
RoTHP: ROTARY POSITION EMBEDDING-BASED TRANSFORMER HAWKES PROCESS

Anningzhe Gao*, Shan Dai *
 Shenzhen Research Institute of Big Data
 {gaoanningzhe, shandai}@sribd.cn

ABSTRACT

Temporal Point Processes (TPPs), especially Hawkes Process are commonly used for modeling asynchronous event sequences data such as financial transactions and user behaviors in social networks. Due to the strong fitting ability of neural networks, various neural Temporal Point Processes are proposed, among which the Neural Hawkes Processes based on self-attention such as Transformer Hawkes Process (THP) achieve distinct performance improvement. Although the THP has gained increasing studies, it still suffers from the sequence prediction issue, i.e., training on history sequences and inferencing about the future, which is a prevalent paradigm in realistic sequence analysis tasks. What's more, conventional THP and its variants simply adopt initial sinusoid embedding in transformers, which shows performance sensitivity to temporal change or noise in sequence data analysis by our empirical study. To deal with the problems, we propose a new Rotary Position Embedding-based THP (RoTHP) architecture in this paper. Notably, we show the translation invariance property and sequence prediction flexibility of our RoTHP induced by the relative time embeddings when coupled with Hawkes process theoretically. Furthermore, we demonstrate empirically that our RoTHP can be better generalized in sequence data scenarios with timestamp translations and in sequence prediction tasks.

Keywords Hawkes Process · Transformer · Rotary Position Embedding · Translation Invariance

1 Introduction

Many natural and artificial systems produce a large volume of discrete events occurring in continuous time, for example, the occurrence of crime events, earthquakes, patient visits to hospitals, financial transactions, and user behavior in mobile applications [1]. It is essential to understand and model these complex event dynamics so that accurate analysis, prediction, or intervention can be carried out subsequently depending on the context.

The occurrence of asynchronous event sequences is often modeled by temporal point processes [2]. They are stochastic processes with (marked) events on the continuous time domain. One special but significant type of temporal point processes is the Hawkes process [3]. A considerable amount of studies have used the Hawkes process as a standard tool to model event streams, including: discovering of patterns in social interactions [4], personalized recommendations based on users' temporal behavior [5], construction and inference on network structure [6, 7] and so on. Hawkes processes usually model the occurrence probability of an event with a so-called intensity function. For those events whose occurrence are influenced by history, the intensity function is specified as history-dependent. However, the simplified assumptions implied by Hawkes process for the complicated dynamics limit the models' practicality. As an example, Hawkes process states that all past events should have positive influences on the occurrence of current events. Meanwhile, the lack of various nonlinear operations in traditional Hawkes process also sets an upper limit for Hawkes process' expressive ability.

Thus there emerge some recent efforts in increasing the expressiveness of the intensity function using nonparametric models like kernel methods and splines [8, 9] and neural networks[10], especially recurrent neural networks [5, 11] due to the sequence modeling ability. RNN-based Hawkes processes use a recurrent structure to summarise history

*The two authors are equally contributed

events, either in the fashion of discrete time [5, 12] or continuous-time [11]. This solution not only makes the historical contributions unnecessarily additive but also allows the modeling of complex memory effects such as delays. However, these RNN-based Hawkes processes also inherit the drawbacks of RNN, for instance, it may take a long time for the patient to develop symptoms due to certain sequel, which has obvious long-term characteristics, such as diabetes, cancer and other chronic diseases, while these RNN-based models are hard to reveal the long-term dependency between the distant events in the sequences [13]. Other likelihood-free methods such as using reinforcement learning [14] and generative adversarial networks [15] to help deal with the complex asynchronous event sequences data are also investigated recently.

Recent developments in natural language processing (NLP) have led to an increasing interest in the self-attention mechanism. [16] present self-attention Hawkes process, furthermore, [17] propose transformer Hawkes process based on the attention mechanism and encoder structure in transformer. This model utilizes pure transformer structure without using RNN and CNN, and achieves state-of-the-art performance. Although the self-attention mechanism applied in the Hawkes process performs empirically superior to RNNs in processing sequences data, most of the existing attention-based Hawkes processes architecture [16, 17, 18, 19] still suffer from timestamp noise sensitivity problem and sequence prediction issue. Specifically, common asynchronous events sequences data generated in reality will be naturally accompanied by temporal noise such as timestamp translation or systematic accuracy deviation due to the limited recording capabilities and storage capacities. For example, in large wireless networks, the event sequences are usually logged at a certain frequency by different devices whose time might not be accurately synchronized and whose accuracy varies within a big range. Conventional THP and its variants simply adopt initial sinusoid embedding in transformers and underexplored the position and time encoding problem in neural Hawkes processes, which is, however, crucial for the modeling of asynchronous events. We will next also demonstrate the timestamp noise sensitivity problem empirically. What's more, to increase the generalizability of transformer in long sequences, various position encoding methods[20, 21, 22]are proposed. The versatility and great success of large language model (LLM) in handling complex tasks and its potential need to work with longer texts further stimulated more recent research on position coding [23, 24] regarding the sequence prediction and generation issue. However, existing attention-based Hawkes process models can't be easily used or generalized in sequences prediction tasks. We will next discuss the problem from the position encoding architecture perspective and illustrate it by using their empirical performance in the related prediction tasks. To overcome the aforementioned problems, we construct a new Rotary Position Embedding-based THP (RoTHP) architecture. By adaptively adjusting the position encoding method to accommodate the crucial temporal information and utilizing the interval characteristics of Hawkes process, we further show the translation invariance property and sequence prediction flexibility of our RoTHP induced by the relative time embeddings when coupled with Hawkes process theoretically. Additional simulation studies are also provided to illustrate the superior performance of our RoTHP compared to the existing attention-based Hawkes process models.

In a nutshell, the contributions of the paper are: (i) We propose a RoTHP architecture to deal with the timestamp noise sensitivity problem and sequence prediction issue suffered by existing attention-based Hawkes process models; (ii) We show the translation invariance property and sequence length flexibility of our proposed RoTHP induced by the relative time embeddings when coupled with Hawkes process theoretically; (iii) We demonstrate empirically that our RoTHP can be better generalized in sequence data scenarios with translation or noise in timestamps and sequence prediction tasks.

Temporal point processes: A temporal point process (TPP) is a stochastic process whose realization consists of a sequence of discrete events localized in continuous time, $\mathcal{H} = \{t_i \in \mathbb{R}^+ \mid i \in \mathbb{N}^+, t_i < t_{i+1}\}$ [1]. TPP can be equivalently represented as a counting process $N(t)$, which records the number of events that have happened till time t . We indicate with $\mathcal{H}_t := \{t' \mid t' < t, t_i \in \mathbb{R}^+\}$ the historical sequence of events that happened before t . Given an infinitesimal time window $[t, t + dt]$, the intensity function of a TPP is defined as the probability of the occurrence of an event t' in $[t, t + dt]$ conditioned on the history of events \mathcal{H}_t :

$$\lambda(t)dt := p(t' : t' \in [t, t + dt] \mid \mathcal{H}_t) = \mathbf{E}(dN(t) \mid \mathcal{H}_t),$$

where $\mathbf{E}(dN(t) \mid \mathcal{H}_t)$ denotes the expected number of events in $[t, t + dt]$ based on the history \mathcal{H}_t . Without loss of generality, we assume that two events do not happen simultaneously, i.e., $dN(t) \in \{0, 1\}$.

Based on the intensity function, it is straightforward to derive the probability density function $f(t)$ and the cumulative distribution function $F(t)$ [25]:

$$\begin{aligned} f(t) &= \lambda(t) \exp \left(- \int_{t_{i-1}}^t \lambda(\tau) d\tau \right), \\ F(t) &= 1 - \exp \left(- \int_{t_{i-1}}^t \lambda(\tau) d\tau \right). \end{aligned} \tag{1}$$

A marked TPP allocates a type to each event. We indicate with $\mathcal{S} = \{(t_i, k_i)\}_{i=1}^n$ an event sequence, where the tuple (t_i, k_i) is the i -th event of the sequence \mathcal{S} , t_i is its timestamp, and $k_i \in \mathcal{U}$ is its event type.

Hawkes Process: An Hawkes process models the self-excitation of events of the same type and the mutual excitation of different event types, in an additive way. Hence, the definition of the intensity function is given as:

$$\lambda(t) = \mu + \sum_{t' \in \mathcal{H}_t} \phi(t - t'), \quad (2)$$

where $\mu \geq 0$, named base intensity, is an exogenous component of the intensity function independent of the history, while $\phi(t) > 0$ is an endogenous component dependent on the history. Besides, $\phi(t)$ is a triggering kernel containing the peer influence of different event types. To highlight the peer influence represented by $\phi(t)$, we write $\phi_{u,v}(t)$, which captures the impact of a historical type- v event on a subsequent type- u event [26]. In this example, the occurrence of a past type- v event increases the intensity function $\phi_{u,v}(t - t')$ for $0 < t' < t$.

To learn the parameters of Hawkes processes, it is common to use Maximum Likelihood Estimation (MLE). Other advanced methods such as adversarial learning [15] and reinforcement learning [14] methods have also been proposed. The log-likelihood of an event sequence \mathcal{S} over a time interval $[0, T]$ is given by:

$$\begin{aligned} \mathcal{L} &= \log \left(\prod_{i=1}^n f(t_i) (1 - F(T)) \right) \\ &= \log \left\{ \prod_{i=1}^n \left[\lambda(t_i) \exp \left(- \int_{t_{i-1}}^{t_i} \lambda(\tau) d\tau \right) \right] \exp \left(- \int_{t_n}^T \lambda(\tau) d\tau \right) \right\} \\ &= \sum_{i=1}^n \log \lambda(t_i) - \int_0^T \lambda(\tau) d\tau \end{aligned} \quad (3)$$

2 Position Embedding in self-attention hawkes process and Discussions

2.1 Existing position embeddings in self-attention hawkes process

2.1.1 Sinusoid embedding in THP

THPs and its variants [18, 27] characterize the temporal information by utilizing a temporal encoding pattern analogous to the “vanilla” absolute positional encoding [28] for each node in the cascade. Denote the temporal encoding as below:

$$[\mathbf{x}(t_i)]_j = \begin{cases} \cos \left(t_i / 10000^{\frac{j-1}{D}} \right), & \text{if } j \text{ is odd}, \\ \sin \left(t_i / 10000^{\frac{j}{D}} \right), & \text{if } j \text{ is even}. \end{cases} \quad (4)$$

We utilize cosine and sine function to obtain the temporal encoding for the time-stamp. For each t_i , we can get its corresponding temporal encoding: $\mathbf{x}(t_i) \in \mathbb{R}^D$. D is the model dimensions we determined. And for the event type encoding, it can be obtained through the embedding matrix $\mathbf{K} \in \mathbb{R}^{D \times C}$, for each type of event, we set its corresponding one-hot vector $\mathbf{c}_i \in \mathbb{R}^C$, thus, we can get the embedding \mathbf{Kc}_i , the dimension of \mathbf{Kc}_i is also D .

For the event sequence $s_n = \{(c_i, t_i)\}_{i=1}^{I_n}$, the corresponding temporal encoding and event encoding are \mathbf{X}^T and $(\mathbf{KC}_n)^T$, where $\mathbf{X} = \{\mathbf{x}(t_1), \mathbf{x}(t_2), \dots, \mathbf{x}(t_{I_n})\} \in \mathbb{R}^{D \times I_n}$ and $\mathbf{C}_n = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_{I_n}] \in \mathbb{R}^{C \times I_n}$. It is worth noting that the temporal encoding matrix and event encoding matrix have same dimensions $\mathbb{R}^{I_n \times D}$, and each row of them correspond

2.1.2 Time-shifted positional encoding in Self-Attentive Hawkes Process

Self-Attentive Hawkes Process adopts a time-shifted positional encoding to inject order information to the modeling sequence. Specifically, for an event (v_i, t_i) , the positional encoding is defined as a K dimensional vector such that the k -th dimension of the position embedding is calculated as:

$$Pos_{(v_i, t_i)}^k = \sin (\omega_k \times i + w_k \times t_i), \quad (5)$$

where i is the absolute position of an event in a sequence, and ω_k is the angle frequency of the k -th dimension, which is pre-defined and will not be changed. While w_k is a scaling parameter that converts the timestamp t_i to a phase shift in

the k -th dimension. Multiple sinusoidal functions with different ω_k and w_k are used to generate the multiple position values, the concatenation of which is the new positional encoding. Even and odd dimensions of the position embedding are generated from sin and cos respectively.

2.2 Discussions

2.2.1 Timestamp noise sensitivity

Here, we first show that the relative time difference property shared by the Temporal Hawkes Processes' likelihood, which will cause the position embedding in the existing self-attention Hawkes process to be sensitive to timestamp noise or translations and thus largely affect its modeling performance to sequence data.

Proposition 1. *Let $\mathcal{H} = \{t_i \in \mathbb{R}^+ \mid i \in \mathbb{N}^+, t_i < t_{i+1}\}$ be a Hawkes process with conditional intensity $\lambda^*(t)$ as defined in (3). If we observe all the arrival times over the time period $[t_1, t_n]$, denoted as t_1, \dots, t_n , then the log-likelihood function \mathcal{L} for \mathcal{H}_t is:*

$$\mathcal{L} = \sum_{i=2}^n \log \left[\mu + \sum_{j=1}^{i-1} \phi(t_i - t_j) \right] - \mu(t_n - t_1) - \sum_{j=1}^{n-1} \int_0^{t_n - t_j} \phi(s) ds, \quad (6)$$

which is a function of timestamp differences $t_i - t_j, i, j \in \{1, 2, \dots, n\}$ and $i \neq j$. Namely, condition on $t_i - t_j$, \mathcal{L} is independent of $t_i, i = 1, \dots, n$.

Proof. According to (3), we have the log-likelihood function \mathcal{L} for \mathcal{H}_t over $[t_1, t_n]$:

$$\mathcal{L} = \sum_{i=1}^n \log \lambda(t_i) - \int_{t_1}^{t_n} \lambda(t) dt. \quad (7)$$

Substituting (2), we get

$$\mathcal{L} = \sum_{i=1}^n \log \left[\mu + \sum_{t' \in \mathcal{H}_t} \phi(t_i - t') \right] - \int_{t_1}^{t_n} \left[\mu + \sum_{t' \in \mathcal{H}_t} \phi(t - t') \right] dt, \quad (8)$$

Since we only consider the arrival times over the time period $[t_1, t_n]$, and $\sum_{t' \in \mathcal{H}_t} \phi(t_i - t') = \sum_{j=1}^{i-1} \phi(t_i - t_j)$ for $i = 2, \dots, n$, we have

$$\sum_{i=1}^n \log \left[\mu + \sum_{t' \in \mathcal{H}_t} \phi(t_i - t') \right] = \sum_{i=2}^n \log \left[\mu + \sum_{j=1}^{i-1} \phi(t_i - t_j) \right]. \quad (9)$$

Also, we have

$$\int_{t_1}^{t_n} \left[\mu + \sum_{t' \in \mathcal{H}_t} \phi(t - t') \right] dt = \mu(t_n - t_1) + \int_{t_1}^{t_n} \left[\sum_{t' \in \mathcal{H}_t} \phi(t - t') \right] dt, \quad (10)$$

Note that $[t_1, t_n] = [t_1, t_2] \cup (t_2, t_3] \cup \dots \cup (t_{n-2}, t_{n-1}] \cup (t_{n-1}, t_n]$, and therefore

$$\begin{aligned} \int_{t_1}^{t_n} \sum_{t' \in \mathcal{H}_t} \phi(t - t') dt &= \int_{t_1}^{t_2} \sum_{t' \in \mathcal{H}_t} \phi(t - t') dt + \int_{t_2}^{t_3} \sum_{t' \in \mathcal{H}_t} \phi(t - t') dt \\ &\quad + \dots + \int_{t_{n-1}}^{t_n} \sum_{t' \in \mathcal{H}_t} \phi(t - t') dt. \end{aligned} \quad (11)$$

For the first two terms in the right hand, we get

$$\int_{t_1}^{t_2} \sum_{t' \in \mathcal{H}_t} \phi(t - t') dt = \int_{t_1}^{t_2} \phi(t - t_1) dt,$$

$$\int_{t_2}^{t_3} \phi(t-t') dt = \int_{t_2}^{t_3} \phi(t-t_1) dt + \int_{t_2}^{t_3} \phi(t-t_2) dt, \quad (12)$$

So, we have

$$\int_0^{t_3} \sum_{t' \in \mathcal{H}_t} \phi(t-t') dt = \int_{t_1}^{t_3} \phi(t-t_1) dt + \int_{t_2}^{t_3} \phi(t-t_2) dt. \quad (13)$$

Similarly, we get

$$\begin{aligned} \int_{t_1}^{t_n} \sum_{t' \in \mathcal{H}_t} \phi(t-t') dt &= \int_{t_1}^{t_n} \phi(t-t_1) dt + \int_{t_2}^{t_n} \phi(t-t_2) dt + \dots + \int_{t_{n-1}}^{t_n} \phi(t-t_{n-1}) dt \\ &= \int_0^{t_n-t_1} \phi(s) ds + \int_0^{t_n-t_2} \phi(s) ds + \dots + \int_{t_1}^{t_n-t_{n-1}} \phi(s) ds \\ &= \sum_{j=1}^{n-1} \int_0^{t_n-t_j} \phi(s) ds, \end{aligned} \quad (14)$$

Combining (8),(9),(10),(14), we complete the proof. \square

By the above proposition, it is evident that the log-likelihood estimation for Temporal Hawkes Processes (THPs) is solely reliant on relative positions (timestamps), denoted as $t_i - t_{i-1}$ for $i = 2, 3, \dots, n$. This suggests that if we substitute the temporal sequence with

$$\{t_1 + \sigma, t_2 + \sigma, \dots, t_n + \sigma\}$$

the likelihood remains invariant (refer to Section 3.2 for a detailed discussion). However, the temporal encoding in THP for the temporal sequence after the translation is

$$Pos(t_1, t_2, \dots, t_n) = \begin{cases} \cos\left(\frac{-t_i+\sigma}{10000^{\frac{j-1}{D}}}\right), & \text{if } j \text{ is odd,} \\ \sin\left(\frac{-t_i+\sigma}{10000^{\frac{j}{D}}}\right), & \text{if } j \text{ is even.} \end{cases} \quad (15)$$

It is important to note that $Pos(t_1, t_2, \dots, t_n) \neq Pos(t_1 + \sigma, t_2 + \sigma, \dots, t_n + \sigma)$, which presents an inconsistency with the likelihood, generally being the main part of the loss function for neural Hawkes process modeling and training. Consequently, for the current timestamp encoding issue, we turn our attention to the relative positional encoding method, a technique extensively employed in Natural Language Processing (NLP) tasks. Specifically, we focus on the Rotary Positional Encoding (RoPE). Our aim is to adapt the RoPE for use in the context of neural Hawkes Point Processes.

2.2.2 Sequence prediction issue

The ability to predict future events based on past data is a critical aspect of many fields, including finance, healthcare, social media analytics, and more. In the context of temporal point processes, this predictive capability becomes even more vital.

The importance of future prediction in temporal point processes is underscored by the need for stability and sensitivity to temporal translations. In the Hawkes process, the Transformer Hawkes Process (THP) is sensitive to temporal translations, meaning that small changes in the input sequence can lead to significant changes in the output. This sensitivity can be problematic when trying to predict future events based on past data, as minor variations in the input can lead to inaccurate predictions.

In practical applications, such as financial transactions, the ability to accurately predict future events can have significant implications. For instance, in finance, accurate predictions can inform investment strategies and risk management decisions. In social media analytics, they can help anticipate user behavior and trends. Therefore, we need a stable model architecture under translations to predict future features.

3 Proposed Model

We will next introduce the proposed Position Embedding-based transformer Hawkes process and its properties.

3.1 Rotary Position Embedding-based transformer Hawkes Process

3.1.1 Model architecture

The key idea in our model design is to apply the rotary position embedding method [22] into temporal process. We fix our notations: Let M be the embedding dimension, K be the number of events. Let $\mathcal{S} = \{(t_i, k_i)\}_{i=1}^n$ be a sequence of Hawkes process.

We use \mathbf{X} to denote the matrix representing the one-hot vector corresponding to the event sequence. $\mathbf{X} \in \mathbb{R}^{K \times L}$, the i th column of \mathbf{X} is a one-hot vector where the j th entry is non-zero if and only if $k_i = j$. We train an event embedding matrix W^E , its i th column is the embedding of the i th event. Hence the embedding of the event is given by $\mathbf{Y} = W^E \mathbf{X}$.

The transformer models have no position information. Unlike the absolute positional embedding used in THP, we consider the **Rotary Temporal Positional Embedding** (RoTPE). Let W^Q, W^K, W^V be the linear transformations corresponds to the $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ vectors, i.e.

$$\begin{aligned}\mathbf{Q} &= \mathbf{Y}W^Q, \\ \mathbf{K} &= \mathbf{Y}W^K, \\ \mathbf{V} &= \mathbf{Y}W^V,\end{aligned}\tag{16}$$

where $W^Q, W^K \in \mathbb{R}^{M \times M_Q}, W^V \in \mathbb{R}^{M \times M_V}, M_K$ in the dimension of the query embedding which is an even number and M_V is the dimension of the value vector. Set

$$\theta_j = 10000^{-2(j-1)/d}, \quad j = 1, 2, \dots, d/2,\tag{17}$$

Let

$$\begin{pmatrix} \cos t_i \theta_1 & \sin t_i \theta_1 & \dots & 0 & 0 \\ -\sin t_i \theta_1 & \cos t_i \theta_1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \cos t_i \theta_{d/2} & \sin t_i \theta_{d/2} \\ 0 & 0 & \dots & -\sin t_i \theta_{d/2} & \cos t_i \theta_{d/2} \end{pmatrix}$$

be the rotary matrix R_{t_i} . We can see that $R_{t_i}^T R_{t_j}$ is just $R_{t_j - t_i}$, measuring the relative temporal difference at position i, j . The attention matrix A is given by

$$\begin{aligned}q_i^T R_{t_i}^T R_{t_j} k_j &= y_i^T (W^Q)^T R_{t_i}^T R_{t_j} W^K y_j \\ &= y_i^T (W^Q)^T R_{t_j - t_i} W^K y_j \\ &= q_i^T R_{t_j - t_i} k_j,\end{aligned}\tag{18}$$

which is only related to the time difference. The attention output is given by

$$O = \text{Softmax}\left(\frac{A}{\sqrt{D_K}}\right)V\tag{19}$$

using the normalization. Then we apply a feed-forward neural network to get the hidden representation $\mathbf{h}(t_j)$ for $1 \leq j \leq n$. Hence we can see from the construction that our RoTHP model only depends on the relative position, which is coincide with the loss function: If we modify our marked temporal sequence $\{(t_1, k_1), (t_2, k_2), \dots, (t_n, k_n)\}$ to $\{(t_1 + \sigma, k_1), (t_2 + \sigma, k_2), \dots, (t_n + \sigma, k_n)\}$ with a translation, both the loss function and our model output keep the same.

Since the rotary matrix are orthogonal, $R_i q_i$ and $R_j k_j$ will have the same length with q_i, k_j , this will make our training more stable.

3.1.2 Training

The intensity function of Neural Hawkes process is given by

$$\lambda(t) = \sum_{k=1}^K \lambda_k(t),\tag{20}$$

where λ_k is the intensity function of the k -th event, and

$$\lambda_k = f_k(\alpha_k(t - t_j) + \mathbf{w}_k^T \mathbf{h}(t_j) + b_k), \quad (21)$$

in which t is defined on interval $t \in [t_j, t_{j+1}]$, and $f_k(x) = \beta_k \log(1 + \exp(x/\beta_k))$ is the softplus function with "softness" parameter β_k . Different from the conditional intensity function setting in the THP [17], here we directly adopt the time difference $t - t_j$ without the normalization by t_j . We will illustrate the translation invariance property under this modified form in the next subsection and show its superior performance compared to the original THP by experiments in Section 4.5.

For the prediction of next event type and timestamp, we train two linear layers W^e, W^t

$$\begin{aligned} \hat{k}_{j+1} &= \text{argmax}(\text{Softmax}(W^e \mathbf{h}(t_j))), \\ \hat{t}_{j+1} &= W^t \mathbf{h}(t_j). \end{aligned} \quad (22)$$

For the sequence \mathcal{S} , we define

$$\begin{aligned} \mathcal{L}_{\text{event}}(\mathcal{S}) &= \sum_{j=1}^{n-1} -\log(\text{Softmax}(W^e \mathbf{h}(t_j))_{k_{j+1}}), \\ \mathcal{L}_{\text{time}}(\mathcal{S}) &= \sum_{j=1}^{n-1} ((t_{j+1} - t_j) - (\hat{t}_{j+1} - \hat{t}_j))^2, \end{aligned} \quad (23)$$

where t_j is the true time stamp of the event j . By definition, $\mathcal{L}_{\text{event}}$ measures the accuracy of the event type prediction and $\mathcal{L}_{\text{time}}$ measures the mean square loss of time prediction. Denote the log-likelihood of \mathcal{S} as \mathcal{L} , then the training loss can be defined by

$$\mathcal{L}(\mathcal{S}) = -\mathcal{L} + \beta_1 \mathcal{L}_{\text{event}}(\mathcal{S}) + \beta_2 \mathcal{L}_{\text{time}}(\mathcal{S}), \quad (24)$$

where β_1, β_2 are hyper-parameters to control the range of event and time losses.

3.2 Translation Invariance Property

In this subsection, we will first show the interval characteristics of the Hawkes process and then illustrate the translation invariance property of our proposed RoTHP.

Proposition 2. (*Translation Invariance Property*): We indicate with $\mathcal{S} = \{(t_i, k_i)\}_{i=1}^n$ a Neural Hawkes process where the tuple (t_i, k_i) is the i -th event of the sequence \mathcal{S} , t_i is its timestamp, and $k_i \in \{1, 2, \dots, K\}$ is its event type, with intensity function as given in (20) and (21). Set

$$\mathcal{S}_\sigma = \{(t_1 + \sigma, k_1), (t_2 + \sigma, k_2), \dots, (t_n + \sigma, k_n)\}$$

be the modified sequence \mathcal{S} with a translation. Then we have the following translation invariance property for our proposed RoTHP:

$$\mathcal{L}(\mathcal{S}) = \mathcal{L}(\mathcal{S}_\sigma) \quad (25)$$

Proof. Observe that the translation doesn't affect the value of timestamp differences $t_i - t_j$, so it's enough for us to prove that $\mathcal{L}(\mathcal{S})$ is a function of $t_i - t_j$, where $i, j \in \{1, 2, \dots, n\}$ and $i \neq j$. Namely, condition on $t_i - t_j$, $\mathcal{L}(\mathcal{S})$ is independent of $t_i, i = 1, \dots, n$.

Since $\mathcal{L}(\mathcal{S}) = -\mathcal{L} + \beta_1 \mathcal{L}_{\text{event}}(\mathcal{S}) + \beta_2 \mathcal{L}_{\text{time}}(\mathcal{S})$, next we complete the proof for \mathcal{L} , $\mathcal{L}_{\text{event}}(\mathcal{S})$, and $\mathcal{L}_{\text{time}}(\mathcal{S})$ respectively.

For \mathcal{L} , we have

$$\begin{aligned} \mathcal{L} &= \log \left[\prod_{i=1}^n f(t_i) f(k = k_i | t_i) \right] \\ &= \log \left[\prod_{i=1}^n \lambda(t_i) \exp \left(- \int_{t_{i-1}}^{t_i} \lambda(\tau) d\tau \right) \frac{\lambda_{k_i}(t_i)}{\lambda(t_i)} \right] \\ &= \sum_{i=1}^n \log \lambda_{k_i}(t_i) - \int_{t_1}^{t_n} \lambda(\tau) d\tau. \end{aligned} \quad (26)$$

By construction of RoTHP, the attention is calculated by the inner product of the Q vector and K vector, we have:

$$\begin{aligned} q_i^T R_{t_i}^T R_{t_j} k_j &= y_i^T (W^Q)^T R_{t_i}^T R_{t_j} W^K y_j \\ &= y_i^T (W^Q)^T R_{t_j - t_i} W^K y_j \\ &= q_i^T R_{t_j - t_i} k_j, \end{aligned} \quad (27)$$

which is function of $t_i - t_j$. Since the $\mathbf{h}(t_j)$ involved in \mathcal{L} is generated by feed-forward neural network with the attention output fed through, So the $\mathbf{h}(t_j)$ is also function of $t_i - t_j$, and we only need to consider the term $t - t_j$ involved in \mathcal{L} . Since the intensity function is chosen as in (20) and (21), which we have proved for general $\phi(t - t_j)$ as in the proof of Proposition 1, \mathcal{L} is also a function of $t_i - t_j$.

For $\mathcal{L}_{event}(\mathcal{S})$, the $\mathbf{h}(t_j)$ has been proved to be function of $t_i - t_j$;

Similarly for $\mathcal{L}_{time}(\mathcal{S})$, it involves the terms $t_i - t_j$ and $\hat{t}_{j+1} - \hat{t}_j$. Since $\hat{t}_{j+1} = W^t \mathbf{h}(t_j)$, $\hat{t}_{j+1} - \hat{t}_j$ is also function of $t_i - t_j$.

Thus we complete the proof. \square

This implies that the timestamp translation will not change the loss function under the modeling of our RoTHP.

For the positional encoding in the THP and its variants, the temporal embedding and event embedding is added before the attention mechanism, hence the model output will change a lot after the translation. This implies that RoTHP is more appropriate for the modeling of general Hawkes process.

Robustness to random noise: we consider the noise given by translation and Gaussian noise. When the translation is applied to the temporal sequence, the model should give the same output since we only consider the temporal interval $[t_1, t_n]$. RoTHP satisfies this property. However, for the absolute positional encoding used in THP, change of the temporal sequence by translation will change the positional encoding and hence change the model output. We will show that as the translation σ changes, the behavior of THP is unstable.

We also consider the Gaussian noise acting on the temporal sequence. For a temporal sequence $\{(t_1, k_1), (t_2, k_2), \dots, (t_n, k_n)\}$, we consider the new sequence $\tilde{\mathcal{S}} = \{(t_1 + \sigma_1, k_1), (t_2 + \sigma_2, k_2), \dots, (t_n + \sigma_n, k_n)\}$ where $\sigma_i \sim \mathcal{N}(0, \epsilon)$. We will show later that our proposed RoTHP will also have better performance under the Gaussian noise.

3.3 Sequence Prediction Flexibility

In multiple Natural Language Processing (NLP) tasks, RoPE has shown the extension property, which means that it can deal with longer sequences. Here we consider the case when we want to use the previous information to predict the future ones. For a temporal sequence $\mathcal{S} = \{(t_1, k_1), (t_2, k_2), \dots, (t_n, k_n)\}$, we pick an index $1 < m < n$, and set $\mathcal{S}_{[1:m]} = \{(t_1, k_1), (t_2, k_2), \dots, (t_m, k_m)\}$ as the training sample, and $\mathcal{S}_{[m+1:n]} = (t_{m+1}, k_{m+1}), (t_{m+2}, k_{m+2}), \dots, (t_n, k_n)$ as the testing sample. This is the case when we need to predict the future case, which is really useful in the real world.

Intuitively, the RoTHP is much more stable in the time translations compared to THP and its variants, which may also lead to its better performance in future event prediction tasks for the Hawkes process. We next first give an analysis here. Consider the input of the models for THP and RoTHP. The training input for THP is the sequence $\mathcal{S}_{[1:m]} = \{(t_1, k_1), (t_2, k_2), \dots, (t_m, k_m)\}$ and the testing input is the sequence $\mathcal{S}_{[m+1:n]} = (t_{m+1}, k_{m+1}), (t_{m+2}, k_{m+2}), \dots, (t_n, k_n)$, so these two sequences have totally different temporal distribution since the model hasn't seen the future time stamps during the training process, which is the essential obstacle of THP in this case. However, for RoTHP, by the translation invariant property, we may view the training input and testing input as $\mathcal{S}_{[1:m]} = \{(0, k_1), (t_2 - t_1, k_2), \dots, (t_m - t_1, k_m)\}$ and $\mathcal{S}_{[m+1:n]} = (0, k_{m+1}), (t_{m+2} - t_{m+1}, k_{m+2}), \dots, (t_n - t_{m+1}, k_n)$, the model thus can extend the distribution of the relative timestamps in the training set to the testing set. So RoTHP can lead to a better prediction. We will show the results for the experiments of future prediction using RoTHP and THP in Section 4.6.

4 Experiment

4.1 Dataset

Our experiments were conducted on three distinct datasets. These datasets were specifically chosen due to their long temporal sequences, which are ideal for our study on encoding methods. Furthermore, the Synthetic dataset, generated from a Hawkes process, provides a suitable case for our investigation.

Synthetic This dataset was generated using Python, following the methodology outlined in [16]. It is a product of a Hawkes process, making it a suitable case for our study. Our synthetic dataset admits 5 event types, with average length 60. The minimal length is 20 and the maximal length is 100.

Financial Transactions [5] This dataset comprises stock transaction records from a single trading day. The sequences in this dataset are lengthy, and the events are categorized into two types: "Buy" and "Sell". The average length of the dataset is 2074 and is appropriate to our experiment.

StackOverflow [29] This dataset is a collection of data from the question-answer website, Stackoverflow. We consider the history of user interactions as a temporal sequence. The average length of the sequences in the dataset is 72, with minimum 41 and maximum 736, and it has 22 event types.

Retweet [30] The data set for Retweets compiles a variety of tweet chains. Every chain comprises an original tweet initiated by a user, accompanied by subsequent response tweets. The accompanying information includes the timing of each tweet and the user's identifier. The average length of the sequences is 109, with minimum 50 and maximum 264. The event types are separated into 3 different types depending on number of the followers: "small", "medium" and "large"

Memetrack [29] This dataset comprises references to 42,000 distinct memes over a period of ten months. It encompasses data from more than 1.5 million documents, including blogs and web articles, sourced from over 5,000 websites. Each sequence in the dataset represents the lifespan of a specific meme, with each occurrence or usage of the meme linked to a timestamp and a website ID.

Mimic-II [31] The MIMIC-II medical dataset compiles data from patients' admissions to an ICU over a span of seven years. Each patient's visits are considered distinct sequences, with each sequence event marked by a timestamp and a diagnosis.

4.2 Baselines

In our research, we juxtapose our devised model with four separate models. The primary focus, however, is concentrated on comparing our model and the THP, in an attempt to comprehend the influence of the rotary temporal positional encoding.

The first model, Recurrent Marked Temporal Point Process (RMTPP)([5]), employs a Recurrent Neural Network (RNN) architecture in predicting the temporal occurrence of the subsequent event. The Neural Hawkes process (NHP) ([11]), infuses neural networks with the Hawkes process to phenomenalize the prediction accuracy. In [16] propound the Self-attentive Hawkes Process (SAHP), which harnesses an attention mechanism in the Hawkes process predictions and incorporates further positional encoding in the model fabrication. Lastly, the Transformer Hawkes process (THP) as operatized in [17], applies the transformer structure to the Hawkes process and incorporates absolute positional encoding in the model construction

4.3 Implementation

The architecture of our proposed RoTHP adopts similar transformer construction and chosen hyper-parameters for the Stackoverflow and financial transaction datasets . For the synthetic dataset, we use the construction of Set 1 in [17] and batch size 4, learning rate 1e-4. We will compare the result of THP using the same architecture and hyper-parameters for Synthetic dataset.

4.4 Result

We can see that the RoTHP outperforms all other models for these three datasets for the log-likelihood, see Table 2 We evaluated the performance of various models, including RMTPP, NHP, SAHP, THP, and our proposed model. The evaluation metrics employed were log-likelihood and accuracy.

Log-Likelihood Analysis Our model exhibited a significant advantage on the Financial dataset, achieving the highest log-likelihood value of 1.076, which is a clear indication of its superior fit compared to the other models. This suggests that our model is particularly adept at capturing the complex patterns and relationships inherent in financial data, which is crucial for accurate prediction and analysis in this domain.

On the SO dataset, our model’s log-likelihood was 0.389. This high log-likelihood reflects a more accurate representation of the data’s distribution, indicating that our model can effectively handle the intricacies of social interactions and networks.

In the case of the Synthetic dataset, our model once again demonstrated its robustness by attaining the highest log-likelihood value of 1.01. This strong performance on synthetic data, which is designed to mimic real-world scenarios, underscores the model’s generalizability and its ability to adapt to various data structures.

For the Retweet dataset, we achieved 2.01, the only model larger than 0, also for the Memetrack dataset, we get the highest score of 1.71. These facts reflects our model’s strong ability in the scenario of infomation spread in the social network.

For the Mimic-II dataset, our model outperforms all other models with log-likelihood 0.64. This implies our model’s ability in the short temporal sequences case is still strong.

Table 1: Log-likelihood

Models	Financial	SO	Synthetic	Retweet	MemeTrick	Mimic-II
RMTPP	-3.89	-2.6	-1.33	-5.99	-6.04	-1.35
NHP	-3.6	-2.55	-	-5.6	-6.23	-1.38
SAHP	-	-1.86	0.59	-4.56	-	-0.52
THP	-1.11	-0.039	0.791	-2.04	0.68	0.48
RoTHP	1.076	0.389	1.01	2.01	1.71	0.64

Accuracy and RMSE We consider the accuracy and Root Mean Square Error (RMSE) estimation on the following datasets: Financial, Mimic-II, and SO.

For the accuracy, we can see that the RoTHP model consistently outperforms the other models across all three datasets. It achieves the highest accuracy on the Financial dataset (62.26), the Mimic-II dataset (85.5), and performs comparably on the SO dataset (46.33) with the highest being THP at 46.4.

In terms of RMSE, which is a measure of error where lower values are better, RoTHP again outperforms all other models across all datasets. It achieves the lowest RMSE on the Financial dataset (0.60), the Mimic-II dataset (0.57), and the SO dataset (1.33).

The RoTHP model demonstrates superior performance in both accuracy and RMSE across all datasets when compared to the other models. This suggests that RoTHP is a more reliable and accurate model for these particular tasks. Its consistent performance across different types of data (Financial, Mimic-II, and SO) indicates its robustness and versatility.

Table 2: Accuracy

Models	Financial	Mimic-II	SO
RMTPP	61.95	81.2	45.9
NHP	62.20	83.2	46.3
THP	62.23	84.9	46.4
RoTHP	62.26	85.5	46.33

Table 3: RMSE

Models	Financial	Mimic-II	SO
RMTPP	1.56	6.12	9.78
NHP	1.56	6.13	9.83
SAHP	-	3.89	5.57
THP	0.93	0.82	4.99
RoTHP	0.60	0.57	1.33

Comparison with THP In this study, we assess the performance of both THP and RoTHP based on their log-likelihood. For the training durations, RoTHP consistently outperforms THP, underscoring the benefits of rotary embedding in the Hawkes process. When applied to a financial transaction dataset, RoTHP exhibits a substantial enhancement at the sixth epoch. Furthermore, RoTHP demonstrates a significantly faster convergence rate for financial transactions, further highlighting its superior performance.

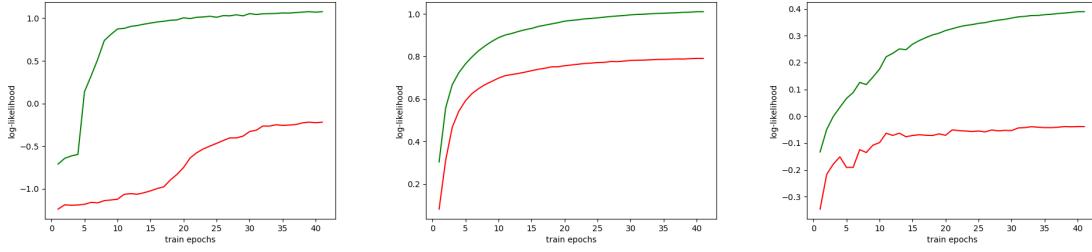


Figure 1: Comparison of log-likelihood between THP and RoTHP. Green lines are RoTHP, red lines are THP. The left figure represent the training process of financial dataset, the middle figure is for the synthetic dataset and the right is for the stackoverflow dataset. We can see in both figures RoTHP outperforms THP

4.5 Robustness study

In this section, we examine the robustness of both THP and RoTHP. As previously discussed, the loss function in the transformer Hawkes process is solely dependent on relative timestamps. We explore two scenarios: Translation and Gaussian noise.

In the Translation scenario, we apply a σ translation to all timestamps in both the training and testing datasets. The model's behavior in financial transactions for σ values ranging from 0 to 10 is presented in Table 4. Remarkably, the log-likelihood of RoTHP remains virtually unchanged, whereas that of THP fluctuates significantly. This observation suggests that RoTHP exhibits greater stability in this context and shows the translation invariant property of our method.

Table 4: Translation log-likelihood of Financial, the table reports the difference between the log-likelihood after the translation and the origin log-likelihood

Models	0	0.2	0.4	0.6	0.8	1	2	5	10
THP	0	0.01	0.018	0.027	0.035	0.044	0.101	0.014	-0.078
RoTHP	0	0	0	0	0	0	0	0	0

In the Gaussian noise scenario, we introduce Gaussian noise $\sigma_i \sim \mathcal{N}(0, \epsilon)$ to each timestamp in every temporal sequence of the training dataset, thereby incorporating Gaussian noise into the training data. We set $\epsilon = 0.01$ for our experiment. The results are displayed in Table 5. We show the difference between the log-likelihood error before and after we add the Gaussian noise on the time stamps. Here, we choose ϵ small because the time gaps between adjacent events are small, and we must choose a noise smaller than the gaps. Otherwise, this will change the order of the events in the sequences. For convenience, we show the ratio of the absolute value of the difference and ϵ to magnify the influence of the Gaussian noise. We pick the Synthetic and SO datasets to see the influence of the Gaussian noise. The reason we pick these two datasets is because we need a long sequence length so that the temporal information will be more important. The Financial dataset has too small time stamp gaps, and the Retweet dataset has integer time stamps, which are not appropriate for this case.

Table 5: Gaussian noise

Models	Synthetic			SO		
	log-likelihood	Accuracy	RMSE	log-likelihood	Accuracy	RMSE
THP	0.906	0	0.104	0.855	0.04	4.379
Ours	0.635	0	0.051	0.234	0.012	1.84

From Table 5, we can see that RoTHP is much stabler than THP in the Gaussian noise case, implying the robustness of the method.

4.6 Predict the future features

In this subsection, we consider the case where we use the previous information to predict the future ones. Intuitively, the THP is sensitive to the translation, and RoTHP is more stable. Hence the RoTHP may perform better than THP. We do the test on financial transaction, synthetic and StackOverflow dataset, and Table 6 shows the result.

Table 6: Future prediction

Models	Financial			Synthetic			SO		
	log-likelihood	Accuracy	RMSE	log-likelihood	Accuracy	RMSE	log-likelihood	Accuracy	RMSE
THP	-1.04	0.6	0.661	0.088	0.38249	0.3405	-0.942	0.4	2.954
Ours	1.24	0.62	0.654	1.08	0.3825	0.3298	0.476	0.403	2.872

The RoTHP model generally performs better than the THP model across all three datasets and metrics. The text also suggests that the THP model may be prone to overfitting, which could explain its poorer performance. Here the translation invariant property of RoTHP plays an important role in this case. By Figure 2, we can see that THP becomes even worse while RoTHP becomes better during the training process. Hence in this case THP meets the overfitting problem.

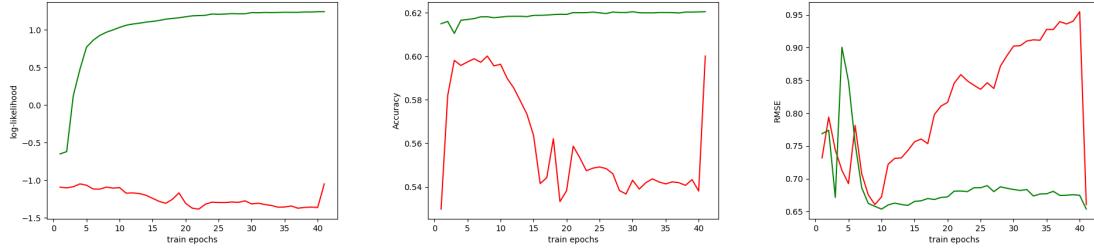


Figure 2: Green lines are RoTHP, red lines are THP. The experiment is for the financial transaction dataset

5 Conclusion

In this paper, we introduce a novel model architecture for the Hawkes process, known as RoTHP, which incorporates the rotary encoding method into the transformer Hawkes process. This results in a model that is both robust and high-performing. We provide an in-depth analysis of why rotary position encoding is more effective for the Hawkes process, examining the loss function and multi-head self-attention mechanism in detail. When tested on three long temporal sequence datasets - Synthetic, financial transaction, and StackOverflow - our model outperformed other models such as RMTPP, THP, NHP, and SAHP. Furthermore, RoTHP demonstrated impressive performance even under conditions of time stamp translation and Gaussian noise.

References

- [1] Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes: volume II: general theory and structure*. Springer Science & Business Media, 2007.
- [2] David Roxbee Cox and Valerie Isham. *Point processes*, volume 12. CRC Press, 1980.
- [3] Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.
- [4] Fangjian Guo, Charles Blundell, Hanna Wallach, and Katherine Heller. The bayesian echo chamber: Modeling social influence via linguistic accommodation. In *Artificial Intelligence and Statistics*, pages 315–323. PMLR, 2015.
- [5] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1555–1564, 2016.
- [6] Shuang-Hong Yang and Hongyuan Zha. Mixture of mutually exciting processes for viral diffusion. In *International Conference on Machine Learning*, pages 1–9. PMLR, 2013.
- [7] Edward Choi, Nan Du, Robert Chen, Le Song, and Jimeng Sun. Constructing disease network and temporal progression model via context-sensitive hawkes process. In *2015 IEEE International Conference on Data Mining*, pages 721–726. IEEE, 2015.

- [8] Ke Zhou, Hongyuan Zha, and Le Song. Learning triggering kernels for multi-dimensional hawkes processes. In *International conference on machine learning*, pages 1301–1309. PMLR, 2013.
- [9] Grace Wahba. *Spline models for observational data*. SIAM, 1990.
- [10] Oleksandr Shchur, Ali Caner Türkmen, Tim Januschowski, and Stephan Günnemann. Neural temporal point processes: A review. *arXiv preprint arXiv:2104.03528*, 2021.
- [11] Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in neural information processing systems*, 30, 2017.
- [12] Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen Chu. Modeling the intensity function of point process via recurrent neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.
- [13] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.
- [14] Shuang Li, Shuai Xiao, Shixiang Zhu, Nan Du, Yao Xie, and Le Song. Learning temporal point processes via reinforcement learning. *Advances in neural information processing systems*, 31, 2018.
- [15] Shuai Xiao, Mehrdad Farajtabar, Xiaojing Ye, Junchi Yan, Le Song, and Hongyuan Zha. Wasserstein learning of deep generative point process models. *Advances in neural information processing systems*, 30, 2017.
- [16] Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive hawkes process. In *International conference on machine learning*, pages 11183–11193. PMLR, 2020.
- [17] Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *International conference on machine learning*, pages 11692–11702. PMLR, 2020.
- [18] Lu-ning Zhang, Jian-wei Liu, Zhi-yan Song, and Xin Zuo. Universal transformer hawkes process with adaptive recursive iteration. *Engineering Applications of Artificial Intelligence*, 105:104416, 2021.
- [19] Liu Yu, Xovee Xu, Goce Trajcevski, and Fan Zhou. Transformer-enhanced hawkes process with decoupling training for information cascade prediction. *Knowledge-Based Systems*, 255:109740, 2022.
- [20] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*, 2020.
- [21] Ofir Press, Noah A Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. *arXiv preprint arXiv:2108.12409*, 2021.
- [22] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- [23] Shouyuan Chen, Sherman Wong, Liangjian Chen, and Yuandong Tian. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023.
- [24] Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023.
- [25] Jakob Gulddahl Rasmussen. Lecture notes: Temporal point processes and the conditional intensity function. *arXiv preprint arXiv:1806.00221*, 2018.
- [26] Mehrdad Farajtabar, Nan Du, Manuel Gomez Rodriguez, Isabel Valera, Hongyuan Zha, and Le Song. Shaping social activity by incentivizing users. *Advances in neural information processing systems*, 27, 2014.
- [27] Lu-ning Zhang, Jian-wei Liu, Zhi-yan Song, and Xin Zuo. Temporal attention augmented transformer hawkes process. *Neural Computing and Applications*, pages 1–15, 2022.
- [28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [29] Jure Leskovec and Rok Sosič. Snap: A general-purpose network analysis and graph-mining library. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(1):1–20, 2016.
- [30] Qingyuan Zhao, Murat A Erdogdu, Hera Y He, Anand Rajaraman, and Jure Leskovec. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1513–1522, 2015.
- [31] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.