ROSFORMER: ROBUST STOCHASTIC CONTINUOUS-TIME TRANSFORMER FOR IRREGULAR MULTIVARIATE TIME SERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

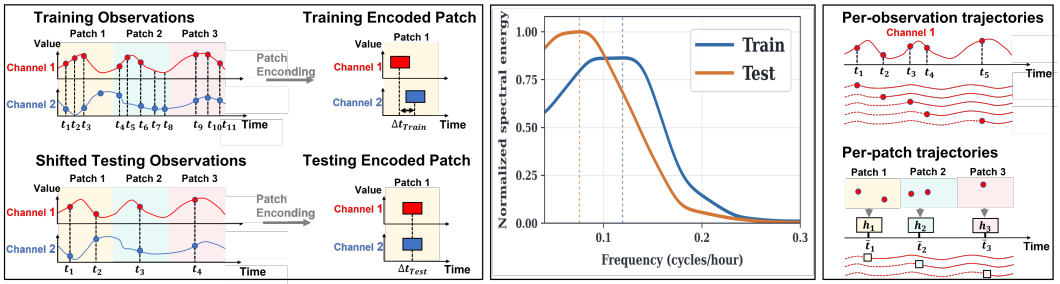
Irregular multivariate time series (IMTS) forecasting is a challenging problem due to the distribution shift incurred by changes in the observation density and missingness patterns. A stochastic continuous-time Transformer incorporates the linear-noise stochastic differential equations (LNSDEs) with a continuous-time attention to attenuate the influence of input perturbations and capture dependencies across observations, which, however, requires a costly numerical calculation of the integration. Patch-based methods, on the other hand, can reduce computation by compressing the local observations into patch embeddings, while the patch compression alone does not ensure robustness to such a shift. This thus motivates us to incorporate the patch-based modeling with a stochastic continuous-time Transformer for a robust and efficient forecasting under distribution shifts. Such an incorporation is non-trivial, as it requires adapting the discrete patch representations to continuous-time modeling, while preserving robustness to input perturbations across both the trajectory evolution and attention aggregation. In this paper, we propose RosFormer, a robust stochastic continuous-time Transformer for an efficient IMTS forecasting under distribution shifts. RosFormer captures the inter-variable dependencies within each patch through a relative-time-aware graph, and uses the resulting embeddings to initialize the LNSDE trajectories at their temporal anchors. These trajectories are then aggregated through compactness-aware attention, with the queries, keys and values sharing one trajectory per variable-level patch to reduce numerical integration costs. We further theoretically analyze the stability of the attention output under bounded perturbations of the initial patch embeddings. Experiments demonstrate the robustness of RosFormer to distribution shifts in the controlled test-time missingness and hospital-disjoint eICU-CRD settings, along with a competitive forecasting and event prediction performance and lower computational cost than the other continuous-time baselines.

1 INTRODUCTION

Irregular multivariate time series (IMTS) analysis plays an important role in many real-world applications, such as healthcare monitoring, climate dynamics modeling, and human activity analysis (De Brouwer et al., 2019; Yao et al., 2018). Different from the regular time series, IMTS contain multiple variables observed asynchronously at nonuniform timestamps, often with missing observations. In practice, changes in the observation density and missingness patterns may introduce input perturbations, leading to distribution shifts between training and deployment, and thus degrading the forecasting performance (Li et al., 2021; Zhou et al., 2023), as illustrated in Figures 1(a) and 1(b). Robustness to these input perturbations is therefore an important requirement for IMTS forecasting.

Continuous-time models provide a way to model the temporal dynamics and alleviate the sensitivity to input perturbations. In particular, linear-noise stochastic differential equations (LNSDEs) (Oh et al., 2024) constrain the diffusion term to be linear in the latent state, with the aim of attenuating the influence of input perturbations during the continuous-time evolution. Continuous-time Transformers, such as ContiFormer (Chen et al., 2023), use Neural ODEs (Chen et al., 2018) to construct the observation-level latent trajectories and aggregate them through attention, thus reducing reliance on the recursive state propagation and its associated error accumulation. Combining LNSDE dynamics

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.



(a) Patch-Level Temporal Misalignment (b) Frequency-Domain Shift (c) Trajectory Reduction

Figure 1: Motivation for patch-level LNSDE modeling under distribution shifts. (a) Changes in observation patterns shift temporal anchors and alter cross-variable temporal alignment within patches. (b) Training and test observations exhibit a frequency-domain shift. (c) Patching reduces the numerical integration cost by evolving one LNSDE trajectory per variable-level patch.

with continuous-time attention forms a **stochastic continuous-time Transformer**, which offers a promising approach to maintaining robustness to input perturbations while alleviating error accumulation during recursive state propagation. Nevertheless, such a Transformer still requires repeated numerical calculation of integration, resulting in a substantial cost during training and inference. Discrete-time patch-based methods, in contrast, can reduce the computational cost by organizing the asynchronous observations into some local temporal windows, and encoding them as the compact patch embeddings (Zhang et al., 2024; Hu et al., 2025). However, changes in observation density and missingness patterns can perturb the patch embeddings, while these methods do not explicitly control how such perturbations propagate through a subsequent temporal modeling.

It is thus promising to combine the computational efficiency of patch-based representations with the robust mechanism of LNSDEs within a continuous-time Transformer. By performing the continuous evolution at the patch level, as shown in Figure 1(c), such a combination can reduce the number of trajectories compared with the observation-level continuous-time attention, while incorporating the stochastic dynamics to address the sensitivity to input perturbations. However, directly incorporating patch-based representations with stochastic continuous-time transformer is non-trivial and presents three challenges. **First, patch representations must be adapted to the continuous-time evolution.** A patch summarizes observations over an interval, whereas a continuous trajectory requires an initial state at a specified time. Different variables may be observed at different times within the same window, and their representative times may change under observation-process shifts, as illustrated in Figure 1(a). **Second, stochastic continuous-time Transformer must remain stable under patch embedding perturbations.** Changes in patch embeddings affect both the evolved trajectories and their attention weights. Stability of individual trajectories does not directly establish stability of the aggregated output. **Third, continuous-time attention must retain the computational benefits of patching.** Even with fewer patch embeddings, separately solving trajectories for different attention branches can introduce repeated numerical integration and limit the computational savings.

To address these challenges, we propose RosFormer, a robust stochastic continuous-time Transformer for the patch-based IMTS forecasting under distribution shifts. **First,** we construct the variable-level patch embeddings with temporal anchors, and use a relative-time-aware graph to model the inter-variable dependencies based on signed differences between anchors. The resulting embeddings can initialize continuous trajectories at their respective anchors. **Second,** we integrate LNSDE dynamics into the continuous-time attention and adjust patch weights according to the temporal compactness of their observations. We further provide a theoretical analysis of attention-output stability under bounded perturbations of the initial patch embeddings, jointly accounting for their effects on the LNSDE trajectories and attention weights. **Finally,** we derive the queries, keys and values from the same LNSDE trajectory for each variable-level patch, avoiding the separate trajectory solved for different attention branches and reducing the numerical integration cost. Empirically, RosFormer demonstrates the robustness to distribution shifts, as shown by the controlled test-time missingness experiments on four IMTS datasets and hospital-disjoint evaluation on eICU-CRD. Additional experiments show a competitive performance on the standard forecasting and event prediction benchmarks. Efficiency analyses further demonstrate the computational benefits of patch-level trajectories and shared $Q/K/V$ generation. Our contributions are as follows:

- To the best of our knowledge, RosFormer is the first framework to incorporate the patch-based graph representations with LNSDE-driven continuous-time attention for IMTS forecasting. It uses stochastic dynamics to address sensitivity to input perturbations, while reducing numerical integration cost through patch-level evolution and shared trajectories.
- We develop time-aware patch graph representations and compactness-aware continuous-time attention to account for asynchronous observations and aggregate patch trajectories. We further provide a theoretical analysis of attention-output stability under bounded perturbations of the initial patch embeddings.
- Extensive experiments demonstrate RosFormer's robustness to distribution shifts, competitive performance on forecasting and event prediction benchmarks, and computational savings from patch-level trajectories and shared $Q/K/V$ generation.

2 Related Work

Continuous-time models. Continuous-time models describe the latent dynamics at arbitrary timestamps through neural differential equations. Neural ODEs (Chen et al., 2018; Rubanova et al., 2019) model a deterministic state evolution, whereas Neural SDEs (Tzen & Raginsky, 2019) introduce diffusion terms to capture the stochasticity. In particular, linear-noise stochastic differential equations (LNSDEs) (Oh et al., 2024) constrain the diffusion term to be linear in the latent state and can attenuate initial-state perturbations. Continuous-time Transformers such as ContiFormer (Chen et al., 2023) evolve latent trajectories from individual observations using neural ODEs and combine them through attention, thereby reducing dependence on recursive state propagation and alleviating the associated error accumulation. Combining LNSDE dynamics with continuous-time attention offers a promising approach to maintaining robustness to input perturbations while alleviating recursive error accumulation. However, the resulting stochastic continuous-time Transformer still requires repeated numerical integration, incurring substantial computational cost during training and inference.

Patch-based models. Patch-based methods reduce the computational cost by aggregating irregular observations within local temporal windows into compact patch embeddings. T-PatchGNN (Zhang et al., 2024) combines variable-specific patches with time-adaptive GNNs. Hi-Patch (Luo et al., 2025) builds hierarchical patch graphs, while VIMTS (Hu et al., 2025) uses time-channel patches and cross-channel graph modeling. TFMixer (Qiu et al., 2026) combines query-based patch mixing with a learnable non-uniform discrete Fourier transform to capture local temporal patterns and global periodicity. However, distribution shifts involving changes in observation density or missingness can perturb patch embeddings and degrade subsequent predictions. Such changes can also shift variable-specific representative times within a fixed window, potentially misaligning cross-variable relations.

3 Proposed Method

3.1 Problem Setup and Design Objectives

An irregular multivariate time series (IMTS) sample with N variables is represented as $\mathcal{S} = \{\mathcal{S}^n\}_{n=1}^{N}$, where $\mathcal{S}^n = \{(t_i^n, \mathbf{x}_i^n)\}_{i=1}^{L_n}$ denotes the observation sequence of variable n. Here, t_i^n and $\mathbf{x}_i^n$ are the timestamp and observed value, with $0 \le t_1^n < \cdots < t_{L_n}^n$. The timestamps and observation counts may differ across variables. Changes in observation density and missingness patterns between the training and deployment stages introduce input perturbations that can induce distribution shifts and degrade the forecasting performance.

Given $\mathcal{S}$ and a set of requested future timestamps $\mathcal{T}^q = \{t_j^q\}_{j=1}^{L_q}$, our goal is to predict the values of all variables at these timestamps while maintaining robustness to input perturbations. We express this forecasting task by using a continuous-time encoder F_θ and a decoder D_ψ:

$$\mathbf{Z}(t) = F_\theta(\mathcal{S}, t), \qquad \widehat{\mathbf{x}}(t) = D_\psi\big(\mathbf{Z}(t)\big), \quad t \in \mathcal{T}^q, \tag{1}$$

where $\mathbf{Z}(t)$ is the latent representation at query time t and $\widehat{\mathbf{x}}(t)$ contains the predicted values.

LNSDEs (Oh et al., 2024) can attenuate the influence of input perturbations during trajectory evolution, while ContiFormer (Chen et al., 2023) aggregates the latent trajectories through continuous-time attention to capture dependencies across observations. Combining these two mechanisms motivates a stochastic continuous-time Transformer that models these dependencies while retaining

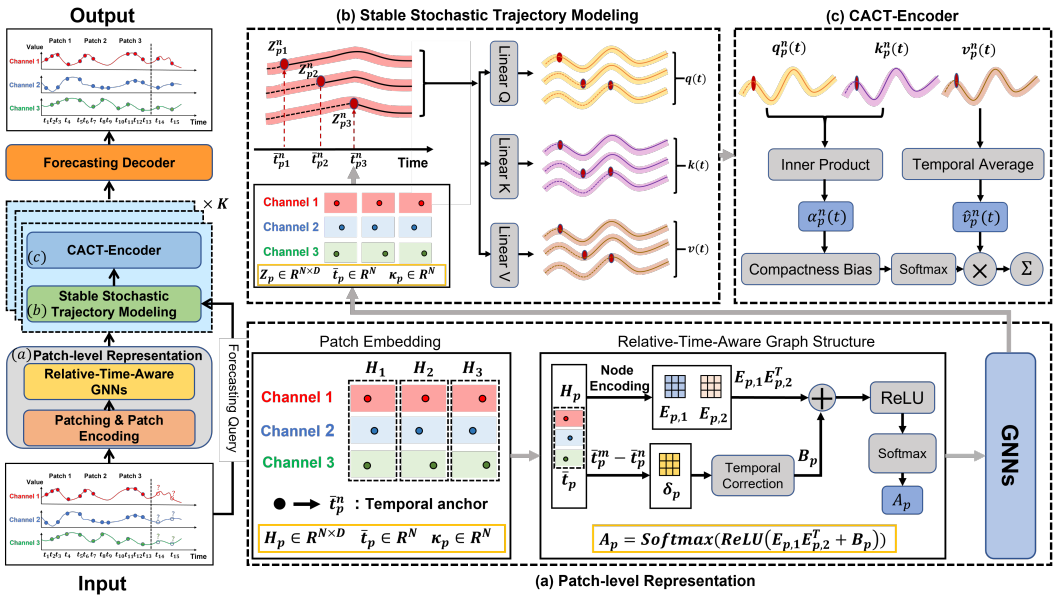


Figure 2: Overview of RosFormer. RosFormer consists of three components: (a) patch-level representation, which converts irregular observations into patch embeddings; (b) stable stochastic trajectory modeling, which evolves patch embeddings into stochastic latent trajectories; and (c) compactness-aware continuous-time encoder (CACT-Encoder),which aggregates patch trajectories.

the perturbation attenuation of LNSDEs. Such a design, however, still relies on the costly numerical integration. This motivates introducing patch-based graph representations to reduce the number of trajectories by initializing them from compact patch embeddings. Though patching offers computational savings, its integration with the stochastic continuous-time Transformer still requires addressing the following three design challenges.

***i)* Adapting patch representations to continuous-time evolution.** A patch representation $\mathbf{Z}_p^n$ summarizes observations of variable n over window p, whereas a continuous trajectory requires an initial state at a specific time. It must therefore be associated with a temporal anchor $\bar{t}_p^n$ to define the initial condition $\mathbf{z}_p^n(\bar{t}_p^n) = \mathbf{Z}_p^n$. However, asynchronous observations give different variables different anchors within the same window. Their signed differences, $\bar{t}_p^m - \bar{t}_p^n$, may also change with observation patterns and must be accounted for when constructing the cross-variable relations.

***ii)* Maintaining attention-output stability under input perturbations.** Input perturbations affect the patch representations $\mathbf{Z}_p^n$ used to initialize the LNSDE trajectories $\mathbf{z}_p^n(t)$. Though LNSDEs can attenuate their influence during evolution, the attention weights computed from these trajectories may still change and affect the stability of the aggregated output $\mathrm{CACT\text{-}ATTN}^n(t)$.

***iii)* Retaining the computational benefits of patching.** Patch-level evolution reduces the number of trajectories. However, generating queries, keys and values from separate trajectories still requires repeated numerical integration for each variable-level patch, limiting the computational savings.

To address these challenges, we propose RosFormer, a robust stochastic continuous-time Transformer for the patch-based IMTS forecasting, with the following three key modules. **1) Patch-level representation:** maps the irregular observations to the patch embeddings, temporal anchors, and compactness scores (Figure 2(a), Section 3.2). It associates each variable-level patch with a temporal anchor, and uses a relative-time-aware graph to account for the anchor differences in cross-variable interactions, thus addressing ***Challenge (i)***. **2) Stable stochastic trajectory modeling:** takes these embeddings and anchors as input and produces the stochastic latent trajectories and their query, key and value trajectories (Figure 2(b), Section 3.3). It evolves each state through an LNSDE and obtains $Q/K/V$ through separate linear projections of the shared trajectory, avoiding redundant numerical integration to address ***Challenge (iii)***. **3) Compactness-aware continuous-time encoder:** uses these trajectories, temporal anchors, and compactness scores to produce the encoded represen-

tations at query times (Figure 2(c), Section 3.4). It aggregates the LNSDE trajectories through a compactness-aware continuous-time attention to reduce the sensitivity of the attention output to the initial patch embedding perturbations, thus addressing ***Challenge (ii)***.The encoded representations are then fed into the forecasting decoder (detailed in Appendix A.3) to generate the final predictions.

3.2 PATCH-LEVEL REPRESENTATION

Patching and Patch Encoding. In RosFormer, we first group each variable's observations into P temporal windows of width W without resampling them onto a dense temporal grid. The window boundaries are shared across variables, while observation times and counts may differ. For a nonempty patch of variable n in window p, the observations are indexed from l_p^n to r_p^n, with $L_p^n = r_p^n - l_p^n + 1$ denoting the number of observations. Following Zhang et al. (2024), we use a TTCN to encode the observed values and their time embeddings into a patch representation: $\mathbf{h}_p^n = \mathrm{TTCN}\left([\phi(t_i^n) \| \mathbf{x}_i^n]_{i=l_p^n}^{r_p^n}\right) \in \mathbb{R}^D$, where $\phi(\cdot)$ is the time embedding function and $\|$ denotes concatenation. The complete TTCN formulation is provided in Appendix A.1.

Each patch representation is associated with a temporal anchor, defined as the mean observation timestamp $\bar{t}_p^n = \frac{1}{L_p^n} \sum_{i=l_p^n}^{r_p^n} t_i^n$. This anchor specifies the starting time of its subsequent continuous trajectory. To also describe how dispersed the observation times are around the anchor, we define temporal compactness as $\kappa_{p,n} = \max\left\{\varepsilon,\, 1 - \frac{2}{W}\sqrt{\frac{1}{L_p^n}\sum_{i=l_p^n}^{r_p^n}\left(t_i^n - \bar{t}_p^n\right)^2}\right\}$. The factor $2/W$ normalizes the standard deviation by its maximum possible value within a window of width W, while the small constant $\varepsilon > 0$ keeps the score positive. A larger $\kappa_{p,n}$ indicates that the observation times are more concentrated around the anchor.

The patch embeddings are then stacked as $\mathbf{H}_p = [\mathbf{h}_p^1; \ldots; \mathbf{h}_p^N] \in \mathbb{R}^{N \times D}$. Together with their temporal anchors and compactness scores, they provide the inputs for inter-variable graph modeling.

Relative-Time-Aware GNN. We then leverage a graph to model the inter-variable dependencies within each temporal window by combining content-based relations with a relative-time correction. The content term captures relations between patch embeddings of different variables, while the correction accounts for differences between their temporal anchors.

To construct the content term, we adapt two sets of learnable static variable embeddings $\mathbf{E}_k^s \in \mathbb{R}^{N \times D_{\mathrm{node}}}$, $k \in \{1, 2\}$, using the current patch contents. A gate $\mathbf{g}_{p,k} = \mathrm{ReLU}\left(\tanh\left([\mathbf{H}_p \| \mathbf{E}_k^s]\mathbf{W}_k^g\right)\right)$ controls the contribution of these contents, giving the node embeddings $\mathbf{E}_{p,k} = \mathbf{E}_k^s + \mathbf{g}_{p,k} \odot (\mathbf{H}_p \mathbf{W}_k^d)$. Here, $\mathbf{W}_k^g \in \mathbb{R}^{(D + D_{\mathrm{node}}) \times D_{\mathrm{node}}}$ and $\mathbf{W}_k^d \in \mathbb{R}^{D \times D_{\mathrm{node}}}$ are learnable weights. We use $\mathbf{E}_{p,1}$ and $\mathbf{E}_{p,2}$ as the target and source embeddings, respectively. Their product $\mathbf{E}_{p,1}\mathbf{E}_{p,2}^\top$ provides the content-based edge scores.

For an edge from source variable m to target variable n, the signed temporal offset is $\delta_{p,nm} = \bar{t}_p^m - \bar{t}_p^n$. We encode it as $\boldsymbol{\phi}_\tau(\delta) = \tanh\left(f_\tau(\delta) - f_\tau(0)\right) \in \mathbb{R}^{d_\tau}$, where f_τ is a shared MLP. Subtracting $f_\tau(0)$ ensures that $\boldsymbol{\phi}_\tau(0) = \mathbf{0}$, so coincident anchors introduce no temporal correction.

The same temporal offset may affect different variable pairs differently. We capture this dependence using both the target and source embeddings:

$$\mathbf{c}_{p,nm} = \tanh\left(\mathbf{E}_{p,1}[n,:]\mathbf{W}_{\mathrm{tar}} + \mathbf{b}_{\mathrm{tar}}\right) \odot \tanh\left(\mathbf{E}_{p,2}[m,:]\mathbf{W}_{\mathrm{src}} + \mathbf{b}_{\mathrm{src}}\right) + \boldsymbol{\gamma}. \tag{2}$$

Here, $[n,:]$ and $[m,:]$ select the target and source rows, $\mathbf{W}_{\mathrm{tar}}, \mathbf{W}_{\mathrm{src}} \in \mathbb{R}^{D_{\mathrm{node}} \times d_\tau}$, and $\mathbf{b}_{\mathrm{tar}}, \mathbf{b}_{\mathrm{src}}, \boldsymbol{\gamma} \in \mathbb{R}^{d_\tau}$.

The interaction between $\mathbf{c}_{p,nm}$ and the encoded time offset determines the edge correction: $B_{p,nm}^\tau = \kappa_{p,m} \frac{\tanh(\eta)}{\sqrt{d_\tau}} \langle \mathbf{c}_{p,nm}, \boldsymbol{\phi}_\tau(\delta_{p,nm}) \rangle$. The source compactness $\kappa_{p,m}$ weakens this correction when its observations are temporally dispersed. The learnable scalar η is initialized to zero, so graph construction initially depends only on content-based relations.

Collecting these corrections into $\mathbf{B}_p^\tau \in \mathbb{R}^{N \times N}$, we combine the content and time terms to obtain the adjacency matrix:

$$\mathbf{A}_p = \mathrm{Softmax}\left(\mathrm{ReLU}\left(\mathbf{E}_{p,1}\mathbf{E}_{p,2}^\top + \mathbf{B}_p^\tau\right)\right). \tag{3}$$

The row-wise softmax normalizes the weights over source variables for each target variable. Graph message passing then updates the patch representations as $\mathbf{Z}_p = \mathrm{ReLU}\left(\mathbf{A}_p \mathbf{H}_p \mathbf{W}^{\mathrm{gcn}}\right) + \mathbf{H}_p \in \mathbb{R}^{N \times D}$, where $\mathbf{W}^{\mathrm{gcn}} \in \mathbb{R}^{D \times D}$.

Let $\mathbf{Z}_p^n$ denote the n-th row of $\mathbf{Z}_p$. The pairs $\{(\bar{t}_p^n, \mathbf{Z}_p^n)\}$ specify the starting times and initial states of the LNSDE trajectories, while $\kappa_{p,n}$ is used in their attention aggregation, as described in Section 3.4.

3.3 Stable Stochastic Trajectory Modeling

Given the patch representations $\{(\bar{t}_p^n, \mathbf{Z}_p^n)\}$ and compactness scores $\{\kappa_{p,n}\}$, RosFormer further constructs an LNSDE trajectory for each variable-level patch and aggregates these trajectories at a query time. Queries, keys, and values share the same trajectory, while compactness scores adjust the contributions of different patches.

LNSDE-based trajectory modeling. Starting from the anchor $\mathbf{Z}_p^n$ at its representative time $\bar{t}_p^n$, RosFormer evolves a stochastic latent trajectory with a linear-noise SDE:

$$\mathbf{z}_p^n(t) = \mathbf{Z}_p^n + \int_{\bar{t}_p^n}^t f_\theta(\tau, \mathbf{z}_p^n(\tau))d\tau + \int_{\bar{t}_p^n}^t g_\sigma(\tau)\mathbf{z}_p^n(\tau)dW_\tau, \tag{4}$$

where W_τ denotes Brownian motion, and f_θ and g_σ parameterize the drift and diffusion. The diffusion term enables a stochastic latent evolution, while the linear-noise structure constrains the diffusion to be linear with respect to the latent state. This provides a controlled stochastic propagation mechanism under observation-process shifts, as will be later analyzed in Section 4.

Shared query-key-value trajectory construction. A naive stochastic continuous-time attention design would construct the independent stochastic paths for the queries, keys, and values, requiring repeated numerical integration. RosFormer instead evolves one shared stochastic trajectory $\mathbf{z}_p^n(t)$ and obtains $\mathbf{q}_p^n(t) = \mathbf{z}_p^n(t)W_Q$, $\mathbf{k}_p^n(t) = \mathbf{z}_p^n(t)W_K$, and $\mathbf{v}_p^n(t) = \mathbf{z}_p^n(t)W_V$. This shared construction couples the query, key, and value trajectories through the same stochastic latent evolution, while substantially reducing the trajectory construction cost.

3.4 Compactness-Aware Continuous-Time Encoder

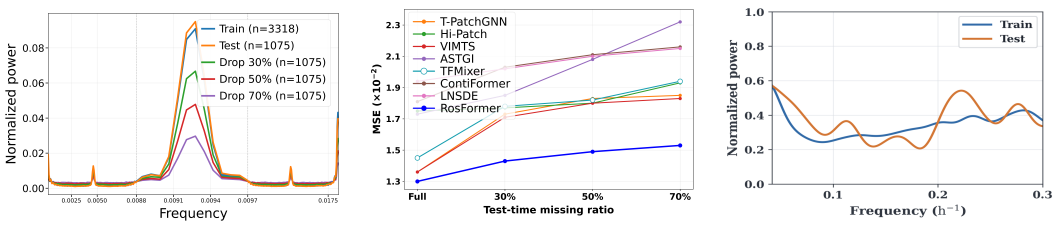
Compactness-aware continuous-time attention. For a query time t, RosFormer then computes the relevance of each anchor trajectory by averaging the query-key similarity over $[\bar{t}_p^n, t]$, i.e., $\alpha_p^n(t) = (t - \bar{t}_p^n)^{-1} \int_{\bar{t}_p^n}^t \mathbf{q}_p^n(\tau)\mathbf{k}_p^n(\tau)^\top d\tau$. The value trajectory is summarized over the same interval as $\hat{\mathbf{v}}_p^n(t) = (t - \bar{t}_p^n)^{-1} \int_{\bar{t}_p^n}^t \mathbf{v}_p^n(\tau)d\tau$. The compactness-aware continuous-time attention output is then

$$\text{CACT-ATTN}^n(t) = \sum_{p=1}^{P} \mathrm{softmax}\left(\frac{\alpha_p^n(t)}{\sqrt{D}} + \log \kappa_{p,n}\right) \hat{\mathbf{v}}_p^n(t), \tag{5}$$

where the softmax is taken over anchor indices p, $\kappa_{p,n}$ denotes the temporal compactness of the p-th patch of variable n, as defined in Section 3.2. For the same trajectory similarity, a temporally compact patch receives a larger aggregation weight, whereas a dispersed patch with a less representative anchor is downweighted.

Multi-head encoder. The CACT-ATTN operator defined above gives a single-head continuous-time attention mechanism. To build the full encoder, RosFormer uses H parallel CACT-ATTN heads, where each head learns a different representation subspace. Their outputs are concatenated and projected as $\text{CACT-MHA}^n(t) = \mathrm{Concat}(head_{(1)}^n(t), \ldots, head_{(H)}^n(t))W^O$, where $head_{(h)}^n(t) = \text{CACT-ATTN}_{(h)}^n(t)$ (Chen et al., 2023).

RosFormer then stacks CACT-MHA layers with a pre-normalized residual Transformer block. Given the input continuous-time representation $\mathbf{Z}^{l-1}(t)$ to the l-th layer, the block updates it by $\tilde{\mathbf{Z}}^l(t) = \mathbf{Z}^{l-1}(t) + \text{CACT-MHA}(\mathrm{LN}(\mathbf{Z}^{l-1}(t)))$ and $\mathbf{Z}^l(t) = \tilde{\mathbf{Z}}^l(t) + \mathrm{FFN}(\mathrm{LN}(\tilde{\mathbf{Z}}^l(t)))$. Here, LN denotes layer normalization and FFN is a position-wise feed-forward network. After stacking multiple blocks, RosFormer obtains the final continuous-time representation $\mathbf{Z}^L(t)$, which can be queried at arbitrary timestamps and fed into the forecasting decoder (detailed in Appendix A.3).



(a) Frequency-domain shift (b) MIMIC-III (c) Cross-hospital shift

Figure 3: Analyses under controlled and natural distribution shifts. (a) Additional test-time observation dropping changes induces a frequency-domain distribution shift. (b) Forecasting MSE on MIMIC-III under increasing test-time missingness. (c) Naturally occurring frequency-domain shift between the hospital-disjoint training and test sets of eICU-CRD.

4 Theoretical Analysis

We analyze the stability of RosFormer under the observation-process shifts. Changes in observation density and missingness patterns can alter the latent states inferred from irregular observations, and such perturbations may be amplified during the continuous-time trajectory propagation. To formalize this effect, we model the observation-process shifts as bounded perturbations to the initial latent states and study how these perturbations affect the output of CACT-ATTN.

We let $\mathrm{CACT\text{-}ATTN}^n(t)$ denote the CACT-ATTN output for variable n at query time t initialized from latent state z^n, and let $\widetilde{\mathrm{CACT\text{-}ATTN}}^n(t)$ denote the corresponding output initialized from a perturbed state $\tilde{z}^n$. We assume the initial perturbation bounded in mean square, i.e., $\sqrt{\mathbb{E}\|z^n - \tilde{z}^n\|^2} \le \rho$, let $t_{\max}$ be the largest timestamp among the input temporal anchors, and consider query times $t > t_{\max}$.

Our analysis relies on the standard regularity conditions commonly used for stable neural differential equations and LNSDEs. Specifically, we assume that the drift and projection functions are Lipschitz, satisfying the standard growth and dissipativity conditions, and that the stochastic trajectories have bounded moments. These assumptions ensure well-posed stochastic dynamics and allow perturbation errors to be controlled. The formal statements are provided in Appendix C.2.

Theorem 4.1 (Robustness Bound of RosFormer)**.** *Under the assumptions above, we consider the LNSDE backbone with a drift Lipschitz constant L_f and effective linear-noise intensity σ_θ. Then, there exists a constant $C_A > 0$ such that, for any $t > t_{\max}$,*

$$\mathbb{E}\left[\left\|\mathrm{CACT\text{-}ATTN}^n(t) - \widetilde{\mathrm{CACT\text{-}ATTN}}^n(t)\right\|\right] \le \frac{C_A(1+\rho)}{\lambda(t - t_{\max})}, \tag{6}$$

where $\lambda = \frac{1}{2}(|\sigma_\theta|^2 - 2L_f)$. Particularly, when $|\sigma_\theta|^2 > 2L_f$, the right-hand side vanishes as $t \to \infty$.

Proof. See Appendix C.5 for the detailed proof. □

Remark 4.2. Theorem 4.1 shows that the CACT-ATTN output deviation scales linearly with the factor $1+\rho$ induced by the initial perturbation and vanishes as the query time moves away from the latest input anchor. This explains how RosFormer controls the trajectory deviations under observation-process shifts. Under the stability condition, the linear-noise stochastic dynamics attenuate trajectory perturbations. After the interval-based aggregation and Lipschitz attention mapping, this stabilizing effect is preserved at the attention-output level, yielding a bound of order $O(1/(t - t_{\max}))$.

In Eq. (43) of Appendix, we further derive the bound for the ODE-based continuous-time attention, showing that its output deviation does not necessarily admit asymptotic decay as the query time moves away from the latest input anchor. This suggests that a deterministic continuous-time attention may still propagate the initial perturbations, whereas our RosFormer controls the attention-output deviations through a structured stochastic trajectory modeling under the distribution shifts.

5 Experiments

We evaluate RosFormer under two distribution-shift settings: controlled test-time missingness on four IMTS forecasting datasets and a hospital-disjoint eICU-CRD setting with naturally differing

Table 1: MIMIC-III forecasting under controlled test-time missingness.Results are averaged over three random seeds. The best results are in **bold**, and the second-best results are in underlined.

Methods	Full		30%		50%		70%	
	MSE×10^{-2}	MAE×10^{-2}	MSE×10^{-2}	MAE×10^{-2}	MSE×10^{-2}	MAE×10^{-2}	MSE×10^{-2}	MAE×10^{-2}
T-PatchGNN (Zhang et al., 2024)	1.36 ± 0.02	6.56 ± 0.11	1.73 ± 0.04	7.60 ± 0.12	1.83 ± 0.03	7.84 ± 0.06	1.85 ± 0.04	7.97 ± 0.08
Hi-Patch (Luo et al., 2025)	1.75 ± 0.26	7.24 ± 0.18	1.77 ± 0.20	7.48 ± 0.16	1.80 ± 0.24	7.66 ± 0.14	1.93 ± 0.20	7.92 ± 0.14
VIMTS (Hu et al., 2025)	1.36 ± 0.02	6.40 ± 0.17	1.71 ± 0.02	7.12 ± 0.10	1.80 ± 0.01	7.34 ± 0.12	1.83 ± 0.02	7.42 ± 0.14
ContiFormer (Chen et al., 2023)	1.81 ± 0.02	7.75 ± 0.11	2.03 ± 0.04	8.64 ± 0.11	2.11 ± 0.02	8.73 ± 0.12	2.16 ± 0.02	8.77 ± 0.10
LNSDE (Oh et al., 2024)	1.94 ± 0.04	7.76 ± 0.10	2.02 ± 0.04	8.10 ± 0.10	2.10 ± 0.03	8.52 ± 0.08	2.15 ± 0.05	8.70 ± 0.12
ASTGI (Liu et al., 2026)	1.32 ± 0.02	6.45 ± 0.11	1.85 ± 0.02	7.67 ± 0.10	2.08 ± 0.03	8.35 ± 0.12	2.32 ± 0.02	9.17 ± 0.14
TFMixer (Qiu et al., 2026)	1.45 ± 0.02	6.72 ± 0.11	1.78 ± 0.02	7.49 ± 0.11	1.82 ± 0.02	7.93 ± 0.12	1.94 ± 0.03	8.74 ± 0.10
RosFormer	**1.30 ± 0.02**	**6.32 ± 0.15**	**1.43 ± 0.01**	**6.68 ± 0.08**	**1.49 ± 0.01**	**6.78 ± 0.10**	**1.53 ± 0.02**	**7.12 ± 0.10**

observation patterns. We further conduct ablations on eICU-CRD. The standard forecasting, event prediction, and computational efficiency results are reported in Appendices A.4, A.6, and B, respectively. Detailed experimental configurations are provided in Appendix A.

5.1 Experimental Setup and Baselines

Implementation details. All the models are implemented in PyTorch and trained on an NVIDIA GeForce RTX 3090 GPU. Following ContiFormer (Chen et al., 2023), we adopt the Gauss–Legendre quadrature to implement Eq. (5). To solve the LNSDE, we employ the Euler–Maruyama algorithm with a fixed step size of 0.1. More implementation details are provided in Appendix A.

Controlled test-time missingness. Following the forecasting settings of VIMTS (Hu et al., 2025), we evaluate on Human Activity, USHCN (Menne et al., 2015), PhysioNet (Silva et al., 2012), and MIMIC-III (Johnson et al., 2016), using a 60:20:20 training, validation, and test split. The baselines include discrete-time methods and continuous-time models in IMTS:T-PatchGNN (Zhang et al., 2024), Hi-Patch (Luo et al., 2025), VIMTS (Hu et al., 2025), TFMixer (Qiu et al., 2026), ASTGI (Liu et al., 2026),ContiFormer (Chen et al., 2023) and LNSDE (Oh et al., 2024). All models are trained on the original training data. At test time, we randomly remove an additional 30%, 50%, or 70% of observed input measurements without changing the prediction targets or retraining the models. We use the mean squared error (MSE) and mean absolute error (MAE) as evaluation metrics.

Hospital-disjoint eICU evaluation. For the real-world setting, we construct a hospital-disjoint split of eICU-CRD (Pollard et al., 2019). Training and validation stays come from source hospitals, whereas test stays come exclusively from hospitals unseen during training. We use 12 physiological and laboratory variables collected during the first 24 hours of each ICU stay, retaining the original timestamps and observation masks without artificially removing observations. Details of the split and the observed differences in sampling patterns are provided in Appendix A.5.2.

5.2 Experimental Results Analysis

Robustness under Controlled Test-Time Missingness Shifts. We first evaluate RosFormer under controlled observation-process shifts induced by test-time missingness. As shown in Fig. 3a, test-time observation removal progressively changes the frequency-domain characteristics of the test data, confirming that increasing missingness induces a non-trivial distribution shift. Figure 3b and Table 1 present MIMIC-III as a representative case. They jointly show that RosFormer maintains more stable forecasting performance as the missingness-induced shift becomes increasingly severe. Although the errors of all methods generally increase with the missing ratio, RosFormer consistently achieves the lowest MSE and MAE across all four settings and maintains its predictive advantage across different missing ratio. It demonstrats that RosFormer has strong robustness as the test-time observation pattern increasingly deviates from the training distribution.Complete results on Human Activity, PhysioNet, and USHCN are provided in Appendix A.5. The overall pattern is consistent across these datasets: RosFormer remains the best or a highly competitive method as test-time missingness increases, achieving the lowest MSE under 70% missingness on all four datasets.

Table 2: Forecasting results on the hospital-disjoint eICU-CRD test set. MSE and MAE are scaled by 10^{-1}.Results are averaged over three independent runs with different random seeds.

Model	MSE (↓)	MAE (↓)
T-PatchGNN (Zhang et al., 2024)	3.91 ± 0.05	4.38 ± 0.04
Hi-Patch (Luo et al., 2025)	4.01 ± 0.10	4.41 ± 0.08
VIMTS (Hu et al., 2025)	3.95 ± 0.04	4.40 ±0.03
ContiFormer (Chen et al., 2023)	4.22 ± 0.04	4.62 ± 0.03
LNSDE (Oh et al., 2024)	4.06 ± 0.05	4.45 ± 0.07
ASTGI (Liu et al., 2026)	3.83 ± 0.04	4.30 ± 0.04
TFMixer (Qiu et al., 2026)	3.87 ± 0.03	4.36 ± 0.05
RosFormer	**3.69 ± 0.02**	**4.21 ± 0.02**

Table 3: Ablation study on the hospital-disjoint eICU-CRD test set. MSE and MAE are scaled by 10^{-1}. Results are averaged over three random seeds.The best results are boldfaced, and the second-best results are underlined.

Variant	MSE (↓)	MAE (↓)
w/o Stochasticity	4.04 ± 0.03	4.44 ± 0.04
w/o Linear-Noise Constraint	3.93 ± 0.01	4.42 ± 0.02
w/o Time-Aware Edges	3.80 ± 0.02	4.30 ± 0.03
w/o Compactness Weighting	3.78 ± 0.02	4.27 ± 0.04
RosFormer	**3.69 ± 0.02**	**4.21 ± 0.02**

Robustness under Natural Hospital-Disjoint Shift. In contrast to the controlled missingness setting, the hospital-disjoint evaluation preserves the original observations and evaluates models on hospitals unseen during training. Figure 3c illustrates differences in the spectral profiles of a representative variable between the hospital-disjoint training and test sets. More comprehensive observation-density comparisons across variables and time intervals are provided in Appendix A.5.2. Together, these analyses provide evidence of a naturally occurring cross-hospital distribution shift in both observation patterns and temporal characteristics. As shown in Table 2, RosFormer achieves the lowest MSE and MAE. These results demonstrate that RosFormer generalizes more robustly to unseen hospitals and that its advantages are not limited to artificially induced missingness shifts.

Ablation Study. To assess how main components contribute to forecasting robustness under distribution shift, we conduct an ablation study with four variants in the hospital-disjoint eICU setting: **i) w/o Stochasticity:** we remove the diffusion term, reducing the LNSDE to a deterministic Neural ODE. **ii) w/o Linear-Noise Constraint:** we replace the linear diffusion term with a generic multi-layer perceptron (MLP). **iii) w/o Relative-Time Edges:** we remove relative temporal-anchor information from edge construction while retaining the GNN and its content-based connections. **iv) w/o Compactness Weighting:** we set compactness coefficients to one only in the attention scores.

As shown in Table 3, relaxing the linear-noise constraint and removing stochasticity both degrade performance, highlighting the importance of LNSDE dynamics. Relative-time-aware edges and compactness weighting in attention also contribute to robustness under distribution shift.

Computational Efficiency Analysis. We evaluate efficiency on Activity dataset with a batch size of 32, with a detailed efficiency analysis provided in Appendix B. Compared with ContiFormer using LNSDEs, RosFormer with independent $Q/K/V$ trajectories substantially reduces runtime through patch-level trajectory construction. Sharing trajectories across $Q/K/V$ further reduces runtime and GPU memory use while maintaining comparable forecasting accuracy (Table 11). The resulting training and inference times of RosFormer are comparable to T-PatchGNN.

5.3 Limitation

RosFormer has two limitations. First, numerical integration of LNSDE trajectories introduces computational overhead despite shared $Q/K/V$ generation. Efficient solvers or solver-free approximations could reduce this cost. Second, patch summarization may lose fine temporal details, especially around abrupt local changes. Adaptive or hierarchical patching could help preserve these details.

6 Conclusion

We presented RosFormer for a robust IMTS forecasting under distribution shifts. Its relative-time-aware graph alleviated the temporal misalignment, while the LNSDE dynamics attenuated the influence of perturbations to patch embeddings. Continuous-time attention aggregated trajectories to help alleviate recursive error accumulation. Under stated stability conditions, we proved that attention output deviations caused by bounded initial perturbations could vanish over time. Patch-level evolution and shared $Q/K/V$ trajectories further reduced computational cost. Experiments demonstrated the robustness, competitive forecasting performance and computational efficiency of RosFormer.

AI USE STATEMENT

Large Language Models (LLMs) were used to assist with the writing and manuscript preparation, including sentence rephrasing, language editing, grammar checking. The LLM was not involved in the ideation, research methodology, or experimental design. The authors reviewed all AI-assisted text and take full responsibility for the methodology, experiments, analyses, and conclusions presented in this paper..

ETHICS STATEMENT

This work adheres to the ICLR Code of Ethics. All datasets used in our experiments, including Human Activity, USHCN, Physionet, MIMIC-III, eICU-CRD, Neonate, Traffic, MIMIC, BookOrder and StackOverflow, are publicly available research datasets and were used in accordance with their intended research purposes and applicable usage guidelines. No new personal data were collected, and no interventions involving human participants or animals were conducted.

REPRODUCIBILITY STATEMENT

The main paper and appendices describe the model architecture, training objectives, numerical integration procedures, and evaluation protocols, including controlled test-time missingness and hospital-disjoint evaluation on eICU-CRD. The real-world datasets are available from their original providers, subject to the corresponding access requirements.

REFERENCES

Marin Biloš, Johanna Sommer, Syama Sundar Rangapuram, Tim Januschowski, and Stephan Günnemann. Neural flows: Efficient alternative to neural odes. *Advances in neural information processing systems*, 34:21325–21337, 2021.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085, 2018.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Yuqi Chen, Kan Ren, Yansen Wang, Yuchen Fang, Weiwei Sun, and Dongsheng Li. Contiformer: Continuous-time transformer for irregular time series modeling. *Advances in Neural Information Processing Systems*, 36:47143–47175, 2023.

Edward De Brouwer, Jaak Simm, Adam Arany, and Yves Moreau. Gru-ode-bayes: Continuous modeling of sporadically-observed time series. *Advances in neural information processing systems*, 32, 2019.

Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1555–1564, 2016.

Bolin Gao and Lacra Pavel. On the properties of the softmax function with application in game theory and reinforcement learning. *arXiv preprint arXiv:1704.00805*, 2017.

Zhangyi Hu, Jiemin Wu, Hua XU, Mingqian Liao, Ninghui Feng, Bo Gao, Songning Lai, and Yutao Yue. IMTS is worth time $\times$ channel patches: Visual masked autoencoders for irregular multivariate time series prediction. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=2mnZ7AjMim.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):160035, 2016.

Leskovec Jure. Snap datasets: Stanford large network dataset collection. *Retrieved December 2021 from http://snap. stanford. edu/data*, 2014.

Rafail Khasminskii. *Stochastic stability of differential equations*. Springer Science & Business Media, 2011.

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.

Aodong Li, Alex Boyd, Padhraic Smyth, and Stephan Mandt. Detecting and adapting to irregular distribution shifts in bayesian online learning. *Advances in neural information processing systems*, 34:6816–6828, 2021.

Xvyuan Liu, Xiangfei Qiu, Hanyin Cheng, Xingjian Wu, Chenjuan Guo, Bin Yang, and Jilin Hu. ASTGI: Adaptive spatio-temporal graph interactions for irregular multivariate time series forecasting. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Wg9Rx5rjgo.

Yicheng Luo, Bowen Zhang, Zhen Liu, and Qianli Ma. Hi-patch: Hierarchical patch gnn for irregular multivariate time series. In *Forty-second International Conference on Machine Learning*, 2025.

Xuerong Mao. *Stochastic differential equations and applications*. Elsevier, 2007.

MJ Menne, C Williams Jr, R Vose, and Data Files. Long-term daily climate records from stations across the contiguous united states, 2015.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

YongKyung Oh, Dongyoung Lim, and Sungil Kim. Stable neural stochastic differential equations in analyzing irregular time series data. In *International Conference on Learning Representations*, volume 2024, pp. 38231–38262, 2024.

Tom Pollard, Alistair Johnson, Jesse Raffa, Leo Anthony Celi, Omar Badawi, and Roger Mark. eICU Collaborative Research Database. *PhysioNet*, April 2019. doi: 10.13026/C2WM1R. URL https://doi.org/10.13026/C2WM1R. Version 2.0.

Xiangfei Qiu, Kangjia Yan, Xvyuan Liu, Xingjian Wu, and Jilin Hu. Bridging time and frequency: A joint modeling framework for irregular multivariate time series forecasting. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=GLPTyKiiYZ.

Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32, 2019.

Mona Schirmer, Mazin Eltayeb, Stefan Lessmann, and Maja Rudolph. Modeling irregular time series with continuous recurrent units. In *International conference on machine learning*, pp. 19388–19405. PMLR, 2022.

Satya Narayan Shukla and Benjamin M Marlin. Multi-time attention networks for irregularly sampled time series. In *... International Conference on Learning Representations*, volume 2021, pp. 14897, 2021.

Ikaro Silva, George Moody, Daniel J Scott, Leo A Celi, and Roger G Mark. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *2012 computing in cardiology*, pp. 245–248. IEEE, 2012.

Nathan J Stevenson, Karoliina Tapani, Leena Lauronen, and Sampsa Vanhatalo. A dataset of neonatal eeg recordings with seizure annotations. *Scientific data*, 6(1):190039, 2019.

Belinda Tzen and Maxim Raginsky. Neural stochastic differential equations: Deep latent gaussian models in the diffusion limit. *arXiv preprint arXiv:1905.09883*, 2019.

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.

Zhen-Jie Yao, Jie Bi, and Yi-Xin Chen. Applying deep learning to individual and community health monitoring data: A survey. *International Journal of Automation and Computing*, 15(6):643–655, 2018.

Jiawen Zhang, Shun Zheng, Wei Cao, Jiang Bian, and Jia Li. Warpformer: A multi-scale modeling approach for irregular clinical time series. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3273–3285, 2023.

Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive hawkes process. In *International conference on machine learning*, pp. 11183–11193. PMLR, 2020.

Weijia Zhang, Chenlong Yin, Hao Liu, Xiaofang Zhou, and Hui Xiong. Irregular multivariate time series forecasting: A transformable patching graph neural networks approach. In *ICML*, pp. 60179–60196, 2024.

Xiang Zhang, Marko Zeman, Theodoros Tsiligkaridis, and Marinka Zitnik. Graph-guided network for irregularly sampled multivariate time series. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Kwm8I7dU-l5.

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.

Helen Zhou, Sivaraman Balakrishnan, and Zachary Lipton. Domain adaptation under missingness shift. In *International Conference on Artificial Intelligence and Statistics*, pp. 9577–9606. PMLR, 2023.

Appendix

This appendix provides supplementary material to the main paper, organized into the following sections. Section A presents the experimental details, including the input processing heads, continuous-time embedding, temporal feature extraction, implementation settings and task-specific training configurations. Section A.4 reports the complete forecasting results on all IMTS forecasting benchmarks. Section A.5 provides additional numerical results and visualizations for the test-time missingness experiment, showing the degradation trends under stronger missingness-induced observation-process shifts. Section A.6 present the complete results for event prediction. Section B provides the computational efficiency analysis, including training latency, inference latency, and GPU memory consumption, and further explains how the shared stochastic trajectory design mitigates the computational overhead of LNSDE-based continuous-time modeling. Section C provides the theoretical details and proofs for the robustness analysis of RosFormer, including the stochastic stability preliminaries, moment boundedness analysis of the LNSDE, error decay results, and the proof of the CACT-ATTN robustness bound.

A Experimental Details

A.1 Input Processing Heads

Continuous Time Embedding. To preserve precise temporal information, we utilize learnable time embeddings to encode continuous-time information Shukla & Marlin (2021), mapping each timestamp t to a high-dimensional vector $\phi(t) \in R^{D_{te}}$:

$$\phi(t)[d] = \begin{cases} \omega_0 t + \alpha_0, & \text{if } d = 0, \\ \sin(\omega_d t + \alpha_d), & \text{if } 0 < d < D_{te}, \end{cases} \tag{7}$$

where ω and α are learnable parameters.

Temporal Feature Extraction. Following the time-wise dividing, we employ a Transformable Time-aware Convolutional Network (TTCN) Zhang et al. (2024) to process variable-length sequences within time intervals into aligned feature patches. For the n-th channel in the p-th window, we concatenate the time embeddings and observations to form the input sequence x_p^n:

$$x_p^n = [\phi(t_i^n) \| x_i^n]_{i=l_p^n}^{r_p^n}, \tag{8}$$

where l_p^n and r_p^n denote the start and end indices of the observations within the section, respectively. To capture intra-section information, we generate adaptive convolution filters f_d^n for the d-th feature map using a content-adaptive mechanism:

$$f_d^n = \left[\frac{\exp(\mathcal{F}_d(x_p^n[i]))}{\sum_{j=1}^{L_p^n} \exp(\mathcal{F}_d(x_p^n[j]))} \right]_{i=1}^{L_p^n}, \tag{9}$$

where $L_p^n = r_p^n - l_p^n + 1$ is the number of points, and $\mathcal{F}_d$ is a learnable meta-filter. Finally, we obtain the aggregated latent vector $h_p^{n'} \in R^{D_{in}}$ via temporal convolution:

$$\mathbf{h}_p^n = \left[\sum_{i=1}^{L_p^n} f_d^n[i]^\top x_p^n[i] \right]_{d=1}^{D_{in}}. \tag{10}$$

Crucially, to anchor this representation in continuous time, we compute the patch's centroid time $\bar{t}_p^n$:

$$\bar{t}_p^n = \frac{1}{L_p^n} \sum_{i=l_p^n}^{r_p^n} t_i^n \tag{11}$$

This yields the aligned pair $(H_p, \bar{t}_p)$, where H_p aggregates $h_p^{n'}$ across channels.

A.2 Task Definitions

Problem 1 (IMTS Forecasting). Given the historical observations $\mathcal{S}$, the forecasting task aims to predict the values of the time series over a future horizon H. Formally, the goal is to learn a mapping function $\mathcal{F} : \mathcal{S} \to \hat{\mathbf{Y}}$, where $\hat{\mathbf{Y}} = \{\hat{\mathbf{x}}_{L+1}, \ldots, \hat{\mathbf{x}}_{L+H}\}$ denotes the predicted observations at future timestamps.

Problem 2 (Event Prediction). In the context of event streams where $\mathbf{x}_i$ represents an event category $k_i \in \{1, \ldots, K\}$, the task is to predict the dynamics of the next event. Unlike standard forecasting which predicts values at fixed or queried steps, event prediction involves estimating the joint probability distribution of the next event's arrival time and its type: $p(t_{L+1}, k_{L+1} \mid \mathcal{S})$.

A.3 Task Decoder

Let the output of the final RosFormer layer be denoted as $Z^l(t)$, which captures the learned stochastic continuous-time latent dynamics. We adopt task-specific decoders as follows.

IMTS Forecasting. Given queried timestamps t_q, we apply an MLP projection head to map latent features to the observation space:

$$\hat{\mathbf{Y}} = \mathrm{Reshape}\big(MLP_{\text{fore}}(Z^l(t_q))\big) \in R^{B \times q_t \times N}, \tag{12}$$

where B is batch size, q_t is the number of query steps, and N is the number of variables. We minimize the mean squared error:

$$\mathcal{L}_{\text{fore}} = \frac{1}{Bq_tN} \sum \big\| \hat{\mathbf{Y}} - \mathbf{Y} \big\|_2^2. \tag{13}$$

Event Prediction. Let t_L denote the last observed time. We read out the last latent state and use two heads:

$$\mathbf{h}_L = \mathrm{Readout}(Z^l(t_L)), \qquad \hat{\Delta t} = MLP_{\Delta t}(\mathbf{h}_L), \qquad \hat{\mathbf{p}} = Softmax(MLP_k(\mathbf{h}_L)). \tag{14}$$

The multi-task objective is:

$$\mathcal{L}_{\text{evt}} = \lambda_t \|\hat{\Delta t} - \Delta t\|_2^2 + \lambda_k \Big(- \sum_{j=1}^{K} \mathbf{1}[k = j] \log \hat{\mathbf{p}}_j \Big). \tag{15}$$

A.4 Full Experimental Results of IMTS Forecasting

Following the settings in VIMTS (Hu et al., 2025), we evaluate the forecasting performance on four real-world datasets from diverse domains: **PhysioNet** (Silva et al., 2012), **MIMIC-III** (Johnson et al., 2016), **Human Activity**, and **USHCN** (Menne et al., 2015). Datasets are split into training, validation, and testing sets with a ratio of 60:20:20. We use the mean squared error (MSE) and mean absolute error (MAE) as evaluation metrics. The forecasting baselines include representative regular time series forecasting models, irregular time series models, and continuous-time models: mTAND (Shukla & Marlin, 2021), Latent ODEs (Rubanova et al., 2019), CRU (Schirmer et al., 2022), LNSDE (Oh et al., 2024), ContiFormer (Chen et al., 2023)[1], T-PatchGNN (Zhang et al., 2024), Hi-Patch (Luo et al., 2025), VIMTS (Hu et al., 2025), and ASTGI (Liu et al., 2026). The results of other methods include TimesNet (Wu et al., 2022), PatchTST (Nie et al., 2022), Crossformer (Zhang & Yan, 2023), GRU-D (Che et al., 2018), RainDrop (Zhang et al., 2022), Warpformer (Zhang et al., 2023),Neural Flows (Biloš et al., 2021)are reported in the Appendix A.4

We report the complete forecasting results on four IMTS forecasting benchmarks in Table 4. The results are presented as mean ± standard deviation. The best results are highlighted in **bold**, and the second-best results are underlined.

[1]On Traffic and BookOrder datasets, our reproduced results for ContiFormer and SAHP are lower than those reported in the original papers. We therefore report the reproduced results here for a fair comparison.

Table 4: Full method benchmarking on IMTS forecasting. The best results are highlighted in **bold**, and the second-best results are underlined. The results are presented as mean ± standard deviation.

Methods	Human Activity		USHCN		PhysioNet		MIMIC-III	
	MSE ($\times 10^{-3}$)	MAE ($\times 10^{-2}$)	MSE ($\times 10^{-1}$)	MAE ($\times 10^{-1}$)	MSE ($\times 10^{-3}$)	MAE ($\times 10^{-2}$)	MSE ($\times 10^{-2}$)	MAE ($\times 10^{-2}$)
TimesNet (Wu et al., 2022)	3.79 ± 0.05	4.28 ± 0.04	5.62 ± 0.12	3.56 ± 0.12	9.30 ± 0.70	5.50 ± 0.34	2.34 ± 0.54	8.09 ± 0.10
PatchTST (Nie et al., 2022)	5.21 ± 0.33	5.10 ± 0.20	5.88 ± 0.10	3.66 ± 0.13	25.56 ± 4.42	10.90 ± 1.08	7.24 ± 0.65	19.80 ± 0.87
Crossformer (Zhang & Yan, 2023)	4.29 ± 0.10	4.89 ± 0.08	5.25 ± 0.07	3.27 ± 0.55	6.66 ± 0.11	4.81 ± 0.11	2.65 ± 0.50	9.56 ± 0.97
GRU-D (Che et al., 2018)	3.94 ± 0.29	4.37 ± 0.21	5.17 ± 0.06	3.21 ± 0.05	5.76 ± 0.34	4.53 ± 0.15	2.35 ± 0.06	8.34 ± 0.22
Raindrop (Zhang et al., 2022)	4.42 ± 0.25	4.65 ± 0.14	5.64 ± 0.10	3.29 ± 0.03	10.63 ± 0.29	6.02 ± 0.19	2.31 ± 0.07	8.61 ± 0.12
Warpformer (Zhang et al., 2023)	2.79 ± 0.02	3.39 ± 0.01	5.25 ± 0.03	3.23 ± 0.04	5.94 ± 0.14	4.21 ± 0.06	1.73 ± 0.30	7.58 ± 0.15
mTAND (Shukla & Marlin, 2021)	3.14 ± 0.09	3.71 ± 0.06	5.03 ± 0.05	3.00 ± 0.06	6.18 ± 0.31	4.44 ± 0.19	2.15 ± 0.05	8.00 ± 0.06
Latent-ODE (Rubanova et al., 2019)	3.32 ± 0.10	3.91 ± 0.08	5.16 ± 0.04	3.21 ± 0.07	6.85 ± 0.28	4.77 ± 0.17	2.11 ± 0.15	7.76 ± 0.08
CRU (Schirmer et al., 2022)	3.03 ± 0.04	3.60 ± 0.04	5.15 ± 0.50	3.18 ± 0.03	6.43 ± 0.62	4.51 ± 0.16	2.23 ± 0.03	7.99 ± 0.22
NeuralFlow (Biloš et al., 2021)	4.29 ± 0.63	4.61 ± 0.43	5.41 ± 0.05	3.35 ± 0.06	7.68 ± 0.37	4.84 ± 0.19	2.34 ± 0.05	8.09 ± 0.09
T-PatchGNN (Zhang et al., 2024)	2.66 ± 0.03	3.15 ± 0.02	5.00 ± 0.04	3.08 ± 0.04	4.98 ± 0.08	3.72 ± 0.03	1.36 ± 0.02	6.56 ± 0.11
Hi-Patch (Luo et al., 2025)	2.57 ± 0.02	3.11 ± 0.03	4.94 ± 0.05	2.96 ± 0.04	4.86 ± 0.03	3.62 ± 0.07	1.75 ± 0.26	7.24 ± 0.18
VIMTS (Hu et al., 2025)	2.65 ± 0.01	3.08 ± 0.01	4.86 ± 0.02	2.98 ± 0.05	4.81 ± 0.02	3.54 ± 0.05	1.36 ± 0.02	6.40 ± 0.17
ContiFormer (Chen et al., 2023)	2.84 ± 0.03	3.26 ± 0.02	4.96 ± 0.04	3.16 ± 0.04	5.14 ± 0.08	4.05 ± 0.03	1.81 ± 0.02	7.75 ± 0.11
ASTGI (Liu et al., 2026)	2.57 ± 0.05	**3.05** ± 0.04	5.23 ± 0.12	3.15 ± 0.12	4.63 ± 0.50	3.66 ± 0.34	1.32 ± 0.02	6.45 ± 0.11
TFMixer (Qiu et al., 2026)	**2.54 ± 0.06**	3.07 ± 0.03	4.88 ± 0.06	2.98 ± 0.12	4.74 ± 0.20	3.68 ± 0.04	1.45 ± 0.02	6.72 ± 0.11
LNSDE (Oh et al., 2024)	3.06 ± 0.05	3.47 ± 0.04	5.05 ± 0.08	3.20 ± 0.10	5.22 ± 0.07	4.17 ± 0.04	1.94 ± 0.04	7.76 ± 0.10
RosFormer	**2.54 ± 0.02**	**3.05 ± 0.02**	**4.84 ± 0.01**	**2.92 ± 0.04**	**4.55 ± 0.03**	**3.46 ± 0.05**	**1.30 ± 0.02**	**6.32 ± 0.15**

A.5 Distribution Shift Experiment

In this section, we evaluate RosFormer under two complementary distribution-shift settings introduced in **Robustness under Distribution Shift** of Sec. 5.2: a controlled test-time missingness shift on four forecasting benchmarks and a natural hospital-disjoint shift on eICU. The former isolates the effect of increasing test-time missingness, whereas the latter evaluates robustness to naturally occurring distribution differences across hospitals.

A.5.1 Controlled Test-Time Missingness Shift

Since the experimental protocol has already been described in the main text, we focus on the detailed performance comparison and the degradation trend under increasing missing ratios. Specifically, all models are trained on the original training split, while additional observations and their corresponding timestamps are randomly removed only from the test inputs. This setting changes the test-time observation density and therefore provides a controlled way to evaluate robustness under missingness-induced distribution shift.

Detailed numerical results. Tables 5, 6, 7, and 8 report the full results on Human Activity, PhysioNet, USHCN, and MIMIC-III, respectively. In the main text, we visualize the degradation trends on MIMIC-III in Fig. 3b, while the complete numerical results for all four forecasting datasets are provided here. Overall, increasing the missing ratio makes the prediction task more difficult, and most methods exhibit clear performance degradation. This confirms that the additional test-time missingness introduces a non-trivial distribution shift. Nevertheless, RosFormer remains highly competitive across different missing ratios and achieves the best performance under the most severe 70% missing setting on MIMIC-III, which is the dataset visualized in the main text.

On MIMIC-III, the MSE of RosFormer increases from 1.30×10^{-2} under the full test input to 1.53×10^{-2} under 70% missingness. Although the prediction error naturally reaches its largest value at the highest missing ratio, the degradation of RosFormer is relatively mild compared with most baselines. In particular, under 70% missingness, RosFormer obtains the lowest MSE and MAE, outperforming strong IMTS baselines such as VIMTS and ASTGI. On PhysioNet, RosFormer shows a similar trend: its MSE increases gradually from 4.55×10^{-3} to 6.73×10^{-3}, while still achieving the best MSE and MAE under the 70% missing setting. These results indicate that RosFormer can better preserve predictive stability when the test-time observation process becomes substantially sparser than that during training.

Robustness trend. To further visualize the degradation pattern, Fig. 4 presents the performance curves under different missing ratios. For better readability, we omit several curves with extremely large errors from the corresponding subfigures: Hi-Patch and ContiFormer are omitted in Fig. 4(a), and Hi-Patch is omitted in Fig. 4(b). Their complete numerical results are still reported in Tables 5 and 6. The curves show that the prediction error of RosFormer increases smoothly as the missing

Table 5: Complete performance results under varying test-time missing rates on Human Activity. The best results are highlighted in **bold**, and the second-best results are in underlined.

Methods	Full		30%		50%		70%	
	MSE$\times 10^{-3}$	MAE$\times 10^{-2}$	MSE$\times 10^{-3}$	MAE$\times 10^{-2}$	MSE$\times 10^{-3}$	MAE$\times 10^{-2}$	MSE$\times 10^{-3}$	MAE$\times 10^{-2}$
T-PatchGNN (Zhang et al., 2024)	2.66 ± 0.03	3.15 ± 0.02	3.00 ± 0.04	3.54 ± 0.03	3.48 ± 0.03	3.95 ± 0.01	4.73 ± 0.04	4.84 ± 0.04
Hi-Patch (Luo et al., 2025)	2.57 ± 0.02	3.11 ± 0.03	382.7 ± 0.22	60.5 ±0.01	382.4 ± 0.22	60.5 ± 0.01	382.3 ± 0.22	60.5 ± 0.01
VIMTS (Hu et al., 2025)	2.65 ± 0.01	3.08 ± 0.01	2.82 ± 0.02	3.23 ± 0.02	3.11 ± 0.01	3.46 ± 0.01	4.30 ± 0.05	4.12 ± 0.06
ContiFormer (Chen et al., 2023)	2.84 ± 0.03	3.26 ± 0.02	52.77 ± 0.50	19.83 ± 0.40	104.77 ± 0.72	30.00 ± 0.82	156.77 ± 0.80	37.58 ± 0.60
LNSDE (Oh et al., 2024)	3.06 ± 0.05	3.47 ± 0.04	3.27 ± 0.04	3.83 ± 0.04	3.52 ± 0.06	4.03 ± 0.04	3.87 ± 0.05	4.38 ± 0.04
ASTGI (Liu et al., 2026)	2.57 ± 0.05	**3.05 ± 0.04**	**2.68 ± 0.05**	**3.17 ± 0.04**	**2.88 ± 0.06**	**3.33 ± 0.04**	3.45 ± 0.04	3.76 ± 0.04
TFMixer (Qiu et al., 2026)	**2.54 ± 0.06**	3.07 ± 0.03	3.04 ± 0.04	3.29 ± 0.02	3.32 ± 0.05	3.46 ± 0.04	4.43 ± 0.04	3.87 ± 0.02
RosFormer	**2.54 ± 0.02**	**3.05 ± 0.02**	2.71 ± 0.02	3.24 ± 0.02	2.89 ± 0.01	3.38 ± 0.02	**3.37 ± 0.02**	**3.68 ± 0.03**

Table 6: Complete performance results under varying test-time missing rates on PhysioNet. The best results are highlighted in **bold**, and the second-best results are in underlined.

Methods	Full		30%		50%		70%	
	MSE$\times 10^{-3}$	MAE$\times 10^{-2}$	MSE$\times 10^{-3}$	MAE$\times 10^{-2}$	MSE$\times 10^{-3}$	MAE$\times 10^{-2}$	MSE$\times 10^{-3}$	MAE$\times 10^{-2}$
T-PatchGNN (Zhang et al., 2024)	4.98 ± 0.08	3.72 ± 0.03	5.30 ± 0.07	4.03 ± 0.03	5.94 ± 0.05	4.30 ± 0.04	7.18 ± 0.08	4.77 ± 0.06
Hi-Patch (Luo et al., 2025)	4.86 ± 0.03	3.62 ± 0.07	170.3 ± 0.32	30.8 ±0.01	169.7 ± 0.24	30.7 ± 0.02	168.5 ± 0.30	30.6 ± 0.01
VIMTS (Hu et al., 2025)	4.81 ± 0.02	3.54 ± 0.05	5.41 ± 0.02	3.89 ± 0.05	6.05 ± 0.02	4.23 ± 0.01	7.27 ± 0.02	4.69 ± 0.02
ContiFormer (Chen et al., 2023)	5.14 ± 0.08	4.05 ± 0.03	6.53 ± 0.05	4.81 ± 0.02	7.68 ± 0.08	5.21 ± 0.02	9.75 ± 0.06	5.70 ± 0.02
LNSDE (Oh et al., 2024)	5.22 ± 0.07	4.17 ± 0.04	5.73 ± 0.06	4.59 ± 0.04	6.22 ± 0.05	4.76 ± 0.04	7.61 ± 0.07	5.46 ± 0.03
ASTGI (Liu et al., 2026)	4.63 ± 0.50	3.66 ± 0.34	5.38 ± 0.40	3.96 ± 0.26	6.21 ± 0.42	4.35 ± 0.30	8.83 ± 0.42	5.30 ± 0.26
TFMixer (Qiu et al., 2026)	4.74 ± 0.20	3.68 ± 0.04	5.94 ± 0.18	4.28 ± 0.04	6.57 ± 0.20	4.54 ± 0.02	8.61 ± 0.12	5.06 ± 0.04
RosFormer	**4.55 ± 0.03**	**3.46 ± 0.05**	**5.00 ± 0.02**	**3.86 ± 0.02**	**5.61 ± 0.03**	**4.08 ± 0.02**	**6.73 ± 0.02**	**4.60 ± 0.03**

ratio grows, which is also consistent with the complete numerical results in Tables 5, 6, 7, and 8. In contrast, some baselines suffer from abrupt error increases when additional observations are removed from the test inputs, suggesting that their learned representations are more sensitive to the observation pattern seen during training. The stable degradation trend of RosFormer supports its robustness under distribution shifts caused by changing sampling density and missingness patterns.

A.5.2 Natural Hospital-Disjoint Shift on eICU

Dataset construction. To complement the controlled missingness experiments, we further evaluate RosFormer under a naturally occurring cross-hospital distribution shift using eICU-CRD Pollard et al. (2019). ICU stays from source hospitals are assigned to the training and validation sets, whereas the test set contains only stays from previously unseen target hospitals. The training, validation, and test sets approximately follow a 6:2:2 ratio. We use a predefined set of 12 physiological and laboratory variables collected during the first 24 hours of each ICU stay, comprising three non-invasive blood pressure measurements (systolic, diastolic, and mean) and nine commonly measured laboratory variables. The original timestamps and observation masks are retained without artificially removing observations, such that the distribution differences arise naturally from the hospital-disjoint setting.

Distribution-shift analysis. We analyze the differences between the training and test sets from two complementary perspectives. Figure 5 compares their observation densities across variables and time intervals. The changes are variable- and time-dependent rather than a uniform increase or decrease in missingness. Such heterogeneous changes may alter both the observations contributing to each patch and the temporal anchors of different variables.

Figure 6 further compares the Lomb–Scargle spectra of the recorded variable values. Differences between the training and test curves indicate that the two hospital groups also exhibit different frequency-domain characteristics. Together, the two analyses show that the hospital-disjoint split contains natural differences in both observation patterns and temporal signal characteristics, without artificially modifying the test data.

Table 7: Complete performance results under varying test-time missing rates on USHCN. The best results are highlighted in **bold**, and the second-best results are in underlined.

Methods	Full		30%		50%		70%	
	MSE×10^{-1}	MAE×10^{-1}	MSE×10^{-1}	MAE×10^{-1}	MSE×10^{-1}	MAE×10^{-1}	MSE×10^{-1}	MAE×10^{-1}
T-PatchGNN (Zhang et al., 2024)	5.00 ± 0.04	3.08 ± 0.04	5.10 ± 0.04	3.12 ± 0.03	5.13 ± 0.02	3.14 ± 0.02	5.16 ± 0.03	3.17 ± 0.01
Hi-Patch (Luo et al., 2025)	4.94 ± 0.05	2.96 ± 0.04	4.95 ± 0.04	2.98 ± 0.02	4.98 ± 0.04	2.96 ± 0.03	5.04 ± 0.04	3.00 ± 0.02
VIMTS (Hu et al., 2025)	4.86 ± 0.02	2.98 ± 0.05	4.93 ± 0.02	2.96 ± 0.03	4.96 ± 0.04	2.96 ± 0.03	4.97 ± 0.05	**2.96 ± 0.04**
ContiFormer (Chen et al., 2023)	4.96 ± 0.04	3.16 ± 0.04	5.31 ± 0.04	3.36 ± 0.04	5.48 ± 0.06	3.45 ± 0.04	5.86 ± 0.02	3.64 ± 0.02
LNSDE (Oh et al., 2024)	5.05 ± 0.08	3.20 ± 0.10	5.07 ± 0.08	3.22 ± 0.08	5.13 ± 0.12	3.24 ± 0.10	5.15 ± 0.14	3.27 ± 0.12
ASTGI (Liu et al., 2026)	5.23 ± 0.12	3.15 ± 0.12	5.30 ± 0.10	3.19 ± 0.08	5.42 ± 0.12	3.27 ± 0.10	5.77 ± 0.15	3.48 ± 0.15
TFMixer (Qiu et al., 2026)	4.88 ± 0.06	2.98 ± 0.12	4.98 ± 0.06	3.11 ± 0.08	5.00 ± 0.08	3.08 ± 0.10	5.07 ± 0.04	3.12 ± 0.05
RosFormer	**4.84 ± 0.01**	**2.92 ± 0.04**	**4.86 ± 0.01**	**2.92 ± 0.02**	**4.92 ± 0.01**	**2.95 ± 0.01**	**4.96 ± 0.02**	2.98 ± 0.01

Table 8: Complete performance results under varying test-time missing rates on MIMIC-III. The best results are highlighted in **bold**, and the second-best results are in underlined.

Methods	Full		30%		50%		70%	
	MSE×10^{-2}	MAE×10^{-2}	MSE×10^{-2}	MAE×10^{-2}	MSE×10^{-2}	MAE×10^{-2}	MSE×10^{-2}	MAE×10^{-2}
T-PatchGNN (Zhang et al., 2024)	1.36 ± 0.02	6.56 ± 0.11	1.73 ± 0.04	7.60 ± 0.12	1.83 ± 0.03	7.84 ± 0.06	1.85 ± 0.04	7.97 ± 0.08
Hi-Patch (Luo et al., 2025)	1.75 ± 0.26	7.24 ± 0.18	1.77 ± 0.20	7.48 ± 0.16	1.80 ± 0.24	7.66 ± 0.14	1.93 ± 0.20	7.92 ± 0.14
VIMTS (Hu et al., 2025)	1.36 ± 0.02	6.40 ± 0.17	1.71 ± 0.02	7.12 ± 0.10	1.80 ± 0.01	7.34 ± 0.12	1.83 ± 0.02	7.42 ± 0.14
ContiFormer (Chen et al., 2023)	1.81 ± 0.02	7.75 ± 0.11	2.03 ± 0.04	8.64 ± 0.11	2.11 ± 0.02	8.73 ± 0.12	2.16 ± 0.02	8.77 ± 0.10
LNSDE (Oh et al., 2024)	1.94 ± 0.04	7.76 ± 0.10	2.02 ± 0.04	8.10 ± 0.10	2.10 ± 0.03	8.52 ± 0.08	2.15 ± 0.05	8.70 ± 0.12
ASTGI (Liu et al., 2026)	1.32 ± 0.02	6.45 ± 0.11	1.85 ± 0.02	7.67 ± 0.10	2.08 ± 0.03	8.35 ± 0.12	2.32 ± 0.02	9.17 ± 0.14
TFMixer (Qiu et al., 2026)	1.45 ± 0.02	6.72 ± 0.11	1.78 ± 0.02	7.49 ± 0.11	1.82 ± 0.02	7.93 ± 0.12	1.94 ± 0.03	8.74 ± 0.10
RosFormer	**1.30 ± 0.02**	**6.32 ± 0.15**	**1.43 ± 0.01**	**6.68 ± 0.08**	**1.49 ± 0.01**	**6.78 ± 0.10**	**1.53 ± 0.02**	**7.12 ± 0.10**

Hospital-disjoint forecasting results. As reported in Table 2, RosFormer achieves the lowest MSE and MAE among all compared methods. These results demonstrate its effectiveness under a real-world hospital-disjoint distribution shift with naturally different observation patterns.

A.6 Full Experimental Results of Event Prediction

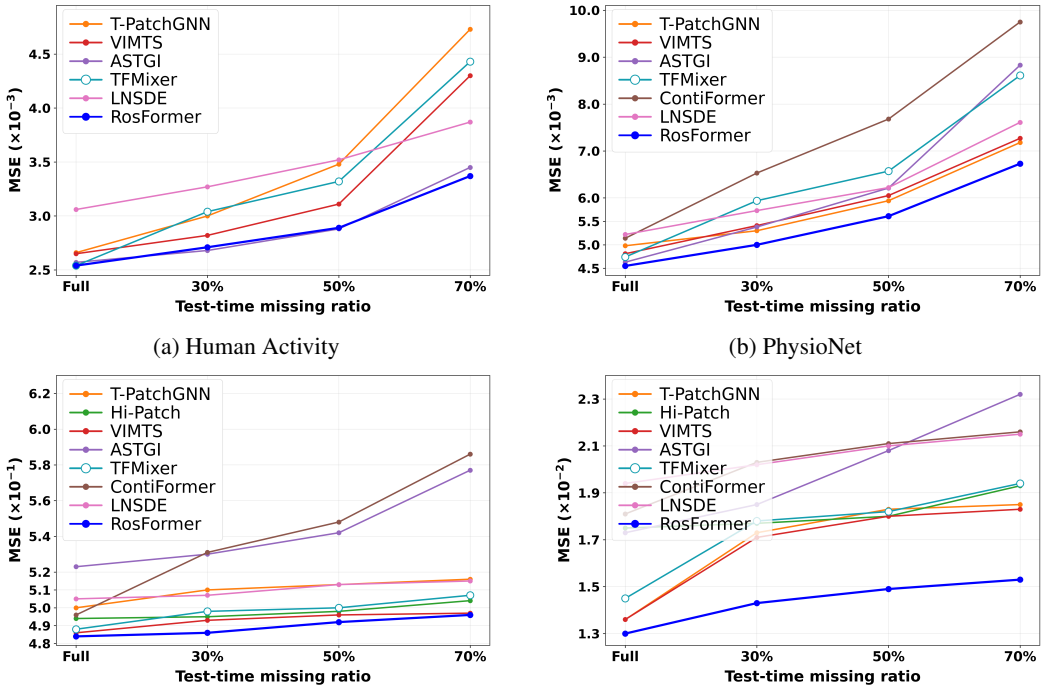
For irregular event prediction, we use one synthetic dataset and five real-world datasets: **Neonate** (Stevenson et al., 2019), **Traffic** (Lai et al., 2018), **MIMIC** (Du et al., 2016), **BookOrder** (Du et al., 2016), and **StackOverflow** (Jure, 2014). These datasets cover various scenarios with complex event dependencies. Besides the irregular time series baselines introduced above, we additionally compare with representative event prediction baselines, including RMTPP (Du et al., 2016), ODE-RNN (Rubanova et al., 2019), and SAHP (Zhang et al., 2020).The results of these three event prediction baselines are reported in the Appendix A.6. The evaluation metrics include log-likelihood (LL), event-type accuracy, and RMSE of time prediction.

We report the complete event prediction results in Table 9. The metrics include log-likelihood (LL), accuracy, and RMSE. Higher LL and accuracy indicate better performance, while lower RMSE indicates better performance.

B Computational Efficiency Analysis

In this appendix, we provide a detailed efficiency analysis of RosFormer. All efficiency results are evaluated on the Activity dataset under the IMTS forecasting setting. For a fair comparison, all models are trained and evaluated with the same batch size of 32. We report the training and inference times per epoch, along with GPU memory consumption, in Table 10.

The results show that standalone LNSDE modeling is computationally expensive, requiring 514.15s training latency and 268.61s inference latency. To examine the cost of trajectory modeling within continuous-time attention, we consider two ContiFormer variants. The first replaces spline inter-



(a) Human Activity (b) PhysioNet

(c) USHCN (d) MIMIC-III

Figure 4: Performance degradation under increasing test-time missing ratios. Lower error indicates better performance. For better readability, curves with extremely large errors are omitted from the corresponding subfigures, including Hi-Patch and ContiFormer in Fig. 4(a), and Hi-Patch in Fig. 4(b). Their complete numerical results are reported in Tables 5, 6, 7, and 8. RosFormer exhibits a smooth error growth trend as the missing ratio increases, showing stronger robustness to observation-process distribution shifts.

polation for Q with an additional ODE, using independent ODEs for Q, K, and V. This increases training and inference latency from 69.69s and 26.27s to 180.65s and 128.42s, respectively. Replacing these three ODEs with independent LNSDEs yields training and inference latencies of 113.26s and 42.35s. Although this stochastic variant is faster than the independent ODE variant, its GPU memory consumption increases from 2865MB to 8212MB.

Introducing patch-level modeling yields the RosFormer variant with independent LNSDEs for Q, K, and V, reducing training latency to 25.32s, inference latency to 9.26s, and GPU memory consumption to 5828MB. Compared with the ContiFormer variant using independent LNSDEs, these represent reductions of approximately 77.6%, 78.1%, and 29.0%, respectively. These gains arise from evolving patch representations, which reduces the number of trajectories requiring numerical integration.

Sharing a single stochastic latent trajectory further reduces training latency to 8.55s, inference latency to 3.76s, and GPU memory consumption to 4048MB. Compared with RosFormer using independent LNSDEs, the shared design reduces these costs by approximately 66.2%, 59.4%, and 30.5%, respectively. It derives Q, K, and V through separate linear projections of the shared trajectory, avoiding redundant numerical integration.

Compared with T-PatchGNN, RosFormer has slightly higher training latency (8.55s versus 7.00s) and slightly lower inference latency (3.76s versus 3.85s), although it requires more GPU memory (4048MB versus 1912MB). Together, patch-level modeling and trajectory sharing substantially reduce the computational overhead of stochastic continuous-time attention.

Continuous-time integration with interval reparameterization. The continuous-time attention score in Eq. in **Compactness-aware continuous-time attention** of 3.4 requires evaluating trajectory-based integrals over different time intervals. Following ContiFormer (Chen et al., 2023),

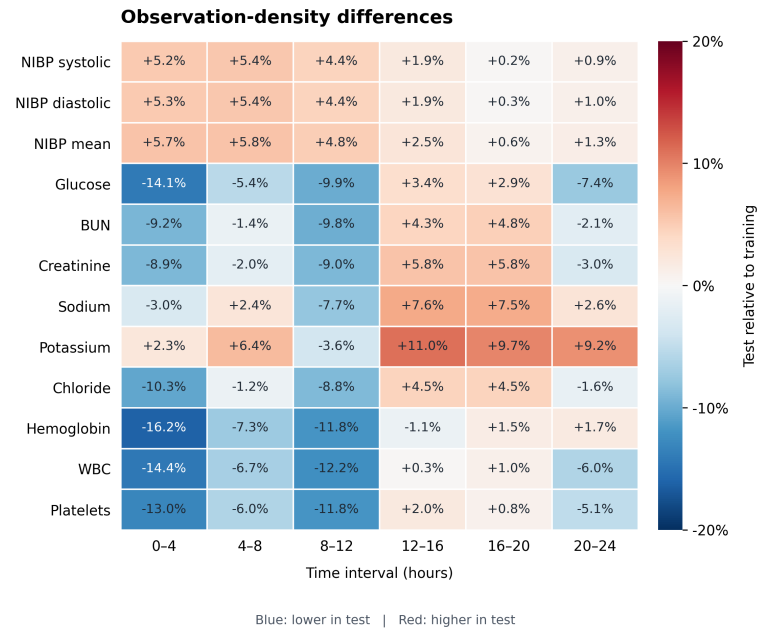


Figure 5: Observation-density differences between the hospital-disjoint training and test sets on eICU. Each cell shows the relative change in test observation density with respect to training for one variable and time interval. Blue and red indicate lower and higher density in the test set, respectively.

we adopt the time-variable reparameterization to transform each interval into a unified interval $[-1, 1]$, and then apply numerical approximation to estimate the corresponding integrals.

Specifically, for the interval $[\bar{t}_p, t]$, we define

$$\tau = \frac{(t - \bar{t}_p)s + \bar{t}_p + t}{2}, \qquad s \in [-1, 1].$$

Then Eq. in **Compactness-aware continuous-time attention** of 3.4 can be rewritten as

$$\begin{aligned}
\alpha_p^n(t) &= \frac{1}{t - \bar{t}_p} \int_{\bar{t}_p}^{t} \mathbf{q}_p^n(\tau)\, \mathbf{k}_p^n(\tau)^\top d\tau \\
&= \frac{1}{2} \int_{-1}^{1} \mathbf{q}_p^n\left(\frac{(t - \bar{t}_p)s + \bar{t}_p + t}{2}\right) \mathbf{k}_p^n\left(\frac{(t - \bar{t}_p)s + \bar{t}_p + t}{2}\right)^\top ds \\
&\approx \frac{1}{2} \sum_{\ell=1}^{M} \gamma_\ell \mathbf{q}_p^n\left(\frac{(t - \bar{t}_p)\xi_\ell + \bar{t}_p + t}{2}\right) \mathbf{k}_p^n\left(\frac{(t - \bar{t}_p)\xi_\ell + \bar{t}_p + t}{2}\right)^\top,
\end{aligned}$$

where $\xi_\ell \in [-1, 1]$ and γ_ℓ denote the quadrature points and weights, respectively.

The same reparameterization is also applied when solving the LNSDE trajectory. Recall that the latent trajectory is defined as

$$z_p^n(t) = z_p^n(\bar{t}_p^n) + \int_{\bar{t}_p^n}^{t} f_\theta(\tau, z_p^n(\tau))\, d\tau + \int_{\bar{t}_p^n}^{t} g_\sigma(\tau)\, z_p^n(\tau)\, dW_\tau. \tag{16}$$

For the interval $[\bar{t}_p^n, t]$, we define

$$\phi_{p,t}^n(s) = \frac{(t - \bar{t}_p^n)s + \bar{t}_p^n + t}{2}, \qquad s \in [-1, 1],$$

and

$$r_{p,t}^n = \frac{t - \bar{t}_p^n}{2}.$$

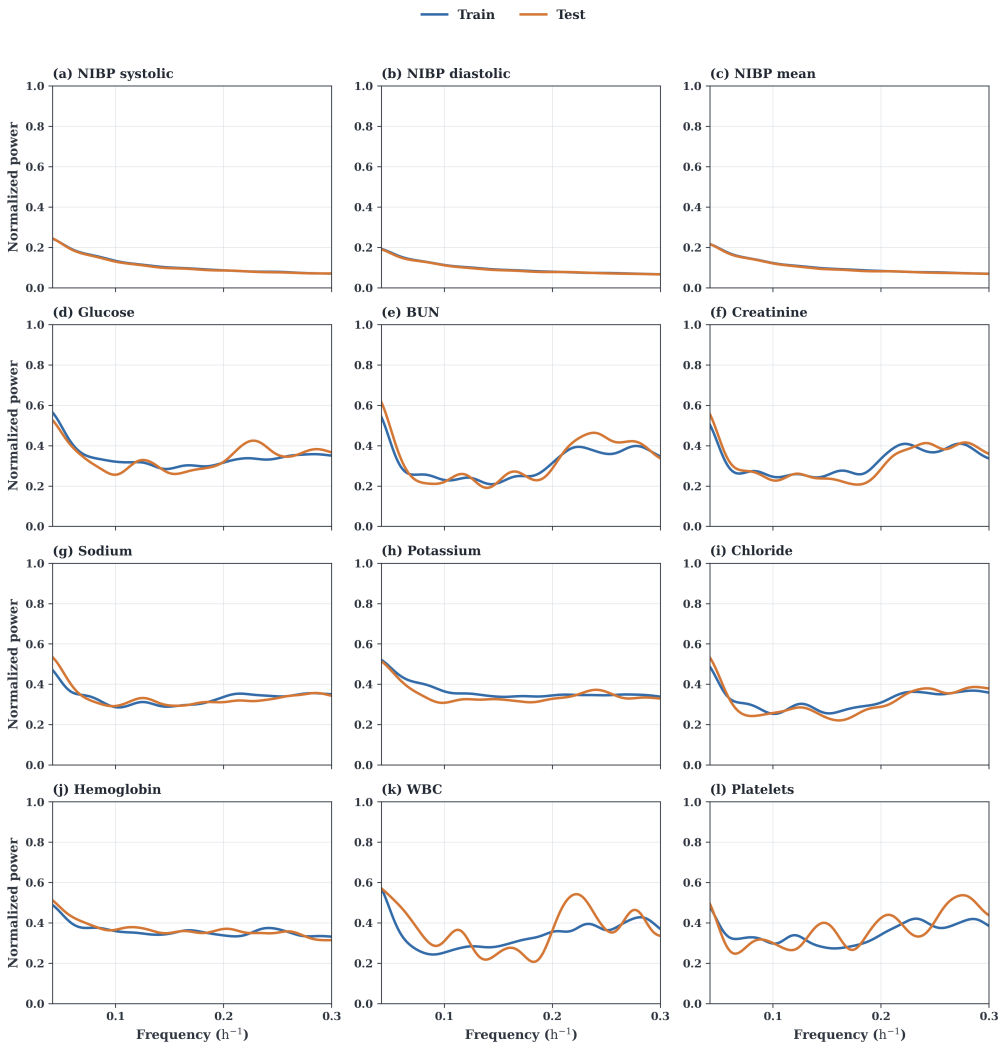


Figure 6: Lomb–Scargle spectra of recorded variable values in the hospital-disjoint training and test sets of eICU-CRD. For each variable, the curves show smoothed, normalized Lomb–Scargle power averaged over eligible ICU stays in each split.

Then the drift integral can be transformed as

$$\int_{\bar{t}_p^n}^{t} f_\theta\big(\tau, z_p^n(\tau)\big)\, d\tau = \int_{-1}^{1} r_{p,t}^n f_\theta\big(\phi_{p,t}^n(s), z_p^n(\phi_{p,t}^n(s))\big)\, ds.$$

For the stochastic integral, the Brownian increment is rescaled under the time change as

$$dW_\tau = \sqrt{|r_{p,t}^n|}\, d\widetilde{W}_s.$$

Therefore, the diffusion integral becomes

$$\int_{\bar{t}_p^n}^{t} g_\sigma(\tau) z_p^n(\tau)\, dW_\tau = \int_{-1}^{1} \sqrt{|r_{p,t}^n|}\, g_\sigma\big(\phi_{p,t}^n(s)\big) z_p^n\big(\phi_{p,t}^n(s)\big) d\widetilde{W}_s.$$

As a result, the LNSDE trajectory can be evaluated on the normalized interval $[-1, 1]$ as

$$\begin{aligned} z_p^n(t) = z_p^n(\bar{t}_p^n) &+ \int_{-1}^{1} r_{p,t}^n f_\theta\big(\phi_{p,t}^n(s), z_p^n(\phi_{p,t}^n(s))\big)\, ds \\ &+ \int_{-1}^{1} \sqrt{|r_{p,t}^n|}\, g_\sigma\big(\phi_{p,t}^n(s)\big) z_p^n\big(\phi_{p,t}^n(s)\big) d\widetilde{W}_s. \end{aligned} \tag{17}$$

Table 9: Full performance comparison on event prediction tasks. The best results are highlighted in **bold**, and the second-best results are underlined. - indicates numerical overflow

Model	Metric	Synthetic	Neonate	Traffic	MIMIC	StackOverflow	BookOrder
RMTPP (Du et al., 2016)	LL (↑)	-1.025 ± .030	-2.817 ± .023	-0.546 ± .012	-1.184 ± .023	-2.374 ± .001	-0.952 ± .007
	Accuracy (↑)	0.841 ± .003	–	0.805 ± .002	0.823 ± .004	0.461 ± .002	0.624 ± .003
	RMSE (↓)	0.369 ± .014	9.517 ± .023	0.337 ± .001	0.864 ± .017	0.955 ± .001	3.647 ± .003
ODE-RNN (Rubanova et al., 2019)	LL (↑)	-1.032 ± .017	-2.732 ± .045	**-0.491 ± .015**	-1.183 ± .038	-2.395 ± .001	-0.988 ± .052
	Accuracy (↑)	0.841 ± .002	–	0.812 ± .001	0.827 ± .005	0.467 ± .003	0.624 ± .004
	RMSE (↓)	0.342 ± .006	9.289 ± .045	0.334 ± .003	0.865 ± .017	0.952 ± .004	3.605 ± .004
SAHP (Zhang et al., 2020)	LL (↑)	-0.619 ± .063	-2.646 ± .057	-0.632 ± .003	-1.110 ± .030	-2.404 ± .002	-0.934 ± .030
	Accuracy (↑)	0.841 ± .004	–	0.780 ± .001	0.830 ± .004	0.455 ± .003	0.622 ± .002
	RMSE (↓)	0.521 ± .055	9.403 ± .060	0.337 ± .002	0.851 ± .006	0.963 ± .004	3.676 ± .011
T-PatchGNN (Zhang et al., 2024)	LL (↑)	-0.683 ± .028	-2.365 ± .026	-0.988 ± .019	-0.975 ± .023	-2.789 ± .001	-1.028 ± .010
	Accuracy (↑)	0.781 ± .003	–	0.732 ± .001	0.726 ± .006	0.430 ± .005	0.553 ± .001
	RMSE (↓)	0.368 ± .005	6.998 ± .033	0.405 ± .001	0.851 ± .007	1.081 ± .003	3.206 ± .020
Hi-Patch (Luo et al., 2025)	LL (↑)	-0.732 ± .024	-2.315 ± .022	-0.805 ± .020	-0.926 ± .020	-2.825 ± .001	-1.061 ± .008
	Accuracy (↑)	0.721 ± .003	–	0.800 ± .001	0.742 ± .004	0.428 ± .002	0.560 ± .001
	RMSE (↓)	0.443 ± .003	4.784 ± .028	0.389 ± .001	**0.786 ± .007**	1.079 ± .004	3.214 ± .025
VIMTS (Hu et al., 2025)	LL (↑)	-0.751 ± .025	-2.336 ± .022	-1.098 ± .014	-1.020 ± .020	-2.833 ± .001	-1.125 ± .010
	Accuracy (↑)	0.814 ± .002	–	0.541 ± .001	0.704 ± .003	0.436 ± .002	0.501 ± .001
	RMSE (↓)	0.454 ± .002	4.949 ± .025	0.416 ± .001	0.883 ± .004	1.073 ± .002	3.232 ± .015
ASTGI (Liu et al., 2026)	LL (↑)	-1.020 ± .030	-2.332 ± .023	-0.885 ± .012	-1.194 ± .023	-2.839 ± .001	-1.926 ± .007
	Accuracy (↑)	0.825 ± .003	–	0.740 ± .002	0.763 ± .004	0.430 ± .003	0.401 ± .004
	RMSE (↓)	0.345 ± .014	4.962 ± .023	0.418 ± .001	0.851 ± .017	1.081 ± .003	3.821 ± .003
TFMixer (Qiu et al., 2026)	LL (↑)	-1.189 ± .020	-2.256 ± .020	-0.576 ± .010	-0.862 ± .018	-2.515 ± .002	-0.925 ± .005
	Accuracy (↑)	0.776 ± .004	–	0.830 ± .003	0.847 ± .002	0.459 ± .002	0.630 ± .002
	RMSE (↓)	0.606 ± .014	4.645 ± .012	0.374 ± .001	0.818 ± .015	1.049 ± .002	3.338 ± .002
ContiFormer (Chen et al., 2023)	LL (↑)	-0.535 ± .032	-2.550 ± .024	-0.857 ± .019	-1.135 ± .026	-2.332 ± .001	-1.076 ± .015
	Accuracy (↑)	0.842 ± .002	–	0.776 ± .001	0.836 ± .003	0.473 ± .003	0.616 ± .001
	RMSE (↓)	0.192 ± .003	9.233 ± .030	0.387 ± .001	0.837 ± .004	0.948 ± .003	3.193 ± .016
LNSDE (Oh et al., 2024)	LL (↑)	-0.992 ± .034	-2.305 ± .020	-0.652 ± .012	-1.185 ± .026	-2.352 ± .001	-1.151 ± .005
	Accuracy (↑)	0.839 ± .003	–	0.778 ± .002	0.767 ± .003	0.470 ± .002	0.500 ± .002
	RMSE (↓)	0.322 ± .016	4.703 ± .020	0.363 ± .001	0.822 ± .015	0.949 ± .001	3.375 ± .001
RosFormer	LL (↑)	**-0.481 ± .020**	**-2.231 ± .022**	-0.510 ± .015	**-0.846 ± .018**	**-2.314 ± .002**	**-0.892 ± .008**
	Accuracy (↑)	**0.845 ± .001**	–	**0.832 ± .002**	**0.853 ± .005**	**0.497 ± .001**	**0.636 ± .002**
	RMSE (↓)	**0.180 ± .004**	**4.614 ± .030**	**0.315 ± .002**	0.825 ± .006	**0.935 ± .001**	**3.158 ± .015**

Table 10: Training/inference latency and GPU memory consumption of models on the Activity dataset. All models are evaluated under the same batch size of 32. ContiFormer (ODE/LNSDE for Q,K,V) means using three different ODEs/LNSDEs to define Q,K,V.

Model	T-PatchGNN	LNSDE	ContiFormer	ContiFormer (Independent ODEs for Q,K,V)
training latency (s)	7.00	514.15	69.69	180.65
inference latency (s)	3.85	268.61	26.27	128.42
GPU memory consumption (MB)	1912	3178	2742	2865

Model	ContiFormer (Independent LNSDEs for Q,K,V)	RosFormer (Independent LNSDEs for Q,K,V)	RosFormer
training latency (s)	113.26	25.32	8.55
inference latency (s)	42.35	9.26	3.76
GPU memory consumption (MB)	8212	5828	4048

In this way, all trajectory integrations are performed on the same normalized interval $[-1, 1]$, while the drift term and the diffusion term are rescaled by $r^n_{p,t}$ and $\sqrt{|r^n_{p,t}|}$, respectively.

Replacing spline-based query construction with trajectory-based modeling. In the original ContiFormer, the key and value functions are modeled as continuous-time trajectories with neural ODEs, while the query function is constructed by a closed-form interpolation function, such as natural cubic spline interpolation. Although this design provides a continuous query function, it

Table 11: Forecasting performance of independent and shared $Q/K/V$ trajectory construction. The independent variant solves three separate LNSDEs for Q, K, and V, whereas the shared variant derives them from one LNSDE trajectory. Lower values are better.

Model	PhysioNet		Human Activity	
	MSE (10^{-3})	MAE (10^{-2})	MSE (10^{-3})	MAE (10^{-2})
RosFormer (Independent $Q/K/V$)	4.58	3.47	**2.52**	**3.02**
RosFormer	**4.55**	**3.46**	2.54	3.05

requires additional interpolation and time-alignment operations over irregular timestamps, which introduces non-negligible computational overhead in practice.

To reduce this cost, we replace the spline-based query construction with trajectory-based modeling. Specifically, instead of constructing $q(t)$ through cubic spline interpolation over discrete query states, we model the latent representation as a continuous-time trajectory and obtain the query trajectory by a learnable linear projection:

$$q_p^n(t) = z_p^n(t)W_Q.$$

This makes the query trajectory share the same continuous-time modeling mechanism as the key and value trajectories, avoiding the additional spline interpolation and time-alignment procedure while preserving the continuous-time attention formulation.

Shared latent trajectory for efficient $Q/K/V$ generation. A straightforward implementation of stochastic continuous-time attention is to construct the query, key, and value trajectories using three independently solved LNSDEs: $\mathbf{q}p^n(t), \mathbf{k}p^n(t), \mathbf{v}_p^n(t)$.

Although this design provides separate dynamics for the three attention components, it requires three stochastic differential equation solves for each patch. The resulting numerical integration and stochastic sampling substantially increase training and inference costs.

RosFormer instead models their common temporal evolution using a single shared latent stochastic trajectory:

$$z_p^n(t) = z_p^n(\bar{t}_p^n) + \int_{\bar{t}_p^n}^t f_\theta\big(\tau, z_p^n(\tau)\big)d\tau + \int_{\bar{t}_p^n}^t g_\sigma(\tau)z_p^n(\tau)dW_\tau.$$

The query, key, and value trajectories are then obtained:

$$q_p^n(t) = z_p^n(t)W_Q, \qquad k_p^n(t) = z_p^n(t)W_K, \qquad v_p^n(t) = z_p^n(t)W_V.$$

Thus, the three attention components share the same stochastic temporal dynamics while retaining distinct query, key, and value representations through their respective projection matrices. This construction is consistent with standard attention, in which queries, keys, and values are projected from a common latent representation.

As shown in Table 10, compared with independently solving three LNSDE trajectories for $Q/K/V$, the shared construction reduces training latency by 66.2%, inference latency by 59.4%, and GPU memory consumption by 30.5%. Table 11 further compares their forecasting performance. The shared construction performs slightly better on PhysioNet, whereas the independent construction performs slightly better on Human Activity, with only small differences on both datasets. These results suggest that the separate projection matrices provide sufficient flexibility for the three attention components, while independent stochastic dynamics introduce substantial computational cost with limited empirical benefit.

Results. Direct stochastic trajectory modeling can be computationally expensive. As shown in Table 10, the standalone LNSDE baseline requires 514.15 s for training and 268.61 s for inference. For comparison, the original ContiFormer requires 69.69 s and 26.27 s, respectively.

RosFormer controls the number of trajectories by evolving variable-level patch embeddings rather than individual observations. Even when separate LNSDEs are solved for Q, K, and V, this variant

requires 25.32 s for training and 9.26 s for inference. The shared-trajectory design further reduces the number of LNSDE solves from three to one per patch. Compared with the independent $Q/K/V$ variant, it reduces training latency by 66.2%, inference latency by 59.4%, and GPU memory consumption by 30.5%.

This reduction in cost comes with only small differences in forecasting accuracy: the shared variant performs slightly better on PhysioNet, while the independent variant performs slightly better on Human Activity, as shown in Table 11. Overall, patch-level evolution and shared $Q/K/V$ construction make stochastic continuous-time attention computationally practical while retaining comparable empirical forecasting performance.

C STOCHASTIC STABILITY AND PROOFS FOR SECTION

C.1 BACKGROUND ON STABILITY OF SDES

Stability is an important concept that describes the behavior of a solution in a given differential equation with respect to changes in the initial conditions. It is well-known that stability of ODEs can be determined without solving the equation by using the Lyapunov direct method. Fortunately, it turns out that the Lyapunov technique for analyzing stability of ODEs can be applied to study stability of SDEs with slight modifications. This is significantly useful since explicit solutions for SDEs cannot be obtained except in special cases. In this Appendix, we provide a brief overview of stochastic stability. We refer to (Mao, 2007; Khasminskii, 2011) for more details.

Consider the d-dimensional SDE:

$$dX(t) = f(t, X(t))dt + g(t, X(t))dW(t), \tag{18}$$

with the initial value $x_0 \in \mathbb{R}^d$. We assume that Eq. 18 has a unique global solution denoted by $X(t; x_0)$ for all $t \geq 0$ with the initial value x_0. Furthermore, assume that

$$f(t, 0) = 0 \quad \text{and} \quad g(t, 0) = 0$$

for $t \geq 0$. Then, $X(t; 0) \equiv 0$ is the solution of Eq. 18 and is called the trivial solution. For $V \in C^{1,2}(\mathbb{R}_+ \times \mathbb{R}^d)$, we define the infinitesimal generator $\mathcal{G}$ of $z(t)$:

$$\mathcal{G}V(t, x) = V_t(t, x) + V_x(t, x)^\top f(t, x) + \frac{1}{2}\text{Tr}[g(t, x)g(t, x)^\top V_{xx}(t, x)] \tag{19}$$

where V_x is the gradient of V with respect to x and V_{xx} is the Hessian of V with respect to x.

In a stochastic system driven by a SDE, stability can be defined in several ways such as stability in probability, almost sure exponential stability, and moment exponential stability. For our purpose, we focus on almost sure exponential stability.

C.2 THEORETICAL ANALYSIS OF ROSFORMER

First, we introduce the definition of LNSDE and the assumptions adopted in our analysis.

Definition C.1. (LNSDE). Consider the d-dimensional Neural Linear Noise Stochastic Differential Equation (LNSDE) defined as:

$$dz(t) = f(t, z(t))dt + \sigma_\theta z(t)dW(t), \quad t \geq 0 \tag{20}$$

with the initial value $z(0) = z_0 \in \mathbb{R}^d$. Here, $W(t)$ is a d_z-dimensional Brownian motion, $f(t, \cdot) : \mathbb{R}^d \to \mathbb{R}^d$ denotes the drift function, and σ_θ represents the diffusion parameter. For our stability analysis, we assume that $g_\sigma(\tau)$ in Eq. 4 is either a constant σ_θ or admits a limiting value, i.e., $\lim_{\tau\to\infty} g_\sigma(\tau) = \sigma_\theta$.

Assumption C.2. For any neural network $s(t, x; \theta_s) : \mathbb{R}_+ \times \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ with parameter θ_s, there exists a positive constant $L_s > 0$ such that for all $t \geq 0$ and $x, x' \in \mathbb{R}^{d_1}$,

$$|s(t, x; \theta_s) - s(t, x'; \theta_s)| \leq L_s|x - x'|.$$

Assumption C.3. For any neural network $s(t, x; \theta_s) : \mathbb{R}_+ \times \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ with parameter θ_s where the last layer is the ReLU or linear function, there exists a positive constant $K_s > 0$ such that

$$|s(t, x; \theta_s)| \leq K_s(1 + |x|), \quad \text{for all } t \geq 0, x \in \mathbb{R}^{d_1}.$$

On the other hand, when tanh function or sigmoid function is applied at the last layer, we have

$$|s(t, x; \theta_s)| \leq K_s, \quad \text{for all } t \geq 0, x \in \mathbb{R}^{d_1}. \tag{21}$$

Assumption C.4. For any neural network $s(t, x; \theta_s) : \mathbb{R}_+ \times \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ with parameter θ_s, there exist positive constants $m, b > 0$ such that for all $t \geq 0, x \in \mathbb{R}^{d_1}$,

$$\langle s(t, x; \theta_s), x \rangle \leq -m|x|^2 + b. \tag{22}$$

We then present some lemmas from prior work, which will be used in our analysis.

Lemma C.5 (Boundedness of Second Moment for LNSDE Oh et al. (2024))**.** *Let $z(t)$ be the solution of the Neural Linear Noise SDE (LNSDE). Under Assumptions 3.1, 3.2, and 3.3, specifically if the drift function γ satisfies the dissipativity condition $\langle \gamma(t, x), x \rangle \leq -m|x|^2 + b$ for some $m > 0, b \geq 0$, then the second moment of the solution is uniformly bounded:*

$$\sup_{t \geq 0} \mathbb{E}[|z(t)|^2] \leq \left(\mathbb{E}[|z(0)|^2] + \frac{b}{m} \right) \exp\left\{ \frac{d_z \sigma_\theta^2}{2m} \right\}, \tag{23}$$

where d_z is the dimension of the latent state and σ_θ is the diffusion parameter.

Lemma C.6 (Exponential Decay of Error for LNSDE Oh et al. (2024))**.** *Consider two solutions $z(t)$ and $\tilde{z}(t)$ of the LNSDE with different initial values corresponding to the original input x and perturbed input $\tilde{x}$. Let $\epsilon(t) = z(t) - \tilde{z}(t)$ be the error process. If the noise intensity satisfies $|\sigma_\theta|^2 > 2L_f$, where L_f is the Lipschitz constant of the drift function, then for sufficiently large T, the expected error decays exponentially:*

$$\mathbb{E}[|\epsilon(T)|] \leq \exp\left\{ -\frac{|\sigma_\theta|^2 - 2L_f}{2} T \right\} (1 + \rho), \tag{24}$$

where ρ represents the bound on the initial perturbation.

Lemma C.7 (Lipschitz Continuity of Softmax Gao & Pavel (2017))**.** *The softmax function $\sigma(z)$ is Lipschitz continuous with respect to the Euclidean norm $||\cdot||_2$, with the Lipschitz constant determined by the inverse temperature parameter L_{soft}. That is, for all $z, z' \in \mathbb{R}^n$:*

$$||\sigma(z) - \sigma(z')||_2 \leq L_{soft} ||z - z'||_2. \tag{25}$$

Corollary C.8 (Expectation Bound for Softmax Differences)**.** *Let z and z' be random vectors (representing the original and perturbed inputs). The root mean square distance of the softmax outputs is bounded by the root mean square distance of the inputs, scaled by λ:*

$$\sqrt{\mathbb{E}\left[||\sigma(z) - \sigma(z')||_2^2\right]} \leq L_{soft} \sqrt{\mathbb{E}\left[||z - z'||_2^2\right]}. \tag{26}$$

Proof. The derivation proceeds in three steps:

1. **Square:** Since both sides of the Lipschitz inequality (Lemma 1) are non-negative, squaring them preserves the inequality:

$$||\sigma(z) - \sigma(z')||_2^2 \leq L_{soft}^2 ||z - z'||_2^2. \tag{27}$$

2. **Expectation:** Assuming z and z' are random variables, we apply the expectation operator $\mathbb{E}[\cdot]$ to both sides. By the monotonicity of expectation (i.e., if $X \leq Y$ almost surely, then $\mathbb{E}[X] \leq \mathbb{E}[Y]$):

$$\mathbb{E}\left[||\sigma(z) - \sigma(z')||_2^2\right] \leq L_{soft}^2 \mathbb{E}\left[||z - z'||_2^2\right]. \tag{28}$$

3. **Square Root:** Finally, taking the square root of both sides. Since the square root function $f(x) = \sqrt{x}$ is monotonically increasing for $x \geq 0$:

$$\sqrt{\mathbb{E}\left[||\sigma(z) - \sigma(z')||_2^2\right]} \leq \sqrt{L_{soft}^2 \mathbb{E}\left[||z - z'||_2^2\right]} = L_{soft} \sqrt{\mathbb{E}\left[||z - z'||_2^2\right]}. \tag{29}$$

This completes the derivation. □

C.3 Proof of Boundedness for Higher Moments

C.3.1 Boundedness of 4th Moment

We provide the proof for the 4th moment ($p = 4$) and the 8th moment ($p = 8$).

Let $Y(t) := |z(t)|^4$ for all $t \geq 0$. Applying Itô's formula to $e^{4mt}Y(t)$, we obtain:

$$\begin{aligned} d(e^{4mt}Y(t)) &= 4me^{4mt}Y(t)dt + e^{4mt}d(|z(t)|^4) \\ &= 4me^{4mt}|z(t)|^4\,dt + e^{4mt}\Bigg(4|z(t)|^2\big\langle z(t), f(t,z(t))\,dt + \sigma_\theta z(t)\,dW_t\big\rangle \\ &\quad + \frac{1}{2}\,\mathrm{Tr}\Big(d_z\sigma_\theta^2 z(t)z(t)^\top\big(8z(t)z(t)^\top + 4|z(t)|^2 I\big)\Big)\,dt\Bigg) \end{aligned}$$

Using the drift condition $\langle f(t,z), z\rangle \leq -m|z|^2 + b$, we have:

$$\begin{aligned} d(e^{4mt}Y(t)) &\leq 4me^{4mt}Y(t)dt + e^{4mt}\left(4|z(t)|^2(-m|z(t)|^2 + b) + 6d_z\sigma_\theta^2|z(t)|^4\right)dt + 4\sigma_\theta e^{4mt}|z(t)|^4 dW_t \\ &= e^{4mt}\left(4b|z(t)|^2 + 6d_z\sigma_\theta^2 Y(t)\right)dt + 4\sigma_\theta e^{4mt}|z(t)|^4 dW_t, \end{aligned}$$

where $4\sigma_\theta e^{4mt}|z(t)|^4 dW_t$ is a martingale term. Integrating from 0 to t:

$$Y(t) \leq e^{-4mt}Y(0) + \int_0^t e^{4m(s-t)}\left(4b|z(s)|^2 + 6d\sigma_\theta^2 Y(s)\right)ds + \text{martingale}.$$

Taking expectations and using the uniform boundedness of the second moment $K_2 := \sup_{t\geq 0}\mathbb{E}[|z(t)|^2]$ (from Theorem C.5):

$$\begin{aligned} \mathbb{E}[Y(t)] &\leq e^{-4mt}\mathbb{E}[Y(0)] + 4bK_2\int_0^t e^{4m(s-t)}ds + 6d\sigma_\theta^2\int_0^t e^{4m(s-t)}\mathbb{E}[Y(s)]ds \\ &= e^{-4mt}\mathbb{E}[Y(0)] + \frac{bK_2}{m}(1 - e^{-4mt}) + 6d\sigma_\theta^2\int_0^t e^{4m(s-t)}\mathbb{E}[Y(s)]ds. \end{aligned}$$

Using Gronwall's inequality leads to:

$$\begin{aligned} \mathbb{E}[Y(t)] &\leq \left(\mathbb{E}[Y(0)] + \frac{bK_2}{m}(1 - e^{-4mt})\right)\exp\left(6d\sigma_\theta^2\int_0^t e^{4m(s-t)}ds\right) \\ &\leq \left(\mathbb{E}[Y(0)] + \frac{bK_2}{m}\right)\exp\left(\frac{6d\sigma_\theta^2}{4m}(1 - e^{-4mt})\right) \\ &\leq \left(\mathbb{E}[Y(0)] + \frac{bK_2}{m}\right)\exp\left(\frac{3d\sigma_\theta^2}{2m}\right). \end{aligned} \tag{30}$$

Thus, $\sup_{t\geq 0}\mathbb{E}[|z(t)|^4] < \infty$.

C.3.2 Boundedness of 8th Moment

The proof for the 8th moment follows the exact same steps by setting $Y(t) := |z(t)|^8$.For $p = 8$, the natural decay factor is e^{8mt}, the drift contribution from b becomes $8b|z(t)|^6$.Using the boundedness of the 6th moment (which follows recursively from the 4th moment bound K_4), we obtain:

$$\mathbb{E}[|z(t)|^8] \leq \left(\mathbb{E}[|z(0)|^8] + \frac{bK_6}{m}\right)\exp\left(\frac{7d\sigma_\theta^2}{2m}\right), \tag{31}$$

which proves the uniform boundedness of the 8th moment.

C.4 Exponential Decay of the Fourth Moment of the Error

Error Dynamics Setup. Let $z(t)$ and $\tilde{z}(t)$ be the solutions of the LNSDE (Eq. 20) corresponding to different initial values $z(0)$ and $\tilde{z}(0)$, respectively. We consider a synchronous coupling, where $z(t)$ and $\tilde{z}(t)$ use the same model parameters and are driven by the same Brownian motion $W(t)$. Define the error process as $\epsilon(t) = z(t) - \tilde{z}(t)$ for all $t \geq 0$. Then, $\epsilon(t)$ satisfies the following Stochastic Differential Equation:

$$d\epsilon(t) = f_\Delta(t, \epsilon(t))dt + g_\Delta(t)\epsilon(t)dW(t), \tag{32}$$

where the drift and diffusion difference terms are defined as:

$$\begin{aligned} f_\Delta(t,\epsilon(t)) &:= f(t,z(t)) - f(t,\tilde{z}(t)), \\ g_\Delta(t)\epsilon(t) &:= \sigma_\theta(z(t)-\tilde{z}(t)) = \sigma_\theta\epsilon(t). \end{aligned} \tag{33}$$

Note that $f_\Delta(t,0)=0$ and $g_\Delta(t,0)=0$ for all t, implying that $\epsilon(t)\equiv 0$ is the trivial solution of Equation (32) when $\epsilon(0)=0$.

Based on Assumptions C.2 and C.2 (Lipschitz continuity and Linear Growth), the difference function f_Δ satisfies the following inequality for all $\epsilon\in\mathbb{R}^d$:

$$|f_\Delta(t,\epsilon)| \le L_f|\epsilon|, \tag{34}$$

where L_f is the Lipschitz constant of the drift function f. This linear growth condition on the error drift is crucial for the subsequent stability analysis.

Step 1: Let $Y(t)=\ln|\epsilon(t)|^4$. According to Ito's formula:

$$dY(t) = \left(\frac{\mathcal{G}V}{V} - \frac{|V_\epsilon^\top g|^2}{2V^2}\right)dt + \frac{V_\epsilon^\top g}{V}dW(t)$$

We calculate the two terms inside the parentheses in detail.

Part 1: The Generator Term $\frac{\mathcal{G}V}{V}$

The formula is $\mathcal{G}V = V_\epsilon^\top f + \frac{1}{2}\text{Tr}[gg^\top V_{\epsilon\epsilon}]$. Substitute $g(\epsilon)=\sigma_\theta\epsilon$.

Drift part:

$$V_\epsilon^\top f = (4|\epsilon|^2\epsilon)^\top f = 4|\epsilon|^2\langle\epsilon, f(t,\epsilon)\rangle$$

(We leave this part unchanged).

Diffusion part :We need to calculate $\frac{1}{2}\text{Tr}[(\sigma_\theta\epsilon)(\sigma_\theta\epsilon)^\top V_{\epsilon\epsilon}]$.

$$\begin{aligned} \text{Diff term} &= \frac{1}{2}\sigma_\theta^2\text{Tr}\left[\epsilon\epsilon^\top(4|\epsilon|^2 I + 8\epsilon\epsilon^\top)\right] \\ &= \frac{1}{2}\sigma_\theta^2\left(\text{Tr}[4|\epsilon|^2\epsilon\epsilon^\top] + \text{Tr}[8\epsilon\epsilon^\top\epsilon\epsilon^\top]\right) \end{aligned}$$

Using the property of the trace $\text{Tr}(AB)=\text{Tr}(BA)$ and $\epsilon^\top\epsilon=|\epsilon|^2$:

Term A: $\text{Tr}[4|\epsilon|^2\epsilon\epsilon^\top] = 4|\epsilon|^2\text{Tr}[\epsilon\epsilon^\top] = 4|\epsilon|^2\cdot|\epsilon|^2 = \mathbf{4}|\epsilon|^\mathbf{4}$

Term B: $\text{Tr}[8\epsilon(\epsilon^\top\epsilon)\epsilon^\top] = 8|\epsilon|^2\text{Tr}[\epsilon\epsilon^\top] = 8|\epsilon|^2\cdot|\epsilon|^2 = \mathbf{8}|\epsilon|^\mathbf{4}$

Combining them:

$$\text{Diff term} = \frac{1}{2}\sigma_\theta^2(4|\epsilon|^4 + 8|\epsilon|^4) = \frac{1}{2}\sigma_\theta^2(12|\epsilon|^4) = \mathbf{6}\sigma_\theta^\mathbf{2}|\epsilon|^\mathbf{4}$$

Calculating $\frac{\mathcal{G}V}{V}$:Dividing by $V=|\epsilon|^4$:

$$\frac{\mathcal{G}V}{V} = \frac{4|\epsilon|^2\langle\epsilon,f\rangle}{|\epsilon|^4} + \frac{6\sigma_\theta^2|\epsilon|^4}{|\epsilon|^4} = \frac{4\langle\epsilon,f\rangle}{|\epsilon|^2} + \mathbf{6}\sigma_\theta^\mathbf{2}$$

.

Part 2: The Ito Correction Term $-\frac{|V_\epsilon^\top g|^2}{2V^2}$.

This term comes from the second-derivative correction of the logarithmic transformation $\ln(\cdot)$ and is always negative.Calculating $V_\epsilon^\top g$:

$$V_\epsilon^\top g = (4|\epsilon|^2\epsilon)^\top(\sigma_\theta\epsilon) = 4\sigma_\theta|\epsilon|^2(\epsilon^\top\epsilon) = 4\sigma_\theta|\epsilon|^4$$

Squaring:

$$|V_\epsilon^\top g|^2 = (4\sigma_\theta|\epsilon|^4)^2 = 16\sigma_\theta^2|\epsilon|^8$$

Dividing by $2V^2$:The denominator is $2(|\epsilon|^4)^2 = 2|\epsilon|^8$.

$$\frac{16\sigma_\theta^2|\epsilon|^8}{2|\epsilon|^8} = \mathbf{8}\sigma_\theta^\mathbf{2}$$

Part3. Final Combination: Cancellation of Positive and Negative TermsNow we put both parts into the dt term:

$$\begin{aligned}\text{Drift of } \ln|\epsilon|^4 &= \left(\frac{\mathcal{G}V}{V}\right) - \left(\frac{|V_\epsilon^\top g|^2}{2V^2}\right) \\ &= \left(\frac{4\langle\epsilon, f\rangle}{|\epsilon|^2} + \mathbf{6}\boldsymbol{\sigma_\theta^2}\right) - \mathbf{8}\boldsymbol{\sigma_\theta^2} \\ &= \frac{4\langle\epsilon, f\rangle}{|\epsilon|^2} - \mathbf{2}\boldsymbol{\sigma_\theta^2}\end{aligned}$$

Part4. Conclusion and Physical MeaningThe final evolution equation is:

$$d(\ln|\epsilon(t)|^4) = \left(\frac{4\langle\epsilon(t), f(t,\epsilon(t))\rangle}{|\epsilon(t)|^2} - 2\sigma_\theta^2\right)dt + 4\sigma_\theta dW(t)$$

Integrate the above equation from time 0 to t:

$$\ln|\epsilon(t)|^4 - \ln|\epsilon(0)|^4 = \int_0^t \left(\frac{4\langle\epsilon(s), f(s,\epsilon(s))\rangle}{|\epsilon(s)|^2} - 2\sigma_\theta^2\right)ds + \int_0^t 4\sigma_\theta dW(s)$$

Since σ_θ is a constant, the stochastic integral term is $\int_0^t 4\sigma_\theta dW(s) = 4\sigma_\theta(W(t) - W(0)) = 4\sigma_\theta W(t)$.Rearranging yields:

$$\ln|\epsilon(t)|^4 = \ln|\epsilon(0)|^4 + \int_0^t \left(\frac{4\langle\epsilon(s), f\rangle}{|\epsilon(s)|^2} - 2\sigma_\theta^2\right)ds + 4\sigma_\theta W(t)$$

Step 2: Introducing the Instability Assumption for the Drift Term (Assumption & Bounding)In the previous discussion (corresponding to the unstable case in the paper), we assumed that the drift term f satisfies the linear growth condition (Lipschitz growth), meaning it attempts to make the system diverge:

$$\langle\epsilon, f(t,\epsilon)\rangle \le L_f|\epsilon|^2$$

Substitute this inequality into the integrand:

$$\frac{4\langle\epsilon(s), f\rangle}{|\epsilon(s)|^2} \le \frac{4L_f|\epsilon(s)|^2}{|\epsilon(s)|^2} = 4L_f$$

Therefore, the integral term is bounded by:

$$\int_0^t \left(\frac{4\langle\epsilon, f\rangle}{|\epsilon|^2} - 2\sigma_\theta^2\right)ds \le \int_0^t (4L_f - 2\sigma_\theta^2)ds = (4L_f - 2\sigma_\theta^2)t$$

The inequality now becomes:

$$\ln|\epsilon(t)|^4 \le \ln|\epsilon(0)|^4 + (4L_f - 2\sigma_\theta^2)t + 4\sigma_\theta W(t)$$

Step 3: Divide by t and Take the Limit (Division & Limit)To determine the exponential growth rate (Lyapunov exponent), we divide both sides of the inequality by t:

$$\frac{1}{t}\ln|\epsilon(t)|^4 \le \underbrace{\frac{\ln|\epsilon(0)|^4}{t}}_{\text{Initial Value Term}} + (4L_f - 2\sigma_\theta^2) + \underbrace{\frac{4\sigma_\theta W(t)}{t}}_{\text{Noise Average Term}}$$

Now let $t \to \infty$ and analyze term by term:Initial Value Term: Since $\ln|\epsilon(0)|^4$ is a constant (assuming the initial value is deterministic and non-zero), as $t \to \infty$:

$$\lim_{t\to\infty} \frac{\ln|\epsilon(0)|^4}{t} = 0$$

Noise Average Term (Key Point): According to the Strong Law of Large Numbers for Brownian Motion, the growth rate of Brownian motion $W(t)$ is much slower than linear time t (in fact, its growth order is $\sqrt{t\ln\ln t}$, the Law of the Iterated Logarithm). Therefore:

$$\lim_{t\to\infty} \frac{W(t)}{t} = 0 \quad \text{a.s. (almost surely)}$$

So:

$$\lim_{t\to\infty} \frac{4\sigma_\theta W(t)}{t} = 0 \quad \text{a.s.}$$

Step 4: Conclusion Discarding the terms that approach 0, the remaining constant term is the upper bound of the limit superior:

$$\limsup_{t\to\infty} \frac{1}{t} \ln |\epsilon(t)|^4 \leq 4L_f - 2\sigma_\theta^2 = -(2\sigma_\theta^2 - 4L_f) \quad \text{a.s.}$$

As we established in the previous almost sure stability analysis, we have the following inequality:

$$\limsup_{t\to\infty} \frac{1}{t} \ln |\epsilon(t)|^4 \leq -(2\sigma_\theta^2 - 4L_f) \quad a.s.$$

Let $\eta = 2\sigma_\theta^2 - 4L$. Assuming the noise is strong enough such that $\sigma_\theta^2 > 2L$, we have $\eta > 0$.Thus, by Egorov's theorem, for any $\delta > 0$, there exists $T^* > 0$ and a measurable set $E \in \mathcal{F}$ such that the convergence is uniform on E. Specifically, we have:

$$\frac{1}{T} \ln |\epsilon(T)|^4 \leq -\eta, \quad \omega \in E,$$

which implies $|\epsilon(T)|^4 \leq e^{-\eta T}$ for $\omega \in E$, with $P(E) \geq 1 - \delta$.Conversely, on the complement set E^c:

$$\frac{1}{T} \ln |\epsilon(T)|^4 > -\eta, \quad \omega \in E^c,$$

with $P(E^c) \leq \delta$ for all $T \geq T^*$.Therefore, we can split the expectation $\mathbb{E}[|\epsilon(T)|^4]$ into two parts:

$$\begin{aligned}
\mathbb{E}[|\epsilon(T)|^4] &= \mathbb{E}[|\epsilon(T)|^4 \mathbf{1}_E] + \mathbb{E}[|\epsilon(T)|^4 \mathbf{1}_{E^c}] \\
&\leq \mathbb{E}[e^{-\eta T} \mathbf{1}_E] + \mathbb{E}[|\epsilon(T)|^4 \mathbf{1}_{E^c}] \\
&\leq e^{-\eta T} P(E) + \mathbb{E}[|\epsilon(T)|^4 \mathbf{1}_{E^c}] \\
&\leq e^{-(2\sigma_\theta^2 - 4L)T} + \mathbb{E}[|\epsilon(T)|^4 \mathbf{1}_{E^c}]
\end{aligned}$$

For the second term, we apply the Cauchy-Schwarz inequality (or Hölder's inequality):

$$\mathbb{E}[|\epsilon(T)|^4 \mathbf{1}_{E^c}] \leq \sqrt{\mathbb{E}[(|\epsilon(T)|^4)^2]} \sqrt{\mathbb{E}[(\mathbf{1}_{E^c})^2]} = \sqrt{\mathbb{E}[|\epsilon(T)|^8]} \sqrt{P(E^c)}$$

Since we have proven that the 8th moment is bounded, i.e., $\sup_{t\geq 0} \mathbb{E}[|\epsilon(t)|^8] \leq C < \infty$, and knowing $P(E^c) \leq \delta$, we derive:

$$\mathbb{E}[|\epsilon(T)|^4 \mathbf{1}_{E^c}] \leq \sqrt{C}\sqrt{\delta}$$

Combining these results, we obtain :

$$\mathbb{E}[|\epsilon(T)|^4] \leq \exp\{-(2\sigma_\theta^2 - 4L_f)T\} + \sqrt{C}\sqrt{\delta}$$

Hence, for any small tolerance $\epsilon > 0$, we can choose a sufficiently small δ (and thus determine a sufficiently large T^*) such that $\sqrt{C}\sqrt{\delta} \leq \epsilon$. For $T \geq T^*$, we satisfy:

$$\mathbb{E}[|\epsilon(T)|^4] \leq \exp\{-(2\sigma_\theta^2 - 4L_f)T\} + \epsilon$$

Since the initial fourth moment is finite, for a fixed perturbation with $\rho > 0$, we can choose a sufficiently large constant $C_4 > 0$, independent of T, such that

$$\mathbb{E}\big[|\epsilon(0)|^4\big] \leq C_4 \rho^4.$$

Finally, since ϵ can be arbitrarily small relative to the exponential term for large T, this leads to the conclusion that for sufficiently large T:

$$\mathbb{E}[|\epsilon(T)|^4] \leq (1 + C_4\rho^4) \exp\{-(2\sigma_\theta^2 - 4L_f)T\} \leq C_4(1+\rho)^4 \exp\{-(2\sigma_\theta^2 - 4L_f)T\} \tag{35}$$

C.5 Error Bound for CACT-ATTN Mechanism

Theorem C.9. *(Error Bound for CACT-ATTN) Let CACT-ATTN$^n(t)$ denote the output of the attention mechanism at time t for the n-th dimension. The original and perturbed trajectories share the same temporal anchors $\{\bar{t}_p^n\}_{p=1}^P$ and compactness scores $\{\kappa_{p,n}\}_{p=1}^P$. Under the assumption of bounded higher moments and Lipschitz continuity of the activation functions, the expected error is bounded by:*

$$\mathbb{E}\left[\left\|\textit{CACT-ATTN}^n(t) - \widetilde{\textit{CACT-ATTN}}^n(t)\right\|\right] \leq \frac{C_A(1+\rho)}{\lambda(t_i - t_{max})}, \tag{36}$$

where t_{max} denote the largest timestamp among the input anchors, C_A is constants depending on the spectral norms of the weight matrices and the bounds of the moments.

Proof. Let the output at query time t_i be defined as CACT-ATTN$^n(t_i) = \sum_{j=1}^P \mathrm{softmax}\Big(\frac{\alpha_j^n(t_i)}{\sqrt{D}} + \log\kappa_{j,n}\Big)\hat{\mathbf{v}}_j^n(t_i) = \sum_{j=1}^P A_{ij}^n V_j^n$(Eq. 5), where α_{ij}^n are the attention weights and V_j^n are the value vectors derived from the trajectory starting at t_j.For computational convenience, we let:$A_{ij}^n = \mathrm{softmax}\Big(\frac{\alpha_j^n(t_i)}{\sqrt{D}} + \log\kappa_{j,n}\Big)$,$V_j^n = \hat{\mathbf{v}}_j^n(t_i)$. We aim to bound the total expected error:

$$\mathcal{E}_{\text{total}}^n = \mathbb{E}\left[\left|\sum_j (A_{ij}^n V_j^n - \tilde{A}_{ij}^n \tilde{V}_j^n)\right|\right].$$

Step 1: Decomposition via Triangle Inequality. Using the property $|a+b| \leq |a| + |b|$ and separating the error into weight-induced and state-induced components:

$$\begin{aligned}
\mathcal{E}_{\text{total}}^n &\leq \sum_j \mathbb{E}\left[\left|A_{ij}^n(V_j^n - \tilde{V}_j^n) + (A_{ij}^n - \tilde{A}_{ij}^n)\tilde{V}_j^n\right|\right] \\
&\leq \sum_j \left(\underbrace{\mathbb{E}\left[|A_{ij}^n| \cdot |V_j^n - \tilde{V}_j^n|\right]}_{\text{Term A: Value Error}} + \underbrace{\mathbb{E}\left[|(A_{ij}^n - \tilde{A}_{ij}^n)\tilde{V}_j^n|\right]}_{\text{Term B: Attention Weight Error}}\right).
\end{aligned}$$

Step 2: Bounding Term A (State Decay). Since attention weights represent a probability distribution, $|A_{ij}^n| < 1$. Noting that $V_j^n = \hat{\mathbf{v}}_j^n(t_i) = \frac{1}{t_i - t_j}\int_{t_j}^{t_i} \mathbf{v}_j^n(\tau)\,d\tau$, $\lambda := \frac{|\sigma_\theta|^2 - 2L_f}{2}$ and employing the Lipschitz continuity of the linear transformation ($L_V = \|W_V\|$) and the established exponential decay of $y(t)$:

$$\begin{aligned}
\text{Term A} &\leq \mathbb{E}\left[|V_j^n - \tilde{V}_j^n|\right] \\
&\leq L_V \cdot \mathbb{E}\left[\left|\frac{1}{t_i - t_j}\int_{t_j}^{t_i} (\mathbf{v}_j^n(\tau) - \tilde{\mathbf{v}}_j^n(\tau))\,d\tau\right|\right] \\
&\leq L_V \cdot \mathbb{E}\left[\left|\frac{1}{t_i - t_j}\int_{t_j}^{t_i} ((1+\rho)e^{-\lambda(\tau - t_j)})\,d\tau\right|\right] \\
&\leq L_V(1+\rho)\frac{1 - e^{-\lambda(t_i - t_j)}}{\lambda(t_i - t_j)} \\
&\leq \frac{L_V(1+\rho)}{\lambda(t_i - t_j)}, \lambda > 0 \quad \text{(by Lemma C.6)}
\end{aligned} \tag{37}$$

Step 3: Bounding Term B (Weight Error via Cauchy-Schwarz). To decouple the weight error from the state magnitude, we apply the Cauchy-Schwarz inequality ($\mathbb{E}[|XY|] \le \sqrt{\mathbb{E}[X^2]}\sqrt{\mathbb{E}[Y^2]}$):

$$\begin{aligned}\text{Term B} &\le \sqrt{\mathbb{E}\left[(A_{ij}^n - \tilde{A}_{ij}^n)^2\right]} \cdot \sqrt{\mathbb{E}\left[|\tilde{V}_j^n|^2\right]} \\ &\le \underbrace{\sqrt{\mathbb{E}\left[(\Delta A_{ij}^n)^2\right]}}_{B_1:\text{Weight RMSE}} \cdot \underbrace{\|W_V^n\|\sqrt{M}}_{B_2:\text{Bounded Moment}},\end{aligned}$$

where B_2 is bounded by a constant M (from Eq. 23 regarding 2nd moment boundedness).

Step 3.1: Analysis of Weight RMSE (B_1). Since $A_{ij}^n = \mathrm{softmax}\Big(\frac{\alpha_j^n(t_i)}{\sqrt{D}} + \log \kappa_{j,n}\Big) = \mathrm{softmax}\big(S_{ij}^n + \log \kappa_{j,n}\big)$(For notational convenience, we let: $S_{ij}^n = \frac{\alpha_j^n(t_i)}{\sqrt{D}}, S_{ij}^n = \frac{\alpha_j^n(t_i)}{\sqrt{D}}, \mathbf{S}_i^n = (S_{i1}^n, \ldots, S_{iP}^n)^\top, \boldsymbol{\kappa}_n = (\kappa_{1,n}, \ldots, \kappa_{P,n})^\top$), and Softmax is Lipschitz continuous with constant L_{soft}, we shift the bound to the score S_{ij}^n:

$$\sqrt{\mathbb{E}\left[(\Delta A_{ij}^n)^2\right]} \le L_{\text{soft}} \sqrt{\mathbb{E}\big[\|\Delta \mathbf{S}_i^n\|_2^2\big]} = L_{\text{soft}} \left(\sum_{\ell=1}^{P} \mathbb{E}\big[|\Delta S_{i\ell}^n|^2\big]\right)^{1/2}.$$

Recall the integral formulation of the score $S_{ij}^n = \frac{\alpha_j^n(t_i)}{\sqrt{D}} = \frac{1}{\sqrt{D}(t_i - t_j)} \int_{t_j}^{t_i} q_j^n(\tau)(k_j^n(\tau))^\top d\tau$. By applying Minkowski's inequality to the integral:

$$\begin{aligned}\sqrt{\mathbb{E}[|\Delta S_{ij}^n|^2]} &\le \frac{1}{\sqrt{D}(t_i - t_j)} \int_{t_j}^{t_i} \sqrt{\mathbb{E}\left[\left|q_j^n (k_j^n)^\top - \tilde{q_j^n}(\tilde{k_j^n})^\top\right|^2\right]} d\tau \\ &= \frac{1}{\sqrt{D}(t_i - t_j)} \int_{t_j}^{t_i} \sqrt{\mathbb{E}\left[\left|q_j^n (\Delta k_j^n)^\top + (\Delta q_j^n)(\tilde{k_j^n})^\top\right|^2\right]} d\tau.\end{aligned}$$

Using the inequality $\sqrt{\mathbb{E}[(X+Y)^2]} \le \sqrt{\mathbb{E}[X^2]} + \sqrt{\mathbb{E}[Y^2]}$ and then applying Cauchy-Schwarz again to separate Q and ΔK:

$$\begin{aligned}\sqrt{\mathbb{E}[|\Delta S|^2]} &\le \frac{1}{\sqrt{D}(t_i - t_j)} \int_{t_j}^{t_i} \left(\sqrt{\mathbb{E}[|q_j^n|^2 |\Delta k_j^n|^2]} + \sqrt{\mathbb{E}[|\Delta q_j^n|^2 |\tilde{k_j^n}|^2]}\right) d\tau \\ &\le \frac{1}{\sqrt{D}(t_i - t_j)} \int_{t_j}^{t_i} \left(\underbrace{\sqrt[4]{\mathbb{E}[|q_j^n|^4]}}_{\text{Bounded}} \cdot \underbrace{\sqrt[4]{\mathbb{E}[|\Delta k_j^n|^4]}}_{\text{Decays Exp.}} + \underbrace{\sqrt[4]{\mathbb{E}[|\Delta q_j^n|^4]}}_{\text{Decays Exp.}} \cdot \underbrace{\sqrt[4]{\mathbb{E}[|\tilde{k_j^n}|^4]}}_{\text{Bounded}}\right) d\tau.\end{aligned}$$

Since the 4th moments of Q^n are bounded (Eq. 30) and the 4th moments of error ΔK^n decays exponentially (Eq. 35), the integral term is dominated by the decay rate, completing the proof.

Step 3.2: Explicit Bound for Score Error (B_1). By Eq. 30, the 4th moment of the query state is uniformly bounded, i.e., $\sqrt[4]{\mathbb{E}[|q_j^n|^4]} \le C_Q$, $\sqrt[4]{\mathbb{E}[|\tilde{k_j^n}|^4]} \le C_K, \lambda := \frac{|\sigma_\theta|^2 - 2L_f}{2}$.

Since the query and key trajectories are obtained by linear projections of the latent trajectory, their perturbations inherit the same exponential decay up to the projection constants L_Q and L_K. Substituting the result from Eq. 35(Exponential Decay of Error), we have: $\sqrt[4]{\mathbb{E}[|\Delta k_j^n|^4]} \le C_4^{\frac{1}{4}} L_K (1+\rho) \exp\{-\frac{1}{4}(2\sigma_\theta^2 - 4L_f)\tau\}$, $\sqrt[4]{\mathbb{E}[|\Delta q_j^n|^4]} \le C_4^{\frac{1}{4}} L_Q (1+\rho) \exp\{-\frac{1}{4}(2\sigma_\theta^2 - 4L_f)\tau\}$.

Substituting these into the integral from Step 3.1:

$$\begin{aligned}\sqrt{\mathbb{E}[(\Delta S_{ij}^n)^2]} &\le \frac{1}{\sqrt{D}(t_i-t_j)}\int_{t_j}^{t_i}(C_Q L_K + C_K L_Q)C_4^{\frac{1}{4}}\cdot(1+\rho)e^{-\frac{2\sigma_\theta^2-4L_f}{4}(\tau-t_j)}d\tau \\ &\le \frac{(C_Q L_K + C_K L_Q)C_4^{\frac{1}{4}}\cdot(1+\rho)}{\sqrt{D}(t_i-t_j)}\int_{t_j}^{t_i} e^{-\frac{\sigma_\theta^2-2L_f}{2}(\tau-t_j)}d\tau \\ &\le C_{\text{score}}(1+\rho)\frac{1-e^{-\lambda(t_i-t_j)}}{\lambda(t_i-t_j)} \\ &\le \frac{C_{\text{score}}(1+\rho)}{\lambda(t_i-t_j)}, \lambda>0\end{aligned} \tag{38}$$

where C_{score} absorbs the constants $C_Q, C_4^{\frac{1}{4}}, C_{\text{int}}$ and the dimension factor. Combining the score bounds with the softmax estimate gives:

$$\sqrt{\mathbb{E}\big[(\Delta A_{ij}^n)^2\big]} \le L_{\text{soft}}\sqrt{P}\max_{1\le\ell\le P}\sqrt{\mathbb{E}\big[|\Delta S_{i\ell}^n|^2\big]} \le \frac{L_{\text{soft}}\sqrt{P}\,C_{\text{score}}(1+\rho)}{\lambda(t_i-t_{\max})}, \tag{39}$$

where $t_{\max}$ denote the largest timestamp among the input anchors. Consequently, the entire Term B is bounded by:

$$\text{Term B} \le L_{\text{soft}}\cdot\frac{\sqrt{P}C_{\text{score}}(1+\rho)}{\lambda(t_i-t_{\max})}\cdot\|W_V\|\sqrt{M} := \frac{C'_B(1+\rho)}{\lambda(t_i-t_{\max})}. \tag{40}$$

Step 4: Aggregation and Final Bound. We now substitute the bounds for Term A (from Step 2) and Term B (from Step 3.2) back into the total error decomposition. Let N_h be the number of history steps (summation over j).

$$\begin{aligned}\mathcal{E}_{\text{total}}^n &\le \sum_j\left(\underbrace{\mathbb{E}[|\alpha_{ij}^n|\cdot|V_j^n-\tilde{V}_j^n|]}_{\text{Term A}}+\underbrace{\mathbb{E}[|(\alpha_{ij}^n-\tilde{\alpha}_{ij}^n)\tilde{V}_j^n|]}_{\text{Term B}}\right) \\ &\le \sum_j\left(\frac{L_V(1+\rho)}{\lambda(t_i-t_j)}+\frac{C'_B(1+\rho)}{\lambda(t_i-t_{\max})}\right) \\ &\le \frac{C_A(1+\rho)}{\lambda(t_i-t_{\max})},\end{aligned} \tag{41}$$

where we simplify the summation by bounding it with effective constants C_A . The decay rate λ is determined by the bottleneck of the error decay in the SDE, satisfying:

$$\lambda \propto \frac{1}{2}(|\sigma_\theta|^2-2L_f). \tag{42}$$

This shows that error decays algebraically with the distance between the query time and the latest input anchor. □

[Bound of continuous-time attention with ODEs] When replacing the LNSDE backbone with a deterministic ODE, i.e., removing the diffusion term by setting $\sigma_\theta = 0$, so that $\lambda < 0$, the same derivation from Eq. 37 and Eq. 38 gives

$$\begin{aligned}&\mathbb{E}\left[\left\|\text{CACT-ATTN}^n(t)-\widetilde{\text{CACT-ATTN}}^n(t)\right\|\right] \\ &\le \sum_j\left(L_V(1+\rho)\frac{1-e^{-\lambda(t_i-t_j)}}{\lambda(t_i-t_j)}+C'_B(1+\rho)\frac{1-e^{-\lambda(t_i-t_{\max})}}{\lambda(t_i-t_{\max})}\right) \\ &\le C_A(1+\rho)\frac{1-e^{-\lambda(t_i-t_{min})}}{\lambda(t_i-t_{max})}.\end{aligned} \tag{43}$$

where t_{max} denote the largest timestamp among the input anchors,t_{min} denote the smallest timestamp among the input anchors,ρ denotes the magnitude of the initial latent perturbation and C_A is a constant satisfying $C_A \ge L_V + C'_B$.

For deterministic ODE dynamics, the diffusion-induced stabilization is removed, so the corresponding propagation rate is no longer positive. As a result, Eq. 43 does not provide a vanishing robustness bound as the query time moves away from the latest input anchor. This indicates that deterministic continuous-time attention may still propagate initial perturbations, whereas the LNSDE-based RosFormer admits the vanishing bound in Theorem 4.1.