767±.103	0.300±.013	0.103±.013	0.101±.014	0.563±.052
	TimeVAE	1.631±.142	0.805±.186	0.215±.035	0.307±.060	0.251±.015
	Diffwave	1.031±.131	0.873±.061	0.232±.032	0.014±.002	0.393±.041
	DiffTime	0.279±.045	0.299±.044	0.236±.074	0.006±.001	0.188±.028
	Cot-GAN	1.039±.028	0.980±.071	0.408±.086	1.337±.068	1.094±.079
Correlational Score (Lower the Better)	MoD	**0.656±.060**	**0.017±.003**	0.010±.003	**0.013±.002**	**0.122±.006**
	KoVAE	0.862±.101	0.045±.006	0.007±.002	0.019±.008	0.203±.031
	Diffusion-TS	0.856±.147	0.049±.008	**0.004±.001**	0.015±.004	0.193±.027
	TimeGAN	4.010±.104	0.210±.006	0.063±.005	0.045±.010	0.886±.039
	TimeVAE	1.688±.226	0.111±.020	0.095±.008	0.131±.010	0.388±.041
	Diffwave	5.001±.154	0.175±.006	0.030±.020	0.022±.005	0.579±.018
	DiffTime	1.158±.095	0.067±.005	0.006±.002	0.017±.004	0.218±.031
	Cot-GAN	3.164±.061	0.249±.009	0.087±.004	0.049±.010	1.042±.007
Discriminative Score (Lower the Better)	MoD	**0.093±.006**	**0.011±.004**	0.025±.019	**0.004±.003**	0.012±.009
	KoVAE	0.142±.005	0.066±.007	**0.009±.011**	0.005±.006	0.075±.003
	Diffusion-TS	0.122±.003	0.061±.009	0.067±.015	0.006±.007	**0.008±.002**
	TimeGAN	0.236±.012	0.114±.055	0.102±.021	0.011±.008	0.238±.068
	TimeVAE	0.499±.000	0.209±.058	0.145±.120	0.041±.044	0.230±.102
	Diffwave	0.493±.004	0.190±.008	0.232±.061	0.017±.008	0.203±.096
	DiffTime	0.445±.004	0.100±.007	0.097±.016	0.013±.006	0.154±.045
	Cot-GAN	0.498±.002	0.325±.099	0.230±.016	0.254±.137	0.426±.022
Predictive Score (Lower the Better)	MoD	**0.250±.000**	**0.119±.001**	**0.036±.000**	**0.093±.000**	0.008±.001
	KoVAE	**0.250±.000**	0.121±.002	0.038±.000	**0.093±.000**	0.039±.001
	Diffusion-TS	**0.250±.000**	0.119±.003	**0.036±.000**	**0.093±.000**	**0.007±.000**
	TimeGAN	0.273±.004	0.124±.001	0.038±.001	0.093±.019	0.025±.003
	TimeVAE	0.292±.000	0.126±.004	0.039±.000	**0.093±.000**	0.012±.002
	Diffwave	0.251±.000	0.130±.001	0.047±.000	**0.093±.000**	0.013±.000
	DiffTime	0.252±.000	0.121±.004	0.038±.001	**0.093±.000**	0.010±.001
	Cot-GAN	0.259±.000	0.129±.000	0.047±.001	0.100±.000	0.068±.009
	Original	0.250±.003	0.121±.005	0.036±.001	0.094±.001	0.007±.001

and 80% to Diffusion-TS, KoVAE, and TimeGAN on MuJoCo. To complement the quantitative analysis, Figure 4 and 1 provide qualitative evaluations using t-SNE, PCA, and KDE visualizations.

The remarkable performance of MoD can be attributed to several key factors. (1) Specialization across diffusion stages through two expert diffusers, each focusing on different diffusion stages. This specialization allows each diffuser to concentrate on specific patterns associated with their respective stages. (2) Enhanced learning capacity: By specializing, MoD improves its overall learning capacity, effectively modeling both global and local patterns within the multivariate time series data. (3) Adaptive contribution of experts: The time-dependent weighting function in MoD dynamically adjusts the influence of each expert, facilitating a smooth transition between them and enhancing the model's flexibility to capture a wide range of patterns over time. (4) Joint training: The combination of MSE loss for temporal relation preservation and KL divergence loss for improved distribution similarity enables MoD to learn a diverse set of representations.

Table 2: Performance on scarce data scenarios. *Size* represents the amount of training and generated data. **Boldface** values indicate the best results. (refer to Table 11 for complete results.)

Size	Metric	Methods	Air	Energy	Sines	Stockv
20%	Discriminative Score (Lower the Better)	MoD	**0.159±.027**	**0.061±.004**	**0.008±.022**	**0.016±.033**
		Diffusion-TS	0.186±.012	0.379±.014	.024±.012	0.118±.011
		TimeVAE	0.350±.089	0.499±.002	0.039±.030	0.176±.208
		TimeGAN	0.355±.045	0.493±.007	0.374±.102	0.042±.068
		T-Forcing	0.500±.000	0.500±.001	0.490±.003	0.372±.241
		RCGAN	0.500±.000	0.500±.000	0.281±.132	0.479±.028
	Predictive Score (Lower the Better)	MoD	**0.006±.000**	**0.195±.000**	**0.092±.001**	0.025±.001
		Diffusion-TS	0.026±.014	0.251±.000	0.093±.000	**0.024±.000**
		TimeVAE	0.019±.003	0.288±.002	0.215±.000	0.052±.001
		TimeGAN	0.007±.002	0.324±.005	0.287±.051	0.050±.001
		T-Forcing	0.139±.061	0.256±.006	0.219±.007	0.091±.024
		RCGAN	0.480±.315	0.751±.434	0.254±.001	0.164±.122
10%	Discriminative Score (Lower the Better)	MoD	**0.101±.023**	**0.056±.019**	**0.009±.017**	**0.053±.029**
		Diffusion-TS	0.187±.025	0.340±.020	0.012±.005	0.104±.017
		TimeVAE	0.425±.067	0.499±.001	0.053±.045	0.080±.108
		TimeGAN	0.257±.093	0.500±.001	0.382±.055	0.068±.106
		T-Forcing	0.500±.000	0.500±.001	0.482±.006	0.474±.071
		RCGAN	0.500±.000	0.500±.000	0.246±.234	0.474±.071
	Predictive Score (Lower the Better)	MoD	**0.003±.000**	**0.188±.000**	**0.092±.000**	**0.026±.000**
		Diffusion-TS	0.013±.012	0.252±.000	0.092±.000	0.027±.000
		TimeVAE	0.005±.003	0.275±.001	0.215±.000	0.075±.001
		TimeGAN	0.003±.001	0.318±.006	0.300±.059	0.081±.008
		T-Forcing	0.157±.027	0.262±.014	0.216±.002	0.118±.040
		RCGAN	0.605±.618	0.740±.371	0.241±.030	0.150±.094

4.2.3 Downstream Tasks

Data Scarcity. To evaluate the model's performance under varying data scarcity conditions, we conduct experiments using the experimental settings outlined in (Desai et al. (2021)). For each dataset, we employ training sets comprising 100%, 20%, 10%, 5%, and 2% of the original data. We then generate synthetic data for various models on four datasets. The quantity of generated data corresponded to the percentage of original training data used to train the generators. Table 2 presents the discriminative and predictive scores for the 20%, and 10% cases (complete results are provided in Table 11 in Appendix E.2). Our model consistently outperforms the baselines across all scarcity levels and datasets. Notably, the lower scores achieved by our model indicate the high quality of the generated data even in challenging scenarios with very limited data (i.e., 2%). Moreover, the predictive scores of our model are nearly indistinguishable from those of the original datasets for most scarcity levels. Importantly, our model demonstrates superior performance on the high-dimensional Energy dataset. To further visualize the quality of the generated data across different scarcity levels, we plot t-SNE, KDE, and PCA visualizations of the generated data of our model in Figures 9, 10, 11, and 12 in Appendix E.2.

Long-term Generation. To evaluate the model's ability to generate long time series sequences with high-quality, we conduct experiments on the Energy and ETTh datasets using sequence lengths of 64, 128, and 256. The results are presented in Table 3 (The extended results are reported in Table 9). Our findings demonstrate the model's capacity to capture underlying patterns and generate realistic samples, even when dealing with long sequences, a challenging aspect of time series data. The model consistently achieves the best scores across most datasets with notable improvements on ETTh dataset. For instance, our model improves the discriminative score by 84% compared to Diffusion-TS and by 92% compared to TimeGAN. Additionally, we plot visualizations of our model performance compared to Diffusion-TS in Figures 6 and 7 in Appendix E.1.

4.3 Ablation Study

To evaluate the contributions of each component within MoD, we conduct an ablation study by removing individual components to assess their impact on performance. We compare the performance of MoD against four key variants: (1) *Base Diffuser*:We replace the mixture of diffusers with a single diffuser throughout the entire diffusion process. (2) *no Time Adaption*: We remove the time-adaptive weighting function that adjusts the contribution of each diffuser based on the diffusion step. (3) w/o

Table 3: Long-term generation performance comparison. Extended results are provided in Table 9

Dataset	Metrics	Length	MoD	Diffusion-TS	TimeGAN	TimeVAE	Diffwave	DiffTime	Cot-GAN
ETTh	Context-FID	64	**0.034±.002**	0.631±.058	1.130±.102	0.827±.146	1.543±.153	1.279±.083	3.008±.277
		128	**0.065±.002**	0.787±.062	1.553±.169	1.062±.134	2.354±.170	2.554±.318	2.639±.427
		256	**0.112±.008**	0.423±.038	5.872±.208	0.826±.093	2.899±.289	3.524±.830	4.075±.894
	Correlational	64	**0.029±.007**	0.082±.005	0.483±.019	0.067±.006	0.186±.008	0.094±.010	0.271±.007
		128	**0.032±.011**	0.088±.005	0.188±.006	0.054±.007	0.203±.006	0.113±.012	0.176±.006
		256	**0.027±.007**	0.064±.007	0.522±.013	0.046±.007	0.199±.003	0.135±.006	0.222±.010
Energy	Context-FID	64	**0.041±.004**	0.135±.017	1.230±.070	2.662±.087	2.697±.418	0.762±.157	1.824±.144
		128	**0.043±.003**	0.087±.019	2.535±.372	3.125±.106	5.552±.528	1.344±.131	1.822±.271
		256	**0.058±.003**	0.126±.024	5.032±.831	3.768±.998	5.572±.584	4.735±.729	2.533±.467
	Correlational	64	**0.411±.033**	0.672±.035	3.668±.106	1.653±.208	6.847±.083	1.281±.218	3.319±.062
		128	**0.243±.026**	0.451±.079	4.790±.116	1.820±.329	6.663±.112	1.376±.201	3.713±.055
		256	**0.247±.052**	0.361±.092	4.487±.214	1.279±.114	5.690±.102	1.800±.138	3.739±.089

Table 4: Ablation study on MoD and variants across multiple datasets.

Metric	Methods	ETTh	Energy	Sines	Stocks
Discriminative Score (Lower the Better)	MoD	0.011±.004	0.093±.006	0.007±.004	0.005±.019
	Base Diffuser	0.499±.000	0.149±.004	0.045±.009	0.169±.008
	no Time Adaption	0.012±.003	0.101±.008	0.011±.003	0.012±.001
	w/o KL	0.013±.004	0.103±.004	0.019±.008	0.045±.041
	w/o MSE	0.497±.000	0.5±.000	0.499±.000	0.5±.000
Correlational Score (Lower the Better)	MoD	0.017±.003	0.656±.060	0.013±.002	0.010±.003
	Base Diffuser	0.039±.002	0.875±.003	0.028±.001	0.109±.002
	no Time Adaption	0.019±.002	0.661±.085	0.017±.003	0.015±.004
	w/o KL	0.019±.003	0.682±.024	0.014±.002	0.012±.002
	w/o MSE	1.029±.002	6.098±.026	0.567±.001	1.25±.000

KL: This variant removes the KL divergence loss from the total loss function. (4) w/o MSE: We omit the MSE loss. The results of this ablation study are presented in Table 4.

The ablation study reveales that removing the MSE loss significantly affects the model's performance across all datasets, demonstrating its critical role in preserving temporal dependencies. The Single Diffuser model exhibits inferior performance across all metrics, highlighting the importance of the dual-stage diffusers in enhancing the model's ability to capture more robust and diverse representations, leading to more realistic generated samples. Removing the KL divergence loss term results in a slight decrease in performance, particularly in terms of discriminative score, indicating its role in enforcing similarity between the true and generated data distributions. By removing the time-dependent weighting function, we observe a decrease in performance across all metrics, suggesting the effectiveness of dynamically adapting the influence of each expert diffuser to capture different patterns at different stages of the diffusion process.

5 CONCLUSION

In this study, we have proposed MoD, a diffusion-based generative model for time series synthesis. By decomposing the diffusion process into specialized expert networks, MoD effectively captures the intricate temporal dynamics and interdependencies inherent in time series data. The segmentation of the diffusion process into two satges, coupled with the joint training regimen, ensures that the generated samples are both coherent and representative of the underlying data distribution. Our empirical evaluation demonstrates MoD's superior performance in both representation learning and time series generation, surpassing baseline models in preserving temporal consistency, capturing data distribution, and facilitating downstream tasks. In addition to its superior performance, MoD offers a computational advantage by employing two specialized expert networks. This approach allows for the use of relatively smaller models, reducing inference time compared to a single large model. By focusing on specific aspects of the data, each expert can capture patterns more efficiently, leading to improved computational efficiency without compromising the quality of the generated samples. For a discussion of limitations and future research directions, please refer to Appendix B.

6 Ethics Statement

Our research adheres to ethical guidelines for fairness and transparency. No personal data was used, and all datasets are either synthetic or publicly available without privacy concerns.

7 Reproducibility Statement

Detailed documentation of the experimental setup, including hyperparameters and model configurations, can be found in the supplementary materials submitted. We have also provided scripts for generating synthetic datasets to ensure that all experiments are fully reproducible.

REFERENCES

Ahmed Alaa, Alex James Chan, and Mihaela van der Schaar. Generative time-series modeling with fourier flows. In *International Conference on Learning Representations*, 2021.

Ahmed Alaa, Boris Van Breugel, Evgeny S. Saveliev, and Mihaela van der Schaar. How faithful is your synthetic data? Sample-level metrics for evaluating and auditing generative models. volume 162 of *Proceedings of Machine Learning Research*, pp. 290–306. PMLR, 17–23 Jul 2022.

Marin Biloš, Kashif Rasul, Anderson Schneider, Yuriy Nevmyvaka, and Stephan Günnemann. Modeling temporal data as continuous functions with stochastic process diffusion, 2023. URL https://arxiv.org/abs/2211.02590.

Defu Cao, Wen Ye, and Yan Liu. Timedit: General-purpose diffusion transformers for time series foundation model. In *ICML 2024 Workshop on Foundation Models in the Wild*, 2024. URL https://openreview.net/forum?id=DA36Myd4HD.

Yu Chen, Wei Deng, Shikai Fang, Fengpei Li, Nicole Tianjiao Yang, Yikai Zhang, Kashif Rasul, Shandian Zhe, Anderson Schneider, and Yuriy Nevmyvaka. Provably convergent schrödinger bridge with applications to probabilistic time series imputation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 4485–4513. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/chen23f.html.

Andrea Coletta, Sriram Gopalakrishnan, Daniel Borrajo, and Svitlana Vyetrenko. On the constrained time-series generation problem. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 61048–61059. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/bfb6a69c0d9e2bc596e1cd31f16fcdde-Paper-Conference.pdf.

Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver. Timevae: A variational auto-encoder for multivariate time series generation, 2021. URL https://arxiv.org/abs/2111.08095.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. URL https://arxiv.org/abs/2105.05233.

Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. Real-valued (medical) time series generation with recurrent conditional gans, 2017. URL https://arxiv.org/abs/1706.02633.

Otto Fabius and Joost R. van Amersfoort. Variational recurrent auto-encoders, 2015. URL https://arxiv.org/abs/1412.6581.

Asadullah Hill Galib, Pang-Ning Tan, and Lifeng Luo. FIDE: Frequency-inflated conditional diffusion model for extreme-aware time series generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=5HQhYiGnYb.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

Daniel Jarrett, Ioana Bica, and Mihaela van der Schaar. Time-series generation by contrastive imitation. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=RHZs3GqLBwg.

Paul Jeha, Michael Bohlke-Schneider, Pedro Mercado, Shubham Kapoor, Rajbir Singh Nirwan, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. PSA-GAN: Progressive self attention GANs for synthetic time series. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Ix_mh42xq5w.

Jinsung Jeon, JEONGHAK KIM, Haryong Song, Seunghyeon Cho, and Noseong Park. Gt-gan: General purpose time series synthesis with generative adversarial networks. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 36999–37010. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/f03ce573aa8bce26f77b76f1cb9ee979-Paper-Conference.pdf.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. URL https://arxiv.org/abs/1312.6114.

Marcel Kollovieh, Abdul Fatir Ansari, Michael Bohlke-Schneider, Jasper Zschiegner, Hao Wang, and Yuyang Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting, 2023. URL https://arxiv.org/abs/2307.11494.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=a-xFK8Ymz5J.

Daesoo Lee, Sara Malacarne, and Erlend Aune. Vector quantized time series generation with a bidirectional prior model, 2023. URL https://arxiv.org/abs/2303.04743.

Hongming Li, Shujian Yu, and Jose Principe. Causal recurrent variational autoencoder for medical time series generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(7): 8562–8570, Jun. 2023. doi: 10.1609/aaai.v37i7.26031. URL https://ojs.aaai.org/index.php/AAAI/article/view/26031.

Xiaomin Li, Vangelis Metsis, Huangyingrui Wang, and Anne Hee Hiong Ngu. Tts-gan: A transformer-based time-series generative adversarial network, 2022. URL https://arxiv.org/abs/2202.02691.

Shujian Liao, Hao Ni, Lukasz Szpruch, Magnus Wiese, Marc Sabate-Vidales, and Baoren Xiao. Conditional sig-wasserstein gans for time series generation, 2023. URL https://arxiv.org/abs/2006.05421.

Haksoo Lim, Minjung Kim, Sewon Park, and Noseong Park. Regular time-series generation using sgm, 2023. URL https://arxiv.org/abs/2301.08518.

Ilan Naiman, Nimrod Berman, Itai Pemper, Idan Arbiv, Gal Fadlon, and Omri Azencot. Utilizing image transforms and diffusion models for generative modeling of short and long time series, 2024a. URL https://arxiv.org/abs/2410.19538.

Ilan Naiman, N. Benjamin Erichson, Pu Ren, Michael W. Mahoney, and Omri Azencot. Generative modeling of regular and irregular time series data via koopman vaes, 2024b. URL https://arxiv.org/abs/2310.02619.

Hao Ni, Lukasz Szpruch, Marc Sabate-Vidales, Baoren Xiao, Magnus Wiese, and Shujian Liao. Sig-wasserstein gans for time series generation, 2021. URL https://arxiv.org/abs/2111.01207.

Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021. URL https://arxiv.org/abs/2102.09672.

Alexander Nikitin, Letizia Iannucci, and Samuel Kaski. Tsgm: A flexible framework for generative modeling of synthetic time series, 2024. URL https://arxiv.org/abs/2305.11567.

Hengzhi Pei, Kan Ren, Yuqing Yang, Chang Liu, Tao Qin, and Dongsheng Li. Towards generating real-world time series data, 2021. URL https://arxiv.org/abs/2111.08386.

Jian Qian, Bingyu Xie, Biao Wan, Minhao Li, Miao Sun, and Patrick Yin Chiang. Timeldm: Latent diffusion model for unconditional time series generation, 2024. URL https://arxiv.org/abs/2407.04211.

Jinwen Qiu, S. Rao Jammalamadaka, and Ning Ning. Multivariate bayesian structural time series model. *Journal of Machine Learning Research*, 19(68):1–33, 2018. URL http://jmlr.org/papers/v19/18-009.html.

Giorgia Ramponi, Pavlos Protopapas, Marco Brambilla, and Ryan Janssen. T-cgan: Conditional generative adversarial network for data augmentation in noisy time series with irregular sampling, 2019. URL https://arxiv.org/abs/1811.08295.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting, 2021. URL https://arxiv.org/abs/2101.12072.

Carl Remlinger, Joseph Mikael, and Romuald Elie. Conditional loss and deep euler scheme for time series generation, 2021. URL https://arxiv.org/abs/2102.05313.

Yulia Rubanova, Ricky T. Q. Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/42a6845a557bef704ad8ac9cb4461d43-Paper.pdf.

Lifeng Shen and James Kwok. Non-autoregressive conditional diffusion models for time series prediction, 2023. URL https://arxiv.org/abs/2306.05043.

Ilya Sutskever, James Martens, and Geoffrey Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 1017–1024, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation, 2021. URL https://arxiv.org/abs/2107.03502.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Lei Wang, Liang Zeng, and Jian Li. Aec-gan: Adversarial error correction gans for auto-regressive long time-series generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8): 10140–10148, Jun. 2023. doi: 10.1609/aaai.v37i8.26208. URL https://ojs.aaai.org/index.php/AAAI/article/view/26208.

Tianlin Xu, Li Kevin Wenliang, Michael Munn, and Beatrice Acciaio. Cot-gan: Generating sequential data via causal optimal transport. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 8798–8809. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/641d77dd5271fca28764612a028d9c8e-Paper.pdf.

Tijin Yan, Hengheng Gong, He Yongping, Yufeng Zhan, and Yuanqing Xia. Probabilistic time series modeling with decomposable denoising diffusion model. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 55759–55777. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/yan24b.html.

Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. Time-series generative adversarial networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019a. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/c9efe5f26cd17ba6216bbe2a7d26d490-Paper.pdf.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, 2019b. URL https://openreview.net/forum?id=S1zk9iRqF7.

Xinyu Yuan and Yan Qiao. Diffusion-TS: Interpretable diffusion for general time series generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4h1apFjO99.

Chen Zhicheng, FENG SHIBO, Zhong Zhang, Xi Xiao, Xingyu Gao, and Peilin Zhao. SDformer: Similarity-driven discrete transformer for time series generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=ZKbplMrDzI.

Linqi Zhou, Michael Poli, Winnie Xu, Stefano Massaroli, and Stefano Ermon. Deep latent state space models for time-series generation, 2023a. URL https://arxiv.org/abs/2212.12749.

Linqi Zhou, Michael Poli, Winnie Xu, Stefano Massaroli, and Stefano Ermon. Deep latent state space models for time-series generation, 2023b. URL https://arxiv.org/abs/2212.12749.

A Related Works

The generation of synthetic time series data has become increasingly important due to its broad applicability, with applications spanning diverse domains such as finance, healthcare, transportation and more. The unique characteristics of time series data, namely its sequential nature and temporal dependencies, necessitate specialized modeling techniques (Jarrett et al. (2021)). While real-world data is often limited or costly to obtain, the ability to synthesize realistic time series enables advancements in simulation, forecasting, and decision-making (Lim et al. (2023)). To address these challenges, various deep generative models have been developed, VAEs with GANs and, more recently, diffusion models emerging as leading approaches in this domain. Generative models have been widely used to create high-quality data in different areas, such as images, text, and audio. However, generating time series data is more difficult because it requires capturing patterns over time and dealing with noisy or incomplete information (Liao et al. (2023)). Early methods for time series generation were mainly based on GANs, which gained prominence for their capacity to produce realistic samples. GANs use two competing neural networks: a generator that creates synthetic data and a discriminator that differentiates between real and generated data (Wang et al. (2023)).

Generative Adversarial Networks (GANs). Many GAN-based models have been proposed to address specific challenges in time series generation. For example, TimeGAN (Yoon et al. (2019a)) incorporates both supervised and unsupervised learning techniques to capture temporal dynamics more effectively using recurrent neural networks (RNNs). RTSGAN (Pei et al. (2021)) and PSA-GAN (Jeha et al. (2022)) use self-attention mechanisms to generate high-quality long univariate time series samples, which is important for capturing long-range patterns. CotGAN (Xu et al. (2020)) leverages causal optimal transport theory to generate sequences with consistent temporal patterns, while RCGAN (Esteban et al. (2017)) introduces an architectural variation by conditioning on additional inputs. TTS-GAN (Li et al. (2022)) integrates a transformer encoder with GANs to generate time series data, demonstrating the utility of transformers for capturing temporal dependencies. GT-GAN (Jeon et al. (2022)) combines GANs, AEs, and differential equation models to model continuous-time flows and generate time series. Sig-WGAN (Ni et al. (2021)), which focuses on financial data, combines a continuous-time probabilistic model with the Wasserstein-1 (W1) metric to address specific challenges in generating financial time series data. Despite these advancements, GAN-based models have limitations, especially when generating long sequences. Standard architectures like RNNs and CNNs struggle to capture long-term patterns, leading to poorer performance on longer time series. Additionally, GANs can suffer from the mode collapse problem, which can hinder the generation of diverse time series samples (Remlinger et al. (2021)). GANs also come with drawback during training where basic architecture

Variational Autoencoders (VAEs). In addition to GANs, VAEs and normalizing flows gained attention for time series generation due to their probabilistic modeling capabilities. VAEs, which learn a latent representation of data by optimizing a lower bound on the data's log-likelihood, offer a powerful framework for capturing the underlying structure of time series data. TimeVAE (Desai et al. (2021)) incorporates an interpretable temporal structure, achieving reasonable performance in generating synthetic time series. Methods based on normalizing flows, which transform complex distributions into simpler ones through invertible mappings, have also been explored. Fourier Flows (Alaa et al. (2021)), a notable approach, leverages a chain of spectral filters followed by an exact likelihood optimization to synthesize time series data. These models provide a flexible, interpretable framework for modeling time series, offering distinct advantages, such as exact likelihood estimation, over traditional methods.

Diffusion models, initially developed for image, video, and text generation (Ho et al. (2020)), have recently emerged as promising alternatives to GANs for time series synthesis. Their ability to generate diverse samples without suffering from mode collapse makes them particularly attractive for this task. The application of diffusion models to time series data is relatively new but has shown significant potential. TimeGrad (Rasul et al. (2021)) uses an autoregressive diffusion process to forecast probabilistic multivariate time series, relying on RNNs to model temporal dependencies. DiffWave (Kong et al. (2021)) has applied CNN-based diffusion architectures to synthesize audio data, outperforming previous GAN-based approaches. Diffusion-TS (Yuan & Qiao (2024)) has integrated interpretability components, such as trend and seasonality, into the diffusion framework to enhance the modeling of time series data.

Several recent studies have further refined diffusion models for time series generation. CSDI (Tashiro et al. (2021)) employs self-supervised masking, inspired by image inpainting techniques, to guide the denoising process for time series. DiffTime (Coletta et al. (2023)) introduces a variant of the diffusion model by approximating the diffusion function based on CSDI, while also incorporating guided diffusion to manage constraints like trend and fixed values without requiring retraining. Biloš et al. (2023) propose an approach for modeling time series data by treating it as a discretization of an underlying continuous function. Instead of adding independent noise to individual data points, the authors introduce the concept of adding noise to the entire function using stochastic processes. Kollovieh et al. (2023) introduce TSDiff for time series modeling using an unconditionally-trained diffusion model. This model leverages a self-guidance mechanism during inference, enabling it to adapt to various downstream tasks like forecasting, imputation, and synthetic data generation without requiring task-specific training. Yan et al. (2024) introduce D3M, a general framework for constructing generative models based on the explicit solutions of linear SDEs. D3M unifies DDPM and continuous flow models, enabling the design of generative models with high generation speed and sampling quality and shows strong performance in probabilistic time series imputation. Chen et al. (2023) explore the Schrödinger bridge problem (SBP) for generative modeling, focusing on its application in probabilistic time series imputation. They provide the first convergence analysis of the approximate iterative proportional fitting (aIPF) algorithm, used to solve SBP with approximated projections. Naiman et al. (2024b) introduce KoVAE, a VAE designed for generating both regular and irregular time series data. The key idea of KoVAE lies in its linear dynamical prior, inspired by Koopman theory, which assumes the latent dynamics of the time series can be represented by a linear map. Zhou et al. (2023b) propose LS4, a generative model for time series data that utilizes a latent space governed by a state-space ordinary differential equation (ODE) to enhance modeling capacity. It leverages a convolutional representation to accelerate computations, surpassing the need to explicitly calculate hidden states. Galib et al. (2024) introduce FIDE, a conditional diffusion model specifically designed to capture the distribution of extreme values in time series generation, where it employs a high-frequency inflation strategy in the frequency domain, ensuring the sustained emphasis on block maxima. Naiman et al. (2024a) propose ImagenTime, a framework for generative modeling of time series data by transforming sequences into images and then leveraging advanced diffusion vision models. Zhicheng et al. (2024) introduce SDformer, a two-stage method for time series generation that leverages the discrete token modeling (DTM) technique.

To enhance training and inference efficiency on resource-constrained devices, latent generative models have been explored for time series generation. These models benefit from the compact and smooth nature of the latent space, enabling more efficient computations. TimeLDM (Qian et al. (2024)) combines a VAE with a latent diffusion model. The VAE encodes time series into a smoothed and informative latent representation, while the latent diffusion model operates in this latent space to generate synthetic samples. Similarly, TimeDiT (Cao et al. (2024)) leverages the transformer architecture to capture long-range temporal dependencies and employs diffusion processes in the latent space to generate high-quality samples without imposing stringent assumptions on the target distribution. These advances demonstrate the versatility of diffusion models in capturing the complex temporal structures inherent in time series data, as they offer both robustness to noise and flexibility in handling high-dimensional data.

B Limitations and future works

While our proposed MoD framework demonstrates effectiveness and superior performance in time series generation, it also hint at potential limitations and avenues for future exploration.

Model Complexity and Computational Cost: Diffusion models, in general, can be computationally expensive to train and infer, especially when applied to high-dimensional data or when a large number of diffusion steps are used. Multivariate time series data, with its complex patterns and dynamic dependencies, poses challenges for a single model to capture the full spectrum of intricacies. While the use of two specialized expert diffusers in the MoD framework improves performance, it also increases the model's complexity and computational demands compared to single-diffuser models. To address this, future research could explore techniques such as knowledge distillation to reduce the model's computational footprint without compromising accuracy.

Conditional Diffusion: Many real-world time series are influenced by factors beyond their intrinsic historical values. External data, such as weather conditions, economic indicators, or social media trends, can provide crucial contextual information for understanding the underlying patterns and dynamics. Incorporating such data can enhance the performance of downstream tasks like forecasting and anomaly detection. The current MoD framework does not explicitly account for external information. To address this limitation and improve performance, exploring conditional diffusion models, which can be conditioned on external data, is a promising avenue for future research.

Irregular Time Series: Irregular time series, characterized by non-uniformly spaced observations, present unique challenges for time series analysis. Extending the MoD framework to effectively handle irregular time series and their downstream tasks is an important direction for future research. Potential approaches include adapting the diffusion process to account for irregular time intervals or employing imputation techniques to create pseudo-regular time series.

C Model Details

C.1 Rationale behind using Dual-stage diffusion

This section delves into the effectiveness of employing dual-stage diffusers within our MoD framework, providing insights into their operation during the reverse diffusion process and supporting the rationale for selecting two specialized diffusers, as opposed to adding more diffusers.

As illustrated in Figures 3 and 5, the attention weight heatmaps reveal distinct patterns for the two specialized diffusers across different diffusion stages. Early-Stage Diffuser, designed to operate effectively at higher noise levels, exhibits higher attention weights during the initial stages of reverse diffusion (notably at T=400), where the overall structure and coarse-grained features of the data are more prominent. This influence gradually diminishes as the reverse diffusion process progresses (i.e., as T decreases from 300 to 100), reflecting the reduced significance of coarse features in later stages.

Conversely, Late-Stage Diffuser assumes a more prominent role as the noise level decreases, particularly from T=200 to T=100. At these later stages, where fine-grained, high-frequency features become more apparent, Late-Stage Diffuser exerts greater attention, effectively capturing the intricate details that emerge as the data becomes less noisy. The observed transition of attention weights between Early-Stage and Late-Stage Diffusers demonstrates their complementary roles throughout the diffusion process, ensuring that both global and local aspects of the data are adequately represented.

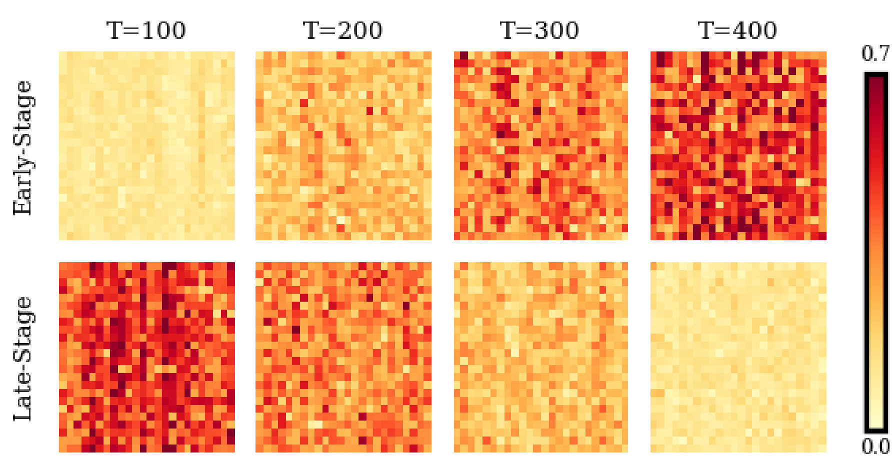


Figure 5: Attention weight heatmaps for both diffusers on Sines dataset.

Table 5: Performance comparison of MoD with varying numbers of diffusers.

Metric	Model	ETTh	Stocks	Sines
Discriminative Score	MoD	0.009	0.008	0.006
	MoD #3	0.010	0.034	0.010
	MoD #4	0.008	0.060	0.008
Predictive Score	MoD	0.121	0.037	0.094
	MoD #3	0.121	0.037	0.093
	MoD #4	0.122	0.038	0.093
Inference Time (ms per sample)	MoD	12.13	8.34	7.93
	MoD #3	18.14	12.58	11.91
	MoD #4	24.21	16.74	16.0

Experimental results, summarized in Table 5, further support the efficacy of this approach. A comparison of MoD with three and four diffusers reveals that while MoD with two diffusers achieves strong performance metrics across all datasets (ETTh, Stocks, Sines), adding a third or fourth diffuser does not significantly improve the discriminative or predictive scores. For instance, the discriminative score for ETTh is 0.009 for MoD and 0.010 for MoD #3, indicating a negligible difference. Similarly, the predictive scores across the models remain virtually unchanged. However, increasing the number of diffusers introduces additional model complexity. As shown in Table 5, moving from two to three or four diffusers leads to a substantial increase in inference time, rising from 12.13 ms (MoD) to 24.21 ms (MoD #4) per sample for ETTh. This increased computational burden is not justified by corresponding gains in predictive or discriminative performance, suggesting a point of diminishing returns.

These results confirm that the use of two expert diffusers strikes an effective balance between model performance and computational efficiency. This dual-stage approach is well-suited for managing varying noise levels during the reverse diffusion process, with one diffuser focusing on recovering coarse-grained features and the other on refining fine-grained details. Adding more diffusers does not yield tangible benefits, indicating that the time-dependent weighting scheme and expertise of each diffuser are well-aligned with the demands of multivariate time series data at different stages. This ensures that the MoD framework remains efficient while maintaining a high modeling capacity, capable of capturing both broad patterns and intricate details in time series data.

C.2 Training and Inference Algorithms

Algorithm 1 Training

Require: Time series data x_0; Diffusion steps T

1: **repeat**
2: $t \sim$ Uniform($\{1, ..., T\}$)
3: $\epsilon \sim \mathcal{N}(0, I)$
4: Sample x_t using Equation 4
5: Compute ϵ_θ using Equation 10
6: Sample x_{t-1} using Equation 9
7: Compute loss using Equation 11
8: Take gradient descent step on $\bigtriangledown_\theta \mathcal{L}$
9: **until** Converged

Algorithm 2 Inference

1: $x_T \sim \mathcal{N}(0, I)$
2: **for** $i = T$ to 1 **do**
3: $z \sim \mathcal{N}(0, I)$ if $t > 1$ else $z = 0$
4: Sample x_{t-1} using Equation 9
5: **end for**
6: **return** x_0

D Experiments Details

D.1 Datasets

The datasets employed in this study encompass a diverse range of temporal patterns and applications:

1. **Sines**: A controlled benchmark dataset with customizable frequencies.
2. **Stock** and **Stockv** (Yahoo and Google): High-volatility, non-stationary financial data spanning the 2008 financial crisis.
3. **ETTh**: Industrial transformer data from China (2016-2018) exhibiting seasonal patterns and long-term trends.
4. **MuJoCo**: Physically-constrained humanoid motion trajectories simulated using a physics engine.
5. **Energy**: Electricity consumption data from Chièvres, Belgium, capturing complex interactions between 28 appliance circuits at 10-minute intervals over approximately 4.5 months.

The datasets used in this study are publicly available at the links provided in Table 5. Their statistical properties are also presented. We employ an overlapping sliding windows mechanism to arrange the data. This technique involves sliding a fixed-size window across the data, one step at a time. By allowing each data point to be part of multiple windows, this approach helps preserve the underlying temporal relationships within the sequence.

Table 6: Dataset Statistics.

Dataset	#Samples	#Features	Link
Sines	10000	5	https://github.com/jsyoon0823/TimeGAN
Stocks	3773	6	https://finance.yahoo.com/quote/GOOG
Stockv	3919	6	https://github.com/abudesai/timeVAE/tree/main/data
ETTh	17420	7	https://github.com/zhouhaoyi/ETDataset
MuJoCo	10000	14	https://github.com/deepmind/dm control
Energy	19711	28	https://archive.ics.uci.edu/ml/datasets

D.2 Baselines

The generative experiments were conducted using the open-source code repositories specified in Table 6, which we adapted as needed.

Table 7: Baseline Source Code References.

Baseline	Link
KoVAE (Naiman et al. (2024b))	https://github.com/azencot-group/KoVAE
Diffusion-TS (Yuan & Qiao (2024))	https://github.com/Y-debug-sys/Diffusion-TS
TimeGAN (Yoon et al. (2019a))	https://github.com/jsyoon0823/TimeGAN
TimeVAE (Desai et al. (2021))	https://github.com/abudesai/timeVAE
Diffwave (Kong et al. (2021))	https://diffwave-demo.github.io/
Cot-GAN (Xu et al. (2020))	https://github.com/tianlinxu312/cot-gan

D.3 Evaluation Metrics

To comprehensively assess the quality of generated synthetic time series data, we employ a suite of metrics designed to evaluate the following key aspects:

- **Distributional Similarity**: The extent to which the synthetic data aligns with the underlying distribution of the real data.
- **Temporal Dependencies**: The preservation of temporal relationships and patterns inherent in the real data.
- **Predictive Utility**: The suitability of synthetic data as input for predictive models.

We utilize the following specific metrics:

1. **Discriminative Score**: A classification model is trained to differentiate between real and synthetic data. A lower discriminative score, ideally approaching 0.5, indicates that the synthetic data is indistinguishable from real data by the discriminator (Yoon et al. (2019a)).
2. **Predictive Score**: A post-hoc sequence model is trained to forecast future values using the Training-on-Synthetic and Testing-on-Real (TSTR) method. A lower predictive score suggests that the synthetic data can effectively support predictive tasks (Yoon et al. (2019a)). This approach validates the preservation of underlying predictive relationships in the synthetic data.
3. **Context-Frechet Inception Distance (Context-FID)**: This metric quantifies the quality of synthetic data by measuring the difference in representations of time series that align with the local context Jeha et al. (2022). The score is computed using both mean and covariance statistics of the feature representations, providing a comprehensive measure of temporal coherence.
4. **Correlational Score**: Evaluates temporal dependency by calculating the absolute error between cross-correlation matrices of the real and synthetic data Ni et al. (2021). This metric is particularly effective at capturing both concurrent and lagged relationships across multiple variables in the time series.

D.4 Model Parameters

To establish default hyperparameters that perform well across various datasets, we conducted a limited hyperparameter tuning process. The parameters explored included batch size (32, 64, 128), the number of attention heads (4, 8), the number of basic dimensions (32, 64, 96, 128), and diffusion steps (50, 200, 500, 1000). Model training was executed on a single Nvidia 4090 GPU. Throughout our experiments, we employed cosine noise scheduling and optimized the network using the Adam optimizer with (β_1, β_2) = (0.9, 0.96). The learning rate was initialized at 8e-4 and followed a linear decay schedule after 500 warmup iterations. For the KL loss, we set the λ_{kl} parameter to 1e-2. Table 7 provides a comprehensive list of the hyperparameter settings used. We employed a dropout rate of 0.1 and a residual dropout rate of 0.1. The Gaussian Error Linear Unit (GELU) activation function was used throughout the model. A weight decay of 0.995 and an update interval of 10 were applied for the Exponential Moving Average (EMA). To improve the reliability and reproducibility of our experiments, all metrics were averaged over 10 runs.

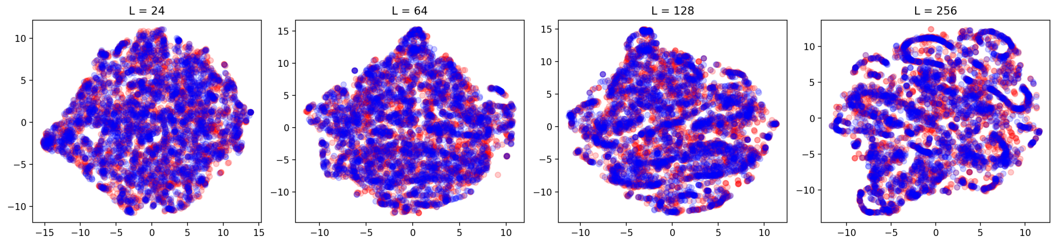
Table 8: Hyperparameters and training details for each dataset

Parameter	Sines	Stocks	ETTh	MuJoCo	Energy
Basic dimension	256	256	256	256	256
Attention heads	4	4	4	4	4
Attention head dimension	64	64	64	64	64
Encoder layers	1	2	3	2	4
MLP dimension	1024	1024	1024	512	1024
Batch size	128	64	128	128	64
Sample size	256	256	256	256	256
Timesteps / sampling steps	500	500	500	1000	1000
Training steps	12000	10000	18000	14000	25000
Training Time/ms per epoch	43.6	41	51.9	217.4	267.5
Inference time / ms per sample	7.93	8.34	12.13	24.49	31.34

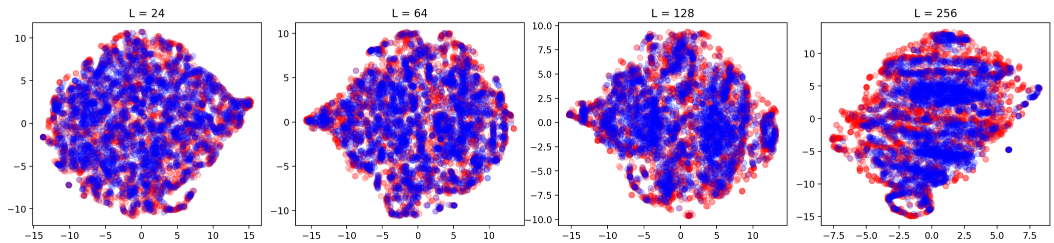
E Additional Experimental Results

E.1 Extended results for long-term generation

To further assess the model's ability to generate long, multivariate time series data, we conducted a comparative analysis using t-SNE visualizations. We focused on two datasets: ETTh and Energy. As illustrated in Figures 6, the model's generated data for the ETTh dataset exhibited a notable



(a) MoD



(b) Diffusion-TS

Figure 6: Effect of Sequence Length on MoD and Diffusion-TS Performance on ETTh..

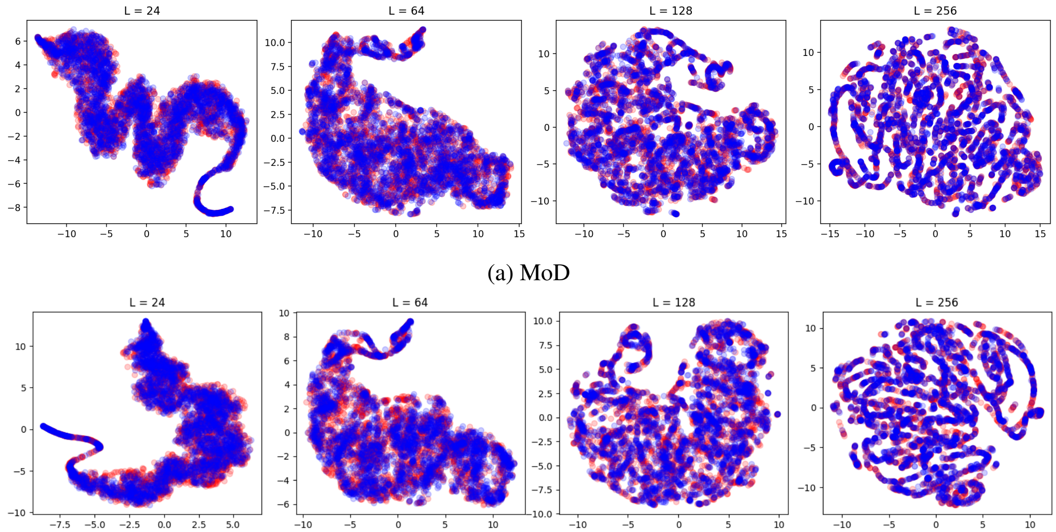
improvement in similarity to the original data as sequence length increased. While the high dimensionality and large number of data points in the Energy dataset made the distinction less apparent, certain regions of the Figure 7, particularly for sequence length 64, revealed a discernible advantage of our model over the Diffusion-TS baseline.

Table 9: Performance on long-term time series generation.

Dataset	Metrics	Length	MoD	Diffusion-TS	TimeGAN	TimeVAE	Diffwave	DiffTime	Cot-GAN
ETTh	Context-FID	64	**0.034±.002**	0.631±.058	1.130±.102	0.827±.146	1.543±.153	1.279±.083	3.008±.277
		128	**0.065±.002**	0.787±.062	1.553±.169	1.062±.134	2.354±.170	2.554±.318	2.639±.427
		256	**0.112±.008**	0.423±.038	5.872±.208	0.826±.093	2.899±.289	3.524±.830	4.075±.894
	Correlational	64	**0.029±.007**	0.082±.005	0.483±.019	0.067±.006	0.186±.008	0.094±.010	0.271±.007
		128	**0.032±.011**	0.088±.005	0.188±.006	0.054±.007	0.203±.006	0.113±.012	0.176±.006
		256	**0.027±.007**	0.064±.007	0.522±.013	0.046±.007	0.199±.003	0.135±.006	0.222±.010
	Discriminative	64	**0.016±.006**	0.106±.048	0.227±.078	0.171±.142	0.254±.074	0.150±.003	0.296±.348
		128	**0.023±.019**	0.144±.060	0.188±.074	0.154±.087	0.274±.047	0.176±.015	0.451±.080
		256	**0.025±.010**	0.060±.030	0.442±.056	0.178±.076	0.304±.068	0.243±.005	0.461±.010
	Predictive	64	**0.113±.006**	0.116±.000	0.132±.008	0.118±.004	0.133±.008	0.118±.004	0.135±.003
		128	**0.104±.006**	0.110±.003	0.153±.014	0.113±.005	0.129±.003	0.120±.008	0.126±.001
		256	0.114±.007	**0.109±.013**	0.220±.008	0.110±.027	0.132±.001	0.118±.003	0.129±.000
Energy	Context-FID	64	**0.041±.004**	0.135±.017	1.230±.070	2.662±.087	2.697±.418	0.762±.157	1.824±.144
		128	**0.043±.003**	0.087±.019	2.535±.372	3.125±.106	5.552±.528	1.344±.131	1.822±.271
		256	**0.058±.003**	0.126±.024	5.032±.831	3.768±.998	5.572±.584	4.735±.729	2.533±.467
	Correlational	64	**0.411±.033**	0.672±.035	3.668±.106	1.653±.208	6.847±.083	1.281±.218	3.319±.062
		128	**0.243±.026**	0.451±.079	4.790±.116	1.820±.329	6.663±.112	1.376±.201	3.713±.055
		256	**0.247±.052**	0.361±.092	4.487±.214	1.279±.114	5.690±.102	1.800±.138	3.739±.089
	Discriminative	64	0.085±.017	**0.078±.021**	0.498±.001	0.499±.000	0.497±.004	0.328±.031	0.499±.001
		128	0.239±.058	**0.143±.075**	0.499±.001	0.499±.000	0.499±.001	0.396±.024	0.499±.001
		256	**0.287±.111**	0.290±.123	0.499±.000	0.499±.000	0.499±.000	0.437±.095	0.498±.004
	Predictive	64	**0.249±.000**	**0.249±.000**	0.291±.003	0.302±.001	0.252±.001	0.252±.000	0.262±.002
		128	**0.247±.000**	**0.247±.000**	0.303±.002	0.318±.000	0.252±.000	0.251±.000	0.269±.002
		256	**0.245±.000**	0.245±.001	0.351±.004	0.353±.003	0.251±.000	0.251±.000	0.275±.004

E.2 Extended results for Data Scarcity

To demonstrate the model's effectiveness in scenarios with limited data, we have included additional experimental results beyond those presented in the main paper due to space constraints. These results encompass datasets with 100%, 5%, and 2% of the original data, representing increasingly



(a) MoD

(b) Diffusion-TS

Figure 7: Effect of Sequence Length on MoD and Diffusion-TS Performance on Energy.

challenging conditions. Our findings consistently indicate that the model can generate high-quality samples even when provided with minimal data.

F Modeling Coarse-grained and Fine-grained Features

To validate our hypothesis, we conducted experiments, the results of which are presented in Figure 8. Before delving into these results, we will elucidate the underlying concept.

It is well-established in image generation that the progressive introduction of noise during the forward diffusion process initially obscures fine-grained details, such as edges, textures, and small objects. These details are highly sensitive to noise perturbations. As the noise level increases, larger-scale features, including overall shapes and extensive regions, become increasingly blurred and distorted. These coarse-grained features, being less susceptible to early noise additions, remain relatively intact for a longer duration. Consequently, during the early stages of the reverse diffusion process, these dominant, coarse-grained features emerge more prominently. As the noise level diminishes, finer details gradually gain prominence.

A similar phenomenon occurs in time series data. Coarse-grained features, characterized by low-frequency, long-term trends, exhibit slower dynamics, rendering them less vulnerable to early noise corruption. Even under high noise conditions, long-term dependencies and global trends maintain a degree of robustness. Conversely, fine-grained features, comprising high-frequency, short-term fluctuations, are quickly obscured by noise, becoming more salient as the noise level decreases in the later stages of the reverse process.

To illustrate this concept, Figure 8 presents an example from the Air dataset, visualizing the diffusion process by our MoD. Time series data can be decomposed into three primary components: trends, seasonality, and residuals. Figure Z depicts the following:

- Row 1: Original time series and its decomposed components.
- Row 2: Generated data and its decomposed components at diffusion timestep T=400. The early-stage diffuser, with a weight of w1=0.9, dominates the contribution to the generated data.
- Row 3: Generated data and its decomposed components at diffusion timestep T=100. The late-stage diffuser, with a weight of w2=0.9, primarily contributes to the generated data.

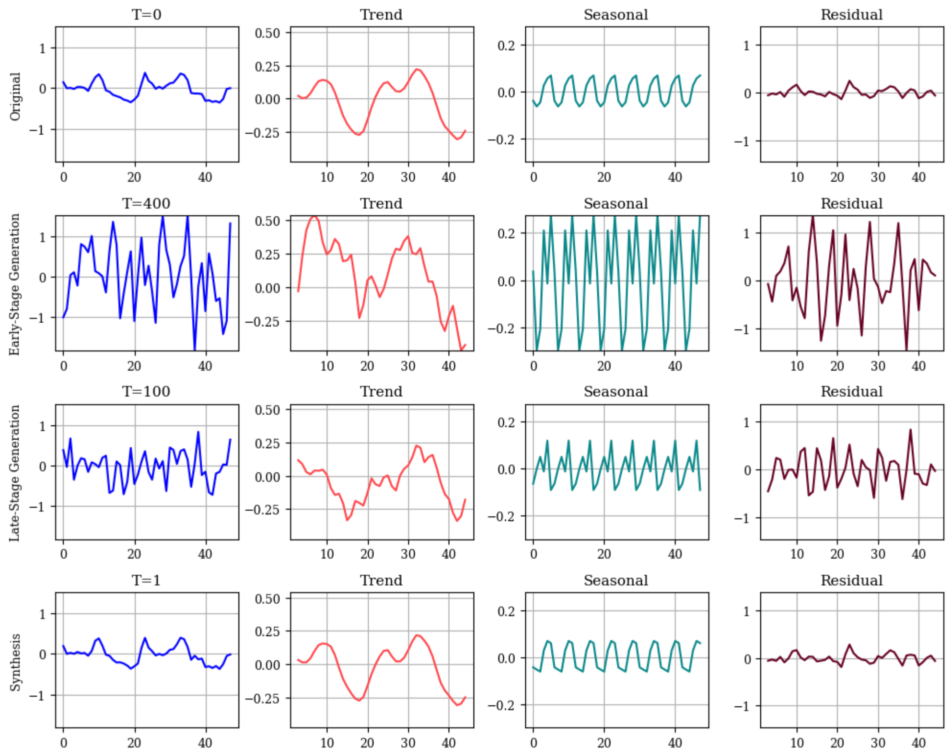


Figure 8: Visualizing the emergence of coarse-grained and fine-grained features in time series data generated by our MoD at different diffusion timesteps. The early-stage diffuser prioritizes coarse-grained features (Row 2), while the late-stage diffuser focuses on fine-grained features (Row 3). The MoD effectively balances both (Row 4).

- Row 4: Final generated data by the Model of Diffusion (MoD) at diffusion timestep T=1. Based on these visualizations, we observe the following:

Feature-Level Analysis: Time series data encompasses two primary feature types:

- Coarse-grained features: Low-frequency components representing long-term trends (Overall upward or downward movements in the data).

- Fine-grained features: High-frequency components representing short-term fluctuations (Rapid changes in the data that occur over short periods).

Early-Stage Generation: At T=400, the generated data exhibits a trend similar to the original data, characterized by two upward and one downward movement. This demonstrates the early-stage diffuser's specialization in modeling coarse-grained features, which are less susceptible to noise. Conversely, the residual component of the generated data differs significantly from the original, highlighting the difficulty of learning fine-grained representations at this stage.

Late-Stage Generation: At T=100, the trend of the generated data aligns more closely with the original, reflecting the refinement provided by the early-stage diffuser. The late-stage diffuser, now dominant, effectively models fine-grained features, resulting in a residual component that more closely resembles the original. MoD Generation: The final generated data, produced by the MoD, represents a combination of the contributions from both diffusers, effectively capturing both coarse-grained and fine-grained features.

Table 10: Performance comparison of MoD and Diffusion-TS variants on Stockv and Sines datasets.

Metric	Methods	Discriminative	Predictive	# Parameters	Train (ms/epoch)	Inference (ms/sample)
Stockv	MoD	**0.007 ±.004**	**0.011±.001**	316.22K	56.45	14.37
	Diffusion-TS	0.187±.009	0.025±.000	203.36K	39.74	9.88
	Diffusion-TS Ensemble	0.118±.002	0.024±.000	406.81K	73.89	20.32
Sines	MoD	**0.005±.003**	**0.093±.000**	307.38K	62.73	20.08
	Diffusion-TS	0.057±.010	0.096±.000	229.75K	50.23	14.35
	Diffusion-TS Ensemble	0.038±.006	0.095±.000	459.47K	96.08	29.61

G Complexity Analysis

In this section, we conduct experiments comparing our MoD model with two variants of the Diffusion-TS model (standard and ensemble versions) across the Stockv and Sines datasets. The comparison considered key metrics such as discriminative and predictive performance, model size (in terms of parameters), and computational efficiency (training and inference time). The results in Table 10 reveal the following:

Performance Metrics: (1) Stockv: MoD achieved the lowest discriminative score, representing a 96.2% improvement over Diffusion-TS and a 94.1% improvement over Diffusion-TS Ensemble. In terms of predictive performance, MoD also outperformed the other models, showing a 56% improvement over Diffusion-TS and a 54% improvement over Diffusion-TS Ensemble. (2) Sines: MoD exhibited a discriminative score 91.2% better than Diffusion-TS and 86.8% better than Diffusion-TS Ensemble. The predictive performance of MoD was slightly superior to both Diffusion-TS and Diffusion-TS Ensemble, with a marginal difference of approximately 3.1% and 2.1%, respectively.

Model Size: MoD had fewer parameters compared to the Diffusion-TS Ensemble, which had a larger model size by 28.6% on Stockv and 44.7% on Sines. The Diffusion-TS model was more compact, requiring 35.6% fewer parameters than MoD on Stockv and 25.5% fewer parameters on Sines. However, despite its smaller size, Diffusion-TS did not achieve the same performance as MoD, particularly in terms of discriminative accuracy.

Computational Efficiency: (1) Train Time: MoD took 42.2% longer to train than Diffusion-TS on Stockv, and 24.9% longer on Sines. However, the Diffusion-TS Ensemble took 30.8% longer to train on Stockv and 47.3% longer on Sines. Compared to the Diffusion-TS Ensemble, MoD was 23.5% more efficient on Stockv and 34.5% more efficient on Sines. (2) Inference Time: MoD required 45.3% more time for inference on Stockv and 39.7% more on Sines compared to Diffusion-TS. However, MoD was 29.6% more efficient than the Diffusion-TS Ensemble on Stockv and 32.4% more efficient on Sines.

In conclusion, our MoD model demonstrates a strong performance-to-complexity ratio across both datasets. It consistently achieves the best discriminative and predictive performance compared to the Diffusion-TS variants, despite having a comparable or slightly larger number of parameters. Moreover, MoD strikes an effective balance between computational efficiency and performance. It requires fewer resources for training and inference compared to the Diffusion-TS Ensemble, while still maintaining a clear advantage in accuracy. Although the standard Diffusion-TS model is faster for training and inference, MoD's superior accuracy makes it the preferred choice when performance is prioritized over raw speed.

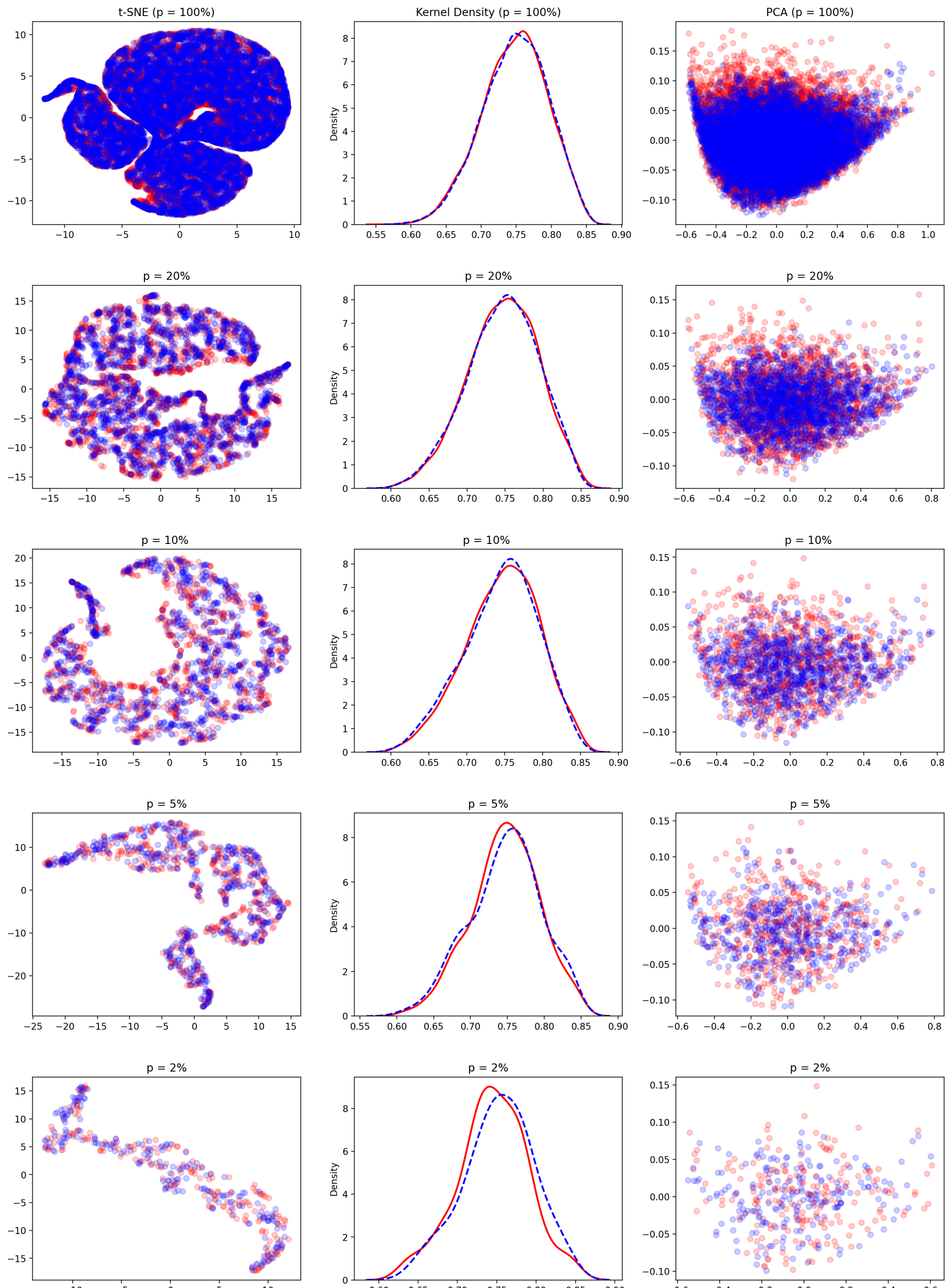


Figure 9: t-SNE, KDE, and PCA visualization of MoD performance on Sines dataset with varying sizes.

Table 11: Detailed results of scarce data generation. (Bold indicates best performance, underline indicates second-best performance).

Train Size	Metric	Methods	Air	Energy	Sines	Stockv
100%	Discriminative Score (Lower the Better)	MoD	**0.212±.005**	**0.087±.008**	**0.005±.003**	**0.007±.004**
		Diffusion-TS	0.177±.014	0.408±.008	0.057±.010	0.187±.009
		TimeVAE	0.381±.037	0.498±.006	0.021±.040	0.009±.009
		TimeGAN	0.387±.026	0.499±.000	0.437±.023	0.011±.013
		T-Forcing	0.495±.010	0.499±.001	0.484±.006	0.450±.099
		RCGAN	0.495±.002	0.500±.000	0.382±.075	0.494±.006
	Predictive Score (Lower the Better)	MoD	**0.005±.000**	**0.251±.000**	**0.093±.000**	**0.011±.001**
		Diffusion-TS	0.014±.014	0.256±.000	0.096±.000	0.025±.000
		TimeVAE	0.013±.002	0.268±.004	0.213±.000	0.019±.001
		TimeGAN	0.005±.000	0.298±.002	0.251±.027	0.021±.001
		T-Forcing	0.101±.028	0.287±.035	0.220±.010	0.082±.004
		RCGAN	0.043±.002	0.277±.011	0.262±.024	0.025±.002
20%	Discriminative Score (Lower the Better)	MoD	**0.159±.027**	**0.061±.004**	**0.008±.022**	**0.016±.033**
		Diffusion-TS	0.186±.012	0.379±.014	0.024±.012	0.118±.011
		TimeVAE	0.350±.089	0.499±.002	0.039±.030	0.176±.208
		TimeGAN	0.355±.045	0.493±.007	0.374±.102	0.042±.068
		T-Forcing	0.500±.000	0.500±.001	0.490±.003	0.372±.241
		RCGAN	0.500±.000	0.500±.000	0.281±.132	0.479±.028
	Predictive Score (Lower the Better)	MoD	**0.006±.000**	**0.195±.000**	**0.092±.001**	0.025±.001
		Diffusion-TS	0.026±.014	0.251±.000	0.093±.000	**0.024±.000**
		TimeVAE	0.019±.003	0.288±.002	0.215±.000	0.052±.001
		TimeGAN	0.007±.002	0.324±.005	0.287±.051	0.050±.001
		T-Forcing	0.139±.061	0.256±.006	0.219±.007	0.091±.024
		RCGAN	0.480±.315	0.751±.434	0.254±.001	0.164±.122
10%	Discriminative Score (Lower the Better)	MoD	**0.101±.023**	**0.056±.019**	**0.009±.017**	**0.053±.029**
		Diffusion-TS	0.187±.025	0.340±.020	0.012±.005	0.104±.017
		TimeVAE	0.425±.067	0.499±.001	0.053±.045	0.080±.108
		TimeGAN	0.257±.093	0.500±.001	0.382±.055	0.068±.106
		T-Forcing	0.500±.000	0.500±.001	0.482±.006	0.474±.071
		RCGAN	0.500±.000	0.500±.000	0.246±.234	0.474±.071
	Predictive Score (Lower the Better)	MoD	**0.003±.000**	**0.188±.000**	**0.092±.000**	**0.026±.000**
		Diffusion-TS	0.013±.012	0.252±.000	0.092±.000	0.027±.000
		TimeVAE	0.005±.003	0.275±.001	0.215±.000	0.075±.001
		TimeGAN	0.003±.001	0.318±.006	0.300±.059	0.081±.008
		T-Forcing	0.157±.027	0.262±.014	0.216±.002	0.118±.040
		RCGAN	0.605±.618	0.740±.371	0.241±.030	0.150±.094
5%	Discriminative Score (Lower the Better)	MoD	**0.178±.043**	**0.061±.007**	**0.005±.010**	**0.035±.001**
		Diffusion-TS	0.136±.026	0.327±.013	0.026±.012	0.062±.054
		TimeVAE	0.292±.207	0.500±.001	0.051±.068	0.191±.141
		TimeGAN	0.355±.230	0.497±.003	0.281±.185	0.040±.029
		T-Forcing	0.497±.006	0.499±.003	0.484±.014	0.449±.069
		RCGAN	0.500±.000	0.500±.000	0.499±.003	0.500±.000
	Predictive Score (Lower the Better)	MoD	**0.005±.014**	**0.179±.000**	**0.090±.000**	**0.025±.000**
		Diffusion-TS	0.009±.001	0.258±.001	0.094±.000	0.028±.000
		TimeVAE	0.040±.002	0.262±.002	0.218±.001	0.084±.004
		TimeGAN	0.046±.005	0.329±.010	0.262±.032	0.080±.001
		T-Forcing	0.185±.012	0.263±.011	0.221±.006	0.131±.026
		RCGAN	0.428±.045	0.499±.054	0.222±.002	0.669±.252
2%	Discriminative Score (Lower the Better)	MoD	**0.055±.038**	**0.067±.038**	**0.013±.020**	**0.062±.002**
		Diffusion-TS	0.063±.040	0.302±.023	0.042±.013	0.119±.060
		TimeVAE	0.154±.163	0.492±.018	0.048±.058	0.300±.147
		TimeGAN	0.167±.242	0.468±.062	0.192±.273	N/A
		T-Forcing	0.500±.000	0.500±.000	0.295±.320	0.300±.316
		RCGAN	0.494±.017	0.500±.000	0.492±.021	N/A
	Predictive Score (Lower the Better)	MoD	**0.005±.000**	**0.172±.001**	**0.091±.002**	**0.024±.002**
		Diffusion-TS	0.011±.002	0.257±.001	0.093±.000	0.027±.000
		TimeVAE	0.056±.005	0.260±.003	0.223±.001	0.153±.008
		TimeGAN	0.057±.003	0.313±.003	0.270±.004	N/A
		T-Forcing	0.185±.086	0.266±.005	0.225±.005	0.155±.003
		RCGAN	0.361±.079	1.451±.024	0.320±.005	N/A

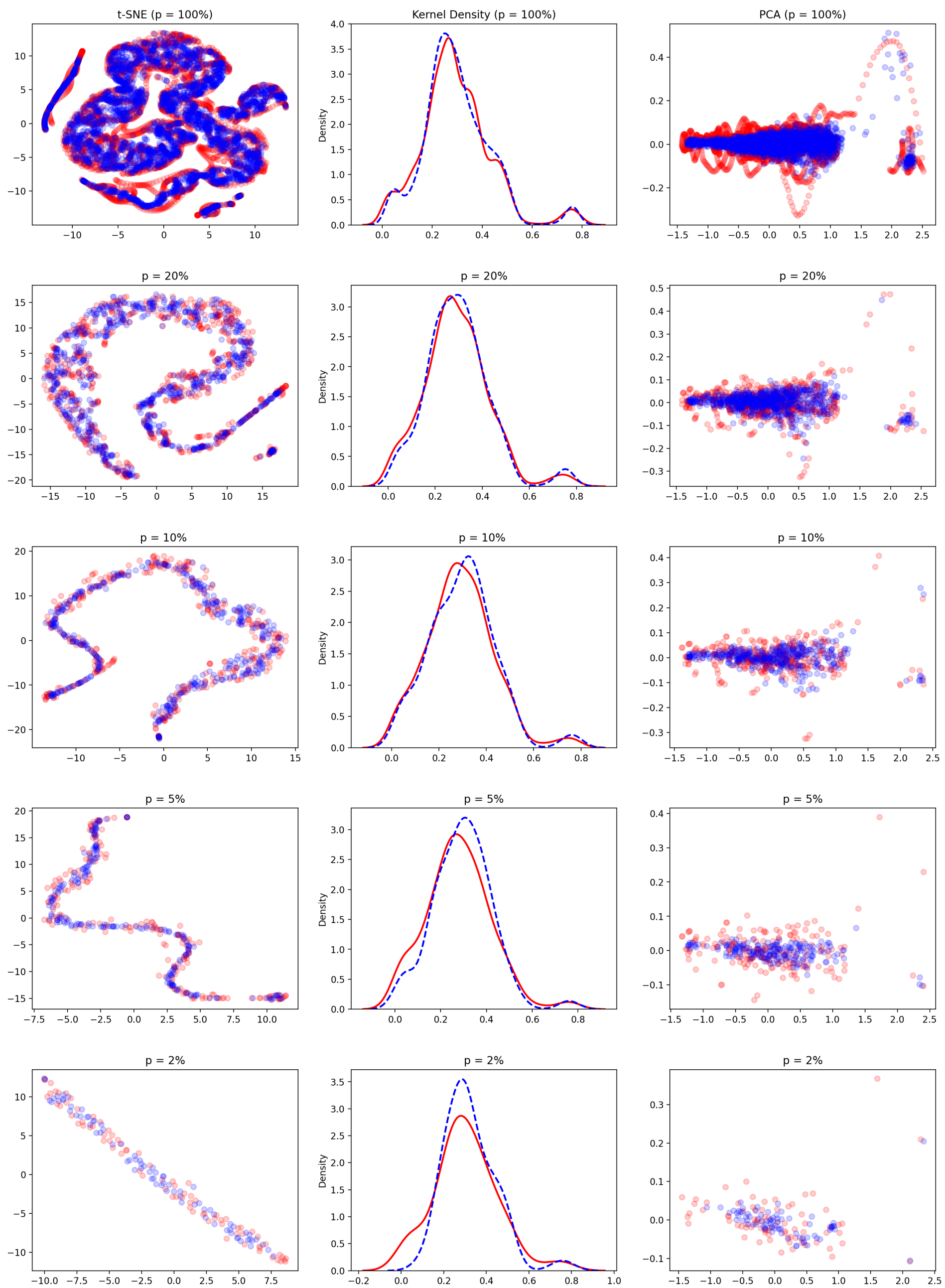


Figure 10: t-SNE, KDE, and PCA visualization of MoD performance on Stockv dataset with varying sizes.

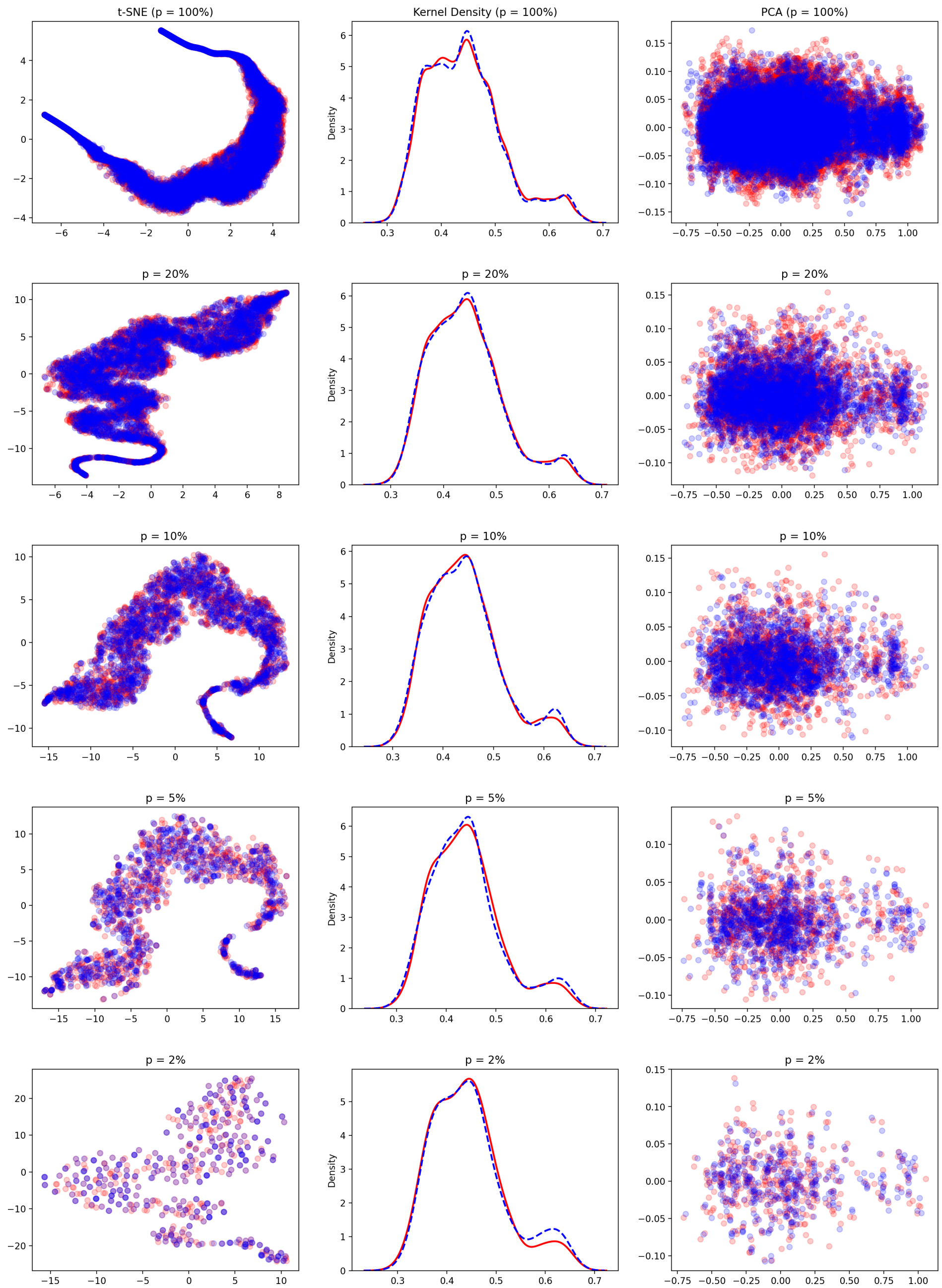


Figure 11: t-SNE, KDE, and PCA visualization of MoD performance on Energy dataset with varying sizes.

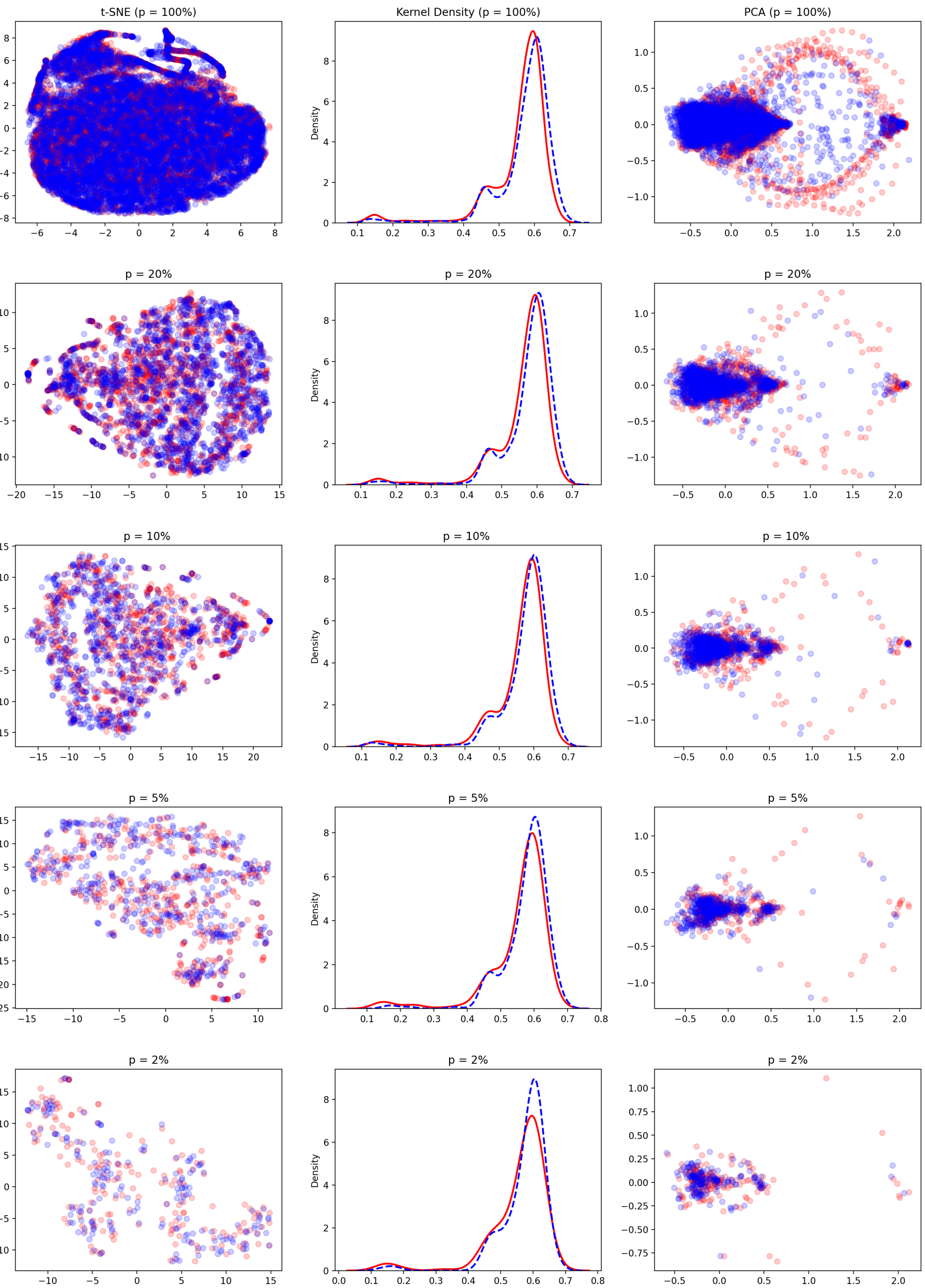


Figure 12: t-SNE, KDE, and PCA visualization of MoD performance on Air dataset with varying sizes.

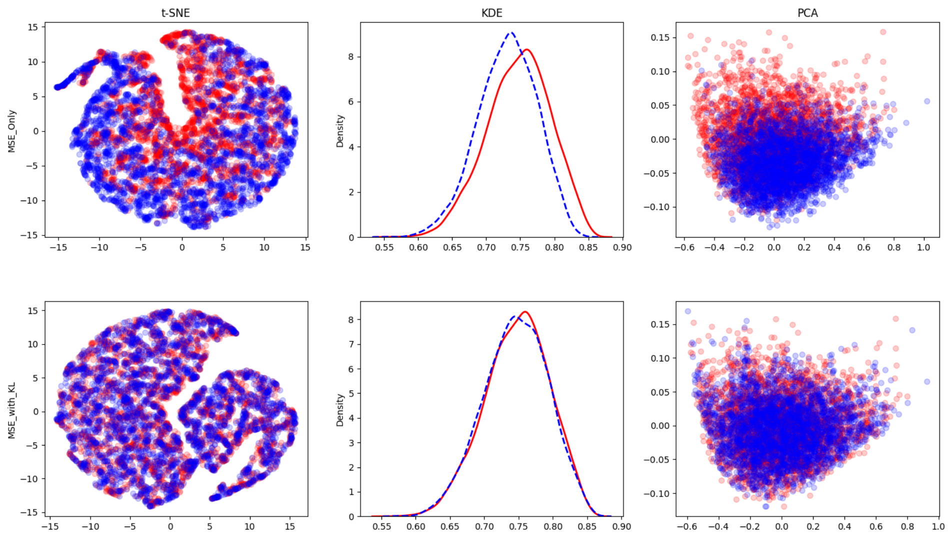


Figure 13: t-SNE, KDE, and PCA visualizations for the Sines dataset. This figure compares the generated samples from the 'MSE Only' and 'MSE with KL' models. The addition of KL divergence results in a more accurate and well-defined distribution of generated samples.

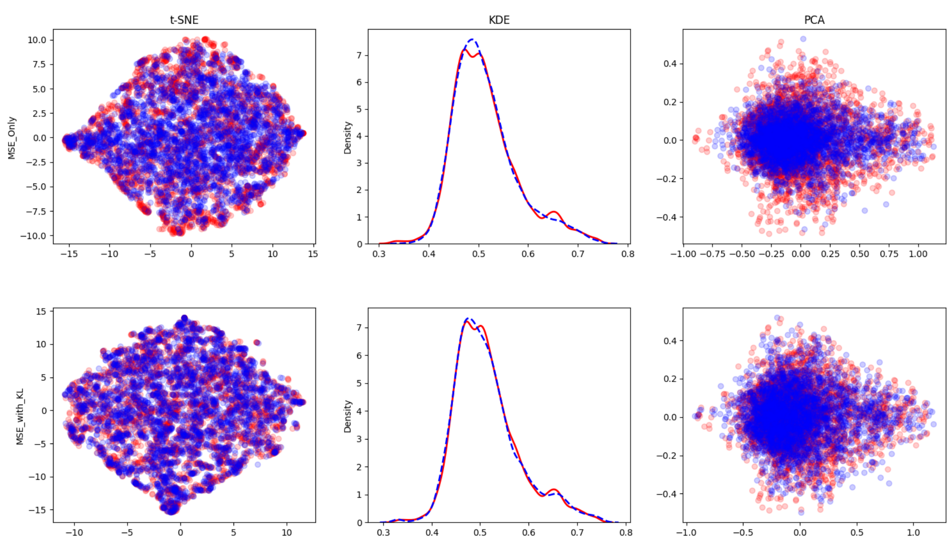


Figure 14: t-SNE, KDE, and PCA visualizations for the ETTh dataset. This figure compares the generated samples from the 'MSE Only' and 'MSE with KL' models. The 'MSE with KL' model exhibits improved performance, particularly in the KDE plot, indicating a closer alignment with the true data distribution.