ON THE EXPRESSIVENESS, PREDICTABILITY AND INTERPRETABILITY OF NEURAL TEMPORAL POINT PROCESSES

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite the fast advance in neural temporal point processes (NTPP) which enjoys high model capacity, there are still some standing gaps to fill including model expressiveness, predictability, and interpretability, especially with the wide application of event sequence modeling. For expressiveness, we first show the incapacity of existing NTPP models for fitting time-varying especially non-terminating TPP, and propose a simple neural model for expressive intensity function modeling. To improve predictability which is not directly optimized by the TPP likelihood objective, we devise our new sampling techniques that enable error metric driven adaptive fine-tuning of the sampling hyperparameter for predictive TPP, based on the event history in training sequences. Moreover, we show how interval-based event prediction can be achieved by our prediction techniques. To achieve interpretable NTPP, we propose an influence definition from one event to the future by comparing the difference between the existence of the event and not, which enables the dependency learning among events and types. Experimental results on synthetic datasets and public benchmarks show the efficacy of our approach.

1 INTRODUCTION

Temporal point process (TPP) (Hawkes, 1971; Zhou et al., 2013) has been a popular and principled tool for modeling and predicting event data in continuous time space, which involves different forms of intensity functions for describing the event occurrence rate over time, with explicit (Xu et al., 2016; Xiao et al., 2016; Zhou et al., 2013) or implicit (Eichler et al., 2017) parametric forms. Most of the traditional TPPs have a time-varying intensity function, as used for predicting futures and analyze the influence among events or their dimensions e.g. types of events for multi-dimensional TPP (Liniger, 2009; Zhou et al., 2013; Wu et al., 2018), which can be used as the causality learning for event sequences. However, most traditional methods are based on specific intensity assumptions e.g. Self-Correcting and Self-Exciting functions, which limits the applicability for diverse datasets.

Neural temporal point process (NTPP) models (Du et al., 2016; Xiao et al., 2017) are becoming increasingly popular for their high capacity to obtain more varying intensity and competitive prediction results. Most of them are designed based on recurrent neural networks (RNN) which endows them with better adaptability to complex event sequences than traditional TPP. Despite their success, existing NTPP models still suffer from several issues. First, their intensity function still follows a specific model, e.g. the popular temporal exponential-decay prior (Du et al., 2016), which limits their expressions to capture rich intensity forms of real-world patterns and causes under-fitting and biased predictions. Besides, existing NTPP models are less interpretable than their classic non-network-based counterparts e.g. Hawkes process. Recent efforts (Wang et al., 2017; Xiao et al., 2019) have been made to introduce neural attention to improve the interpretability, while many studies (Xiao et al., 2019) have shown that attention may not be the panacea for achieving stable and reliable relation finding in sequence learning, and it has an unclear connection to the Bayesian framework.

Therefore, in this paper, we are mainly concentrated on solving the following two problems: i) can we design a flexible form for fitting intensity function of NTPP and even get better predictions compared with existing NTPP models; ii) can we increase the interpretability of NTPP and thus directly use them for analyzing the influences between events? We aim to model a more flexible intensity-based

NTPP and design an interpretable framework to fill the interpretability gap and evaluate the infectivity matrix and triggering matrix as traditional Hawkes processes do. **The main highlights are**:

1) We choose a simple, effective yet in literature unused neural model (to our best knowledge, see Table 1) for encoding time-varying intensify function $\lambda(t|\mathcal{H}_j)$. Specifically, it is a more general form than the specific parameterization of neural intensity in (Du et al., 2016; Mei & Eisner, 2017), thus is less prior dependent. Note that similar to ours, FullyNN (Omi et al., 2019) also directly uses a neural net but to encode the intensity integral $\Lambda(t|\mathcal{H}_j)$ instead of $\lambda(t|\mathcal{H}_j)$, which can not guarantee the satisfaction of $\Lambda(t_j|\mathcal{H}_j) = 0$ by neuralizing $\Lambda(t|\mathcal{H}_j)$.

2) We propose a prediction error metric driven sampling approach based on the Time Change Theorem (Brown et al., 2002). This approach is applicable to TPP models either based on intensity modeling (Du et al., 2016) or its integral (i.e. intensity measure) (Omi et al., 2019). We further show that when the inverse of the intensity measure exists, for popular TPP models including both classic (Isham & Westcott, 1979) and neural models (Du et al., 2016), the event prediction can be fulfilled in a closed-form from a tuned sampling variable u based on history sequences.

3) Inspired by the counter-factual causality learning scheme (Tsushima et al., 2020), we propose a novel technique for realizing interpretable neural TPP, regarding the dependency between temporal events as well as event types. We define the influence from one event to the future by comparing the difference between the existence of the event without specific assumptions on model design. We will show that our interpretable technique is coherent with the infectivity matrix of traditional Hawkes process, which cannot be achieved by the attention-based NTPP (Xiao et al., 2017).

Throughout the paper, we term the full version of our neural TPP model as **E**xpressive, **P**redictive, and **I**nterpretable Neural TPP, namely **EPI-NTPP**. The structure is shown in Fig. 6 in Appendix. When one or two of its components are disabled as they are orthogonal to each other, e.g. Expressive NTPP without the prediction and interpretation modules, we call it **E-NTPP** as shown in Fig. 1.

2 Backgrounds and Related Works

Here we give the background to ease the later presentation. TPPs are described with a conditional intensity function over time, being a direct way to represent the event occurrence. Given the event history $\mathcal{H}_j = \{t_1, t_2, \ldots, t_j\}$, the conditional intensity function for event $j+1$ is:

$$\lambda(t|\mathcal{H}_j) = \frac{f(t|\mathcal{H}_j)}{1 - F(t|\mathcal{H}_j)}, \tag{1}$$

where $f(t|\mathcal{H}_j)$ is the conditional density function and $F(t|\mathcal{H}_j)$ is the corresponding cumulative probability function for event $j+1$. Note here we use the single-dimension TPP for notation brevity. Then the intensity measure, which can also call the cumulative intensity function, is specified as

$$\Lambda\left(t|\mathcal{H}_j\right) = \int_{t_j}^{t} \lambda\left(\tau|\mathcal{H}_j\right) d\tau. \tag{2}$$

The Hawkes process (Hawkes, 1971) is a mutually exciting multivariate point process, which is widely used to learn event streams. The definition of the intensity function can be simply specified as

$$\lambda_u(t|\mathcal{H}_j) = \mu_u + \sum_{t_i<t} \alpha_{uu_i} g(t - t_i), \tag{3}$$

where μ_u is the base intensity of the type u and the coefficient α_{uu_i} captures the mutually exciting property between type u and u_i. $g(t) = \exp\left(-wt\right)$ is the kernel function which captures time-varying decaying information. To learn the Hawkes process, the EM algorithm (Zhou et al., 2013) is widely used which alternatively calculates the triggering probability $\{p_{ij}\}$ and the infectivity matrix $\{\alpha_{uu'}\}$. The Hawkes process takes advantage of its sound interpretability of event inter-effects by mathematical derivation, but it suffers limited fitting capacity due to its restricted intensity form.

Several NTPPs have been recently proposed to model the event sequences. RMTPP (Du et al., 2016) is one of most popular NTPP models, which uses recurrent neural networks (RNNs) (Elman, 1990) to model the TPP by concatenating the time feature $\mathbf{t}_j$ and marker embedding $\mathbf{y}_j = \mathbf{W}_{em}\mathbf{u}_j + \mathbf{b}_{em}$ as input vector. Denote $\mathbf{h}_j$ the output of RNN, the intensity is specified as (Du et al., 2016):

$$\lambda(t|\mathcal{H}_j) = \exp\left(\mathbf{v}^\top \cdot \mathbf{h}_j + w(t - t_j) + b\right), \tag{4}$$

Table 1: Comparison of the three features as discussed in this paper. For the difficulty of combining methods with our proposed techniques (see the last two columns), 'difficult' means more computing overhead than the 'moderate' methods. 'easy' means that the methods do not need to calculate the numerical integral for P-NTPP, which costs less than the 'moderate' methods. See details in Appendix J. $^\diamond$**Note for combination, it refers to E-NTPP rather than the complete EPI-NTPP.**

Methods	Expressiveness	Predictability	Interpretability	Combine P-NTPP	Combine I-NTPP
RMTPP	Eq. 4: exp-form intensity	inadequate integration	little (see Sec. 2)	easy	moderate
NHP	Eq. 5: exp-link-form intensity	inadequate integration	no	moderate	moderate
LogNormMix	Eq. 6: fail to fit self-correcting PP	adequate integration	no	difficult	difficult
FullyNN	Eq. 7: NN-form intensity integral	inadequate integration	no	convenient	need differentiation
AttentionPP	Eq. 5: exp-link-from intensity	inadequate integration	attention score	moderate	moderate
EPI-NTPP$^\diamond$	Eq. 10: NN-form intensity	adequate integration	probability	moderate	moderate

where $\mathbf{v}^\top \cdot \mathbf{h}_j$ denotes the accumulative influence, $w(t-t_j)$ emphasizes the influence of current event j and the rest parameter b gives a base level. Though simple form as RMTPP is, the w parameter can be interpreted as inhibiting ($w > 0$) or triggering($w < 0$) the occurrence of future events given the past, which is in fact a nontrivial advantage of our I-NTPP models.

The Neural Hawkes Processes (NHP) (Mei & Eisner, 2017) develop a continuous-time LSTM to model self-modulating Hawkes processes, whose intensity is as:

$$\begin{aligned}\lambda_u(t|\mathcal{H}_j) =&\phi_u(\mathbf{w}_k^\top \mathbf{h}(t)), \quad \mathbf{h}(t) = \mathbf{o}_j(2\sigma(2(\mathbf{c}(t)) - 1), \\ \mathbf{c}(t) =&\bar{\mathbf{c}}_{j+1} + (\mathbf{c}_{j+1} - \bar{\mathbf{c}}_{j+1}) \exp(-\delta_{j+1}(t - t_j))\end{aligned} \tag{5}$$

where ϕ_u and σ are the link functions and $\mathbf{o}_j, \bar{\mathbf{c}}_{j+1}, \mathbf{c}_{j+1}, \bar{\mathbf{c}}_{j+1}$ and δ_{j+1} are the parameters learned with neural network. Here k in above equations is about the specific design in the cited work continuous-time LSTM (Mei & Eisner, 2017), which can be understood as the index of the cell. However, these two mainstreams of intensity-based methods are both based on exponentially time-varying kernel with several link functions, which is still specific and can cause the the biased prediction problem as will be discussed in Sec. 3.1.

There are also efforts indirectly learning the NTPP model. For instance, the LogNormMix (Shchur et al., 2020) suggests generally learning the conditional probability density with a log-normal mixture:

$$f(t|\mathcal{H}_j) = \sum_{k=1}^{K} w_k \frac{1}{\sqrt{2\pi}(t - t_j) \cdot s_k} \exp\left(-\frac{(\log(t - t_j) - \mu_k)^2}{2s_k^2}\right), \tag{6}$$

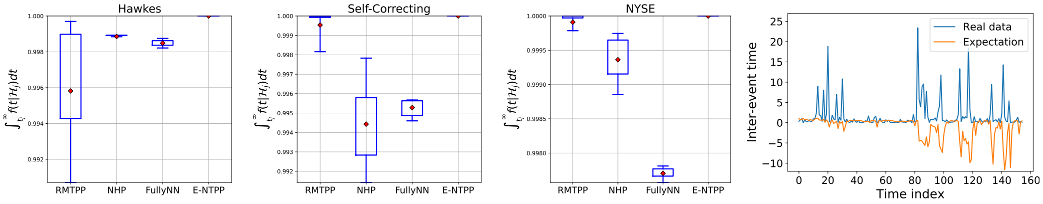
where w are the mixture weights, μ are the mixture means, and s are the standard deviations. Exactly, LogNormMix models the NTPP with density, which has more advantages in optimization of log-likelihood and expectation-based future prediction. However, it is inconvenient to get the intensity function and thus cause trouble to explore more information e.g. dimension/event influence. More importantly, the expressiveness of LogNormMix is limited, which fails to model some specific point processes such as Self-correcting point process. We discuss more about LogNormMix in Appendix A.

FullyNN (Omi et al., 2019) first uses a neural network to learn the intensity measure $\Lambda(t|\mathcal{H}_j)$ as defined in Eq. 2 in which the weights are all constrained to be positive to guarantee the network output is monotonically increasing w.r.t. the elapsed time t and utilizes autograd mechanism of deep learning framework to compute the intensity $\lambda(t|\mathcal{H}_j)$ to avoid numerical integration of the intensity.

$$\Lambda\left(t|\mathcal{H}_j\right) = \text{NN}\left(t - t_j, \text{RNN}(\mathcal{H}_j)\right), \quad \lambda\left(t|\mathcal{H}_j\right) = \frac{\partial}{\partial t}\Lambda\left(t|\mathcal{H}_j\right) = \frac{\partial}{\partial t}\text{NN}(t, \mathcal{H}_j). \tag{7}$$

In this setting, though the model enjoys the convenience brought by the universal approximation of neural network, directly modeling the intensity measure may have numerical flaws like the basic requirement for a TPP: $\Lambda(t_j^+|\mathcal{H}_j) = 0$, and also especially for conflicting the non-terminating assumption as will be detailed in Sec. 3.1 and further in Appendix B: the output of Λ network cannot automatically satisfy the basic necessary condition: $\Lambda(\infty|\mathcal{H}_j) \to \infty$.

Exactly, all of the above models exist their own advantages, but may fail at some parts of the perspective of expressiveness, predictability and interpretability. In this paper, we propose three independent parts in Sec. 3, 4 and 5. We hope our methods (especially predictability and interpretability parts) can be useful as orthogonal technologies for existing works. Table 1 summarizes and compares the popular TPP models in the three aspects, and also the combination readiness with our techniques.



(a) Hawkes Simulation (b) SC Simulation (c) NYSE Data (d) Inter-event time

Figure 1: **Left three:** compared with RMTPP, NHP and FullyNN, our E-NTPP (here only using the expressive module) can more effectively approximate the limit of integral density $\bar{F}$ in Eq. 8, namely the accumulated probability for next event occurrence is 1 for non-terminating TPP as embodied by Hawkes, Self-Correcting simulation data and NYSE real-word data. **Right most:** actual inter-event time $d_{j+1} = t_{j+1} - t_j$ in blue and its prediction $\hat{d}_{j+1} = \hat{t}_{j+1} - t_j$ with expectation predictions in orange by Eq. 9 for fitting Hawkes process by RMTPP. Note in this example, most prediction $\hat{d}_{j+1}$ are below zero i.e. the prediction is earlier than the current time and meaningless.

3 On Expressive and Sound Neural TPP

We first show that many existing NTPP models (Du et al., 2016; Mei & Eisner, 2017; Omi et al., 2019) still have modeling limitations especially for non-terminating processes i.e. there is always the occurrence of the next event. Table 1 gives an overview.

3.1 Soundness of Intensity Function

For designing a powerful and sound intensity function, we start with a motivating observation for the undesirable effects in the non-terminating process as encountered by many existing intensity models. As shown in Fig. 1(a), the event data is simulated from a Hawke process with $\mu = 0.2, \alpha = 0.8, \beta = 1$ being a non-terminating point process defined by Eq. 3, i.e. the next event must occur within a limited time period. However, as shown in Fig. 1(a), we can find that the density integral (i.e. the upper bound of the cumulative probability distribution function) is less than 1 for many methods: RMTPP, NHP, and FullyNN, and the case also holds for other data source from Self-Correcting process and real-world NYSE dataset as shown in Fig. 1. Specifically, the density integral is given by:

$$\bar{F} = \int_{t_j}^{\infty} f(t|\mathcal{H}_j)dt < 1, \tag{8}$$

It means these NTPPs inherently are unable to exactly learn the non-terminating sequences. Specifically, RMTPP with intensity in Eq. 4, can be strictly proved as a terminating point process when $w < 0$ (more details are discussed in appendix), which fails to learn the non-terminating TPP.

Besides, the corresponding expectation prediction exists the theoretical bias for predicting the future due to the inadequate integration of expectation. The expectation prediction can be written by:

$$\hat{t}_{j+1} = \int_{t_j}^{+\infty} t \cdot f(t|\mathcal{H}_j)dt, \tag{9}$$

We again take RMTPP as an example for prediction by Eq. 9 in Fig. 1(d), whereby the estimated next event time can be earlier than the current time: $\hat{t}_{j+1} < t_j$. The larger t_j is, the larger expectation bias will be. One way to mitigate this issue is to calculate the inter-event time directly instead of calculating $\hat{t}_{j+1}$ first (see details in Appendix C). It reduces bias yet is still a lower estimation.

Why we care about non-terminating for NTPP model? Most NTPP models are based on RNN or Transformer (Zuo et al., 2020) architecture, which inputs a sequence of events(for example $t_1, t_2, \ldots, t_n$) to model the intensity and the next event(i.e. $t_2, t_3, \ldots, t_{n+1}$) are use to get the likelihood. However, if the TPP is a terminating one (for example terminating at t_k), then rest inputs(i.e. $t_{k+1}, \ldots, t_n$) will become unreasonable for the model. Thus these RNN or Transformer based methods have a basic assumption of non-terminating requirements before the end event t_n.

To address all these issues, we propose a new NTPP intensity function, which is less human prior dependent and is shown can learn the non-terminating point process effectively.

3.2 Intensity Form Unspecified Neural Point Process

We herein propose our expressive neural point process, using a neural network to capture the arbitrary form of intensity function. The output of the intensity neural network is ensured to be positive by an activation function ζ e.g. Softplus as used in our experiments which is associated with dimension u:

$$\lambda_u(t|\mathcal{H}_j) = \zeta(NN_u(t - t_j, RNN(\mathcal{H}_j))), \tag{10}$$

where RNN$(\mathcal{H}_j)$ embeds the history information to the vector and NN$(\cdot,\cdot)$ capture the free form of intensity function with a fully connected network, which can fit all the forms of function theoretically. The summation of intensity over all dimensions is $\lambda(t|\mathcal{H}_j) = \sum_u \lambda_u(t|\mathcal{H}_j)$, which can be used to predict the next event time in Sec. 4 and we still apply the log-likelihood to optimize the neural networks as used in (Du et al., 2016; Mei & Eisner, 2017).

Different from the FullyNN, which designs the intensity measure directly and calculates the intensity function by the differentiation of the intensity measure, we design the intensity with free form and get the intensity measure with numerical integration. Thus we can avoid the drawback that $\Lambda(0|\mathcal{H}_j) \neq 0$ (recall the discussion in Sec. 2).

Models e.g. RMTPP are essentially terminating NTPP due to the design flaws of the intensity. So given the event sequences of non-terminating TPP (e.g. Hawkes), the intensity is not expressive enough, leading to $\bar{F} < 1$, which causes poor expectation prediction as shown in Fig. 1(d). In Fig. 1(a) 1(b) 1(c), E-NTPP outperforms when modeling non-terminating TPP ($\bar{F}$ is closer to 1).

4 On Sampling based Prediction for Neural TPP

Given a trained model e.g. E-NTPP, RMTPP etc., we propose a sampling based method to improve the predictability i.e. prediction ability. In particular, it is focused on the next event timing and interval estimation, but not event marker. The algorithm is shown in Appendix G.

4.1 Adaptive Prediction via Hyperparameter Tuning on Training Data

Most existing prediction methods (Du et al., 2016; Mei & Eisner, 2017; Shchur et al., 2020) use expectation prediction methods as shown in Eq. 9 to predict the next event. In this paper, we develop a prediction error metric driven sampling approach for event prediction over a time window. Note in this section, we only need a trained NTPP model with its intensity or intensity measure, then this technology can be applied as the sampling or prediction for the event sequences.

With the Time Change Theorem (Brown et al., 2002), the intensity measure $\Lambda(t_{j+1}|\mathcal{H}_j) = \int_{t_j}^{t_{j+1}} \lambda(\tau|\mathcal{H}_j)d\tau$ is exponentially distributed with the parameter 1 (see the details in Appendix D). Then given the current event time t_j and a parameter $u \sim \text{uniform}(0,1)$, we can easily derive:

$$\Lambda(t|\mathcal{H}_j) = \int_{t_j}^{t} \lambda(\tau|\mathcal{H}_j)d\tau = -\log(1-u), \tag{11}$$

where $-\log(1-u)$ is the sampling of the exponential distribution. The above formula paves the road to predicting time. At first, when the inverse of the intensity measure $\Lambda^{-1}(t|\mathcal{H}_j)$ exists, for some popular TPP models including both classic (Isham & Westcott, 1979) and neural models (Du et al., 2016), the event sampling can be efficiently performed in a closed-form with $\hat{t}^u_{j+1} = \Lambda^{-1}(t|\mathcal{H}_j)(-\log(1-u))$ given the event history $\mathcal{H}_j$. For example, with the intensity of RMTPP in Eq. 4, we can get an analytical solution with sampling methods (recall $u \sim \text{uniform}(0,1)$):

$$\hat{t}_{j+1}(u) = t_j + \frac{\log\left(\exp(l_j) - w\log(1-u)\right) - l_j}{w}, \tag{12}$$

where $l_j = \mathbf{v}^\top \mathbf{h}_j + b$. For other NTPP models such as Neural Hawkes (Mei & Eisner, 2017), FullyNN (Omi et al., 2019), we can get the sampling result by using a root finding method. Let $K(t) = \Lambda(t|\mathcal{H}_j) + \log(1-u)$, we cannot obtain the inverse of K directly like RMTPP. However, given a certain u, we can find that $K(0) < 0$, $K(\infty) \to \infty$ and $K'(t) = \lambda(t|\mathcal{H}_j) > 0$. Hence t has the unique solution for $K(t) = 0$, which can be find by the Newton's iterative method:

$$t^{k+1} = t^k - \frac{K(t^k)}{K'(t^k)} = t^k - \frac{\Lambda(t^k|\mathcal{H}_j) + \log(1-u)}{\lambda(t^k|\mathcal{H}_j)}. \tag{13}$$

Sometimes uncertainty prediction (i.e. sampling) is not enough for predicting. People may prefer a certain next event prediction instead of an uncertain one like expectation.

A simple way is to set the value $u = 0.5$ to calculate the median prediction as done in (Omi et al., 2019). However, the median prediction has been proven as a biased prediction such as Poisson process (Streit, 2010). So instead of setting the value $u = 0.5$, we can search u from the training data adaptively, to better fit the prediction metric (e.g. MSE) in testing or validation data.

Specifically, we assume that the training data and the testing data are independently and identically distributed. Then given the training data$\{t_j^{tr}\}_{j=1}^{N}$ and testing data $\{t_j^{te}\}_{j=1}^{M}$, we can estimate the adaptive value u_{ada} from training sequences, which is specified as:

$$u_{\text{ada}} = \arg\min_{u\in(0,1)} \sum_{j=1}^{N} \left(t_{j+1}^{tr} - \hat{t}_{j+1}^{tr}(u)\right)^2, \tag{14}$$

where $\hat{t}_{j+1}^{tr}(u)$ is the sampling results for next event time based on training sequence. As shown in Algorithm 1 in Appendix G, bisection search is used to obtain an optimal value u_{ada} to fit the behavior of training sequences in terms of an optimal u. Then for the testing sequences, we can simply predict the next event $\hat{t}_{j+1}^{te}$ as $\hat{t}_{j+1}^{te}(u_{\text{ada}})$, which is the MSE-guide prediction result.

Thus, given a trained NTPP model, we calculate the adaptive value u_{ada} from training sequences by Eq. 14, and get the adaptive predictions with Eq. 13 by setting $u = u_{\text{ada}}$ for testing sequences.

Remarks. Note our adaptive prediction is based on the Time Change (Brown et al., 2002) (see Theorem 1 in Appendix) and its derived inverse sampling method, and cannot use the popular thinning method (Guan & Loh, 2007) instead. In fact, the thinning method is more suitable to generate whole sequences from scratch instead of predicting next event, as it fails to get the 1-to-1 mapping between u and the next event, which needs multiple random values to sample the next event.

4.2 Further Discussion

In terms of predictability referred in this paper, predicting the time interval is also important if not more important than time point prediction: e.g. people may be interested in the probability p that the next event occurs in the interval $[\tau_1, \tau_2]$. Exactly, for the three values (p, τ_1, τ_2), if two of them are known, then the rest can be calculated with the above sampling method. Given the first event time τ_1 and the probability p, we have $\tau_2 = \hat{t}_{j+1}^{u_2}$ where u_2 can be specified as:

$$u_2 = p + \big(1 - \exp\left(-\Lambda(\tau_1|\mathcal{H}_j)\right)\big). \tag{15}$$

The proof is given in Appendix F. Here $1 - \exp\left(-\Lambda(\tau_1|\mathcal{H}_j)\right) = u_1$ refers to the first sampling value of τ_1, u_2 which calculated above refers to the second sampling value of τ_2 and Λ is the intensity measure defined in Eq. 2. Then the next event t_{j+1} occurs in $[\tau_1, \tau_2]$ with the probability $p = u_2 - u_1$. Thus given τ_1, p, we can calculate the rest parameter τ_2 which means that the next event happens in the interval $[\tau_1, \tau_2]$ with the probability p according to the model.

In practice, given the specific pair of sampling values (u_1, u_2), it is known for the corresponding probability p and interval $[\tau_1, \tau_2]$. Then for a NTPP model, we can estimates the interval $[\tau_1, \tau_2]$ given $p = u_1 - u_2$. Besides, for every event, we can also test the model by comparing the theoretical probability and the frequency of next event (i.e. $t_{j+1} \in [\tau_1, \tau_2]$ for every event).

Quantile Interval Estimation (QIE). Let $(u_1, u_2) = (0.05, 0.95), \quad (0.1, 0.9), \ldots, \quad (0.45, 0.55)$, which has the corresponding theoretical probability p as $0.9, 0.8, \ldots, 0.1$. Then we can do the estimation of $[\tau_1, \tau_2]$ and test the model with frequency of the events. Specifically, given the dataset $\{t_j\}_{j=1}^{J+1}$, the statistical frequency can calculated as

$$v_{QIE}(\alpha) = \frac{\sum_{j=1}^{J} 1\left(t_{j+1} \in (\hat{t}_{j+1}^{(\frac{\alpha}{2})}, \hat{t}_{j+1}^{(1-\frac{\alpha}{2})}]\right)}{J}, \tag{16}$$

where $1(\cdot)$ is the characteristic function and $\alpha = 1 - p$ is the significance level. By setting $u_1 = \frac{\alpha}{2}$ and $u_2 = 1 - \frac{\alpha}{2}$ where $\alpha = 0.1, 0.2, \ldots, 0.9$, we can compare the consistence of v_{QIE} and its corresponding probability p for model test. We give such a study result in experiments with different α in Fig. 7(a) and Fig. 7(b) in Appendix.

Compact Interval Estimation (CIE). We also discuss how to estimate the most compact i.e. shortest interval for a given probability p, which can be useful for time-sensitive decision making. However, compact interval is hard to estimate and obviously quantile interval may not be the most compact one. It is interesting to find that we can directly estimate compact interval as $[\hat{t}_{j+1}^{0}, \hat{t}_{j+1}^{(1-\alpha)}]$ when $w < 0$ for RMTPP as a special case. We discuss more in Appendix F.

5 On Counter-factual Interpretable Neural TPP

Now we provide a new perspective to show how the existing NTPP models can explain the point process with clear interpretability. So our method in this section can be view as a supplement to the interpretability of the previous works which lack the interpretability before. Similar with the method in Sec. 4, what we need is a trained NTPP model with its continuous intensity function.

5.1 Influence among Dimensions

We aim to establish a definition of influence between events and give quantitative estimation, which is often missing in neural TPP literature, and often only holds for specific TPP like Hawkes process.

Given the history $\mathcal{H}_{j-1}$, we explore what the event j influences at (current) time t on dimension u. We discuss the two exclusive cases separately: **i)** event j occur at time t_j, then we can get the rate of event occurrence as $\lambda_u(t|\mathcal{H}_j)$; **ii)** event j does not occur, then the rate of event occurrence at (current) time t is $\lambda_u(t|\mathcal{H}_{j-1})$. In the sense of counter-factual, between the occurrence and nonoccurrence of event j, we define the **influence** of event j by the difference of the two intensities:

$$I_u^j(t) = \lambda_u(t|\mathcal{H}_j) - \lambda_u(t|\mathcal{H}_{j-1}). \tag{17}$$

Then we show the rationality of the definition, by taking the example of traditional Hawkes processes, in which the concept of influence is often be used. The intensity of Hawkes processes can be specified as: $\lambda_u(t|\mathcal{H}_j) = \mu_u + \sum_{t_j<t} \alpha_{uu_j} e^{-\beta(t-t_j)}$. For $t = t_j$ in Eq. 17, the influence I_u^j is exactly as:

$$I_u^j(t_j) = \sum_{t_i<t_j^+} \alpha_{uu_i} e^{-\beta(t_j-t_i)} - \sum_{t_i<t_j} \alpha_{uu_i} e^{-\beta(t_j-t_i)} = \alpha_{uu_j}. \tag{18}$$

Thus $I_u^j(t_j)$ is exactly equal to the parameter of infectivity matrix α_{uu_j} in Hawkes process, which can be interpreted as the influence from dimension u_j to dimension u. Therefore, our definition in Eq. 17 is not contradictory to the 'influence' understood by the traditional TPP. Then for a trained NTPP model, we can estimate the influence between type u and u' with the mean of I_u^j on u' as

$$\hat{\alpha}_{uu'} = \sum_j I_u^j(t_j) / \sum_{j:u_j=u'} 1, \tag{19}$$

We call the matrix $\{\hat{\alpha}_{uu'}\}$ as infectivity matrix here. Different from Hawkes processes, the mean of $I_u^j(t_j)$ (i.e. $\hat{\alpha}_{uu'}$) is allowed to be negative which means the event j suppresses event on dimension u.

Besides, we propose another way to estimate $\hat{\alpha}_{uu'}$ given a trained NTPP, with intensity specified as $\tilde{\lambda}_u(t) = \lambda_u(t) + |c_u|$ where $\lambda_u(t)$ is the intensity of RMTPP and c_u is the trainable parameter. Note the above NTPP is in line with Hawkes processes and thus every parameter of Hawkes can be estimated (see Appendix K). Given the Hawkes simulation data, Fig. 5 shows the estimation is quite accurate, suggesting that the NTPP really understands Hawkes' generation mechanism.

5.2 Influence among Events

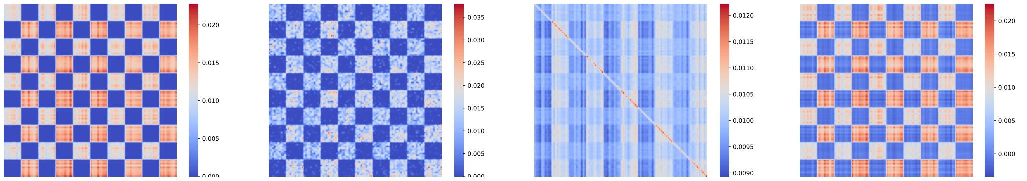
People are often interested in finding the triggering effects between temporal events. For example, which historical event results in the occurrence of the current event. In multi-dimensional Hawkes processes, there exist similar results. By using EM algorithm to estimate the parameters (Zhou et al., 2013), we can get a event triggering matrix $\{p_{ij}\}$ as specified as

$$p_{ij} = \frac{\alpha_{u_i u_j} g\left(t_i - t_j\right)}{\mu_{u_i} + \sum_{j=1}^{i-1} \alpha_{u_i u_j} g\left(t_i - t_j\right)}, \quad p_{ii} = \frac{\mu_{u_i}}{\mu_{u_i} + \sum_{j=1}^{i-1} \alpha_{u_i u_j} g\left(t_i - t_j\right)}, \tag{20}$$

where p_{ij} denotes the probability that event i is triggered by event j and p_{ii} represents the probability that event i is self-triggered. However, the triggering is mainly limited to the $\{\alpha_{uv}\}$ and the kernel $g(t)$, which can neither get a long dependence from history nor express the negative influence. By the definition of influence in Eq. 17, we propose the triggering/suppressing 'probability' matrix $\bar{p}_{ij}$:

$$\bar{p}_{ij} = \frac{\lambda_{u_i}(t_i|\mathcal{H}_{t_j}) - \lambda_{u_i}(t_i|\mathcal{H}_{t_{j-1}})}{\lambda_{u_i}(t_i|\mathcal{H}_{t_{i-1}})}, \quad \bar{p}_{ii} = 1 - \sum_{i \neq j} \bar{p}_{ij}. \tag{21}$$

Exactly $\lambda_{u_i}(t_i|\mathcal{H}_{t_j}) - \lambda_{u_i}(t_i|\mathcal{H}_{t_{j-1}}) = I_{u_i}^j(t_i)$, which means the influence of event j at time t_i on dimension u_i. Thus $\bar{p}_{ij}$ is the influence 'ratio' from event i to event j, which can evaluate the importance of event i given the occurrence of event j. Besides, $\{p_{ij}\}$ in Eq. 20 is a special case of $\bar{p}_{ij}$ given the Hawkes intensity, which can be interpreted as the impact of event i to event j.



(a) Ground Truth (b) ADM4 2013 (c) Attention 2017 (d) EI-NTPP

Figure 2: Comparison of infectivity matrix estimation for multi-dimensional Hawkes process.

Table 2: Negative log-likelihood and RMSE ($\times 10^{-4}$) results for synthetic and real-world data. Note the first, second and third columns are Expectation, Median and Adaptive prediction methods.

Synthetic dataset / Method	Hawkes				Self-Correcting				Renewal			
	NLL↓	RMSE↓			NLL↓	RMSE↓			NLL↓	RMSE↓		
		exp	med	adp		exp	med	adp		exp	med	adp
RMTPP (Du et al., 2016)	-2.004	780	806	777	-1.404	829	884	829	-1.782	975	1010	975
NHP (Mei & Eisner, 2017)	-2.019	771	804	771	-1.329	835	890	835	-1.849	975	1048	968
FullyNN (Omi et al., 2019)	-2.000	779	810	779	-1.391	838	870	838	-1.786	975	1013	978
LogNormMix (Shchur et al., 2020)	-2.015	777	-	-	-1.403	829	-	-	-1.882	970	-	-
EP-NTPP (ours)	-2.021	774	796	775	-1.415	817	827	823	-1.850	974	1043	971
Real dataset / Method	ATM				StackOverflow				NYSE			
	NLL↓	RMSE↓			NLL↓	RMSE↓			NLL↓	RMSE↓		
		exp	med	adp		exp	med	adp		exp	med	adp
RMTPP (Du et al., 2016)	-1.425	556	638	548	-0.448	648	663	655	-1.171	1502	1536	1503
NHP (Mei & Eisner, 2017)	-1.340	565	657	553	-0.323	643	675	648	-1.295	1542	1624	1545
FullyNN (Omi et al., 2019)	-1.733	556	2209	806	-0.395	644	672	643	-1.176	1522	1558	1527
LogNormMix (Shchur et al., 2020)	-1.408	674	-	-	-0.525	654	-	-	-1.243	1543	-	-
EP-NTPP (ours)	-1.988	545	537	537	-0.492	643	669	642	-1.347	1498	1592	1508

6 Experiments and Discussion

We fit and evaluate our various models on several simulation and real-world datasets. We apply Min-max normalization on all datasets before experiments to overcome the numerical overflow problem caused by exp operation in RMTPP (Du et al., 2016) and NHP (Mei & Eisner, 2017). Besides, median absolute deviation (MAD) based anomaly detection method (see details in Appendix I) is also applied for data splitting (partition sequences at large time intervals).

In the experiments, for evaluation of expressiveness, E-NTPP is used to compare with existing works such as RMTPP (Du et al., 2016), NHP (Mei & Eisner, 2017), FullyNN (Omi et al., 2019), and LogNormMix (Mei & Eisner, 2017) with negative log-likelihood (NLL). Besides, for the predictability, the adaptive prediction method (i.e. P-NTPP) is evaluated compared with median and expectation with root mean square error (RMSE). At last, for interpretability, the event and dimension influence (I-NTPP) is also estimated with counter-factual interpretable methods compared with ADM4 (Zhou et al., 2013) and attention-based method (Xiao et al., 2017).

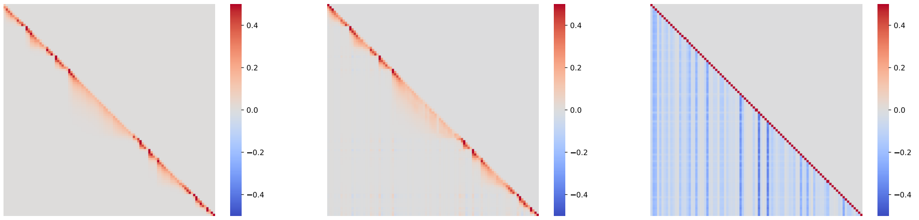
6.1 Protocols and Datasets

To show the effectiveness, we first test our proposed methods in synthetic data to show the expressiveness, predictability and interpretability of our approaches. The datasets are simulated with Hawkes processes, Self-Correcting processes (SC) and Renewal processes. The setting of the hyperparameter is shown in Appendix I. Specifically, 0.1 million events (640 sequences) are simulated for each generated dataset and we use 60% of them for training, 20% for validation and 20% for testing. In addition to NLL, we also evaluate by RMSE of expectation, median and adaptive prediction in Table 2. Besides, the interpretability is also evaluated to show our method's superiority.

1) NYSE. A book order from NYSE with 0.7 million transactions (Du et al., 2016). Each transaction contains time (in millisecond) and possible action (buy/sell). We cut sequence with 1,929,600 events for training and 482,400 for test. The action type is as a marker and we predict when an action occurs.

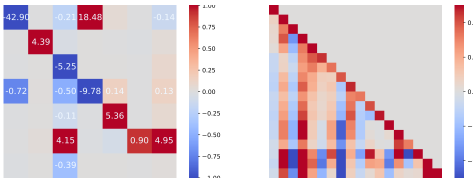
2) ATM. The dataset is mainly for the predictive ATM maintenance problem, which is comprised of the event logs involving error reporting and failure tickets (Xiao et al., 2017). The type of ATM has 2 main types 'ticket' and 'error' within 7 months in America. Besides the 'error' is divided into 6 subtypes: printer (PRT), cash dispenser module (CNG), internet data center (IDC), communication part (COMM), printer monitor (LMTP), miscellaneous. We treat them as 7 types to train the model.

3) Stack Overflow. It contains question answering records on Stack Overflow days (Paranjape et al., 2017). Every question answering event is recorded whose dimension refers to the user ID. We collect the data in a two-year period and treat the reward history of each user as the event sequence.



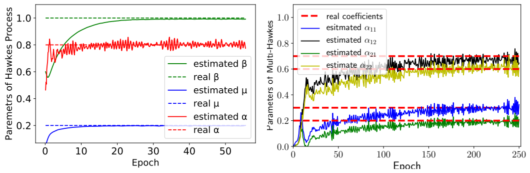
(a) ADM4(Hawkes) (b) EI-NTPP(Hawkes) (c) EI-NTPP(SC)

Figure 3: Event Influence by calculating $\{p_{ij}\}$ with EI-NTPP for Hawkes and Self-Correcting data.



(a) Dimension influence (b) Event influence

Figure 4: Influence estimation over event types and event records for ATM dataset.



(a) 1D-Hawkes estimation (b) 2D-Hawkes estimation

Figure 5: Parameters learning for 1-D and multi-dimensional Hawkes process via NTPP.

6.2 Experimental Results and Discussion

Results of NLL and RMSE of expectation. Table 2 shows the NLL and RMSE compared with different models. Our EP-NTPP outperforms on NLL and RMSE of expectation prediction on Hawkes, Self-Correcting simulation data and ATM, NYSE real-world data. Besides, in Renewal and StackoverFlow datasets, our EP-NTPP is also ranked in the top two of all models. For the synthetic data, we can find our EP-NTPP performs the top two among Hawkes, Self-Correcting and Renewal simulated datasets. Specifically, EP-NTPP models best with the lowest NLL and RMSE in Self-Correcting simulated sequences and has the best NLL in Hawkes simulations. For real-world data, EP-NTPP outperforms on ATM and NYSE datasets both w.r.t. NLL and RMSE metric.

Comparison of expectation, median and adaptive prediction. In particular, expectation, median prediction and adaptive prediction are also given in Table 2. The simplest median prediction used in (Omi et al., 2019) performs worst. Note Adaptive prediction outperforms in some cases, which has no concerns of bias. For the models like RMTPP and FullyNN, which can calculate the intensity measure without numerical integration, sampling based methods have more computational advantageous and do not need to calculate numerical integration twice, that is, calculating the first numerical integration to get $f(t|\mathcal{H}_j)$ and then calculating the second integration for expectation.

Performance of interpretability. Then we show that the efficacy of interpretability of EI-NTPP, which can capture the influence among dimensions and events. As shown in Fig. 2, we simulated multi-dimensional Hawkes processes with a particular infectivity matrix as the ground truth in Fig. 2(a). we can see that our method which calculates the influence among dimensions by Eq. 19 with EI-NTPP performs the best. For the influence among events, the results are shown in Fig. 3. We can find that the influence between events simulated by Hawkes processes is similar to the traditional method(i.e. ADM4), the calculated $\{p_{ij}\}$ value reflects the occurrence mechanism of events: self-generated or triggered in Hawkes. And for the evaluation of Self-Correcting simulated data, the red diagonal represents that the event occurrence are more likely to be self-generated and the historical events have negative effects on the occurrence of current events.

In ATM experiment, Fig. 4 shows a counter-factual-based interpretation model for neural TPP (event/dimension influence mining), which is new to attention-based TPP models. The influences of dimensions are calculated with E-NTPP proposed in Sec. 3.2, whose dimension types includes printer (PRT), cash dispenser module (CNG), internet data center (IDC), communication part (COMM), printer monitor (LMTP), miscellaneous, and ticket in Fig. 4(a). Fig. 4(b) reflects the mutual triggering or suppressing effects between events(i.e. $\{p_{ij}\}$), and show the long-term dependency of 100 events. In addition, accuracy of event type prediction is also reported on different datasets in Appendix H.

REFERENCES

Emery N Brown, Riccardo Barbieri, Valérie Ventura, Robert E Kass, and Loren M Frank. The time-rescaling theorem and its application to neural spike train data analysis. *Neural computation*, 14(2):325–346, 2002.

N. Du, H. Dai, R. Trivedi, U. Upadhyay, M. Gomez-Rodriguez, and L. Song. Recurrent marked temporal point processes: Embedding event history to vector. In *KDD*, 2016.

M. Eichler, R. Dahlhaus, and J. Dueck. Graphical modeling for multivariate hawkes processes with nonparametric link functions. *Journal of Time Series Analysis*, 2017.

J. L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

Yongtao Guan and Ji Meng Loh. A thinned block bootstrap variance estimation procedure for inhomogeneous spatial point patterns. *Journal of the American Statistical Association*, 102(480): 1377–1386, 2007.

A. Hawkes. Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 438–443, 1971.

Hengguang Huang, Hao Wang, and Brian Mak. Recurrent poisson process unit for speech recognition. In *AAAI*, 2019.

V. Isham and M. Westcott. A self-correcting point process. *Stochastic Processes and Their Applications*, 8(3):335–347, 1979.

Shuang Li, Shuai Xiao, Shixiang Zhu, Nan Du, Yao Xie, and Le Song. Learning temporal point processes via reinforcement learning. In *NIPS*, 2018.

T. J. Liniger. Multivariate hawkes processes. *PhD thesis, Swiss Federal Institute Of Technology, Zurich*, 2009.

H. Mei and J. Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. In *NIPS*, 2017.

Takahiro Omi, Naonori Ueda, and Kazuyuki Aihara. Fully neural network based model for general temporal point processes. In *NeurIPS*, 2019.

Ashwin Paranjape, Austin R. Benson, and Jure Leskovec. Motifs in temporal networks. In *WSDM*, 2017.

Jakob Gulddahl Rasmussen. Lecture notes: Temporal point processes and the conditional intensity function. *arXiv preprint arXiv:1806.00221*, 2018.

Oleksandr Shchur, Marin Biloš, and Stephan Günnemann. Intensity-free learning of temporal point processes. In *ICLR*, 2020.

Roy L Streit. *Poisson point processes: imaging, tracking, and sensing*. Springer Science & Business Media, 2010.

Kanae Tsushima, O. Chitil, and Joanna Sharrad. Type debugging with counter-factual type error messages using an existing type checker. 2020.

Yongqing Wang, Huawei Shen, Shenghua Liu, Jinhua Gao, and Xueqi Cheng. Cascade dynamics modeling with attention-based recurrent neural network. In *IJCAI*, 2017.

W. Wu, J. Yan, X. Yang, and H. Zha. Decoupled learning for factorial marked temporal point processes. In *KDD*, 2018.

S. Xiao, J. Yan, C. Li, B. Jin, X. Wang, X. Yang, S. Chu, and H. Zha. On modeling and predicting individual paper citation count over time. In *IJCAI*, 2016.

S. Xiao, J. Yan, X. Yang, H. Zha, and S. Chu. Modeling the intensity function of point process via recurrent neural networks. In *AAAI*, 2017.

Shuai Xiao, Junchi Yan, Mehrdad Farajtabar, L. Song, X. Yang, and H. Zha. Learning time series associated event sequences with recurrent point process networks. *IEEE Transactions on Neural Networks and Learning Systems*, 30:3124–3136, 2019.

H. Xu, W. Wu, S. Nemati, and H. Zha. Icu patient flow prediction via discriminative learning of mutually-correcting processes. *TKDE*, 2016.

K. Zhou, H. Zha, and L. Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *AISTATS*, 2013.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and H. Zha. Transformer hawkes process. In *ICML*, 2020.

Appendix

A The shortcomings of existing NTPP

Here we show some shortcomings for LogNormMix. First, though the intensity can be calculated with the density and survival functions, we can not get the closed-form intenisty for LogNormix. In fact, it depends on whether density can be integrated with a closed-form, and $F(t|\mathcal{H}_j)$ is not in closed-form for LogNormix. To see this, one can rewrite the intensity of LogNormix as

$$\lambda(t|\mathcal{H}_j) = \frac{f(t|\mathcal{H}_j)}{1 - F(t|\mathcal{H}_j)}$$

Here $F(t|\mathcal{H}_j)$ is CDF of Gaussian mixture, which can be specified as

$$F(t|\mathcal{H}_j) = \sum w_i \Phi_i(t)$$

where $\Phi_i(t)$ is CDF of Gaussian with no closed-form. Thus the intensity is not in closed-form.

Then due to the non-closed-form problem, LogNorMix can not get the continuous results of intensity function, which causes inaccuracy and high-cost problems for calculating every $\lambda(t_i)$ (t_i may be not the discrete-time point). Other intensity/intensity measure-based methods will not suffer from this problem.

Besides, due to the Log-Gaussian based form of density for LogNormMix, the corresponding intensity may be a restricted form. For example, when $t \to \infty$, the intensity is specified as

$$\lim_{t\to\infty} \lambda(t|\mathcal{H}_j) = \lim_{t\to\infty} \frac{f(t|\mathcal{H}_j)}{1 - F(t|\mathcal{H}_j)} = \lim_{t\to\infty} \frac{f'(t|\mathcal{H}_j)}{-f(t|\mathcal{H}_j)}$$

For simplicity, we use one LogNorm for $f(t|\mathcal{H}_j)$ (The mixed model is similar), which can be specified as

$$f(t|\mathcal{H}_j) = \frac{1}{\tau s\sqrt{2\pi}} \exp\left(-\frac{(\log\tau - \mu)^2}{2s^2}\right)$$

Here $\tau = t - t_j$. Then

$$\lim_{t\to\infty} \lambda(t|\mathcal{H}_j) = \lim_{t\to\infty} \frac{-\frac{1}{\tau} f(t|\mathcal{H}_j) + \frac{-2(\log\tau-\mu)}{2\tau s^2} f(t|\mathcal{H}_j)}{-f(t|\mathcal{H}_j)} = \lim_{t\to\infty} \frac{s^2 + (\log\tau - \mu)}{\tau s^2}$$

and use L'Hôpital's rule again:

$$\lim_{t\to\infty} \lambda(t|\mathcal{H}_j) = \lim_{t\to\infty} \frac{1}{\tau s^2} \to 0$$

So LogNormMix may be not a good way to capture the history information of all sequences such as Self-correcting based events because the corresponding intensity is **restricted** and **not free** enough.

B The Terminating Problem

For a recurrent-based model, there exists a basic assumption that the next event must occur, which is non-terminating for the point process. Thus the design of the intensity function requires theoretical basis to guarantee the mathematical properties as indicated in the following theorem.

Proposition 1 *A conditional intensity $\lambda(t|\mathcal{H}_j)$ uniquely defines a non-terminating point process if it satisfies the following conditions for any $\{t_1, t_2, \ldots, t_j\}$ and $t > t_j$: 1) $\lambda(t|\mathcal{H}_j)$ is non-negative and integrable in the interval $[t_j, \infty)$; 2) $\int_{t_j}^{t} \lambda^*(\tau)d\tau \to \infty$ for $t \to \infty$.*

The non-terminating point process are discussed in detail in (Rasmussen, 2018), which require that the upper bound of the cumulative probability function

$$\overline{F} = \int_{t_j}^{\infty} f^*(\tau)d\tau = 1 - \exp\left(-\int_{t_j}^{\infty} \lambda^*(\tau)d\tau\right), \tag{22}$$

tends to 1(i.e. $\overline{F} = 1$). However, several existing NTPP model may not satisfy the condition2 such as FullyNN (Omi et al., 2019) and RMTPP (Du et al., 2016), which leads to $\overline{F} < 1$ and thus do not meet the basic assumptions of recurrent-based point process model.

In detail, we take an example for RMTPP model, whose intensity is given in Eq. 4. To analyze the terminating problem, there exist three situations about the parameter w:

i) When $w > 0$, the intensity increases until the next event occurs, whose form is similar to Self-Correcting point process (Isham & Westcott, 1979) and satisfies Proposition 1 as non-terminating point process.

ii) When $w = 0$, the point process becomes the Recurrent Poisson Process with exponential link function as used in (Li et al., 2018; Huang et al., 2019) which satisfies Proposition 1 as a non-terminating point process.

iii) However, when $w < 0$, there is an issue with Eq. 4. Consider the integration of the conditional intensity function for the next event:

$$\lim_{t\to\infty}\int_{t_j}^{t}\lambda(\tau)d\tau = \lim_{t\to\infty}\frac{\exp\left(l_j + w(t-t_j)\right) - \exp(l_j)}{w} = -\frac{1}{w}\exp(l_j) < \infty, \tag{23}$$

where $l_j = \mathbf{v}^\top \mathbf{h}_j + b$. It means condition 2 of Proposition 1 is not satisfied. Thus when $w < 0$, the point process with intensity Eq. 4 may terminate at the current event for $\int_{t_j}^{\infty} f\left(t|\mathcal{H}_j\right)dt < 1$, which does not satisfy the implied assumptions in RNN model. Then for RMTPP model, we can rewrite Eq. 22 as

$$\overline{F} = 1 - \exp\left(\frac{1}{w}e^{l_j}\right) < 1, \tag{24}$$

which means that the point process continues with probability $\overline{F}$, and terminates with probability $1 - \overline{F}$. Exactly, most real-world event sequences are more likely to show the situation of $w < 0$ when Learning RMTPP model. Thus we should not easily ignore this terminating problem. Besides, there also exist this terminating problem for other more complex NTPP models, whose experimental results are given in Fig. 1.

C The Biased Prolem

The expectation prediction of next event can be written as

$$\hat{t}_{j+1} = \int_{t_j}^{+\infty} t \cdot f(t|\mathcal{H}_j)dt \tag{25}$$

However, the estimation above is biased because

$$\bar{F} = \int_{t_j}^{\infty} f(t|\mathcal{H}_j)dt < 1, \tag{26}$$

This problem can cause high bias prediction as shown in Fig. 1(d). The prediction of inter-event time $\hat{d}_j = \hat{t}_{j+1} - t_j$ are easily below zero meaning their prediction is earlier than the current time and meaning less given the small $\bar{F}$ and high current event t_j.Though the prediction Eq. 25 is used in their paper (Du et al., 2016; Mei & Eisner, 2017), most researchers use

$$\hat{d}_{j+1} = \int_{0}^{+\infty} \tau \cdot f(\tau + t_j|\mathcal{H}_j)d\tau \tag{27}$$

in their code to predict the inter-event time directly, which can mitigate the bias problem. We use this modified expectation prediction in Tab. 2. However, note this method still is a biased prediction method theoretically.

D Time Change Theorem

Here we rewrite the Time Change Theorem, which is used in Sec. 4.

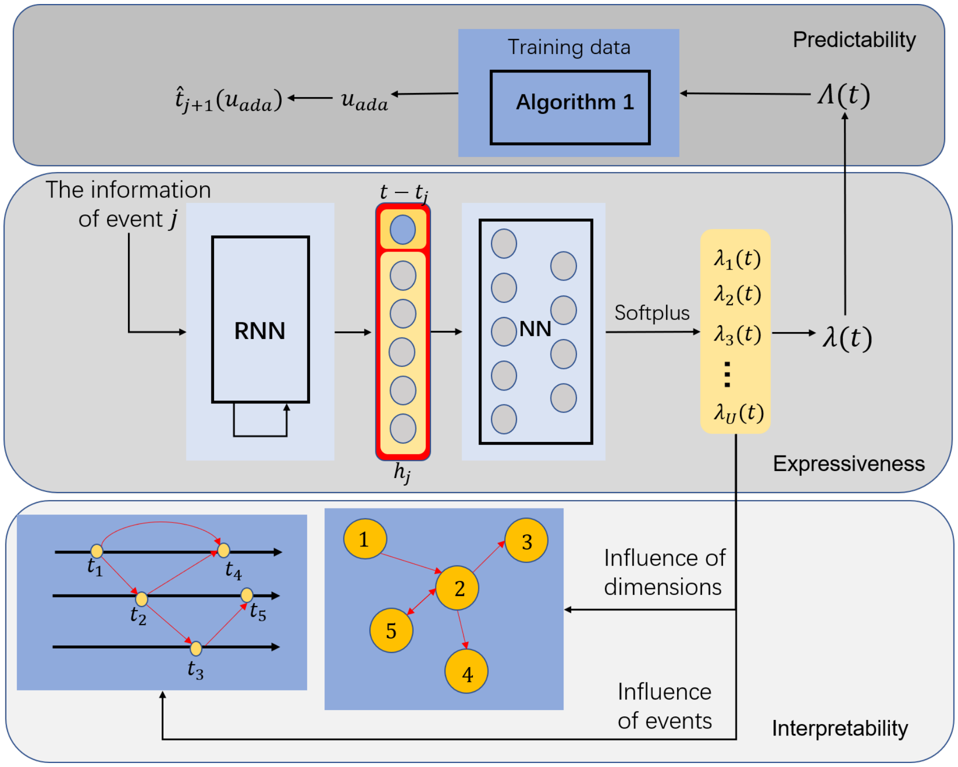


Figure 6: Overview of EPI-NTPP. We propose E-NTPP to improve the expressiveness, P-NTPP to improve the predictability and I-NTPP to improve the interpretability of neural TPP. These three components are orthogonal to each other, which can be used together to improve the overall performance.

Theorem 1 ***Time Change Theorem (TCT)*** *(Brown et al., 2002) For a temporal point process* $\{t_1, t_2, ..., t_j, ...\}$ *with conditional intensity* $\lambda(t|\mathcal{H}_j)$*, the integrated conditional intensity in* $(t_j, t_{j+1}]$*, i.e. intensity measure, has the form:*

$$\Lambda(t_{j+1}|\mathcal{H}_j) = \int_{t_j}^{t_{j+1}} \lambda^*(t)dt. \tag{28}$$

Then $\Lambda(t_j, t_{j+1})$ *obeys the exponential distribution with parameter* 1.

The theorem above reveals the essence of numerical value for intensity function and intensity measure. It is often used in point process based analysis such as QQ-plot for testing, Least Square (LS) loss etc.

E THE OVERVIEW OF OUR METHODS

We show the overview of our EPI-NTPP in Fig. 6. Note our three technologies(i.e. E-NTPP,P-NTPP and I-NTPP) are are orthogonal to each other. Note P-NTPP and I-NTPP can also be applied to other NTPP models such as RMTPP, NHP and FullyNN.

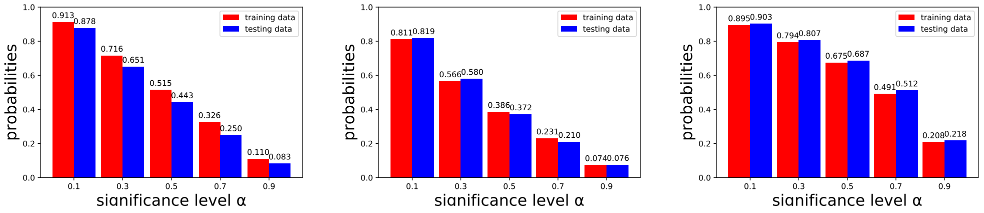
F SAMPLING METHOD FOR INTERVAL PREDICTION

Here we discuss more about Sampling Method for interval prediction of neural point process.

F.1 THE PROOF OF EQ. 15

It is evident about the probability p with $p = u_2 - u_1$. Due to Eq. 11, we can get that

$$u_1 = 1 - \exp\left(-\Lambda(\tau_1|\mathcal{H}_j)\right).$$



(a) Self-Correcting data for QIE (b) Hawkes process data for QIE (c) Hawkes process data for CIE

Figure 7: Quantile/Compact interval estimation (Q/CIE) for simulation data generated by Self-Correcting and Hawkes processes.

Then Eq. 15 can be derived as

$$u_2 = p + u_1 = p + (1 - \exp -\Lambda(\tau_1|\mathcal{H}_j))$$

F.2 Discussions about CIE

Besides, for CIE, we provide some discussions here. Given the density function, we have

$$f(t|\mathcal{H}_j) = \lambda(t|\mathcal{H}_j) \cdot \exp(-\int_{t_j}^{t} \lambda(\tau|\mathcal{H}_j))d\tau$$

then take the derivative of f w.r.t. t for RMTPP, we have

$$f'(t|\mathcal{H}_j) = \lambda'(t|\mathcal{H}_j) \cdot \exp(-\int_{t_j}^{t} \lambda(\tau|\mathcal{H}_j)d\tau) - \lambda^2(t|\mathcal{H}_j) \cdot \exp(-\int_{t_j}^{t} \lambda(\tau|\mathcal{H}_j)d\tau),$$

where $\lambda(t|\mathcal{H}_j) = \exp\left(l_j + w(t - t_j)\right)$ for RMTPP. And we can get that $\lambda'(t|\mathcal{H}_j) = w \cdot \lambda(t|\mathcal{H}_j)$, thus we have :

$$f'(t|\mathcal{H}_j) = f(t|\mathcal{H}_j) \cdot (w - \lambda(t|\mathcal{H}_j)) = f(t|\mathcal{H}_j) \cdot (w - \exp\left(l_j + w(t - t_j)\right)) \tag{29}$$

We rewrite Eq. 12 here for simplicity:

$$\hat{t}^{u}_{j+1} = t_j + \frac{\log\big(\exp(l_j) - w\log(1-u)\big) - l_j}{w}$$

By Eq. 29, if $w < 0$, then $f'\left(t|\mathcal{H}_j\right) < 0$. It means that the probability decrease with the increment of t. So given a fixed probability p or the significance α(note $p = 1 - \alpha$), the most compact interval must be $[t^{(0)}_{j+1}, \hat{t}^{(1-\alpha)}_{j+1}]$ where $t^{(0)}_{j+1} = t_j$ and $\hat{t}^{(1-\alpha)}_{j+1}$ can be calculated given significance α:

$$\widehat{t}^{(1-\alpha)}_{j+1} = t_j + \frac{\log\big(\exp(l_j) - w\log(\alpha)\big) - l_j}{w} \quad (w < 0). \tag{30}$$

Then we can test the model by its theoretical probability and statistical frequency that

$$v_{CIE}(\alpha) = \frac{\sum_{j=1}^{J} 1\left(t_{j+1} \in (\hat{t}^{(0)}_{j+1}, \hat{t}^{(1-\alpha)}_{j+1}]\right)}{J}. \tag{31}$$

However, for $w > 0$, $f\left(t|\mathcal{H}_j\right)' > 0$ if $w > \exp\left(l_j + w(t - t_j)\right)$ else $f\left(t|\mathcal{H}_j\right)' < 0$. It means the probability that an events occur will increase first and decrease after $t > l_j + (\log(w) - l_j)/w$. So we cannot calculate the compact interval directly.

Here we give simple experimental results of CIE and QIE with Hawkes simulation data as shown in Fig. 7. Given the significance level α, the frequency of real next event in the interval are shown in Fig. 7 an thus we can see the rationality of CIE and QIE, where the frequency is close to $1 - \alpha$. Note that the deviation of CIE is larger than QIE, which shows the expressive limitation of RMTPP.

G ALGOTITHM IN SEC. 4

Algorithm 1: MSE-guided adaptive predictions by calculating adaptive value

```
Input: the training data $\{t_1^{tr}, t_2^{tr}, \ldots, t_N^{tr}\}$, the testing data $\{t_1^{te}, t_2^{te}, \ldots, t_M^{te}\}$, and the trained
       parameters of NTPP model
Output: the adaptive value $u_{\mathrm{ada}}$ and the corresponding predicted sequences on testing data
        $\{\hat{t}_2^{te}, \hat{t}_3^{te}, \ldots, \hat{t}_{M+1}^{te}\}$
Initialize $u_1 = 0.1$, $u_2 = 0.9$, $\delta = 0.001$ and $\epsilon = 0.001$
calculate $\mathrm{dRMSE}_1 = \sum_{j=1}^{N} \left(t_{j+1}^{tr} - \hat{t}_{j+1}^{tr}(u_1)\right)^2 - \sum_{j=1}^{N} \left(t_{j+1}^{tr} - \hat{t}_{j+1}^{tr}(u_1 + \delta)\right)^2$
calculate $\mathrm{dRMSE}_2 = \sum_{j=1}^{N} \left(t_{j+1}^{tr} - \hat{t}_{j+1}^{tr}(u_2)\right)^2 - \sum_{j=1}^{N} \left(t_{j+1}^{tr} - \hat{t}_{j+1}^{tr}(u_2 + \delta)\right)^2$
/* Here we can check that $\mathrm{dRMSE}_1 \cdot \mathrm{dRMSE}_2 < 0$ */
$\bar{u} = (u_1 + u_2)/2$
while $u_2 - u_1 < \epsilon$ do
    calculate $\mathrm{dRMSE}_m = \sum_{j=1}^{N} \left(t_{j+1}^{tr} - \hat{t}_{j+1}^{tr}(\bar{u})\right)^2 - \sum_{j=1}^{N} \left(t_{j+1}^{tr} - \hat{t}_{j+1}^{tr}(\bar{u} + \delta)\right)^2$
    if $\mathrm{dRMSE}_1 \cdot \mathrm{dRMSE}_m < 0$ then
        $u_2 = \bar{u}$
        $\mathrm{dRMSE}_2 = \mathrm{dRMSE}_m$
    else if $\mathrm{dRMSE}_2 \cdot \mathrm{dRMSE}_m < 0$ then
        $u_1 = \bar{u}$
        $\mathrm{dRMSE}_1 = \mathrm{dRMSE}_m$
    else
        break
    $\bar{u} = (u_1 + u_2)/2$
the adaptive variable $u_{\mathrm{ada}} = \bar{u}$
for $j = 1, 2, \ldots, M$ do
    calculate $\hat{t}_{j+1}^{te} = \hat{t}_{j+1}^{te}(u_{\mathrm{ada}})$
```

H ACCURACY RESULTS OF TYPE PREDICTION

We further explicitly write the full multi-dim version for event type prediction (Mei & Eisner, 2017), which will be added in new version:

$$u_{j+1} = \arg\max_{u} \int_{t_j}^{\infty} \frac{\lambda_u(t|\mathcal{H}_j)}{\lambda(t|\mathcal{H}_j)} f(t|\mathcal{H}_j) dt,$$

For event type prediction, it uses the same sample of t values for each k in the argmax, which reduces sampling variance and also lets the model share the $\lambda_u(t|\mathcal{H}_j)$ and $f(t|\mathcal{H}_j)$ computations across all dimensions u. The above description is also generally discussed in Section 4 of [12], which is not the focus of this paper (as our prediction improvement is concerned with event time).

Here we compare the event type prediction accuracy on ATM and StackOverflow datasets (we also add another popular MIMIC dataset), and our method EP-NTPP slightly outperforms the state-of-the-art methods, which we think is owing to the expressive modeling part. Note our sampling-based prediction part relates only to time prediction and has nothing to do with type prediction. However, the expressive modeling part is related to likelihood fitting, which directly improves NLL, but not necessarily on event type prediction.

Table 3: ACC Results for real-world data.

Datasets	RMTPP	LogNormMix	FullyNN	NHP	EP-NTPP (ours)
ATM	87.58%	89.15%	89.42%	89.23%	89.94%
NYSE	64.32%	64.32%	64.54%	64.22%	64.54%
stackoverflow	45.25%	46.54%	46.55%	46.59%	46.72%
MIMIC	95.28%	94.79%	96.28%	96.52%	96.30%

I Training Details and Error Bars

In this appendix, we explain the details of experiments, including data preprocessing, hyperparameter selection, and error bars of the experiments in Sec. 6.

As mentioned in Sec. 6, in the data preprocessing stage, we do two things, we apply Min-max normalization on time interval values and utilize median absolute deviation(MAD) based anomaly detection method to split event sequences. In this setting, the order of magnitude of each dataset is unified in the range of 0 to 1 and the abnormal large time intervals are eliminated.

The routine of median absolute deviation(MAD) based anomaly detection algorithm is shown as:

```
Algorithm 2: Median Absolute Deviation (MAD) to find outliers
Input: the unsplit time interval sequence $seq = \{t_1, t_2, \dots, t_N\}$, the scaling factor $n$.
Output: indices of outliers in sequence.
Function MAD ($seq$, $n$):
    $median$ = get_median ($seq$)
    $deviations$ = get_abs ($seq$ - $median$)
    $mad$ = get_median ($deviations$)

    $indices$ = find(get_abs ($seq$ - $median$) < $n$ * $mad$)
    return $indices$
```

We implement our model EPI-NTPP with PyTorch. We use a single-layer LSTM as the RNN encoder, we set its hidden size to be 64. We use ReLU as the activation function of the intensity function network, the hidden size of this MLP module is also set to be 64. In the input layer, the marker embedding size is set to be 16. Mini-batch size is set to be 16 and Adam algorithm with learning rate 0.001 is used for optimization.

All the NTPP models including EPI-NTPP run on a single RTX-2080Ti (11GB) GPU and ADM4 (which we use as a baseline for infectivity matrix estimation in Sec. 6) runs on a core of Intel i9-7920X CPU @ 2.90GHz with 128GB RAM.

As shown in Fig. 8, the error bars shows that our model outperforms others especially in the aspect of **expressiveness**, a lower NLLLoss indicates that the intensity function we modeled captures more complicated patterns to fit the dataset.

J Details for Table 1

Comparison on Expressiveness. As is known, the intensity of RMTPP is exponential form and thus the expressiveness is limited. NHP and AttentionPP adopt the intensity with a link function after the exponential form, which improve the robustness. LogNormMix use the Gaussian-form density, which is not a exponential form-based model. However, as discuss in Appendix A, given any parameters of LogNormMix, it satisfies that $\lambda(t|\mathcal{H}_j) \to 0$, which results the failure of modeling some special TPP model such as Poisson and self-correcting TPP. Besides, EPI-NTPP and FullyNN are both NN-based intensity without a specified form theoretically. The difference is that FullyNN models on intensity integral (also called intensity measure), while EPI-NTPP models on the intensity directly.

Comparison on Predictability. As discussed in Sec. 3.1 and Appendix C, the expectation will suffer from biased problem due to inadequate integral for some NTPP model such as RMTPP, NHP, FullyNN and etc. Compared with them, LogNormMix is a non-terminating model which design the density directly and thus overcome this baised problem. Besides, the method of P-NTPP is also a strategy to solve this problem.

Comparison on Interpretability. Among the methods given in Table 1, only AttentionPP and our EPI-NTPP gives some results of interpretability. The difference is that AttentionPP is based on attention score to evaluate the influence, while our methods is based on the difference of intensity, which is a probabilistic based method. Note we can get some interpretable results with RMTPP in a sense. w in Eq. 4 can interpret by inhibiting ($w > 0$) or triggering($w < 0$) the occurrence of future events given the past.

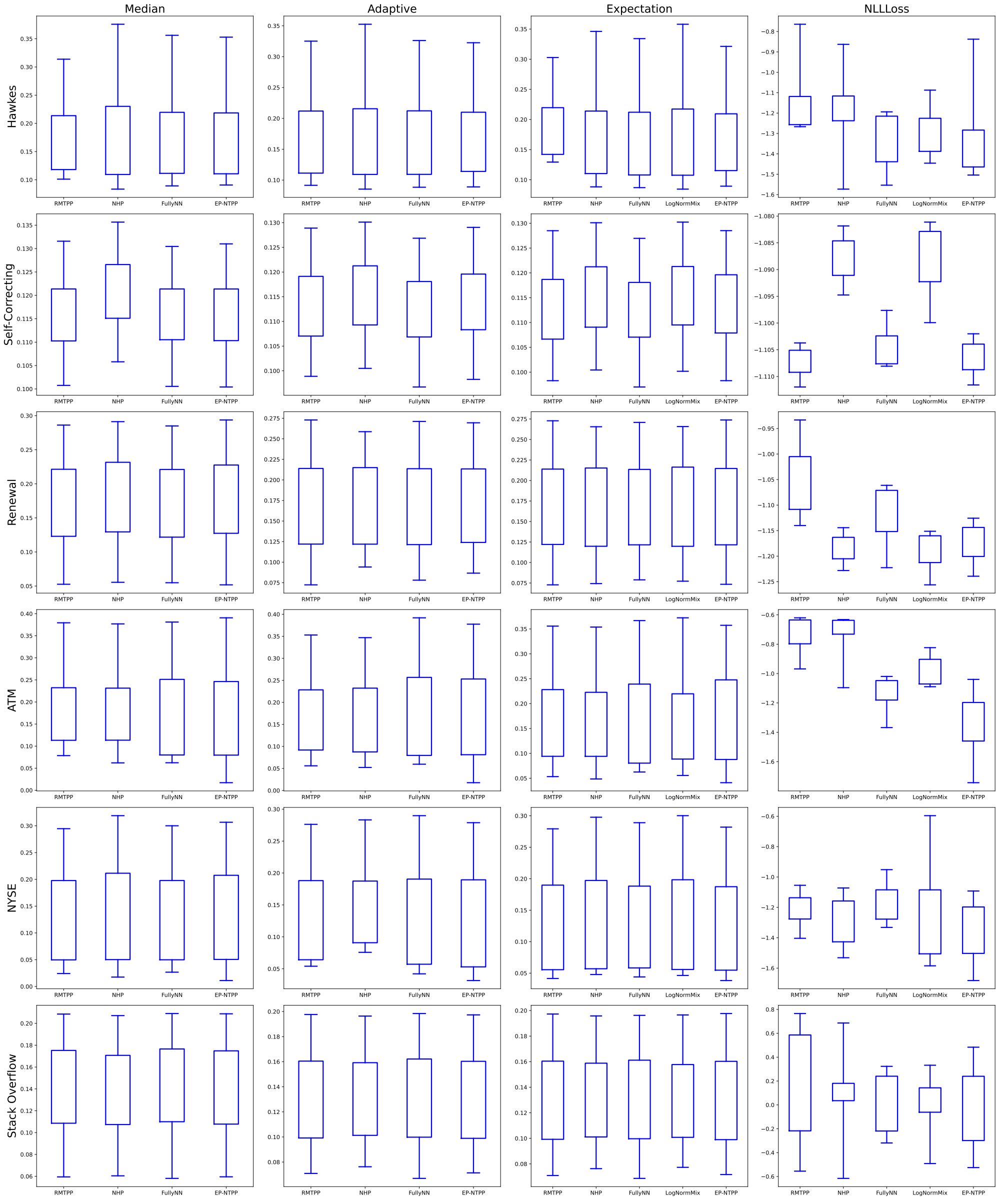


Figure 8: Error bars for RMSE results and negative log-likelihood on synthetic and real-world datasets. Median, adaptive and expectation predictions are shown in first three columns and the last column is for NLLLoss. These error bars are statistically obtained by testing models on a large number of different batches of data.

K A Simple Modification of RMTPP

As discussed in the end of Sec. 5, we modify the intensity for RMTPP Eq. 4 as follows,

$$\lambda(t|\mathcal{H}_j) = \exp\left(l_j + w(t - t_j)\right) + |c| \tag{32}$$

where $|c| > 0$ is the base intensity and $l_j = \mathbf{v}^\top \cdot \mathbf{h}_j + b$ is the accumulative past influence based on RNN. This leads to the following intensity measure,

$$\Lambda(t|\mathcal{H}_j) = \frac{1}{w}\exp\bigl(l_j + w(t - t_j)\bigr) - \frac{1}{w}\exp(l_j) + |c| \cdot (t - t_j) \tag{33}$$

When $t \to \infty$, the intensity measure $\Lambda(t_j, t) \to \infty$ for all values of w because $|c|(t - t_j) \to +\infty$ and $\frac{1}{w}\exp\bigl(l_j + w(t - t_j)\bigr) > 0$. Then the cumulative density function

$$F(t|\mathcal{H}_j) = \exp\left(-\Lambda(t|\mathcal{H}_j)\right) \tag{34}$$

tends to 1 when $t \to +\infty$, which solve the inadequate integral problem of terminating problem. Thus the modified RMTPP is a sound NTPP as a non-terminating TPP model. Then the conditional density function for next event can be written as

$$f(t|\mathcal{H}_j) = \big(\exp\left(l_j + w(t-t_j)\right) + |c|\big) \cdot \exp\left(\frac{1}{w}\exp(l_j) - \frac{1}{w}\exp\big(l_j + w(t-t_j)\big) - |c|(t-t_j)\right). \tag{35}$$

Thus we can estimate the expectation of next time as a time prediction with Eq. 9 with adequate integral.

Besides, the modification can expand to multi-dimensional case, which can be specified as

$$\lambda_d(t|\mathcal{H}_j) = \exp\left(l_j^d + w(t-t_j)\right) + |c_d| \tag{36}$$

where $d = 1, 2, \ldots, D$ and $l_j^d = \mathbf{v}_{:,d}^\top \cdot \mathbf{h}_j + b_d$ with the trainable parameter $\mathbf{v}_{:,d}$ and b_d. Then we show the method of estimating 1D and MD Hawkes parameters with the modified model.

With the modification above, one can estimate the parameters of Hawkes process while training. According to Eq. 32, one can easily find that the parameters of modified RMTPP can be in line with Hawkes process

$$\begin{cases} \hat{\mu} = |c|, \\ \hat{\beta} = -w, \\ \hat{\alpha} = \frac{1}{n}\sum_j \frac{\exp\left(l_j\right)}{\sum_{k=1}^{j}\exp\left(w \cdot (t_j - t_k)\right)}, \end{cases} \tag{37}$$

Fig. 5(a) is the result of evaluating 1D Hawkes parameters while training modified RMTPP. we can find that $\hat{\mu}, \hat{\beta}, \hat{\alpha}$ can get the convergence to real parameters. Note the parameter $\hat{\alpha}$ is oscillatory because of different batch inputs.

For a multi-dimensional case, the muti-Hawkes with its intensity can be specified as

$$\lambda_d(t|\mathcal{H}_j) = \mu_d + \sum_{t_k < t} \alpha_{dd_j} e^{-\beta(t-t_k)}, \tag{38}$$

Then μ_d and β is easily estimated as shown in Eq. 37. However, for the estimation of the matrix of $\{\alpha_{dd'}\}$, i.e. infectivity matrix, there exist several methods. In addition to the influence-based estimation proposed by Eq. 19 in the paper, we can also estimate from another view, that is, the least square method. Based on Eq. 36 and Eq. 38, we can get that

$$\sum_{k=1}^{j} \alpha_{dd_j} e^{w(t_j - t_k)} = \exp\left(l_j^d - wt_j\right) \tag{39}$$

which is the linear equation about $\alpha_{dd'}$. Then linear regression can be used to estimate $\alpha_{dd'}$ based on least square method. So we can estimated infectivity matrix which people may be interested in. Fig. 5(b) shows the results of multi-dimensional case.