XTSFormer: Cross-Temporal-Scale Transformer for Irregular-Time Event Prediction in Clinical Applications

Tingsong Xiao[1], Zelin Xu[1], Wenchong He[1], Zhengkun Xiao[1], Yupu Zhang[1], Zibo Liu[1], Shigang Chen[1], My T. Thai[1], Jiang Bian[2, 3], Parisa Rashidi[4], Zhe Jiang[1]*

[1]Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, USA
[2]Department of Biostatistics and Health Data Science, Indiana University, Indianapolis, IN, USA
[3]Regenstrief Institute, Indianapolis, IN, USA
[4]J. Crayton Pruitt Family Department of Biomedical Engineering, University of Florida, Gainesville, FL, USA
{xiaotingsong, zelin.xu, whe2, xiaoz, y.zhang1, ziboliu, sgchen, mythai, parisa.rashidi, zhe.jiang}@ufl.edu, bianji@iu.edu

Abstract

Adverse clinical events related to unsafe care are among the top ten causes of death in the U.S. Accurate modeling and prediction of clinical events from electronic health records (EHRs) play a crucial role in patient safety enhancement. An example is modeling de facto care pathways that characterize common step-by-step plans for treatment or care. However, clinical event data pose several unique challenges, including the irregularity of time intervals between consecutive events, the existence of cycles, periodicity, multi-scale event interactions, and the high computational costs associated with long event sequences. Existing neural temporal point processes (TPPs) methods do not effectively capture the multi-scale nature of event interactions, which is common in many real-world clinical applications. To address these issues, we propose the cross-temporal-scale transformer (XTSFormer), specifically designed for irregularly timed event data. Our model consists of two vital components: a novel Feature-based Cycle-aware Time Positional Encoding (FCPE) that adeptly captures the cyclical nature of time, and a hierarchical multi-scale temporal attention mechanism, where different temporal scales are determined by a bottom-up clustering approach. Extensive experiments on several real-world EHR datasets show that our XTSFormer outperforms multiple baseline methods.

Code — https://github.com/spatialdatasciencegroup/XTSFormer

Extended version — https://arxiv.org/abs/2402.02258

Introduction

Adverse events related to unsafe care are among the top ten causes of death in the US (Dingley et al. 2008; Weinger et al. 2003). The large volume of electronic health record (EHR) data being collected in hospitals, along with recent advancements in machine learning and artificial intelligence, provides unique opportunities for data-driven and evidence-based clinical decision-making systems (Sutton et al. 2020). One specific example is the learning of de facto clinical care pathways, which are detailed step-by-step plans for the treatment or care of surgical patients. For instance, in multimodal post-surgery pain management, clinical event sequences involving different types of analgesic and anesthetic medications from perioperative EHR data reveal the practical treatment plans adopted in a hospital. Encoding such sequential patterns (care pathways) plays a crucial role in evidence-based interventions and management, improving the quality of care, reducing variability in practice, and optimizing pain management outcomes.

Traditionally, analyzing care pathways has been done manually based on clinicians' knowledge and experience. In recent years, data-driven methods have been developed to automatically extract de facto care pathways from EHR data, including process mining, machine learning, stochastic models, and simulations (Manktelow et al. 2022; Aspland, Gartner, and Harper 2021). Unfortunately, these methods typically focus only on relatively simple care pathways. This paper focuses on learning complex temporal patterns from noisy clinical event data.

The problem presents several technical challenges. First, the irregularity of time intervals between events makes common time series prediction methods insufficient (e.g., standard transformer models (Vaswani et al. 2017)). Second, event sequence patterns often exhibit cycles, periodicity, and multi-scale effects. For example, clinical operational events such as medication administration in operating rooms occur on a fine scale, typically within minutes. Conversely, events that occur pre- or post-operation are on a coarser scale, often spanning hours or days. Figure 1 presents an illustrative example where the event sequence represents a patient's medication administration sequence. In this sequence, medication type A is taken nearly every 12 hours, while medication type B is taken approximately every two days. This scenario exhibits the multi-scale and cyclic patterns commonly observed in healthcare event data. Accurately modeling these complex patterns, especially within extended event sequences, can incur high computational costs.

Existing methods are generally based on the temporal point processes (TPPs), a common framework for modeling asynchronous event sequences in continuous time (Cox and Isham 1980; Schoenberg, Brillinger, and Guttorp 2002). Tra-

*Corresponding author.

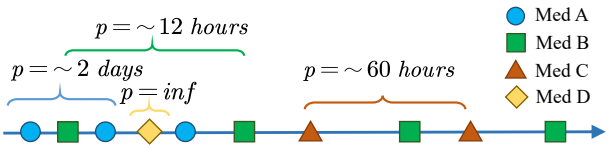


Figure 1: A clinical example of a medication administration sequence for a patient in EHRs.

ditional statistical TPP models (Daley and Vere-Jones 2008) characterize the stochastic nature of event timing but can only capture simple patterns in event occurrences, such as self-excitation (Hawkes 1971a). More recently, deep learning methods, also known as neural TPPs, have gained popularity due to their ability to model complex event dependencies in the intensity function (Eom, Lee, and Choi 2022; Li et al. 2020; Lin et al. 2022; Bae et al. 2022; Zhang, Lipani, and Yilmaz 2021; Wang et al. 2023; Zhou et al. 2023). One category of neural TPPs is based on recurrent neural networks (RNNs), such as the Recurrent Marked Temporal Point Process (RMTPP) (Du et al. 2016), continuous-time LSTM (CT-LSTM) (Mei and Eisner 2017), and intensity Function-based models (Xiao et al. 2017; Omi, Aihara et al. 2019). While LSTM-based approaches address challenges like vanishing gradients, they still face issues such as unresolved long-range dependencies. Transformers-based TPPs, such as Transformer Hawkes Process (THP) (Zuo et al. 2020), Self-Attentive Hawkes Process (SAHP) (Zhang et al. 2020), and (Yang, Mei, and Eisner 2022), can capture the long-range dependency by allowing direct interactions between all events in a sequence. However, these methods do not capture the critical multi-scale patterns within event sequences. While some work has been done on multi-scale transformers, e.g., Scaleformer (Shabani et al. 2023) and Pyraformer (Liu et al. 2022), as well as efficient transformers, e.g., LogTrans (Li et al. 2019), efficient ViT (Dehghani et al. 2023), and Informer (Zhou et al. 2021), and (Hu et al. 2022; Dai et al. 2022) for time series data, these methods assume regular time intervals and are therefore not suitable for predicting irregular time events. Neural ODE-based models can handle irregular time series (Chen et al. 2018; Kidger et al. 2020; Rubanova, Chen, and Duvenaud 2019; Weerakody et al. 2021), but they typically capture random variables as continuous-time functions (e.g., temperature over a time interval) and thus cannot be directly applied to discrete event sequences, where events do not occur at every time point in continuous time. A few works (Jia and Benson 2019; Chen, Amos, and Nickel 2020) have modified neural ODE models for discrete event sequences. However, these methods assume that the event dynamics follow an unknown mathematical system, which may not hold true in real-world applications.

To address these challenges, we propose a novel cross-temporal-scale transformer (XTSFormer) for irregular time event prediction. Our XTSFormer integrates Feature-based Cycle-aware Time Positional Encoding (FCPE) and cross-scale attention within a multi-scale time hierarchy. Specifically, we define the time scale on irregular time event sequences through bottom-up clustering, where events with shorter intervals (at smaller scales) are merged earlier. We designed a cross-scale attention mechanism by selecting a key set as nodes within the same scale level. In summary, this paper makes the following contributions: **1)** We introduce XTSFormer, a neural Temporal Point Process model that incorporates multi-scale temporal interactions of event features, crucial for practical applications in clinical event analysis. **2)** The model introduces two novel components: a feature-based cycle-aware time positional encoding, which captures complex temporal patterns by incorporating both feature and cyclical information, and a cross-temporal-scale attention mechanism, which improves time efficiency compared to standard all-pair attention. **3)** Extensive experiments on two actual EHR datasets demonstrate that our proposed model outperforms several benchmarks.

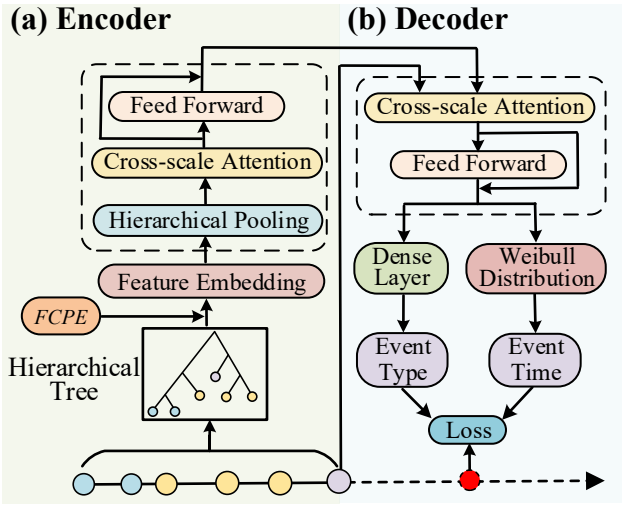


Figure 2: The flowchart of the proposed XTSFormer.

Methodology

Problem definition

Consider a temporal sequence $\mathcal{Q}$ of events denoted as $\langle e_1, ..., e_i, ..., e_L \rangle$, where L represents the length of the sequence. Each event, e_i, can be characterized by a pair (t_i, k_i): t_i signifies the event time, and $k_i \in \{1, 2, ..., K\}$ indicates the event type, with K denoting the total number of type classes. The objective of the event prediction problem is to predict the subsequent event $e_{L+1} = (t_{L+1}, k_{L+1})$. It is important to note that the time of each event, t_i, is irregular, which means that events do not occur at fixed intervals. These event times can exhibit patterns across various temporal scales. For instance, clinical operational events like medication administration may be recorded at minute intervals within an operation room but may be recorded every few hours during the pre-operation or post-operation phases.

Overall model architecture

This section introduces our proposed cross-temporal-scale transformer (XTSFormer) model. As illustrated in Figure 2,

the model consists of two parts: (a) the construction of a hierarchical tree, feature-based cycle-aware time positional encoding, and cross-temporal-scale module (encoder); and (b) event time and type prediction (decoder). Our main idea is to establish a multi-scale time hierarchy and perform cross-scale attention with selective key sets at each scale. Latent features are processed using pooling operations across multiple scale levels. Specifically, starting from the irregular time event sequence, we first conduct a bottom-up clustering to define the multi-scale hierarchy of event points. This is done during the preprocessing phase. Within the framework of our model, the procedure begins with embedding operations, incorporating both our FCPE and semantic feature embedding. The model then progresses through the cross-temporal-scale module, moving from the smallest scale to the largest. At each scale, the model performs hierarchical pooling according to the tree hierarchy, applies cross-scale attention, and concatenates the pooled clusters with those at the subsequent scale. These iterations continue until they reach the root node. This process allows the model to learn complex multi-scale representations within a multi-level hierarchy without sacrificing granularity or specificity. Additionally, this approach enhances computational efficiency by reducing the size of the key set in cross-attention operations, as shown in Figure 2(a).

Feature-based Cycle-aware Time Positional Encoding

Positional encoding is crucial in transformer-based models to capture the relative temporal order of events in TPPs. Existing methods can be classified into fixed (Vaswani et al. 2017) and learned encoding (Kazemi et al. 2019; Xu et al. 2020; Zhang et al. 2020; Xu et al. 2019; Li et al. 2021; Dikeoulias, Amin, and Neumann 2022; Shaw, Uszkoreit, and Vaswani 2018; Raffel et al. 2020), but they fail to learn event cycles based on event features. Research highlights the importance of incorporating semantic features to accurately represent periodic patterns in real-world phenomena (Ke, He, and Liu 2021; Zhang, Lee, and Lee 2019). To effectively capture complex cyclic patterns in irregular time sequences, we introduce a novel Feature-based Cycle-aware Time Positional Encoding (FCPE), which integrates these essential semantic aspects into the encoding of time intervals between events.

Formally, time positional encoding can be described as a function $\mathcal{P}: T \rightarrow \mathbb{R}^{d\times 1}$, mapping the time domain $T \subset \mathcal{R}$ to a d-dimensional vector space. In attention mechanisms, it is the dot product of time positional encodings that carries significance(Xu et al. 2019). Therefore, the relative timespan $|t_a - t_b|$ between events a and b implies crucial temporal information, where t_a and t_b represent the occurrence times of events a and b, respectively. Considering events a and b, we define a temporal kernel $\mathcal{K}: T \times T \rightarrow \mathbb{R}$, such that

$$\mathcal{K}\left(t_a, t_b\right) = \mathcal{P}\left(t_a\right) \cdot \mathcal{P}\left(t_b\right) = \mathcal{F}\left(t_a - t_b\right), \qquad (1)$$

where $\mathcal{F}$ is a location invariant function of the timespan.

The kernel $\mathcal{K}$ defined above satisfies the assumptions of Bochner's Theorem (Veech 1967). Given this, the kernel $\mathcal{K}$ can be represented as in Eq. (2):

$$\mathcal{K}\left(t_a, t_b\right) = \mathcal{F}\left(t_a - t_b\right) = \int_{-\infty}^{\infty} e^{iw(t_a - t_b)} p\left(w\right) dw. \quad (2)$$

Different from (Xu et al. 2020), which uses the Monte Carlo integral (Rahimi and Recht 2007) to approximate the expectation of $\mathcal{F}$, we sample the probability density $p(w_k)$ at several frequencies w_k and learn $p(w_k)$ based on the event feature, where $k = 0, ..., \frac{d}{2} - 1$ (with d as an even integer). The frequencies w_k are learnable parameters, initialized as $\frac{2\pi k}{\frac{d}{2}}$, corresponding to the Discrete Fourier Transform (DFT) of the spectral density function as follows:

$$\begin{aligned}\mathcal{F}\left(t_a - t_b\right) &\approx \sum_{k=1}^{\frac{d}{2}} \mu\left(k\right) e^{iw(t_a - t_b)} \\ &= \sum_{k=1}^{\frac{d}{2}} \mu^k \cos\left(w_k\left(t_a - t_b\right)\right), \end{aligned} \qquad (3)$$

where $\mu(k)$ (representing $p(w_k)$) is the non-negative power spectrum at frequency index k and $\frac{d}{2}$ denotes the number of frequencies. Since w_k is learnable, μ^k is the learned probability density, also referred to as 'intensity' corresponding to frequency w_k.

Thus, following the above conditions and to satisfy Eq. (1) and Eq. (3), we propose the final FCPE function $\mathcal{P}(t_i)$ for time t_i, as shown in Eq. (4),

$$\mathcal{P}(t_i) = \begin{bmatrix} \mu_i^1 \cos\left(w_1 t_i\right) \\ \mu_i^1 \sin\left(w_1 t_i\right) \\ \mu_i^2 \cos\left(w_2 t_i\right) \\ \mu_i^2 \sin\left(w_2 t_i\right) \\ \vdots \\ \mu_i^{\frac{d}{2}} \cos\left(w_{\frac{d}{2}} t_i\right) \\ \mu_i^{\frac{d}{2}} \sin\left(w_{\frac{d}{2}} t_i\right) \end{bmatrix} \in \mathbb{R}^{d\times 1}, \qquad (4)$$

where d is the encoding dimension, w_k is the k-th sampled frequency, and μ_i^k is the learned feature-based intensity corresponding to w_k. Specifically, $\mu_i = [\mu_i^1, \mu_i^2, ..., \mu_i^{\frac{d}{2}}]^T$ can be expressed as $\mu_i = W^\mu \mathbf{k_i}$, where $W^\mu \in \mathbb{R}^{\frac{d}{2}\times K}$ is a learnable parameter matrix, and $\mathbf{k_i} \in \mathbb{R}^{K\times 1}$ is the one-hot encoding of event type k_i.

The advantages of our FCPE are twofold. First, it is based on the premise that any point in time can be represented as a vector derived from a series of sine and cosine functions, capturing the cyclical nature of time with varying intensities and frequencies. This approach is particularly suitable for modeling irregular time intervals. Second, we propose learning the intensities associated with each sampled frequency based on the event's semantic features (e.g., event type) at a particular time. Ideally, event types that occur more frequently will be reflected in higher density values μ^k at higher frequencies w_k. FCPE's translation invariance ensures stability, maintaining performance even when there are shifts in the input features.

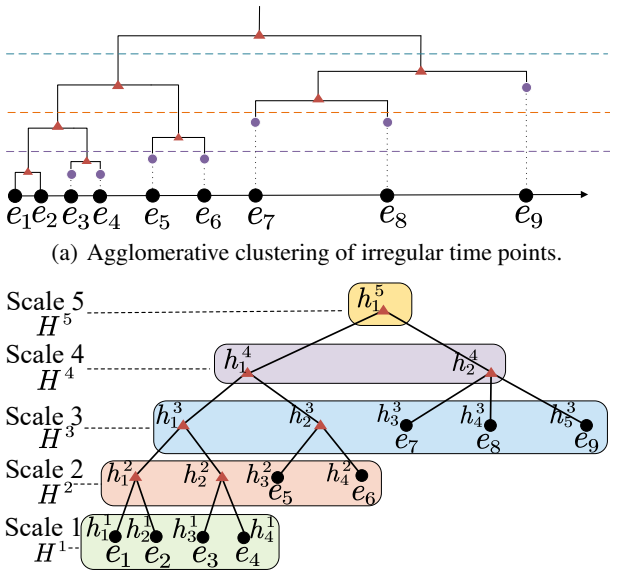
Following the temporal positional encoding $\mathcal{P}(t_i)$, we merge it with a non-temporal feature representation, i.e., $f_i = W^k \mathbf{k}_i + \mathcal{P}(t_i)$, where f_i is the entire embedding i-th event, and $W^k \in \mathbb{R}^{d \times K}$ is a learnable parameter matrix for non-temporal embedding.

Cross-temporal-scale Module on Irregular Event Sequence

The cross-temporal-scale module comprises hierarchical pooling and cross-scale attention, as shown in Figure 2(a). A unique challenge in designing cross-scale attention for irregular time event sequences is the lack of a clear definition of temporal scales. Unlike regular time series data, where temporal scales can be easily defined based on original or down sampled resolutions, irregular time sequences require a different approach. Intuitively, events occurring within short intervals interact at a smaller time scale (e.g., medications administered every few minutes in an operating room), while events with longer intervals operate at a larger time scale (e.g., medication given every few days post-operation). To establish the concept of temporal scales in irregular time sequences, we employ hierarchical clustering.

Hierarchical pooling layer. We define temporal scales for irregular time points using a bottom-up hierarchical clustering approach, such as agglomerative clustering (Day and Edelsbrunner 1984) with the Ward linkage method and Euclidean distance. The agglomerative algorithm starts by treating each time point as an individual cluster, then recursively merges the two closest clusters (measured by minimum, maximum, or centroid distance) until all clusters merge into one. The algorithm's greedy criterion ensures that time points closer together (on smaller scales) are merged earlier. Thus, the temporal scale can be determined based on the cluster merging order within the multi-level hierarchy. Figure 3 illustrates this process with nine events, e_1 to e_9. Figure 3(a) shows the bottom-up clustering, with one merging operation at a time. The levels of the vertical bars indicate the merging order of intermediate clusters. In this example, e_1 and e_2 merge first, followed by e_3 and e_4, then e_5 and e_6. The leftmost clusters merge next, and the process continues until all clusters merge into a root node. The merging order shows that e_7 and e_8 are at a larger time scale than e_1 and e_2, which aligns with our intuition based on the point distribution.

To quantify the time scales of event points, we can vertically slice the merging order of all initial and intermediate clusters into different intervals. Clusters that merge in the s-th vertical interval from the bottom belong to scale s. For example, in Figure 3(a), three thresholds split the merging operations into four intervals. Within each interval, we examine the initial clusters before any merging and the final clusters before the interval ends. For instance, e_1 and e_2 as well as e_3 and e_4 are merged into two internal nodes (red triangles) in the first (bottom) interval, placing them in scale 1. Similarly, e_7, e_8, and e_9 are merged in the third interval, placing them in scale 3. The cumulative merging process is summarized in a hierarchical tree structure, as shown in Figure 3(b), where the **temporal scale** of a tree node is defined



(a) Agglomerative clustering of irregular time points.

(b) Scale hierarchy based on merging order.

Figure 3: An illustration of multi-scale hierarchy on irregular time points by bottom-up clustering.

by its level.

A key consideration is how to choose the slicing thresholds, as they control the granularity of the multiple scales. For finer-grained multi-scale levels, more thresholds (intervals or tree levels) are needed. In the extreme, the number of levels could equal the number of event points. For instance, the temporal scale for medication events in the operating room (intra-operation) might be on the order of minutes, while pre-operation and post-operation events span several hours or days. Using such domain knowledge, we can set slicing thresholds like 5 minutes, 30 minutes, 1 hour, 8 hours, and 24 hours. This approach helps create a multi-scale temporal hierarchy with varying levels of granularity.

In practice, choosing scales based solely on time intervals may be inefficient. To better control the number of points (leaf or internal) at each scale level, slicing thresholds can be configured based on the number of merging operations. For example, the multi-scale hierarchy in Figure 3(b) can be configured with 2, 2, 3, and 1 merging operations at each respective level. This approach helps manage the number of intermediate clusters at each scale.

We denote the latent representations at different tree nodes in each scale level s as $H^s = [h_1^s, h_2^s, \ldots, h_{n_s}^s]$, where h_j^s is the j-th node in scale s, and n_s is the number of nodes in scale s. Initially, for a leaf node e_j, $h_j^s = f_j$ (the raw embedding). In Figure 3(b), there are four node representations at the 1st scale, four at the 2nd scale, and so on.

To aggregate features across different scale levels, we conduct an average pooling operation based on the tree hierarchy, followed by concatenation of the pooled clusters with the clusters in the next scale.

Cross-scale attention layer. We now introduce our attention operation in the multi-scale time hierarchy. In the com-

mon all-pair attention, each time point (query) computes attention weights for all other points (keys). In our cross-scale temporal attention, for each tree node (query), we only do temporal attention on a selective key set, i.e., nodes in the same scale level. The cross-attention operation is expressed in Eq. (5), where $\tilde{h_j^s}$ is the representation of h_j^s after cross-attention, $\boldsymbol{q}_j^s$ is the query vector for jth node at scale s, $\boldsymbol{k}_l^s$ is the key vector for the l-th node, $\boldsymbol{v}_l$ is the value vector, $\mathcal{N}_j^s$ is the selective key set of h_j^s, and D_K is the dimension of key and query vectors as a normalizing term. Consider the example in Figure 3(b). The key set for e_6 (h_4^2) has four nodes (h_1^2, h_2^2, h_3^2, and h_4^2), including itself. This reduces the total number of keys from 9 to 4.

$$\tilde{h_j^s} = \sum_{l \in \mathcal{N}_j^s} \frac{\exp(\boldsymbol{q}_j^s \boldsymbol{k}_l^{sT} / \sqrt{D_K}) \boldsymbol{v}_l}{\sum_{l \in \mathcal{N}_j^s} \exp(\boldsymbol{q}_j^s \boldsymbol{k}_l^{sT} / \sqrt{D_K})}. \tag{5}$$

Time cost analysis: Assume the number of input temporal points is $\hat{L}$, batch size is B, number of heads is $\hat{h}$, and hidden dimension is $\hat{d}$. In our attention computation, the query matrix dimensions are represented by $Q \in \mathbb{R}^{B \times \hat{h} \times \hat{L} \times \hat{d}}$. For each query point $\mathcal{Q}_i$ (where $1 \leq i \leq \hat{L}$), its selective key set size is M (depending on the threshold in each level). Thus the key matrix results from concatenating key sets of all query points, yielding a dimension of $\mathbb{R}^{B \times \hat{h} \times \hat{L} \times M \times \hat{d}}$. Consequently, the Flops of the attention computation with our approach is $B \cdot \hat{h} \cdot \hat{L} \cdot M \cdot \hat{d}$, with the key set size M scaling $O(log\hat{L})$. This results in significantly more efficient attention computation Flops, specifically $B \cdot \hat{h} \cdot \hat{L} \cdot \log \hat{L} \cdot \hat{d}$, compared to the vanilla transformer computation of $B \cdot \hat{h} \cdot \hat{L}^2 \cdot \hat{d}$.

Decoder and Loss Function

Our decoder comprises two parts: predicting event type and event time, as shown in Figure 2(b). First, we apply cross-scale attention at the topmost (largest) scale, using the last element as the query and the others as keys to capture the temporal and sequential nature of the upcoming event. Following this, we obtain H_L, the comprehensive latent representation of the entire past event sequence, by applying a dense layer to H^s. Since H^s incorporates features from multiple scales, it effectively identifies the temporal patterns of the forthcoming event.

Event type prediction The prediction of the next event type, based on the latent embedding H_L of past events, is achieved through a dense transformation layer followed by a softmax function. This process generates the predicted probability distribution of the event type. We calculate the cross-entropy loss $\mathcal{L}_p$ using the true event type labels y_i and the predicted probability distribution of event type P_i , *i.e.,* $\mathcal{L}_p = -\sum_i y_i \log(P_i)$, thus optimizing the model for accurate event type prediction.

Event time prediction For predicting event time, we add another dense layer on top of H_L to learn the distribution parameters of the temporal point process, specifically the scale parameter λ and shape parameter γ. The proposed framework can use the Weibull distribution (Rinne 2008) to model the intensity function. The exponential distribution, a specific case of the Weibull distribution with $\gamma = 1$, has a constant intensity function suggesting events occur with a uniform likelihood, irrespective of past occurrences. This characteristic makes it less suitable for scenarios where historical events are influential. Conversely, the Weibull distribution, with its variable hazard function that can be increasing, decreasing, or constant, offers a flexible approach to modeling how past events impact future probabilities. The comparative effectiveness of these two distributions as intensity functions is explored in the experimental section of our study.

We use the negative log-likelihood (NLL) of event time as the loss function for event time prediction:

$$\mathcal{L}_t = -\log P\left(t'; \lambda, \gamma\right) = -\log\left(\frac{\gamma}{\lambda}\left(\frac{t'}{\lambda}\right)^{\gamma-1} e^{-\left(\frac{t'}{\lambda}\right)^{\gamma}}\right), \tag{6}$$

where t' is the label time. The final loss is $\mathcal{L} = (1-\alpha)\mathcal{L}_t + \alpha\mathcal{L}_p$, where α is a hyperparameter for trade-off.

Experimental Evaluation

Goals: The goal of the evaluation section is to compare our proposed XTSFormer with baseline models in neural TPPs in prediction performance for both event time and event type. Additionally, we conducted an ablation study, computational experiments, sensitivity analysis, and an interpretable case study.

Evaluation metrics: For the event type prediction task, we utilized the accuracy and the macro F1-score as evaluation metrics. Meanwhile, for the event time prediction task, the root mean square error (RMSE) and negative log-likelihood (NLL) were chosen as the performance metrics.

Datasets: In the experiments, we used two EHR datasets—Medications and Providers—from our university hospital. The **Medications** dataset includes 5,080 patient encounters, each treated as a sequence detailing medication events across 86 distinct classes, with a total of 355,490 records. The **Providers** dataset, structured similarly to the Medications dataset, includes 56,262 patient encounters, each representing a sequence of interactions with 48 distinct provider classes, totaling 704,496 records. These timelines cover pre-operative, intra-operative, and post-operative periods. More dataset details and evaluation results are in the arXiv version.

Baselines: Our baseline methods include traditional TPP model called Hawkes Processes (HP) (Hawkes 1971b), two RNN-based neural TPP models (RMTPP (Du et al. 2016) and CT-LSTM (Mei and Eisner 2017)), two neural ODE-based models (NJSDE (Jia and Benson 2019) and ODETPP (Chen, Amos, and Nickel 2020)), and five Transformer-based algorithms (SAHP (Zhang et al. 2020), THP (Zuo et al. 2020), and A-NHP (Yang, Mei, and Eisner 2022).

Comparison on Prediction Performance

Table 1 summarizes the accuracy, F1-score, RMSE, and NLL of all evaluated methods on all datasets. It is observed that the traditional HP model exhibits the lowest accuracy in predicting event types. The RNN-based models perform

Methods	Medications				Providers			
	Accuracy (%)	F1-score (%)	RMSE	NLL	Accuracy (%)	F1-score (%)	RMSE	NLL
HP	$21.9_{\pm 1.1}$	$18.1_{\pm 2.1}$	$2.78_{\pm 0.33}$	$3.54_{\pm 0.38}$	$32.1_{\pm 2.5}$	$31.9_{\pm 2.6}$	$5.17_{\pm 1.30}$	$2.19_{\pm 0.13}$
RMTPP	$23.4_{\pm 0.6}$	$20.1_{\pm 1.8}$	$1.87_{\pm 0.77}$	$3.10_{\pm 0.18}$	$35.7_{\pm 2.1}$	$33.2_{\pm 2.7}$	$4.11_{\pm 1.40}$	$2.23_{\pm 0.11}$
CTLSTM	$22.5_{\pm 0.6}$	$19.2_{\pm 1.7}$	$1.61_{\pm 0.41}$	$3.23_{\pm 0.18}$	$34.5_{\pm 1.4}$	$32.5_{\pm 1.9}$	$3.12_{\pm 1.50}$	$1.93_{\pm 0.08}$
NJSDE	$29.5_{\pm 0.4}$	$25.2_{\pm 0.9}$	$1.40_{\pm 0.22}$	$2.33_{\pm 0.19}$	$37.9_{\pm 1.2}$	$34.1_{\pm 1.1}$	$2.95_{\pm 1.17}$	$1.89_{\pm 0.07}$
ODETPP	$24.6_{\pm 0.5}$	$23.1_{\pm 0.9}$	$1.99_{\pm 0.20}$	$2.60_{\pm 0.21}$	$33.4_{\pm 1.5}$	$29.0_{\pm 0.8}$	$3.81_{\pm 1.21}$	$2.33_{\pm 0.08}$
SAHP	$28.4_{\pm 0.9}$	$25.5_{\pm 2.1}$	$1.81_{\pm 0.30}$	$2.44_{\pm 0.21}$	$38.0_{\pm 1.9}$	$\mathbf{37.2}_{\pm 2.1}$	$3.55_{\pm 1.93}$	$2.10_{\pm 0.09}$
THP	$27.1_{\pm 0.7}$	$26.1_{\pm 1.3}$	$1.41_{\pm 0.33}$	$2.49_{\pm 0.19}$	$37.5_{\pm 2.2}$	$33.8_{\pm 1.9}$	$2.84_{\pm 1.48}$	$1.82_{\pm 0.09}$
A-NHP	$30.2_{\pm 0.5}$	$25.5_{\pm 0.8}$	$1.57_{\pm 0.29}$	$2.54_{\pm 0.22}$	$38.9_{\pm 1.5}$	$34.9_{\pm 1.5}$	$2.89_{\pm 1.54}$	$1.83_{\pm 0.11}$
XTSFormer	$\mathbf{33.5}_{\pm 0.8}$	$\mathbf{29.4}_{\pm 1.1}$	$\mathbf{1.12}_{\pm 0.24}$	$\mathbf{2.23}_{\pm 0.20}$	$\mathbf{43.9}_{\pm 1.3}$	$37.2_{\pm 1.5}$	$\mathbf{2.33}_{\pm 1.74}$	$\mathbf{1.75}_{\pm 0.10}$

Table 1: Results (average $\pm$ std) of all methods on Medications and Providers dataset, with the best results in bold.

slightly better than HP in overall accuracy and F1-score. The Transformer-based models are generally more accurate than the RNN-based models. Their overall accuracy is around 4% to 5% higher than RNN-based models. Among transformer models, XTSFormer performs the best, whose overall accuracy is 3% to 5% higher than other transformers. This could be explained by the fact that our model captures the multi-scale temporal interactions among events. For event time prediction, we observe similar trends, except that the RMSE of event time prediction for SAHP is somehow worse than other transformers (close to the RNN-based models). The reason could be that the event sequences in our real-world datasets do not contain self-exciting patterns as assumed in the Hawkes process.

Ablation Study

To evaluate the effectiveness of our proposed model components, we conducted an ablation study on various datasets. The study investigates the impact of FCPE, multi-scale temporal attention, and choice of event time distribution (Exponential or Weibull). Specifically, we compare two kinds of positional encoding (PE), *i.e.,* traditional positional encoding (written as 'base') (Zuo et al. 2020) and our FCPE. Moreover, we compare our model with (w/) and without (w/o) multi-scale parts. Meanwhile, we compare two kinds of distribution, *i.e.,* exponential distribution and Weibull distribution. Table 2 shows the accuracy results of the ablation study. Introducing multi-scale attention further enhances predictive accuracy. The model with multi-scale attention consistently outperforms its counterpart without it. This demonstrates the significance of modeling interactions at different temporal scales, which is crucial for capturing complex event dependencies.

Computational Time Costs

To evaluate our method's efficiency on long event sequences, we conducted computational experiments using a synthetic dataset of 64 sequences, each with 100,000 events. Each event, consisting of a type ($\{0, 1, \ldots, 9\}$) and occurrence time, was generated using Poisson distributions, with higher event types having lower frequencies (e.g., type 0 occurs every 2 hours, type 9 every 10 days). Time intervals were sampled from exponential distributions and perturbed with Gaussian noise ($\sigma = 0.1\times$ mean interval) to simulate real-world irregularities. We set the batch size to 1 and the hidden dimension to 32, progressively increasing the sequence length and recording time costs (in minutes). Figure 4(a) shows that XTSFormer becomes increasingly efficient compared to Vanilla Transformer as sequences grow longer, while Figure 4(b) demonstrates that XTSFormer maintains lower time costs across embedding dimensions, highlighting its scalability for large-scale data with limited computational resources.

PE	MS	Dist	Medications	Providers
base	w/o	Exponential	$25.2_{\pm 0.3}$	$36.7_{\pm 1.3}$
FCPE	w/o	Exponential	$27.8_{\pm 0.2}$	$38.9_{\pm 0.9}$
base	w/	Exponential	$28.3_{\pm 0.5}$	$37.9_{\pm 1.0}$
FCPE	w/	Exponential	$30.9_{\pm 0.6}$	$38.1_{\pm 0.8}$
base	w/o	Weibull	$26.8_{\pm 0.6}$	$37.1_{\pm 0.8}$
FCPE	w/o	Weibull	$28.9_{\pm 0.3}$	$39.6_{\pm 0.6}$
base	w/	Weibull	$29.3_{\pm 0.6}$	$40.2_{\pm 1.3}$
FCPE	w/	Weibull	$33.5_{\pm 0.8}$	$43.9_{\pm 1.3}$

Table 2: Accuracy results (average $\pm$ std) in percentages. PE: positional encoding, MS: Multi-scale, Dist: Distribution.

Sensitivity Analysis

We investigated parameter sensitivity by varying the largest scale $S \in \{1, 3, 5, 7, 9\}$ and report the accuracy results on two datasets in Figure 4(c). Notably, our method displays sensitivity to the largest scale S, which determines the multi-scale intensity. For instance, when $S = 1$, the absence of multiple scales leads to suboptimal performance.

Interpretable Case Study

To visualize the captured event cycles, we conducted an interpretable case study focusing on the learned cyclical intensities μ^j in Eq. (4), which indicates the importance of the frequency w_j for event j. We selected an anonymized patient who underwent cardiac surgery and displayed their medication administration sequence across three days, from

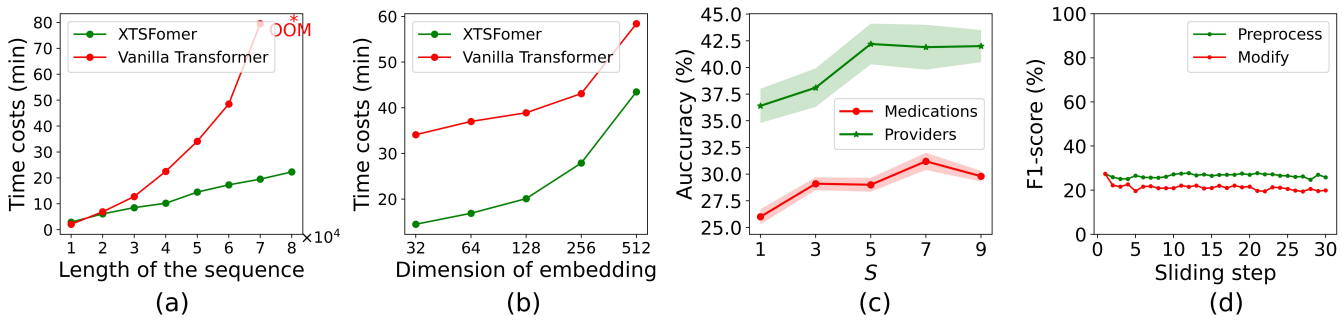
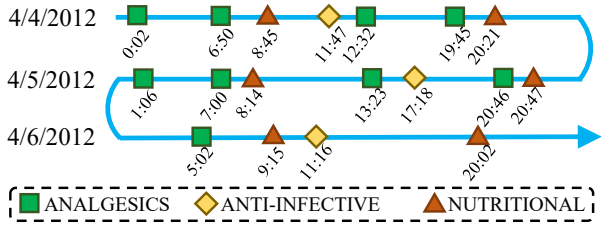


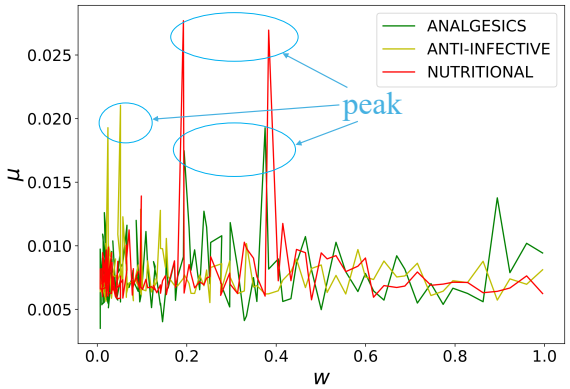
Figure 4: (a) is the time cost on different lengths of sequence (OOM indicates 'out of memory'). (b) is the time cost of different dimensions of embeddings. (c) is the comparison of accuracy on various S scales. (d) is the F1-score of different sliding steps on Medications.

4/4/2012 to 4/6/2012, as shown in Figure 5(a). For clarity, we focused on three specific medication classes: 'ANALGESICS', 'ANTI-INFECTIVE', and 'NUTRITIONAL', excluding other classes from the analysis. We can observe that 'ANALGESICS' were administered approximately every 6 hours, 'NUTRITIONAL' every 11 hours, and 'ANTI-INFECTIVE' every 24 hours. These observed administration cycles align well with the theoretical cycles depicted in Figure 5(b). Specifically, the theoretical cycle for 'ANTI-INFECTIVE' is shown as the inverse of its peak frequency, approximately 23 hours, closely matching the 24-hour administration pattern. For 'ANALGESICS', the theoretical cycle is around 5.3 hours, which is slightly shorter than the observed 6-hour interval but still within a reasonable range given potential variations in clinical practice. The 'NUTRITIONAL' medications exhibited two theoretical cycles at 2.5 hours and 5.5 hours. These shorter cycles may suggest overlapping administration patterns in practice, resulting in the observed 11-hour interval, likely due to the combined effect of multiple dosing schedules or nutritional assessments.

The results show that our feature-based time cycle-aware position encodings learn real-world cyclic patterns in clinical events. In contrast, existing cycle-based position encodings do not learn the varying intensity of different cycle frequencies.



(a) Medication administration sequence for a specific patient over three days.



(b) Learned intensities across frequencies for three medications.

Figure 5: A case study illustrating the learned intensities across frequencies in medication administration sequences.

Limitations

Our model currently predicts one event at a time, which can be inefficient for consecutive event prediction. Predicting multiple consecutive events requires repeating the hierarchical multi-scale clustering steps, adding significant preprocessing time. One strategy to mitigate this overhead is to delay the reconstruction of the hierarchical tree, updating the multi-scale hierarchy incrementally by inserting and deleting event nodes as needed. Preliminary results in Figure 4(d) suggest that this approach somehow impacts prediction accuracy. Further research is needed to develop an end-to-end module that can learn the multi-scale hierarchy without preprocessing.

Conclusion and Future Work

The paper proposed XTSFormer, a neural TPP model with feature-based cycle-aware time positional encoding and cross-scale temporal attention. Time scales are derived from a bottom-up clustering, prioritizing shorter interval events at smaller scales and the cross-scale attention mechanism assigns the key set as nodes at the same scale levels. Extensive experiments on two real-world EHRs validated the model's effectiveness. In future works, we will continue to focus on model interpretability and its generalization to consecutive event prediction.

Ethical Statement

The proposed model has the potential for implementation as a clinical decision-support tool to enhance patient safety. For instance, in post-surgery pain management, the tool can analyze de-facto clinical care pathways within pain medication sequences from electronic health record data. These learned pathways can assist clinicians in identifying anomalous events, preventing errors, and designing treatment plans that better align with patient needs to optimize outcomes. By offering insights into real-world care patterns and supporting standardized, evidence-based practices, this tool contributes to improving healthcare quality and operational efficiency in hospitals.

This research was approved by the Institutional Review Board (IRB) of the authors' institution. Data collection adhered to ethical standards, with all data de-identified to ensure participant confidentiality. To mitigate risks, we implemented a comprehensive data security and privacy protection framework, which includes data anonymization and execution of models and codes on a security-verified computing platform.

Acknowledgments

This material is based upon work supported by the National Science Foundation (NSF) under Grant No. IIS-2147908, IIS-2207072, CNS-1951974, OAC-2152085, OAC-2402946, and OAC-2410884, SCH-2123809, and the National Institutes of Health (NIH) grant R01 LM014027. In addition, we thank Raymond Opoku, Ronald Ison, Jason Petho, and Patrick Tighe from UF Health, as well as Jim Su, for their assistance with data preprocessing.

References

Aspland, E.; Gartner, D.; and Harper, P. 2021. Clinical pathway modelling: a literature review. *Health Systems*, 10(1): 1–23.

Bae, W.; Ahmed, M. O.; Tung, F.; and Oliveira, G. L. 2022. Meta Temporal Point Processes. In *The Eleventh International Conference on Learning Representations*.

Chen, R. T.; Amos, B.; and Nickel, M. 2020. Neural Spatio-Temporal Point Processes. In *International Conference on Learning Representations*.

Chen, R. T.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018. Neural ordinary differential equations. *Advances in neural information processing systems*, 31.

Cox, D. R.; and Isham, V. 1980. *Point processes*, volume 12. CRC Press.

Dai, R.; Das, S.; Kahatapitiya, K.; Ryoo, M. S.; and Brémond, F. 2022. MS-TCT: multi-scale temporal convtransformer for action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20041–20051.

Daley, D. J.; and Vere-Jones, D. 2008. *An introduction to the theory of point processes: volume II: general theory and structure*. Springer.

Day, W. H.; and Edelsbrunner, H. 1984. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of classification*, 1: 7–24.

Dehghani, M.; Djolonga, J.; Mustafa, B.; Padlewski, P.; Heek, J.; Gilmer, J.; Steiner, A. P.; Caron, M.; Geirhos, R.; Alabdulmohsin, I.; et al. 2023. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, 7480–7512. PMLR.

Dikeoulias, I.; Amin, S.; and Neumann, G. 2022. Temporal Knowledge Graph Reasoning with Low-rank and Model-agnostic Representations. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, 111–120.

Dingley, C.; Daugherty, K.; Derieg, M. K.; and Persing, R. 2008. Improving Patient Safety Through Provider Communication Strategy Enhancements. *Advances in Patient Safety: New Directions and Alternative Approaches (Vol. 3: Performance and Tools)*.

Du, N.; Dai, H.; Trivedi, R.; Upadhyay, U.; Gomez-Rodriguez, M.; and Song, L. 2016. 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*, 1555–1564.

Eom, D.; Lee, S.; and Choi, J. 2022. Variational Neural Temporal Point Process. *arXiv preprint arXiv:2202.10585*.

Hawkes, A. G. 1971a. Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 33(3): 438–443.

Hawkes, J. 1971b. On the Hausdorff dimension of the intersection of the range of a stable process with a Borel set. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 19(2): 90–102.

Hu, H.; Dong, S.; Zhao, Y.; Lian, D.; Li, Z.; and Gao, S. 2022. Transrac: Encoding multi-scale temporal correlation with transformers for repetitive action counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19013–19022.

Jia, J.; and Benson, A. R. 2019. Neural jump stochastic differential equations. *Advances in Neural Information Processing Systems*, 32.

Kazemi, S. M.; Goel, R.; Eghbali, S.; Ramanan, J.; Sahota, J.; Thakur, S.; Wu, S.; Smyth, C.; Poupart, P.; and Brubaker, M. 2019. Time2vec: Learning a vector representation of time. *arXiv preprint arXiv:1907.05321*.

Ke, G.; He, D.; and Liu, T.-Y. 2021. Rethinking Positional Encoding in Language Pre-training. In *International Conference on Learning Representations*.

Kidger, P.; Morrill, J.; Foster, J.; and Lyons, T. 2020. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems*, 33: 6696–6707.

Li, S.; Jin, X.; Xuan, Y.; Zhou, X.; Chen, W.; Wang, Y.-X.; and Yan, X. 2019. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems*, 32.

Li, S.; Wang, L.; Zhang, R.; Chang, X.; Liu, X.; Xie, Y.; Qi, Y.; and Song, L. 2020. Temporal logic point processes. In *International Conference on Machine Learning*, 5990–6000. PMLR.

Li, Y.; Si, S.; Li, G.; Hsieh, C.-J.; and Bengio, S. 2021. Learnable fourier features for multi-dimensional spatial positional encoding. *Advances in Neural Information Processing Systems*, 34: 15816–15829.

Lin, H.; Wu, L.; Zhao, G.; Pai, L.; and Li, S. Z. 2022. Exploring Generative Neural Temporal Point Process. *Transactions on Machine Learning Research*.

Liu, S.; Yu, H.; Liao, C.; Li, J.; Lin, W.; Liu, A. X.; and Dustdar, S. 2022. Pyraformer: Low-Complexity Pyramidal Attention for Long-Range Time Series Modeling and Forecasting. In *International Conference on Learning Representations*.

Manktelow, M.; Iftikhar, A.; Bucholc, M.; McCann, M.; and O'Kane, M. 2022. Clinical and operational insights from data-driven care pathway mapping: a systematic review. *BMC medical informatics and decision making*, 22(1): 43.

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.

Omi, T.; Aihara, K.; et al. 2019. Fully neural network based model for general temporal point processes. *Advances in neural information processing systems*, 32.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1): 5485–5551.

Rahimi, A.; and Recht, B. 2007. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20.

Rinne, H. 2008. *The Weibull distribution: a handbook*. CRC press.

Rubanova, Y.; Chen, R. T.; and Duvenaud, D. K. 2019. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32.

Schoenberg, F. P.; Brillinger, D. R.; and Guttorp, P. 2002. Point processes, spatial-temporal. *Encyclopedia of environmetrics*, 3: 1573–1577.

Shabani, A.; Abdi, A.; Meng, L.; and Sylvain, T. 2023. Scaleformer: Iterative Multi-scale Refining Transformers for Time Series Forecasting. In *International Conference on Learning Representations*.

Shaw, P.; Uszkoreit, J.; and Vaswani, A. 2018. Self-Attention with Relative Position Representations. In *Proceedings of NAACL-HLT*, 464–468.

Sutton, R. T.; Pincock, D.; Baumgart, D. C.; Sadowski, D. C.; Fedorak, R. N.; and Kroeker, K. I. 2020. An overview of clinical decision support systems: benefits, risks, and strategies for success. *NPJ digital medicine*, 3(1): 17.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Veech, W. A. 1967. On a theorem of Bochner. *Annals of Mathematics*, 117–137.

Wang, Q.; Cheng, M.; Yuan, S.; and Xu, H. 2023. Hierarchical Contrastive Learning for Temporal Point Processes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8): 10166–10174.

Weerakody, P. B.; Wong, K. W.; Wang, G.; and Ela, W. 2021. A review of irregular time series data handling with gated recurrent neural networks. *Neurocomputing*, 441: 161–178.

Weinger, M. B.; Slagle, J.; Jain, S.; and Ordonez, N. 2003. Retrospective data collection and analytical techniques for patient safety studies. *Journal of biomedical informatics*, 36(1-2): 106–119.

Xiao, S.; Yan, J.; Yang, X.; Zha, H.; and Chu, S. 2017. Modeling the intensity function of point process via recurrent neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Xu, D.; chuanwei ruan; evren korpeoglu; sushant kumar; and kannan achan. 2020. Inductive representation learning on temporal graphs. In *International Conference on Learning Representations (ICLR)*.

Xu, D.; Ruan, C.; Korpeoglu, E.; Kumar, S.; and Achan, K. 2019. Self-attention with functional time representation learning. *Advances in neural information processing systems*, 32.

Yang, C.; Mei, H.; and Eisner, J. 2022. Transformer Embeddings of Irregularly Spaced Events and Their Participants. In *Proceedings of the Tenth International Conference on Learning Representations (ICLR)*.

Zhang, D.; Lee, K.; and Lee, I. 2019. Semantic periodic pattern mining from spatio-temporal trajectories. *Information Sciences*, 502: 164–189.

Zhang, Q.; Lipani, A.; Kirnap, O.; and Yilmaz, E. 2020. Self-attentive Hawkes process. In *International conference on machine learning*, 11183–11193. PMLR.

Zhang, Q.; Lipani, A.; and Yilmaz, E. 2021. Learning neural point processes with latent graphs. In *Proceedings of the Web Conference 2021*, 1495–1505.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 11106–11115.

Zhou, W.-T.; Kang, Z.; Tian, L.; and Su, Y. 2023. Intensity-free Convolutional Temporal Point Process: Incorporating Local and Global Event Contexts. *Information Sciences*, 119318.

Zuo, S.; Jiang, H.; Li, Z.; Zhao, T.; and Zha, H. 2020. Transformer hawkes process. In *International conference on machine learning*, 11692–11702. PMLR.