# Mitigating Performance Saturation in Neural Marked Point Processes: Architectures and Loss Functions

Tianbo Li*[†][‡]
Sea AI Lab
Singapore
litb@sea.com

Tianze Luo*
Nanyang Technological University
Singapore
tianze001@e.ntu.edu.sg

Yiping Ke
Nanyang Technological University
Singapore
ypke@ntu.edu.sg

Sinno Jialin Pan
Nanyang Technological University
Singapore
sinnopan@ntu.edu.sg

## ABSTRACT

Attributed event sequences are commonly encountered in practice. A recent research line focuses on incorporating neural networks with the statistical model—marked point processes, which is the conventional tool for dealing with attributed event sequences. Neural marked point processes possess good interpretability of probabilistic models as well as the representational power of neural networks. However, we find that performance of neural marked point processes is not always increasing as the network architecture becomes more complicated and larger, which is what we call the *performance saturation* phenomenon. This is due to the fact that the generalization error of neural marked point processes is determined by both the network representational ability and the model specification at the same time. Therefore we can draw two major conclusions: first, simple network structures can perform no worse than complicated ones for some cases; second, using a proper probabilistic assumption is as equally, if not more, important as improving the complexity of the network. Based on this observation, we propose a simple graph-based network structure called GCHP, which utilizes only graph convolutional layers, thus it can be easily accelerated by the parallel mechanism. We directly consider the distribution of interarrival times instead of imposing a specific assumption on the conditional intensity function, and propose to use a likelihood ratio loss with a moment matching mechanism for optimization and model selection. Experimental results show that GCHP can significantly reduce training time and the likelihood ratio loss with interarrival time probability assumptions can greatly improve the model performance. [1]

## CCS CONCEPTS

• **Information systems → Data stream mining**; • **Mathematics of computing →** *Stochastic processes*; • **Computing methodologies →** Neural networks.

## KEYWORDS

Neural point processes, Hawkes processes, event sequential analysis

**ACM Reference Format:**
Tianbo Li, Tianze Luo, Yiping Ke, and Sinno Jialin Pan. 2021. Mitigating Performance Saturation in Neural Marked Point Processes: Architectures and Loss Functions. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3447548.3467436

## 1 INTRODUCTION

*Attributed event sequences* are one of the most commonly encountered data objects in real-world applications. An attributed event sequence contains not only the timestamps of asynchronously generated events but also event features/attributes.[2] It is naturally generated from databases and event logfiles, and has been applied to various application scenarios and disciplines including financial transactions [2], natural language processing [25], and spatial dependence among trees [22], etc. Existing methods that deal with attributed event sequences are usually based on the statistical tool—marked point processes, which have been used for recommendation [8], network inference [15], fake news mitigation [9], and many other tasks [23, 27, 33].

To endow the probabilistic methods with better flexibility and effectiveness, some researchers [6, 18, 21, 29] have explored the idea of incorporating marked point processes with neural networks, especially recurrent neural networks (RNNs), as they are applicable to the sequential nature. The recurrent architecture of these models, however, makes it difficult to be accelerated by parallel mechanisms.

*Both authors contributed equally to this research.
[†]Corresponding author.
[‡]This work was done when he was a student at Nanyang Technological University, Singapore.

[1] The source code is available at https://github.com/ltz0120/Graph-Convolutional-Hawkes-Processes-GCHP.
[2] In this paper, we interchangeably use these two terms. Sometimes they are also referred to as "marks" in the literature of stochastic processes.

*KDD '21, August 14–18, 2021, Virtual Event, Singapore.*

ACM ISBN 978-1-4503-8332-5/21/08…$15.00
https://doi.org/10.1145/3447548.3467436

As an alternative, the attention mechanism has been applied to the learning of point processes in recent studies [34, 38]. In addition to the recurrent and attentive network architectures, a graph-based neural point process model [26] has been applied to consider the geometry structure of Hawkes processes. Despite these architectures are believed to be more effective and have better representational power, it is still not clear whether more complex network architectures will do better for learning attributed event sequences. The other drawback of the current neural marked point process models is that the aforementioned models are often designated to particular forms of the conditional intensity function. For example, the RMTPP model [6] utilizes an exponential form, whereas NHPP [18] utilizes a sigmoid function. Despite the computational convenience that these assumptions bring about, the representational capability of neural networks are also restricted. Moreover, existing approaches often involve the Monte Carlo integration [3] for predicting the next event, which is rather time-consuming instead.

In this paper, we are trying to answer one of the most fundamental questions regarding neural point processes: how can we improve the model performance? Can we get a better model by making the network architecture more complicated? A short answer is NO. Neural point process models often exhibit what we call the *performance saturation* phenomenon — performance of the model increases and then stagnates at a certain point, as we increase the number of the network. So what causes the performance saturation of the neural marked point processes? The reason is that, the generalization error of neural point processes can be decomposed into network estimation error, model specification error (inductive bias) and some irreducible error caused by the randomness of the ground truth model. As we utilize more parameters in the network, which is equivalent to increase the dimension of network function space, the network estimation error can be reduced, but not the model specification error. This tells us an important fact regarding neural point processes: defining a good probabilistic structure of the point process, sometimes is more important than chasing after fancy network architectures.

Based on this observation, we try to improve the neural marked point processes in two ways: architectures and loss functions. We compare architectures among recurrent, attentive, graph-based ones and the combinations among them. We propose a novel temporal-graph-based neural marked point processes, called graph convolutional Hawkes process (GCHP), which can achieve similar performance as the start-of-the-arts, but takes much less training time. The model falls into the category of nonlinear marked Hawkes process with multiplicative kernels. We also apply a convenient likelihood ratio loss based on moment matching approach, which can avoid the high time complexity that the Monte Carlo integration in the traditional methods brings about. Instead of using the conditional intensity function, we directly considers the conditional distributions of the interarrival times, and link up the loss functions with conditional intensity functions and conditional distributions of the interarrival times.

The **contributions** of this paper are summarized as below.

- **We introduce the performance saturation phenomenon in neural point processes for the first time.** In this paper, we describe the *performance saturation* phenomenon that performance of neural network stop increasing as the neural network gets more complicated and has more parameters, which is different than the *double descent* [20] phenomenon as in classical neural networks. We provide an explanation based the generalization error decomposition.
- **We propose a simple graph-based network architecture for neural point process.** We propose a simple method, called GCHP, which is based on the temporal graph of the event history and can be easily incorporated by graph convolutional networks. This method helps not only significantly reduce the training time, but also improve the performance of existing methods.
- **We present an easy-to-use and effective loss function based on likelihood ratio test.** We directly consider the distribution of interarrival times instead of imposing a specific assumption on the conditional intensity function. We propose to use a likelihood ratio type loss function which take into account both the model complexity and the likelihood of observations. We link up the equivalence among loss functions and intensity functions. Experimental results show that the our method can significantly improve the prediction accuracy.

## 2 RELATED WORK

Existing works on marked point processes can be classified into non-neural and neural-based ones.

**Non-neural marked point processes.** Models related to non-neural marked point processes are usually from the perspective of traditional statistical learning [12, 14, 28, 31, 36]. These works carry out improvements in terms of incorporating statistical techniques such as regularization and non-parametric methods. Zhou et al. [36] introduces nuclear and rank norm to the likelihood of multi-dimensional Hawkes processes, so that the sparse and low-rank pattern of the infectivity matrix can be recovered. Xu et al. [31] imposes a more general assumption of the decay kernels, which uses a series of basis functions such as exponential and Gaussian. Wang et al. [28] modulates the intensity function by an additional nonlinear link function, in order to capture the nonlinear effects. Another major development of marked Hawkes processes is Bayesian Hawkes processes [13, 25, 33]. These models are usually fused with mixture models, especially in the context of natural language processing. Representative works are the Dirichlet-Hawkes Processes proposed in [7, 32], which take into account both textual contents and temporal information. Both models assume a Dirichlet prior distribution for the parameters, and therefore they are applicable to clustering tasks. More recently, some works [13, 25] propose hierarchical Bayesian Hawkes processes to deal with continuous features associated with events. One common drawback in Bayesian Hawkes processes is the poor scalability. The inference process for a Bayesian model is relatively time-consuming, and thus neural-based methods are getting more and more attractive.

**Neural-based marked point processes.** An active research line is to learn point processes with neural networks. The RMTPP [6] model views the intensity function as a nonlinear function of the history, and uses a recurrent neural network to learn a representation of influences from the event history. Experimental results show that the model has better performance in both model fitting and prediction than traditional methods. [18] proposes a neural

**Table 1: Summary of some neural marked point processes.**

| Model | Network architecture | Intensity function |
|---|---|---|
| RMTPP [6] | Recurrent | Exponential |
| IRNN [29] | Recurrent | – |
| Neural HP [18] | Recurrent | Sigmoid |
| FulNN [21] | Recurrent | Softplus |
| GeoHP [26] | Graph-based, recurrent | Linear |
| Transformer HP [38] | Attentive | Softplus, exponential |
| Self-attentive HP [34] | Attentive | Softplus |

Hawkes process model named NHPP, which considers the interactions between events. The IRNN model [29] uses an intensity recurrent architecture that synergistically models time series and event sequence, making it able to capture both background and history effect. All of the above methods define respective intensity functions to be a specific parametric form. The fully neural point process model (FulNN) [21] relaxes the assumption of a parametric intensity function, and uses a fully connected neural network to output the cumulative hazard functions, which avoids defining a specific form of the intensity function. However, the model fails to consider the features associated with each event. The geometric Hawkes process (GeoHP) model [26] treats the parameter estimation of a vanilla Hawkes process as a matrix completion problem, and uses graph convolutional recurrent neural networks [19] to solve it. Note that though GCN layers are used in GeoHP, they are used for learning the user/item embeddings. The main architecture of GeoHP is still RNN. Besides, the intensity function of the model is linear with even fewer parameters than the vanilla Hawkes process. More recently, some studies [35, 38] investigate the incorporation of attention mechanism for Hawkes processes.

Existing neural marked point processes can be categorized into three types in terms of the network architecture: recurrent, attentive and graph-based, as shown in Figure 1. In Table 1, we summarize some important models in terms of the network architecture and the intensity function.

## 3 PRELIMINARIES

In this section, we briefly introduce two main preliminary techniques of our model.

**Marked point processes.** Marked point processes (MPP) [4] are commonly used for modeling the temporal dynamics of attributed event sequences. A marked Hawkes process is a point process $\mathcal{N}(\cdot,\cdot)$ on $\mathcal{T}\times\mathcal{M}$, where $\mathcal{T}=[0,T]$ is the observation window and $\mathcal{M}$ the mark (feature) space. It is worth noting that if $\mathcal{M}$ is finite discrete, $\mathcal{N}$ is degenerated to a multi-dimensional Hawkes processes. In this paper, we assume that $\mathcal{M}$ can be continuous, i.e., $\mathcal{M}=\mathbb{R}^p$. The continuous assumption is more general and common in real

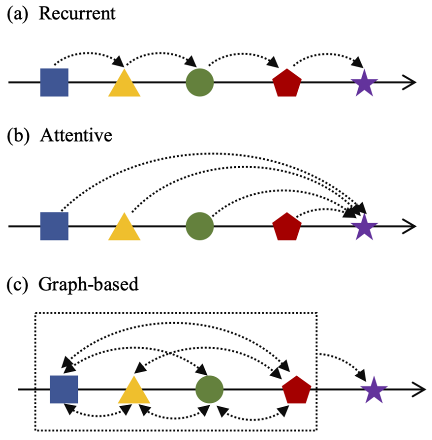

**Figure 1: An illustration of the three message passing methods for neural point processes: (a) recurrent, (b) attentive, (c) graph-based network structure. The dashed lines with arrow heads denoted the direction of message passing.**

world. *Spatio-temporal Hawkes processes* [24] are a good example of continuous mark space, as the location of a point (latitude and longitude) is in $\mathbb{R}^2$. Given the *nature history* $\mathcal{H}_{t-}$, which is defined by the $\sigma$-algebra: $\mathcal{H}_{t-}=\sigma\{\mathcal{N}(s,\mathcal{M};\omega):0<s<t\}$, where $\omega$ is a sampled path, the *conditional intensity function* of a marked point process is defined by

$$\lambda(t,m|\mathcal{H}_{t-})=\lim_{\Delta_t,\Delta_m\to 0}\frac{\mathbb{E}[\mathcal{N}\left([t,t+\Delta_t)\times B(m,\Delta_m)\right)|\mathcal{H}_{t-}]}{\Delta_t|B(m,\Delta_m)|},$$

where $|B(m,\Delta_m)|$ is the Lebesgue measure of the ball $B(m,\Delta_m)$ with radius $\Delta_m$. It can be decomposed by [4]

$$\lambda(t,m|\mathcal{H}_{t-})=\lambda_g(t|\mathcal{H}_{t-})p(m|t,\mathcal{H}_{t-}), \tag{1}$$

where $\lambda_g(t|\mathcal{H}_{t-})$ is the marginal intensity w.r.t. time, often referred to as the *ground intensity*. The conditional marked and ground intensity function is often abbreviated to $\lambda^*(t,m)$ and $\lambda_g^*(t)$, respectively ,where the notation $*$ represents the intensity function is conditioned on the history $\mathcal{H}_{t-}$. $p(m|t,\mathcal{H}_{t-})$ is the conditional mark density which refers to the distribution to be anticipated at the end of a time interval, not immediately after the next interval has begun. Given a realization of attributed event sequence $\{(t_i,m_i):i=1,\dots,N\}$, the log-likelihood function is given by [4]

$$\ell=\sum_{i=1}^{N}\log\lambda_g(t_i|\mathcal{H}_{t_i-})-\int_{t_{i-1}}^{t_i}\lambda_g(t|\mathcal{H}_{t_i-})dt+\log p(m_i|t_i,\mathcal{H}_{t_i-}).$$

**Graph convolutional networks.** In recent years, GCNs [5, 11] have obtained great success as an efficient and effective model for graph-structured data. Given an input graph with an adjacency matrix $\mathcal{A}$ and a feature matrix $\boldsymbol{X}$, GCNs encode both the topological information and the node attributes and produce an output with node embeddings. The most representative model applies the new layer-wise propagation rule [11]:

$$H^{(l+1)}=\sigma(\tilde{A}H^{(l)}W^{(l)}),$$

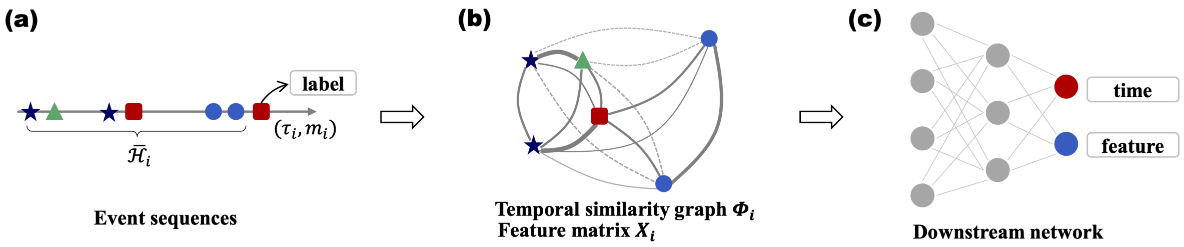

**Figure 2: An illustration of the modeling flow of graph convolutional Hawkes processes (GCHP). (a)→(b): transform into the attributed graph $(\Phi_i, X_i)$. (b)→(c): input the data into the GCHP model.**

where $\tilde{A}$ is a normalized adjacency matrix, $H^{(l)}$ is the output of the $l$-th layer, $W^{(l)}$ is a layer-specific trainable weight matrix, and $\sigma$ is a non-linear activation function.

## 4 A SIMPLE TEMPORAL-GRAPH-BASED ARCHITECTURE FOR NEURAL MARKED POINT PROCESSES

In this section, we present a simple temporal-graph-based architecture for learning marked point processes. This method, which just utilizes graph convolutional layers, is easy and convenient to implement, and achieves as good performance as existing methods with much less training time. It can be viewed as a special case of a nonlinear marked Hawkes process with multiplicative kernel, therefore, we refer to our model as graph convolutional Hawkes processes (GCHP).

**The model.** Figure 2 illustrates the overarching modeling process of our GCHP method. We first scan the input attributed event sequence. For each event $(t_i, m_i)$, we obtain its trimmed history $\tilde{\mathcal{H}}_i$ with a preset number of prior events. We then transform the trimmed history $\tilde{\mathcal{H}}_i$ into a temporal similarity graph $\Phi_i$ and a feature matrix $\boldsymbol{X}_i$, which are then passed to graph convolutional layers. Unlike [6, 18] that assume a specific form of the intensity function, we use a moment matching strategy to approximate the intensity. To be specific, our GCHP model with two graph convolutional layers can be written as

$$\begin{cases} \widehat{\tau}_i, \quad \widehat{m}_i = F(H_i^{(2)}, \tilde{\Phi}_i), \\ H_i^{(2)} = \sigma(\tilde{\Phi}_i H_i^{(1)} W^{(1)}), \\ H_i^{(1)} = \sigma(\tilde{\Phi}_i \boldsymbol{X_i} W^{(0)}), \end{cases}$$

where $\tilde{\Phi}_i = D_i^{-1/2} \Phi_i D_i^{-1/2}$, and $D_i$ is diagonal matrix of the degrees of $\Phi_i$. ":" denotes the concatenation of the two matrices. $F$ denotes fully connected layers, and $\sigma$ is an activation function, such as the ReLU.

**The construction of temporal similarity graph $\Phi$.** The temporal similarity graph $\Phi$ plays a crucial role in our model. As suggested by its name, it measures the similarity between events in the time domain. The use of temporal similarity graph is not arbitrary. It is essentially an important component in the intensity function of the nonlinear marked Hawkes processes with multiplicative kernels.

Given a symmetric kernel $\phi$, the weight between two events $(t_i, m_i)$ and $(t_j, m_j)$ can be defined by $\phi(t_i - t_j)$. The dimension of $\Phi$ is determined by the length of the trimmed history, which is preset by fixing the influential range, i.e., the number of past events that the next event is relevant to. Presetting the range makes the temporal similarity graphs and feature matrices of different events aligned. Trimmed history is also considered a reasonable approximation of the full history as the major influence comes from the closest events due to the decay of influence.

**Nonliear marked Hawkes process with multiplicative kernels.** We define a special type of marked Hawkes processes that incorporates multiplicative kernels for time and marks, whose intensity can be written as

$$\lambda(t, m) = \mu p(m) + \int_0^t \int_{\mathcal{M}} (\phi\kappa) * dN, \tag{2}$$

where $\mu$ is the base intensity and $p$ is a deterministic density function w.r.t. the mark $m$. $\phi$ and $\kappa$ are two positive definite kernel functions for arrival time and marks. $*$ denotes the convolution operation. It can be seen that such intensity is a linear convolution function. To relax the assumption of the linearity of intensity function, Eq. (2) can be extended to nonlinearity:

$$\lambda(t, m) = h\left(\mu p(m) + \int_0^t \int_{\mathcal{M}} (\phi\kappa) * dN\right), \tag{3}$$

where $h : \mathbb{R} \to \mathbb{R}^+$ is a non-negative function. It can be verified that the likelihood of such process can be viewed as a function of matrices $\Phi$ and $\mathcal{K}$. The former matrix is composed of $\phi(t_i - t_j)$'s. We call it the *temporal similarity graph*, as it measures the similarity between each two events. It can be seen that the feature kernel $\mathcal{K}$ provides an embedding method for the marks, in accordance with the theory of reproducing kernel Hilbert space. Therefore, the estimation of the next interarrival time $\tau$ and mark $m$, which is calculated from the estimated parameters by maximizing the likelihood, can be viewed as a function (denoted by $g$) of $\Phi$ and $\mathcal{K}$. The estimation of the next interarrival time $\hat{\tau}$ and mark $\widehat{m}$ can be written by

$$\hat{\tau}, \quad \widehat{m} = g(\Phi \odot \mathcal{K}),$$

where $\Phi$ is the temporal similarity graph, $\mathcal{K}$ the Gram matrix of the features (marks) and $\odot$ denotes the Hadamard product. The process can be interpreted in a sense that the next event is determined by the topology of the temporal similarity graph and the similarity of features. The closer two events are, the more similar their corresponding features will be.

**Table 2: Equivalence among conditional distributions, loss functions, and conditional intensity functions. Φ and Γ are the cdf of a standard normal distribution and an upper incomplete gamma function, respectively.**

| **Distribution** $p(\tau_i \vert \mathcal{H}_i)$ | **Equivalent loss function** $\ell_t(\tau_i, \hat{\tau}_i)$ | **Conditional intensity functions** $\lambda(\tau_i \vert \mathcal{H}_i)$ |
|---|---|---|
| Exponential($\lambda$) | $\sum_{i=1}^{N} \frac{\tau_i}{\hat{\tau}_i} + \log \hat{\tau}_i$ | $1/\hat{\tau}_i$ |
| Gaussian($\mu$, $\sigma^2$) | $\frac{1}{\sigma\ i} \sum_{i=1}^{N} (\hat{\tau}_i - \tau_i)^2 - 2N \log \sigma_i$ | $\frac{\exp\left((\hat{\tau}_i - \tau_i)^2/2\sigma_i^2\right)}{\sqrt{2\pi}\sigma\left(1 - \Phi((\hat{\tau}_i - \tau_i)/\sigma_i\right)}$ |
| Gamma($k$, $\theta$) | $\sum_{i=1}^{N} \log \Gamma(\frac{\hat{\tau}_i}{\theta_i}) + \frac{\tau_i}{\theta_i} - \frac{\hat{\tau}_i}{\theta_i} \log \frac{\tau_i}{\theta_i}$ | $\frac{\left(\frac{\tau_i}{\theta_i}\right)^{\left(\frac{\hat{\tau}_i}{\theta_i}\right)} \tau^{-1} e^{-\frac{\tau_i}{\theta_i}}}{\Gamma\left(\frac{\hat{\tau}_i}{\theta_i}, \frac{\tau_i}{\theta_i}\right)}$ |
| Laplacian($\mu$, $\sigma$) | $\frac{1}{\sigma\ i} \sum_{i=1}^{N} \vert\hat{\tau}_i - \tau_i\vert$ | $\begin{cases} \frac{1}{2\sigma_i \exp\left(-\frac{\hat{\tau}_i - \tau_i}{\sigma_i}\right) - \sigma_i}, & \hat{\tau}_i \leq \tau_i \\ \frac{1}{\sigma_i} \exp\left(-\frac{-2(\hat{\tau}_i - \tau_i)}{\sigma_i}\right), & \hat{\tau}_i > \tau_i \end{cases}$ |

**Complexity analysis.** Our model has a running time complexity of $O(Nm^2p)$ for each epoch, where $N$ is the number of events, $m$ and $p$ are the length of the trimmed history and the dimension of features, respectively. Note that $m \ll N$. For long sequences when $m \ll N$ does not hold, the temporal similarity graph (shown in Figure 2) becomes sparse with the similarity values concentrate around diagonal entries. Therefore, the complexity of our model becomes $O(Nmp)$ for long histories. This complexity is superior to [31, 36, 37] whose complexity is $O(N^3p)$. It is also better than [1], which has a complexity of $O(Np^2)$ for high-dimensional Hawkes processes where $p \gg m > 0$. The THP [38] has the complexity of $O(Nm^2p)$, which is much worse than our model for long histories.

## 5 LIKELIHOOD RATIO AND LOSS FUNCTION

Generally, learning a stochastic process by maximizing the log-likelihood is viewed as an unsupervised task. For neural point processes, each event $(t_i, m_i)$ in the input sequence is treated as a label. The general objective function can be written as

$$\text{loss} = \sum_{i=1}^{N} \left(\ell_m(m_i, \widehat{m}_i) + c\ell_t(\tau_i, \hat{\tau}_i)\right).$$

Here $\widehat{m}_i$ and $\hat{\tau}_i$ are the outputs for the feature and interarrival time of the $i$-th event, given the history $\bar{\mathcal{H}}_i$. $\tau_i = t_i - t_{i-1}$, $t_0 = 0$ and $m_i$ is the actual interarrival time and feature of the next event. $c$ is a hyper-parameter controlling the weight of time. $\ell_m$ and $\ell_t$ are the respective loss functions. It is worth noting that, the maximum likelihood estimator of neural point processes also admits the form, as a result of the invariance property.

One of the most challenging parts in applying marked point processes is that the exact form of the conditional intensity $\lambda^*(t, m)$ is not known. Traditionally, the loss function is designed by assuming a specific form of the conditional intensity function. There are several attempts made by researchers to designate some specific forms for the intensity. RMTPP [6] uses an exponential form, whereas NHP [18] adopts sigmoid. Such choices, however, may restrict the expressive power of neural networks. Moreover, the calculation of the expectation of the next interarrival time usually does not have analytic solutions, and thus one has to turn to numerical methods, such as Monte Carlo simulation, which is computationally unfriendly. Recently, [21] proposes an approach that avoids the specification of the intensity. It first models the integral of the intensity using a feedforward neural network and then obtains the intensity function as its derivative. However, this method is unable to perform long-term predictions, as the derivatives for future events are not available. In this paper, we propose to consider the distribution of the interarrival times, instead of the intensity function, as a result the Monte Carlo integration can be avoided. The next lemma states that the probabilistic structure of a point process can be equivalently defined by its conditional intensity function as well as the conditional density of the interarrival times.

Lemma 1 (Equivalence between conditional intensity function and conditional density of the interarrival times [4]). *A regular point process is specified uniquely by the conditional intensity function $\lambda_i^*(t)$ if and only if the conditional probability densities of the next arrival time satisfy that $p_i^*(t) = \lambda_i^*(t) \exp\left\{-\int_{t_{i-1}}^{t} \lambda_i^*(s)ds\right\}$, for all $i = 1, 2, \cdots$.*

The proof of this lemma can be found in [4]. This result tells that we do not have to designate the form of the conditional intensity function, but instead we can directly impose the probabilistic assumptions on the interarrival times.

**Likehihood ratio loss function**. As we would like to jointly optimize the network structure and the parameters, we propose a loss function based on the likelihood ratio statistic, which takes into account both the likelihood of the observations and the complexity (in terms of the number of free parameters in a network) of the model. The optimal network structure and parameter can be obtained by optimizing the likelihood ratio loss function, which can by written as,

$$\boldsymbol{w}^*, \Xi^* = \arg \inf_{\boldsymbol{w}, \Xi \in \mathbb{F}} \inf_{p \in \mathbb{P}} \log \frac{\sup_{\boldsymbol{\theta} \in \Theta_0} \prod_{i=1}^{N} p(\tau_i, m_i | \boldsymbol{\theta})}{\sup_{\boldsymbol{\theta} \in \Theta_1} \prod_{i=1}^{N} p_i(\tau_i, m_i | \mathcal{H}_i; \boldsymbol{\theta}) \chi^2_{\alpha}(N - d_{\Xi})}$$

where $\Theta_0$ is the null parameter space, $\Theta_1$ the parameter space restricted by the neural network, $\mathbb{P}$ the distribution family, $\boldsymbol{w}$ the network parameter, and $\Xi$ the network structure. According to the traditional model selection theory, this likelihood ratio has a $\chi^2$ distribution with degree of $N - d_{\Xi}$, where $N$ is the number of samples, and $d_{\Xi}$ is the number of free parameters in model $d_{\Xi}$. Therefore, we introduce a $\chi^2$ coefficient in the loss function, where $\chi^2_{\alpha}(N - d_{\Xi})$ represents the $\alpha$-percentile of the $\chi^2$ distribution with degree $N - d_{\Xi}$. This loss function measures the goodness-of-fit, and it can be reduced to a generalized likelihood ratio test problem.

**Moment matching.** We propose to use a moment matching mechanism in the loss function. We directly output the expectation of the next interarrival time $\hat{\tau}_i = \mathbb{E}(\tau_i | \bar{\mathcal{H}}_i)$ and feature $\hat{m}_i = \mathbb{E}(m_i | \bar{\mathcal{H}}_i)$, which are the first-order moment of the next interarrival time and mark, respectively. This method has two benefits. First, it is convenient for the network to predict the next event. Second, it reduces the number of free parameters to estimate, making the model more robust to overfitting.

**Discussion.** It is worth noting that this model selection method can be regarded as a traditional statistical goodness-of-fit problem. It is only valid for the classical underparameterized situations where the number of the free parameters in the network is smaller than the number of events for training.

## 6 THE SATURATION PHENOMENON FOR THE LEARNING OF NEURAL POINT PROCESSES

The neural networks for a probabilistic model can be viewed as an estimator for the unknown parameters. Classical learning theory [10] indicates as the number of parameters in a model increases, the model becomes prone to overfitting and the test error gets larger. Recent studies [20] find that many deep learning tasks exhibit a "double descent" phenomenon where model performance initially gets worse and then gets better as the number of free parameters in the model increases. Therefore, many believes that a mammoth neural network model always means a good opportunity to obtain better performance. However, this conjecture is not valid when it comes to neural marked point processes. Instead, the performance of neural networks often becomes "saturated" – no matter how much efforts are put into making the neural network more complicated and has more parameters, the performance just stop increasing.

The reason of this phenomenon is that the representation capability of the neural network is capped by the assumptions of point process. An extreme example would be a neural homogeneous Poisson process, which is like "to break a butterfly upon a wheel" – no matter how delicate the network is, its generalization ability is still rather weak. We present an explanation in Figure 3. It can be seen that the generalization error can be decomposed into three parts: network estimation error, model specification error and some irreducible error caused by the randomness of the ground truth. In the underparameterized case, the function space defined by the neural network, ie. $\mathbb{F}$, may not include the function space defined by the point process probability structure, ie. $\mathbb{M}$, leading to the network estimation error. This error can be reduced by making the network to be more complicated and have more parameters, until the optimal estimation $\hat{p}_{\mathbb{P}}$ is included in $\mathbb{F}$. After that, however, all the efforts put into enlarging $\mathbb{F}$ are all in vain, as $\hat{p}_{\mathbb{P}}$ is already achieved. As a result, the "double descent" phenomenon does not occur when it comes to neural point processes.

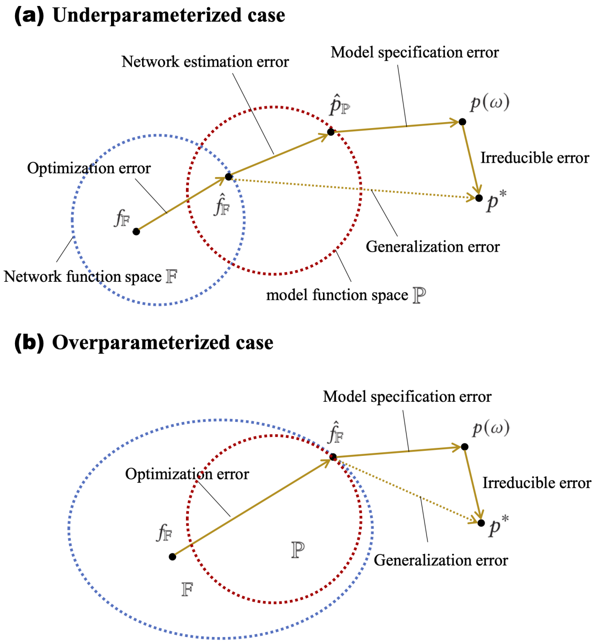

**Figure 3: An illustration of the generalization error decomposition and the performance saturation phenomenon. $\mathbb{F}$ and $\mathbb{P}$ are the function spaces defined by the neural network and the point process probabilistic assumptions. $f_{\mathbb{F}}$ is a network model with arbitrary parameter. $\hat{f}_{\mathbb{F}}$ represents the optimal estimation learned by neural network, and $\hat{p}_{\mathbb{P}}$ denotes the optimal estimation of under the probabilistic assumption. $p^*$ and $p(\omega)$ indicate the ground truth and a realization, respectively. (a) In the underparameterized case, network function space $\mathbb{F}$ is not large enough to find the optimal estimation given the probabilistic model restrictions. (b) In the underparameterized case, $\mathbb{F}$ is large enough to obtain the optimal estimation, where the network estimation error is eliminated. However, the performance cannot be further improved as the model specification error cannot be reduced by modifying the neural network only.**

We perform an example experiment on a synthetic dataset. The dataset is simulated from a 10-dimensional Hawkes process. The description of the dataset can be found in Section 7. We present the accuracy of mark prediction on test dataset using three different network structures in Figure 4. When the network is underparameterized, the performance continuously increases as the size of network expands, until the performance reaches certain point. In the overparameterized regime, all the networks have similar performance. Neither descent nor "double ascent" is observed.

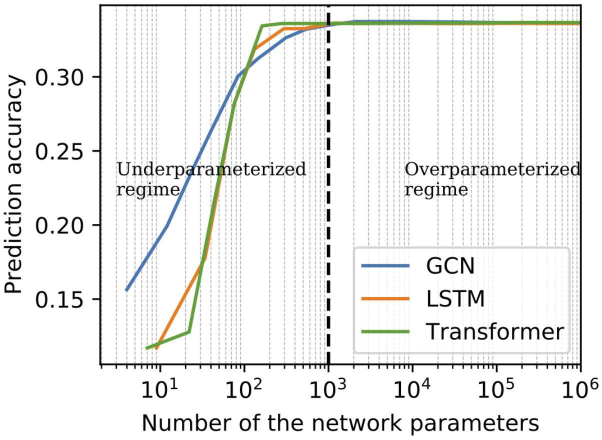

**Figure 4: The performance saturation phenomenon for neural point processes. It can be seen that the model performance on test dataset stagnates after as the number of network parameters increases.**

**Table 3: Statistics of datasets.**

| Dataset | # of events | | # of sequences | | # of event types $K$ |
|---|---|---|---|---|---|
| | Train set | Test set | Train set | Test set | |
| Hawkes | 36k | 7k | 100 | 40 | 10 |
| ATM | 370k | 182k | 1085 | 469 | 7 |
| Weeplace | 98k | 31k | 21 | 8 | 8 |
| IPTV | 731k | 243k | 227 | 75 | 16 |

## 7 EXPERIMENTS

In this section, we evaluate our model against some state-of-the-art baselines on one synthetic and threes real-world datasets.

**Datasets.** The datasets we use are listed as follows. We summarize the statistics of the datasets in Table 3.

- *Hawkes*: a synthetic dataset with categorical features. The event sequences are generated from a 10-dimensional Hawkes process with uniformly sampled parameters.
- *IPTV* [17]: a real-world dataset with categorical features. The dataset consists of IPTV viewing events with timestamps and categories.
- *Weeplace* [16]: a real-world dataset with both categorical and continuous features. The dataset contains the check-in histories of users at different locations (longitudes and latitudes).
- *ATM* [29]: a real-world dataset with categorical features. The dataset is composed of the event logs of error reporting and failure tickets.

**Experimental environment.** All the experiments were conducted on a server with 64G RAM, a 16 logical cores CPU (AMD Ryzen Threadripper 1900X) and 4 GPUs (Nvidia GeForce GTX 1080 Ti) for acceleration.

**Table 4: Performance on prediction.**

| Dataset | Model | Accuracy (feature) | RMSE (time) | Average Running time (s) |
|---|---|---|---|---|
| Hawkes | RMTPP [6] | 32.46% | 5.565 | 0.451 |
| | IRNN [29] | 33.40% | 4.395 | 0.475 |
| | NHPP [18] | 33.61% | 4.480 | 46.47 |
| | MAHP [30] | 10.01% | 4.898 | 1.794 |
| | GeoHP [26] | 22.91% | 12.62 | 38.94 |
| | THP [38] | 33.27% | 35.01 | 122.7 |
| | 1-layer GCN | 33.75% | 4.506 | **0.0792** |
| | 2-layer GCN | 33.81% | **4.385** | 0.0888 |
| | GCN + LSTM | 33.16% | 4.374 | 0.1258 |
| | GCN + TFM | **33.97%** | 4.392 | 0.2551 |
| ATM | RMTPP [6] | 76.64% | 7.150 | 5.756 |
| | IRNN [29] | 76.19% | 2.793 | 6.299 |
| | NHPP [18] | 33.78% | 7.558 | 660.52 |
| | MAHP [30] | 41.91% | 3.202 | 24.876 |
| | GeoHP [26] | 14.91% | 9.268 | 872.40 |
| | THP [38] | 68.76% | 4.534 | 14.612 |
| | 1-layer GCN | 76.56% | 2.825 | **0.2611** |
| | 2-layer GCN | 90.88% | **2.612** | 0.3993 |
| | GCN + LSTM | **91.41%** | 2.899 | 0.5061 |
| | GCN + TFM | 91.08% | 2.767 | 1.5754 |
| IPTV | RMTPP [6] | 57.57% | 34.382 | 11.281 |
| | IRNN [29] | 58.63% | 34.311 | 11.065 |
| | NHPP [18] | 31.05% | 19.929 | 1070.15 |
| | MAHP [30] | 18.02% | 36.738 | 28.213 |
| | GeoHP [26] | 43.12% | 25.421 | 907.91 |
| | THP [38] | 71.94% | 31.325 | 10.031 |
| | 1-layer GCN | 75.28% | 11.162 | **1.8634** |
| | 2-layer GCN | 75.35% | **10.866** | 2.0753 |
| | GCN + LSTM | **76.11%** | 11.133 | 3.1946 |
| | GCN + TFM | 76.01% | 11.139 | 7.6419 |
| Weeplace | RMTPP [6] | 22.07% | 7.162 | 1.400 |
| | IRNN [29] | 23.37% | **6.448** | 1.434 |
| | NHPP [18] | 25.71% | 6.773 | 140.26 |
| | MAHP [30] | 15.13% | 6.969 | 5.210 |
| | GeoHP [26] | 17.74% | 28.28 | 42.89 |
| | THP [38] | 29.24% | 51.78 | 51.15 |
| | 1-layer GCN | 31.61% | 6.498 | **0.1831** |
| | 2-layer GCN | 31.81% | 6.493 | 0.2090 |
| | GCN + LSTM | **32.09%** | 6.525 | 0.2832 |
| | GCN + TFM | 30.05% | 6.563 | 0.6990 |

### 7.1 Task 1: Comparison Among Network Structures

In this task, we compare the performance of different neural network architectures, to demonstrate the advantage of applying GCN networks in marked point processes.

**Baselines.** We compare our model with six state-of-the-art neural-based methods: RMTPP [6] (RNN-based model), IRNN [29], NHPP [18] (LSTM-based model), MAHP [30] and GeoHP [26], and THP [38] (transformer-based model). Meanwhile, we also compare with some variants of our model including the one-layer GCN and the combinations of the GCN with LSTM and the GCN with transformer.

**Metrics.** We assess the performance of each model in three aspects: time prediction, feature prediction and training time. We use **RMSE** for the time prediction, and we measure the categorical features prediction by the percentage of correct predictions (**Accuracy**). A higher accuracy and a lower RMSE indicate a better performance. The training time per epoch is also recorded, as a measure of the model’s efficiency.

**Experimental settings.** We apply likelihood ratio loss to train our model. The hyper-parameters of all models were tuned for the best performance. We use a single fully connected layer after the graph convolutional layers in our model, to predict the time and event category.

**Discussion.** The experimental results are shown in Table 4. From the experimental results, we can observe that our GCN-based model outperforms the baseline methods in terms of time prediction error, category prediction accuracy and training time. In addition, by combining with GCN networks, the performance of LSTM and transformer are greatly improved. We contribute the performance improvements into three aspects: (1). The GCN model encodes the event correlations into the temporal similarity graph, which can better encode the relations among each event than other models. (2). The lightweight of GCN can greatly speed up the training processes. (3). The likelihood ratio loss fits the task much better than other intensity losses used in the baseline methods. We furthermore show this point in Task 2.

## 7.2 Task 2: Comparison Among Loss Functions

To test the model’s performance under different optimization objective, i.e. loss function, we conduct experiments on the state-of-the-art models such as LSTM and transformer with different loss types. The experiment results are shown in Table 5.

**Experimental settings.** We select some state-of-the-art models as the representatives of the combination of certain network architecture and loss type: RMTPP [6] represents the RNN-based model with exponential intensity loss, NHPP [18] represents the LSTM-based model with softmax intensity loss, THP [38] represents the transformer-based model with exponential intensity loss. We further implement two models: the LSTM-based model which has one layer LSTM, and transformer-based model which has one layer transformer, based on our proposed likelihood ratio loss which minimizes the exponential interarrival loss(discussed in Section 5). We carefully tune the hyper-parameters to enable all the models achieve their best performance.

**Discussion.** From Table 5, we can infer that the optimization objective can greatly affect the performance of the model. With adopting our proposed likelihood ratio loss function, the prediction accuracy is consistently enhanced among four datasets, which demonstrates the significance of the assumption on the exponential distribution of interarrival times. In addition, the baselines [6, 18, 38] apply Monte Carlo integration to approximate their intensity, which may slowdown the entire inference process. By adopting our exponential distribution assumption with moment matching, we can significantly speed up the training process.

## 8 CONCLUSION

In this paper, we describe an interesting performance saturation phenomenon when training neural marked point process models: performance often becomes stagnated at some point, and cannot be improved any more by making the network more complicated. From the generalization error analysis and experimental results, we conclude our paper with two suggestions for using neural marked point process models: first, for some cases, a simple network structure can perform as well as complicated ones, but more efficiently; second, using a proper probabilistic assumption is as equally, if not more, important as improving the network structure. In the future, we would like to investigate the reason of this phenomenon theoretically.

**Table 5: Performance comparison with different loss types. Exp interarrival represents the exponential distribution assumption on the interarrival times. Exp/Sigmoid/Softplus intensity represents the respective form assumptions on the conditional intensity function.**

| Dataset | Model | Accuracy (feature) | RMSE (time) |
|---|---|---|---|
| Hawkes | **LSTM + Exp interarrival** | 33.68% | 4.436 |
| | RNN + Exp intensity [6] | 32.46% | 5.565 |
| | LSTM + Sigmoid intensity [18] | 33.61% | 4.480 |
| | **TFM + Exp interarrival** | 33.57% | 4.508 |
| | TFM + Softplus intensity [38] | 33.27% | 35.01 |
| ATM | **LSTM + Exp interarrival** | 92.51% | 3.105 |
| | RNN + Exp intensity [6] | 76.64% | 7.150 |
| | LSTM + Sigmoid intensity [18] | 33.78% | 7.558 |
| | **TFM + Exp interarrival** | 90.60% | 3.245 |
| | TFM + Softplus intensity [38] | 68.76% | 4.534 |
| IPTV | **LSTM + Exp interarrival** | 76.20% | 11.238 |
| | RNN + Exp intensity [6] | 57.57% | 34.382 |
| | LSTM + Sigmoid intensity [18] | 31.05% | 19.929 |
| | **TFM + Exp interarrival** | 76.20% | 10.188 |
| | TFM + Softplus intensity [38] | 71.94% | 31.325 |
| Weeplace | **LSTM + Exp interarrival** | 31.86% | 6.777 |
| | RNN + Exp intensity [6] | 22.07% | 7.162 |
| | LSTM + Sigmoid intensity [18] | 25.71% | 6.773 |
| | **TFM + Exp interarrival** | 32.62% | 6.571 |
| | TFM + Softplus intensity [38] | 29.24% | 51.78 |

## ACKNOWLEDGMENT

This research/project is supported by: (1) the National Research Foundation, Singapore under its Industry Alignment Fund – Prepositioning (IAF-PP) Funding Initiative; (2) NTU Singapore Nanyang Assistant Professorship (NAP) grant M4081532.020; (3) Singapore MOE AcRF Tier-1 grant 2018-T1-002-143 (RG131/18 (S)); and (4) Alibaba-NTU Singapore Joint Research Institute, Nanyang Technological University. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

## REFERENCES

[1] Massil Achab, Emmanuel Bacry, Stéphane Gaïffas, Iacopo Mastromatteo, and Jean-François Muzy. 2017. Uncovering causality from multivariate Hawkes integrated cumulants. *The Journal of Machine Learning Research* 18, 1 (2017), 6998–7025.

[2] Emmanuel Bacry, Iacopo Mastromatteo, and Jean-François Muzy. 2015. Hawkes processes in finance. *Market Microstructure and Liquidity* 1, 01 (2015), 1550005.

[3] Kurt Binder, David M Ceperley, J-P Hansen, MH Kalos, DP Landau, D Levesque, H Mueller-Krumbhaar, D Stauffer, and J-J Weis. 2012. *Monte Carlo methods in statistical physics*. Vol. 7. Springer Science & Business Media.

[4] Daryl J Daley and David Vere-Jones. 2003. An introduction to the theory of point processes, volume 1: Elementary theory and methods. *Verlag New York Berlin Heidelberg: Springer* (2003).

[5] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*. 3844–3852.

[6] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. 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*. ACM, 1555–1564.

[7] Nan Du, Mehrdad Farajtabar, Amr Ahmed, Alexander J Smola, and Le Song. 2015. Dirichlet-hawkes processes with applications to clustering continuous-time document streams. In *SIGKDD'15*. ACM, 219–228.

[8] Nan Du, Yichen Wang, Niao He, Jimeng Sun, and Le Song. 2015. Time-sensitive recommendation from recurrent user activities. In *Advances in Neural Information Processing Systems*. 3492–3500.

[9] Mehrdad Farajtabar, Jiachen Yang, Xiaojing Ye, Huan Xu, Rakshit Trivedi, Elias Khalil, Shuang Li, Le Song, and Hongyuan Zha. 2017. Fake news mitigation via point process based intervention. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 1097–1106.

[10] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

[11] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.

[12] Tianbo Li and Yiping Ke. 2019. Thinning for accelerating the learning of point processes. *Advances in Neural Information Processing Systems* 32 (2019), 4091–4101.

[13] Tianbo Li and Yiping Ke. 2020. Tweedie-Hawkes Processes: Interpreting the Phenomena of Outbreaks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 4699–4706.

[14] Tianbo Li, Pengfei Wei, and Yiping Ke. 2018. Transfer Hawkes Processes with Content Information. In *2018 IEEE International Conference on Data Mining (ICDM)*. 1116–1121.

[15] Scott Linderman and Ryan Adams. 2014. Discovering latent network structure in point process data. In *International Conference on Machine Learning*. 1413–1421.

[16] Bin Liu, Yanjie Fu, Zijun Yao, and Hui Xiong. 2013. Learning geographical preferences for point-of-interest recommendation. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1043–1051.

[17] Dixin Luo, Hongteng Xu, Yi Zhen, Xia Ning, Hongyuan Zha, Xiaokang Yang, and Wenjun Zhang. 2015. Multi-task multi-dimensional hawkes processes for modeling event sequences. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[18] Hongyuan Mei and Jason M Eisner. 2017. The neural hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*. 6754–6764.

[19] Federico Monti, Michael Bronstein, and Xavier Bresson. 2017. Geometric matrix completion with recurrent multi-graph neural networks. In *Advances in Neural Information Processing Systems*. 3697–3707.

[20] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. 2019. Deep Double Descent: Where Bigger Models and More Data Hurt. (2019).

[21] Takahiro Omi, Naonori Ueda, and Kazuyuki Aihara. 2019. Fully Neural Network based Model for General Temporal Point Processes. In *Advances in Neural Information Processing Systems*.

[22] Antti Penttinen, Dietrich Stoyan, and Helena M Henttonen. 1992. Marked point processes in forest statistics. *Forest science* 38, 4 (1992), 806–824.

[23] Michael D Porter, Gentry White, et al. 2012. Self-exciting hurdle models for terrorist activity. *The Annals of Applied Statistics* 6, 1 (2012), 106–124.

[24] Alex Reinhart et al. 2018. A review of self-exciting spatio-temporal point processes and their applications. *Statist. Sci.* 33, 3 (2018), 299–318.

[25] Yeon Seonwoo, Alice Oh, and Sungjoon Park. 2018. Hierarchical Dirichlet Gaussian Marked Hawkes Process for Narrative Reconstruction in Continuous Time Domain. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 3316–3325.

[26] Jin Shang and Mingxuan Sun. 2019. Geometric Hawkes Processes with Graph Convolutional Recurrent Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4878–4885.

[27] Long Tran, Mehrdad Farajtabar, Le Song, and Hongyuan Zha. 2015. Netcodec: Community detection from individual activities. In *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM, 91–99.

[28] Yichen Wang, Bo Xie, Nan Du, and Le Song. 2016. Isotonic hawkes processes. In *International conference on machine learning*. 2226–2234.

[29] Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen M Chu. 2017. Modeling the intensity function of point process via recurrent neural networks. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[30] Hongteng Xu, Xu Chen, and Lawrence Carin. 2018. Superposition-assisted stochastic optimization for hawkes processes. *arXiv preprint arXiv:1802.04725* (2018).

[31] Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. 2016. Learning granger causality for hawkes processes. In *International Conference on Machine Learning*. 1717–1726.

[32] Hongteng Xu and Hongyuan Zha. 2017. A Dirichlet Mixture Model of Hawkes Processes for Event Sequence Clustering. NIPS.

[33] Shuang-Hong Yang and Hongyuan Zha. 2013. Mixture of mutually exciting processes for viral diffusion. In *ICML*. 1–9.

[34] Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. 2020. Self-attentive hawkes process. In *International Conference on Machine Learning*. PMLR, 11183–11193.

[35] Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. 2020. Self-attentive Hawkes processes. In *International Conference on Machine Learning*.

[36] Ke Zhou, Hongyuan Zha, and Le Song. 2013. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *Artificial Intelligence and Statistics*. 641–649.

[37] Ke Zhou, Hongyuan Zha, and Le Song. 2013. Learning triggering kernels for multi-dimensional hawkes processes. In *International Conference on Machine Learning*. 1301–1309.

[38] Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. 2020. Transformer Hawkes Process. In *International Conference on Machine Learning*.