RAF−DIFF: RATIONAL ADAPTIVE FUNCTIONAL DIFFUSION FOR TIME SERIES ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series analysis tasks such as forecasting, imputation, anomaly detection, and classification are crucial for applications spanning climate science, finance, retail, and cloud infrastructure. We present Rational Adaptive Functional Diffusion (RAF−Diff), a conditional diffusion model that introduces two innovations: (i) a Rational Adaptive Correlation Projection (RACP) in the forward process and (ii) a Rational Denoising Transformer (R-DiT) architecture, specifically designed for time series modeling. Unlike traditional fixed Gaussian noise based forward process, our RACP module makes the forward process trainable and data-adaptive, capturing complex inter-feature correlations through learned rational function transformations. Our R-DiT architecture with its rational function layers provides superior expressiveness for capturing periodic time series signals. RAF−Diff achieves superior performance across four fundamental time series analysis tasks, significantly outperforming existing prominent models in forecasting (eight datasets), imputation (six datasets), classification (ten datasets) and anomaly detection (five datasets). Comprehensive ablation studies across all tasks validate the utility of each RAF−Diff component, demonstrating the model's robustness in diverse time series challenges.

1 INTRODUCTION

Time series analysis stands at the heart of countless critical applications across diverse domains. From financial market forecasting (Kim, 2003) and economic modeling (Henrique et al., 2019) to transportation planning (Huang et al., 2023), energy management (Dumas et al., 2022), and climate prediction (Li et al., 2024a; Rasul et al., 2022), the ability to understand and predict time series patterns drives decision-making in our most vital systems. Beyond forecasting, critical tasks include missing data imputation (Friedman, 1962; Qu et al., 2024; Zhan et al., 2021), anomaly detection for industrial maintenance (Xu et al., 2021), and classification for applications such as trajectory-based action recognition (Franceschi et al., 2019).

The field has evolved from conventional statistical methods such as ARIMA and state-space models to embrace deep neural architectures including NLinear (Zeng et al., 2023), recurrent networks (Hewamalage et al., 2021), convolutional networks (Yue et al., 2022), and transformers (Vaswani et al., 2017). Most recently, diffusion models have emerged as a powerful paradigm, achieving unprecedented performance in image synthesis (Ho et al., 2020; Dhariwal & Nichol, 2021b), video generation (Harvey et al., 2022; Blattmann et al., 2023), and multi-modal applications (Saharia et al., 2022). This success has sparked intense interest in applying diffusion to time series, leading to frameworks such as TimeDiff (Shen & Kwok, 2023) with future mixup conditioning, TimeGrad (Rasul et al., 2021) combining diffusion with RNN hidden states and CSDI (Tashiro et al., 2021) employing self-supervised masking for imputation. More about related work is in Appendix A.

However, despite sophisticated conditioning mechanisms, current approaches suffer from two fundamental limitations. First, they employ a fixed forward diffusion process that adds linear Gaussian noise according to a predefined schedule that cannot adapt to time series characteristics. Implicitly time series exhibit inherent correlations among variables; however, standard forward diffusion processes ignore these dependencies, leading to noise corruption that significantly distorts the underlying dynamics. This rigidity forces the reverse process to follow a potentially suboptimal trajectory for complex time series dynamics. Second, they inherit the limitations of MLPs as the primary

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

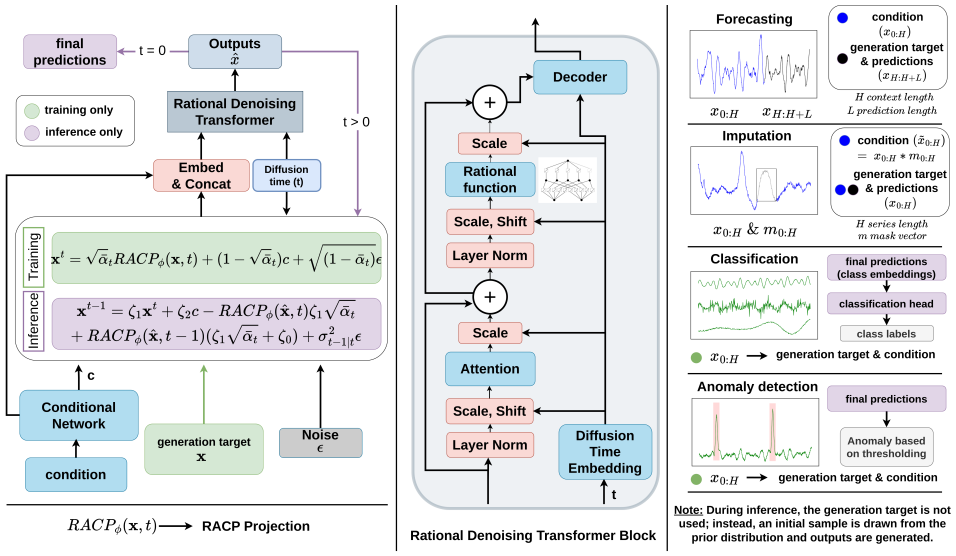


Figure 1: **Overview of the proposed `RAF-Diff` architecture:** The left panel illustrates the Rational Adaptive Correlation Projection (RACP) conditional diffusion process used during training and inference, where noisy inputs are progressively denoised based on condition during inference. The center panel shows our Rational Denoising Transformer (R-DiT) block that integrates diffusion-time embeddings with rational block via scale shift modulation and normalization. The right panel presents task-specific prediction heads and condition inputs, enabling forecasting, imputation, classification, and anomaly detection using a single architecture. During inference, the generation targets is not used; instead, samples are drawn from the learnable prior distribution and predictions are generated through iterative denoising.

computational block. Although theoretically universal function approximators (Hornik et al., 1989), MLPs demonstrably struggle with periodic functions common in temporal signals (Yu et al., 2024; Yang & Wang, 2024) and high-frequency components where gradient descent converges slowly (Rahaman et al., 2019; Basri et al., 2020). In addition, no existing diffusion model provides a single framework for all major tasks such as forecasting, imputation, classification, and anomaly detection. The existing diffusion based time series models have been shown in the literature for up to two of the four time series tasks.

To address these limitations, we adopt rational functions as learnable basis functions (Boullé et al., 2020). This choice yields higher parameter efficiency, enabling complex functions to be modeled using fewer parameters (Yang & Wang, 2024), and has shown notable advantages in mathematical and symbolic regression tasks (Aghaei, 2024). Theoretically, rational functions can approximate a broader class of functions including those with singularities or sharp variations, more efficiently and accurately than polynomials (Walsh, 1935; Baker Jr & Gammel, 1961), thereby offering greater expressive power for modeling intricate time series dynamics that conventional MLPs find difficult to capture.

Building upon these insights, this paper introduces the `RAF-Diff` architecture that uses a Rational Denoising Transformer (R-DiT) with a Rational Adaptive Correlation Projection (RACP) in the forward process. `RAF-Diff` achieves state-of-the-art performance across all four time series analysis tasks, representing the first diffusion model to simultaneously excel in forecasting (8 datasets), imputation (6 datasets), classification (10 datasets), and anomaly detection (5 datasets). The RACP in the forward process learns complex feature correlations specific to time series data, whereas the R-DiT effectively captures periodic patterns through rational functions. Comprehensive ablation studies validate each component's utility, and our analysis reveals how the RACP capture meaningful correlations (Section 4) that enhance the diffusion process. Figure 1 provides an architectural overview.

The main contributions of this paper are:
(1) We propose a Rational Adaptive Correlation Projection (RACP) mechanism within the forward diffusion process, along with a loss formulation and a Rational Denoising Transformer (R-DiT) architecture for time series analysis.
(2) Extensive experiments demonstrate the general applicability of `RAF-Diff` across four major time series analysis tasks, achieving strong performance against leading task-specific and diffusion-based models.
(3) To the best of our knowledge, `RAF-Diff` is the first diffusion-based framework to achieve state-of-the-art performance across all four major time series analysis tasks: forecasting, imputation, classification, and anomaly detection.

2 METHODOLOGY

Consider an N-dimensional multivariate time series observed over a time horizon H, denoted as $x_{0:H} = \{x_0, x_1, \ldots, x_H\} \in \mathbb{R}^{N \times H}$, where the variables at each time step may exhibit dependencies that are important for downstream time series analysis. `RAF-Diff` follows a common diffusion-based architecture across forecasting, imputation, classification, and anomaly detection, while the generation target and conditioning input are defined according to each task, as summarized in Appendix Table 11. In **forecasting**, the model predicts the future sequence $x_{H:H+L}$ conditioned on the historical observations $x_{0:H}$. In **imputation**, the complete sequence $x_{0:H}$ serves as the reconstruction target, while the masked sequence $\tilde{x}_{0:H}$ is provided as the condition. In **classification**, the full sequence $x_{0:H}$ is used to learn representations for class prediction, whereas in **anomaly detection**, the full sequence $x_{0:H}$ is used to learn representations from which anomaly scores are computed.

Existing conditional diffusion models for time series transform inherently correlated temporal distributions into an isotropic Gaussian prior. This transformation is achieved by incrementally adding fixed linear Gaussian noise to data (see Appendix B.1 and B.2). Despite their varying conditioning strategies, these models share two fundamental limitations that critically restrict their modeling capacity. First, the forward diffusion process operates under a fully predetermined fixed noise schedule that is non-trainable and entirely agnostic to the underlying temporal structure, forcing the reverse process to follow a potentially suboptimal trajectory that poorly reflects the true complexity of temporal data distributions. Second, conventional denoising architectures rely on fixed-activation MLP layers as their primary computational backbone, which are well-documented to struggle with the highly nonlinear, periodic, and high-frequency dependencies inherent to real-world time series (Han et al., 2024b; Zhou et al., 2024; Huang et al., 2025; Long & Qin, 2025; Vaca-Rubio et al., 2024). Compounding these issues, no existing diffusion framework simultaneously addresses both limitations, nor has a common diffusion-based framework been shown to achieve state-of-the-art performance across the four major time series analysis tasks of forecasting, imputation, classification, and anomaly detection.

To tackle these problems, we introduce `RAF-Diff`, a conditional diffusion framework built around two core architectural innovations. The first is the Rational Adaptive Correlation Projection (RACP), which fundamentally departs from conventional fixed noise schedules by rendering the forward diffusion process trainable and data-adaptive, enabling the model to capture complex inter-feature correlations specific to temporal data. The second is the Rational Denoising Transformer (R-DiT), which is an Enhansed version of DiT (Diffusion Transformer) designed using rational function transformations.

2.1 RATIONAL ADAPTIVE CORRELATION PROJECTION (RACP) AND OBJECTIVE

Let us define the RACP of the data for marginal distributions for generation target $\mathbf{x}$ (left panel of Figure 1) as follows

$$q_\phi(\mathbf{x}^t|\mathbf{x}) = \mathcal{N}\left(\mathbf{x}^t; \sqrt{\bar{\alpha}_t} RACP_\phi(\mathbf{x}, t), (1-\bar{\alpha}_t)I\right),$$

where $RACP_\phi(\mathbf{x}, t) : \mathbb{R}^{d\times H} \times [0, T] \mapsto \mathbb{R}^{d\times H}$ is a nonlinear RACP projection (see Appendix B.3 for mathematical formulation) parameterized by ϕ that applies a time-dependent projection to the data point $\mathbf{x}$ which helps to learn complex correlations between time series data (Section 4). We now introduce a learnable condition to the forward process, denoted c (Refer to right panel of Figure 1,

Table 1: Summary of dataset benchmarks.

Tasks	Benchmark Datasets	Evaluation Metrics	Series Length
Forecasting	Norpool, Casio, Traffic, Electricity, Weather, Exchange, ETTh1, ETTm1	MSE, MAE	96 to 720
Imputation	ETT (4 subsets), Electricity, Weather	MSE, MAE	96
Classification	UEA (10 subsets)	Accuracy	29 to 1751
Anomaly Detection	SMD, MSL, SMAP, SWaT, PSM	Precision, Recall, F1-Score	100

inspired by a similar formulation used in image diffusion Pandey et al. (2022). Consequently, the marginal distribution is

$$q_\phi(\mathbf{x}^t \mid \mathbf{x}, c) = \mathcal{N}\Big(\mathbf{x}^t;\, \sqrt{\bar{\alpha}}_t\, RACP_\phi(\mathbf{x}, t) + (1 - \sqrt{\bar{\alpha}}_t)\, c,\, (1 - \bar{\alpha}_t)\, I\Big). \tag{1}$$

For $t = T$ along with an appropriately regulated noise schedule α_t, $\bar{\alpha}_T \approx 0$, our formulation results in $q_\phi(\mathbf{x}^T|\mathbf{x}, c) \approx \mathcal{N}(c, I)$. This serves as our learnable prior distribution and inference requires executing our reverse process from $\mathcal{N}(c, I)$.

We derived the following posterior distribution $q_\phi(\mathbf{x}^{t-1}|\mathbf{x}^t, \mathbf{x}, c)$, that satisfies Eq.(1) (Appendix C.1):

$$\mathbf{x}^{t-1} = \zeta_1 \mathbf{x}^t + \zeta_2 c - \zeta_1 \sqrt{\bar{\alpha}_t}\, RACP_\phi(\mathbf{x}, t) + \left(\zeta_1 \sqrt{\bar{\alpha}_t} + \zeta_0\right) RACP_\phi(\mathbf{x}, t-1) + \sigma_{t-1|t}\, \boldsymbol{\epsilon}. \tag{2}$$

Consequently, the corresponding loss formulation of `RAF-Diff` is represented by the following (Derivation in Appendix C.2).

$$\begin{aligned} \mathcal{L}_{\texttt{RAF-Diff}} = \mathbb{E}_{q_\phi}\Bigg[& \underbrace{D_{\mathrm{KL}}\left(q_\phi\left(\mathbf{x}^T|\mathbf{x}, c\right) \| p\left(\mathbf{x}^T|c\right)\right)}_{\mathcal{L}_{\text{prior}}} - \underbrace{\sum_{t=1}^{T} \log p_\theta\left(\mathbf{x}|\mathbf{x}^t, c\right)}_{\mathcal{L}_{\text{rec}}} \\ & + \underbrace{\sum_{t=1}^{T} D_{\mathrm{KL}}\left(q_\phi\left(\mathbf{x}^{t-1}|\mathbf{x}^t, \mathbf{x}, c\right) \| p_\theta\left(\mathbf{x}^{t-1}|\mathbf{x}^t, c\right)\right)}_{\mathcal{L}_{\text{diff}}} \Bigg]. \end{aligned} \tag{3}$$

The details of each loss term and the detailed derivation are provided in the Appendix C.3.

2.2 Rational Denoising Transformer

Beyond modifying the diffusion formulation through RACP projection, we fundamentally redesign the denoising architecture to better capture temporal dependencies in time series data. Traditional diffusion models employ convolutional U-Net architectures as denoisers (Ho et al., 2020), with various architectural modifications proposed, including adaptive normalization layers for conditional information injection Dhariwal & Nichol (2021a); Perez et al. (2018). Recently, transformer architectures have gained prominence in diffusion models through the introduction of Diffusion Transformers (DiT) Peebles & Xie (2023); Esser et al. (2024), demonstrating superior scalability and performance in computer vision tasks.

However, standard transformer architectures face critical limitations when applied to complex time series data Han et al. (2024a). At their core, transformers rely on two fundamental components: attention mechanisms and MLPs. Although extensive research has explored alternatives to conventional attention mechanisms Liu et al. (2021); Wang et al. (2021), these variants predominantly depend on traditional MLPs as their computational backbone. This dependence becomes problematic for time series modeling, where MLPs face several inherent limitations despite their theoretical universal approximation capabilities Hornik et al. (1989).

Interestingly, despite their central role in denoising, comparatively little attention has been devoted to improving the expressiveness of MLPs within diffusion architectures (Shazeer, 2020). We developed the Rational Denoising Transfomer (R-DiT) (center panel in Figure 1) designed using learnable rational basis functions B.3, thereby enriching its capacity to model complex interactions. In

Table 2: **Forecasting**: MSEs of Multivariate prediction on eight real-world time series datasets (subscripts are the rank). CSDI and TiDE ran out of memory on Traffic and Electricity. Results of all baselines are from (Shen et al., 2024). Baselines marked with * are implemented by us using TSLib. (Wang et al., 2024b; Wu et al., 2022)

Method	NorPool	Caiso	Traffic	ECL	Weather	Exchange	ETTh1	ETTm1	Rank
Ours	0.544 (2)	**0.089** (1)	0.374 (2)	**0.144** (1)	**0.293** (1)	0.016 (2)	**0.401** (1)	0.338 (2)	1.5
CnDiff*	**0.531** (1)	0.094 (2)	0.374 (2)	0.145 (2)	0.296 (2)	0.016 (2)	0.405 (2)	0.340 (3)	2.0
mr-Diff	0.645 (5)	0.127 (5)	0.474 (11)	0.155 (7)	0.296 (2)	0.016 (2)	0.411 (5)	0.340 (3)	5.0
TimeDiff	0.665 (7)	0.136 (11)	0.564 (17)	0.193 (16)	0.311 (4)	0.018 (15)	0.407 (3)	**0.336** (1)	9.2
SSSD	0.872 (23)	0.195 (25)	0.642 (23)	0.255 (27)	0.349 (19)	0.061 (30)	0.726 (33)	0.464 (26)	25.8
CSDI	1.010 (34)	0.253 (34)	–	–	0.356 (22)	0.077 (34)	0.497 (19)	0.529 (32)	29.2
TimeGrad	1.152 (36)	0.258 (35)	1.745 (37)	0.736 (37)	0.392 (27)	0.079 (35)	0.993 (38)	0.874 (36)	35.1
D3VAE	0.745 (13)	0.241 (32)	0.928 (31)	0.286 (30)	0.375 (24)	0.200 (37)	0.504 (21)	0.362 (13)	25.1
CPF	1.613 (39)	0.383 (37)	1.625 (36)	0.793 (38)	1.390 (39)	0.016 (2)	0.730 (34)	0.482 (29)	31.8
PSA-GAN	1.501 (38)	0.510 (39)	1.614 (35)	0.535 (36)	1.220 (37)	0.018 (15)	0.623 (32)	0.537 (33)	33.1
N-Hits	0.716 (11)	0.131 (7)	0.386 (4)	0.152 (5)	0.323 (10)	0.017 (10)	0.498 (20)	0.353 (10)	9.6
FiLM	0.723 (12)	0.179 (21)	0.628 (22)	0.210 (20)	0.327 (12)	0.016 (2)	0.426 (10)	0.347 (7)	13.2
NBeats	0.832 (17)	0.141 (12)	**0.373** (1)	0.269 (28)	1.344 (38)	0.016 (2)	0.586 (29)	0.391 (21)	18.5
Depts	0.662 (6)	0.106 (4)	1.018 (34)	0.319 (32)	0.761 (35)	0.020 (18)	0.579 (27)	0.380 (19)	21.9
TimeXer*	0.715 (10)	0.145 (14)	0.435 (9)	0.153 (6)	0.315 (7)	0.017 (12)	0.424 (9)	0.345 (6)	9.1
Crossformer*	0.833 (19)	0.127 (6)	0.491 (13)	0.149 (4)	0.321 (8)	0.069 (32)	0.452 (15)	0.348 (8)	13.1
PAttn*	0.832 (18)	0.162 (18)	0.492 (14)	0.183 (15)	0.346 (17)	**0.015** (1)	0.427 (11)	0.363 (14)	13.5
MultiPatchFormer*	0.836 (20)	0.155 (17)	0.441 (10)	0.167 (10)	0.347 (18)	0.017 (14)	0.418 (7)	0.369 (16)	14.0
iTransformer*	0.864 (22)	0.176 (20)	0.429 (6)	0.160 (8)	0.354 (21)	0.017 (11)	0.440 (14)	0.384 (20)	15.2
TimesNet*	0.827 (15)	0.146 (15)	0.595 (19)	0.179 (14)	0.337 (15)	0.021 (20)	0.460 (16)	0.376 (18)	16.5
Nonstationary*	0.576 (3)	0.134 (9)	0.671 (27)	0.179 (13)	0.352 (20)	0.023 (22)	0.575 (26)	0.463 (25)	18.1
TiDE*	0.947 (28)	0.236 (31)	–	0.203 (19)	0.459 (29)	0.016 (8)	0.435 (12)	0.404 (22)	24.8
FedFormer	0.873 (24)	0.205 (26)	0.591 (18)	0.238 (25)	0.342 (16)	0.133 (36)	0.541 (24)	0.426 (24)	24.1
PatchTST	0.851 (21)	0.193 (24)	0.831 (30)	0.225 (22)	0.782 (36)	0.047 (28)	0.526 (23)	0.372 (17)	25.1
Scaleformer	0.983 (29)	0.207 (28)	0.618 (21)	0.195 (17)	0.462 (30)	0.036 (25)	0.613 (31)	0.481 (28)	26.1
Autoformer	0.940 (26)	0.226 (30)	0.688 (28)	0.201 (18)	0.360 (23)	0.056 (29)	0.516 (22)	0.565 (34)	26.2
ETSformer*	1.011 (35)	0.180 (22)	0.955 (32)	0.241 (26)	0.473 (31)	0.025 (23)	0.599 (30)	0.501 (31)	28.8
Pyraformer	1.008 (33)	0.273 (36)	0.659 (24)	0.273 (29)	0.394 (28)	0.032 (24)	0.579 (27)	0.493 (30)	28.9
Transformer	1.004 (32)	0.206 (27)	0.671 (26)	0.328 (33)	0.388 (26)	0.062 (31)	0.759 (35)	0.992 (38)	31.0
Informer	0.985 (30)	0.251 (33)	0.664 (25)	0.298 (31)	0.385 (25)	0.073 (33)	0.775 (36)	0.673 (35)	31.0
Reformer*	0.907 (25)	0.166 (19)	0.713 (29)	0.332 (34)	0.682 (34)	0.674 (39)	0.953 (37)	0.874 (37)	31.8
NLinear	0.707 (9)	0.135 (10)	0.430 (7)	0.147 (3)	0.313 (5)	0.019 (17)	0.410 (4)	0.349 (9)	8.0
SCINet	0.613 (4)	0.095 (3)	0.434 (8)	0.171 (12)	0.329 (13)	0.036 (25)	0.465 (17)	0.359 (12)	11.8
MICN*	0.770 (14)	0.133 (8)	0.517 (16)	0.168 (11)	0.322 (9)	0.017 (13)	0.435 (13)	0.357 (11)	11.9
TimeMixer*	0.828 (16)	0.183 (23)	0.499 (15)	0.164 (9)	0.337 (14)	0.016 (9)	0.420 (8)	0.366 (15)	13.6
DLinear	0.670 (8)	0.461 (38)	0.389 (5)	0.215 (21)	0.488 (32)	0.022 (21)	0.415 (6)	0.345 (5)	17.0
LightTS*	0.943 (27)	0.151 (16)	0.601 (20)	0.225 (23)	0.324 (11)	0.021 (19)	0.465 (18)	0.411 (23)	19.6
TSMixer*	0.990 (31)	0.144 (13)	0.476 (12)	0.225 (24)	0.314 (6)	0.037 (27)	0.550 (25)	0.478 (27)	20.6
LSTMa	1.481 (37)	0.217 (29)	0.966 (33)	0.414 (35)	0.662 (33)	0.403 (38)	1.149 (39)	1.030 (39)	35.4

addition, the rational blocks are coupled with adaptive layer normalization through dimension-wise scale-and-shift operations, enabling dynamic modulation of feature magnitudes. This results in a denoising architecture that is better aligned with the structured and highly correlated nature of time-series data. Appendix D has detailed formulation of our R-DiT. Algorithms for both training and inference are detailed in Appendix G.

3 EXPERIMENTS

Implementation Details Table 1 summarizes all benchmark datasets. Further information about these datasets are provided in Appendix E. Our `RAF-Diff` model is optimized with Adam using a learning rate of 10^{-4}. Training is carried out with a batch size of 64, incorporates early stopping with a patience of 10, and is run for 100 epochs. The diffusion process follows a quadratic variance schedule, with $\beta_1 = 10^{-4}$ at the start and increasing to $\beta_T = 10^{-1}$ at the final step. All experiments are conducted on a single Nvidia RTX A5000 GPU with 24GB of memory. The model architecture of `RAF-Diff` is depicted in Figure 1 and Standard conditional network is implemented as in Shen et al. (2024); Rishi et al. (2025). Further details on the baselines and hyperparameters are provided in Appendix F and Appendix H, respectively.

3.1 FORECASTING

Setup We performed experiments on eight real-world time series datasets comprising daily, weekly, and monthly forecasts following the established standards in literature Shen et al. (2024); Fan et al. (2022); Zhou et al. (2021); Wu et al. (2021). For all the baselines and our model, the context length of the look-back window H is selected from the set $\{96, 192, 336, 720, 1440\}$, determined by performance evaluations on the validation dataset averaged over ten runs with a fixed prediction

Table 3: **Imputation**: For the Imputation task, we randomly took different proportions of time points specifically 12.5%, 25%, 37.5%, and 50% within 96-length time series and evaluated the performance using MSE. Average outcomes across four distinct masking ratios are presented, indicated by subscripts denoting ranks. For detailed outcomes, please refer to Tables 20 and 21, which present the full results and the MAE metrics. Results of all baselines are implemented using TSLib (Wang et al., 2024b; Wu et al., 2022).

Models	ECL	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Rank
Ours	**0.075** (1)	**0.065** (1)	**0.037** (1)	0.024 (2)	**0.016** (1)	0.032 (3)	1.5
Nonstationary	0.088 (2)	0.075 (2)	0.049 (2)	0.026 (3)	0.021 (2)	0.030 (2)	2.2
TimesNet	0.094 (3)	0.089 (4)	0.051 (3)	0.027 (4)	0.022 (3)	**0.029** (1)	3.0
Crossformer	0.097 (4)	0.148 (12)	0.151 (5)	0.060 (9)	0.079 (5)	0.043 (9)	7.3
LightTS	0.108 (6)	0.159 (14)	0.152 (6)	0.068 (10)	0.076 (4)	0.046 (10)	8.3
iTransformer	0.099 (5)	0.148 (13)	0.139 (4)	0.071 (11)	0.082 (6)	0.051 (12)	8.5
SSSD	0.208 (15)	0.078 (3)	0.410 (17)	**0.023** (1)	0.152 (13)	0.032 (4)	8.8
Reformer	0.162 (11)	0.101 (5)	0.226 (12)	0.033 (5)	0.165 (14)	0.039 (7)	9.0
Pyraformer	0.189 (14)	0.116 (8)	0.201 (10)	0.037 (7)	0.128 (10)	0.034 (5)	9.0
Transformer	0.165 (12)	0.108 (6)	0.256 (14)	0.034 (6)	0.203 (15)	0.036 (6)	9.8
TiDE	0.129 (8)	0.169 (15)	0.163 (8)	0.090 (14)	0.101 (7)	0.052 (13)	10.8
DLinear	0.128 (7)	0.170 (16)	0.162 (7)	0.090 (15)	0.101 (8)	0.052 (14)	11.2
Informer	0.171 (13)	0.114 (7)	0.366 (16)	0.049 (8)	0.245 (16)	0.042 (8)	11.3
ETSformer	0.136 (10)	0.138 (10)	0.234 (13)	0.073 (13)	0.142 (12)	0.048 (11)	11.5
FiLM	0.129 (9)	0.174 (17)	0.169 (9)	0.090 (16)	0.102 (9)	0.057 (15)	12.5
FEDformer	0.215 (16)	0.128 (9)	0.224 (11)	0.072 (12)	0.129 (11)	0.073 (16)	12.5
Autoformer	0.231 (17)	0.147 (11)	0.266 (15)	0.740 (17)	0.862 (17)	0.210 (17)	15.7

length (Table 9) for fair and robust comparison. In this setting, given the condition $x_{0:H}$, the future L time steps $x_{H:H+L}$ are predicted.

Results Table 2 presents the mean squared errors (MSEs) for multivariate forecasting. We see that our `RAF-Diff` has a rank of 1 in 4 out of the 8 datasets. Notable improvements are observed, particularly in complex datasets like Weather and ETTh1. In the other four datasets, `RAF-Diff` consistently holds Rank 2, with a very narrow margin from Rank 1 in each case. On average, the `RAF-Diff` model achieves a ranking of 1.5, demonstrating its state-of-the-art performance in comparison to other leading models in forecasting tasks. The results with mean absolute error (MAE) metrics are presented in Table 19 of the Appendix L.

Table 4: **Classification**: Accuracy of the classification for 10 Datasets followed by average accuracy and Rank. **to be read as former (e.g., In** is Informer). FiLM ran out of memory for PEMS-SF. Baselines are implemented using TSLib (Wang et al., 2024b).

Method	Ethanol	FaceDet	Handwriting	Heartbeat	Japanese	PEMS-SF	SCP1	SCP2	SpokenArab	UWave	Avg	Rank
Ours	0.316 (2)	**0.692** (1)	**0.382** (1)	0.750 (8)	**0.978** (1)	**0.898** (1)	0.914 (2)	**0.578** (1)	**0.994** (1)	**0.884** (1)	0.738	1.9
TimesNet	0.304 (4)	0.674 (5)	0.327 (3)	**0.809** (1)	**0.978** (1)	0.884 (2)	0.914 (2)	0.555 (3)	0.988 (3)	0.881 (2)	0.731	2.6
Trans**	0.277 (10)	0.686 (2)	0.377 (2)	0.780 (2)	**0.978** (1)	0.832 (7)	0.907 (7)	0.533 (9)	0.989 (2)	0.865 (4)	0.722	4.7
In**	0.258 (13)	0.669 (7)	0.317 (5)	0.780 (2)	0.970 (6)	0.826 (8)	0.914 (2)	0.561 (2)	0.986 (5)	0.862 (5)	0.714	5.6
Re**	0.281 (8)	0.681 (3)	0.316 (6)	0.775 (4)	0.970 (6)	0.815 (9)	0.897 (8)	0.533 (9)	0.985 (6)	0.868 (3)	0.712	6.0
iTrans**	0.269 (12)	0.669 (7)	0.271 (8)	0.756 (6)	0.975 (5)	0.872 (3)	**0.921** (1)	0.544 (5)	0.980 (7)	0.859 (6)	0.712	6.1
FED**	0.292 (6)	0.670 (6)	0.235 (10)	0.765 (5)	**0.978** (1)	0.843 (6)	0.597 (13)	0.527 (11)	0.987 (4)	0.587 (13)	0.648	7.5
LightTS	0.304 (4)	0.665 (9)	0.205 (12)	0.751 (7)	0.962 (12)	0.867 (4)	0.911 (6)	0.538 (7)	0.977 (8)	0.831 (9)	0.701	7.8
Pyra**	0.311 (3)	0.555 (14)	0.325 (4)	0.663 (13)	0.970 (6)	0.554 (13)	0.802 (11)	0.538 (7)	0.968 (11)	0.825 (10)	0.651	9.2
PatchTST	0.273 (11)	0.679 (4)	0.265 (9)	0.678 (11)	0.959 (13)	0.861 (5)	0.849 (10)	0.516 (12)	0.966 (12)	0.850 (7)	0.690	9.6
FiLM	0.281 (8)	0.645 (10)	0.136 (14)	0.731 (9)	0.913 (14)	-	0.890 (9)	0.550 (4)	0.977 (8)	0.781 (12)	0.656	9.8
Cross**	**0.395** (1)	0.625 (12)	0.276 (7)	0.546 (14)	0.970 (6)	0.809 (10)	0.754 (12)	0.461 (14)	0.949 (14)	0.834 (8)	0.662	9.9
DLinear	0.254 (14)	0.628 (11)	0.221 (11)	0.678 (11)	0.967 (10)	0.793 (11)	0.914 (2)	0.544 (5)	0.964 (13)	0.821 (11)	0.678	10.1
Auto**	0.285 (7)	0.586 (13)	0.185 (13)	0.721 (10)	0.964 (11)	0.786 (12)	0.552 (14)	0.511 (13)	0.975 (10)	0.518 (14)	0.608	11.8

3.2 IMPUTATION

Setup For imputation experiments, we use six datasets from the electricity and weather domains as benchmarks, including ETT (Zhou et al., 2021), Electricity [1], and Weather [2], all of which frequently encounter missing data issues. To evaluate model performance under varying levels of missing data, we randomly mask time points with ratios of 12.5%, 25%, 37.5%, 50%. For this task, masked time series $\tilde{x}_{0:H}$ is used as the condition for `RAF-Diff` and the process of noising and denoising are restricted to the masked portions in the diffusion process.

[1] `https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014`
[2] `https://www.bgc-jena.mpg.de/wetter/`

Table 5: **Anomaly detection**: We calculate the F1-score (presented as a percentage) for each dataset. A higher F1-score signifies enhanced performance. For comprehensive results, refer to Table 22. All Baselines were executed with TSLib (Wang et al., 2024b; Wu et al., 2022).

Model	MSL	PSM	SMAP	SMD	SWaT	Avg F1	Rank
Ours	**89.74** (1)	95.76 (4)	**91.84** (1)	83.79 (4)	90.16 (2)	90.26	2.4
TimesNet	81.80 (7)	**97.40** (1)	69.35 (8)	84.59 (3)	**92.48** (1)	85.12	4.0
SSSD	85.55 (2)	92.97 (9)	70.47 (5)	**88.88** (1)	88.05 (5)	85.18	4.4
MICN	79.41 (14)	95.93 (3)	70.52 (4)	84.82 (2)	89.85 (3)	84.11	5.2
DLinear	81.87 (6)	96.63 (2)	68.97 (15)	83.00 (7)	85.73 (7)	83.24	7.4
iTransformer	72.53 (16)	94.32 (5)	69.31 (9)	81.41 (9)	89.24 (4)	81.36	8.6
AnomalyTrans^{+}	83.59 (3)	92.68 (10)	70.15 (6)	78.50 (11)	79.24 (17)	80.83	9.4
FiLM	63.88 (18)	93.70 (7)	69.01 (14)	83.25 (5)	85.72 (8)	79.11	10.4
LightTS	80.99 (11)	93.41 (8)	68.92 (17)	82.70 (8)	81.74 (9)	81.55	10.6
Transformer	80.90 (12)	90.64 (16)	73.26 (3)	71.19 (14)	79.53 (10)	79.10	11.0
KANAD	81.07 (10)	90.53 (17)	69.10 (12)	83.09 (6)	79.30 (11)	80.62	11.2
Reformer	81.10 (8)	92.33 (11)	69.26 (11)	71.04 (16)	79.30 (11)	78.61	11.4
Autoformer	82.59 (4)	88.23 (18)	74.01 (2)	71.16 (15)	79.19 (18)	79.04	11.4
Pyraformer	77.19 (15)	92.24 (13)	69.41 (7)	78.97 (10)	79.29 (14)	79.42	11.8
Informer	81.08 (9)	92.32 (12)	69.31 (9)	71.04 (16)	79.29 (14)	78.61	12.0
Crossformer	80.61 (13)	94.30 (6)	69.06 (13)	74.50 (13)	79.28 (16)	79.55	12.2
FEDformer	82.22 (5)	91.40 (14)	68.96 (16)	69.93 (18)	79.30 (11)	78.36	12.8
ETSformer	68.34 (17)	91.09 (15)	68.57 (18)	76.15 (12)	86.76 (6)	78.18	13.6

Results Table 3 presents the results for the imputation with the evaluation metric as the average MSE over various random masking ratios. Our model consistently delivers top-tier performance, recording the lowest average rank considering all datasets, with notably marked improvements on datasets such as ECL, ETTh1, ETTh2, ETTm1, and ETTm2. For example, in ETTm2, our method achieves an error rate of 0.016, significantly exceeding the nearest competitor. Overall, `RAF-Diff` secures an average rank of 1.5, clearly outperforming robust baselines like Nonstationary, TimesNet, and Crossformer, which hold ranks of 2.2, 3.0, and 7.3, respectively. These findings emphasize our model's superior ability to manage missing data and accurately reconstruct masked time series portions with diverse temporal dynamics. The results with MSE for individual masking rations and MAE as the metric are provided in Tables 20 and 21 in the appendix L.

3.3 Classification

Setup For classification experiments, we perform sequence-level classification using `RAF-Diff` to learn the representations that are fed into a classification head. The benchmark datasets are chosen to be 10 multivariate datasets from the UEA Time Series Classification Archive (Bagnall et al., 2018), which include tasks such as gesture, action, and audio recognition, as well as medical diagnosis through heartbeat monitoring, among other practical applications. Subsequently, we pre-process these datasets following (Zerveas et al., 2021), noting that different subsets exhibit varying sequence lengths. The time series that must be classified is used as a condition ($x_{0:H}$) for diffusion, and noise is added to this series, allowing the `RAF-Diff` to learn representations which are fed to the classification head. Training is performed end-to-end, including the classification head (Refer to right panel of Figure 1).

Results Table 4 presents accuracy as the metric for the classification. Our `RAF-Diff` consistently attains the highest average accuracy of 0.738 along with the best average ranking of 1.9, surpassing all other models. In most datasets, `RAF-Diff` achieves the highest accuracy (marked in bold) or ranks among the top performers. Notably, it achieves the best accuracy on FaceDet (0.692), Handwriting (0.382), PEMS-SF (0.898), SpokenArab (0.994), and UWave (0.884). In summary `RAF-Diff` stands out with its superior average ranking and overall accuracy. This indicates that our new method is more effective and robust for the time series classification compared to existing state-of-the-art models.

3.4 Anomaly Detection

Setup Experiments are performed for anomaly detection in time series, aiming to pinpoint abnormal time instances. Five popular anomaly detection benchmarks are used: SMD (Su et al., 2019), MSL (Hundman et al., 2018a), SMAP (Hundman et al., 2018b), SWaT (Mathur & Tippenhauer, 2016), and PSM (Abdulaal et al., 2021), which encompass applications in service monitoring, space and

Table 6: Correlation Comparison Before and After RACP Projection

Task	Dataset (window)	Stage	Mean	Median	Std	P95	Max
Forecasting	Electricity (168)	Before	0.5888	0.5889	0.0008	0.5898	0.5903
	Electricity (168)	After	**0.8800**	**0.8789**	0.0688	0.9361	0.9375
Forecasting	ETTm1 (192)	Before	-0.0587	-0.0581	0.0029	-0.0540	-0.0518
	ETTm1 (192)	After	**0.1033**	**0.1054**	0.0109	0.1185	0.1238
Anomaly Detection	SMAP (100)	Before	0.9482	0.9619	0.0405	0.9930	0.9939
	SMAP (100)	After	**0.7684**	**0.7826**	0.0689	0.8486	0.8643
Anomaly Detection	MSL (100)	Before	0.8934	0.8981	0.0532	0.9773	0.9867
	MSL (100)	After	**0.5782**	**0.5766**	0.0607	0.7010	0.7170
Imputation	ETTh1 (96)	Before	0.5306	0.5280	0.0083	0.5450	0.5469
	ETTh1 (96)	After	**0.9984**	**0.9997**	0.0026	0.9999	0.9999

earth exploration, and water treatment. Using the pre-processing techniques of Anomaly Transformer (Xu et al., 2021), we segment the data set into successive non-overlapping portions using a sliding window approach. In prior research, reconstruction has served as a standard method for unsupervised point-wise representation learning, where reconstruction error acts as a natural anomaly indicator. To ensure a fair evaluation, we modify only the base model used for reconstruction with our model, while maintaining the classical reconstruction error as the universal anomaly criterion across all our and baseline experiments. For anomaly detection, a linear layer serves as the conditioning network in our model.

Results Table 5 summarizes the results of anomaly detection tasks using F1 score as evaluation metric. `RAF-Diff` excels as the leading method in most of the datasets. It achieves a notable margin over other leading models for MSL and SMAP datasets. For the other datasets, the F1 score are within top 4 ranks. Overall, `RAF-Diff` achieves the highest average F1 score of 90.26% with an impressive average rank of 2.4. Additionally, Table 22 presents a comparison of the Accuracy, Recall, and Precision metrics. Furthermore, the We provide detailed sections with Extended Quantitative results (Appendix L) and Qualitative results (Appendix O).

4 Model Analysis

Rational Adaptive Functional Projection. To understand how the RACP component facilitates improved correlation modeling in time series, we analyze the correlations in the learned latent representation space before and after applying the RACP projection. This analysis is conducted across all timesteps, multiple temporal windows, and diverse datasets spanning various tasks, as summarized in Table 6. These results highlight several key observations. First, the observed correlation shifts are consistent across multiple datasets and tasks, including Electricity, ETTm1, SMAP, MSL, and ETTh1, indicating that the effect is robust and not dataset-specific. These findings support our hypothesis (Section 1) that increased correlation and enhanced time-dependent adaptability enable a more effective diffusion process for time series modeling. Notably, similar observations regarding the role of structured correlations in latent spaces improving diffusion performance have also been reported in image and video diffusion literature (Ge et al., 2023). In the space of diffusion models for time series, learning the correlation between feature and temporal space has been reported to be necessary for time series imputation tasks Tashiro et al. (2021).

Second, the behavior is task-dependent and systematic: for forecasting and imputation, correlations increase significantly, indicating improved alignment with target structures, whereas for anomaly detection, correlations decrease, which is desirable, as anomalies correspond to deviations from normal correlated patterns. Third, the magnitude of change is substantial, for example, Electricity shows a large increase in mean correlation ($0.5888 \rightarrow 0.8800$), and ETTh1 achieves near-perfect correlation after conditioning. Finally, the inclusion of multiple statistical measures (mean, median, standard deviation, tail statistics) computed over all timesteps further confirms that these effects are robust. Refer to Appendix N for visual example.

Ablation study of different `RAF-Diff` components for time series analysis. We conducted an ablation study to evaluate the impact of our architectural contributions.

RACP vs Without RACP: Table 7 presents a comprehensive comparison between `RAF-Diff` trained with and without the proposed RACP projection across multiple time-series analysis tasks. RACP

Table 7: Comparison of "with and without RACP Projection" across a variety of tasks. Various evaluation metrics are utilized, such as MSE for Imputation (25 % mask) and Forecasting, Accuracy for Anomaly Detection and classification.

Task & Metric	Dataset	w/ RACP	w/o RACP
Forecasting (MSE)	ETTh1	**0.4015**	0.9495
	Weather	**0.2930**	0.6960
Imputation (MSE)	ETTh1	**0.0520**	0.0649
	ETTm1	**0.0210**	0.0454
	ETTm2	**0.0140**	0.0309
Classification (Accuracy)	Handwriting	**0.3822**	0.2115
	PEMS	**0.8984**	0.8359
	SCP2	**0.5781**	0.4922
Anomaly Detection (Accuracy)	MSL	**0.9791**	0.9765
	PSM	**0.9783**	0.9725
	SMAP	**0.9793**	0.9797
	SMD	**0.9876**	0.9863
	SwAT	**0.9783**	0.9703

Table 8: Ablation study comparing of each architecture component of `RAF-Diff` across different tasks. Results show forecasting performance (MSE), anomaly detection capability (F1 score), and classification accuracy for `RAF-Diff` with different neural network components

Denoiser →	DiT		R-DiT	
Task	Projection Type			
Dataset, Metric	MLP	RACP	MLP	RACP
Forecasting Weather, MSE	0.325	0.320	0.308	**0.293**
Imputation ETTm2, MSE	0.035	0.028	0.034	**0.014**
Anomaly Detection SwAT, F1	80.80	88.33	87.35	**90.16**
Classification PEMS-SF, Acc.	0.828	0.838	0.844	**0.898**

consistently improves performance across tasks. For forecasting and imputation (MSE), it greatly lowers errors (e.g., ETTh1: 0.40 vs 0.95; Weather: 0.29 vs 0.70), indicating more accurate modeling of temporal dependencies, similarly, in imputation (MSE), RACP yields lower errors across all datasets. For classification and anomaly detection (accuracy), it yields higher scores (e.g., Handwriting: 0.38 vs 0.21) and consistent gains across all benchmarks. Overall, these results demonstrate that integrating RACP projection systematically improves `RAF-Diff` performance across diverse time-series tasks by enabling more effective representation learning.

While RACP demonstrably improves model performance, an important question remains: why specifically a rational projection? To justify this design choice, we conduct experiments comparing both DiT and R-DiT under two projection schemes, standard MLP projection and the proposed RACP projection, as presented in the following section.

DiT vs R-DiT with various Projection types: Table 8 shows four variations: *(i)* Transitioning from DiT to R-DiT with MLP projection layers (column 1 $\rightarrow$ 3) and with *(ii)* RACP (column 2 $\rightarrow$ 4), both leads to improved performance irrespective of projection layers, underscoring the effectiveness of our novel R-DiT module. Using RACP over MLP Projection in both the *(iii)* DiT (column 1 $\rightarrow$ 2) and *(iv)* R-DiT (column 3 $\rightarrow$ 4) configurations also yields performance gains, emphasizing our second contribution RACP over an MLP projection. Overall R-DiT + RACP configuration (column 4) achieves the largest improvement among all in all time series tasks, clearly demonstrating the combined strength. This evidences both the standalone strength of each innovation and their enhanced performance when combined.

In addition to these ablation studies, we also evaluated the efficiency and robustness of our model via Computational time analysis (App J), Statistical significance (App K), Sensitivity analysis for different diffusion hyperparameters (App I.1), and for various look-back windows (App I.2).

5 Conclusion

This paper introduces `RAF-Diff`, a diffusion-based framework featuring two key innovations: Rational Adaptive Correlation Projection (RACP) in the forward diffusion process and the Rational Denoising Transformer (R-DiT) architecture. `RAF-Diff` is the first diffusion-based model to achieve superior performance across all four time series analysis tasks, outperforming 38 baselines for Forecasting (Table 2; 8 datasets), 15 for Imputation (Table 3; 6 datasets), 14 for Classification (Table 4; 10 datasets), and 17 for Anomaly Detection (Table 5; 5 datasets). R-DiT addresses limitations of traditional MLPs in capturing complex temporal patterns, while learnable RACP enables effective feature correlation modeling. Validated through extensive ablation studies, `RAF-Diff` establishes a new paradigm bridging generative modeling and time series analysis.

AI USE STATEMENT

In this work, we used generative AI tools solely for language editing and polishing the manuscript, including improving grammar, clarity, readability, and sentence structure. Generative AI tools were not used to generate research ideas, develop the methodology, conduct experiments, analyze results, produce scientific claims, or create figures. All AI-assisted edits were carefully reviewed by the authors to ensure that they preserved the intended technical meaning and accurately reflected the work performed. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide the source code as part of the supplementary material. The main paper describes the proposed RAF-Diff framework, training objectives, experimental setup, and evaluation protocols, while the appendix provides additional details on datasets, model configurations, hyperparameters, algorithms, sensitivity analyses, computational analysis, and extended results. Together with the released code, these materials are intended to enable reproduction of the reported experiments and facilitate further evaluation of `RAF-Diff` across forecasting, imputation, classification, and anomaly detection tasks.

REFERENCES

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

Alireza Afzal Aghaei. rkan: Rational kolmogorov-arnold networks. *arXiv preprint arXiv:2406.14495*, 2024.

Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

George A Baker Jr and John L Gammel. The padé approximant. *Journal of Mathematical Analysis and Applications*, 2(1):21–30, 1961.

Grigory Bartosh, Dmitry Vetrov, and Christian A Naesseth. Neural diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 3073–3095, 2024.

Ronen Basri, Meirav Galun, Amnon Geifman, David Jacobs, Yoni Kasten, and Shira Kritchman. Frequency bias in neural networks for input of non-uniform density. In *International conference on machine learning*, pp. 685–694. PMLR, 2020.

Yaniv Benny and Lior Wolf. Dynamic dual-output diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11482–11491, 2022.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22563–22575, 2023.

Nicolas Boullé, Yuji Nakatsukasa, and Alex Townsend. Rational neural networks. *Advances in neural information processing systems*, 33:14243–14253, 2020.

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.

Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 6989–6997, 2023.

Si-An Chen, Chun-Liang Li, Sercan O Arik, Nathanael Christian Yoder, and Tomas Pfister. Tsmixer: An all-mlp architecture for time series forecast-ing. *Transactions on Machine Learning Research*, 2023.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*, 2023a.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*, 2023b.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021a.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021b.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

Jonathan Dumas, Antoine Wehenkel, Damien Lanaspeze, Bertrand Cornélusse, and Antonio Sutera. A deep generative model for probabilistic energy forecasting in power systems: normalizing flows. *Applied Energy*, 305:117871, 2022.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

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. In *International Conference on Learning Representations*, 2022.

Shibo Feng, Chunyan Miao, Zhong Zhang, and Peilin Zhao. Latent diffusion transformer for probabilistic time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11979–11987, 2024.

Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. *Advances in neural information processing systems*, 32, 2019.

Milton Friedman. The interpolation of time series by related series. *Journal of the American Statistical Association*, 57(300):729–757, 1962.

Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22930–22941, 2023.

Xiao Han, Xinfeng Zhang, Yiling Wu, Zhenduo Zhang, and Zhe Wu. Are kans effective for multivariate time series forecasting? *arXiv preprint arXiv:2408.11306*, 2024a.

Xiao Han, Xinfeng Zhang, Yiling Wu, Zhenduo Zhang, and Zhe Wu. Are kans effective for multivariate time series forecasting? *arXiv preprint arXiv:2408.11306*, 2024b.

William Harvey, Saeid Naderiparizi, Vaden Masrani, Christian Weilbach, and Frank Wood. Flexible diffusion modeling of long videos. *Advances in Neural Information Processing Systems*, 35: 27953–27965, 2022.

Robert Hecht-Nielsen. Kolmogorov's mapping neural network existence theorem. In *Proceedings of the international conference on Neural Networks*, volume 3, pp. 11–14. IEEE press New York, NY, USA, 1987.

Bruno Miranda Henrique, Vinicius Amorim Sobreiro, and Herbert Kimura. Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*, 124:226–251, 2019.

Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37 (1):388–427, 2021.

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

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Songtao Huang, Zhen Zhao, Can Li, and Lei Bai. Timekan: Kan-based frequency decomposition learning architecture for long-term time series forecasting. *arXiv preprint arXiv:2502.06910*, 2025.

Xingshuai Huang, Di Wu, and Benoit Boulet. Metaprobformer for charging load probabilistic forecasting of electric vehicle charging stations. *IEEE Transactions on Intelligent Transportation Systems*, 24(10):10445–10455, 2023.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 387–395, 2018a.

Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 387–395, 2018b.

Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

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 *The Tenth International Conference on Learning Representations*, 2022.

Kyoung-jae Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1-2):307–319, 2003.

Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013a.

Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013b.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

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.

Lizao Li, Robert Carver, Ignacio Lopez-Gomez, Fei Sha, and John Anderson. Generative emulation of weather forecast ensembles with diffusion models. *Science Advances*, 10(13):eadk4489, 2024a.

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.

Yuxin Li, Wenchao Chen, Xinyue Hu, Bo Chen, Mingyuan Zhou, et al. Transformer-modulated diffusion models for probabilistic multivariate time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024b.

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, 2022a.

Qihao Liu, Zhanpeng Zeng, Ju He, Qihang Yu, Xiaohui Shen, and Liang-Chieh Chen. Alleviating distortion in image generation via multi-resolution diffusion models and time-dependent layer normalization. *Advances in Neural Information Processing Systems*, 37:133879–133907, 2024a.

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 *# PLACEHOLDER_PARENT_METADATA_VALUE#*, 2022b.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in neural information processing systems*, 35: 9881–9893, 2022c.

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.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks. *arXiv preprint arXiv:2404.19756*, 2024b.

Yunji Long and Xue Qin. A multivariate time series prediction model based on the kan network. *Scientific Reports*, 15(1):23621, 2025.

Juan Miguel Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured atate apace models. *Transactions on machine learning research*, pp. 1–36, 2023.

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

Alejandro Molina, Patrick Schramowski, and Kristian Kersting. Pad\'e activation units: End-to-end learning of flexible activation functions in deep networks. *arXiv preprint arXiv:1907.06732*, 2019.

Vahid Naghashi, Mounir Boukadoum, and Abdoulaye Banire Diallo. A multiscale model for multivariate time series forecasting. *Scientific Reports*, 15(1):1565, 2025.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2019.

Kushagra Pandey, Avideep Mukherjee, Piyush Rai, and Abhishek Kumar. Diffusevae: Efficient, controllable and high-fidelity generation from low-dimensional latents. *Transactions on Machine Learning Research*, 2022.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Xinji Qu, Zhuo Liu, Chase Q Wu, Aiqin Hou, Xiaoyan Yin, and Zhulian Chen. Mfgan: multimodal fusion for industrial anomaly detection using attention-based autoencoder and generative adversarial network. *Sensors*, 24(2):637, 2024.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pp. 5301–5310. PMLR, 2019.

Syama Sundar Rangapuram, Shubham Kapoor, Rajbir Singh Nirwan, Pedro Mercado, Tim Januschowski, Yuyang Wang, and Michael Bohlke-Schneider. Coherent probabilistic forecasting of temporal hierarchies. In *International Conference on Artificial Intelligence and Statistics*, pp. 9362–9376. PMLR, 2023.

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.

Kashif Rasul, Young-Jin Park, Max Nihlén Ramström, and Kyung-Min Kim. Vq-ar: Vector quantized autoregressive probabilistic time series forecasting. *arXiv preprint arXiv:2205.15894*, 2022.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pp. 1278–1286. PMLR, 2014.

J Rishi, GVS Mothish, and Deepak Subramani. Conditional diffusion model with nonlinear data transformation for time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022.

J Taylor Sean and J Taylor. Forecasting at scale. *Am. Stat*, 72(1):37–45, 2018.

Mohammad Amin Shabani, Amir H Abdi, Lili Meng, and Tristan Sylvain. Scaleformer: Iterative multi-scale refining transformers for time series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

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

Lifeng Shen, Weiyu Chen, and James Kwok. Multi-resolution diffusion models for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

Robert H Shumway, David S Stoffer, Robert H Shumway, and David S Stoffer. Arima models. *Time series analysis and its applications: with R examples*, pp. 75–163, 2017.

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.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2828–2837, 2019.

Mingtian Tan, Mike Merrill, Vinayak Gupta, Tim Althoff, and Tom Hartvigsen. Are language models actually useful for time series forecasting? *Advances in Neural Information Processing Systems*, 37:60162–60191, 2024.

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

Cristian J Vaca-Rubio, Luis Blanco, Roberto Pereira, and Màrius Caus. Kolmogorov-arnold networks (kans) for time series analysis. In *2024 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6. IEEE, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Joseph Leonard Walsh. *Interpolation and approximation by rational functions in the complex domain*, volume 20. American Mathematical Soc., 1935.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The eleventh international conference on learning representations*, 2023.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. *arXiv preprint arXiv:2405.14616*, 2024a.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 568–578, 2021.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. 2024b.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. Timexer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems*, 37:469–498, 2024c.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*, 2022.

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.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint arXiv:2110.02642*, 2021.

Xingyi Yang and Xinchao Wang. Kolmogorov-arnold transformer. *arXiv preprint arXiv:2409.10594*, 2024.

Runpeng Yu, Weihao Yu, and Xinchao Wang. Kan or mlp: A fairer comparison. *arXiv preprint arXiv:2407.16674*, 2024.

Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8980–8987, 2022.

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.

George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 2114–2124, 2021.

Peng Zhan, Shaokun Wang, Jun Wang, Leigang Qu, Kun Wang, Yupeng Hu, and Xueqing Li. Temporal anomaly detection on iiot-enabled manufacturing. *Journal of Intelligent Manufacturing*, 32 (6):1669–1678, 2021.

T Zhang, Y Zhang, W Cao, J Bian, X Yi, S Zheng, and J Li. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures. arxiv 2022. *arXiv preprint arXiv:2207.01186*, 2022.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *The eleventh international conference on learning representations*, 2023.

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.

Quan Zhou, Changhua Pei, Fei Sun, Jing Han, Zhengwei Gao, Dan Pei, Haiming Zhang, Gaogang Xie, and Jianhui Li. Kan-ad: time series anomaly detection with kolmogorov-arnold networks. *arXiv preprint arXiv:2411.00278*, 2024.

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, 2022a.

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, 2022b.

A RELATED WORK

Classical time series methods include ARIMA (Shumway et al., 2017), Holt-Winter (Hyndman & Athanasopoulos, 2018), and Prophet (Sean & Taylor, 2018). Real-world time series often exhibit a complexity that exceeds these predefined frameworks, which restricts the effectiveness of these conventional approaches in practical applications. Thus several deep learning methods have been proposed for time series modeling.

Basis expansion has been utilized by several models for time-series analysis. FiLM (Zhou et al., 2022a) employed Fourier analysis and low-rank matrix approximation to reduce noise, along with Legendre polynomial projection to maintain historical data representations. NBeats (Oreshkin et al., 2019) is an interpretable architecture that combines polynomial trend modeling with Fourier techniques for detecting seasonality. Depts (Fan et al., 2022) builds on NBeats by adding a periodicity module for periodic series, while N-Hits (Challu et al., 2023) enhances the approach using multi-scale hierarchical interpolation.

Models such as SCINet (Liu et al., 2022a) utilize a recursive strategy involving downsampling, convolution, and interaction, to harness temporal dependencies in downsampled subsequences. NLinear (Zeng et al., 2023) implements a basic approach by normalizing the time series before using a linear layer for prediction. DLinear (Zeng et al., 2023) employs a seasonal-trend decomposition akin to Autoformer (Wu et al., 2021).

Transformer Based Models: Several models utilized the transformer (Vaswani et al., 2017) and its variations for time series analysis. Informer (Zhou et al., 2021) used sparse attention to minimize computational load and employed a generative-style decoder for rapid long-sequence in one forward pass. Autoformer (Wu et al., 2021) substituted traditional self-attention with an autocorrelation layer, while Fedformer (Zhou et al., 2022b) incorporated frequency domain mapping through a frequency-enhanced module. Pyraformer (Liu et al., 2022b) introduced pyramidal attention for multi-resolution representation, and Scaleformer (Shabani et al., 2023) adopted a shared-weight multilevel forecasting approach, from broad to fine scales. Inspired by vision transformers (Dosovitskiy et al., 2020), PatchTST (Nie et al., 2023) partitions time series data into subseries patches and uses self-supervised pre-training to extract local semantic features.

Diffusion Based Models: TimeGrad (Rasul et al., 2021) pioneered the use of conditional diffusion models, leveraging autoregressive prediction informed by RNN hidden states. CSDI (Tashiro et al., 2021), applies non-autoregressive generation through self-supervised masking but relies on dual transformers, facing boundary inconsistencies and computational challenges with large datasets. SSSD (Lopez Alcaraz & Strodthoff, 2023) aims to reduce CSDI's computational load using structured state-space models, but it continues with masking-based conditioning, preserving boundary issues. TimeDiff (Shen & Kwok, 2023) introduced future mixup and autoregressive initialization within an encoder-decoder scheme for denoising. TMDM (Li et al., 2024b) integrates transformers with a diffusion process for probabilistic multivariate time series forecasting. TimeDiT (Cao et al., 2024) is a foundational model for time series, which employed a transformer type architecture to capture temporal dependencies and employs diffusion processes to generate samples. mr-Diff(Shen et al., 2024) is a recent study utilizing a multi-resolution strategy, incorporating fine-to-coarse patterns as latent variables to aid in denoising. NDMBartosh et al. (2024) applies a non-linear time-dependent transformation for image diffusion models with the Markovian reverse process. CN-Diff (Rishi et al., 2025) introduced a generative framework that employs novel non-linear transformations and learnable conditions in the forward process for time series forecasting. Unlike CN-Diff, which introduces a nonlinear learnable transformation in the forward diffusion process mainly for forecasting, RAF-Diff uses a rational-function-based adaptive correlation projection (RACP) designed to explicitly model inter-feature dependencies in time series. RAF-Diff further introduces a Rational Denoising Transformer (R-DiT) and extends the framework consistently across forecasting, imputation, classification, and anomaly detection. In addition, the current study focuses on task-specific time series modeling rather than large-scale pretrained foundation models. Training such foundation models typically requires substantially greater computational resources and large-scale pretraining across heterogeneous datasets like Timedit Cao et al. (2024), which is beyond the scope of this work.

B PRELIMINARIES

B.1 DIFFUSION MODEL FORMULATION

Diffusion models are generative frameworks that progressively corrupt data through a forward noising process and then learn to reverse this process to generate realistic samples. For a data point $x \sim q(x)$, the forward process produces increasingly noisy latent variables $x^0, x^1, \ldots, x^T$, until x^T resembles pure Gaussian noise. The generative model is trained to reverse this chain, starting from noise and reconstructing $x^0 \approx x$.

In the standard diffusion model, the forward process follows a linear Gaussian Markov chain (Sohl-Dickstein et al., 2015; Ho et al., 2020). In denoising diffusion probabilistic models (DDPM), at step t, x^t is generated by modifying the previous state x^{t-1} (multiplied by $\sqrt{\alpha_t}$) with zero-mean Gaussian noise and variance $(1-\alpha_t)$, that is, $q(x^t|x^{t-1}) = \mathcal{N}\left(x^t; \sqrt{\alpha_t}x^{t-1}, (1-\alpha_t)I\right)$. We derive the marginal distribution as $q(x^t|x) = \mathcal{N}(x^t; \sqrt{\bar{\alpha}_t}x, (1-\bar{\alpha}_t)I)$, where $\bar{\alpha}_t$ is defined as $\prod_{s=1}^{t} \alpha_s$. Hence, $x^t = \sqrt{\bar{\alpha}_t}x + \sqrt{1-\bar{\alpha}_t}\epsilon$, with $\epsilon \sim \mathcal{N}(0, I)$. Leveraging these marginal distributions, the joint distribution for the latent variables $x^0, x^1, x^2, \ldots, x^T$ is

$$q(x^{0:T}|x) = \prod_{t=1}^{T} q(x^t \mid x^{t-1}) \,.$$

The forward process is typically fixed, lacking trainable parameters, and it is designed so that $q(x^0|x) \approx \delta(x^0 - x)$ and $q(x^T|x) \approx \mathcal{N}(x^T; 0, I)$. If accessing $q(x^{t-1}|x^t)$ were feasible, we could sample from $x^T \sim \mathcal{N}(x^T; 0, I)$ and reverse it to yield $x^0 \sim q(x^0) \approx q(x)$. The distribution $q(x^{t-1}|x^t)$ depends implicitly on $q(x)$, forming a complex relationship. Hence, we approximate the reverse process through a Markov chain adopting the form

$$p_\theta(x^{0:T}) = p(x^T) \prod_{t=1}^{T} p_\theta(x^{t-1}|x^t),$$

with $p(x^T) = \mathcal{N}(x^T; 0, I)$. The integration of the forward process q with the reverse process p_θ is akin to a (hierarchical) variational autoencoder (Kingma, 2013a; Rezende et al., 2014). During training, the standard variational bound of the negative log-likelihood is minimized. In the case of DDPM (Ho et al., 2020), the loss to be minimized is

$$\mathcal{L} = \mathbb{E}_q \left[\underbrace{D_{\mathrm{KL}}\left(q(x^T|x) \,\|\, p(x^T)\right)}_{\mathcal{L}_{\text{prior}}} \underbrace{- \log p_\theta(x|x^0)}_{\mathcal{L}_{\text{rec}}} + \sum_{t=1}^{T} \underbrace{D_{\mathrm{KL}}\left(q(x^{t-1}|x^t, x) \,\|\, p_\theta(x^{t-1}|x^t)\right)}_{\mathcal{L}_{\text{diff}}} \right].$$

Given the fixed nature of process q and distribution $p_\theta(x^T) = p(x^T)$, the prior term $\mathcal{L}_{prior}$ can be disregarded as it does not depend on parameters θ. Since $\log p_\theta(x|x^0)$ is often modeled by a Gaussian distribution with low variance, the reconstruction term $\mathcal{L}_{rec}$ also remains unaffected by θ. Consequently, the diffusion term $\mathcal{L}_{diff}$ is the only part that the model parameters θ influence. $\mathcal{L}_{diff}$ is the sum of Kullback–Leibler (KL) divergences between the posterior distribution in the forward process $q(x^{t-1}|x^t, x)$ and the assumed normal distribution $p_\theta(x^{t-1}|x^t) = \mathcal{N}(x^{t-1}; \mu_\theta(x^t, t), \Sigma_\theta(x^t, t))$ in the reverse process. Here, the variance $\Sigma_\theta(x^t, t)$ is set to $\sigma_t^2 I$, while the mean $\mu_\theta(x^t, t)$ is learned by a neural network with parameters θ. This process is typically viewed as a noise estimation or data prediction problem (Benny & Wolf, 2022). To estimate noise, the network ϵ_θ forecasts the noise in the diffused input x^t and then computes $\mu_\theta(x^t, t)$ using the formula $\left(\frac{1}{\sqrt{\alpha_t}}x^t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}}\epsilon_\theta(x^t, t)\right)$. The parameter θ is learned by minimizing the loss function $\mathcal{L}_\epsilon = \mathbb{E}_{x,\epsilon,t}\left[\|\epsilon - \epsilon_\theta(x^t, t)\|^2\right]$. Alternatively, in the data prediction approach, a denoising network x_θ is employed to derive an estimate $x_\theta(x^t, t)$ of the clean data x^0 from x^t, and then set to

$$\mu_\theta(x^t, t) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x^t + \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_\theta(x^t, t) \,.$$

Here, the parameter θ is learned by minimizing the loss..

$$\mathcal{L}_x = \mathbb{E}_{x,\epsilon,t}\left\|x - x_\theta(x^t, t)\right\|^2 .$$

For time series diffusion models, forecasting x_θ has been found to be more effective than predicting ϵ_θ, as shown in (Feng et al., 2024; Shen & Kwok, 2023).

B.2 Conditional Diffusion models for time series modeling

In time series analysis, the prediction target x_{output} depends on the specific task at hand. For forecasting, the goal is to predict future values $x_{1:H}$ given past observations $x_{-L+1:0} \in \mathbb{R}^{d\times L}$, where L is the lookback window, H is the forecast horizon, and d is the number of variables. For imputation, the aim is to reconstruct missing regions within the input sequence. For classification, the model learns a latent representation of the input time series that is subsequently mapped to class labels through a classification head. For anomaly detection, the objective is to identify irregular or abnormal patterns within the sequence.

When employing conditional diffusion models for these tasks, the generative distribution is expressed as $p_\theta(x^{0:T}_{\text{output}}|c) = p_\theta(x^T_{\text{output}})\prod_{t=1}^T p_\theta(x^{t-1}_{\text{output}}|x^t_{\text{output}}, c)$, where $x^T_{\text{output}} \sim \mathcal{N}(0, I)$ and c denotes the condition, which varies according to the task (e.g., past observations for forecasting, masked inputs for imputation, the full sequence for classification and anomaly detection) (Rasul et al., 2021; Shen & Kwok, 2023; Shen et al., 2024; Li et al., 2024b).

At each step t, the denoising transition is modeled as $p_\theta(x^{t-1}_{\text{output}}|x^t_{\text{output}}, c) = \mathcal{N}\left(x^{t-1}_{\text{output}}; \mu_\theta(x^t_{\text{output}}, t|c), \sigma_t^2 I\right)$. During inference, generation begins with $\hat{x}^T_{\text{output}} \sim \mathcal{N}(0, I)$. By iteratively applying the denoising step until $t = T$, we obtain the final prediction $\hat{x}_{\text{output}}$, which corresponds to future values in forecasting, completed sequences in imputation, class-discriminative features in classification, or anomaly-indicative representations in detection tasks.

B.3 Rational Adaptive Correlation Projection (RACP)

The Kolmogorov-Arnold representation theorem Hecht-Nielsen (1987) states that any multivariate continuous function f, defined on a bounded domain, can be expressed as a finite composition of continuous univariate functions and addition. Specifically, for a smooth function $f : [0, 1]^n \to \mathbb{R}$, it can be represented as:

$$f(x_1, \ldots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \Phi_{q,p}(x_p) \right).$$

Here, each function $\Phi_{q,p} : [0, 1] \to \mathbb{R}$ and $\Phi_q : \mathbb{R} \to \mathbb{R}$ are continuous. This means that the $(2d+1)(d+1)$ univariate functions Φ_q and $\phi_{q,p}$ are enough for an exact representation of a d-variate function. Unlike traditional MLPs, which employ fixed activation functions at each node, (Liu et al., 2024b) define a generalized Kolmogorov-Arnold layer which replaces each weight parameter with a univariate function. The resulting functional form for deeper KAN can be expressed as : $\text{KAN}(x) = (\Phi_{L-1} \circ \Phi_{L-2} \circ \cdots \circ \Phi_0)(x)$ where Φ_l, $l \in [0, 1, \cdots, L-1]$ is a KAN layer, and the output dimension of each KAN layer can be expressed as: $[n_0, n_1, \cdots, n_{L-1}]$. Therefore, the transform process of j-th feature in l-th layer can be formed as:$\mathbf{x}_{l,j} = \sum_{i=1}^{n_{l-1}} \Phi_{l-1,j,i}(\mathbf{x}_{l-1,i}), \quad j = 1, \cdots, n_l.$ where Φ consists of two parts: the spline function and the residual activation function with learnable parameters w_a, w_b: $\Phi(x) = w_a \,\text{SiLU}(x) + w_b \,\text{Spline}(x)$. where $\text{Spline}(\cdot)$ is a linear combination of B-spline functions $\text{Spline}(x) = \sum_i c_i B_i(x)$.

In our method, we use the rational function (Boullé et al., 2020; Aghaei, 2024) as the base function for the KAN layer, instead of the B-spline. Specifically, we parameterize the function $\phi(x)$ on each edge as a rational function over polynomials $P(x)$ and $Q(x)$ of order m and n:

$$\phi(x) = wF(x) = w\frac{P(x)}{Q(x)} = w\frac{a_0 + a_1 x + \cdots + a_m x^m}{b_0 + b_1 x + \cdots + b_n x^n}$$

Here, a_i and b_j are coefficients of the rational function, and w is a scaling factor. This function is said to have degree m/n. We aim to learn the coefficients $\{a_i\}$, $\{b_j\}$, and w through end-to-end backpropagation.

To avoid instability caused by poles, where $Q(x) \to 0$ and $\phi(x) \to \pm\infty$, we employ a Safe Padé Activation Unit (PAU) (Molina et al., 2019) as our basis, which is a modified form of the standard

rational function:

$$F(x) = \frac{a_0 + a_1 x + \cdots + a_m x^m}{1 + |b_1 x + \cdots + b_n x^n|}$$

C Formal Derivations

C.1 Forward Posterior

Given:

$$q_\phi(\mathbf{x}^t|\mathbf{x}, c) = \mathcal{N}(\mathbf{x}^t; \sqrt{\bar{\alpha}_t} RACP_\phi(\mathbf{x}, t) + (1 - \sqrt{\bar{\alpha}_t})c, (1 - \bar{\alpha}_t)I) \tag{4}$$

From (4), We can write,

$$\begin{aligned}
\mathbf{x}^t &= \sqrt{\bar{\alpha}_t} RACP_\phi(\mathbf{x}, t) + (1 - \sqrt{\bar{\alpha}_t})c + \sqrt{1 - \bar{\alpha}_t}\epsilon \\
&= \sqrt{\bar{\alpha}_t} RACP_\phi(\mathbf{x}, t) + (1 - \sqrt{\bar{\alpha}_t})c + \sqrt{1 - \alpha_t \bar{\alpha}_{t-1}}\epsilon \\
&= \sqrt{\bar{\alpha}_t} RACP_\phi(\mathbf{x}, t) + (1 - \sqrt{\bar{\alpha}_t})c + \sqrt{1 - \alpha_t} \epsilon + \sqrt{\alpha_t - \alpha_t \bar{\alpha}_{t-1}}\epsilon \\
&= \sqrt{\bar{\alpha}_t} RACP_\phi(\mathbf{x}, t) + (1 - \sqrt{\bar{\alpha}_t})c + \sqrt{1 - \alpha_t} \epsilon + \sqrt{\alpha_t}\sqrt{1 - \bar{\alpha}_{t-1}}\epsilon \\
&= \sqrt{\bar{\alpha}_t} RACP_\phi(\mathbf{x}, t) + (1 - \sqrt{\bar{\alpha}_t})c + \sqrt{1 - \alpha_t} \epsilon \\
&+ \sqrt{\alpha_t} \left(\mathbf{x}^{t-1} - \sqrt{\bar{\alpha}_{t-1}} RACP_\phi(\mathbf{x}, t-1) - (1 - \sqrt{\bar{\alpha}_{t-1}})c \right) \\
&= \sqrt{\alpha_t} \mathbf{x}^{t-1} + \sqrt{\bar{\alpha}_t} \left(RACP_\phi(\mathbf{x}, t) - RACP_\phi(\mathbf{x}, t-1) \right) + (1 - \sqrt{\alpha_t})c + \sqrt{1 - \alpha_t}\epsilon
\end{aligned} \tag{5}$$

By introducing nonlinear, time-dependent projection of data, along with condition, results in a forward process that is non-Markovian as:

$$\begin{aligned}
q_\phi(\mathbf{x}^t|\mathbf{x}^{t-1}, \mathbf{x}, c) = \mathcal{N}(\mathbf{x}^t; &\sqrt{\alpha_t}\mathbf{x}^{t-1} + \sqrt{\bar{\alpha}_t} \left(RACP_\phi(\mathbf{x}, t) - RACP_\phi(\mathbf{x}, t-1) \right) \\
&+ (1 - \sqrt{\alpha_t})c, (1 - \alpha_t)I)
\end{aligned} \tag{6}$$

From (6 and 4), Posterior distribution can be derived as:

$$\begin{aligned}
q_\phi \left(\mathbf{x}^{t-1}|\mathbf{x}^t, \mathbf{x}, c \right) &\propto \frac{q_\phi(\mathbf{x}^{t-1}, \mathbf{x}^t, \mathbf{x}, c)}{q_\phi(\mathbf{x}^t, \mathbf{x}, c)} \\
&\propto \frac{q_\phi(\mathbf{x}^t|\mathbf{x}^{t-1}, \mathbf{x}, c)}{q_\phi(\mathbf{x}^t, \mathbf{x}, c)} q_\phi(\mathbf{x}^{t-1}, \mathbf{x}, c) \\
&\propto q_\phi(\mathbf{x}^t|\mathbf{x}^{t-1}, \mathbf{x}, c) q_\phi(\mathbf{x}^{t-1}|\mathbf{x}, c)
\end{aligned} \tag{7}$$

$$\begin{aligned}
&\propto \exp\left(-\frac{1}{2} \left(\frac{\left(\mathbf{x}^t - \sqrt{\alpha_t}\mathbf{x}^{t-1} - \sqrt{\bar{\alpha}_t}(RACP_\phi(\mathbf{x}, t) - RACP_\phi(\mathbf{x}, t-1)) - (1 - \sqrt{\alpha_t})c \right)^2}{1 - \alpha_t} \right.\right. \\
&\left.\left. + \frac{\left(\mathbf{x}^{t-1} - \sqrt{\bar{\alpha}_{t-1}} RACP_\phi(\mathbf{x}, t-1) - (1 - \sqrt{\bar{\alpha}_{t-1}})c \right)^2}{1 - \bar{\alpha}_{t-1}} \right) \right) \\
&\propto \exp\left(-\frac{1}{2} \left((\frac{\alpha_t}{1 - \alpha_t} + \frac{1}{1 - \bar{\alpha}_{t-1}})(\mathbf{x}^{t-1})^2 \right.\right. \\
&- \frac{\left(2\sqrt{\alpha_t}\mathbf{x}^{t-1}(\mathbf{x}^t - \sqrt{\bar{\alpha}_t}(RACP_\phi(\mathbf{x}, t) - RACP_\phi(\mathbf{x}, t-1)) - (1 - \sqrt{\alpha_t})c \right)}{1 - \alpha_t} \\
&\left.\left. - \frac{2\mathbf{x}^{t-1}}{1 - \bar{\alpha}_{t-1}} \left(\sqrt{\bar{\alpha}_{t-1}} RACP_\phi(\mathbf{x}, t-1) + (1 - \sqrt{\bar{\alpha}_{t-1}})c \right) \right) \right)
\end{aligned} \tag{8}$$

By reformulating the above propotional form, akin to normal distribution, we can see that,

$$q_\phi(\mathbf{x}^{t-1}|\mathbf{x}^t, \mathbf{x}, c) = \mathcal{N}(\mathbf{x}^{t-1}; \mu_{t-1|t}, \sigma^2_{t-1|t} I) \tag{9}$$

Where,

$$\mu_{t-1|t} = \left(\frac{\sqrt{\alpha_t}\mathbf{x}^t - \sqrt{\bar{\alpha}_t}\sqrt{\alpha_t}(RACP_\phi(\mathbf{x}, t) - RACP_\phi(\mathbf{x}, t-1)) - \sqrt{\alpha_t}(1-\sqrt{\alpha_t})c}{1-\alpha_t} + \frac{\sqrt{\bar{\alpha}_{t-1}} RACP_\phi(\mathbf{x}, t-1) + (1-\sqrt{\bar{\alpha}_{t-1}})c}{1-\bar{\alpha}_{t-1}} \right) \sigma^2_{t-1|t} \tag{10}$$

$$\sigma^2_{t-1|t} = \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \tag{11}$$

Mean($\mu_{t-1|t}$) can be simplified as, $\mu_{t-1|t} =$

$$\begin{aligned}
&\left(\frac{\sqrt{\alpha_t}\mathbf{x}^t - \sqrt{\bar{\alpha}_t}\sqrt{\alpha_t}(RACP_\phi(\mathbf{x}, t) - RACP_\phi(\mathbf{x}, t-1)) - \sqrt{\alpha_t}(1-\sqrt{\alpha_t})c}{(1-\alpha_t)} \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \right. \\
&\qquad \left. + \frac{\sqrt{\bar{\alpha}_{t-1}} RACP_\phi(\mathbf{x}, t-1) + (1-\sqrt{\bar{\alpha}_{t-1}})c}{(1-\bar{\alpha}_{t-1})} \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \right) \\
&= \left(\sqrt{\alpha_t}\mathbf{x}^t - \sqrt{\bar{\alpha}_t}\sqrt{\alpha_t}(RACP_\phi(\mathbf{x}, t) - RACP_\phi(\mathbf{x}, t-1)) - \sqrt{\alpha_t}(1-\sqrt{\alpha_t})c \right) \frac{(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \\
&\qquad + \left(\sqrt{\bar{\alpha}_{t-1}} RACP_\phi(\mathbf{x}, t-1) + (1-\sqrt{\bar{\alpha}_{t-1}})c \right) \frac{(1-\alpha_t)}{1-\bar{\alpha}_t}
\end{aligned} \tag{12}$$

$$\mu_{t-1|t} = \zeta_1 \mathbf{x}^t + \zeta_2 c - RACP_\phi(\mathbf{x}, t)\zeta_1\sqrt{\bar{\alpha}_t} + RACP_\phi(\mathbf{x}, t-1)(\zeta_1\sqrt{\bar{\alpha}_t} + \zeta_0)$$

Where,

$$\zeta_0 = \frac{1-\alpha_t}{1-\bar{\alpha}_t}\sqrt{\bar{\alpha}_{t-1}}; \qquad \zeta_1 = \frac{(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\sqrt{\alpha_t}; \qquad \zeta_2 = \left(1 + \frac{(\sqrt{\bar{\alpha}_t}-1)(\sqrt{\alpha_t}+\sqrt{\bar{\alpha}_{t-1}})}{1-\bar{\alpha}_t}\right);$$

C.2 Loss Formulation

Although our intention was not to explicitly make the forward process non-Markovian, our formulation has resulted in a non-Markovian forward process Eq.6. So, we can make use of joint distribution from DDIM (Song et al., 2020) as:

$$q(x^{0:T}|x) = q(x^T|x) \prod_{t=1}^{T} q(x^{t-1}|x^t, x),$$

Next, we define Non-Markovian trainable generative process $p_\theta(x_{0:T})$ as:

$$p_\theta(x^{0:T}) = p(x^T) \prod_{t=1}^{T} p_\theta(x^{t-1}|x^t) p_\theta(x^0|x^t), \tag{13}$$

(14)

This formulation is similar to Equation (55) in DDIM (Song et al., 2020). However, the objective in the referenced work is to minimize the number of sampling time steps; therefore, certain time steps are employed for image generation, and others are included in the variational objective, ultimately demonstrating that the loss formulation aligns with DDPM.

Our definition of trainable reverse generative process (14) is a result of introducing a RACP in the marginal distribution, which makes it challenging to learn from the reverse Markovian process. However, even without non-linearity, the resulting loss will be similar to that of DDPM,

Now, by incorporating this into our variational loss term, the variational loss is given as:

$$\mathbb{E}_{q_\phi}\Bigg[\underbrace{D_{\mathrm{KL}}\left(q_\phi\left(\mathbf{x}^T|\mathbf{x},c\right)\|p\left(\mathbf{x}^T|c\right)\right)}_{\mathcal{L}_{\text{prior}}}-\underbrace{\sum_{t=1}^{T}\log p_\theta\left(\mathbf{x}|\mathbf{x}^t,c\right)}_{\mathcal{L}_{\text{rec}}}$$
$$+\underbrace{\sum_{t=1}^{T}D_{\mathrm{KL}}\left(q_\phi\left(\mathbf{x}^{t-1}|\mathbf{x}^t,\mathbf{x},c\right)\|p_\theta\left(\mathbf{x}^{t-1}|\mathbf{x}^t,c\right)\right)}_{\mathcal{L}_{\text{diff}}}\Bigg]. \tag{15}$$

C.3 Variational Objective

To calculate the diffusion term $\mathcal{L}_{diff}$ of the objective (3), we need to compute the KL divergence between the forward posterior distribution $q_\phi(\mathbf{x}^{t-1}|\mathbf{x}^t,\mathbf{x},c)$ and the reverse distribution $p_\theta(\mathbf{x}^{t-1}|\mathbf{x}^t,c)$. Since we use parameterization $p_\theta(\mathbf{x}^{t-1}|\mathbf{x}^t,c)=q_\phi(\mathbf{x}^{t-1}|\mathbf{x}^t,\hat{\mathbf{x}}_\theta(x^t,t),c)$, both of these distributions are normal distributions with the same variance, so we can evaluate the KL divergence between them analytically as follows:

$$D_{KL}\left(q_\phi(\mathbf{x}^{t-1}|\mathbf{x}^t,\mathbf{x},c)\|p_\theta(\mathbf{x}^{t-1}|\mathbf{x}^t,c)\right)=$$
$$\frac{1}{2\sigma^2_{t-1|t}}\Big\|\zeta_1\sqrt{\bar{\alpha}_t}\Big(RACP_\phi(\mathbf{x},t)-RACP_\phi(\hat{\mathbf{x}}_\theta(\mathbf{x}^t,t),t)\Big)$$
$$-(\zeta_1\sqrt{\bar{\alpha}_t}+\zeta_0)\Big(RACP_\phi(\hat{\mathbf{x}}_\theta(\mathbf{x}^t,t-1),t)-RACP_\phi(\mathbf{x},t-1)\Big)\Big\|_2^2. \tag{16}$$

We can compute the prior term as follows:

$$\begin{aligned}
&D_{KL}\left(q_\phi(\mathbf{x}^T|\mathbf{x},c)\|p(\mathbf{x}^T|c)\right)=\\
&\frac{1}{2}\left[\log\frac{|I|}{\sigma_T^2 I}-d+\mathrm{Tr}\{I^{-1}\sigma_T^2 I\}+\left\|\sqrt{\bar{\alpha}_T}RACP_\phi(\mathbf{x},T)+(1-\sqrt{\bar{\alpha}_T})c-c\right\|_2^2\right]\\
&=\frac{1}{2}\left[-d\log\sigma_T^2+d\sigma_T^2+\left\|\sqrt{\bar{\alpha}_T}RACP_\phi(\mathbf{x},T)-\sqrt{\bar{\alpha}_T}c\right\|_2^2\right]\\
&=\frac{1}{2}\left(d\left(\sigma_T^2-\log\sigma_T^2-1\right)+\bar{\alpha}_T\left\|RACP_\phi(\mathbf{x},T)-c\right\|_2^2\right).\\
&=\frac{1}{2}\bar{\alpha}_T\left\|RACP_\phi(\mathbf{x},T)-c\right\|_2^2.
\end{aligned} \tag{17}$$

The reconstruction term is considered for all latent variables similar to VAE(Kingma, 2013b).

D Rational Denoising Transformer Block Equations

Adaptive Layer Normalization: Following the success of adaLN-Zero in DiT (Peebles & Xie, 2023), we employ adaptive layer normalization (adaLN) for injecting conditional information. The conditioning vector, derived from timestep t and condition c, modulates both the normalization parameters:

$$\text{adaLN}(\mathbf{h},\mathbf{c})=\gamma_{\mathbf{c}}\odot\frac{\mathbf{h}-\mu(\mathbf{h})}{\sigma(\mathbf{h})}+\beta_{\mathbf{c}} \tag{18}$$

where $\gamma_{\mathbf{c}}$ and $\beta_{\mathbf{c}}$ are regression outputs from the conditioning vector $\mathbf{c}$.

Temporal Conditioning Mechanism: For time series analysis, we enhance the conditioning mechanism to specifically handle temporal dependencies. The timestep embedding t undergoes sinusoidal encoding followed by

$$\mathbf{e}_t=(\text{SinusoidalEmbed}(t)) \tag{19}$$

Scale and Shift Operations: We incorporate additional scale and shift operators within each transformer block, similar to recent advances in diffusion transformers (Liu et al., 2024a). These operations provide fine-grained control over feature magnitudes and enable more stable training dynamics as mentioned in Section 2.2.

The complete Rational Denoising Transformer block can be expressed as:

$$\mathbf{h}' = \text{adaLN}(\mathbf{h}, \mathbf{c}) \tag{20}$$

$$\mathbf{h}_{\text{attn}} = \text{MultiHeadAttention}(\mathbf{h}') + \mathbf{h} \tag{21}$$

$$\mathbf{h}_{\text{norm}} = \text{adaLN}(\mathbf{h}_{\text{attn}}, \mathbf{c}) \tag{22}$$

$$\mathbf{h}_{\text{out}} = \text{Rational}(\mathbf{h}_{\text{norm}}) + \mathbf{h}_{\text{attn}} \tag{23}$$

E Extended Dataset Details

E.1 Forecasting

These datasets are *(1)* Caiso[3] - hourly electricity loads over eight years from various California regions; *(2)* Traffic[4] - hourly road occupancy rates from San Francisco Bay area sensors; *(3)* Electricity[5]- hourly electricity use of 321 clients over two years; *(4)* Weather[6] - 10-minute meteorological data for 2020-2021; *(5)* NorPool[7] - eight years of hourly energy production data from various European countries; *(6)* Exchange[8] - daily exchange rates for eight countries: Australia, United Kingdom, Canada, Switzerland, China, Japan, New Zealand and Singapore Lai et al. (2018); *(7)* ETTh1 and *(8)* ETTm1[9] are benchmarks for China's electricity transformer temperature data, over two years, ETTh1 reported hourly and ETTm1 every 15 minutes Zhou et al. (2021). Additional details regarding the dataset's characteristics are available in Table 9.

Following Shen et al. (2024), we adopt a train:validation:test ratio of 7:1:2 for Exchange, Weather, Wind, Traffic and Electricity data sets, and 6:2:2 for ETTh1 and Ettm1 data sets. For Norpool, the training data consists of observations before April 1, 2020; validation data spans from April 1 to October 1, 2020; and testing data is after October 1, 2020. For Caiso, the training set includes data before January 1, 2020, the validation period extends from January 1 to October 1, 2020, and the test set is post-October 1, 2020. Given that data sets exhibit different sampling intervals, we focus on prediction tasks with suitable prediction lengths based on the characteristics of the dataset (Shen & Kwok, 2023; Shen et al., 2024).

Table 9: Dataset characteristics for Forecasting

dataset	dim	#observations	freq.	H (steps)
NorPool	18	70,128	1 hour	1 month (720)
Caiso	10	74,472	1 hour	1 month (720)
Weather	21	52,696	10 mins	1 week (672)
ETTm1	7	69,680	15 mins	2 days (192)
Wind	7	48,673	15 mins	2 days (192)
Traffic	862	17,544	1 hour	1 week (168)
Electricity	321	26,304	1 hour	1 week (168)
ETTh1	7	17,420	1 hour	1 week (168)
Exchange	8	7,588	1 day	2 weeks (14)

E.2 Imputation, Classification and Anomaly Detection

The descriptions of the datasets utilized for Imputation, Classification, and Anomaly Detection are presented in Table 10. These configurations are derived from TSLib (Wu et al., 2022; Wang et al., 2024b).

[3] https://www.energyonline.com/Data/

[4] https://pems.dot.ca.gov/

[5] https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

[6] https://www.bgc-jena.mpg.de/wetter/

[7] https://data.nordpoolgroup.com/power-system/production/

[8] https://github.com/laiguokun/multivariate-time-series-data

[9] https://github.com/zhouhaoyi/ETDataset

Table 10: Dataset descriptions. The dataset size is organized in (Train, Validation, Test).

Tasks	Dataset	Dim	Series Length	Dataset Size	Information (Frequency)
Imputation	ETTm1, ETTm2	7	96	(34465, 11521, 11521)	Electricity (15 mins)
	ETTh1, ETTh2	7	96	(8545, 2881, 2881)	Electricity (15 mins)
	Electricity	321	96	(18317, 2633, 5261)	Electricity (15 mins)
	Weather	21	96	(36792, 5217, 10540)	Weather (10 mins)
Classification (UEA)	EthanolConcentration	3	1751	(261, 0, 263)	Alcohol Industry
	FaceDetection	144	62	(5890, 0, 3524)	Face (250Hz)
	Handwriting	3	152	(150, 0, 850)	Handwriting
	Heartbeat	61	405	(204, 0, 205)	Heart Beat
	JapaneseVowels	12	29	(270, 0, 370)	Voice
	PEMS-SF	963	144	(267, 0, 173)	Transportation (Daily)
	SelfRegulationSCP1	6	896	(268, 0, 293)	Health (256Hz)
	SelfRegulationSCP2	7	1152	(200, 0, 180)	Health (256Hz)
	SpokenArabicDigits	13	93	(6599, 0, 2199)	Voice (1025Hz)
	UWaveGestureLibrary	3	315	(120, 0, 320)	Gesture
Anomaly Detection	SMD	38	100	(566724, 141681, 708420)	Server Machine
	MSL	55	100	(14653, 11664, 73729)	Spacecraft
	SMAP	25	100	(108146, 27037, 427617)	Spacecraft
	SWaT	51	100	(396000, 99000, 449919)	Infrastructure
	PSM	25	100	(105984, 26497, 87841)	Server Machine

Algorithm 1 `RAF-Diff` Training

Input: target $\mathbf{x} = x_{1:H}$, history $x_{-L+1:0}$, condition c
repeat
  $\epsilon \sim \mathcal{N}(0, I)$, $t \sim \text{Uniform}(\{1, \dots, T\})$
  $\mathbf{x}^t \sim q_\phi(\mathbf{x}^t \mid \mathbf{x}, c)$ ▷ via ϵ
  Compute $\mathcal{L}_{\texttt{RAF-Diff}}$ (Eq. 3)
  Gradient step on $\nabla_{\theta,\phi}\mathcal{L}_{\texttt{RAF-Diff}}$
until converged

Algorithm 2 `RAF-Diff` Inference

$\mathbf{x}^T \sim \mathcal{N}(c, I)$
for $t = T$ **to** 1 **do**
  $\epsilon \sim \mathcal{N}(0, I)$ if $t > 1$, else $\epsilon = 0$
  $\hat{\mathbf{x}} = \hat{\mathbf{x}}_\theta(\mathbf{x}^t, t, c)$
  $\mathbf{x}^{t-1} = \zeta_1 \mathbf{x}^t + \zeta_2 c - \zeta_1 \sqrt{\bar{\alpha}_t}\, \text{RACP}_\phi(\hat{\mathbf{x}}, t) + (\zeta_1 \sqrt{\bar{\alpha}_t} + \zeta_0)\, \text{RACP}_\phi(\hat{\mathbf{x}}, t-1) + \sigma_{t-1|t}\, \epsilon$
end for
return $\mathbf{x}^0$

F BASELINES

We have extensively evaluated our `RAF-Diff` model by benchmarking it against various baseline models using a range of methodologies across different tasks, from basis expansion methods to diffusion techniques. These evaluations include: (i) models based on diffusion (Shen et al., 2024; Shen & Kwok, 2023; Rasul et al., 2021; Tashiro et al., 2021; Lopez Alcaraz & Strodthoff, 2023; Rishi et al., 2025); (ii) basis expansion methods, as described in (Challu et al., 2023; Zhou et al., 2022a; Fan et al., 2022; Oreshkin et al., 2019); (iii) other generative approaches such as GAN and VAE, referenced in (Li et al., 2022; Rangapuram et al., 2023; Jeha et al., 2022). We have also compared `RAF-Diff` with transformer models (Shabani et al., 2023; Nie et al., 2023; Zhou et al., 2022b; Wu et al., 2021; Liu et al., 2022b; Zhou et al., 2021; Kitaev et al., 2020; Liu et al., 2022c; Woo et al., 2022; Wang et al., 2024c; Tan et al., 2024; Naghashi et al., 2025; Liu et al., 2023; Wu et al., 2022; Liu et al., 2022c; Das et al., 2023a; Woo et al., 2022; Kitaev et al., 2020; Zhang & Yan, 2023) and additional deep learning methodologies (Liu et al., 2022a; Zeng et al., 2023; Bahdanau et al., 2015; Zhou et al., 2024; Chen et al., 2023; Wang et al., 2024a; 2023; Zhang et al., 2022; Xu et al., 2021).

G ALGORITHM

The training (Algorithm 1) and inference (Algorithm 2) methods employed in our paper are as follows,

In addition, we have included DDPM inference (Alg 3) to highlight the parallels between our inference process and that of DDPM.

Algorithm 3 DDPM Inference

$\mathbf{x}^T \sim \mathcal{N}(0, I)$
for $t = T$ **to** 1 **do**
  $\epsilon \sim \mathcal{N}(0, I)$ if $t > 1$, else $\epsilon = 0$
  $\hat{\mathbf{x}} = \hat{\mathbf{x}}_\theta(\mathbf{x}^t, t)$
  $\mathbf{x}^{t-1} = \frac{\sqrt{\bar{\alpha}_{t-1}}\,\beta_t}{1-\bar{\alpha}_t}\hat{\mathbf{x}} + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}^t + \sigma_t\epsilon$
end for
return $\mathbf{x}^0$

Table 11: Task-wise generation targets and conditioning inputs used in RAF-Diff.

Task	Generation Target	Conditioning Input
Forecasting	Future observations, $\mathbf{x}_{H:H+L}$	Past observations, $\mathbf{x}_{0:H}$
Imputation	Full observations, $\mathbf{x}_{0:H}$	Masked observations with missing values, $\tilde{\mathbf{x}}_{0:H}$
Classification	Full observations used to generate class embeddings	Full observations, $\mathbf{x}_{0:H}$
Anomaly Detection	Full observations used to compute anomaly scores	Full observations, $\mathbf{x}_{0:H}$

H Implementation Details

The table 11 summarizes how RAF-Diff adapts its generation target and conditioning input to the objective of each time series task. While forecasting and imputation use different target-condition pairs, classification and anomaly detection use the full observed sequence to learn task-specific representations.

The MLP modules within the embeddings of the condition and noised input, as well as the decoder, are from TiDE (Das et al., 2023a). This MLP framework is acknowledged as an effective component for universal time series analysis models (Das et al., 2023b). In the R-DiT, we designed an enhanced version of DiT using rational function, with the diffusion time step adjusted by adaptive layer normalization (Peebles & Xie, 2023; Esser et al., 2024). Hyperparameters are tuned specifically regarding embedding layers, encoder layers, time steps and hidden dimensions while other hyperparameters remain constant as detailed in Section 3. Optimal hyperparameters are selected via cross-validation and summarized for various datasets and tasks in Table 12. In the condition network, for condition c, a dense layer is used for tasks such as forecasting and anomaly detection, while a transformer is utilized for imputation and classification tasks. The implementation of rational function is inspired by Yang & Wang (2024).

I Sensitivity Analysis

I.1 Diffusion Hyperparameters

Table 13 presents the metrics for different tasks involving `RAF-Diff`, dependent on T (diffusion timesteps). In Classification, the accuracy is evaluated, along with Forecasting via MSE and MAE, and for Anomaly Detection, both Accuracy and F1 score are assessed. Notably, setting T = 200 leads to improved outcomes in both classification and anomaly detection. Meanwhile, T = 100 is more effective for forecasting.

A quadratic variance schedule is employed to establish the variance of Gaussian noise β_k. In this study, the parameter β_T is selected from the set $\{0.001, 0.01, 0.1, 0.9\}$, while β_1 remains fixed at 10^{-4}. As shown in Table 14, the optimal setting with $\beta_T = 0.1$ consistently yields improved results.

I.2 Look back window for forecasting

Table 15 presents the prediction MSE and MAE for `RAF-Diff` utilizing various lengths of the lookback window. The values considered are L = $\{96, 192, 336, 720, 1440\}$. It is evident that on the datasets ETTh1, and weather (corresponding to 168-step, and 672-step-ahead predictions, respectively), satisfactory performance is achieved when L is set to 720.

Table 12: Best hyperparameters for all our time series tasks.

Task	Dataset	Timesteps	Hidden Dim	Depth	Emb	Heads
Forecasting	Weather	100	256	1	4	8
	Caiso	100	128	3	5	8
	ETTh1	100	128	3	5	8
	Exchange	200	64	2	2	4
	ECL	100	256	3	5	8
	Norpool	200	256	2	1	4
	ETTm1	100	128	1	4	8
	Traffic	200	256	3	2	8
Imputation	ETTh1	200	128	2	1	1
	ETTh2	200	128	2	1	1
	ETTm2	200	128	2	1	1
	ECL	200	128	2	1	4
	ETTm1	200	128	2	1	8
	weather	200	32	2	1	4
Classification	EthanolConcentration	200	128	4	4	8
	FaceDetection	200	256	1	4	8
	Handwriting	100	256	1	1	8
	Heartbeat	200	256	4	4	1
	PEMS-SF	200	256	1	1	8
	SelfRegulationSCP2	100	256	4	1	8
	SpokenArabicDigits	200	256	1	1	8
	UWaveGestureLibrary	200	128	2	1	8
	JapaneseVowels	1000	32	1	1	8
	SelfRegulationSCP1	200	256	1	1	8
Anomaly Detection	MSL	100	32	4	1	4
	PSM	100	64	1	1	8
	SMAP	100	64	4	1	4
	SMD	100	32	1	1	1
	SWaT	100	32	4	4	4

Table 13: Sensitivity Study with varying diffusion time steps

Diffusion time	Classification (Ethanol)	Anomaly Detection (MSL)		Forecasting (ETTm1)	
Datasets	Accuracy	Acc	F1	MSE	MAE
50	0.261	0.969	0.839	0.344	0.380
100	0.261	0.965	0.815	**0.337**	**0.375**
200	**0.316**	**0.979**	**0.897**	0.339	0.377
500	0.234	0.979	0.899	0.351	0.390

Table 14: Sensitivity study by varying variance upper bound (β_T)

Tasks (Datasets)	Classification (Ethanol)	Anomaly Detection (MSL)		Forecasting (ETTm1)	
β_T Metric	Accuracy	Acc	F1	MSE	MAE
0.001	0.246	0.974	0.868	0.345	0.381
0.01	0.261	0.973	0.867	0.339	0.376
0.1	**0.316**	**0.979**	**0.897**	**0.337**	**0.375**
0.9	0.250	0.964	0.806	0.337	0.376

Table 15: Predicting MSE and MAE for different Lookback windows (L) for Multivariate forecasting

Look back window	96		192		336		720		1440	
Datasets	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.518	0.487	0.445	0.443	0.428	0.439	**0.401**	**0.425**	0.426	0.443
Weather	0.365	0.368	0.342	0.362	0.331	0.361	**0.307**	**0.336**	0.394	0.412

J Computational Analysis

We evaluate the computational efficiency of our proposed `RAF-Diff` in terms of training time, inference time, and model size, with results summarized in Tables 16 and 17.

Importantly, `RAF-Diff` achieves competitive accuracy while maintaining a significantly smaller parameter footprint (0.5M parameters), which translates into faster inference and lower memory requirements compared to most diffusion-based baselines. These results highlight the potential of `RAF-Diff` as a lightweight yet effective framework for time series analysis, offering a promising trade-off between accuracy, efficiency, and scalability.

Table 16: Training time (ms) for different models and sequence lengths (H) for ETTh1 univariate. Times are reported for one iteration.

	$H = 96$	$H = 168$	$H = 192$	$H = 336$	$H = 720$	MSE
`RAF-Diff`	0.92	0.90	0.97	0.84	0.71	0.066
CN-Diff	0.21	0.26	0.27	0.31	0.39	0.066
mr-Diff	0.59	0.69	0.71	0.74	0.82	0.066
TimeDiff	0.71	0.75	0.77	0.82	0.85	0.066
TimeGrad	2.11	2.42	3.21	4.22	5.93	0.078
CSDI	5.72	7.09	7.59	10.59	17.21	0.083
SSSD	16.98	19.34	22.64	32.12	52.93	0.097

Table 17: Inference time (ms) for different models and sequence lengths (H) for ETTh1 univariate. Times are reported for full inference of one sample.

Model	Trainable Params	$H = 96$	$H = 168$	$H = 192$	$H = 336$	$H = 720$
`RAF-Diff`	0.5M	10.1	9.9	9.8	10.0	9.8
CN-Diff	1.1M	6.2	6.7	6.9	7.8	9.1
mr-Diff	1.4M	12.5	14.3	14.9	16.8	27.5
TimeDiff	1.7M	16.2	17.3	17.6	26.5	34.6
TimeGrad	3.1M	870.2	1620.9	1854.5	3119.7	6724.1
CSDI	10M	90.4	128.3	142.8	398.9	513.1
SSSD	32M	418.6	590.2	645.4	1054.2	2516.9

K Statistical Significance Analysis

Acknowledging that the performance of deep models in time series analysis can be affected by various random initializations. Table 18 presents the outcomes from various runs across several tasks, including forecasting, imputation, anomaly detection, and classification. In the forecasting task (ETTh1) and imputation task (ETTm1), we measure performance using the mean squared error (MSE). For anomaly detection (MSL), we utilize the F1-score for evaluation, and for classification (Hand Writing), accuracy is employed as the metric. We note that classification accuracy and the F1-score for anomaly detection are consistent across different runs, indicating stable and reliable performance reflecting that multiple runs are predicting same categories. Conversely, the MSE

for forecasting and imputation exhibits minor fluctuations across runs, highlighting the statistical significance of our model.

Table 18: Statistical significance analysis across five different seed runs over four different tasks

	Forecasting	Imputation	Anomaly Detection	Classification
Datasets(metric)	ETTh1(MSE)	ETTm1(MSE)	MSL(F1)	Hand Writing(Accuracy)
0	0.4016211	0.0214681	0.8974237	0.3822115
1	0.4016621	0.0213752	0.8974237	0.3822115
2	0.4016901	0.0215876	0.8974237	0.3822115
3	0.4016121	0.0209875	0.8974237	0.3822115
4	0.4016319	0.0223561	0.8974237	0.3822115
mean	0.40164346	0.0215549	0.8974237	0.3822115
std deviation	0.0000287742	0.0004482766	0.0	0.0

L Full Results

L.1 Forecasting

Table 19 illustrates the mean absolute error (MAE) outcomes for multivariate time series prediction tasks, with an average rank of 2.6. It is evident that `RAF-Diff` continues to demonstrate a superior overall performance for MAE as the evaluation metric.

L.2 Imputation

Table 20 and 21 provides a comprehensive analysis of the MSE and MAE of several imputation models across various datasets with differing levels of missing data (12.5%, 25%, 37.5% and 50%). Our model's excellence is evident in the "Avg Rank" row, where it secures the top position with a rank of 1.5. These findings collectively underscore the robustness of our imputation method, particularly as the proportion of missing data increases.

L.3 Anomaly Detection

The results for the anomaly detection task are summarized in the table 22, which compares the performance of various models across multiple datasets, including MSL, PSM, SMAP, SMD and SWaT. The evaluation metrics used are Accuracy (A), Recall (R), and Precision (P), with higher values indicating superior performance. Our proposed model consistently demonstrates state-of-the-art performance, achieving the highest scores in accuracy, recall, and precision on nearly all datasets. This is further validated by its overall average rank of 2.2, which is significantly better than the next best model, TimesNet, at 3.8. These findings collectively highlight the robust and superior capability of our model in effectively identifying anomalies across diverse time series data.

M `RAF-Diff` to DDPM

In Appendix C, we explicitly derived each step used to obtain the loss formulation for our proposed method.
From equation 6 in Appendix C.1 we can clearly see.

$$q_\phi(\mathbf{x}^t|\mathbf{x}, c) = \mathcal{N}(\mathbf{x}^t; \sqrt{\bar{\alpha}}_t RACP_\phi(\mathbf{x}, t) + (1 - \sqrt{\bar{\alpha}}_t)c, (1 - \bar{\alpha}_t)I) \tag{24}$$

Table 19: Multivariate prediction of MAEs on eight real-world time series datasets (subscripts are the rank). CSDI and TiDE run out of memory on Traffic and Electricity. Results of all baselines are from Shen et al. (2024). Baselines with * are run using TSLib (Wang et al., 2024b; Wu et al., 2022)

Method	NorPool	Caiso	Traffic	Electricity	Weather	Exchange	ETTh1	ETTm1	Rank
Ours	0.565 (2)	**0.189** (1)	0.268 (2)	0.243 (2)	0.325 (4)	**0.079** (1)	0.425 (7)	0.372 (2)	2.6
CnDiff*	**0.554** (1)	0.191 (2)	0.270 (6)	0.243 (2)	0.324 (2)	**0.079** (1)	0.421 (3)	0.378 (7)	3.0
mr-Diff	0.604 (5)	0.219 (5)	0.320 (14)	0.252 (6)	0.324 (2)	0.082 (7)	0.422 (4)	0.373 (3)	5.8
TimeDiff	0.611 (6)	0.234 (12)	0.384 (20)	0.305 (18)	**0.312** (1)	0.091 (16)	0.430 (10)	**0.372** (1)	10.5
TimeGrad	0.821 (35)	0.339 (34)	0.849 (37)	0.630 (37)	0.381 (27)	0.193 (34)	0.719 (37)	0.605 (37)	34.8
CSDI	0.777 (32)	0.345 (35)	–	–	0.374 (25)	0.194 (35)	0.438 (12)	0.442 (27)	27.7
SSSD	0.753 (26)	0.295 (25)	0.398 (27)	0.363 (27)	0.350 (16)	0.127 (25)	0.561 (32)	0.406 (21)	24.9
D3VAE	0.692 (14)	0.331 (32)	0.483 (31)	0.372 (29)	0.380 (26)	0.301 (37)	0.502 (26)	0.391 (15)	26.2
CPF	0.889 (37)	0.424 (37)	0.714 (36)	0.643 (38)	0.781 (39)	0.082 (7)	0.597 (35)	0.472 (29)	32.2
PSA-GAN	0.890 (38)	0.477 (38)	0.697 (35)	0.533 (36)	0.578 (37)	0.087 (14)	0.546 (30)	0.488 (33)	32.6
N-Hits	0.643 (10)	0.221 (6)	0.268 (2)	0.245 (4)	0.335 (7)	0.085 (10)	0.480 (20)	0.388 (12)	8.9
FiLM	0.646 (11)	0.278 (22)	0.398 (27)	0.320 (20)	0.336 (8)	**0.079** (1)	0.436 (11)	0.374 (4)	13.0
Depts	0.611 (6)	0.204 (4)	0.568 (34)	0.401 (32)	0.394 (30)	0.100 (18)	0.491 (24)	0.412 (23)	21.4
NBeats	0.832 (36)	0.235 (13)	**0.265** (1)	0.370 (28)	0.420 (31)	0.081 (6)	0.521 (27)	0.409 (22)	20.5
Scaleformer	0.769 (30)	0.310 (27)	0.379 (18)	0.304 (17)	0.438 (32)	0.138 (27)	0.579 (34)	0.475 (30)	26.9
PatchTST	0.710 (20)	0.293 (24)	0.411 (30)	0.348 (26)	0.555 (36)	0.147 (28)	0.489 (23)	0.392 (16)	25.4
FedFormer	0.744 (23)	0.317 (29)	0.385 (21)	0.341 (24)	0.347 (14)	0.233 (36)	0.484 (21)	0.413 (24)	24.0
Autoformer	0.751 (25)	0.321 (30)	0.392 (26)	0.313 (19)	0.354 (17)	0.167 (30)	0.484 (21)	0.496 (34)	25.2
Pyraformer	0.781 (33)	0.371 (36)	0.390 (22)	0.379 (30)	0.385 (28)	0.112 (23)	0.493 (25)	0.435 (26)	27.9
Informer	0.757 (27)	0.336 (33)	0.391 (24)	0.383 (31)	0.364 (21)	0.192 (33)	0.605 (36)	0.542 (35)	30.0
Transformer	0.765 (28)	0.321 (30)	0.410 (29)	0.405 (33)	0.370 (24)	0.178 (32)	0.567 (33)	0.592 (36)	30.6
Crossformer	0.704 (19)	0.225 (8)	0.269 (5)	0.247 (5)	0.369 (23)	0.170 (31)	0.459 (17)	0.390 (14)	15.2
NonStationary*	0.581 (3)	0.232 (11)	0.382 (19)	0.281 (14)	0.356 (18)	0.105 (22)	0.528 (28)	0.443 (28)	17.9
Timexer*	0.647 (12)	0.230 (10)	0.275 (7)	0.252 (8)	0.333 (6)	0.086 (11)	0.429 (9)	0.376 (6)	8.6
Timesnet*	0.696 (15)	0.239 (14)	0.320 (15)	0.281 (15)	0.349 (15)	0.103 (21)	0.456 (16)	0.394 (18)	16.1
Multipatchformer*	0.701 (18)	0.245 (16)	0.292 (8)	0.256 (10)	0.342 (11)	0.086 (13)	0.420 (2)	0.387 (11)	11.1
PAttn*	0.698 (17)	0.248 (17)	0.317 (12)	0.267 (11)	0.341 (10)	0.080 (5)	0.429 (8)	0.384 (10)	11.2
Reformer*	0.717 (22)	0.272 (21)	0.390 (23)	0.409 (34)	0.613 (38)	0.650 (39)	0.740 (38)	0.689 (38)	31.6
iTransformer*	0.711 (21)	0.264 (19)	0.292 (9)	0.252 (7)	0.346 (13)	0.086 (12)	0.439 (13)	0.397 (19)	14.1
ETSFormer*	0.783 (34)	0.291 (23)	0.525 (33)	0.341 (25)	0.500 (34)	0.112 (24)	0.555 (31)	0.484 (32)	29.5
TiDE*	0.745 (24)	0.315 (28)	–	0.291 (16)	0.392 (29)	0.079 (4)	0.424 (6)	0.399 (20)	18.1
SCINet	0.601 (4)	0.193 (3)	0.335 (16)	0.280 (13)	0.344 (12)	0.137 (26)	0.463 (19)	0.389 (13)	13.2
NLinear	0.636 (8)	0.223 (7)	0.293 (10)	**0.239** (1)	0.328 (5)	0.091 (16)	**0.418** (1)	0.375 (5)	6.6
DLinear	0.640 (9)	0.497 (39)	0.268 (2)	0.336 (22)	0.444 (33)	0.102 (19)	0.442 (14)	0.378 (7)	18.1
LSTMa	0.974 (39)	0.305 (26)	0.510 (32)	0.444 (35)	0.501 (35)	0.534 (38)	0.782 (39)	0.699 (39)	35.4
MICN*	0.682 (13)	0.230 (9)	0.306 (11)	0.277 (12)	0.365 (22)	0.089 (15)	0.449 (15)	0.393 (17)	14.2
TSMixer*	0.774 (31)	0.244 (15)	0.369 (17)	0.336 (23)	0.357 (19)	0.149 (29)	0.538 (29)	0.477 (31)	24.2
LightTS*	0.765 (29)	0.253 (18)	0.391 (25)	0.327 (21)	0.362 (20)	0.102 (20)	0.462 (18)	0.418 (25)	22.0
TimeMixer*	0.698 (16)	0.265 (20)	0.319 (13)	0.255 (9)	0.341 (9)	0.084 (9)	0.422 (5)	0.381 (9)	11.2

From (24), We can write,

$$
\begin{aligned}
\mathbf{x}^t &= \sqrt{\bar{\alpha}_t} RACP_\phi(\mathbf{x}, t) + (1 - \sqrt{\bar{\alpha}_t})c + \sqrt{1 - \bar{\alpha}_t}\epsilon \\
&= \sqrt{\bar{\alpha}_t} RACP_\phi(\mathbf{x}, t) + (1 - \sqrt{\bar{\alpha}_t})c + \sqrt{1 - \alpha_t \bar{\alpha}_{t-1}}\epsilon \\
&= \sqrt{\bar{\alpha}_t} RACP_\phi(\mathbf{x}, t) + (1 - \sqrt{\bar{\alpha}_t})c + \sqrt{1 - \alpha_t}\epsilon + \sqrt{\alpha_t - \alpha_t \bar{\alpha}_{t-1}}\epsilon \\
&= \sqrt{\bar{\alpha}_t} RACP_\phi(\mathbf{x}, t) + (1 - \sqrt{\bar{\alpha}_t})c + \sqrt{1 - \alpha_t}\epsilon + \sqrt{\alpha_t}\sqrt{1 - \bar{\alpha}_{t-1}}\epsilon \\
&= \sqrt{\bar{\alpha}_t} RACP_\phi(\mathbf{x}, t) + (1 - \sqrt{\bar{\alpha}_t})c + \sqrt{1 - \alpha_t}\epsilon \\
&+ \sqrt{\alpha_t}\left(\mathbf{x}^{t-1} - \sqrt{\bar{\alpha}_{t-1}} RACP_\phi(\mathbf{x}, t-1) - (1 - \sqrt{\bar{\alpha}_{t-1}})c\right) \\
&= \sqrt{\alpha_t}\mathbf{x}^{t-1} + \sqrt{\bar{\alpha}_t}\left(RACP_\phi(\mathbf{x}, t) - RACP_\phi(\mathbf{x}, t-1)\right) + (1 - \sqrt{\alpha_t})c + \sqrt{1 - \alpha_t}\epsilon
\end{aligned}
\tag{25}
$$

Here, if we replace RACP = 1d and c = 0 in the above equation, we get

$$
\mathbf{x}^t = \sqrt{\alpha_t}\mathbf{x}^{t-1} + \sqrt{1 - \alpha_t}\epsilon \tag{26}
$$

$$
q_\phi(\mathbf{x}^t | \mathbf{x}, c) = \mathcal{N}(\mathbf{x}^t; \sqrt{\bar{\alpha}_t}\mathbf{x} + (1 - \bar{\alpha}_t)I) \tag{27}
$$

Which is DDPM formulation.

Table 20: Full results for the imputation task with MSE metrics. We randomly mask 12.5%, 25%, 37.5% and 50% time points to compare the model performance under different missing degrees. + in the Transformers indicates the name of + former .

Dataset		**Ours**	NS+	TimesNet	Cross+	LightTS	iTrans+	SSSD	Pyra+	Re+	Trans+	TiDE	DLinear	In+	ETS*	FED*	FiLM	Auto+
ECL	0.125	**0.060**	0.078	0.088	0.079	0.077	0.073	0.197	0.177	0.148	0.149	0.085	0.085	0.149	0.117	0.184	0.085	0.196
	0.25	**0.070**	0.085	0.091	0.092	0.099	0.090	0.205	0.184	0.155	0.161	0.114	0.113	0.165	0.127	0.206	0.114	0.215
	0.375	**0.079**	0.091	0.096	0.104	0.119	0.107	0.214	0.190	0.166	0.170	0.142	0.141	0.179	0.141	0.225	0.142	0.242
	0.5	**0.092**	0.099	0.102	0.114	0.137	0.128	0.217	0.204	0.178	0.178	0.174	0.173	0.193	0.158	0.247	0.174	0.269
	Avg	**0.075**	0.088	0.094	0.097	0.108	0.099	0.208	0.189	0.162	0.165	0.129	0.128	0.171	0.136	0.215	0.129	0.231
ETTh1	0.125	0.047	0.047	0.062	0.120	0.114	0.098	**0.046**	0.077	0.060	0.065	0.111	0.111	0.071	0.086	0.072	0.117	0.094
	0.25	**0.052**	0.064	0.080	0.139	0.146	0.125	0.065	0.102	0.084	0.118	0.150	0.150	0.095	0.118	0.103	0.154	0.118
	0.375	**0.077**	0.083	0.097	0.156	0.175	0.156	0.090	0.128	0.113	0.113	0.186	0.188	0.127	0.152	0.140	0.191	0.160
	0.5	**0.083**	0.107	0.116	0.176	0.203	0.212	0.110	0.156	0.146	0.135	0.229	0.229	0.165	0.196	0.197	0.233	0.217
	Avg	**0.065**	0.075	0.089	0.148	0.159	0.148	0.078	0.116	0.101	0.108	0.169	0.170	0.114	0.138	0.128	0.174	0.147
ETTh2	0.125	**0.029**	0.039	0.040	0.126	0.113	0.095	0.184	0.141	0.144	0.182	0.108	0.108	0.279	0.132	0.127	0.110	0.212
	0.25	**0.025**	0.046	0.047	0.143	0.140	0.119	0.336	0.186	0.201	0.252	0.143	0.144	0.354	0.189	0.176	0.150	0.219
	0.375	**0.037**	0.052	0.054	0.155	0.166	0.148	0.477	0.221	0.246	0.281	0.179	0.179	0.402	0.264	0.247	0.188	0.274
	0.5	**0.055**	0.060	0.061	0.181	0.189	0.192	0.642	0.257	0.313	0.308	0.221	0.217	0.430	0.351	0.348	0.227	0.360
	Avg	**0.037**	0.049	0.051	0.151	0.152	0.139	0.410	0.201	0.226	0.256	0.163	0.162	0.366	0.234	0.224	0.169	0.266
ETTm1	0.125	0.017	0.018	0.019	0.051	0.052	0.045	**0.016**	0.027	0.024	0.022	0.056	0.056	0.028	0.042	0.034	0.056	1.103
	0.25	0.021	0.022	0.023	0.054	0.062	0.060	**0.020**	0.034	0.033	0.030	0.076	0.076	0.040	0.061	0.051	0.077	0.811
	0.375	0.026	0.028	0.029	0.060	0.073	0.077	**0.024**	0.040	0.035	0.037	0.099	0.099	0.056	0.084	0.079	0.100	0.561
	0.5	**0.032**	0.034	0.036	0.067	0.086	0.101	0.032	0.049	0.047	0.045	0.128	0.128	0.073	0.104	0.125	0.129	0.484
	Avg	0.024	0.026	0.027	0.060	0.068	0.071	**0.023**	0.037	0.033	0.034	0.090	0.090	0.049	0.073	0.072	0.090	0.740
ETTm2	0.125	**0.011**	0.018	0.019	0.069	0.062	0.052	0.088	0.077	0.105	0.178	0.066	0.066	0.174	0.081	0.060	0.066	1.059
	0.25	**0.014**	0.020	0.021	0.069	0.073	0.070	0.128	0.101	0.142	0.215	0.089	0.088	0.244	0.115	0.090	0.089	0.957
	0.375	**0.018**	0.022	0.023	0.087	0.084	0.090	0.236	0.142	0.182	0.206	0.111	0.111	0.285	0.163	0.131	0.112	0.726
	0.5	**0.022**	0.025	0.026	0.091	0.087	0.117	0.158	0.192	0.230	0.213	0.138	0.138	0.277	0.209	0.237	0.138	0.705
	Avg	**0.016**	0.021	0.022	0.079	0.076	0.082	0.152	0.128	0.165	0.203	0.101	0.101	0.245	0.142	0.129	0.102	0.862
weather	0.125	0.028	0.025	**0.023**	0.039	0.038	0.038	0.036	0.029	0.031	0.030	0.038	0.038	0.036	0.036	0.042	0.041	0.304
	0.25	0.031	**0.027**	0.028	0.042	0.043	0.046	0.029	0.032	0.036	0.034	0.047	0.047	0.039	0.044	0.058	0.050	0.237
	0.375	0.032	0.031	0.031	0.044	0.048	0.055	**0.030**	0.036	0.040	0.040	0.056	0.056	0.044	0.052	0.077	0.062	0.151
	0.5	0.036	0.036	0.034	0.047	0.053	0.067	**0.033**	0.040	0.048	0.044	0.066	0.066	0.050	0.061	0.114	0.076	0.148
	Avg	0.032	0.030	**0.029**	0.043	0.046	0.051	0.032	0.034	0.039	0.036	0.052	0.052	0.042	0.048	0.073	0.057	0.210
Avg MSE		**0.041**	0.048	0.049	0.095	0.102	0.104	0.151	0.118	0.114	0.128	0.117	0.117	0.165	0.129	0.140	0.120	0.408
Avg Rank		**1.5**	2.2	3.0	7.3	8.3	8.5	8.8	9.0	9.0	9.8	10.8	11.2	11.3	11.5	12.5	12.5	15.7

Table 21: Full results for the imputation task with MAE metrics. We randomly mask 12.5%, 25%, 37.5% and 50% time points to compare the model performance under different missing degrees. + in the Transformers indicates the name of + former .

Dataset		**Ours**	NS+	TimesNet	Cross+	iTrans+	LightTS	SSSD	Re+	Pyra+	Trans+	DLinear	TiDE	In+	ETS+	FED+	FiLM	Auto+
ECL	0.125	**0.168**	0.193	0.203	0.199	0.190	0.197	0.304	0.276	0.298	0.277	0.207	0.207	0.276	0.246	0.322	0.207	0.333
	0.25	**0.184**	0.201	0.208	0.217	0.214	0.228	0.308	0.282	0.299	0.285	0.243	0.244	0.288	0.257	0.340	0.243	0.345
	0.375	**0.195**	0.209	0.213	0.231	0.235	0.252	0.312	0.290	0.303	0.291	0.273	0.274	0.297	0.270	0.354	0.274	0.365
	0.5	**0.212**	0.218	0.221	0.244	0.258	0.271	0.313	0.297	0.310	0.296	0.303	0.305	0.306	0.285	0.370	0.304	0.382
	Avg	**0.190**	0.205	0.211	0.223	0.224	0.237	0.309	0.286	0.302	0.287	0.256	0.257	0.292	0.264	0.347	0.257	0.356
ETTh1	0.125	0.152	**0.146**	0.168	0.237	0.219	0.235	0.151	0.172	0.200	0.181	0.232	0.231	0.188	0.209	0.194	0.238	0.222
	0.25	**0.156**	0.169	0.191	0.258	0.249	0.267	0.177	0.205	0.229	0.245	0.269	0.268	0.219	0.245	0.233	0.273	0.249
	0.375	**0.189**	0.192	0.209	0.277	0.279	0.294	0.205	0.239	0.259	0.238	0.302	0.299	0.252	0.277	0.275	0.303	0.288
	0.5	**0.195**	0.217	0.227	0.298	0.326	0.318	0.232	0.273	0.285	0.262	0.332	0.329	0.287	0.315	0.327	0.334	0.337
	Avg	**0.173**	0.181	0.199	0.268	0.268	0.278	0.191	0.222	0.243	0.232	0.284	0.282	0.236	0.261	0.257	0.287	0.274
ETTh2	0.125	**0.121**	0.130	0.131	0.237	0.210	0.227	0.310	0.268	0.276	0.311	0.222	0.222	0.389	0.256	0.242	0.224	0.311
	0.25	**0.111**	0.142	0.144	0.254	0.238	0.256	0.417	0.317	0.328	0.374	0.259	0.258	0.451	0.316	0.290	0.263	0.327
	0.375	**0.136**	0.152	0.155	0.267	0.265	0.280	0.501	0.355	0.350	0.389	0.289	0.289	0.480	0.375	0.340	0.295	0.359
	0.5	0.168	**0.164**	0.165	0.290	0.302	0.300	0.581	0.411	0.377	0.406	0.319	0.321	0.503	0.442	0.399	0.326	0.413
	Avg	**0.134**	0.147	0.149	0.262	0.254	0.266	0.452	0.338	0.333	0.370	0.272	0.273	0.456	0.347	0.318	0.277	0.353
ETTm1	0.125	**0.084**	0.090	0.092	0.158	0.147	0.156	0.088	0.110	0.113	0.103	0.161	0.161	0.117	0.144	0.130	0.162	0.875
	0.25	**0.091**	0.098	0.101	0.164	0.172	0.174	0.095	0.129	0.126	0.121	0.190	0.190	0.141	0.175	0.160	0.190	0.719
	0.375	**0.101**	0.110	0.111	0.171	0.195	0.190	0.103	0.130	0.139	0.134	0.217	0.217	0.168	0.205	0.196	0.217	0.565
	0.5	**0.110**	0.122	0.124	0.181	0.225	0.207	0.119	0.151	0.153	0.148	0.246	0.246	0.194	0.230	0.251	0.246	0.504
	Avg	**0.096**	0.105	0.107	0.171	0.184	0.182	0.101	0.127	0.133	0.126	0.204	0.203	0.155	0.189	0.185	0.204	0.666
ETTm2	0.125	**0.071**	0.078	0.082	0.174	0.151	0.165	0.210	0.230	0.200	0.310	0.171	0.171	0.300	0.199	0.166	0.171	0.819
	0.25	**0.080**	0.083	0.087	0.174	0.178	0.182	0.256	0.272	0.236	0.342	0.199	0.200	0.368	0.241	0.206	0.200	0.748
	0.375	0.091	**0.089**	0.092	0.196	0.203	0.196	0.341	0.318	0.272	0.348	0.225	0.225	0.406	0.294	0.244	0.226	0.630
	0.5	0.102	**0.096**	0.099	0.195	0.232	0.202	0.272	0.359	0.307	0.343	0.252	0.251	0.386	0.331	0.331	0.251	0.587
	Avg	**0.086**	0.087	0.090	0.185	0.191	0.186	0.270	0.295	0.254	0.336	0.212	0.212	0.365	0.266	0.237	0.212	0.696
weather	0.125	0.049	0.046	**0.046**	0.094	0.086	0.089	0.085	0.077	0.065	0.076	0.089	0.089	0.092	0.092	0.102	0.089	0.402
	0.25	0.054	**0.051**	0.052	0.101	0.105	0.102	0.067	0.087	0.075	0.086	0.108	0.108	0.094	0.109	0.133	0.111	0.334
	0.375	**0.056**	0.056	0.057	0.104	0.121	0.108	0.062	0.092	0.082	0.095	0.122	0.122	0.102	0.122	0.163	0.131	0.257
	0.5	0.062	0.064	**0.061**	0.110	0.141	0.114	0.065	0.105	0.090	0.100	0.136	0.137	0.107	0.138	0.209	0.152	0.254
	Avg	0.055	0.054	**0.054**	0.102	0.113	0.103	0.070	0.090	0.078	0.088	0.114	0.114	0.099	0.115	0.152	0.120	0.312
Avg MAE		**0.122**	0.130	0.133	0.201	0.213	0.209	0.232	0.219	0.224	0.229	0.224	0.223	0.267	0.241	0.249	0.226	0.443
Avg Rank		**1.3**	2.2	3.0	7.0	8.2	8.5	8.8	9.2	9.3	9.8	11.0	11.0	11.5	12.0	12.2	12.2	15.8

N VISUALISATIONS OF RACP COMPONENT

To understand what is learned by RACP, we use a trained model to investigate how features are correlated within the learned latent representation space of RACP at a randomly selected timestep of the diffusion model. We observed that there are increasing correlations between the features in learned

Table 22: Full results for the anomaly detection task. The A, R and P represent the accuracy, recall and precision (%) respectively. A higher value of A, R and P indicates a better performance.

Model	MSL			PSM			SMAP			SMD			SWaT			Average	
	A	R	P	A	R	P	A	R	P	A	R	P	A	R	P	A	Rank
Ours	97.91	86.85	92.82	97.83	93.34	98.30	97.93	90.62	93.09	98.76	76.92	92.01	97.83	82.92	98.78	**98.05**	**2.2**
TimesNet	96.47	75.29	89.55	98.57	96.30	98.52	93.62	56.41	90.01	98.77	81.54	87.89	98.15	93.54	91.45	97.12	3.8
SSSD	97.18	79.35	92.80	96.34	87.14	99.62	93.76	58.23	89.23	99.12	84.66	93.53	97.35	80.29	97.48	96.75	4.4
MICN	96.06	71.98	88.56	97.81	93.24	98.79	93.78	58.14	89.61	98.83	78.95	91.65	97.60	87.48	92.35	96.82	5.4
DLinear	96.48	75.30	89.69	98.17	94.69	98.65	93.51	56.38	88.82	98.71	75.84	91.65	96.75	80.36	91.88	96.72	7.4
iTransformer	95.00	62.63	86.15	96.99	90.26	98.77	93.62	56.33	90.07	98.61	73.45	91.31	97.47	86.29	92.39	96.34	8.2
AnomalyT	96.85	76.29	92.43	96.20	86.75	99.48	93.71	57.80	89.22	98.40	70.08	89.22	95.82	65.73	99.75	96.20	9.4
LightTS	96.33	74.08	89.32	96.54	88.54	98.84	93.56	55.80	90.12	98.69	75.41	91.55	96.25	69.16	99.93	96.27	9.4
FiLM	93.88	51.30	84.62	96.68	89.02	98.88	93.52	56.40	88.86	98.72	76.59	91.19	96.75	80.36	91.85	95.91	10.2
KANAD	96.42	72.82	91.43	95.15	83.54	98.81	93.56	56.31	89.43	98.71	76.25	91.27	95.83	65.73	99.95	95.93	10.8
Reformer	96.38	73.71	90.15	96.04	85.97	99.71	93.56	56.76	88.84	97.95	60.39	86.25	95.83	65.73	99.94	95.95	11.4
Informer	96.37	73.69	90.12	96.04	85.90	99.77	93.57	56.79	88.93	97.95	60.37	86.30	95.83	65.73	99.92	95.95	11.4
Crossformer	96.31	72.78	90.32	96.99	89.72	99.36	93.52	56.51	88.77	98.15	64.86	87.52	95.83	65.73	99.89	96.16	11.4
Transformer	96.33	73.66	89.70	95.24	83.15	99.62	94.27	61.37	90.86	97.80	65.24	78.32	95.87	66.10	99.81	95.90	11.6
Autoformer	96.63	75.80	90.71	94.16	78.95	99.99	94.38	62.53	90.65	97.81	65.10	78.45	95.81	65.56	99.96	95.76	11.8
Pyraformer	95.73	68.47	88.46	95.99	86.03	99.41	93.58	56.96	88.84	98.43	71.05	88.87	95.83	65.73	99.90	95.91	11.8
FEDformer	96.57	75.23	90.65	95.58	84.66	99.31	93.51	56.38	88.76	97.89	59.10	85.61	95.84	65.73	99.95	95.88	12.4
ETSformer	94.56	55.68	88.46	95.41	84.65	98.58	93.49	55.49	89.72	98.30	65.29	91.36	96.98	81.60	92.61	95.75	13.6

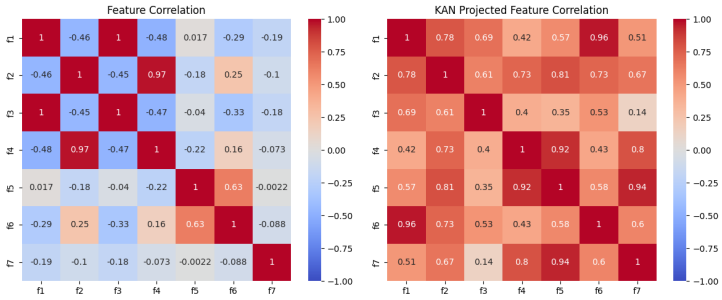


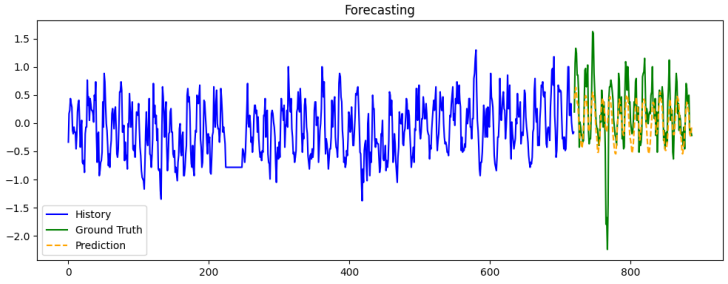
Figure 2: Correlation Matrix of Features for Actual Data and RACP Data in ETTm1 for the Forecasting Task.

latent space as shown in Figure 2. Thus, we hypothesize that this increased correlation and time-dependent adaptability perhaps facilitate a more effective diffusion process for time series analysis. Similar observations of correlation in latent space that help diffusion models have been made in image and video diffusion works (Ge et al., 2023). In the space of diffusion models for time series, Tashiro et al. (2021) has noted that, learning the correlation between feature and temporal space is necessary for time series imputation tasks. From this analysis, we hypothesize that incorporating a RACP in the forward process can markedly decrease the disparity between the log-likelihood and the variational bound.

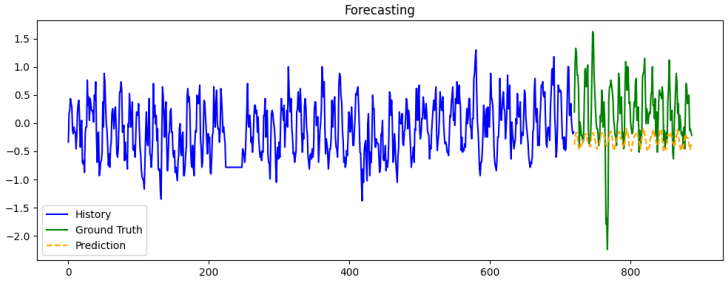
O Qualitative Results

O.1 Forecasting

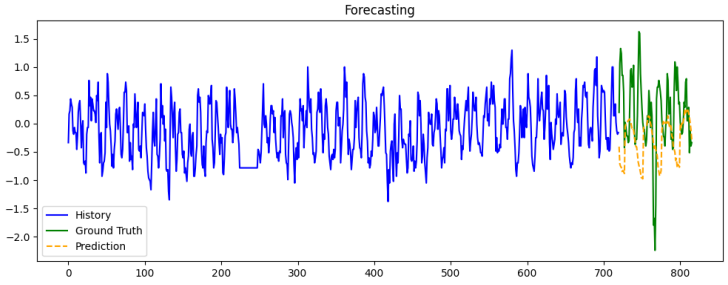
The forecasting visualizations in Figures 3 and 4 demonstrate the predictive capabilities of the proposed `RAF-Diff` model across various datasets. Figure 3 provides a comparative analysis on the ETTh1 dataset, where `RAF-Diff`'s predictions exhibit a high degree of accuracy, closely following the ground truth values and outperforming the forecasting accuracy of other baseline models such as TimesNet, Reformer, and Nonstationary Transformer. These results collectively validate the effectiveness of the `RAF-Diff` architecture for long-term time series forecasting. Also shown in Figure 4, our model effectively captures the underlying patterns and trends of distinct time series, including electricity consumption (ECL, Caiso, Norpool) and weather data. The close alignment between the ground truth (green line) and the prediction (dashed orange line) in the forecasting window suggests the model's ability of forecasting different datasets accurately.



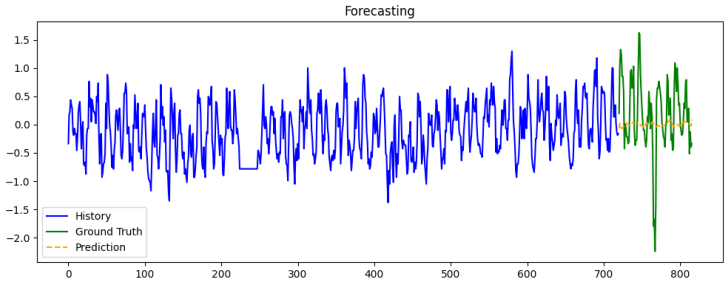
(a) RAF−Diff



(b) TimesNet



(c) Reformer

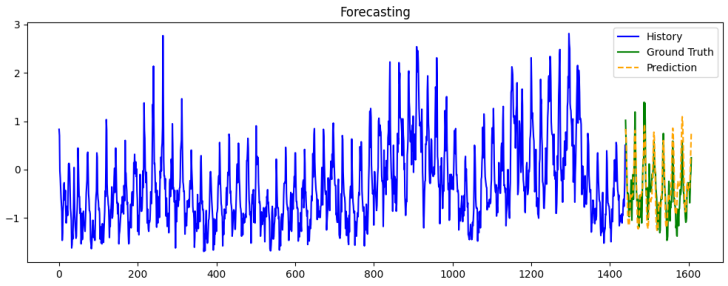


(d) Nonstationary Transformer

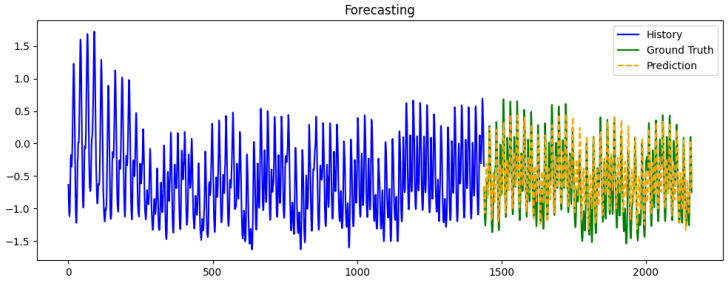
Figure 3: Forecasting visualizations of ETTh1 dataset predictions by (a) RAF−Diff, (b) TimesNet Wu et al. (2022), (c) Reformer Kitaev et al. (2020) and (d) Nonstationary Transformer Liu et al. (2022c)

O.2 Imputation

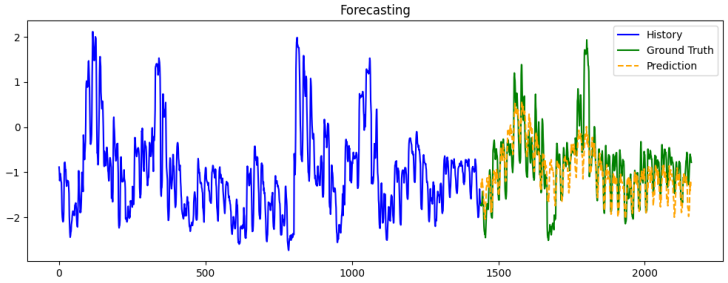
To evaluate the effectiveness of our proposed method for imputation, we present qualitative comparisons through imputation visualizations across multiple datasets and masking scenarios. Figure 5 demonstrate the imputation performance of various time series models like DLinear, Crossformer, NonStationary Transformer on the Ettm1 dataset and Figure 6 Weather, Etth1, Etth2 and ECL at 50% and 25% mask ratios respectively. These visualizations clearly show that our method achieves



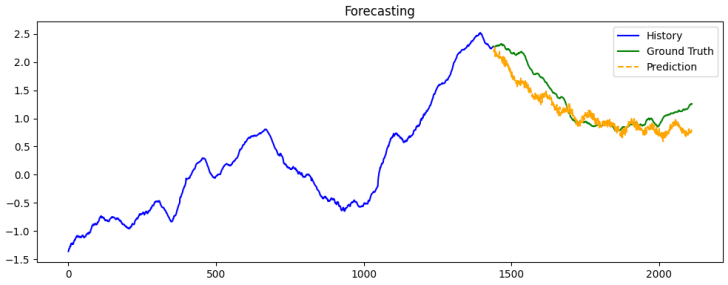
(a) ECL



(b) Caiso



(c) Norpool



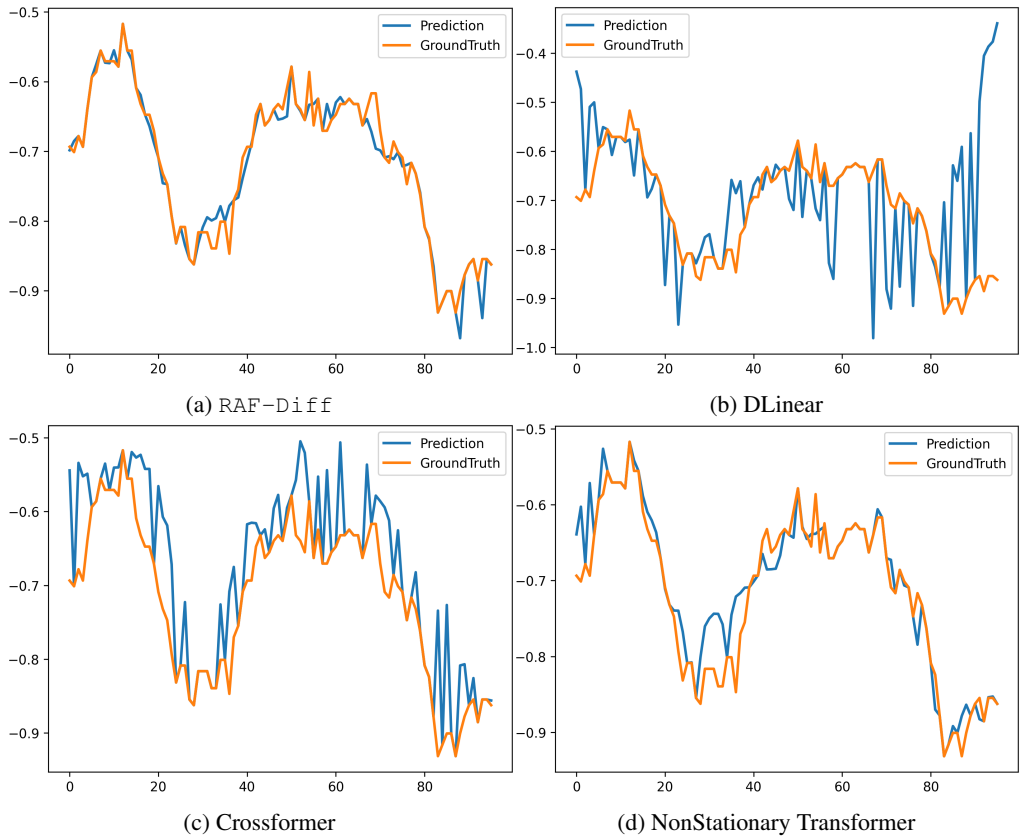
(d) Weather

Figure 4: Forecasting visualizations results of different datasets for `RAF-Diff`

superior imputation quality compared to baseline approaches, maintaining better temporal continuity and more accurately capturing the underlying patterns of the original time series.

O.3 Anomaly Detection

Figure 7 demonstrates the superior anomaly detection capabilities of our proposed method on the SMAP dataset across different baseline models including DLinear, Crossformer, and NonStationary Transformer. The visualization clearly shows that our method successfully identifies and highlights



(a) RAF−Diff (b) DLinear

(c) Crossformer (d) NonStationary Transformer

Figure 5: Imputation visualizations for ETTm1 dataset at 50% mask ratio for (a) Ours, (b) Dlinear (Zeng et al., 2023), (c) Crossformer (Zhang & Yan, 2023) and (d) NonStationary Transformer (Liu et al., 2022c)

the anomalous regions (indicated by the yellow shaded areas), while the baseline methods fail to detect these critical anomalies. This superior detection performance validates the effectiveness of our method in capturing meaningful temporal dependencies essential for accurate anomaly identification in time series data.

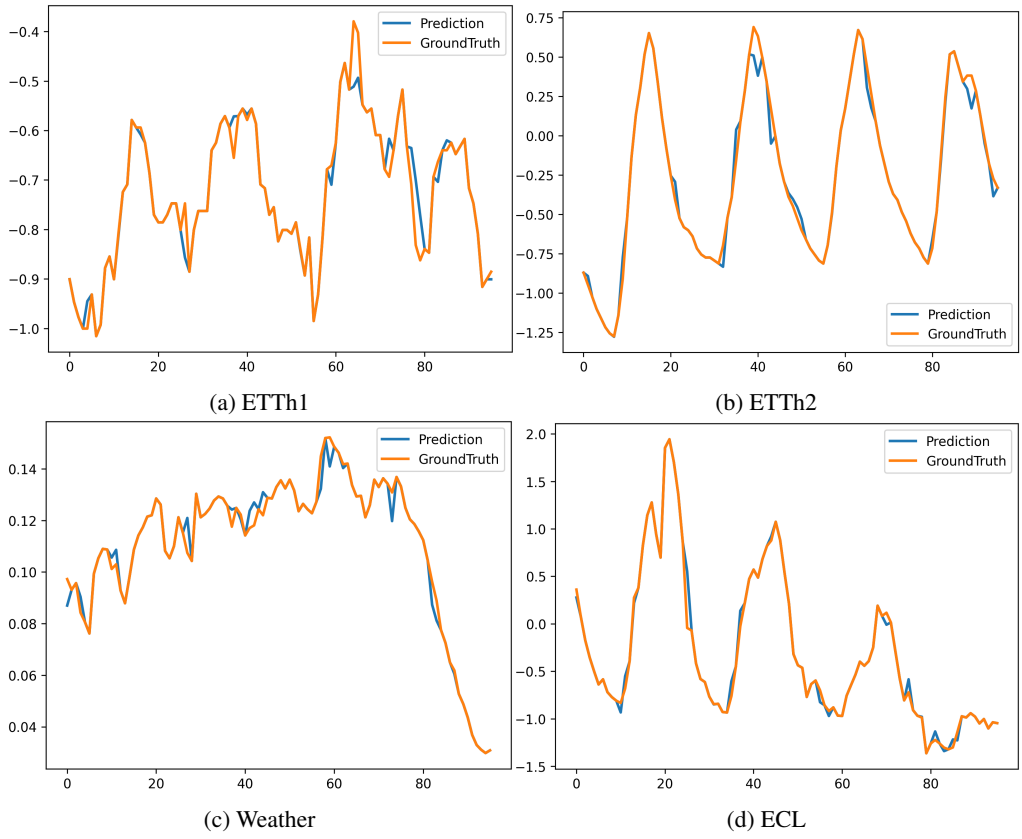
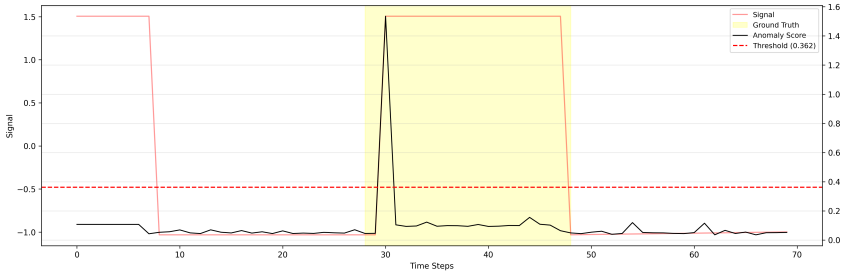
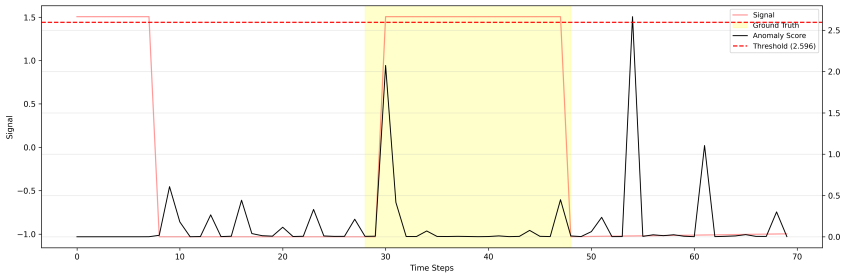


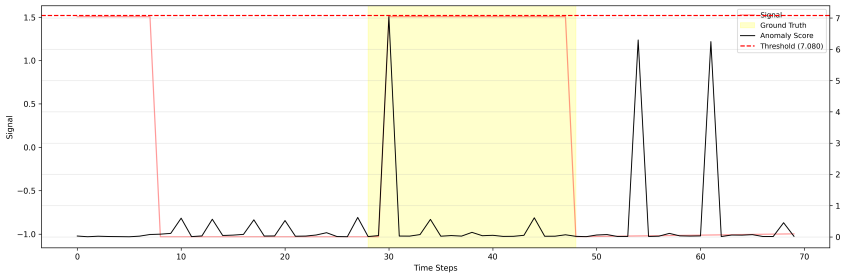
Figure 6: Imputation visualizations of `RAF-Diff` for ETTh1, ETTh2, Weather and ECL datasets at 25% mask ratio.



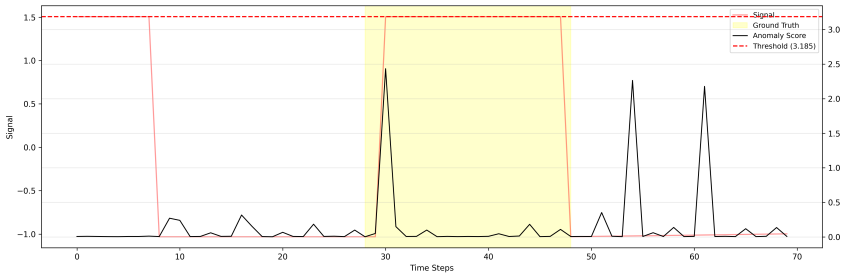
(a) RAF−Diff



(b) DLinear



(c) Crossformer



(d) NonStationary Transformer

Figure 7: Anomaly Detection visualizations for SMAP dataset for (a) Ours, (b) Dlinear (Zeng et al., 2023), (c) Crossformer (Zhang & Yan, 2023) and (d) NonStationary Transformer (Liu et al., 2022c)