
Retrieval-Augmented Diffusion Models for Time Series Forecasting

Jingwei Liu^{1,2*} Ling Yang^{3 †} Hongyan Li^{1,2 ‡} Shenda Hong^{3,4,5 ‡}

¹School of Intelligence Science and Technology, Peking University

²National Key Laboratory of General Artificial Intelligence, Peking University

³Institute of Medical Technology, Peking University Health Science Center

⁴National Institute of Health Data Science, Peking University

⁵Institute for Artificial Intelligence, Peking University

jingweiliu1996@163.com, yangling0818@163.com

{leehy, hongshenda}@pku.edu.cn

Abstract

While time series diffusion models have received considerable focus from many recent works, the performance of existing models remains highly unstable. Factors limiting time series diffusion models include insufficient time series datasets and the absence of guidance. To address these limitations, we propose a Retrieval-Augmented Time series Diffusion model (RATD). The framework of RATD consists of two parts: an embedding-based retrieval process and a reference-guided diffusion model. In the first part, RATD retrieves the time series that are most relevant to historical time series from the database as references. The references are utilized to guide the denoising process in the second part. Our approach allows leveraging meaningful samples within the database to aid in sampling, thus maximizing the utilization of datasets. Meanwhile, this reference-guided mechanism also compensates for the deficiencies of existing time series diffusion models in terms of guidance. Experiments and visualizations on multiple datasets demonstrate the effectiveness of our approach, particularly in complicated prediction tasks. Our code is available at <https://github.com/stanliu96/RATD>

1 Introduction

Time series forecasting plays a critical role in a variety of applications including weather forecasting [15, 11], finance forecasting [7, 5], earthquake prediction [19] and energy planning [6]. One way to approach time series forecasting tasks is to view them as conditional generation tasks [32, 42], where conditional generative models are used to learn the conditional distribution $P(\mathbf{x}^P | \mathbf{x}^H)$ of predicting the target time series \mathbf{x}^P given the observed historical sequence \mathbf{x}^H . As the current state-of-the-art conditional generative model, diffusion models [12] have been utilized in many works for time series forecasting tasks [28, 36, 30].

Although the performance of the existing time series diffusion models is reasonably well on some time series forecasting tasks, it remains unstable in certain scenarios (an example is provided in 1(c)). The factors limiting the performance of time series diffusion models are complex, two of them are particularly evident. First, most time series lack direct semantic or label correspondences, which often results in time series diffusion models lacking meaningful **guidance** during the generation

*Contact: Jingwei Liu, jingweiliu1996@163.com

†Contributed equally.

‡Corresponding Authors: Hongyan Li, Shenda Hong

process(such as text guidance or label guidance in image diffusion models). This also limits the potential of time series diffusion models.

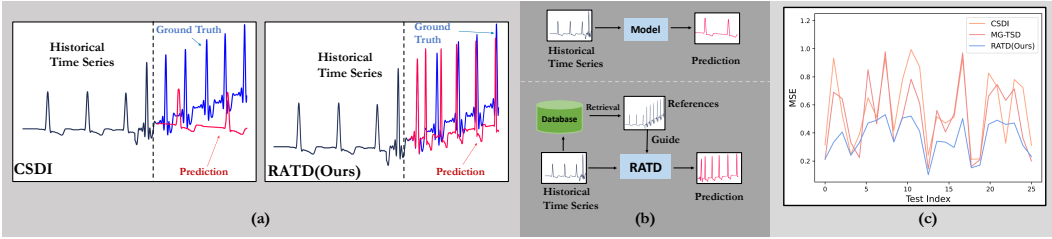


Figure 1: (a) The figure shows the differences in forecasting results between the CSDI [36] (left) and RATD (right). Due to the very small proportion of such cases in the training set, CSDI struggles to make accurate predictions, often predicting more common results. Our method, by retrieving meaningful references as guidance, makes much more accurate predictions. (b) A comparison between our method’s framework(bottom) and the conventional time series diffusion model framework(top). (c) We randomly selected 25 forecasting tasks from the electricity dataset. Compared to our method, CSDI and MG-TSD [9] exhibited significantly higher instability. This indicates that the RATD is better at handling complex tasks that are challenging for the other two methods.

The second limiting factor arises from two shortcomings of the time series datasets: **size insufficient** and **imbalanced**. Compared to image datasets, time series datasets typically have a smaller scale. Popular image datasets (such as LAION-400M) contain 400 million sample pairs, while most time series datasets usually only contain tens of thousands of data points. Training a diffusion model to learn the precise distribution of datasets with insufficient size is challenging. Additionally, real-world time series datasets exhibit significant imbalance. For example, in the existing electrocardiogram dataset MIMIC-IV, records related to diagnosed pre-excitation syndrome (PS) account for less than 0.025% of the total records. This imbalance phenomenon may cause models to overlook some extremely rare complex samples, leading to a tendency to generate more common predictions during training, thus making it difficult to handle complex prediction tasks, as illustrated in Figure 1.

To address these limitations, we propose the Retrieval-Augmented Time series Diffusion Model (RATD) for complex time series forecasting tasks. Our approach consists of two parts: the embedding-based retrieval and the reference-guided diffusion model. After obtaining a historical time series, it is input into the embedding-based retrieval process to retrieve the k nearest samples as references. The references are utilized as guidance in the denoising process. RATD focuses on making maximum utilization of existing time series datasets by finding the most relevant references in the dataset to the historical time series, thereby providing meaningful guidance for the denoising process. RATD focuses on maximizing the utilization of insufficient time series data and to some extent mitigates the issues caused by data imbalance. Meanwhile, this reference-guided mechanism also compensates for the deficiencies of guidance in existing time series diffusion models. Our approach demonstrates strong performance across multiple datasets, particularly on more complex tasks.

To summarize, our main contributions are summarized as follows:

- To handle complex time series forecasting, we for the first time introduce Retrieval-Augmented Time series Diffusion (RATD), allowing for greater utilization of the dataset and providing meaningful guidance in the denoising process.
- Extra Reference Modulated Attention (RMA) module is designed to provide reasonable guidance from the reference during the denoising process. RMA effectively simply integrates information without introducing excessive additional computational costs.
- We conducted experiments on five real-world datasets and provided a comprehensive presentation and analysis of the results using multiple metrics. The experimental results demonstrate that our approach achieves comparable or better results compared to baselines.

2 Related Work

2.1 Diffusion Models for Time Series Forecasting

Recent advancements have been made in the utilization of diffusion models for time series forecasting. In TimeGrad [28], the conditional diffusion model was first employed as an autoregressive approach for prediction, with the denoising process guided by the hidden state. CSDI [36] adopted a non-autoregressive generation strategy to achieve faster predictions. SSSD [1] replaced the noise-matching network with a structured state space model for prediction. TimeDiff [30] incorporated future mix-up and autoregressive initialization into a non-autoregressive framework for forecasting. MG-TSD [9] utilized a multi-scale generation strategy to sequentially predict the main components and details of the time series. Meanwhile, mr-diff [31] utilized diffusion models to separately predict the trend and seasonal components of time series. These methods have shown promising results in some prediction tasks, but they often perform poorly in challenging prediction tasks. We propose a retrieval-augmented framework to address this issue.

2.2 Retrieval-Augmented Generation

The retrieval-augmented mechanism is one of the classic mechanisms for generative models. Numerous works have demonstrated the benefits of incorporating explicit retrieval steps into neural networks. Classic works in the field of natural language processing leverage retrieval augmentation mechanisms to enhance the quality of language generation [16, 10, 4]. In the domain of image generation, some retrieval-augmented models focus on utilizing samples from the database to generate more realistic images [2, 44]. Similarly, [3] employed memorized similarity information from training data for retrieval during inference to enhance results. MQ-ReTCNN [40] is specifically designed for complex time series forecasting tasks involving multiple entities and variables. ReTime [13] creates a relation graph based on the temporal closeness between sequences and employs relational retrieval instead of content-based retrieval. Although the aforementioned three methods successfully utilize retrieval mechanisms to enhance time series forecasting results, our approach still holds significant advantages. This advantage stems from the iterative structure of the diffusion model, where references can repeatedly influence the generation process, allowing references to exert a stronger influence on the entire conditional generation process.

3 Preliminary

The forecasting task and the background knowledge about the conditional time series diffusion model will be discussed in this section. To avoid conflicts, we use the symbol “s” to represent the time series, and the “t” denotes the t-th step in the diffusion process.

Generative Time Series Forecasting. Suppose we have an observed historical time series $\mathbf{x}^H = \{s_1, s_2, \dots, s_l \mid s_i \in \mathbb{R}^d\}$, where l is the historical time length, d is the number of features per observation and s_i is the observation at time step i . The \mathbf{x}^P is the corresponding prediction target $\{s_{l+1}, s_{l+2}, \dots, s_{l+h} \mid s_{l+i} \in \mathbb{R}^d\}$ ($d' \leq d$), where h is the prediction horizon. The task of generative time series forecasting is to learn a density $p_\theta(\mathbf{x}^P \mid \mathbf{x}^H)$ that best approximates $p(\mathbf{x}^P \mid \mathbf{x}^H)$, which can be written as:

$$\min_{p_\theta} D(p_\theta(\mathbf{x}^P \mid \mathbf{x}^H) \parallel p(\mathbf{x}^P \mid \mathbf{x}^H)), \quad (1)$$

where θ denotes parameters and D is some appropriate measure of distance between distributions. Given observation x the target time series can be obtained directly by sampling from $p_\theta(\mathbf{x}^P \mid \mathbf{x}^H)$. Therefore, we obtain the time series $\{s_1, s_2, \dots, s_{n+h}\} = [\mathbf{x}^H, \mathbf{x}^P]$.

Conditional Time Series Diffusion Models. With observed time series \mathbf{x}^H , the diffusion model progressively deconstructs target time series \mathbf{x}_0^P (equals to the \mathbf{x}^P mentioned in the previous context) by injecting noise, then learns to reverse this process starting from \mathbf{x}_T^P for sample generation. For the convenience of expression, in this paper, we use \mathbf{x}_t to refer to the t-th time series in the diffusion process, with the letter “P” omitted. The forward process can be formulated as a Gaussian process with a Markovian structure:

$$\begin{aligned} q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) &:= \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \mathbf{x}^H, \beta_t \mathbf{I}), \\ q(\mathbf{x}_t \mid \mathbf{x}_0) &:= \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, \mathbf{x}^H, (1 - \bar{\alpha}_t) \mathbf{I}), \end{aligned} \quad (2)$$

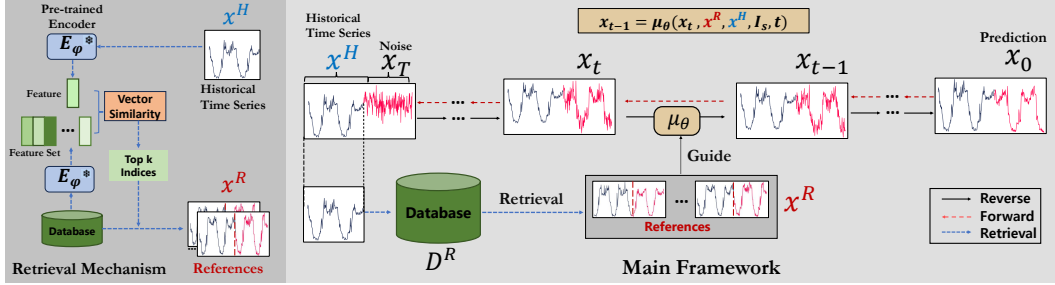


Figure 2: **Overview** of the proposed RATD. The historical time series x^H is inputted into the retrieval module to for the corresponding references x^R . After that, x^H is concatenated with the noise as the main input for the model μ_θ . x^R will be utilized as the guidance for the denoising process.

where β_1, \dots, β_T denotes fixed variance schedule with $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^t \alpha_s$. This forward process progressively injects noise into data until all structures are lost, which is well-approximated by $\mathcal{N}(0, I)$. The reverse diffusion process learns a model $p_\theta(x_{t-1}|x_t, x^H)$ that approximates the true posterior:

$$p_\theta(x_{t-1}|x_t, x^H) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t), \Sigma_\theta(x_t), x^H), \quad (3)$$

where μ_θ and Σ_θ are often computed by the Transformer. Ho *et al.* [12] improve the diffusion training process and optimize following objective:

$$\mathcal{L}(x_0) = \sum_{t=1}^T \mathbb{E}_{q(x_t|x_0|x^H)} \|\mu_\theta(x_t, t|x^H) - \hat{\mu}(x_t, x_0|x^H)\|^2, \quad (4)$$

where $\hat{\mu}(x_t, x_0|x^H)$ is the mean of the posterior $q(x_{t-1}|x_0, x_t)$ which is a closed form Gaussian, and $\mu_\theta(x_t, t|x^H)$ is the predicted mean of $p_\theta(x_{t-1}|x_t, x^H)$ computed by a neural network.

4 Method

We first describe the overall architecture of the proposed method in 4.1. Then we will introduce the strategy of building datasets in Section 4.2. The embedding-based retrieval mechanisms and reference-guided time series diffusion model are introduced in Section 4.3.

4.1 Framework Overview

Figure 2(a) shows the overall architecture of RATD. We built the entire process based on DiffWave [17], which combines the traditional diffusion model framework and a 2D transformer structure. In the forecasting task, RATD first retrieves motion sequences from the database base \mathcal{D}^R based on the input sequence of historical events. These retrieved samples are then fed into the Reference-Modulated Attention (RMA) as references. In the RMA layer, we integrate the features of the input $[x^H, x^t]$ at time step t with side information \mathcal{I}_s and the references x^R . Through this integration, the references guide the generation process. We will introduce these processes in the following subsections.

4.2 Constructing Retrieval Database for Time Series

Before retrieval, it is necessary to construct a proper database. We propose a strategy for constructing databases from time series datasets with different characteristics. Some time series datasets are size-insufficient and are difficult to annotate with a single category label (*e.g.*, electricity time series), while some datasets contain complete category labels but exhibit a significant degree of class imbalance (*e.g.*, medical time series). We use two different definitions of databases for these two different types of datasets. For the first definition, the entire training set is directly defined as the database \mathcal{D}^R :

$$\mathcal{D}^R := \{x_i | \forall x_i \in \mathcal{D}^{\text{train}}\} \quad (5)$$

where $x_i = \{s_i, \dots, s_{i+l+h}\}$ is the time series with length $l + h$, and $\mathcal{D}^{\text{train}}$ is the training set. In the second way, the subset containing samples from all categories in the dataset is defined as the database

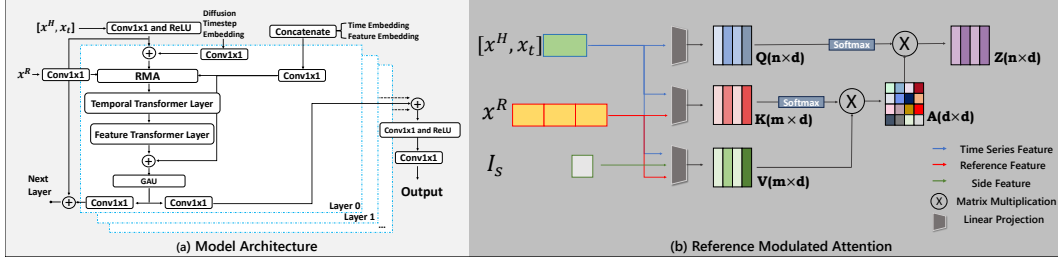


Figure 3: The structure of μ_θ . (a) The main architecture of μ_θ is the time series transformer structure that proved effective. (b) The structure of the proposed RMA. We integrate three different features through matrix multiplication.

$\mathcal{D}^{R'}$:

$$\mathcal{D}^{R'} = \{\mathbf{x}_i^c, \dots, \mathbf{x}_q^c | \forall c \in \mathcal{C}\} \quad (6)$$

where \mathbf{x}_i^k is the i -th sample in the k -th class of the training set, with a length of $l+h$. \mathcal{C} is the category set of the original dataset. For brevity, we represent both databases as \mathcal{D}^R .

4.3 Retrieval-Augmented Time Series Diffusion

Embedding-Based Retrieval Mechanism For time forecasting tasks, the ideal references $\{s_i, \dots, s_{i+h}\}$ would be samples where preceding n points $\{s_{i-n}, \dots, s_{i-1}\}$ is most relevant to the historical time series $\{s_j, \dots, s_{j+n}\}$ in the \mathcal{D}^R . In our approach, the overall similarity between time series is of greater concern. We quantify the reference between time series using the distance between their embeddings. To ensure that embeddings can effectively represent the entire time series, pre-trained encoders E_ϕ are utilized. E_ϕ is trained on representation learning tasks, and the parameter set ϕ is frozen in our retrieval mechanism. For time series (with length $n+h$) in \mathcal{D}^R , their first n points are encoded, thus the \mathcal{D}^R can be represented as $\mathcal{D}_{\text{emb}}^R$:

$$\mathcal{D}_{\text{emb}}^R = \{\{i, E_\phi(\mathbf{x}_{[0:n]}^i), \mathbf{x}_{[n:n+h]}^i\} | \forall \mathbf{x}^i \in \mathcal{D}^R\} \quad (7)$$

where $[p:q]$ refers to the subsequence formed by the p -th point to the q -th point in the time series. The embedding corresponding to the historical time series can be represented as $\mathbf{v}^H = E_\phi(\mathbf{x}^H)$. We calculate the distances between \mathbf{v}^H and all embeddings in $\mathcal{D}_{\text{emb}}^R$ and retrieve the references corresponding to the k smallest distances. This process can be expressed as:

$$\text{index}(\mathbf{v}^H) = \arg \min_{\mathbf{x}^i \in \mathcal{D}_{\text{emb}}^R} \|\mathbf{v}^H - E_\phi(\mathbf{x}_{[0:n]}^i)\|^2 \quad (8)$$

$$\mathbf{x}^R = \{\mathbf{x}_{[n:n+h]}^j | \forall j \in \text{index}(\mathbf{v}^H)\}$$

where $\text{index}(\cdot)$ represents retrieved index given \mathbf{v}_D . Thus, we obtain a subset \mathbf{x}^R of \mathcal{D}^R based on a query \mathbf{x}^H , i.e. $\zeta_k: \mathbf{x}^H, \mathcal{D}^R \rightarrow \mathbf{x}^R$, where $|\mathbf{x}^R| = k$.

Reference-Guided Time Series Diffusion Model In this section, we will introduce our reference-guided time series diffusion model. In the diffusion process, the forward process is identical to the traditional diffusion process, as shown in Equation (2). Following [34, 12, 35] the objective of the reverse process is to infer the posterior distribution $p(\mathbf{z}^{\text{tar}} | \mathbf{z}^c)$ through the subsequent expression:

$$p(\mathbf{x} | \mathbf{x}^H) = \int p(\mathbf{x}_T | \mathbf{x}^H) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}^H, \mathbf{x}^R) \mathcal{D}\mathbf{x}_{1:T}, \quad (9)$$

where $p(\mathbf{x}_T | \mathbf{x}^H) \approx \mathcal{N}(\mathbf{x}_T | \mathbf{x}^H, \mathbf{I})$, $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}^H, \mathbf{x}^R)$ is the reverse transition kernel from \mathbf{x}_t to \mathbf{x}_{t-1} with a learnable parameter θ . Following most of the literature in the diffusion model, we adopt the assumption:

$$p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t, \mathbf{x}) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, \mathbf{x}^H, \mathbf{x}^R, t), \Sigma_\theta(\mathbf{x}_t, \mathbf{x}^H, \mathbf{x}^R, t)) \quad (10)$$

where μ_θ is a deep neural network with parameter θ . After similar computations as those in [12], $\Sigma_\theta(\mathbf{x}_t, \mathbf{x}^H, \mathbf{x}^R, t)$ in the backward process is approximated as fixed. In other words, we can achieve reference-guided denoising by designing a rational and robust μ_θ .

Denoising Network Architecture Similar to DiffWave [17] and CSDI [36], our pipeline is constructed on the foundation of transformer layers, as shown in Figure 3. However, the existing framework cannot effectively utilize the reference as guidance. Considering attention modules to integrate the \mathbf{x}^R and \mathbf{x}_t as a reasonable intuition, we propose a novel module called Reference Modulated Attention (RMA). Unlike normal attention modules, we realize the fusion of three features in RMA: the current time series feature, the side feature, and the reference feature. To be specific, RMA was set at the beginning of each residual module Figure 3. We use 1D-CNN to extract features from the input \mathbf{x}_t , references \mathbf{x}^R , and side information. Notably, we concatenate all references together for feature extraction. Side information consists of two parts, representing the correlation between variables and time steps in the current time series dataset Appendix B. We adjust the dimensions of these three features with linear layers and fuse them through matrix dot products. Similar to text-image diffusion models [29], RMA can effectively utilize reference information to guide the denoising process, while appropriate parameter settings prevent the results from overly depending on the reference.

Training Procedure To train RATD (*i.e.*, optimize the evidence lower bound induced by RATD), we use the same objective function as previous work. The loss at time step $t - 1$ are defined as follows respectively:

$$\begin{aligned} L_{t-1}^{(x)} &= \frac{1}{2\hat{\beta}_t^2} \|\mu_\theta(\mathbf{x}_t, \hat{\mathbf{x}}_0) - \hat{\mu}(\mathbf{x}_t, \hat{\mathbf{x}}_0)\|^2 \\ &= \gamma_t \|\mathbf{x}_0 - \hat{\mathbf{x}}_0\| \end{aligned} \quad (11)$$

where $\hat{\mathbf{x}}_0$ are predicted from \mathbf{x}_t , and $\gamma_t = \frac{\bar{\alpha}_{t-1}\beta_t^2}{2\hat{\beta}_t^2(1-\bar{\alpha}_t)^2}$ are hyperparameters in diffusion process. We summarize the training procedure of RATD in Algorithm 1 and highlight the differences from the conventional models, in cyan. The process of sampling is shown in Appendix A.

Algorithm 1 Training Procedure of RATD

Require: Time series dataset $\mathcal{D}^{\text{train}}$, neural network μ_θ , , diffusion step T , external database \mathcal{D}^R , pre-trained encoder E_ϕ , number of references k

- 1: Retrieve references with top- k high similarity from \mathcal{D}^R using E to obtain \mathbf{x}^R as described in Section 4.3
 - 2: **while** ϕ_θ not converge **do**
 - 3: Sample diffusion time $t \in \mathcal{U}(0, \dots, T)$
 - 4: Compute the side feature \mathcal{I}_s
 - 5: Perturb \mathbf{x}_0 to obtain \mathbf{x}_t
 - 6: Predict $\hat{\mathbf{x}}_0$ from \mathbf{x}_t , \mathcal{I}_s and \mathbf{x}^R (Equation (10))
 - 7: Compute loss L with $\hat{\mathbf{x}}_0$ and \mathbf{x}_0 (Equation (11))
 - 8: Update θ by minimizing L
 - 9: **end while**
-

5 Experiments

5.1 Experimental Setup

Datasets Following previous work [45, 38, 8, 30], experiments are performed on four popular real-world time series datasets: (1) *Electricity*^{*}, which includes the hourly electricity consumption data from 321 clients over two years.; (2) *Wind* [20], which contains wind power records from 2020-2021. (3) *Exchange* [18], which describes the daily exchange rates of eight countries (Australia, British, Canada, Switzerland, China, Japan, New Zealand, and Singapore); (4) *Weather*[†], which documents 21 meteorological indicators at 10-minute intervals spanning from 2020 to 2021.; Besides, we also applied our method to a large ECG time series dataset: MIMIC-IV-ECG [14]. The MIMIC-IV-ECG dataset contains clinical electrocardiogram data from over 190,000 patients and 450,000 hospitalizations at Beth Israel Deaconess Medical Center (BIDMC).

^{*}<https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014>

[†]<https://www.bgc-jena.mpg.de/wetter/>

Table 1: Performance comparisons on four real-world datasets in terms of MSE, MAE, and CRPS. The best is in bold, while the second best is underlined.

Dataset	Exchange			Wind			Electricity			Weather		
Metric	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS
RATD (ours)	0.013	0.073	0.339	0.784	0.579	0.673	<u>0.151</u>	<u>0.246</u>	0.373	0.281	0.293	0.301
TimeDiff	0.018	0.091	0.589	0.896	0.687	0.917	0.193	0.305	0.490	0.327	<u>0.312</u>	0.410
CSDI	0.077	0.194	0.397	1.066	0.741	0.941	0.379	0.579	0.480	0.356	0.374	0.354
mr-Diff	<u>0.016</u>	0.082	0.397	<u>0.881</u>	0.675	0.881	0.173	0.258	0.429	0.296	0.324	0.347
D ₃ VAE	0.200	0.301	0.401	1.118	0.779	0.979	0.286	0.372	<u>0.389</u>	0.315	0.380	0.381
Fedformer	0.133	0.233	0.631	1.113	0.762	1.235	0.238	0.341	0.561	0.342	0.347	0.319
FreTS	0.039	0.140	0.440	1.004	0.703	0.943	0.269	0.371	0.634	0.351	0.354	0.391
FiLM	<u>0.016</u>	0.079	0.349	0.984	0.717	0.798	0.210	0.320	0.671	0.327	0.336	0.556
iTransformer	<u>0.016</u>	<u>0.074</u>	<u>0.343</u>	0.932	0.676	<u>0.811</u>	0.192	0.262	0.402	0.358	0.401	<u>0.318</u>
Autoformer	0.056	0.167	0.769	1.083	0.756	1.201	1.026	0.313	0.602	0.360	0.354	0.754
Pyraformer	0.032	0.112	0.532	1.061	0.735	0.994	0.273	0.379	0.732	0.394	0.385	0.485
Informer	0.073	0.192	0.631	1.168	0.772	1.065	0.292	0.383	0.749	0.385	0.364	0.821
PatchTST	0.047	0.153	0.629	1.001	<u>0.672</u>	1.026	0.225	0.394	0.801	0.782	0.670	0.370
SCINet	0.038	0.137	0.624	1.055	0.732	0.997	0.171	0.280	0.499	0.329	0.344	0.814
DLinear	0.022	0.102	0.538	0.899	0.686	0.957	0.215	0.336	0.527	0.488	0.444	0.791
NLinear	0.019	0.091	0.481	0.989	0.706	0.974	0.147	0.239	0.419	0.369	0.328	0.738
TimesNet	0.023	0.120	0.520	0.982	0.771	1.001	0.141	0.361	0.403	<u>0.313</u>	0.364	0.491
NBeats	<u>0.016</u>	0.081	0.399	1.069	0.741	0.981	0.269	0.370	0.697	0.744	0.420	0.871

Baseline Methods To comprehensively demonstrate the effectiveness of our method, we compare RATD with four kinds of time series forecasting methods. Our baselines include (1) Time series diffusion models, including CSDI [36], mr-Diff [31], D³VAE [20], TimeDiff [30]; (2) Recent time series forecasting methods with frequency information, including FiLM [46], Fedformer [47] and FreTS [41]; (3) Time series transformers, including PatchTST [25], Autoformer [38], Pyraformer [22], Informer [45] and iTransformer [23]; (4) Other popular methods, including TimesNet [39], SciNet [21], NLinear [43], DLinear [43] and NBeats [26].

Evaluation Metric To comprehensively assess our proposed methodology, our experiment employs three metrics: (1) Probabilistic forecasting metrics: Continuous Ranked Probability Score (CRPS) on each time series dimension [24]. (2) Distance metrics: Mean Squared Error (MSE), and Mean Average Error (MAE) are employed to measure the distance between predictions and ground truths.

Implementation Details The length of the historical time series was 168, and the prediction lengths were (96, 192, 336), with results averaged. All experiments were conducted on an Nvidia RTX A6000 GPU with 40GB memory. During the experiments, the second strategy of conducting \mathcal{D}^R was employed for the MIMIC dataset, while the first strategy was utilized for the other four datasets. To reduce the training cost, we preprocessed the retrieval process by storing the reference indices of each sample in the training set in a dictionary. During the training on the diffusion model, we accessed this dictionary directly to avoid redundant retrieval processes. More details are shown in Appendix B.

5.2 Main Results

Table 1 presents the primary results of our experiments on four daily datasets. Our approach surpasses existing time series diffusion models. Compared to other time series forecasting methods, our approach exhibits superior performance on three out of four datasets, with competitive performance on the remaining dataset. Notably, we achieve outstanding results on the wind dataset. Due to the lack of clear short-term periodicity (daily or hourly), some prediction tasks in this dataset are exceedingly challenging for other models. Retrieval-augmented mechanisms can effectively assist in addressing these challenging prediction tasks.

Figure 4 presents a case study randomly selected from our experiments on the wind dataset. We compare our prediction with iTransformer and two popular open-source time series diffusion models, CSDI and D₃VAE. Although CSDI and D₃VAE provide accurate predictions in the initial short-term period, their long-term predictions deviate significantly from the ground truth due to the lack of guidance. ITransformer captures rough trends and periodic patterns, yet our method offers higher-quality predictions than the others. Furthermore, through the comparison between the predicted

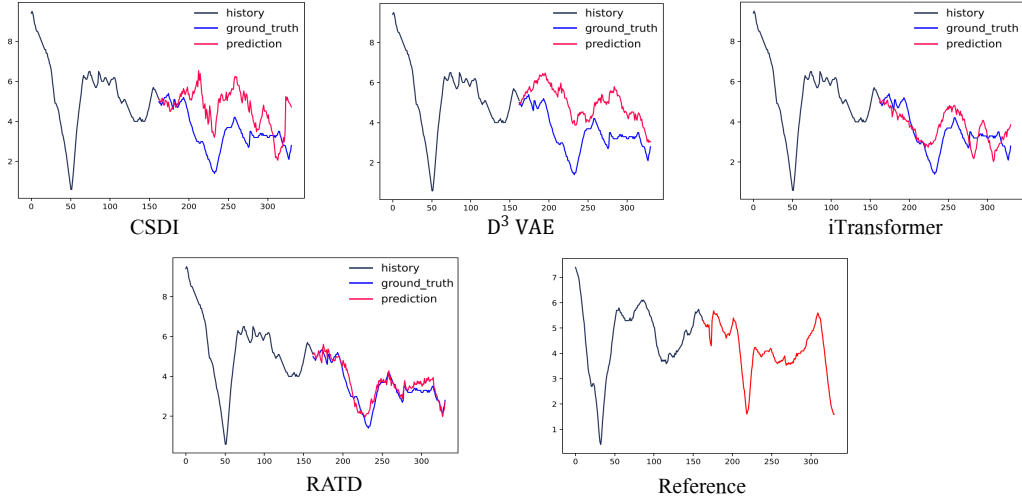


Figure 4: Visualizations on *wind* by CSDI, D_3 VAE, iTransformer and the proposed RATD (with reference).

results and references in the figure, although references provide strong guidance, they do not explicitly substitute for the entire generated results. This further validates the rationality of our approach.

Table 2 presents the testing results of our method on the MIMIC-IV-ECG dataset. We selected some powerful open-source methods as baselines for comparison. Our experiments are divided into two parts: in the first part, we evaluate the entire test set, while in the second part, we select rare cases (those accounting for less than 2% of total cases) from the test set as a subset for evaluation. Prediction tasks in the second part are more challenging for deep models. In the first experiment, our method achieved results close to iTransformer, while in the second task, our model significantly outperformed other methods, demonstrating the effectiveness of our approach in addressing challenging tasks.

5.3 Model Analysis

Influence of Retrieval Mechanism To investigate the impact of the retrieval augmentation mechanism on the generation process, we conducted an ablation study and presented the results in Table 3. The study addresses two questions: whether the retrieval augmentation mechanism is effective and which retrieval method is most effective. Firstly, we removed our retrieval augmentation mechanism from the RATD as a baseline. Besides, the model with random time series guidance is another baseline. The references retrieved by other methods have all positively impacted the prediction results. This suggests that reasonable references are highly effective in guiding the generation process.

We also compared two different retrieval mechanisms: correlation-based retrieval and embedding-based retrieval. The first method directly retrieves the reference in the time domain (*e.g.*, using Dynamic Time Warping (DTW) or Pearson correlation coefficient). Our approach adopts the second mechanism: retrieving references through the embedding of time series. From the results, the correlation-based methods are significantly inferior to the embedding-based methods. The former methods fail to capture the key features of the time series, making it difficult to retrieve the best references for forecasting. We also evaluate the embedding-based methods with various encoders for comparison. The comprehensive results show that methods with different encoders do not significantly differ. This indicates that different methods can all extract meaningful references, thereby producing similar improvements in results. TCN was utilized in our experiment because TCN strikes the best balance between computational cost and performance.

Effect of Retrieval Database We conducted an ablation study on two variables, n and k , to investigate the influence of the retrieval database \mathcal{D}^R in RATD, where n represents the number of samples in each category of the database, and k represents the number of reference exemplars. The results in Figure 5q can benefit the model in terms of prediction accuracy because a larger \mathcal{D}^R brings higher diversity, thereby providing more details beneficial for prediction and enhancing the generation

Table 2: Performance comparisons on MIMIC datasets with popular time series forecasting methods. Here, "MIMIC-IV (All)" refers to the model’s testing results on the complete test set, while "MIMIC(Rare)" indicates the model’s testing results on a rare disease subset.

Method	iTransformer			PatchTST			TimesNet			CSDI			RATD		
Metric	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS
MIMIC-IV (All)	<u>0.174</u>	0.263	<u>0.299</u>	0.219	0.301	0.307	0.193	0.311	0.310	0.268	0.331	0.369	0.172	<u>0.270</u>	0.293
MIMIC-IV (Rare)	<u>0.423</u>	<u>0.315</u>	<u>0.379</u>	0.483	0.379	0.407	0.627	0.359	0.464	0.499	0.359	<u>0.374</u>	0.206	0.299	0.301

Table 3: Ablation study on different retrieval mechanisms. "-" means no references was utilized and "Random" means references are selected randomly. Others refer to what model we use for retrieval references.

Dataset	Exchange			Wind			Electricity			Weather		
Metric	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS
-	0.077	0.194	0.397	1.066	0.741	0.941	0.379	0.579	0.480	0.356	0.374	0.354
Random	0.153	0.203	0.599	1.593	0.903	0.996	0.471	0.639	0.701	0.431	0.473	0.461
DTW	0.075	0.195	0.403	1.073	0.791	0.942	0.357	0.564	0.449	0.361	0.375	0.356
Pearson	0.091	0.207	0.411	1.099	0.831	0.953	0.361	0.571	0.483	0.370	0.364	0.391
DLinear	0.022	0.081	0.361	0.941	0.735	0.895	0.159	0.250	0.390	0.297	0.304	0.332
Informer	<u>0.019</u>	0.078	0.371	0.841	0.645	0.861	0.170	0.263	0.411	0.291	0.305	0.330
TimesNet	0.013	<u>0.074</u>	<u>0.341</u>	0.781	0.572	0.669	<u>0.167</u>	0.263	0.397	<u>0.286</u>	<u>0.295</u>	0.311
TCN	0.013	0.073	0.339	<u>0.784</u>	<u>0.579</u>	<u>0.673</u>	0.161	<u>0.256</u>	<u>0.391</u>	0.281	0.293	<u>0.313</u>

process. Simply increasing k does not show significant improvement, as utilizing more references may introduce more noise into the denoising process. In our experiment, the settings of n and k are 256 and 3, respectively.

Inference Efficiency In this experiment, we evaluate the inference efficiency of the proposed RATD in comparison to other baseline time series diffusion models (TimeGrad, MG-TSD, SSSD). Figure 6 illustrates the inference time on the multivariate *weather* dataset with varying values of the prediction horizon (h). While our method introduces an additional retrieval module, the sampling efficiency of the RATD is not low due to the non-autoregressive transformer framework. It even slightly outperforms other baselines across all h values. Notably, TimeGrad is observed to be the slowest, attributed to its utilization of auto-regressive decoding.

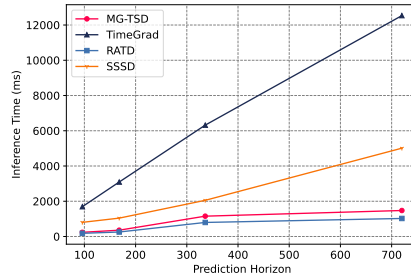
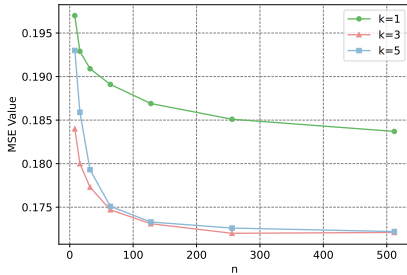


Figure 5: The effect of hyper-parameter n and k . Figure 6: Inference time (ms) on the Electricity with different prediction horizon h

Effectiveness of Reference Modulated Attention To validate the effectiveness of the proposed RMA, we designed additional ablation experiments. In these experiments, we used the CSDI architecture as the baseline method and added extra fusion modules to compare the performance of these modules (linear layer, cross-attention layer, and RMA). The results are shown in the Table 4.

Through our experiments, we found that compared to the basic cross-attention-based approach, RMA can integrate an edge information matrix (representing correlations between time and feature dimensions) more effectively. The extra fusion is highly beneficial in experiments, guiding the model to capture relationships between different variables. In contrast, linear-based methods concatenate inputs and references initially, which prevents the direct extraction of meaningful information from references, resulting in comparatively modest performance.

Table 4: Performance comparison(MSE) between CSDI-based methods, CSDI represents the basic network framework, CSDI+Linear denotes the approach where inputs and references are concatenated via a linear layer and fed into the network together, CSDI+CrossAttention signifies the use of cross attention to fuse features from inputs and references, and finally, CSDI+RMA, which incorporates an additional RMA.

Dataset	Exchange	Electricity	Wind	Weather	Solar	MIMIC-IV
CSDI	0.077	0.379	1.066	0.356	0.381	0.268
CSDI+Linear	0.075	0.316	0.932	0.349	0.369	0.265
CSDI+Cross Attention	0.028	0.173	0.829	0.291	0.340	0.183
CSDI+RMA	0.013	0.151	0.784	0.281	0.327	0.172

Predicting x_0 vs Predicting ϵ . Following the formulation in Section 4.3, our network is designed to forecast the latent variable x_0 . Since some existing models [28, 36] have been trained by predicting an additional noise term ϵ , we conducted a comparative experiment to determine which approach is more suitable for our framework. Specifically, we maintained the network structure unchanged, only modifying the prediction target to be ϵ . The results are presented in Table 5. Predicting x_0 proves to be more effective. This may be because the relationship between the reference and x_0 is more direct, making the denoising task relatively easier.

Table 5: MSEs of two denoising strategies: Predicting x_0 vs predicting ϵ .

denoising strategy	Wind	Weather	Exchange
x_0	0.784	0.281	0.013
ϵ	0.841	0.331	0.018

RMA position We investigate the best position of RMA in the model. Front, middle, and back means we set the RMA in the front of, in the middle of, and the back of two transformer layers, respectively. We found that placing RMA before the bidirectional transformer resulted in the most significant improvement in model performance. This also aligns with the intuition of network design: cross-attention modules placed at the front of the model tend to have a greater impact.

Table 6: Ablation study on different RMA positions. The best is in bold.

Dataset	Exchange			Wind			Electricity			Weather		
	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS
-	0.077	0.194	0.397	1.066	0.741	0.941	0.379	0.579	0.480	0.356	0.374	0.354
Back	0.031	0.105	0.373	0.673	0.611	0.842	0.267	0.434	0.426	0.301	0.321	0.322
Middle	0.057	0.141	0.381	0.799	0.631	0.833	0.291	0.481	0.451	0.333	0.331	0.336
Front	0.013	0.063	0.331	0.784	0.579	0.673	0.161	0.256	0.391	0.281	0.293	0.313

6 Discussion

Limitation and Future Work As a transformer-based diffusion model structure, our approach still faces some challenges brought by the transformer framework. Our model consumes a significant amount of computational resources dealing with time series consisting of too many variables. Additionally, our approach requires additional preprocessing (retrieval process) during training, which incurs additional costs on training time (around ten hours).

Conclusion In this paper, we propose a new framework for time series diffusion modeling to address the forecasting performance limitations of existing diffusion models. RATD retrieves samples most relevant to the historical time series from the constructed database and utilize them as references to guide the denoising process of the diffusion model, thereby obtaining more accurate predictions. RATD is highly effective in solving challenging time series prediction tasks, as evaluated by experiments on five real-world datasets.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.62172018, No.62102008) and Wuhan East Lake High-Tech Development Zone National Comprehensive Experimental Base for Governance of Intelligent Society.

References

- [1] Juan Miguel Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *arXiv preprint arXiv:2208.09399*, 2022.
- [2] Andreas Blattmann, Robin Rombach, Kaan Oktay, Jonas Müller, and Björn Ommer. Retrieval-augmented diffusion models. *Advances in Neural Information Processing Systems*, 35:15309–15324, 2022.
- [3] Giovanni Bonetta, Rossella Cancelliere, Ding Liu, and Paul Vozila. Retrieval-augmented transformer-xl for close-domain dialog generation. *arXiv preprint arXiv:2105.09235*, 2021.
- [4] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pp. 2206–2240. PMLR, 2022.
- [5] Jian Cao, Zhi Li, and Jian Li. Financial time series forecasting model based on ceemdan and lstm. *Physica A: Statistical mechanics and its applications*, 519:127–139, 2019.
- [6] Jui-Sheng Chou and Duc-Son Tran. Forecasting energy consumption time series using machine learning techniques based on usage patterns of residential householders. *Energy*, 165:709–726, 2018.
- [7] Alexiei Dingli and Karl Sant Fournier. Financial time series forecasting—a deep learning approach. *International Journal of Machine Learning and Computing*, 7(5):118–122, 2017.
- [8] Wei Fan, Shun Zheng, Xiaohan Yi, Wei Cao, Yanjie Fu, Jiang Bian, and Tie-Yan Liu. Depts: deep expansion learning for periodic time series forecasting. *arXiv preprint arXiv:2203.07681*, 2022.
- [9] Xinyao Fan, Yueying Wu, Chang Xu, Yuhao Huang, Weiqing Liu, and Jiang Bian. Mg-tds: Multi-granularity time series diffusion models with guided learning process. *arXiv preprint arXiv:2403.05751*, 2024.
- [10] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.
- [11] Pradeep Hewage, Ardhendu Behera, Marcello Trovati, Ella Pereira, Morteza Ghahremani, Francesco Palmieri, and Yonghuai Liu. Temporal convolutional neural (tcn) network for an effective weather forecasting using time-series data from the local weather station. *Soft Computing*, 24:16453–16482, 2020.
- [12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [13] Baoyu Jing, Si Zhang, Yada Zhu, Bin Peng, Kaiyu Guan, Andrew Margenot, and Hanghang Tong. Retrieval based time series forecasting. *arXiv preprint arXiv:2209.13525*, 2022.
- [14] Alistair EW Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, et al. Mimic-iv, a freely accessible electronic health record dataset. *Scientific data*, 10(1):1, 2023.
- [15] Zahra Karevan and Johan AK Suykens. Transductive lstm for time-series prediction: An application to weather forecasting. *Neural Networks*, 125:1–9, 2020.
- [16] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*, 2019.
- [17] 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*, 2020.

- [18] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.
- [19] S Sri Lakshmi and RK Tiwari. Model dissection from earthquake time series: A comparative analysis using modern non-linear forecasting and artificial neural network approaches. *Computers & Geosciences*, 35(2):191–204, 2009.
- [20] Yan Li, Xinjiang Lu, Yaqing Wang, and Dejing Dou. Generative time series forecasting with diffusion, denoise, and disentanglement. *Advances in Neural Information Processing Systems*, 35:23009–23022, 2022.
- [21] Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. Scinet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35:5816–5828, 2022.
- [22] Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International conference on learning representations*, 2021.
- [23] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.
- [24] James E Matheson and Robert L Winkler. Scoring rules for continuous probability distributions. *Management science*, 22(10):1087–1096, 1976.
- [25] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.
- [26] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.
- [27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
- [28] Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, pp. 8857–8868. PMLR, 2021.
- [29] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.
- [30] Lifeng Shen and James Kwok. Non-autoregressive conditional diffusion models for time series prediction. *arXiv preprint arXiv:2306.05043*, 2023.
- [31] Lifeng Shen, Weiyu Chen, and James Kwok. Multi-resolution diffusion models for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2023.
- [32] Zhipeng Shen, Yuanming Zhang, Jiawei Lu, Jun Xu, and Gang Xiao. Seriesnet: a generative time series forecasting model. In *2018 international joint conference on neural networks (IJCNN)*, pp. 1–8. IEEE, 2018.
- [33] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 3531–3539, 2021.
- [34] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

- [35] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.
- [36] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csd: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021.
- [37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [38] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.
- [39] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *The eleventh international conference on learning representations*, 2022.
- [40] Sitan Yang, Carson Eisenach, and Dhruv Madeka. Mqretnn: Multi-horizon time series forecasting with retrieval augmentation. *arXiv preprint arXiv:2207.10517*, 2022.
- [41] Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain mlps are more effective learners in time series forecasting. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [42] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32, 2019.
- [43] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.
- [44] Mingyuan Zhang, Xinying Guo, Liang Pan, Zhongang Cai, Fangzhou Hong, Huirong Li, Lei Yang, and Ziwei Liu. Remodiffuse: Retrieval-augmented motion diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 364–373, 2023.
- [45] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.
- [46] Tian Zhou, Ziqing Ma, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, Rong Jin, et al. Film: Frequency improved legendre memory model for long-term time series forecasting. *Advances in Neural Information Processing Systems*, 35:12677–12690, 2022.
- [47] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, pp. 27268–27286. PMLR, 2022.

A Sampling Procedure

Like Algorithm 1, we summarize the sampling procedure of RATD in Algorithm 2 and highlight the differences from conventional diffusion models in .

Algorithm 2 Sampling Procedure of RATD

Require: The historical time series \mathbf{x}^H , the learned model μ_θ , external database \mathcal{D}^R , pre-trained E_ϕ , the number of references k

Ensure: Prediction \mathbf{x}^P corresponding to the history \mathbf{x}^H

- 1: Sample initial target time series \mathbf{x}_T
 - 2: Embed \mathbf{x}^H into \mathbf{v}^H
 - 3: Retrieval the reference \mathbf{x}^R with \mathbf{v}^H
 - 4: Compute the side feature \mathcal{I}_s
 - 5: **for** t in $T, T - 1, \dots, 1$ **do**
 - 6: Predict $\hat{\mathbf{x}}_0$ from $\mathbf{x}_t, \mathcal{I}_s$ and \mathbf{x}^R (Equation (10))
 - 7: Sample \mathbf{x}_{t-1} from the posterior $q(\mathbf{x}_t|\mathbf{x}_0)$ (Equation (2))
 - 8: **end for**
-

B Impletion Details

B.1 Training Details

Our dataset is split in the proportion of 7:1:2 (Train: Validation: Test), utilizing a random splitting strategy to ensure diversity in the training set. We sample the ECG signals at 125Hz for the MIMIC-IV dataset and extract fixed-length windows as samples. For training, we utilized the Adam optimizer with an initial learning rate of 10^{-3} , $\text{betas} = (0.95, 0.999)$. During the training process of shifted diffusion, the batch size was set to 64, and early stopping was applied for a maximum of 200 epochs. The diffusion steps T were set to 100.

B.2 Side Information

We combine temporal embedding and feature embedding as side information \mathbf{v}_s . We use 128-dimensions temporal embedding following previous studies [37]:

$$s_{embedding}(s_c) = \left(\sin(s_c/\tau^{0/64}), \dots, \sin(s_c/\tau^{63/64}), \cos(s_c/\tau^{0/64}), \dots, \cos(s_c/\tau^{63/64}) \right) \quad (12)$$

where $\tau = 10000$. Following [36], s_l represents the timestamp corresponding to the l -th point in the time series. This setup is designed to capture the irregular sampling in the dataset and convey it to the model. Additionally, we utilize learnable embedding to handle feature dimensions. Specifically, feature embedding is represented as 16-dimensional learnable vectors that capture relationships between dimensions. According to [17], we combine time embedding and feature embedding, collectively referred to as side information \mathcal{I}_s .

The shape of \mathcal{I}_s is not fixed and varies with datasets. Taking the Exchange dataset as an example, the shape of forecasting target \mathbf{x}^R is [Batchsize (64), 7(number of variables), 168 (time-dimension), 12 (time-dimension)] and the corresponding shape of \mathcal{I}_s is [Batchsize (64), total channel(144(time:128 + feature:16)), 320 (frequency-dimension*latent channel), 12 (time-dimension)].

B.3 Transformers Details

Our approach employs the Transformer architecture from CSDI, with the distinction of expanding the channel dimension to 128. The network comprises temporal and feature layers, ensuring the comprehensiveness of the model in handling the time-frequency domain latent while maintaining a relatively simple structure. Regarding the transformer layer, we utilized a 1-layer Transformer encoder implemented in PyTorch [27], comprising multi-head attention layers, fully connected layers, and layer normalization. We adopted the "linear attention transformer" package[‡], to enhance

[‡]<https://github.com/lucidrains/linear-attention-transformer>

computational efficiency. The inclusion of numerous features and long sequences prompted this decision. The package implements an efficient attention mechanism [33], and we exclusively utilized the global attention feature within the package.

B.4 Metrics

We will introduce the metrics in our experiments. We summarize them as below:

CRPS. CRPS [24] is a univariate strictly proper scoring rule which $\hat{\cdot}$ measures the compatibility of a cumulative distribution function F with an observation x as:

$$CRPS(F, x) = \int_R (F(y) - \mathbb{1}_{(x \leq y)})^2 dy \quad (13)$$

where $\mathbb{1}_{(x \leq y)}$ is the indicator function, which is 1 if $x \leq y$ and 0 otherwise. The CRPS attains the minimum value when the predictive distribution F same as the data distribution.

MAE and MSE. MAE and MSE are calculated in the formula below, $\hat{\mathbf{x}}^P$ represents the predicted time series, and \mathbf{x}^P represents the ground truth time series. MAE calculates the average absolute difference between predictions and true values, while MSE calculates the average squared difference between predictions and true values. A smaller MAE or MSE implies better predictions.

$$\begin{aligned} MAE &= mean(|\hat{\mathbf{x}}^P - \mathbf{x}^P|) \\ MSE &= \sqrt{mean(|\hat{\mathbf{x}}^P - \mathbf{x}^P|^2)} \end{aligned} \quad (14)$$

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: The abstract and introduction include the paper's contributions and scope.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discuss the limitations of our approach at Section 6

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: The paper does not include theoretical results

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide all the information needed in the Experiment part.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The full code will be uploaded in the future.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide all the details about the dataset in appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We do not report the error bars in our experiemnt.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The paper provide sufficient information on the computer resources in paragraph implement details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research conducted in the paper conform with the NeurIPS Code of Ethics

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: There is no obvious societal impact of the work performed

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All the original work are cited properly.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset’s creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: The paper does not release new assets

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.