

# Recurrent Neural Goodness-of-Fit Test for Time Series

Aoran Zhang

Carnegie Mellon University

Wenbin Zhou

Carnegie Mellon University

Liyan Xie

University of Minnesota

Shixiang Zhu

Carnegie Mellon University

## Abstract

Time series data are crucial across diverse domains such as finance and healthcare, where accurate forecasting and decision-making rely on advanced modeling techniques. While generative models have shown great promise in capturing the intricate dynamics inherent in time series, evaluating their performance remains a major challenge. Traditional evaluation metrics fall short due to the temporal dependencies and potential high dimensionality of the features. In this paper, we propose the REcurrent NeurAL (RENAL) Goodness-of-Fit test, a novel and statistically rigorous framework for evaluating generative time series models. By leveraging recurrent neural networks, we transform the time series into conditionally independent data pairs, enabling the application of a chi-square-based goodness-of-fit test to the temporal dependencies within the data. This approach offers a robust, theoretically grounded solution for assessing the quality of generative models, particularly in settings with limited time sequences. We demonstrate the efficacy of our method across both synthetic and real-world datasets, outperforming existing methods in terms of reliability and accuracy. Our method fills a critical gap in the evaluation of time series generative models, offering a tool that is both practical and adaptable to high-stakes applications.

## 1 INTRODUCTION

Time series data are pervasive across various domains, playing a critical role in applications requiring tem-

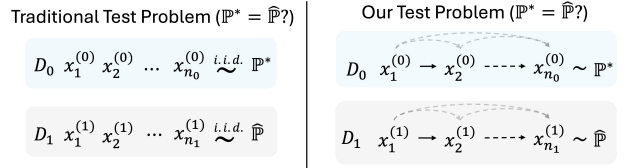


Figure 1: An illustration of our problem setup. In traditional test problems, both the real data ( $D_0$ ) and the model-generated data ( $D_1$ ) are assumed to be *i.i.d.*, following the underlying distributions  $\mathbb{P}^*$  and  $\hat{\mathbb{P}}$ , respectively. However, in our test problem, both  $D_0$  and  $D_1$  exhibit general temporal dependencies.

poral insights, such as forecasting stock prices, optimizing trading strategies in finance, healthcare monitoring, and managing power generation and demand in energy systems (Han et al., 2019; Zhu et al., 2021b; Wu et al., 2024; Zhang et al., 2024). As the complexity and volume of time series data continue to rise, the demand for sophisticated modeling techniques grows in parallel. In this context, generative time series models have emerged as powerful tools, enabling probabilistic time series prediction (Li et al., 2022; Liu et al., 2023; Albergo and Vanden-Eijnden, 2023), the simulation of future scenarios (Dong et al., 2023; Sattarov et al., 2023), and the generation of synthetic data that captures complex temporal dependencies (Yoon et al., 2019; Zhu et al., 2021a, 2022; Suh et al., 2023, 2024).

However, despite their growing popularity, evaluating the performance of generative time series models remains a significant challenge. In image and text domains, evaluation often relies on heuristic metrics, such as Fréchet Inception Distance (FID) for images (Heusel et al., 2017) and BLEU score for text (Papineni et al., 2002), or even subjective visual inspection. These methods, however, are not readily applicable for time series data due to the intrinsic temporal dependencies between observations. As a result, the absence of established and rigorous evaluation methods for generative time series models poses a significant barrier to their further adoption, particularly in high-stakes domains where reliability is paramount. Current metrics either fail to account for the dynamic, evolving nature

of time series or are built on assumptions that often do not hold in real-world settings. This underscores the need for robust, statistically sound evaluation techniques tailored to time series models.

Goodness-of-fit (GOF) tests have long been a cornerstone for evaluating the accuracy of statistical models (D’Agostino, 2017). However, applying these tests to modern time series data presents significant challenges: (i) *Complex temporal dependencies*: The inherent temporal structure of time series data often violates the strict stationarity assumption that underpins many traditional tests (Chen et al., 2003). In practice, we often only have access to a small number of sequences – or even a single time series – which does not provide enough samples to observe stationarity. A naive attempt to address this issue is to partition the time series into smaller segments, but this creates dependent subsequences, which restricts the effectiveness of standard tests that require a larger set of independent samples Baum et al. (2023). (ii) *High dimensionality*: Contemporary time series datasets often exhibit high dimensionality, complicating both estimation and hypothesis testing, which can result in diminished statistical power. These challenges underscore the necessity for new approaches that account for the inherent temporal structure and can operate effectively with a limited number of time sequences while providing statistically sound evaluations of model performance.

In this paper, we present a novel REcurrent Neural (RENAL) Goodness-of-Fit test tailored for general time series data, including both regularly and irregularly sampled time series. Our approach overcomes the challenges posed by temporal dependencies by transforming time series into conditionally independent data pairs using recurrent neural architectures such as Long Short-Term Memory (LSTM) and Transformers (Graves and Graves, 2012; Vaswani, 2017). We begin by leveraging recurrent networks to encode the time series into low-dimensional history embeddings, preserving temporal structure while mapping the original distribution into a low-dimensional Markov Chain. This enables us to decompose dependent observations into independent embedding pairs. We then propose a novel chi-square test to compare the empirical transition probabilities of history embeddings from real data and the generative model of interest. This method provides a theoretically grounded and practical tool for evaluating complex, potentially high-dimensional time series models. We demonstrate its strong empirical performance across various types of time series data, including both synthetic and real-world datasets, spanning traditional time series and discrete event sequences.

Our key contributions are summarized as follows.

1. We introduce the first goodness-of-fit test for general time series data, by transforming the problem into testing differences in transition probabilities of learned history embeddings. The resulting test offers an effective tool for evaluating the quality of generative time series models.
2. We present the asymptotic distribution of test statistics under the null hypothesis, providing guarantees on the accuracies of the proposed test.
3. We empirically validate our method, demonstrating its superior performance over state-of-the-art methods across both synthetic and real-world datasets, including both regularly and irregularly sampled time series.

**Related Work** The evaluation of time-series generative models remains challenging due to the lack of a comprehensive and widely accepted framework. Some measures have been proposed, including specialized metrics such as Fréchet Inception Distance (FID) for images (Heusel et al., 2017) and BLEU score for text (Papineni et al., 2002), and more general distributional measures like Kullback-Leibler (KL) divergence (Moreno et al., 2003; Chan and Vasconcelos, 2005), Wasserstein distance (Xiao et al., 2017), and Maximum Mean Discrepancy (MMD) (Li et al., 2018) have been adopted. However, these measures are not inherently designed for goodness-of-fit (GOF) of time series data (D’Agostino, 2017).

Most GOF tests for time series are parametric. One common approach is to conduct residual-based tests that measure the discrepancy between residual distributions or assess residuals’ autocorrelation (Ljung and Box, 1978; Box and Pierce, 1970; Escanciano, 2006; Gallagher and Fisher, 2015). Another approach is to conduct likelihood ratio-based tests to compare nested models (Chernoff, 1954; Akaike, 1974; Burnham and Anderson, 2004; Chen et al., 2003). Stein-based methods assess model fit via Stein discrepancy, with applications in point processes (Chwialkowski et al., 2016; Yang et al., 2019). These parametric methods typically impose strict assumptions about data distributions and are unsuitable for black-box generative models with unknown structures.

In contrast, non-parametric GOF tests offer more flexibility without strict distributional assumptions. A popular approach measures the distance between kernel embeddings of distributions in a reproducing kernel Hilbert space using MMD (Gretton et al., 2012, 2009; Jitkrittum et al., 2016; Chwialkowski et al., 2015; Schrab et al., 2023) or Fisher discriminant analysis (Eric et al., 2007). The kernel can be enhanced with deep neural networks capture more complex structures (Liu et al., 2020; Kirchler et al., 2020; Zhu et al., 2023).

Other methods use scores like Hyvärinen scores (Wu et al., 2022) or the Generalized Score (Wei et al., 2021) to measure the discrepancies. However, most of these non-parametric tests overlook temporal dependencies and assume the data is *i.i.d.* or strictly stationary. While a non-parametric test based on Kernel Stein Discrepancy (KSD) exists for sequential models, it focuses on dependencies for neighboring events only and is designed for discrete-time series (Baum et al., 2023). We aim to design a GOF test to tackle continuous-time time series with long-term dependencies.

Our work offers several key advantages over prior approaches. Due to its parametric specification, it is highly scalable compared to non-parametric methods when applied to large-scale data. Meanwhile, it can achieve high performance in complex and high-dimensional settings (*e.g.*, generative models) compared to other traditional parametric methods, which can be credited to the strong modeling power of the recurrent neural networks that it incorporates.

## 2 RECURRENT NEURAL GOODNESS-OF-FIT

In this section, we present a flexible framework for testing general time series data, termed REcurrent NEural (RENAL) Goodness-of-Fit. The overall architecture is summarized in Figure 2.

### 2.1 Problem Setup

Consider two stationary time series datasets:  $D_0 = \{x_1^{(0)}, \dots, x_{n_0}^{(0)}\}$ , obtained from observations of a real-world process, and  $D_1 = \{x_1^{(1)}, \dots, x_{n_1}^{(1)}\}$ , synthetic data sequences generated by a learned time series model. Here,  $n_0$  and  $n_1$  are the lengths of the respective sequences, and each observation  $x_i^{(k)}$  belongs to the sample space  $\mathcal{X} \subseteq \mathbb{R}^d$ , where  $k \in \{0, 1\}$  indicates the dataset index and  $d$  is the data dimension. We assume that the data sequences are generated sequentially with strict temporal ordering. This means each observation  $x_i^{(k)}$  depends solely on the preceding observations within its own sequence:  $x_1^{(k)}, \dots, x_{i-1}^{(k)}$ .

Let  $\mathbb{P}^*$  denote the true (but unknown) probability distribution governing the real-world time series  $D_0$ , and let  $\hat{\mathbb{P}}$  represent the probability distribution learned by the time series model, from which  $D_1$  is sampled. Our objective is to assess whether the learned time series model  $\hat{\mathbb{P}}$  accurately captures the underlying distribution of the real-world data. Formally, we aim to test the following hypotheses:

$$H_0 : \mathbb{P}^* = \hat{\mathbb{P}} \quad \text{versus} \quad H_1 : \mathbb{P}^* \neq \hat{\mathbb{P}}. \quad (1)$$

This constitutes a *goodness-of-fit* (GOF) test for a time series model. In other words, we aim to test whether the real sequences  $D_0$  are likely to be sampled from the tentative model  $\hat{\mathbb{P}}$ . Note that  $\hat{\mathbb{P}}$  is also unknown except the sequences  $D_1$  generated from it. To assess the effectiveness of a GOF test, we use the Type-I and Type-II accuracy as the evaluation metric. We also choose the average of Type-I and Type-II accuracy as an overall metric<sup>1</sup>.

Two key challenges arise in this task: (i) Each observation depends on all previous observations in its sequence, meaning the data are not independent. When we have a very limited number of sequences – possibly even just one sequence for each distribution ( $D_0$  and  $D_1$ ) – the lack of independence restricts the applicability of standard statistical tests. (ii) The sample space  $\mathcal{X}$  may be high-dimensional, which complicates the estimation and comparison of the distributions  $\mathbb{P}^*$  and  $\hat{\mathbb{P}}$ . The curse of dimensionality implies that the number of required samples increases exponentially with the dimensionality  $d$ , making distributional comparisons even more challenging.

### 2.2 Recurrent Neural Representation

To address the challenges posed by temporal dependencies and high dimensionality in time series data, our main idea is to transform the original interdependent data sequences into a set of low-dimensional and conditionally independent data pairs. Rather than directly testing the raw time series data – which is complicated by sequential dependencies – we leverage recurrent neural architectures to create history embeddings that encapsulate temporal information.

We consider the time series modeled by Neural Ordinary Differential Equations (ODEs), a flexible and powerful framework capable of capturing continuous-time dynamics, making it well-suited for both regularly and irregularly sampled time series data (Chen et al., 2018). Specifically, for a continuous-time series  $\{x(t), t \geq 0\}$ , we define a low-dimensional *history embedding*  $h(t) \in \mathcal{H} \subseteq \mathbb{R}^p$ , where  $p \leq d$  and  $\mathcal{H}$  denotes the embedding space. This latent embedding captures the key information about the historical observations up to time  $t$ . The evolution of the embedding is governed by the differential equation:

$$\frac{dh(t)}{dt} = f(h(t), x(t)), \quad (2)$$

where  $f$  is the update function modeling the underlying dynamics over continuous time. Here,  $h(t)$  serves

<sup>1</sup>We may also use the weighted average of Type-I and Type-II accuracies, depending on which accuracy is prioritized.

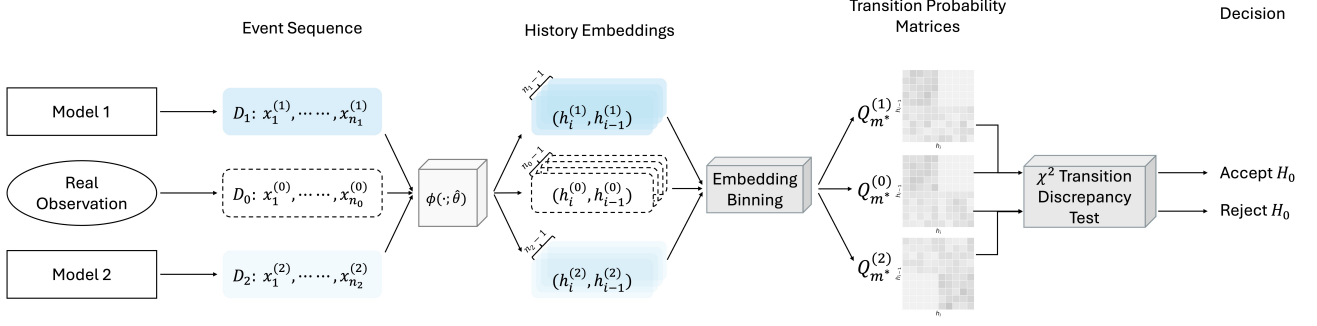


Figure 2: Architecture of the proposed framework. Real-world observations are compared to model-generated sequences, with darker blue indicating better fits. We first use a recurrent neural network  $\phi$  to extract conditionally independent history embeddings. Then we construct their transition probability matrices using these embeddings and evaluate the fit with a chi-square discrepancy test.

as a summarization of the time series  $x(t)$  up to time  $t$ , effectively capturing temporal dependencies in a condensed form. Given the embedding, the observation  $x(t)$  is a random variable with mean value  $g(h(t))$ , where  $g : \mathcal{H} \mapsto \mathcal{X}$  is a decoding function.

In practice, we work with  $n$  discrete observations  $\{t_i\}_{i=1}^n$  and adopt the Euler method to approximate the solution to the ODE model (2) (Chen et al., 2018):

$$h_{i+1} = h_i + f(h_i, x_i) \Delta t_i, \quad (3)$$

where  $h_i$  and  $h_{i+1}$  are the history embeddings at times  $t_i$  and  $t_{i+1}$ , respectively, and  $\Delta t_i = t_{i+1} - t_i$  is time step between them.

Inspired by the model (3), we parameterize the history embedding updates as

$$h_{i+1} = \phi(x_i, h_i; \theta), \quad (4)$$

using an embedding function  $\phi(\cdot, \cdot; \theta) : \mathcal{X} \times \mathcal{H} \mapsto \mathcal{H}$ , which is modeled as a neural network with  $\theta$  representing the network’s weights. The embedding function updates the history embedding  $h_{i+1}$  based on the current observation  $x_i$  and the previous history  $h_i$ .

*Learning.* The embedding network  $\phi(\cdot, \cdot; \theta)$ , together with the decoding network  $g(\cdot; \varphi)$  can be learned by minimizing the one-step predictive loss. Specifically, we let  $h_{i+1} = \phi(x_i, h_i; \theta)$  as in (4) and  $\hat{x}_{i+1} = g(h_{i+1}; \varphi)$ , and the model parameters  $\theta$  and  $\varphi$  are jointly learned by minimizing a predictive loss, e.g., the mean square error  $\sum_i \|\hat{x}_i - x_i\|^2$ . We denote the learned network as  $\phi(\cdot, \cdot; \hat{\theta})$  and  $g(\cdot; \hat{\varphi})$ .

### 2.3 Reformulation of GOF

To facilitate the application of the GOF test in (1), we impose the following assumption on the learned embedding function.

**Assumption 1** (Consistency). *The learned embedding function  $\phi(\cdot, \cdot; \hat{\theta})$  is consistent, meaning it approximates the true underlying embedding function in (3) as closely as necessary. Therefore, the learned embedding  $h_{i+1}$  captures all relevant information from  $x_i$  and  $h_i$ .*

This assumption ensures that the history embeddings produced by our model are sufficient representations of the observed history, effectively summarizing all relevant past information required for accurate time series modeling.

Prior research has demonstrated the empirical validity of this assumption across various recurrent neural architectures, including RNNs, LSTMs, and Transformers (Mei and Eisner, 2017; Chen et al., 2018; Zuo et al., 2020; Dong et al., 2023). Furthermore, even in the case when this assumption does not hold, our numerical experiments indicate that our method remains robust in performance. This resilience stems from the binary nature of the GOF test, where a clear distinction between null and alternative sequences is typically sufficient to trigger an alert.

Under Assumption 1, we show in the following Lemma that the sequence of history embeddings  $\{h_i\}_{i=1}^n$  possesses the *Markov* property and is *homogeneous* over time.

**Lemma 1.** *The history embedding sequence  $\{h_i\}_{i=1}^n$  is a homogeneous Markov chain, i.e., for any set  $B \subset \mathcal{H}$ :*

$$\mathbb{P}\{h_i \in B | h_{i-1}, h_{i-2}, \dots, h_1\} = \mathbb{P}\{h_i \in B | h_{i-1}\},$$

and

$$\mathbb{P}\{h_{i+1} \in B | h_i\} = \mathbb{P}\{h_i \in B | h_{i-1}\}.$$

The proof is provided in Appendix A.1.

Therefore, the distributional behavior of the learned history embeddings can be fully characterized by their

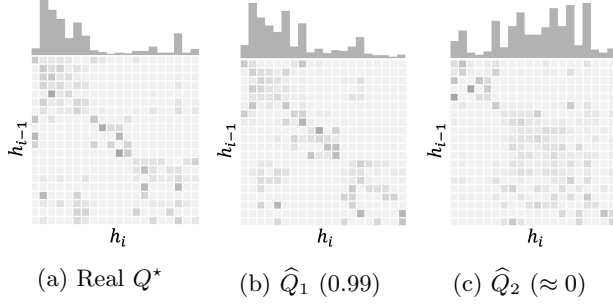


Figure 3: Transition probability matrices of history embeddings  $Q$  from (a) real data, (b) data generated by Model 1, and (c) data generated by Model 2. Model 1 exhibits a better fit compared to Model 2, as evidenced by the closer resemblance between the histograms in (a) and (b). The number in the parentheses indicates the corresponding testing score.

one-step *transition* density function  $Q : \mathcal{H} \times \mathcal{H} \mapsto \mathbb{R}_{\geq 0}$ . Specifically, for any subset  $B \subseteq \mathcal{H}$  and for all  $i$ , we have

$$\mathbb{P}\{h_{i+1} \in B | h_i = h\} = \int_{h' \in B} Q(h, h') dh',$$

where  $Q(h, \cdot)$  serves as the conditional probability density function of  $h_{i+1}$  given  $h_i = h$ .

This formulation allows us to pivot our original GOF test in (1) to test the equality of the one-step transition probabilities for history embeddings. By leveraging the Markov property, we simplify the problem as shown in the following proposition.

**Proposition 1.** *The original GOF test in (1) can be reformulated as:*

$$H_0 : Q^* = \hat{Q} \quad \text{versus} \quad H_1 : Q^* \neq \hat{Q}, \quad (5)$$

where  $Q^*$  denotes the true transition density function of the history embeddings derived from the real data, and  $\hat{Q}$  denotes the transition density function derived from the model-generated data.

Figure 3 provides an illustrative comparison of transition probability matrices estimated from three different datasets. Panel (a) displays the matrix derived from the real data used for training, which is a sequence generated by a self-exciting point process (SE). Panels (b) and (c) show the transition probability matrices estimated from data generated by two fitted models, denoted as Model 1 and Model 2, respectively. We set Model 1 to be the same as SE, while Model 2 represents a self-correcting point process (SC), which differs significantly from SE. Observing these matrices, we can see that Model 1 in panel (b) offers a better fit compared to Model 2 in panel (c), as the pattern of its matrix more closely resembles that of the real data shown in panel (a).

## 2.4 Proposed Algorithm

To operationalize the reformulated GOF test in (5), we proceed by partitioning the embedding space and calculating both transition probability matrices  $Q^*$  for the real data and  $\hat{Q}$  for the model-generated data. We then perform a chi-square test using these matrices to assess the fit of the model. The complete testing procedure is outlined in Algorithm 1.

We begin by learning the embedding function  $\phi$  using the real-world observations  $D_0$ . After training, we apply the learned embedding function to both datasets  $D_0$  and  $D_1$  and obtain their embedding sequences  $\{h_i^{(0)}\}$  and  $\{h_i^{(1)}\}$ .

**Step 1: Embedding Binning** We partition the embedding space  $\mathcal{H}$  into  $m$  bins of equal size, denoted as  $\mathcal{H}_1, \dots, \mathcal{H}_m$ . Such discretization allows us to approximate the continuous transition density function  $Q$  with a discrete transition matrix. For a given embedding sequence  $\{h_i\}_{i=1}^n$ , we assign each embedding to a bin, resulting in a sequence of bin indices  $\{y_i\}_{i=1}^n$ , where  $y_i = \sum_{u=1}^m u \cdot \mathbb{1}\{h_i \in \mathcal{H}_u\}$ , and  $\mathbb{1}$  denotes the indicator function, which equals 1 if the condition is true and 0 otherwise. We then compute the transition count matrix  $C_m = (c_{uv})_{u,v=1}^m$ , where each entry represents the number of transitions from bin  $u$  to bin  $v$  in the embedding sequence, for any  $1 \leq u, n \leq m$ ,

$$c_{uv} = \sum_{i=1}^{n-1} \mathbb{1}\{h_i \in \mathcal{H}_u \text{ and } h_{i+1} \in \mathcal{H}_v\}. \quad (6)$$

We then construct the empirical transition probability matrix  $Q_m := (q_{uv})_{u,v=1}^m$  by normalizing each row:

$$q_{uv} = \frac{c_{uv}}{\sum_{v'=1}^m c_{uv'}}. \quad (7)$$

This matrix represents the estimated probabilities of transitioning from bin  $u$  to bin  $v$ .

In Lemma 2, we establish that this binning method effectively transforms the original continuous-state Markov chain into a discrete-state Markov chain. The proof is provided in Appendix A.2.

**Lemma 2.** *For a given continuous-state homogeneous Markov chain  $\{h_i\}_{i=1, \dots, n}$ , the corresponding discretized sequence  $\{y_i\}_{i=1, \dots, n}$  forms a homogeneous Markov chain in a discrete state space.*

We denote the empirical transition count and probability matrices obtained from  $D_0$  as  $C_m^{(0)}$  and  $Q_m^*$ , following (6)-(7), and the corresponding matrices from  $D_1$  as  $C_m^{(1)}$  and  $\hat{Q}_m$ . By comparing the transition probability matrices  $Q_m^*$  and  $\hat{Q}_m$ , we can perform the chi-square test to determine if there is a significant difference between the two distributions (Billingsley, 1961).

*Optimal Bin Number Selection.* Selecting an appropriate number of bins  $m$  is crucial for effectively capturing the distributional differences between the two data sequences. An inadequate choice of  $m$  can either oversimplify the data (if  $m$  is too small) or introduce noise and sparsity issues (if  $m$  is too large), adversely affecting the power of the statistical test.

To determine the optimal number of bins  $m$ , we formulate the following optimization problem, inspired by Zhu et al. (1998):

$$\max_{m \geq 1} \left\{ \|Q_m^* - \hat{Q}_m\|_F + \lambda \left( S(Q_m^*) + S(\hat{Q}_m) \right) \right\},$$

where  $\|\cdot\|_F$  denotes the Frobenius norm that measures the discrepancy between two transition matrices,  $\lambda$  is a user-defined parameter that control the trade-off between the discrepancy and the smoothness (or roughness) of the matrices, and  $S(\cdot)$  is the smoothness measure of the transition matrix and is defined as:

$$S(Q_m) = -\sqrt{\sum_{u=2}^{m-1} \sum_{v=2}^{m-1} (\nabla^2 q_{uv})^2}.$$

Here  $\nabla^2 q_{uv}$  denotes the second derivatives of  $q_{uv}^{(k)}$ , and in our binning scenario it is calculated as  $\nabla^2 q_{uv} = q_{u+1,v} + q_{u-1,v} + q_{u,v+1} + q_{u,v-1} - 4q_{uv}$ .

The rationale behind this optimization is to select the optimal number of bins  $m$  by balancing two objectives: maximizing the discrepancy between the transition matrices of the real and generated data (to enhance the test’s sensitivity to differences), and promoting smoothness in these matrices (to prevent noise and sparsity), thereby achieving a trade-off that leads to strong testing power. By optimizing this balance, we ensure the transition matrices capture meaningful differences without being affected by overfitting or random fluctuations. We validate this procedure empirically and find that it enables the transition matrices to capture enough information about the differences in the trajectories, ultimately leading to strong testing power. The details of this procedure are provided in Appendix B.

**Step 2:  $\chi^2$  Transition Discrepancy Test** To assess the differences between the transition probability matrices  $Q_m^*$  and  $\hat{Q}_m$ , we adopt the chi-square statistics, denoted by  $W_m$ . The chi-square test is well-suited for measuring discrepancies between categorical distributions and, in our context, directly quantifies the differences between the transition matrices. We also establish its asymptotic distribution under temporal regimes following Billingsley (1961) in Proposition 2, with the proof provided in Appendix A.4.

**Proposition 2.** *Given the transition count matrices  $C_m^{(k)}$  and the corresponding  $Q_m^* = (q_{uv}^*)$  and  $\hat{Q}_m = (\hat{q}_{uv})$  from (7), and total event counts  $n_k$  for  $k \in \{0, 1\}$ , a chi-square test statistics  $W_m$  is given by*

$$W_m = \sum_{u=1}^m \sum_{v=1}^m \frac{c_u^{(0)} c_v^{(1)}}{c_{uv}^{(0)} + c_{uv}^{(1)}} (q_{uv}^* - \hat{q}_{uv})^2 \quad (8)$$

where  $c_u^{(k)} = \sum_{v=1}^m c_{uv}^{(k)}$  is the total count of transitions from state  $u$ . Under the null hypothesis  $H_0 : Q^* = \hat{Q}$ , as  $n_0, n_1 \rightarrow \infty$ , the test statistics  $W_m$  asymptotically follows a chi-square distribution with  $m(m-1)$  degrees of freedom:

$$W_m \sim \chi_{m(m-1)}^2.$$

Given the fitted chi-squared statistics  $W_m$ , we test the reformulated null hypothesis in (5) by setting the rejection region of significance level  $\alpha$  as:

$$R_m(\alpha) = \left\{ W_m \geq \chi_{m(m-1)}^2(\alpha) \right\}, \quad (9)$$

where  $\chi_{m(m-1)}^2(\alpha)$  denotes the  $(1 - \alpha)$ -quantile of the chi-square distribution with  $m(m-1)$  degrees of freedom, and  $\alpha$  is the pre-specified significance level. If the null hypothesis is rejected, we conclude that the sequences  $D_1$  are not generated from the distribution of  $D_0$ ; otherwise, we infer the opposite. As a direct consequence of Proposition 2, the type-I error rate (the probability of incorrectly rejecting  $H_0$  when it is true) associated with this rejection region is asymptotically bounded by  $\alpha$ . This underscores the effectiveness of the proposed testing procedure in controlling false positives under the asymptotic regime.

We highlight RENAL’s strong computational scalability in large-scale data settings due to the following factors: (i) Several acceleration techniques can enhance RENAL’s scalability, such as parallel processing with attention mechanisms (Bahdanau, 2014) and leveraging pre-trained models (Franceschi et al., 2019) to reduce the number of training iterations required for convergence. (ii) RENAL extracts a low-dimensional representation of temporal dependencies using a compact transition matrix, which minimizes the computational overhead of the binning optimization procedure, even with large training datasets. (iii) As a parametric GOF testing method, RENAL’s inference cost scales with the number of model parameters rather than the number of training samples, making it more efficient than nonparametric methods in large-scale applications.

### 3 EXPERIMENTS

In this section, we validate our proposed goodness-of-fit test against competing baselines (Section C.1)

---

**Algorithm 1:** RENAL Goodness-of-Fit Test

---

**Input** :  $D_0 = \{x_1^{(0)}, \dots, x_{n_0}^{(0)}\}$ ,  $D_1 = \{x_1^{(1)}, \dots, x_{n_1}^{(1)}\}$ ,  
smoothing parameter  $\lambda$ , set of bin numbers  
 $M$ , significance level  $\alpha$ .

**Initialization:**  $h_0^{(0)}, h_0^{(1)}, m^* \leftarrow 2, \ell^* \leftarrow -\infty$ ;  
 $\hat{\theta} \leftarrow$  Learn model  $\phi(\cdot; \theta)$  using  $D_0$ ;  
**for**  $k \in \{0, 1\}$  **do**  
    **for**  $i = 1$  **to**  $n_k$  **do**  
         $h_i^{(k)} = \phi(h_{i-1}^{(k)}, x_i^{(k)}; \hat{\theta})$ ;  
    **end**  
**end**  
 $m^* \leftarrow$   
     $\operatorname{argmax}_{m'} \{\|Q_{m'}^* - \hat{Q}_{m'}\| + \lambda (S(Q_{m'}^*) + S(\hat{Q}_{m'}))\}$ ;  
where  $Q_m^*$  and  $\hat{Q}_m$  are computed using  $C_m^{(0)}$  and  $C_m^{(1)}$ ,  
according to (6) and (7);  
Compute test statistics  $W_{m^*}$  using (8);  
**if**  $W_{m^*} > \chi_{m^*(m^*-1)}^2(\alpha)$  **then**  
    **return** Reject;  
**else**  
    **return** Accept;

---

across a broad spectrum of sequential data paradigms. Synthetic and real data studies are shown in Section 3.2 and Section 3.3 respectively.

### 3.1 Experiment Setup

We first introduce the baselines, evaluation metrics, and choices of neural network models used in the experiments. More detailed baselines and specific configurations are provided in Appendix C.1 and C.2.

**Baselines** We consider a broad range of existing GOF tests including: (i) Empirical Likelihood (EL): a test using kernel smoothing to assess the empirical likelihood for an  $\alpha$ -mixing process (Chen et al., 2003), (ii) Portmanteau test with Weighted McLeod–Li statistics (PT- $Q_w$ ): a portmanteau test using the trace of squared autocorrelation matrices for stable statistics (Gallagher and Fisher, 2015), (iii) Sieve-based Cramer-von Mises (S-CvM): a sieve-bootstrap based test to approximate  $p$ -values of the Cramer-von Mises test (Escanciano, 2006), (iv) Maximum Mean Discrepancy (MMD): a kernel-based test measuring MMD distance (Gretton et al., 2012), (v) Stein-based Bootstrap test (Stein): a test using a Stein operator value along with a wild bootstrap (Chwialkowski et al., 2016), (vi) Kernel Stein Discrepancy (KSD): a kernel Stein test specifically designed for general point processes (Yang et al., 2019). We also consider other baselines using recurrent neural representations with different partitions with details available in Appendix C.1.

**Evaluation Metrics** We use standard Type-I and Type-II accuracies to assess the performance of the proposed method. Type-I accuracy refers to the prob-

ability of correctly accepting the null hypothesis when it is true, while Type-II accuracy reflects the probability of correctly rejecting the null when it is false. To compare overall performance across different scenarios, we also compute the ‘‘Average’’ accuracies in the Tables below, by simulating multiple data sequence pairs from both  $H_0$  and  $H_1$ . The average accuracy is defined as the proportion of correct decisions, i.e., accepting  $H_0$  when true and rejecting it when false.

**Model Configuration** For our method, we use LSTM (Graves and Graves, 2012) as the embedding function  $\phi$  for regularly sampled time series data, and Neural Hawkes (Mei and Eisner, 2017) for irregularly sampled time series data, such as temporal point processes. The neural network configurations and discretization settings are described in Appendix C.2. All baseline methods are applied following their original implementations, with set-up details available in Appendix C.1. The significance level for all chi-square tests was set at  $\alpha = 0.05$ , which is used to determine the rejection region as in (9).

### 3.2 Synthetic Studies

We investigate the performance of RENAL through synthetic experiments with predefined ground truths and fitted models. More detailed descriptions of the testing procedures are deferred to Appendix C.4.

**Synthetic Data** We consider three types of time series data in our synthetic study: (i) *Time Series Data* consists of data generated with three models: two stationary ARMA(2,1) and ARMA(2,2) models, and a GARCH (1, 1) model, (ii) *Temporal Point Processes (TPP) Data* consists of data generated from two TPP models: self-exciting point process (SE) and self-correcting (SC) point process, where the occurrence of one event increases/decreases the likelihood of future events correspondingly, (iii) *Spatial-Temporal Point Process (STPP) Data* consists of data generated with two STPP models: the standard diffusion (STD) model which describes symmetric event spread in all directions; the Gaussian diffusion (GAU) (Musmeci and Vere-Jones, 1992) model with diffusion shape and orientation varying with location. More details of these models are provided in Appendix C.3.

**Result** The test accuracies under various scenarios are summarized in Table 1. It shows that RENAL consistently outperforms other methods. It is worthwhile mentioning that when  $P^* = \hat{P}$ , the reported accuracy corresponds to Type-I accuracy, whereas when  $P^* \neq \hat{P}$ , it corresponds to Type-II accuracy. Additionally, we report the overall accuracy for each of the three types of time series by averaging the results across all scenarios in Table 1. Due to the limited sample size

Table 1: Testing accuracy on synthetic dataset with  $\alpha = 0.05$ . “–” indicates the method is not applicable.

Methods	Time Series						TPP					STPP					
	$P^*$	Average	ARMA(2, 1)	ARMA(2, 1)	ARMA(2, 2)	ARMA(2, 1)	ARMA(2, 2)	Average	SE	SC	SE	SC	Average	STD	GAU	STD	GAU
	$\hat{P}$		ARMA(2, 1)	ARMA(2, 2)	ARMA(2, 2)	GARCH(1, 1)	GARCH(1, 1)		SE	SC	SE	SC		STD	GAU	GAU	STD
EL		0.37	0.08	0.33	0.23	0.76	0.64	–	–	–	–	–	–	–	–	–	–
PT- $Q_W$		0.52	0.91	0.15	0.93	0.95	0.08	–	–	–	–	–	–	–	–	–	–
S-CvM		0.44	0.98	0.08	0.97	0.02	0.04	–	–	–	–	–	–	–	–	–	–
Stein		–	–	–	–	–	–	0.51	0.32	0.47	0.51	0.74	–	–	–	–	–
KSD		–	–	–	–	–	–	0.56	0.66	0.78	0.72	0.08	–	–	–	–	–
MMD		0.62	0.82	0.15	0.72	0.75	0.82	0.45	0.53	0.61	0.31	0.35	0.43	0.65	0.68	0.10	0.26
EWD-2		0.55	0.70	0.10	0.68	0.70	0.84	0.53	0.80	0.69	0.25	0.37	0.44	0.75	0.80	0.07	0.13
EWD-4		0.61	0.45	0.65	0.37	0.98	0.87	0.53	0.46	0.33	0.63	0.68	0.48	0.13	0.09	0.93	0.75
EWD-6		0.66	0.60	0.52	0.58	0.95	0.91	0.56	0.54	0.63	0.54	0.52	0.38	0.68	0.63	0.07	0.13
EWD-8		0.66	0.80	0.15	0.76	0.94	0.84	0.53	0.65	0.76	0.38	0.39	0.48	0.83	0.88	0.12	0.17
EWD-10		0.67	0.95	0.15	0.85	0.77	0.65	0.52	0.80	0.80	0.22	0.29	0.46	0.94	0.85	0.05	0.09
Scott		0.56	0.97	0.08	0.47	0.72	0.58	0.43	0.99	1	0.03	0.01	0.49	0.98	1	0.005	0.001
RENAL		<b>0.72</b>	0.71	0.60	0.65	0.92	0.95	<b>0.61</b>	0.64	0.62	0.58	0.56	<b>0.60</b>	0.70	0.57	0.67	0.44

 Table 2: Testing accuracy on real data with  $\alpha = 0.05$ . For earthquake data, we consider two cases: TPP (time only) and STPP (spatio-temporal).

Model	Weather Time Series						Earthquake TPP					Earthquake STPP				
	$P^*$	Average	PIT	SFO	SFO	PIT	Average	JP	NC	NC	JP	Average	JP	JP	NC	NC
	$\hat{P}$		PIT	SFO	PIT	SFO		JP	NC	JP	NC		JP	NC	JP	NC
MMD		0.52	1	0.98	0.02	0.01	0.37	0.24	0.47	0.42	0.34	0.51	0.15	0.05	0.98	0.97
RENAL		<b>0.66</b>	0.65	0.67	0.58	0.75	<b>0.57</b>	0.57	0.63	0.52	0.54	<b>0.62</b>	0.6	0.37	0.72	0.67

(single sequence), all methods do not exactly meet the pre-specified Type I error rate of  $\alpha = 0.05$  used to define the rejection region. Despite this, our method, **RENAL**, consistently achieves one of the highest Type I accuracies and best Type II accuracies with superior performance since: (i) It exhibits the highest overall accuracy and balanced performance across all scenarios; (ii) It requires few model specifications or assumptions, and (iii) It is applicable across all settings while some baseline methods are limited to certain cases.

For *Time Series Data*, **RENAL** achieves strong overall performance in both Type I and Type II accuracies. In contrast, while some baseline methods can easily distinguish ARMA from GARCH due to their structural differences, they exhibit low accuracy when differentiating between two ARMA models. For *TPP* and *STPP Data*, **RENAL** also effectively capture time-dependency, achieving balanced Type I and II accuracies. However, the baseline methods either have low Type-I accuracy or fail to distinguish different models as reflected by their low type-II accuracy. Furthermore, both **EWD- $m$**  and **Scott** suffer from imbalance accuracies and highly depend on the number of partitions – denser partitions lead to sparse transition matrices and thus lower Type II accuracy, while coarser partitions may miss nuanced transition probabilities and reduce the Type I accuracy. **RENAL** address these issues with adaptive binning.

Additionally, the results shown in Table 1 support our earlier claim that **RENAL** is robust to assumption violations. This is because **RENAL** consistently achieves

higher and more stable accuracy than baseline methods, even when the sample size is relatively small (Table 4 in Appendix C) with only a single training data sequence of moderate length, which could potentially violate Assumption 1.

### 3.3 Real Data Study

We also assess the effectiveness of **RENAL** on real data including earthquake and weather data. Details of the testing procedures are provided in Appendix C.4.

**Data Description** We include two datasets for our study on real data: (i) *Earthquake Data* contains all earthquake data with magnitudes greater than 3 in North California (Waldhauser and Schaff, 2008) and Japan (from Japan Meteorological Agency (JMA) earthquake catalog) from January 1, 1993, to December 31, 2011, (ii) *Weather Data* includes daily temperature records from Pittsburgh International Airport and San Francisco International Airport starting from January 1st, 1994 (Vose et al., 2014).

**Results** In real-world data studies, we will compare our method only to **MMD**, as most other baselines are inapplicable without knowledge of the ground truth data distribution.

The result is shown in Table 2. It can be seen that across all datasets **RENAL** consistently achieves higher average accuracy, and strikes a better balance between Type I accuracy and Type II accuracy compared with **MMD**. This is because **MMD** fails to model temporal dependencies in data sequences, resulting in extreme



cases of either predicting nearly all false positives or all false negatives, which on the other hand highlights our method’s efficacy in tackling complex time series testing problems. We note that both RENAL and MMD show imbalanced results for the earthquake STPP data, this is likely due to inherent high spatial variation in the NC data.

## 4 CONCLUSION

We proposed a novel framework called the REcurrent NeurAL Goodness-of-Fit test (RENAL), tailored to effectively assess the quality of generative time series models. Our approach transforms the time series into conditionally independent pairs of history embeddings using recurrent neural networks, followed by a chi-square-based goodness-of-fit test to evaluate the dependencies between real and generated data. We provide theoretical justification for this framework and demonstrate that optimizing the discretization of history embeddings enhances its performance. In our numerical studies, RENAL consistently outperforms state-of-the-art methods, achieving higher and more balanced testing accuracy across synthetic and real-world datasets, including time series, point processes, and spatio-temporal point processes.

## References

- Akaike, H. (1974). A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723.
- Albergo, M. S. and Vanden-Eijnden, E. (2023). Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*.
- Bahdanau, D. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Baum, J., Kanagawa, H., and Gretton, A. (2023). A kernel stein test of goodness of fit for sequential models. In *International Conference on Machine Learning*, pages 1936–1953. PMLR.
- Billingsley, P. (1961). Statistical methods in markov chains. *The annals of mathematical statistics*, pages 12–40.
- Box, G. E. and Pierce, D. A. (1970). Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association*, 65(332):1509–1526.
- Burnham, K. P. and Anderson, D. R. (2004). Multimodel inference: understanding aic and bic in model selection. *Sociological methods & research*, 33(2):261–304.
- Chan, A. B. and Vasconcelos, N. (2005). Probabilistic kernels for the classification of auto-regressive visual processes. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, volume 1, pages 846–851. IEEE.
- Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in neural information processing systems*, 31.
- Chen, S. X., Härdle, W., and Li, M. (2003). An empirical likelihood goodness-of-fit test for time series. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 65(3):663–678.
- Chernoff, H. (1954). On the distribution of the likelihood ratio. *The Annals of Mathematical Statistics*, pages 573–578.
- Chwialkowski, K., Strathmann, H., and Gretton, A. (2016). A kernel test of goodness of fit. In *International conference on machine learning*, pages 2606–2615. PMLR.
- Chwialkowski, K. P., Ramdas, A., Sejdinovic, D., and Gretton, A. (2015). Fast two-sample testing with analytic representations of probability measures. *Advances in Neural Information Processing Systems*, 28.
- D’Agostino, R. B. (2017). *Goodness-of-fit-techniques*. Routledge.
- Darwin, J. H. (1959). Note on the comparison of several realizations of a markoff chain. *Biometrika*, 46(3/4):412–419.
- Dong, Z., Fan, Z., and Zhu, S. (2023). Conditional generative modeling for high-dimensional marked temporal point processes. *arXiv preprint arXiv:2305.12569*.
- Eric, M., Bach, F., and Harchaoui, Z. (2007). Testing for homogeneity with kernel fisher discriminant analysis. *Advances in Neural Information Processing Systems*, 20.
- Escanciano, J. C. (2006). Goodness-of-fit tests for linear and nonlinear time series models. *Journal of the American Statistical Association*, 101(474):531–541.
- Feller, W. and Morse, P. M. (1958). An introduction to probability theory and its applications.
- Franceschi, J.-Y., Dieuleveut, A., and Jaggi, M. (2019). Unsupervised scalable representation learning for multivariate time series. *Advances in neural information processing systems*, 32.
- Gallagher, C. M. and Fisher, T. J. (2015). On weighted portmanteau tests for time-series goodness-of-fit. *Journal of Time Series Analysis*, 36(1):67–83.

- Graves, A. and Graves, A. (2012). Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45.
- Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. (2012). A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773.
- Gretton, A., Fukumizu, K., Harchaoui, Z., and Sriperumbudur, B. K. (2009). A fast, consistent kernel two-sample test. *Advances in neural information processing systems*, 22.
- Han, Z., Zhao, J., Leung, H., Ma, K. F., and Wang, W. (2019). A review of deep learning models for time series prediction. *IEEE Sensors Journal*, 21(6):7833–7848.
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Jitkrittum, W., Szabó, Z., Chwialkowski, K. P., and Gretton, A. (2016). Interpretable distribution features with maximum testing power. *Advances in Neural Information Processing Systems*, 29.
- Kirchler, M., Khorasani, S., Kloft, M., and Lippert, C. (2020). Two-sample testing using deep learning. In *International Conference on Artificial Intelligence and Statistics*, pages 1387–1398. PMLR.
- Li, S., Xiao, S., Zhu, S., Du, N., Xie, Y., and Song, L. (2018). Learning temporal point processes via reinforcement learning. *CoRR*, abs/1811.05016.
- Li, Y., Lu, X., Wang, Y., and Dou, D. (2022). Generative time series forecasting with diffusion, denoise, and disentanglement. *Advances in Neural Information Processing Systems*, 35:23009–23022.
- Liu, F., Xu, W., Lu, J., Zhang, G., Gretton, A., and Sutherland, D. J. (2020). Learning deep kernels for non-parametric two-sample tests. In *International conference on machine learning*, pages 6316–6326. PMLR.
- Liu, X., Gong, C., and qiang liu (2023). Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*.
- Ljung, G. M. and Box, G. E. (1978). On a measure of lack of fit in time series models. *Biometrika*, 65(2):297–303.
- Mei, H. and Eisner, J. M. (2017). The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in neural information processing systems*, 30.
- Moreno, P., Ho, P., and Vasconcelos, N. (2003). A kullback-leibler divergence based kernel for svm classification in multimedia applications. *Advances in neural information processing systems*, 16.
- Musmeci, F. and Vere-Jones, D. (1992). A space-time clustering model for historical earthquakes. *Annals of the Institute of Statistical Mathematics*, 44:1–11.
- Ogata, Y. (1981). On lewis’ simulation method for point processes. *IEEE transactions on information theory*, 27(1):23–31.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Pearson, K. (1900). X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175.
- Sattarov, T., Schreyer, M., and Borth, D. (2023). Fintdiff: Diffusion models for financial tabular data generation. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pages 64–72.
- Schrab, A., Kim, I., Albert, M., Laurent, B., Guedj, B., and Gretton, A. (2023). Mmd aggregated two-sample test. *Journal of Machine Learning Research*, 24(194):1–81.
- Scott, D. W. (1979). On optimal and data-based histograms. *Biometrika*, 66(3):605–610.
- Suh, N., Lin, X., Hsieh, D.-Y., Honarkhah, M., and Cheng, G. (2023). Autodiff: combining auto-encoder and diffusion model for tabular data synthesizing. *arXiv preprint arXiv:2310.15479*.
- Suh, N., Yang, Y., Hsieh, D.-Y., Luan, Q., Xu, S., Zhu, S., and Cheng, G. (2024). Timeautodiff: Combining autoencoder and diffusion model for time series tabular data synthesizing. *arXiv preprint arXiv:2406.16028*.
- Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.
- Vose, R. S., Applequist, S., Squires, M., Durre, I., Menne, M. J., Williams, C. N., Fenimore, C., Gleason, K., and Arndt, D. (2014). Improved historical temperature and precipitation time series for us climate divisions. *Journal of Applied Meteorology and Climatology*, 53(5):1232–1251.

- Waldhauser, F. and Schaff, D. P. (2008). Large-scale relocation of two decades of northern california seismicity using cross-correlation and double-difference methods. *Journal of Geophysical Research: Solid Earth*, 113(B8).
- Wei, S., Zhu, S., Zhang, M., and Xie, Y. (2021). Goodness-of-fit test for mismatched self-exciting processes. In *International Conference on Artificial Intelligence and Statistics*, pages 1243–1251. PMLR.
- Wu, S., Diao, E., Elkhailil, K., Ding, J., and Tarokh, V. (2022). Score-based hypothesis testing for unnormalized models. *IEEE Access*, 10:71936–71950.
- Wu, S., Zhou, W., Chen, M., and Zhu, S. (2024). Counterfactual generative models for time-varying treatments. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 3402–3413, New York, NY, USA. Association for Computing Machinery.
- Xiao, S., Farajtabar, M., Ye, X., Yan, J., Song, L., and Zha, H. (2017). Wasserstein learning of deep generative point process models. *Advances in neural information processing systems*, 30.
- Yang, J., Rao, V., and Neville, J. (2019). A stein-papangelou goodness-of-fit test for point processes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 226–235. PMLR.
- Yoon, J., Jarrett, D., and Van der Schaar, M. (2019). Time-series generative adversarial networks. *Advances in neural information processing systems*, 32.
- Zhang, K., Wen, Q., Zhang, C., Cai, R., Jin, M., Liu, Y., Zhang, J. Y., Liang, Y., Pang, G., Song, D., et al. (2024). Self-supervised learning for time series analysis: Taxonomy, progress, and prospects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Zhu, S., Li, S., Peng, Z., and Xie, Y. (2021a). Imitation learning of neural spatio-temporal point processes. *IEEE Transactions on Knowledge and Data Engineering*, 34(11):5391–5402.
- Zhu, S., Wang, H., Cheng, X., and Xie, Y. (2022). Neural spectral marked point processes. In *International Conference on Learning Representations*.
- Zhu, S., Yao, R., Xie, Y., Qiu, F., Qiu, Y., and Wu, X. (2021b). Quantifying grid resilience against extreme weather using large-scale customer power outage data. *arXiv preprint arXiv:2109.09711*.
- Zhu, S., Yuchi, H. S., Zhang, M., and Xie, Y. (2023). Sequential adversarial anomaly detection for one-class event data. *INFORMS Journal on Data Science*, 2(1):45–59.
- Zhu, W., Wang, Y., and Zhu, Q.-F. (1998). Second-order derivative-based smoothness measure for error concealment in dct-based codecs. *IEEE Transactions on Circuits and Systems for video Technology*, 8(6):713–718.
- Zuo, S., Jiang, H., Li, Z., Zhao, T., and Zha, H. (2020). Transformer hawkes process. In *International conference on machine learning*, pages 11692–11702. PMLR.

## Checklist

- For all models and algorithms presented, check if you include:
  - A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
  - An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Not Applicable]
  - (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes/No/Not Applicable]
- For any theoretical claim, check if you include:
  - Statements of the full set of assumptions of all theoretical results. [Yes]
  - Complete proofs of all theoretical results. [Yes]
  - Clear explanations of any assumptions. [Yes]
- For all figures and tables that present empirical results, check if you include:
  - The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
  - All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
  - A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
  - A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
- If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
  - Citations of the creator If your work uses existing assets. [Yes]
  - The license information of the assets, if applicable. [Not Applicable]

- (c) New assets either in the supplemental material or as a URL, if applicable. [**Not Applicable**]
  - (d) Information about consent from data providers/curators. [**Not Applicable**]
  - (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [**Not Applicable**]
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
- (a) The full text of instructions given to participants and screenshots. [**Not Applicable**]
  - (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [**Not Applicable**]
  - (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [**Not Applicable**]

## A PROOFS

### A.1 Proof of Lemma 1

For any subset  $B \subset \mathcal{H}$ , the left-hand side of the Markov property becomes

$$\begin{aligned} & \mathbb{P}(h_{i+1} \in B \mid h_i, h_{i-1}, \dots, h_1) \\ &= \int_{x \in \mathcal{X}} \mathbb{P}(h_{i+1} \in B \mid h_i, h_{i-1}, \dots, h_1, X_i = x) f_{X_i}(x \mid h_i, h_{i-1}, \dots, h_1) dx, \end{aligned}$$

where  $f_{X_i}(\cdot \mid h_i, h_{i-1}, \dots, h_1)$  is the conditional density function of  $X_i$ , the  $i$ -th observation in the time series, conditioned on all past embeddings. Since  $h_{i+1}$  is determined by  $h_i$  and  $X_i$ , we have:

$$\mathbb{P}(h_{i+1} \in B \mid h_i, h_{i-1}, \dots, h_1, X_i = x) = \mathbb{P}(h_{i+1} \in B \mid h_i, X_i = x).$$

Moreover, note that the distribution of  $X_i$  is fully determined by  $h_i$  in the decoding process. Thus we have

$$f_{X_i}(x \mid h_i, h_{i-1}, \dots, h_1) = f_{X_i}(x \mid h_i).$$

Therefore, we simplify the left-hand side expression to

$$\mathbb{P}(h_{i+1} \in B \mid h_i, h_{i-1}, \dots, h_1) = \int_{x \in \mathcal{X}} \mathbb{P}(h_{i+1} \in B \mid h_i, X_i = x) f_{X_i}(x \mid h_i) dx = \mathbb{P}(h_{i+1} \in B \mid h_i).$$

To prove homogeneity, we first note that  $\phi$  is homogeneous across different time steps (namely we are using the same embedding network across times), *i.e.*,  $h_{i+1} = \phi(x_i, h_i; \theta)$  for all  $i$ . Therefore, for any  $y \in \mathcal{H}$  and  $x \in \mathcal{X}$ ,

$$\begin{aligned} \mathbb{P}(h_i \in B \mid h_{i-1} = y, X_{i-1} = x) &= \mathbb{P}(\phi(X_{i-1}, h_{i-1}) \in B \mid h_{i-1} = y, X_{i-1} = x) \\ &= \mathbb{P}(\phi(X_i, h_i) \in B \mid h_i = y, X_i = x) \\ &= \mathbb{P}(h_{i+1} \in B \mid h_i = y, X_i = x). \end{aligned} \tag{10}$$

Since the conditional distribution of  $X_i$ , given  $h_i$  is also stationary, as the distribution is determined by the embedding function  $g$ , we have

$$\begin{aligned} \mathbb{P}(h_{i+1} \in B \mid h_i = y) &= \int_{x \in \mathcal{X}} \mathbb{P}(h_{i+1} \in B \mid h_i = y, X_i = x) f_{X_i \mid h_i}(x \mid y) dx \\ &\stackrel{(i)}{=} \int_{x \in \mathcal{X}} \mathbb{P}(h_i \in B \mid h_{i-1} = y, X_{i-1} = x) f_{X_{i-1} \mid h_{i-1}}(x \mid y) dx \\ &= \mathbb{P}(h_i \in B \mid h_{i-1} = y), \end{aligned}$$

where the equality (i) used the homogeneous property in (10).

### A.2 Proof of Lemma 2

Using Markov property of  $\{h_i\}_{i \in \mathbb{N}}$  and the fact that  $y_i$  deterministically determined by  $h_i$ ,

$$\begin{aligned} & \mathbb{P}(y_i = a_i \mid y_{i-1} = a_{i-1}, \dots, y_1 = a_1) \\ &= \mathbb{P}(h_i \in \mathcal{H}_{a_i} \mid h_{i-1} \in \mathcal{H}_{a_{i-1}}, \dots, h_1 \in \mathcal{H}_{a_1}) \\ &= \int_{\eta \in \mathcal{H}_{a_{i-1}}} \mathbb{P}(h_i \in \mathcal{H}_{a_i} \mid h_{i-1} = \eta) f_{h_{i-1}}(\eta \mid h_{i-1} \in \mathcal{H}_{a_{i-1}}, \dots, h_1 \in \mathcal{H}_{a_1}) d\eta \\ &= \int_{\eta \in \mathcal{H}_{a_{i-1}}} \mathbb{P}(h_i \in \mathcal{H}_{a_i} \mid h_{i-1} = \eta) f_{h_{i-1}}(\eta \mid h_{i-1} \in \mathcal{H}_{a_{i-1}}) d\eta \\ &= \mathbb{P}(h_i \in \mathcal{H}_{a_i} \mid h_{i-1} \in \mathcal{H}_{a_{i-1}}) \\ &= \mathbb{P}(y_i = a_i \mid y_{i-1} = a_{i-1}). \end{aligned}$$

Furthermore, when assuming the sequence  $\{h_i\}_{i \in \mathbb{N}}$  is stationary, using the homogeneity of  $\{h_i\}_{i \in \mathbb{N}}$  (Lemma 1) and the deterministic nature of  $y_i$ , for arbitrary discrete states  $a$  and  $\zeta$ ,

$$\begin{aligned} \mathbb{P}(y_{i+1} = a | y_i = \zeta) &= \mathbb{P}(h_{i+1} \in \mathcal{H}_a | h_i \in H_\zeta) = \int_{y \in H_\zeta} \mathbb{P}(h_{i+1} \in \mathcal{H}_a | h_i = y) f_{h_i}(y | h_i \in H_\zeta) dy \\ &= \int_{y \in H_\zeta} \mathbb{P}(h_i \in \mathcal{H}_a | h_{i-1} = y) f_{h_{i-1}}(y | h_{i-1} \in H_\zeta) dy \\ &= \mathbb{P}(y_i = a | h_{i-1} = \zeta). \end{aligned}$$

Thus,  $\{y_i\}_{i \geq 0}$  has the Markovian and homogeneous property.

### A.3 Proof of Proposition 1

According to Lemma 1, the embedding sequence  $\{h_i\}_{i=1}^n$  forms a homogeneous Markov chain. This implies that the distribution  $\mathbb{P}$  can be fully characterized by the transition densities between successive history embeddings. Thus, for two data sequences  $D_0 \sim \mathbb{P}^*$  and  $D_1 \sim \hat{\mathbb{P}}$ , the distributions  $\mathbb{P}^*$  and  $\hat{\mathbb{P}}$  can be uniquely and sufficiently represented by their respective transition mappings  $Q^*$  and  $\hat{Q}$ . The hypothesis test (5) that tests the equality of these transition mappings is therefore equivalent to testing whether  $\mathbb{P}^*$  and  $\hat{\mathbb{P}}$  are the same.

### A.4 Proof of Proposition 2

The proof basically follows the corresponding proof in (Billingsley, 1961), with some shortening and elaborations to aid understanding. We will first set up and prove two lemmas, Lemma 3 and Lemma 4, which are useful to prove Proposition 2.

**Lemma 3** (Lemma 3.2 in (Billingsley, 1961)). *Let  $\{x_1, x_2, \dots, x_n\}$  be a sequence of samples from a first-order stationary discrete Markov chain with  $m$  states with a transition count matrix  $C_m = (c_{uv})_{u,v=1}^m$  and transition probability matrix  $Q_m = (q_{uv})_{u,v=1}^m$ . Denote  $c_u := \sum_{v=1}^m c_{uv}$  as total counts of state  $u$  and  $q_u := \sum_{v=1}^m q_{uv}$  as the total probability of state  $u$ . Then the weak law of large number holds, i.e., for arbitrary state  $u$ , as  $n \rightarrow \infty$ ,*

$$\frac{c_u}{n} \xrightarrow{P} q_u, \text{ as } n \rightarrow \infty. \quad (11)$$

*Proof.* Define the random variable  $z_s(u) := \mathbb{1}(x_s = u)$ . Then  $c_u = \sum_{s=1}^n z_s(u)$ . From stationarity of the chain, it follows that  $\mathbb{E}[z_s(u)] = q_u$ , so  $\mathbb{E}[c_u/n] = n^{-1} \mathbb{E}[\sum_{s=1}^n z_s(u)] = q_u$ ,

$$\text{Cov}(z_s(u), z_{s'}(u)) = \mathbb{E}[(z_s(u) - q_u)(z_{s'}(u) - q_u)] = P(x_s = u, x_{s'} = u) - q_u^2 = \begin{cases} q_u - q_u^2 & \text{if } s' = s \\ q_u q_{uu}^{|s'-s|} - q_u^2 & \text{otherwise} \end{cases},$$

and

$$\begin{aligned} \text{Var}\left(\frac{c_u}{n}\right) &= \frac{1}{n^2} \sum_{s=1}^n \sum_{s'=1}^n \text{Cov}(z_s(u), z_{s'}(u)) \\ &= \frac{1}{n^2} \left\{ n(q_u - q_u^2) + 2 \sum_{s=1}^n \sum_{s'=s+1}^n (q_u q_{uu}^{|s'-s|} - q_u^2) \right\} \\ &\leq \frac{1}{n^2} \left\{ n(q_u - q_u^2) + 2 \sum_{s=1}^n \sum_{s'=s+1}^n (q_u^2 - q_u^2) \right\} \quad (0 \leq q_{uu} \leq q_u \leq 1) \\ &\leq \frac{1}{n}. \end{aligned}$$

Therefore  $\text{Var}(\frac{c_u}{n}) \rightarrow 0$  as  $n \rightarrow \infty$ , then by Chebyshev's inequality, for  $\epsilon > 0$ ,

$$\mathbb{P}\left(\left|\frac{c_u}{n} - q_u\right| \geq \epsilon\right) \leq \frac{\text{Var}(\frac{c_u}{n})}{\epsilon^2} \leq \frac{1}{n\epsilon^2} \rightarrow 0, \text{ as } n \rightarrow \infty.$$

Therefore, as  $n \rightarrow \infty$ ,  $\frac{c_u}{n} \xrightarrow{P} q_u$ . □

**Lemma 4** (Theorem 3.1 from (Billingsley, 1961)). *Let  $\{x_1, x_2, \dots, x_n\}$  be a sequence of samples from a first-order stationary discrete Markov chain with  $m$  states with a transition count matrix  $C_m = (c_{uv})_{u,v=1}^m$  and transition probability matrix  $Q_m = (q_{uv})_{u,v=1}^m$ . Denote  $c_u := \sum_{v=1}^m c_{uv}$  as the total counts at state  $u$ . Let*

$$\xi_{uv} = \frac{c_{uv} - c_u q_{uv}}{c_u^{1/2}}.$$

*The distribution of the  $m^2$ -dimensional random vector  $\xi = (\xi_{uv})$  converges as  $n \rightarrow \infty$  to the normal distribution.*

*Proof.* The proof follows the proof for Theorem 3.1 in (Billingsley, 1961). The process  $\{x_n\}$  having a sequence of states  $\{a_n\}$  can be viewed as having been generated in the following fashion. Consider an independent collection of random variables  $x_1$  and  $r_{un}$  (for  $u = 1, 2, \dots, m$  and  $n = 1, 2, \dots$ ) such that  $P(x_1 = u) = q_u$  and  $P(r_{un} = v) = q_{uv}$ .

We begin by sampling  $x_1$ .  $x_2$  is defined to be  $x_2 = r_{x_1 1}$ . If  $x_1, x_2, \dots, x_n$  have been defined,  $x_{n+1}$  is defined to be  $r_{x_n, w}$ , where  $w - 1$  is the number of  $l$  ( $1 \leq l < n$ ) such that  $x_l = x_n$ . By definition,  $\{x_i = a_i, 1 \leq i \leq n\} = \{x_1 = a_1, r_{a_{i-1} w_{i-1}} = a_i, 2 \leq i \leq n\}$  where  $w_i - 1$  is the number of elements among  $\{a_1, \dots, a_{i-1}\}$  that are equal to  $a_i$ . Since the variables involved are all distinct and independent,

$$P(x_i = a_i, 1 \leq i \leq n) = P(x_1 = a_1)P(r_{a_1 w_1} = a_2) \cdots P(r_{a_{n-1} w_{n-1}} = a_n) = q_{a_1} q_{a_1 a_2} \cdots q_{a_{n-1} a_n}.$$

By process of generation,  $(c_{u1}, c_{u2}, \dots, c_{um})$  is the frequency count of  $\{r_{u1}, \dots, r_{uc_u}\}$ . By Lemma 3, as  $n \rightarrow \infty$ ,  $c_u \rightarrow nq_u$ .

We can then compare  $(c_{u1}^{(0)}, \dots, c_{um}^{(0)})$  with the frequency count  $(c_{u1}^{(1)}, \dots, c_{um}^{(1)})$  of  $\{r_{u1}, \dots, r_{u[nq_u^{(0)}]}\}$ . From the independence of the array  $\{r_{un}\}$  and the central limit theorem for multinomial trials, it follows that the  $m^2$  random variables  $\left( \frac{c_{uv}^{(1)} - [nq_u^{(0)}]q_{uv}^{(0)}}{\sqrt{nq_{uv}^{(0)}}} \right)$  have asymptotic jointly normal distribution. Define  $e_w$  as  $e_w = \mathbf{1}(r_{uw} = v) - q_{uv}^{(0)}$  and put  $R_w = e_1 + \dots + e_w$ . It is clear that  $e_w$ 's are independent and identically distributed with mean 0 and variance  $\sigma^2 = q_{uv}^{(0)}(1 - q_{uv}^{(0)})$ . Now

$$\frac{c_{uv}^{(1)} - [nq_u^{(0)}]q_{uv}^{(0)}}{n^{1/2}} - \frac{c_{uv}^{(0)} - c_u^{(0)}q_{uv}^{(0)}}{n^{1/2}} = \frac{R_{[nq_u^{(0)}]} - R_{c_u^{(0)}}}{n^{1/2}}. \quad (12)$$

By Lemma 3,  $\lim_{n \rightarrow \infty} \frac{c_u^{(0)}}{n} = q_u^{(0)}$ , for arbitrary  $\epsilon > 0$ ,  $\exists N$  sufficiently large such that  $\forall n > N$ ,  $P(|c_u^{(0)} - [nq_u^{(0)}]| > n\epsilon^3) < \epsilon$ , then

$$\begin{aligned} & P \left( \left| \frac{c_{uv}^{(1)} - [nq_u^{(0)}]q_{uv}^{(0)}}{n^{1/2}} - \frac{c_{uv}^{(0)} - c_u^{(0)}q_{uv}^{(0)}}{n^{1/2}} \right| > \epsilon \right) \\ &= P \left( \left| \frac{R_{[nq_u^{(0)}]} - R_{c_u^{(0)}}}{n^{1/2}} \right| > \epsilon \right) \\ &\leq P \left( |c_u^{(0)} - [nq_u^{(0)}]| > n\epsilon^3 \right) + P \left( \max_{|w - [nq_u^{(0)}]| \leq n\epsilon^3} |R_{[nq_u^{(0)}]} - R_w| > \epsilon n^{1/2} \right) \\ &\leq \epsilon + 2P \left( \max_{1 \leq w \leq n\epsilon^3} |R_w| > \frac{1}{2} \epsilon n^{1/2} \right) \\ &\leq \epsilon + 2 \left( \frac{4}{n\epsilon^2} \right) (n\epsilon^3 \sigma^2) = (1 + 8\sigma^2)\epsilon. \quad (\text{by Kolmogorov's inequality in (Feller and Morse, 1958)}) \end{aligned}$$

Since  $\epsilon$  can be arbitrarily small,  $\frac{c_{uv}^{(1)} - [nq_u^{(0)}]q_{uv}^{(0)}}{n^{1/2}} - \frac{c_{uv}^{(0)} - c_u^{(0)}q_{uv}^{(0)}}{n^{1/2}} \rightarrow 0$ , i.e.,  $\frac{c_{uv}^{(1)} - [nq_u^{(0)}]q_{uv}^{(0)}}{(nq_u^{(0)})^{1/2}} - \frac{c_{uv}^{(0)} - c_u^{(0)}q_{uv}^{(0)}}{(nq_u^{(0)})^{1/2}} \rightarrow 0$  and  $\frac{c_{uv}^{(0)} - c_u^{(0)}q_{uv}^{(0)}}{(nq_u^{(0)})^{1/2}}$  has the same limiting distribution of  $\frac{c_{uv}^{(1)} - [nq_u^{(0)}]q_{uv}^{(0)}}{(nq_u^{(0)})^{1/2}}$ . By Lemma 3,  $\xi_{uv} = \frac{c_{uv}^{(0)} - c_u^{(0)}q_{uv}^{(0)}}{c_u^{1/2}}$  has the same limiting distribution of  $\frac{c_{uv}^{(0)} - c_u^{(0)}q_{uv}^{(0)}}{(nq_u^{(0)})^{1/2}}$ .  $\xi = (\xi_{uv})$  converges to normal distribution as  $n \rightarrow \infty$ .  $\square$

Now recall that we assume two data sequences  $D_0$  and  $D_1$ , and obtain the transition count matrices  $C_m^{(k)} = (c_{uv}^{(k)})_{u,v=1}^m$ ,  $k = 0, 1$ , from  $D_0$  and  $D_1$ , respectively. Under the null hypothesis that  $D_0$  and  $D_1$  follow the same distribution, we have the population transition probability matrices for the learned historical embedding should equal to each other, i.e.,  $Q^* = \hat{Q}$ , we denote  $Q^* = \hat{Q} = (q_{uv})_{u,v=1}^m$ .

Define  $\xi_{uv} = \frac{c_{uv}^{(0)} - c_u^{(0)} q_{uv}}{c_u^{(0) \frac{1}{2}}}$ . By Lemma 4,  $\xi = (\xi_{uv})$  converges to normal distribution as  $n \rightarrow \infty$ . The frequency counts  $(c_{u1}^{(k)}, c_{u2}^{(k)}, \dots, c_{um}^{(k)})$  for  $k \in \{0, 1\}$  have multinomial distributions with probabilities  $(q_{u1}, q_{u2}, \dots, q_{um})$  and  $\mathbb{E}(c_{uv}^{(k)}) = c_u^{(k)} q_{uv}$ . By (Pearson, 1900), as  $c_u^{(k)} \rightarrow \infty$ ,

$$\sum_{v=1}^m \frac{(c_{uv}^{(k)} - c_u^{(k)} q_{uv})}{c_u^{(k)} q_{uv}} \sim \chi_{(m-1)}^2, \quad \text{and} \quad \sum_{u=1}^m \sum_{v=1}^m \frac{(c_{uv}^{(k)} - c_u^{(k)} q_{uv})^2}{c_u^{(k)} q_{uv}} \sim \chi_{m(m-1)}^2.$$

Under the null hypothesis, it is intuitively clear that the common estimate is  $\hat{q}_{uv} = \frac{c_{uv}^{(0)} + c_{uv}^{(1)}}{c_u^{(0)} + c_u^{(1)}}$  by (Billingsley, 1961; Darwin, 1959). Plugging in the common estimate, we have

$$\sum_{u=1}^m \sum_{v=1}^m \frac{(c_{uv}^{(0)} - c_u^{(0)} \frac{c_{uv}^{(0)} + c_{uv}^{(1)}}{c_u^{(0)} + c_u^{(1)}})^2}{c_u^{(0)} \frac{c_{uv}^{(0)} + c_{uv}^{(1)}}{c_u^{(0)} + c_u^{(1)}}} + \sum_{u=1}^m \sum_{v=1}^m \frac{(c_{uv}^{(1)} - c_u^{(1)} \frac{c_{uv}^{(0)} + c_{uv}^{(1)}}{c_u^{(0)} + c_u^{(1)}})^2}{c_u^{(1)} \frac{c_{uv}^{(0)} + c_{uv}^{(1)}}{c_u^{(0)} + c_u^{(1)}}} \sim \chi_{m(m-1)}^2.$$

The above expression can be simplified as,

$$\begin{aligned} & \sum_{uv} \frac{\left( c_{uv}^{(0)} - c_u^{(0)} \frac{c_{uv}^{(0)} + c_{uv}^{(1)}}{c_u^{(0)} + c_u^{(1)}} \right)^2}{c_u^{(0)} \frac{c_{uv}^{(0)} + c_{uv}^{(1)}}{c_u^{(0)} + c_u^{(1)}}} + \sum_{uv} \frac{\left( c_{uv}^{(1)} - c_u^{(1)} \frac{c_{uv}^{(0)} + c_{uv}^{(1)}}{c_u^{(0)} + c_u^{(1)}} \right)^2}{c_u^{(1)} \frac{c_{uv}^{(0)} + c_{uv}^{(1)}}{c_u^{(0)} + c_u^{(1)}}} \\ &= \sum_{uv} \frac{c_u^{(0)} c_u^{(1)}}{(c_{uv}^{(0)} + c_{uv}^{(1)}) (c_u^{(0)} c_u^{(1)})^2 (c_u^{(0)} + c_u^{(1)})} \left( c_u^{(0)} + c_u^{(1)} \right) \left( c_u^{(0)} c_{uv}^{(1)} - c_u^{(1)} c_{uv}^{(0)} \right)^2 \\ &= \sum_{uv} \frac{c_u^{(0)} c_u^{(1)}}{c_{uv}^{(0)} + c_{uv}^{(1)}} \left( \frac{c_{uv}^{(0)}}{c_u^{(0)}} - \frac{c_{uv}^{(1)}}{c_u^{(1)}} \right)^2 \\ &= \sum_{uv} \frac{c_u^{(0)} c_u^{(1)}}{c_{uv}^{(0)} + c_{uv}^{(1)}} \left( q_{uv}^{(0)} - q_{uv}^{(1)} \right)^2 \sim \chi_{m(m-1)}^2. \end{aligned}$$

## B ADDITIONAL TEST STATISTICS DETAILS

**Smoothing Constraints for Objective Function** The complete equation for the second derivative is shown below:

$$S(Q_m) = -\sqrt{\sum_{u=2}^{m-1} \sum_{v=2}^{m-1} (\nabla^2 q_{uv})^2} \approx -\sqrt{\sum_{u=2}^{m-1} \sum_{v=2}^{m-1} (q_{u+1,v} + q_{u-1,v} + q_{u,v+1} + q_{u,v-1} - 4q_{uv})^2}.$$

This smoothing constraint is chosen as the second derivative, which measures the curvature or the rate of change in the transition probabilities across states. Abrupt transitions (i.e., sudden large changes in transition probabilities between consecutive states) result in large second-order differences. Therefore, minimizing the square of the second-order differences (or equivalently, maximizing the negative square of the second-order differences) encourages smoother transitions. This smoothing regularization results in more stable transitions between states by penalizing abrupt changes, leading to gradual and consistent shifts in transition probabilities.



Table 3: Neural Network Specifications and Discretization Settings.

Type of Models	Neural Network					Discretization	
	Network Type	Hidden Dimension	$\ell$	Optimizer	$e$	$N_b$	$\lambda$
Time Series	LSTM	6	0.001	Adam	100	6	0.1
TPP	NeuralHawkes	4	0.00025	SGD	150	20	0.08
STPP	NHPP	4	0.0005	SGD	200	20	0.08

Incorporating the hyperparameters  $\lambda$  to control for the weights of smoothing, the objective is then:

$$\begin{aligned} \text{Objective} &= \|\{Q_m^*\} - \hat{Q}_m\|_F + \lambda \left( S(Q_m^*) + S(\hat{Q}_m) \right) \\ &= \sqrt{\sum_{u,v} (q_{uv}^* - \hat{q}_{uv})^2} - \lambda \left( \sqrt{\sum_{u=2}^{m-1} \sum_{v=2}^{m-1} (\nabla^2 q_{uv}^*)^2} + \sqrt{\sum_{u=2}^{m-1} \sum_{v=2}^{m-1} (\nabla^2 \hat{q}_{uv})^2} \right). \end{aligned}$$

## C EXPERIMENTS DETAILS

The following sections provide more details of the synthetic and real-world experiments. The code is available in <https://github.com/aoranzhangmia/Neural-GoF-Time>.

### C.1 Baseline Details

For **MMD**, **KSD**, and **Stein**, input sequences are split into 50 subsequences and the radial basis function kernel is used with bandwidth set as the median of pairwise squared Euclidean distances among data points. **KSD** utilizes the trapezoidal rule with 101 points for integration. For **Stein** test, the bootstrap flip probability is set at 0.01.

For discretization baselines, we construct **EWD**– $m$  and **Scott** discretization for a history embedding sequence  $\{h_i\} \subset \mathcal{H} \subseteq \mathbb{R}^p$  as follows: (i) **EWD**– $m$ : for each dimension  $p' = 1, 2, \dots, p$  of the latent space  $\mathcal{H} \subseteq \mathbb{R}^p$ , we partition the range of  $\{h_i\}_{p'}$  (the  $p'$ -th dimension of  $\{h_i\}$ ) into  $m$  equally sized intervals (bins). (ii) **Scott**: For each dimension  $p'$ , the number of bins ( $m^{p'}$ ) for the equal-sized binning is determined by Scott’s rule (Scott, 1979):

$$m^{p'} = \left\lceil \frac{\max(\{h_i\}_{p'}) - \min(\{h_i\}_{p'})}{3.5 \cdot \sigma_{\{h_i\}_{p'}} \cdot n^{-1/3}} \right\rceil,$$

where  $\sigma_{\{h_i\}_{p'}}$  is the standard deviation of the sequence  $\{h_i\}_{p'}$  and  $n$  is the length of the  $\{h_i\}$ .

### C.2 Time Series Generative Model Details

The detailed setting for the time series generative models used in the experiment can be referred to Table 3. Here  $\ell$  denotes learning rate,  $e$  denotes epoches,  $N_b$  represents the maximum number of bins per dimension, and  $\lambda$  sets the corresponding smoothing constraints. The maximum number of discretized states is 100, and ensure that transition count matrices have at least 15% non-zero entries to prevent overly sparse transition probability matrices. Note that as the dimension of latent space increase, the proportion of non-zero entries and  $\lambda$  should be increased accordingly to avoid too sparse transition probability matrices.

### C.3 Datasets Details

The general setting is presented in Table 4. **Type** denotes the model type if the dataset is synthetic and denotes the kind of data if the dataset is obtained from real observation. **Range** denotes the time range (one-dimensional) if the data is time series data, and time  $\times$  longitude  $\times$  latitude space (three dimensions) if it is spatial-temporal data. **Average Length** denotes the average number of data points in a single trajectory.

The model specification for synthetic data and specific information of real data are listed below:

Table 4: Synthetic and real dataset description.

	Time Series			TPP		STPP		Real Data	
Types	ARMA <sub>1</sub>	ARMA <sub>2</sub>	GARCH	SE	SC	STD	GAU	Earthquake	Weather
Range	[0,50)	[0,50)	[0,50)	[0,100)	[0,100)	Time: [0, 1), Space: [0, 1) × [0, 1)		Time: [0, 1), Space: [0, 1) × [0, 1)	[0,365)
Average Length	500	500	500	500	500	500	500	400	365
# of Trajectories	500	500	500	500	500	200	200	200	60

**Time Series Data** For an ARMA(p,q) model with autoregressive (AR) coefficients  $\phi_{arma} = (\phi_1, \dots, \phi_p)$ , moving average (MA) coefficients  $\theta_{arma} = (\theta_1, \dots, \theta_q)$ , and variance of the white noise ( $\sigma^2$ ), the model is given by:

$$x_i = \phi_1 x_{i-1} + \phi_2 x_{i-2} + \dots + \phi_p x_{i-p} + \epsilon_i + \theta_1 \epsilon_{i-1} + \dots + \theta_q \epsilon_{i-q},$$

where  $x_i$  is the value of the time series at index  $i$  and  $\epsilon_i$  is the white noise term at index  $i$  with  $i \sim N(0, \sigma^2)$ .

ARMA models we experiment are specified as:

1. *ARMA<sub>1</sub>*: ARMA(2,1) with  $\phi_{arma} = [0.5, 0.4]$ ,  $\theta_{arma} = 0.65$  and  $\sigma = 1$ .
2. *ARMA<sub>2</sub>*: ARMA (2,2) with  $\phi_{arma} = [0.5, -0.4]$ ,  $\theta_{arma} = [0.3, -0.2]$  and  $\sigma = 1$ .

Note that the neural ODE framework can be used to approximately model the ARMA time series. AR(1) model can be exactly modeled using our framework by letting  $h_i = \phi(h_{i-1}, x_{i-1}) = x_{i-1} = h_{i-1}$  and  $x_i = \alpha h_i + \epsilon_i$ . For AR(2),  $h_i = (x_{i-1}, x_{i-2})$ , thus  $h_i = \phi(h_{i-1}, x_{i-1})$  and  $x_i = \phi^\top h_i + \epsilon$ , *i.e.*, the  $g$  function is the linear function  $\phi^\top h_i$ . For ARMA models, the MA part involves past errors that make it complicated, but the neural ODE can still approximate how these errors propagate over time, thus implicitly capturing the time dependencies within the process.

The GARCH( $p, q$ ) model measuring the volatility over a time range is defined with the following parameters: the mean of the time series is  $\mu$ , the constant term representing the baseline level of volatility is  $\omega$ , the ARCH terms capturing the impact of past shocks are  $\alpha_{garch} = (\alpha_1, \dots, \alpha_p)$ , the GARCH terms capturing the persistence of volatility are  $\beta_{garch} = (\beta_1, \dots, \beta_q)$ , the leverage terms capturing the asymmetry for negative shocks are  $\gamma_{garch} = (\gamma_1, \dots, \gamma_p)$ , and the innovation term (shock) at time  $i$  is  $\eta_i$ . Given the parameters, GARCH( $p, q$ ) is defined with mean and variance equations:

$$\text{Mean Equation: } x_i = \mu + \eta_i,$$

$$\text{Variance Equation: } \sigma_i^2 = \omega + \sum_{j=1}^p \alpha_j \eta_{i-j}^2 + \sum_{j=1}^q \beta_j \sigma_{i-j}^2 + \sum_{j=1}^p \gamma_j \eta_{i-j} \mathbb{1}(\eta_{i-j} < 0),$$

where  $\eta_i = \epsilon_i \sigma_i$  represents the innovation with  $\epsilon_i \sim N(0, 1)$  and  $\sigma_i$  being the conditional standard deviation at index  $i$ . In the experiment, the model GARCH(1,1) is defined with  $\mu = 0.03$ ,  $\omega = 0.04$ ,  $\alpha_{garch} = 0.04$ ,  $\gamma_{garch} = 0.02$ ,  $\beta_{garch} = 0.9$ , and  $\eta = 1.0$ .

**TPP Data** Sequence are simulated using Ogata's thinning algorithm (Ogata, 1981). The models are set as follows:

1. *Self-Exciting process (SE)*: The occurrence of past events triggers future events. The conditional intensity function is given by  $\lambda(t|\mathcal{H}_i) = \mu + \alpha \sum_{i:t_i < t} \exp\{-\beta(t - t_i)\}$ , where  $\mu = 1$ ,  $\alpha = 1$ ,  $\beta = 1.25$  and  $\bar{\lambda} = 1e + 2$ .
2. *Self-Correcting Process (SC)*: The occurrence of events reduces the likelihood of future ones. The conditional intensity function is  $\lambda(t) = \exp(\mu + \alpha t - \sum_{i:t_i < t} \beta)$ , where  $\mu = 2.5$ ,  $\alpha = 0.05$ ,  $\beta = 0.25$ , and  $\bar{\lambda} = 1e + 2$ .

**STPP Data** The standard diffusion kernel function is defined as:

$$h(t, s, h_i, h_s) = C \frac{\exp(-\beta \Delta t)}{2\pi \sigma_x \sigma_y \Delta t} \exp \left\{ -\frac{1}{2\Delta t} \left( \frac{(\Delta x)^2}{\sigma_x^2} + \frac{(\Delta y)^2}{\sigma_y^2} \right) \right\},$$

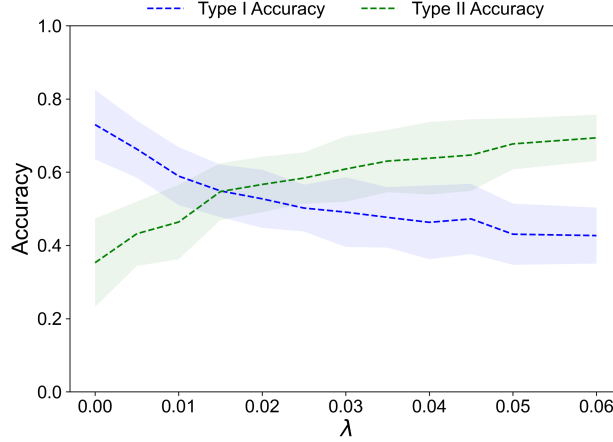


Figure 4: Ablation study of hyper-parameter  $\lambda$  selection.

and the Gaussian diffusion kernel function is defined as:

$$h(t, s, h_i, h_s) = \frac{C \exp(-\beta \Delta t)}{2\pi\sigma_x\sigma_y\Delta t\sqrt{1-\rho^2}} \times \exp\left(-\frac{(\Delta x - \mu_x)^2/\sigma_x^2 + (\Delta y - \mu_y)^2/\sigma_y^2 - (2\rho(x - \mu_x)(\Delta y - \mu_y)) / (\sigma_x\sigma_y)}{2\Delta t(1 - \rho^2)}\right),$$

where  $h_i$  is a historical time,  $h_s$  is a historical location,  $\Delta t = t - h_i$ ,  $\Delta s = s - h_s$ ,  $\Delta x = \Delta s_x$ , and  $\Delta y = \Delta s_y$ . Models are defined with the following parameters:

1.  $STPP_{std}$ :  $\mu = 1$ ,  $\sigma_x = \sigma_y = 0.5$ ,  $\beta = 0.25$ ,  $C = 1$ ,  $\bar{\lambda} = 1e + 4$ .
2.  $STPP_{gau}$ :  $\mu = 2.5$ ,  $\mu_x = \mu_y = 0.1$ ,  $\sigma_x = \sigma_y = 1$ ,  $\rho = 0$ ,  $\beta = 1$ ,  $C = 1$ ,  $\bar{\lambda} = 1e + 4$ .

#### C.4 Testing Procedure Details

**Synthetic Data** For each dataset, we perform 500 iterations of testing between randomly selected pairs of a designated ground-truth model and a fitted model, where  $P^*$  represents the ground-truth model and  $\hat{P}$  represents the fitted model.

**Real Data** Observations are clipped randomly and normalized for each iteration of the experiment with an average length 400. We evaluate the performance of goodness-of-fit test statistics for four cases: (i) North California earthquake data sequences as the ground truth and Japan California earthquake data as the fitted model; (ii) the reverse configuration; (iii) North California data as both observe and fitted model; (iv) Japan data for both observe and fitted model. 200 iterations are performed for each scenario. Similar to experiments on Earthquake dataset, we assess the performance of proposed test in each scenario for 200 iterations.

**Ablation Study** This study examines the impact of hyperparameter choices on the testing performance of the NR statistic. We focus on the smoothing constraints  $\lambda$  for discretization. We set the ground-truth model as SE and the fitted model as SC using the same experimental setup as the TPP synthetic experiment. 200 iterations are performed for each  $e$  and  $\lambda$ . Now the dimension of the hidden states is 2 and  $e$  is 150.

*Smoothing Constraints  $\lambda$ :* As  $\lambda$  increases, we observe a general increase in Type II accuracies and a decrease in Type I accuracies. Type II accuracies begin around 0.35, increase to 0.5 by  $\lambda = 0.01$ , and continue to increase until  $\lambda = 0.015$ , where the trend of increase become slower until it stabilizes beyond  $\lambda = 0.05$ . Type II accuracies start near 0.75 for small  $\lambda$ , drop below 0.6 by  $\lambda = 0.075$  and continue the decreasing trend until it becomes stable beyond  $\lambda = 0.06$ . Type I and II accuracies intersect at  $\lambda = 0.015$ , achieving the best balance between Type I and Type II Accuracy.