HAWKES PROCESSES MODELING, INFERENCE AND CONTROL: AN OVERVIEW*

RAFAEL LIMA†

Abstract. Hawkes Processes are a type of point process which model self-excitement among time events. It has been used in a myriad of applications, ranging from finance and earthquakes to crime rates and social network activity analysis. Recently, a surge of different tools and algorithms have showed their way up to top-tier Machine Learning conferences. This work aims to give a broad view of the recent advances on the Hawkes Processes modeling and inference to a newcomer to the field. The parametric, nonparametric, Deep Learning and Reinforcement Learning approaches are broadly discussed, along with the current research challenges on the topic and the real-world limitations of each approach. Illustrative application examples in the modeling of Retweeting behaviour, Earthquake aftershock occurence and COVID-19 spreading are also briefly discussed.

Key words. Hawkes Processes, Point Processes, Machine Learning

AMS subject classifications. 68T99, 62M20, 60G55

1. Introduction.

Point Processes are tools for modeling the arrival of time events. They have been broadly used to model both natural and social phenomena related to arrival of events in a continuous time-setting, such as the queueing of customers in a given store, the arrival of earthquake aftershocks[59, 29], the failure of machines at a factory, the request of packages over a communication network, and the death of citizens in Ancient societies [16].

Predicting, and thus being able to effectively intervene in all these phenomena is of huge commercial and/or societal value, and thus there has been an intensive investigation of the theoretical foundations of this area.

Hawkes Processes (HP) [26] are a type of point process which models self- and mutual-excitation, i.e., when the arrival of an event makes future events more likely to happen. They are suitable for capturing epidemic, clustering, and faddish behaviour on social and natural time-varying phenomena. The excitation effect is represented by an additional function to the intensity of the process (i.e., the expected arrival rate of events): the triggering kernel, which quantifies the influence of events of a given process in the self- and mutual triggering of its associated intensity functions. Much of the Hawkes Processes' research has been devoted to modeling the triggering kernels, handling issues of scalability to large number of concurrent processes and quantity of data, as well as speed and tractability of the inference procedure.

Regarding the learning of the triggering kernels, one of the methods involves the assumption that it can be defined by simple parametric functions, s.a. one or multiple exponentials, Gaussians, Rayleigh, Mittag-Leffler [10] functions and power-laws. Much of the work dealing with this type of approach concerned with enriching these parametric models [39, 79, 86, 85, 19], scaling them for high dimensions, i.e., multivariate processes of large dimensions [5, 41], dealing with distortions related to restrictions on the type of available data [87], and proposing adversarial losses as a complement to the simple Maximum Likelihood Estimation (MLE) [89].

Another way of learning the triggering function is by assuming that it is represented by a finite grid, in which the triggering remains constant along each of its

*Preprint. Work in Progress.

Funding: This work was not supported by any organization.

†Samsung R&D Institute Brazil, Campinas-SP, Brazil (rafael.goncalves.lima at gmail.com).

subintervals. Regarding this piecewise constant (or non-parametric) approach, most notably developed in [55, 6], the focus has been on speeding up the inference through parallelization and online learning of the model parameters [91, 1].

A more recent approach, enabled by the rise to prominence of Deep Learning models and techniques, which accompany the increase of computational power and availability of data of the recent years, regards modeling the causal triggering effect through the use of neural network models, most notably RNNS, LSTMs and GANs. These models allow for less bias and more flexibility than the parametric models for the modeling of the triggering kernel, while taking advantage of the numerous training and modeling techniques developed by the booming connectionist community.

In addition, regarding the control of self-exciting point processes,i.e., the modification of the process parameters towards more desirable configurations, while taking into account an associated 'control cost' to the magnitude of these modifications, recent works either make use of Dynamic Programming (Continuous Hamilton-Jacobi-Bellman Equation-based approach) [94, 93], Kullback-Leibler Divergence penalization (a.k.a. 'Information Bottleneck') [78], and Reinforcement Learning-based, as well as Imitation Learning-based techniques [75, 44].

Although there have been some interesting reviews and tutorials regarding domain-specific applications of Hawkes Processes in Finance [27, 5] and Social Networks [65], a broad view of the inference and modeling approachs is still lacking. A close work to ours is the one presented in [88] which, although very insightful, lacks coverage of important advances, such as the previously mentioned control approaches, as well as the richer variants of Neural Network-based models. Furthermore, a concise coverage of the broader class of Temporal Point Processes is given in [67], while reviews for parametric spatiotemporal formulations of HP are given in [63] and [92].

In the following, we introduce the mathematical definitions involving HPs, then carefully describe the advances on each of the aforementioned approachs, then finish by a summarization, along with some considerations.

2. Theoretical Background. In this and the next section, we present the mathematical definitions used throughout the remaining sections of the paper. The Hawkes Processes were originally introduced in [26] and [25].

In the present work, we restrain ourselves to the Marked Temporal Point Processes (MTPP), i.e., the point process in which each event is defined by a time coordinate and a mark (or label). An intuitive example of a MTPP is shown in Figure 1.

The key definitions are those of: Counting Process, Intensity Function, Triggering Kernel, Impact Matrix, (Log-)likelihood, Covariance, Bartlet Spectrum, Higher-order Moments and Branching Structure.

2.1. Multivariate Marked Temporal Point Processes. Realizations of univariate MTPPs, here referred to by $\mathcal{S}$ are one or more sequences of events e_i, each composed by the time coordinate t_k and the mark m_k, s.a.:

$$\mathcal{S} = \{(t_0, m_0), ..., (t_S, m_S)\}, \tag{2.1}$$

where S is the total number of events. Marks may represent, for example, a specific user in a social network or a specific geographic location, among others. For more complex problems, as in the check-in times prediction of [90], a composite mark may represent an user of interest and a specific location.

An easy way to generalize this notation would be to refer to multiple realizations of multivariate MTPPs as $\mathcal{S} = \{\mathcal{S}_{i,j}\}$, where i would refer to the dimension of the

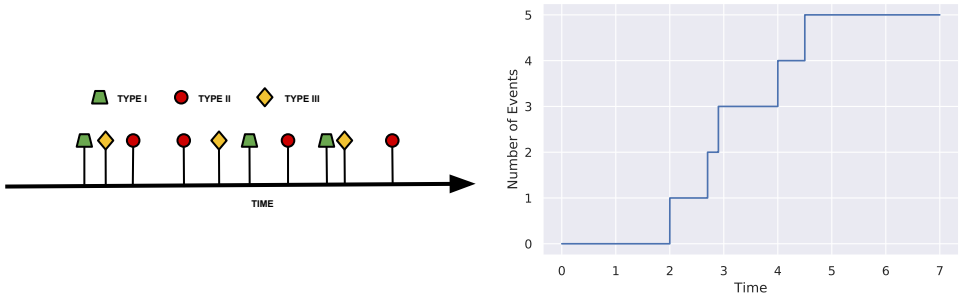

Fig. 1: Left: Intuitive diagram of a Marked Temporal Point Process (MTPP) with three types of event (marks). Right: Example of a counting process on the time interval [0,7].

process, while j would refer to the index of the sequence. Now, regarding only the purely temporal portion of the process, i.e., the time coordinates t_k, it is also common to express them by means of a Counting Process $N(t)$, which is simply the cumulative number of event arrivals up to time t:

$$\int_{0^-}^{t} dN_s, \tag{2.2}$$

where:

- $dN_{t_k} = 1$, if there is an event at t_k;
- $dN_{t_k} = 0$, otherwise.

This is illustrated in Figure 1.

Associated to each temporal point process, there is an Intensity Function, which is the expected rate of arrival of events:

$$\lambda(t)dt = E\left\{dN_t = 1\right\}, \tag{2.3}$$

which may depend or not on the history of past events. Such dependence results in a so-called "Conditional Intensity Function" (CIF):

$$\lambda(t)dt = E\left\{dN_t = 1|\mathcal{H}\right\}, \tag{2.4}$$

where $\mathcal{H}$ is the history of all events up to time t:

$$\mathcal{H} : \{t_{i,j} \in \mathcal{S}|t_{i,j} < t\} \tag{2.5}$$

This concept will be further discussed in the next section.

2.2. Hawkes Processes. The simplest example of a temporal point process is the Homogeneous Poisson Process (HPP), in which the intensity is a positive constant:

$$\lambda(t) = \mu, \tag{2.6}$$

for $\mu \in \mathbb{R}^+$.

In the case of an Inhomogeneous Poisson Process (IPP), the intensity $\lambda(t)$ is allowed to vary. Both HPP and IPP are shown in Figure 2.

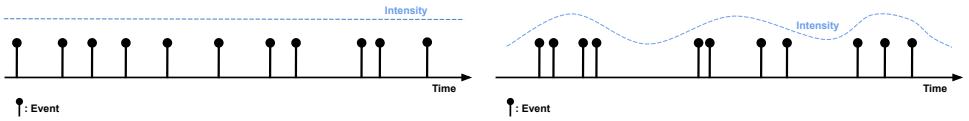

Fig. 2: Illustrative examples of Homogeneous (left) and Inhomogeneous (right) Poisson Processes. The events are represented by black dots.

Both the Homogeneous and the Inhomogeneous Poisson Processes have, in common, the fact that each consecutive event interval sampled from the Intensity Function is independent of the previous ones. When analyzing several natural phenomena, one may wish to model how events in each dimension i of the process, which may be representing a specific social network user, an earthquake shock at a given geographical region, or a percentual jump in the price of a given stock, just to cite a few examples - affect the arrival of events in all the dimensions of the process, including its own.

In particular, we are interested in the cases where the arrival of one event makes further events more likely to happen, which is reflected as an increase in the value of the intensity function after the time of said event. When this increase happens to the Intensity Function of the same i-th dimension of the event, the effect is denominated self-excitation. When the increase happens to the intensity of other dimensions, we refer to it as mutual excitation.

Hawkes Processes (HP) model self-excitation in an analytical expression for the intensity through the insertion of an extra term, which is designed to capture the effect of all the previous events of the process in the current value of the CIF. For a univariate Hawkes, we have:

$$\lambda_{HP}(t) = \underbrace{\mu}_{\text{baseline intensity}} + \underbrace{\sum_{t_i<t} \phi(t-t_i)}_{\text{self-excitation term}} \quad , \tag{2.7}$$

while, for the multivariate case, with dimension D, we are going to have both self-excitation ($\phi_{ii}(t)$) and mutual-excitation terms ($\phi_{ij}(t)$, s.t. $(i \neq j)$):

$$\lambda^i_{HP}(t) = \mu_i + \sum_{j=1}^{D} \sum_{t_{ij}<t} \phi_{ij}(t-t_{ij}). \tag{2.8}$$

The assumptions of

1. Causality:
$$\phi(t) = 0 \qquad \forall t < 0$$
2. Positivity:
$$\phi(t) \geq 0 \qquad \forall t \geq 0$$

are usually held for all $\phi_{ij}(t)$ [1].

In the case that the kernel matrix $\mathbf{\Phi}(t) = [\phi_{ij}(t)]^{d,n}_{i,j=0}$ can be factored into $\mathbf{\Phi}(t) = \boldsymbol{\alpha} \odot \boldsymbol{\kappa}(t)$, with $\boldsymbol{\alpha} = [\alpha_{ij}]^{d,n}_{i,j=0}$ and $\boldsymbol{\kappa}(t) = [\kappa_{ij}(t)]^{d,n}_{i,j=0}$, where "$\odot$" correspond to the Hadamard (element-wise) product.

[1] The case in which $\phi(t) < 0$ for $t \geq 0$ is referred to as an "Inhibiting Process", and not usually considered in Hawkes Processes works.

$\boldsymbol{\alpha}$, here denominated Impact Matrix, can implicitly capture a myriad of different patterns of self- and mutual excitation, as exemplified in Figure 3. This factorization is particularly convenient when adding penalization terms related to network properties to the loglikelihood-based loss of a Multivariate HP, such as in the works of [86] and [48].

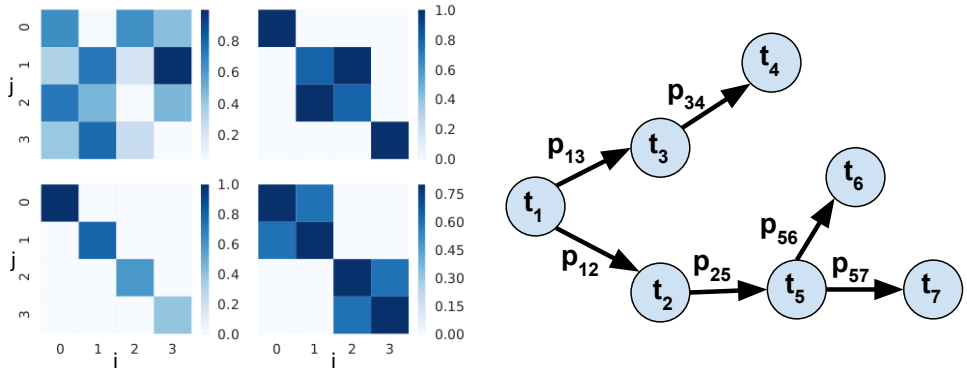

Fig. 3: Left: Four examples of 4×4 Impact Matrices $\boldsymbol{\alpha}$. Each $\alpha_{ij}(t)$ has the corresponding value indicated according to the color scale on the side. Right: Illustrative example of the concept of Branching Structure. A given edge $t_i \rightarrow t_j$ means that t_i triggered t_j with the probability $p_j i$.

For being of practical value, realizations of HPs are constrained to having finite number of events for any sub-interval of the simulation horizon $[0, T]$. This corresponds to having the kernel function (or kernel matrix) satisfying the following stationarity condition:

$$\boldsymbol{Spr}\left(||\boldsymbol{\Phi}(t)||\right) = \boldsymbol{Spr}\left(\{||\phi_{ij}(t)||\}_{1 \leq i,j \leq D}\right) < 1, \tag{2.9}$$

where $||\phi(t)||$ corresponds to $||\phi(t)|| = \int_0^\infty \phi(t)dt$ and $\boldsymbol{Spr}(\cdot)$ corresponds to the spectral radius of the matrix, i.e., the largest value among its eigenvalues.

If this stationarity condition is satisfied, we have that the process will reach weakly stationary state, i.e., when the properties of the process, most notably its "moments", vary only as a function of the relative distance, here referred to as "τ", of its points.

The first order moment, or statistics of the HP, is defined as:

$$\Lambda_i = E\{\lambda^i_{HP}(t)\} = \lim_{T \to \infty} \frac{1}{T} \int_0^T \lambda^i_{HP}(t)dt = (\mathbb{I} - ||\boldsymbol{\Phi}(\boldsymbol{t})||)^{-1}\mu_i \tag{2.10}$$

while the second-order statistics, or stationary covariance, is defined as:

$$\nu^{ij}(t' - t)dtdt' = E\{dN^i_t dN^j_{t'}\} - \Lambda_i \Lambda_j dtdt' - \epsilon_{ij}\Lambda_i \delta(t' - t)dt, \tag{2.11}$$

where ϵ_{ij} is 1, if $i = j$, and 0, otherwise, while $\delta(t)$ refers to the Dirac delta distribution.

The Fourier Transform of this stationary covariance is referred to as *Bartlett Spectrum*. Sometimes, a different transform, the Laplace Transform, is used for the same

purpose. In Hawkes' seminal paper [26], high importance is given to the fact that, when assuming some specific parametric functions for the excitation matrix $\boldsymbol{\Phi(t)}$, it is possible to find simple formulas for the covariance of process in the frequency domain. One example is the univariate case for $\phi(t)$ defined as the frequently used "parametric exponential kernel": $\phi(t) = \alpha e^{-\beta t}$, for $\alpha, \beta \in \mathbb{R}$.

For this choice, we have:

$$\nu^*(s) = \mathcal{L}\{\nu\}(s) = \frac{\alpha\mu(2\beta - \alpha)}{2(\beta - \alpha)(s + \beta - \alpha)} \qquad (s \in \mathbb{C}) \tag{2.12}$$

where $\mathcal{L}\{\cdot\}(s)$ refers to the Laplace Transform [2]. The detailed steps of this computation can be found in [26].

Going beyond the first- and second-order statistics, it is also possible to define statistics of higher orders, see [47, 16]. Although they become less and less intuitive and tractable, as their order increases, the work in [1] makes use of 3rd-order statistics, K^{ijk}, in a specific application of the Generalized Method of Moments for learning the impact matrix of multivariate HPs. It is defined as:

$$\begin{aligned} K^{ijk}dt = \int\int_{\tau,\tau' \in \mathbb{R}^2} \Big(& \mathbb{E}(dN_t^i dN_{t+\tau}^j dN_{t+\tau'}^k) \\ & -2\mathbb{E}(dN_t^i)\mathbb{E}(dN_{t+\tau}^j)\mathbb{E}(dN_{t+\tau'}^k) - \mathbb{E}(dN_t^i dN_{t+\tau}^j)\mathbb{E}(dN_{t+\tau'}^k) \\ & -\mathbb{E}(dN_t^i dN_{t+\tau'}^k)\mathbb{E}(dN_{t+\tau}^j) - \mathbb{E}(dN_{t+\tau}^j dN_{t+\tau'}^k)\mathbb{E}(dN_t^i) \Big), \end{aligned} \tag{2.13}$$

for $1 \leq i, j, k \leq D$, and it is connected to the skewness of N_t.

Now, regardless of the function family chosen for modeling $\boldsymbol{\Phi(t)}$ and $\boldsymbol{\mu}$, its fitness will be computed by measuring its likelihood over a set of sequences similar to the set of sequences used for training the model. Let be a set $\boldsymbol{\mathcal{S}}$ of M sequences, each with a total number of N_j events, considered over the interval $[0, T]$, such that:

$$\boldsymbol{\mathcal{S}} = \{\mathcal{S}^j\}_{j=1}^M = \{\Big[(t_1^j, m_0^j), ..., (t_{N_j}^j, m_{N_j}^j)\Big]\}_{j=1}^M, \tag{2.14}$$

with $m_j^k \in \{1, 2, ..., D\}$, $\forall j, k \in \mathbb{Z}_+$.

And let be a family $\mathbb{F}$ of Multivariate HPs with dimension $D \geq 1$ and parametric exponential kernels assumed for the shape of the excitation functions, such that the CIF $\lambda_i^j(t)$ of each i-th node defined over the sequence $\mathcal{S}^j$ is given by [3]:

$$\lambda_i^j(t) = \mu_i + \sum_{t_k^j \leq t} \alpha_{m_k^j i} \mathrm{e}^{-\beta_{m_k^j i}(t - t_k^j)} \qquad (t_k^j \in \mathcal{S}^j) \tag{2.15}$$

Given parameter vectors

$$\boldsymbol{\mu} = \{\mu_m\}_{m=1}^D \in \mathbb{R}_+^D \qquad \theta = \{(\alpha_{mn}, \beta_{mn})\}_{m=1,n=1}^{D,D} \in \mathbb{R}_+^{2D^2},$$

the likelihood function given in the logarithmic form, i.e., the loglikelihood, of a multivariate HP over a set $\mathcal{S}$ of M sequences considered over the interval $[0, T]$, is

[2] The Laplace Transform $\mathcal{L}\{f\}(s)$ of a function $f(t)$ defined for $t \geq 0$ is computed as $\mathcal{L}\{f\}(s) = \int_{0^-}^{\infty} f(t)\mathrm{e}^{-st}$, for some $s \in \mathbb{C}$.

[3] Equivalent definitions of $\mathbb{F}$ can be given to families of HPs defined by other types of HPs, such as those with power-law kernels, or those with the corresponding CIF modeled by a recurrent neural network.

given by:

$$llh_{\mathcal{S}}(\boldsymbol{\mu}, \theta, \mathbb{F}) = \sum_{j=1}^{M} \left(\sum_{i=1}^{D} \sum_{k \in N_j} \log \lambda_i^j(t_k^j) - \underbrace{\sum_{i=1}^{D} \int_0^T \lambda_i^j(t) dt}_{\text{Compensator}} \right) \tag{2.16}$$

Thus, the goal of learning a HP over $\mathcal{S}$, is the act of finding vectors $\boldsymbol{\mu}$ and θ s.t.:

$$(\boldsymbol{\mu}, \theta) = \operatorname{argmax} llh_{\mathcal{S}}(\boldsymbol{\mu}, \theta, \mathbb{F}) \tag{2.17}$$

A more rigorous and complete derivation of Equation 2.17 can be found in [7].

Another concept which is of relevance in some inference methods is that of a Branching Structure ($\mathcal{B}$). It defines the ancestry of each event in a given sequence, i.e., specifies the probability that the i-th event t_i was caused by the effect of a preceding event t_j in the CIF (p_{ji}, for $0 \leq i < n$) or by the baseline intensity μ (p_{0i}). The probabilities p_{ji} and $p_0 i$ can be given by:

$$p_{ji} = \frac{\phi(t_i - t_j)}{\lambda(t_i)}, \text{ for } (j \geq 1) \quad \text{and} \quad p_{0i} = \frac{\mu}{\lambda(t_i)}. \tag{2.18}$$

As an example, consider the example illustrated in Figure 3, in which event t_1, the first of the event series, causes events t_2 and t_3; event t_2 causes event t_5; event t_3 causes event t_4; and event t_5 causes events t_6 and t_7. The corresponding Branching Structure $\mathcal{B}_E$ implied by these relations among the events has an associated probability given by:

$$p(\mathcal{B}_E) = p_{01} * p_{12} * p_{13} * p_{25} * p_{34} * p_{56} * p_{67} \tag{2.19}$$

2.3. Simulation Algorithms. Regarding the experimental aspect of HPs, synthetic data may be generated through the following methods:

1. **Ogata's Modified Thinning Algorithm** [58]: It starts by sampling the first event at time t_0 from the baseline intensity. Then, each posterior event t_i is obtained by sampling it from a HPP with intensity fixed as the value calculated at t_{i-1}, and then:
   - Accepting it with probability $\frac{\lambda(t_i)}{\lambda^*(t_{i-1})}$, where $\lambda^*(t_{i-1})$ is the value of the intensity at time t_{i-1}, while λ_{t_i} is the value calculated through Equation 2.7.
   - Or rejecting it and proceed to resampling a posterior event candidate;
2. **Perfect Simulation** [56]: It derives from the fact the HP may be seen as a superposition of Poisson Processes. It proceeds by sampling events from the baseline intensity, taken as the initial level, and then sampling levels of descendant events for each of the events sampled at the previous level. From each event $t_{0,i}$ sampled from the baseline intensity, we associate an IPP with intensity defined as $\phi(t - t_{0,i})$, and then sample its descendant events. Next, we take each descendant event sampled and also associate it with its corresponding IPP, and so on, until all the levels were explored over the simulation horizon $[0, T]$.

More detailed, step-by-step descriptions of each of these two algorithms are shown in pseudocode format, in Algorithm 2.1 and Algorithm 2.2, both for the case of

Algorithm 2.1 Ogata's Modified Thinning Algorithm (Univariate Case)

Input μ, $\phi(t)$, T
Define t = 0
Sample t_1 from exponential distribution with rate μ
Update $t = t + t_1$
Define $n = 1$
while $t < T$ **do**
  $\lambda_n = \mu + \sum_{i=1}^{n} \phi(t - t_n)$
  Sample t_{n+1} from exponential distribution with rate λ_n
  $\lambda_{n+1} = \mu + \sum_{i=1}^{n} \phi(t + t_{n+1} - t_n)$
  Sample u from Uniform Distribution over $[0, 1]$
  if $\frac{\lambda_{n+1}}{\lambda_n} < u$ **then**
    Update $t = t + t_{n+1}$
    Update $n = n + 1$
  end if
end while
return $\{t_i\}_{i=1}^{n}$

Algorithm 2.2 Perfect Simulation of Hawkes Processes (Univariate Case)

Input μ, $\phi(t)$, T
Define j = 0
Simulate HPP with $\lambda = \mu$ over $[0, T]$ to obtain $\{t_j^i\}_{i=1}^{n_j}$
while $\exists t_j^i \quad (\forall i \leq n_j)$ **do**
  for $(i = 1 \; ; \; i \leq n_j \; ; \; i++)$ **do**
    Simulate IPP with $\lambda = \phi(t - t_j^i)$ over $t \in [0, T]$ to obtain $\{t_{(j+1),k}^i\}_{k=1}^{n_{j+1}^i}$
  end for
  Update $n_{j+1} = \sum_{i=1}^{n_j} n_{j+1}^i$
  Update $\{t_{(j+1)}^i\}_{i=1}^{n_{j+1}} = \bigcup_{i=1}^{n_{j+1}} \{t_{(j+1),k}^i\}_{k=1}^{n_{j+1}^i}$
  Update $j = j + 1$
end while
return $\bigcup_{l=0}^{j} \{t_l^i\}_{i=1}^{n_l}$ (After sorting)

Univariate HPs. In the next section, we focus on the HP models which assume simple parametric forms for the excitation functions, along with its variants, which we will refer to as "Parametric HPs".

3. Parametric HPs. In the present section, we discuss the HP models which assume simple parametric forms for the excitation functions. Figure 4 shows some examples of commonly used functions for parametric HPs. Much has been done recently in terms of incrementing these models for dealing with specific aspects of some domains, such as social networks [98, 39], audio streaming [79], medical check-ups [87], among others. In the following subsections, we aim to give a broad view of how the parametric modeling of HPs are used and improved throughout a series of possible ideas. We divided the recent research on parametric HPs as focusing on three different strategies, accompanied by working examples. The referred strategies are:

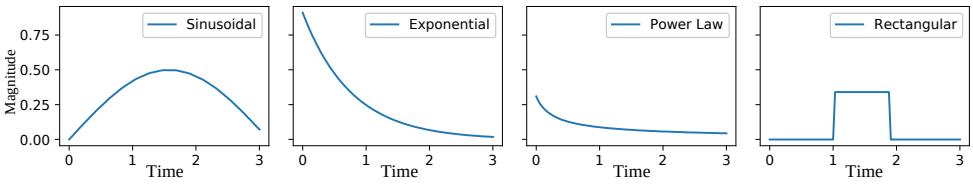

Fig. 4: Four examples of parametric HP kernels ($\phi(t)$). Each of them is used to model a different type of interaction among events of a given HP.

1. Enhancing and composing simple parametric kernels, to adapt the model to specific modeling situations and datasets;
2. Improving scalability of parametric HP models, to the multivariate cases with many nodes and sequences with many jumps;
3. Improving robustness of training over worst-case scenarios and defective data.

Further examples on each strategy are also briefly mentioned in Section 11.

3.1. Enhanced and Composite Triggering Kernels. As a way of modeling the daily oscillations of the triggering effects on Twitter data, [39] proposes a time-varying excitation function for HPs. The probability $\mathbb{P}$ of getting a retweet over the time interval $[t, t+\delta t]$, with small δ, is modeled as:

$$\mathbb{P}(\text{Retweet in } [t, t+\delta t]) = \lambda(t)\delta t, \tag{3.1}$$

in which the time-dependent rate is dependent on previous events as:

$$\lambda(t) = p(t) \sum_{t_i < t} d_i \phi(t - t_i), \tag{3.2}$$

with $p(t)$ being the infectiousness rate, t_i as the time corresponding to the i-th retweet arrival, and d_i as the number of followers of the i-th retweeting individual.

Furthermore, the memory kernel $\phi(s)$, a probability distribution for the time intervals between a tweet by the followee and its retweet by the follower, has been shown to be heavily tailed in a variety of social networks [98]. It is fitted to the empirical data by the function:

$$\phi(s) = \begin{cases} 0, & \text{for } s < 0 \\ c_0, & \text{for } 0 \le s \le s_0 \\ c_0 (s/s_0)^{-(1+\theta)}, & \text{for } s > s_0 \end{cases}$$

where the parameters c_0, s_0 and θ are known.

The model is defined so that the daily cycles of human activity are naturally translated into cycles of retweet activity. The time dependence of the infectious rate is, therefore, defined as:

$$p(t) = p_0 \left\{ 1 - r_0 \sin\left(\frac{2\pi}{T_m}(t + \phi_0) \right) \right\} \sqrt[\tau_m]{e^{-(t-t_0)}} \tag{3.3}$$

The parameters p_0, r_0, ϕ_0 and τ_m correspond to the intensity, the relative amplitude of the oscillation, its phase, and the characteristic time of popularity decay

respectively. Those are fitted through a Least Square Error (LSE) minimization procedure over the aggregation of retweet events over time bins δt.

Another improvement over the traditional parametric froms for HPs involve the addition of a nonlinearity on the expression for the CIF:

$$\lambda_{HP}(t) = g\left(\mu + \sum_{t_i<t} \phi(t - t_i)\right), \tag{3.4}$$

in which $g(\cdot)$ correspond to a so-called *link function*, e.g., sigmoid:

$$g(x) = \frac{1}{1 + e^{-x}}. \tag{3.5}$$

The work in [79] proposes a procedure for simultaneously learning $g(\cdot)$, μ and $\phi(t) = \alpha\kappa(t)$, with $\kappa(t)$ taken as e^{-t}, for assuring convergence, of the algorithm.

The procedure is formulated using a moment-matching idea over a piecewise-constant approximation for $g(\cdot)$, which leads to the definition of the objective function as a summation:

$$\min_{g\in\mathcal{G},\boldsymbol{W}} \frac{1}{n}\sum_{i=1}^{n}\left(N_i - \int_0^{t_i} g(w \cdot x_t)dt\right)^2, \tag{3.6}$$

with $w = (\mu, \alpha)^T$, and $x_t = (1, \sum_{t_i\in\mathcal{H}_t} \kappa(t - t_i))^T$.

The algorithm is run by recursively updating the estimates $\hat{w}$, with the Isotron Algorithm (see [35]), and $\hat{g}$, with a projected gradient descent step.

Theoretical bounds for the approximation error of the method are also given, along with extensions of the algorithm for general point processes, monotonically decreasing nonlinearities, low-rank processes and multidimensional HPs.

Another possible enhancement for modeling the parametric HP is through using a composition of Gaussian kernels of different bandwidths, as in [86]. The core idea is that the maximum nonzero frequency component of the kernel is bounded as the same value of the intensity function, since this one is simply a weighted sum of the basis functions. Therefore, for every value of the tolerance ξ, it is possible to find a frequency value ω_0 s.t.:

$$\int_{\omega_0}^{\infty} |\hat{\lambda}(\omega)|d\omega \leqslant \xi \tag{3.7}$$

From this ω_0, the method then defines the triggering function $\phi(t)$ as a composition of $\tilde{D}_\phi$ Gaussian functions, with $\tilde{D}_\phi$ equally spaced values of bandwidth over the interval $[0, \omega_0]$.

The estimate of the value of the intensity function, for transforming into the frequency domain, is done through a kernel density estimation with gaussian kernels of bandwidth fixed as the Silverman's rule of thumb:

$$h = \sqrt[5]{\left(\frac{4\hat{\sigma}^5}{3n}\right)} \approx \frac{1.06\hat{\sigma}}{\sqrt[5]{n}}, \tag{3.8}$$

where $\hat{\sigma}$ is the standard deviation of the time intervals, and n is the number of events of a given sequence.

After the Gaussian functions are defined, it remains to estimate the model coefficients Θ, with the impact matrix, now an impact tensor $A = \{\alpha_{ijk}\}$, through a

convex surrogate loss penalized by parameters related to the sparsity of the matrix, the temporal sparsity of the kernels, and the pairwise similarity:

$$\underset{\Theta \succcurlyeq 0}{\operatorname{argmin}} -\mathcal{L}_\Theta + \gamma_{\mathcal{S}}||A||_1 + \gamma_{\mathcal{G}}||A||_{1,2} + \gamma_{\mathcal{P}}E(A), \tag{3.9}$$

where:

- $||A||_1 = \sum_{i,j,k} |\alpha_{ijk}|$ is the l1-norm of the tensor, which is related to its temporal sparsity, which causes the excitation functions to go to zero at infinity, therefore maintaining the stability of the process;
- $||A||_{1,2} = \sum_{i,j} ||\{\alpha_{ij1}, ..., \alpha_{ij\tilde{D}_\phi}\}||_2$ is related to the sparsity over the $\tilde{D}_\phi$ basis functions of a given node of the process, and it enforces the local independence of the process;
- $E(A) = \sum_i \sum_{i' \in \mathcal{C}_i} ||\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_{i'}||_F^2 + ||\boldsymbol{\alpha}^i - \boldsymbol{\alpha}^{i'}||_F^2$ is a coefficient to enforce the pairwise similarity of the process, in which $\mathcal{C}_i$ corresponds to the cluster to which node i belongs, $||\cdot||_F$ is the Frobenius norm, $\boldsymbol{\alpha}_i = \{\alpha_{ijk}\}$, for fixed i, and $\boldsymbol{\alpha}^i = \{\alpha_{ijk}\}$, for fixed j. It means that, if i and i' are similar types of events, then their mutual excitation effects should be similar as well;
- $\gamma_{\mathcal{S}}$, $\gamma_{\mathcal{G}}$ and $\gamma_{\mathcal{P}}$ are coefficients to be tuned for the model.

The estimation is done through an Expectation-Maximization procedure close to those of [55] and [101], which first randomly initializes the impact tensor and the vector of baseline intensities $\boldsymbol{\mu}$, and then iterates through:

1. Estimating the probability that each event was generating by each of the compositional basis kernels, as well as the baseline intensity;
2. Averaging the probabilities over all events of all training sequences for updating the coefficients of each basis function and the baseline intensity.

The two steps are repeated until convergence of the parameter estimates.

3.2. Scalability. Another setting in which the parametric choice of kernels is highly convenient is with respect to the scalabity of the inference procedure for high dimensional networks and sequences with large number of events, which occur in several domains, such as social interactions data, which is simultaneously large (i.e., large number of posts), high-dimensional (numerous users) and structured (i.e., the users interactions are not in a random fashion but, instead, present some regularities).

One interesting inference method towards this direction is the work presented in [41], which achieves a complexity $O(nD)$, with D as the number of events comprised by the process history, and D as the dimension of the impact matrix of the process.

The referred method, entitled *Scalable Low-Rank Hawkes Processes* (SLRHP), takes advantage of the memoryless property of the exponential and the underlying regularity of large networks connected to social events: The memoryless property, which means that, in HPs with exponential excitation functions, the effect of all past events over the intensity value of a given point can be computed by just the time of the last event before said point, speeds up the intensity computing portion of the inference procedure iterations, while the underlying regularity of large impact matrices associated with the social phenomena allow the dynamics of large-dimensional HPs to be captured by impact matrices of much smaller magnitudes.

The baseline rates and excitation functions of model are then defined using a low-rank approximation:

$$\mu_i = (t) = \sum_{j=1}^{E} P_{ij}\tilde{\mu}_j \tag{3.10}$$

$$\phi_{mi}(t) = \sum_{j,l=1}^{E} P_{ij} P_{ml} \tilde{\phi}_{lj}(t), \tag{3.11}$$

in which $P \in \mathbb{R}_{+}^{D \times E}$ is a projection matrix from the original D-dimensional space to a low-dimensional space E ($E << D$). This projection can also be seen as a low-rank approximation of the excitation function matrix $\mathbf{\Phi}$, in which:

$$\mathbf{\Phi} = P \tilde{\mathbf{\Phi}} P^T \tag{3.12}$$

Through having that $E << D$, the formulated low-rank approximated inference algorithm SLRHP manages to:

1. Capture a simplified underlying regularity impose inferred intensity rates' parameters by adopting sparsity-inducing constraints to the model parameters;
2. Lower the number of parameters for both the baseline rates and excitation kernels, with the D natural rates and D^2 triggering kernels are lowered to r and E^2, respectively. This advantage is diminished slightly by the additional cost of inferring the $(D \times E)$-sized projection matrix P.

Another way of dealing with scalability issues of multivariate HPs, both in terms of the total number of events in the sequences and the number of nodes, is through the mean-field treatment, as described in [4]. Compared with the SLRHP method, which focuses on reducing the dimensionality of underlying network, the mean-field treatment focuses on finding closed-form expressions for approximate estimations involved in the optimization defined over the network in its real size. The key step of this method is to consider that the arrival intensities of each node of the process is wide-sense stationary, which implies the stability condition for the excitation matrix, and fluctuates only slightly around its mean value.

This last assumption, entitled *Mean-Field Hypothesis*, posits that, if $\lambda^i(t)$ corresponds to the intensity of the i-th node, and $\tilde{\Lambda}^i$ corresponds to the empirical estimator of the first-order statistics of said node,

$$\tilde{\Lambda}^i = \frac{N_T^i}{T}, \tag{3.13}$$

where N_T^i corresponds to the total number of events arrived at node i up to the final time of the simulation horizon $[0, T]$, then we have that:

$$\frac{|\lambda^i(t) - \tilde{\Lambda}^i|}{\tilde{\Lambda}^i} << 1 \qquad \forall t \in [0, T] \tag{3.14}$$

The condition defined by Equation 3.14 is met when: (i) $||\phi(t)|| << 1$, independently of the shape of $\phi(t)$; (ii) when the dimensionality of the MHP is sufficiently high; and also (iii) when $\phi(t)$ changes sufficiently slowly, so that the influence of past events average to a near constant value.

From this, we can recover the parameters θ^i from the intensity function λ_t^i, and the intensity function from the first-order statistics $\tilde{\Lambda^i}$, as:

$$\log \lambda_t^i \simeq \log \tilde{\Lambda}^i + \frac{\lambda^i(t) - \tilde{\Lambda}^i}{\tilde{\Lambda}^i} - \frac{(\lambda^i(t) - \tilde{\Lambda}^i)^2}{2(\tilde{\Lambda}^i)^2} \tag{3.15}$$

and

$$\lambda^i(t) = \mu^i + \int_{0^-}^{t} \sum_{j=1}^{D} \phi^{ji}(t) dN_t^i \tag{3.16}$$

The method yields mean-field estimates for the parameters with error which decays proportionally to the inverse of the final time T of the sequences:

$$\mathbb{E}(\theta^i) \approx \theta^i_{MF} \tag{3.17}$$

$$\boldsymbol{cov}(\theta^{ji}, \theta^{j'i'}) \sim \frac{1}{T}, \tag{3.18}$$

where θ^i_{MF} refers to the mean-field estimator of the parameters.

3.3. Training. Elaborating further on some difficulties of inferring parametric kernels from real-data, an interesting method regarding truncated sequences is described in [87].

In the case of learning HP parameters from real-data, one often has to deal with sequences which are only partially observed, i.e., the time event arrivals are only available over a finite time-window.

This poses a challenge concerning the robustness of the learning algorithms, since the triggering pattern from unobserved events are not considered: The inference deals with the error induced by computing intensity values over finite time windows, i.e., by computing intensity values along simulation horizons $[0, T]$, the closed-form equation would assume that its value at 0 is simply the baseline rate μ.

From the expression for the CIF:

$$\lambda_{HP}(t) = \mu + \sum_{t_i<t} \phi(t - t_i), \tag{3.19}$$

for $t \in [0, T]$, we have that, taking some value $T' \in [0, T]$, we may split the triggering effect term in two parts:

$$\lambda_{HP}(t) = \mu + \sum_{t_i<T'} \phi(t - t_i) + \sum_{T'\leq t_i<T} \phi(t - t_i), \tag{3.20}$$

for $t \in [T', T]$.

If we are observing the sequences only over the interval $[T', T]$, the second term is implicitly ignored, what may have severe degradation of the learning procedure, specially in the case that the excitation functions decay slowly.

The method proposed handles this issue through a *sequence-stitching method*. The trick is to sample "candidate predecessor events" and choosing the most likely one from their similarity w.r.t. the observed events. The augmented sequences would then be use for the actual HP parameters' learning algorithm.

In practice, this means that, given M sequences $\mathcal{S}_m$ realized over $[T', T]$, the method would **not** learn from the regular MLE formula

$$\theta^* = \underset{\theta}{\operatorname{argmax}}\, llh(\{\mathcal{S}_m\}_{m=1}^M, \theta), \tag{3.21}$$

but, instead, from some expression which takes into account the expected influence of unobserved predecessor events

$$\theta^*_{SDC} = \underset{\theta}{\operatorname{argmax}}\, \mathbb{E}_{s\ \mathcal{H}_{T'}} llh([\mathcal{S}, \mathcal{S}_m], \theta), \tag{3.22}$$

where $\mathcal{H}_{T'}$ corresponds to the distribution over all possible sequences of events happening before time T'. In practice, this expectation is computed not over the real

distribution, but over some finite number of (relatively few) samples, such as 5 or 10. This finite sample approximation converts Equation 3.22 onto

$$\theta^*_{SDC} = \underset{\theta}{\operatorname{argmax}} \sum_{\mathcal{S}_{stitch} \in \mathcal{K}} p(\mathcal{S}_{stitch}) llh(\mathcal{S}_{stitch}, \theta), \tag{3.23}$$

where $p(\mathcal{S}_{stitch})$ is the probability of the stitched sequence obtained from the concatenation of the original observed sequence and one of the sample predecessor candidate sequence. This probability is obtained by normalizing over similarity values over the candidate sequences obtained from some similarity function $\mathbb{S}(\cdot, \cdot)$ of the form:

$$\mathbb{S}(\mathcal{S}_k, \mathcal{S}) = \sqrt[\psi]{\mathrm{e}^{-||f(\mathcal{S}_k) - f(\mathcal{S})||^2}}, \tag{3.24}$$

where $f(\cdot)$ is some feature of the event sequence, and $\psi \in \mathbb{R}^+$ is some scale parameter. By fixing the excitation pattern matrix $\boldsymbol{\kappa}$ as composed of exponentials $\kappa(t) = e^{-\beta t}$ and imposing sparsity constraint $||\boldsymbol{\alpha}||_1 = \sum_{i,j} |\alpha_{ij}|$ over the impact matrix, the equivalent problem,

$$\theta^* = \underset{\mu \geqslant 0, \boldsymbol{\alpha} \succ 0}{\operatorname{argmax}} \sum_{\mathcal{S}_{stitch} \in \mathcal{K}} p(\mathcal{S}_{stitch}) llh(\mathcal{S}_{stitch}, \theta) + \gamma ||\boldsymbol{\alpha}||_1 \tag{3.25}$$

is shown to be solved through Expectation-Maximization updated for both μ and $\boldsymbol{\alpha}$. The method is more suitable for slowly-decaying excitation patterns, in which the influence of the unobserved events would be more prominent. In the case of exponentials with large values for the decay factor β, the improvement margins mostly vanish.

Another interesting improvement regarding the training procedure of MLE for HP parametric functions involves the complementary use of adversarial and discriminative learning, as in [89]. Although adversarial training has gained an ever increasing relevance for neural-network based models in the last years, due to the popularization of Generative Adversarial Networks [22] and its variants, as will be discussed further in Section 5, keeping the assumption of a simple parametric shape for the excitation function is a way to insert domain-specific knowledge in the inference procedure.

The key idea of this complementary training is that, while the discriminative loss, here defined as the Mean-Squared Error (MSE) between discretized versions of predicted and real sequences, would direct the parameter updates towards smoother prediction curves, the adversarial loss would tend to push the temporally evenly distributed sampled sequences towards those more realistic-looking.

For the Gradient Descent-based updates, a discretization of the point process is carried out, so as to approximate the predictions through a recursive computation of the integral of the intensity function (the *compensator* portion of the loglikelihood function) in a closed-form expression. The parameters of the complementary training are actually initialized by a purely MLE procedure, which was found to be more insensitive to initial points. The $MLE + GAN$ training updates then follows. The full procedure can be summarized by the following steps:

1. Subdivide the M original sequences on $[0, T]$ into training (on $[0, T^{tn}]$), validation (on $[T^{tn}, T^{vd}]$) and test (on $[T^{vd}, T]$) portions, using previously defined parameters T^{tn} and T^{vd};
2. Initialize the parameters of the model through the 'purely' MLE procedure;
3. Sample M sequences from the model over the interval $[0, T^{tn}]$;

4. By choosing a specific parameter shape for the excitation function and binning both the original and simulated sequences over equally-spaced intervals, define the closed-form expression for the MSE (discriminative) loss $\mathcal{L}_{MSE}$ over all the sequences and dimensions of the process;
5. Define the GAN (adversarial) loss of the model over the sequences as:

$$\mathcal{L}_{GAN} =$$

$$\begin{cases} E_{\mathcal{S}^{tn}\ \mathbb{P}(\mathcal{S}^{tn})}\left[F_W(Y_{\theta_S}(\mathcal{S}^{tn}))\right] - E_{\mathcal{S}^{tn}\ \mathbb{P}(\mathcal{S}^{tn})}\left[F_W(Y_{\theta_S}(\mathcal{S}^{tn}))\right], \\ \text{for the critic network } F_W \\ -E_{\mathcal{S}^{tn}\ \mathbb{P}(\mathcal{S}^{tn})}\left[F_W(Y_{\theta_S}(\mathcal{S}^{tn}))\right], \\ \text{for the training model (generator),} \end{cases} \tag{3.26}$$

where $\mathbb{P}(\mathcal{S}^{tn})$ corresponds to the underlying probability distribution we assumed to have generated the original sequences $\mathcal{S}^{tn}$, $F_W\{\cdot\}$ refers to a neural network which can compute the so-called Wasserstein distance, a metric for difference among distributions which will be further explained in Section 5, and $Y_{\theta_S}(\cdot)$ corresponds to the parametric model for sequence generation.
6. Compute the joint loss for MLE and GAN portions as:

$$\boldsymbol{L}_{MLE+GAN} = \gamma_{GAN}\boldsymbol{L}_{MLE} + (1-\gamma_{GAN})\boldsymbol{L}_{GAN},\ \text{for } \gamma_{GAN} \in [0,1] \tag{3.27}$$

7. Compute gradients of $\boldsymbol{L}_{MLE+GAN}$ over each parameter of model Y_{θ_S};
8. Update parameter estimates of training model with $\eta\nabla_{\theta_S}(\boldsymbol{L}_{MLE+GAN})$, for some learning rate η;
9. Repeat steps 3 to 8 until convergence.

The method is an interesting combination of the enriched dynamical modeling from the adversarial training strategy with the robustness over small training sets of the parametric-based MLE estimation for HPs.

In this section, we provided a comprehensive analysis of the progresses in HP modeling and inference for excitation functions assumed to be of a simple parametric shape, along with their compositions and variants. In the next section, we will discuss about advances in nonparametric HP excitation function strategies.

4. Nonparametric HPs. Nonparametric HPs consider that some rigid and simple parametric assumption for the triggering kernel may not be enough to capture all the subtleties of the excitation effects that could not be retrieved from the data. They may be broadly divided between two main approaches:

1. Frequentist
2. Bayesian

We will briefly discuss their variants in this section.

4.1. Frequentist Nonparametric HPs. The frequentist approach to HP modeling and inference consists in assuming that the excitation function (or matrix) can be defined as a binned grid (or a set of grids), in which the values of the functions would be taken as piecewise constant inside each bin, and the width of the bin will be (hopefully) expressive enough to model the local variations of the self-excitation effect.

They were first developed in the works of [43], [3] and [6]. In the case of [43], the final values of the bins were found by solving a discretized Ordinary Differential

Equation, implied by the branching structure of the discretized triggering kernel and background rate over the data, through iterative methods. The approach in [3] and [6], on the other side, recovers the piecewise constant model by exploiting relations, in the frequency domain, between the triggering kernel, the background rate and the second-order statistics of the model, also obtained in a discretized way over the data.

The increased expressiveness of this type of excitation model incurs in two main drawbacks:

- The bin division grid concept is close to that of a histogram over the distance among events, what usually requires much larger datasets for leading to accurate predictions, as opposed to parametric models, which would behave better on shorter and fewer sequences but most likely underfit on large sequence sets;
- The time of the inference procedure may also be much larger, since it involves sequential binning computation procedures which can not take advantage of the markov property of parametric functions such as the exponentials.

Two improvements, to be discussed in the present subsection, deal exactly with these drawbacks through:

- Acceleration of the computations over each sequence and/ or over each bin of the excitation matrix/function;
- Reduction of the times of binning computational procedures through a so-called online update of the bin values.

4.2. Acceleration of Impact Matrix Estimation through Matching of Cumulants. One Acceleration strategy is developed in [1], which replaces the task of estimating the excitation functions directly through estimating their cumulative values, i.e., their integrated values from zero up to infinity, what would be enough to quantify the causal relationships among each node. That is, instead of estimating $\phi_{ij}(t)$, for each node, the method would estimate a matrix $||\boldsymbol{\Phi}(t)|| = \{||\phi_{ij}(t)||\}$, in which:

$$||\phi_{ij}(t)|| = \int_0^\infty \phi_{ij}(t)dt, \, \forall (i,j) \in D \times D \tag{4.1}$$

The method, entitled Nonparametric Hawkes Process Cumulant (NPHC), then proceeds to compute, from the sequences, moment estimates $\hat{\boldsymbol{R}}$ up to the third-order. It then finds some estimate $\left|\left|\hat{\boldsymbol{\Phi}}(t)\right|\right|$ of this cumulant matrix which minimizes the L^2 squared error between these estimated moments and the actual moments $\boldsymbol{R}(||\boldsymbol{\Phi}(t)||)$, which are uniquely determined from $||\boldsymbol{\Phi}(t)||$:

$$\left|\left|\hat{\boldsymbol{\Phi}}(t)\right|\right| = \underset{||\boldsymbol{\Phi}(t)||}{\operatorname{argmin}} ||\boldsymbol{R}(||\boldsymbol{\Phi}(t)||) - \hat{\boldsymbol{R}}||^2 \tag{4.2}$$

This L^2 minimization comes from the fact that, by defining:

$$\boldsymbol{V} = (\mathbb{I}^D - \left|\left|\hat{\boldsymbol{\Phi}}(t)\right|\right|)^{-1}, \tag{4.3}$$

one may express the first-, second- and third-order moments of the process as:

$$\Lambda^i = \sum_{m=1}^{D} V^{im}\mu^m \tag{4.4}$$

$$\nu^{ij} = \sum_{m=1}^{D} \Lambda^m V^{im} V^{jm} \tag{4.5}$$

$$\begin{aligned} K^{ijk} = \sum_{m=1}^{D} \big(& V^{im} V^{jm} \nu^{km} + V^{im} \nu^{jm} V^{km} \\ & + \nu^{im} V^{jm} V^{km} - 2\Lambda^m V^{im} V^{jm} V^{km} \big), \end{aligned} \tag{4.6}$$

and thus we may find an estimator $\hat{\boldsymbol{V}} = \operatorname{argmin}_{\boldsymbol{V}} \boldsymbol{L}_{NPHC}(\boldsymbol{V})$, with $\boldsymbol{L}_{NPHC}(\boldsymbol{V})$ defined as:

$$\boldsymbol{L}_{NPHC}(\boldsymbol{V}) = (1 - \gamma_{NPHC}) ||\boldsymbol{K^c}(\boldsymbol{V}) - \hat{\boldsymbol{K}}^{\boldsymbol{c}}||_2^2 + \gamma_{NPHC} ||\boldsymbol{\nu}(\boldsymbol{V}) - \hat{\boldsymbol{\nu}}||_2^2, \tag{4.7}$$

where γ_{NPHC} is a weighting parameter, $|| \cdot ||_2^2$ is the Frobenius norm and $\boldsymbol{K^c} = \{K^{iij}\}_{1 \leq i,j \leq D}$ is a two-dimensional compression of the tensor $\boldsymbol{K}$. From this expression, by setting

$$\gamma_{NPHC} = \frac{||\hat{\boldsymbol{K}}^{\boldsymbol{c}}||_2^2}{||\hat{\boldsymbol{K}}^{\boldsymbol{c}}||_2^2 + ||\hat{\boldsymbol{\nu}}||_2^2}, \tag{4.8}$$

one may arrive to

$$\left|\left| \hat{\boldsymbol{\Phi}}(t) \right|\right| = \mathbb{I}^D - \hat{\boldsymbol{V}}^{-1} \tag{4.9}$$

The estimates of moments in the algorithm are actually computed through truncated and discretized (binned) countings along a single realization of the process and, since the real-data estimates are usually not symmetric, the estimates are averaged along positive and negative axis.

Also, for $D = 1$, the estimate $\left|\left| \hat{\boldsymbol{\Phi}}(t) \right|\right|$ can be estimated solely from the second-order statistics. For higher-dimensional processes, it is the skewness of the third-order moment which uniquely fixes $\left|\left| \hat{\boldsymbol{\Phi}}(t) \right|\right|$.

4.3. Online Learning. Another improvement of the method consists in updating the parameters of the discretized estimate of the excitation function through a single pass over the event sequence, i.e., an online learning procedure [91].

In the case of the referred algorithm, the triggering function is assumed to:

1. be positive,

$$\phi(t) \geq 0, \forall t \in \mathbb{R} \tag{4.10}$$

2. have a decreasing tail, i.e.,

$$\sum_{k=m}^{\infty} (t_k - t_{k-1}) sup_{x \in (t_{k-1}, t_k]} |f(y)| \leq \zeta_f(t_{i-1}), \forall i > 0, \tag{4.11}$$

   for some bounded and continuous $\zeta_f : \mathbb{R}^+ \mapsto \mathbb{R}^+$ s.t. $\lim_{t \to \infty} \zeta_f(t) = 0$
3. belong to a Reproducing Kernel Hilbert Space, which here is used as a tool for embedding similarity among high-dimensional and complex distributions onto lower-dimensional ones.

The method proceeds by taking the usual expression for the loglikelihood function

$$llh_{\tilde{T}}(\boldsymbol{\lambda}) = -\sum_{d=1}^{D}\left(\int_0^{\tilde{T}} \lambda_d(s)ds - y_{d,k}\log\lambda_d(t_k)\right), \tag{4.12}$$

and optimizing, instead, over a discretized version of it

$$llh_{\tilde{T}}(\boldsymbol{\lambda}) = \sum_{d=1}^{D}\sum_{k=1}^{M(t)}\left(\int_{\chi_{k-1}}^{\chi_k} \lambda_d(s)ds - y_{d,k}\log\lambda_i(t_k)\right) \tag{4.13}$$

$$= \sum_{d=1}^{D}\Delta L_{d,t}(\lambda_d), \tag{4.14}$$

with $(t_1, ..., t_{n(t)})$ denoting the event arrival times over an interval $[0, \tilde{T}]$, and with a partitioning $\{0, \chi_1, ..., \chi_{M(t)}\}$ of this interval $[0, \tilde{T}]$ such that

$$\chi_{k+1} = \min_{t_i \geq \chi_k}\{\iota * \lfloor \chi_k/\iota \rfloor + \iota, t_i\}, \tag{4.15}$$

for some small $\iota > 0$. The discretized version can then be expressed as

$$llh^{(\iota)}_{\tilde{T}}(\boldsymbol{\lambda}) = \sum_{d=1}^{D}\sum_{k=1}^{M(t)}\left(\int_{\chi_{k-1}}^{\chi_k} (\chi_k - \chi_{k-1})\lambda_d(\chi_k)\right. \tag{4.16}$$

$$\left. - y_{d,k}\log\lambda_i(\chi_k)\right) = \sum_{d=1}^{D}\Delta L^{(\iota)}_{d,\tilde{T}}(\lambda_d) \tag{4.17}$$

The optimization procedure is done at each slot of the $M(t)$ partition, taking into account:

- A truncation over the intensity function effect, i.e.,

$$\phi(t) = 0, \forall t > t_{max}, \tag{4.18}$$

  so as to simplify the optimization of the integral portion of the loss. The error over this approximation is shown to be bounded by the decreasing tail assumption.
- A Tikhonov regularization over the coefficients μ_d and $\phi_{d,d}$, which is simply the addition of weighted $||\mu_d||^2$ and $||\phi_{d,d}||^2$ terms to the loss function, so as to keep their resulting values small;
- A projection step for the triggering function optimization part, so as to keep them all positive.

The recent improvements over frequentist nonparametric HP estimation were shown to focus on two main strategies:

- Speeding up inference through replacement of the excitation matrix as objective by the matrix of cumulants, which were shown to be enough to capture the mutual influence among each pair of nodes;
- And an online learning procedure, which used some assumptions over the kernels (positive, decreasing tail RKHS) so as to recover estimates of the HP parameters over a single pass on some partitioning of the event arrival timeline.

Method	Total Complexity
ODE HP [101]	$\mathcal{O}(Iter * \tilde{D}_\phi(n_{max}^3 D^2 + M * (n_{max} D + n_{max}^2)))$
Granger Causality HP[86]	$\mathcal{O}(Iter * \tilde{D}_\phi n_{max}^3 D^2)$
Wiener-Hopf Eq. HP [6]	$\mathcal{O}(n_{max} D^2 M + D^4 M^3)$
NPHC [1]	$\mathcal{O}(n_{max} D^2 + Iter * D^3)$
Online Learning HP [91]	$\mathcal{O}(Iter * D^2)$

Table 1: Comparison of Computational Complexity of parametric and nonparametric HP estimation methods, extracted from [1]. *Iter* is the number of iterations of the optimization procedure, $\tilde{D}_\phi$ is the number of composing basis kernels of $\phi(t)$, D is the dimensionality of the MHP, n_{max} is the maximum number of events per sequence, and M is the number of components of the discretization applied to $\phi(t)$. Complexities obtained from [1] and [91].

	MHP Estimation Method			
Performance Metric	ODE HP [101]	Granger Causality HP [86]	ADM4 [100]	NPHC [1]
Relative Error	0.162	0.19	0.092	0.071
Estimation Time (s)	2944	2780	2217	38

Table 2: Performance comparison of several Multivariate HP Estimation methods in the MemeTracker [42] dataset, extracted from [1]. The relative error between a ground truth impact matrix $\boldsymbol{\alpha} = \{\alpha_{ij}\}$ and its estimate $\hat{\alpha} = \{\hat{\alpha}_{ij}\}$ is simply $\sum_{i,j} |\alpha_{ij} - \hat{\alpha}_{ij}|/|\alpha_{ij}| \mathbb{1}_{\{\alpha_{ij} \neq 0\}} + |\hat{\alpha}_{ij}| \mathbb{1}_{\{\alpha_{ij}=0\}}$.

A comparison among the complexity of several parametric and frequentist nonparametric HP estimation methods is shown in Table 1, and performance metrics are shown in Table 2. In general, it is possible to see a focus of more recent methods, such as NPHC and the Online Learning approach, in reducing the complexity per iteration of the resulting estimation procedure through approximation assumptions on the underlying model.

4.4. Bayesian Nonparametric HPs. Another nonparametric treatment of HPs revolves around the assumption of the triggering kernel and the background rates to be modeled by distributions (or mixtures of distributions) from the so-called "Exponential Family", which, through their conjugacy relationships, allow for closed-form computations of the sequential updates in the model. These were mainly proposed in the works of [19], [17] and [97].

In [19], HPs have also been used for modeling clustering of documents streams, capturing the dynamics of arrival time patterns, for being used together with textual content-based clustering.

The logic is that news and other media-related information sources revolving around a given occurrence- such as a natural catastrophe, a political action, or a celebrity scandal- are related not only regarding its word content, but also their time occurrences, as journalists tend to release more and more contents about a topic of high public interest, but tend to slow down the pace of publication as this interest gradually vanishes or shifts toward other subjects.

The main idea is to unite both Bayesian Nonparametric Inference, which is a

scalable clustering method which allows for new clusters to be added as the number of samples grow, with Hawkes Processes. The corresponding Bayesian Nonparametric model, the Dirichlet Process, would capture the diversity of event types, while the HPs would capture the temporal dynamics of the event streams.

A Dirichlet Process $DP(\alpha, G_0)$ can be roughly described as a probability distribution over probability distributions. It is defined by a concentration parameter α, proportional to the level of discretization ('number of bins') of the underlying sampled distribution, and a base distribution G_0, which is the distribution to be discretized. As an example, for α equal to 0, the distribution is completely concentrated at a single value, while, in the limit that α goes to infinity, the sampled distribution becomes continuous.

The corresponding hybrid model, the Dirichlet-Hawkes Process (DHP) is defined by:

- μ, an intensity parameter;
- $\mathbb{P}_0^{DHP}(\theta_{DHP})$, a base distribution over a given parameter space $\theta_{DHP} \in \Theta_{DHP}$;
- A collection of excitation functions $\phi_{\theta_{DHP}}(t, t')$.

After an initial time event t_1 and an excitation function parameter θ_{DHP}^1 are sampled from these base parameters μ and $\mathbb{P}_0^{DHP}(\theta_{DHP})$, respectively, the DHP is then allowed to alternate between:

1. Sampling new arrival events t_i from the current value of θ_{DHP} for the excitation function, with probability

$$\frac{\mu}{\mu + \sum_{i=1}^{n-1} \phi_{\theta_{DHP}^i}(t_n, t_i)} \tag{4.19}$$

2. Or sampling a new value for θ_{DHP} with probability

$$\frac{\sum_{i=1}^{n-1} \phi_{\theta_{DHP}^i}(t_n, t_i)}{\mu + \sum_{i=1}^{n-1} \phi_{\theta_{DHP}^i}(t_n, t_i)} \tag{4.20}$$

In this way, you are dealing with a superposition of HPs, in which the arrival events tend towards processes with higher intensities, i.e., the preferential attachment, but which also allows for diversity, since there is always a nonzero probability of sampling a new HP from the baseline intensity μ.

By defining the excitation functions ϕ_θ as a summation of parametric kernels

$$\phi_{\theta_{DHP}}(t_i, t_j) = \sum_{l=1}^{K} \alpha_{\theta_{DHP}}^l \kappa_{DHP}^l (t_i - t_j) \tag{4.21}$$

the model can be made even more general.

The approach in [17], besides the random histogram assumption for the triggering kernel, similar to the frequentist case, also considers the case of it being defined by a mixture of Beta distributions, and have the model being updated through a sampling procedure (Markov chain Monte Carlo).

The work in [97] proposes a Gamma distribution over the possible values of μ and the triggering kernel to be modeled as:

$$\phi(\cdot) = \frac{\mathcal{GP}(\cdot)^2}{2}, \tag{4.22}$$

where $\mathcal{GP}(\cdot)$ is a Gaussian Process [62].

These assumptions allow for closed-form updates over the posterior distributions over the background rate and the triggering kernel, which are claimed to be more scalable and efficient than the plain binning of the events.

In the next section, we will explore the neural architectures, which were introduced for modeling and generation of HPs through a more flexible representation of the effect of past events in the intensity function.

5. Neural Network-based HPs. In this section, we discuss the neural network-based formulations of HP modeling. The main idea is to capture the influence of past events on the intensity function in a nonlinear, and thus hopefully more flexible, way.

This modeling approach makes use of recurrent models, which, in their simplest formulation, encode sequences of states

$$(z_0^s, z_1^s, ..., z_N^s) \tag{5.1}$$

and outputs

$$(z_0^o, z_1^o, ..., z_N^o) \tag{5.2}$$

in a way that each state z_{i+1}^s can be obtained by a composition of the immediately preceding state z_i^s and a so-called hidden state h_i which captures the effect of the other past states:

$$h_i = \sigma_h(W_s z_i^s + W_h h_{i-1} + b_h) \tag{5.3}$$

$$z_i^o = \sigma_o(W_o z_i^o + b_o), \tag{5.4}$$

in which W_o, W_s, W_h, b_h and b_o are parameters to be fitted by the optimization procedure, while σ_h and σ_o are nonlinearities, such as a sigmoid or a hyperbolic tangent function.

In the case of HPs, the state to be modeled would be the intensity function along a sequence of time event arrivals, and an additional assumption would be that its intensity value decays exponentially between consecutive events.

In the case of most NN-based models, the inference also counts with a mark distribution for the case of marked HPs, in which a multinomial, or some other multi-class distribution, is fitted together with the recurrent intensity model.

Arguably the first from such type of models, the Recurrent Marked Temporal Point Process [18] jointly models marked event data by using a RNN with exponentiated output for modeling the intensity.

For sequences of the type $\{t_i, y_i\}_{i=1}^N$, in which t_i corresponds to the time of the i-th event arrival, while z_i^o refers to the type of event or mark, we have a hidden cell h_i described by:

$$\boldsymbol{h_i} = \max\left\{W_o z_i^o + W^t t_i + W_h h_{i-1} + b_h, 0\right\}, \tag{5.5}$$

and a Conditional Intensity Function defined as a function of this hidden state

$$\lambda(t) = exp\left(\boldsymbol{v^t h_i} + w_t(t - t_i) + b^t\right), \tag{5.6}$$

while a K-sized mark set can have its probability modeled by a Softmax distribution:

$$P(z_{i+1}^o = k|\boldsymbol{h_i}) = Softmax(k, \boldsymbol{V_k^{zo} h_i} + b_k^{zo}) = \frac{exp\left(\boldsymbol{V_k^{zo} h_i} + b_k^{zo}\right)}{\sum_{k=1}^K exp\left(\boldsymbol{V_k^{zo} h_i} + b_k^{zo}\right)} \tag{5.7}$$

As the likelihood of the whole sequence can be defined as a product of conditional density functions for each event:

$$llh\left(\{t_i, z_i^o\}_{i=1}^N\right) = \prod_{i=1}^{N} f(t_i, z_i^o), \tag{5.8}$$

with

$$f_\lambda(t) = \lambda(t) exp\left(-\int_{t_n}^{t} \lambda(t)dt\right), \tag{5.9}$$

where t_n is the latest event ocurred before time t. From this $f(t)$, we may estimate the time of the next event as:

$$t_{i+1} = \int_{t_i}^{\infty} t f_\lambda(t)dt \tag{5.10}$$

This allows us to optimize the parameters of the RNN model over the loss equal to this likelihood function, composed by these conditional density functions.

Given a set of M training sequences $\mathcal{S}^j = \{t_i^j, y_i^j\}_{i=1}^{N^j}$, we want to optimize the weight parameters over a loss function defined as:

$$llh(\{\mathcal{S}^j\}_{j=1}^M) = \sum_{j}^{M} \sum_{i}^{N_j} \log\left(P(z_{i+1}^{o,j}|\boldsymbol{h_i}) + \log f_\lambda(t_{i+1}^j - t_i^j|\boldsymbol{h_i})\right) \tag{5.11}$$

This optimization procedure is usually done through the Backpropagation Through Time (BPTT) algorithm, which proceeds by 'unrolling' the RNN cells by a fixed number of steps, then calculating the cumulative loss along all these steps, together with the gradients over each of W's, w's, v's and b's, then updating these parameters with a pre-defined learning rate until convergence.

One improvement for the RNN-based modeling approach is described in [83], referred to as "Time Series Event Sequence" (TSES), which consists of treating the mark sequences as derived from another RNN model, instead of the multinomial distribution. This RNN for the marks is then jointly trained with the RNN for event arrival times.

Another improvement over this NN-based modeling approach is the Neural Hawkes Process [52], which uses a variant of the basic RNN, entitled Long Short-Term Memory (LSTM) [30], applied to the intensity function modeling.

The Neural Hawkes Process models the intensity function of a multi-type event sequence by associating each k-th event type with a corresponding intensity function $\lambda_k(t)$, s.t.:

$$\lambda_k(t) = f_k(w_k^T \boldsymbol{z^h}(t)), \tag{5.12}$$

with

$$\boldsymbol{z^h}(t) = \boldsymbol{o_i} \odot (2g(2\boldsymbol{c}(t) - 1)), \tag{5.13}$$

The variables $\boldsymbol{o_i}$ and $\boldsymbol{c}(t)$ are defined through the following update rules:

$$\boldsymbol{c_{i+1}} = f_{i+1} \odot \boldsymbol{c}(t_i) + \boldsymbol{i}_{i+1} \odot \boldsymbol{z}_{i+1}, \tag{5.14}$$

The variables $\boldsymbol{i}_{i+1}$,$\boldsymbol{f}_{i+1}$, $\boldsymbol{z}_{i+1}$ and $\boldsymbol{o}_{i+1}$ are defined similarly to the gated variables of a standard LSTM cell. The reader is invited to read the original paper for their full definition. And the value of the $\boldsymbol{c}(t)$ is assumed to decay exponentially among consecutive events as:

$$\boldsymbol{c}(t) = \overline{\boldsymbol{c}}_{\boldsymbol{i+1}} + (\boldsymbol{c}_{i+1} - \overline{\boldsymbol{c}}_{\boldsymbol{i+1}})exp(-\boldsymbol{\delta}_{i+1}(t - t_i)), \tag{5.15}$$

for

$$\overline{\boldsymbol{c}}_{\boldsymbol{i+1}} = \overline{f}_{i+1} \odot \overline{\boldsymbol{c}}(t_i) + \overline{\boldsymbol{i}}_{i+1} \odot \boldsymbol{z}_{i+1} \tag{5.16}$$

$$\boldsymbol{\delta}_{i+1} = g(\boldsymbol{W}_{\boldsymbol{\delta}}\boldsymbol{k}_{\boldsymbol{i}} + \boldsymbol{U}_{\boldsymbol{\delta}}\boldsymbol{h}(t_i) + \boldsymbol{d}_{\boldsymbol{\delta}}). \tag{5.17}$$

The W's, U's and d's of the model are trained so as to maximize the loglikelihood over a set of sequences. Compared with the previous RNN-based model, in the Neural Hawkes Process:

1. The baseline intensity μ_k is not implicitly considered constant, but instead is allowed to vary;
2. The variations of the cell intensity are not necessarily monotonic, because the influences of each event type on the cell values may decay at a different rate;
3. The sigmoid functions along the composition equations allow for an enriched behaviour of the intensity values.

All this contributes to an increased expressiveness of the model. Besides, as in the regular LSTM models, the "forget" gates $\boldsymbol{f}_{i+1}$ are trained so as to control how much influence the past values of $\boldsymbol{c}(t)$ will have on its present value, thus allowing the model to possess a "long-term" memory.

Another variant of the RNN-based HPs, introduced in [84], models the baseline rate and the history influence as separate RNNs each. The baseline rate is taken as a time series, with its corresponding RNN updating its state at equally spaced intervals, such as five days. The event history influence RNN updates its state at each event arrival. This has been shown to increase the time and mark prediction performance, as demonstrated in Table 3.

Both the background rate time series $\{\mu(t)\}_{t=1}^{T}$ and the marked event sequence $\{m_i, t_i\}_{i=1}^{N}$ are modeled by LSTM cells:

$$(\boldsymbol{h}^{\mu}(t), \boldsymbol{z}_c^{\mu}(t)) = \text{LSTM}_{\mu}(\boldsymbol{\mu}(t), \boldsymbol{h}^{\mu}(t-1) + \boldsymbol{z}_c^{\mu}(t-1)), \tag{5.18}$$

$$(\boldsymbol{h}^{m}(i), \boldsymbol{z}_c^{m}(i)) = \text{LSTM}_{m}(\boldsymbol{m}_i, \boldsymbol{h}^{m}(i) + \boldsymbol{z}_c^{m}(i-1)), \tag{5.19}$$

These h and c states correspond to the hidden state and the long-term dependency terms, respectively, similarly to the Neural Hawkes Process. Both terms are concatenated in a single variable $\boldsymbol{z}_e(t)$, for jointly training both RNN models:

$$\boldsymbol{z}_e(t) = tanh(\boldsymbol{W}_f\,[\boldsymbol{h}^{\mu}(t), \boldsymbol{h}^{m}] + \boldsymbol{b}_f) \tag{5.20}$$

$$\boldsymbol{U}(t) = Softmax(\boldsymbol{W}_U\boldsymbol{z}_e(t) + \boldsymbol{b}_U), \tag{5.21}$$

$$\boldsymbol{u}(t) = Softmax(\boldsymbol{W}_u\,[\boldsymbol{z}_e(t), \boldsymbol{U}(t)] + \boldsymbol{b}_u) \tag{5.22}$$

$$z_s = \boldsymbol{W}_s \boldsymbol{z}_e(t) + b_s, \tag{5.23}$$

with U and u denoting the main event types and subtypes, respectively, and z_s denoting the composed timestamp of each event. The loss over which the model is trained is defined in a cross-entropy way:

$$\begin{aligned} &\sum_{j=1}^{N} \left(-\boldsymbol{W}_U(j) \log(U(t,j)) - w_u(j) \log(u(t,j)) \right. \\ &\left. - \log\left(f(z_s(t,j)|h(t-1,j))\right)\right), \end{aligned} \tag{5.24}$$

with

$$f\left(z_s(t,j)|h(t-1,j)\right) = \frac{1}{\sqrt{2\pi\sigma}} \mathrm{e}^{\left(-\frac{(z_s(t,j) - z_{\tilde{s}}(t,j))^2}{2\sigma^2}\right)}, \tag{5.25}$$

where $z_{\tilde{s}}(t,j)$ is the model predicted output for the corresponding event $z_s(t,j)$.

The model weights are then jointly trained, over the total loss function and under some correction for the frequency ratio of each event type, for both the background rate time series values and the event arrivals RNN, and are shown to outperform more 'rigid' models.

Now, regarding the generation of HP sequences, both RMTPP and Neural Hawkes Processes have in common the fact that they intend to model the intensity function of underlying process, so that new sequences may be sampled in a way to reproduce the behaviour of the original dataset. This intensity modeling, however, has three main drawbacks:

1. It may be unnecessary, since the sequences may be simply produced by unrolling cells of corresponding RNN models;
2. The sequences from these intensity-modeling approaches are sampled using a 'Thinning algorithm' [58], which may incur in slowed-down simulations, in the case of repeatedly rejected event intervals.
3. These methods are trained by maximizing the loglikelihood over the training sequences, which is asymptotically equivalent to minimizing the KL-Divergence over original and model distributions. This MLE approach is not robust in the case of multimodal distributions

The model in [81] proposes approximating a generative model for generating event sequences by using an alternative metric of difference among distributions, the Wasserstein (or Earth-Moving) distance, already discussed in Section 3.

In the model, entitled Wasserstein Generative Adversarial Temporal Point Process (WGANTPP), this Wasserstein loss is shown to be equal to:

$$L^{'} = \left[\frac{1}{m}\sum_{i=1}^{m} F_w(G_\Theta(\mathcal{S}_s^i)) - \frac{1}{m}\sum_{i=1}^{m} F_w(\mathcal{S}_r^i)\right] \tag{5.26}$$

$$+ \nu \sum_{i,j=1}^{m} \frac{|f_w(\mathcal{S}_r^i) - F_w(G_\Theta(\mathcal{S}_r^j))|}{|\mathcal{S}_r^i - G_\Theta(\mathcal{S}_s^j)|_*}, \tag{5.27}$$

where the second term, along with the constant ν correspond to the so-called Lipschitz constraints, related to the continuity of the models.

$\{\mathcal{S}_r^i\}_{i=1}^m$ are real data sequences, while $\{\mathcal{S}_s^i\}_{i=1}^n$ are sequences sampled from a Homogeneous Poisson Process with a rate λ_{HPP} which is simply the expected arrival

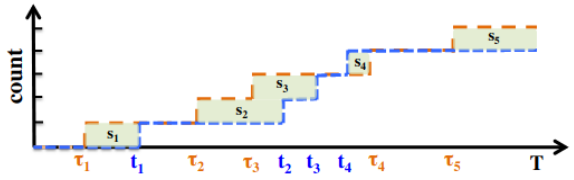

Fig. 5: Intuition behind the distance metric $|\cdot|_*$ among two given event sequences $\{t_i\}$ and $\{\tau_i\}$. Extracted from [81].

rate over all training sequences. The generator network G_Θ and discriminator F_w are defined as:

$$G_\Theta(\mathcal{S}_r) = \tilde{\mathcal{S}} = \{t_1, ..., t_n\} \tag{5.28}$$

with

$$t(i) = g_G^x(f_G^x(h(i))) \text{ and } h(i) = g_G^h(f_G^h(z_s, h(i-1))) \tag{5.29}$$

$$F_w(\tilde{\mathcal{S}}) = \sum_{i=1}^{n} a(i) \tag{5.30}$$

with

$$a(i) = g_D^a(f_D^a(h(i))) \text{ and } h(i) = g_D^h(f_D^h(t_i, h(i-1))) \tag{5.31}$$

with the g's defined as nonlinearities, and $\tilde{\mathcal{S}}$ as some example time event sequence. The f's are linear transformations, as in a standard RNN cell, with their corresponding weight matrices and bias vectors to be tuned by a Stochastic Gradient Descent procedure.

The distance metric $|\cdot|_*$ of two sequences $\{t_i\}$ and $\{\tau_i\}$, for the case of purely temporal point processes in $[0, T)$, can be shown to be equivalent to

$$\sum_{i=1}^{n} |t_i - \tau_i| + (m-n) \times T - \sum_{i=n+1}^{m} \tau_i, \tag{5.32}$$

which has an intuitive graphical interpretation, as shown in Figure 5.

As previously discussed, the generator is trained so as to "fool" the discriminator, while the discriminator is trained so as to distinguish generated sequences from those of real data. This adversarial training procedure is roughly equivalent to gradient updates with opposite signs over their respective parameters: positive sign for the discriminator and negative sign for the generator.

In the end of the training procedure, one hopefully gets a generator network capable of producing sequences virtually indistinguishable from the real data ones.

This method, however, consists of training a network for generating entire sequences, and so the generator model learned may not accurately generate conditional output sequences from input ones. Another model, described in [82], deals with this task by generating, from partially observed sequences, the future events of these same

sequences conditioned on their history, i.e., instead of aiming to capture the underlying distribution of a set of full sequences, the model performs a "sample agnostic" in-sample prediction.

Analogously to one of the parametric models described in Section 3, the learning procedure of this in-sample neural-network based prediction model takes advantages of both types of divergence measures: MLE loss (or KL Divergence) and Wasserstein distance.

The former aims for a rigid and unbiased parameter matching among two given probabilistic distributions, which is sensitive to noisy samples and outliers, while the latter has biased parameter updates, but is sensitive to underlying geometrical discrepancies among sample distributions. This combined loss is a way to balance both sets of priorities. In the case of long-term predictions, in which initial prediction errors propagate and magnify themselves throughout the whole stream, this joint loss was found to strengthen the effectiveness of the inference procedure.

The proposed model borrows on the seq2seq architecture ([73]) and aims to model endings of individual sequences conditioned on their partially observed history of initial events, and inserts an adversarial component in the training to increase the accuracy of long-term predictions. A network, designated as generator, encodes a compact representation of the initial partial observation of the sequence and outputs a decoded remaining of this same sequence. That is to say that, for a full sequence:

$$\{t_1, t_2, ..., t_{n+m}\}, \tag{5.33}$$

the seq2seq modeling approach learns a mapping

$$G_\Theta(\mathcal{S}^{1,n}) = \mathcal{S}^{m,n} \tag{5.34}$$

such that

$$\mathcal{S}^{1,n} = \{t_1, t_2, ..., t_n\} \text{ and } \mathcal{S}^{m,n} = \{t_{n+1}, t_{n+2}, ..., t_{n+m}\}. \tag{5.35}$$

This mapping is defined through a composition of RNN cells:

$$\boldsymbol{h}_i = \eta_g^h(f_A^h(t_i, \boldsymbol{h}_{i-1}) \text{ and } t_{i+1} = \eta_g^x f_g^x(\boldsymbol{h}_i) \tag{5.36}$$

with the η's defined as nonlinear activation functions, and the f's as linear transformations with trainable weight matrices and bias vectors.

The learning procedure consists of tuning the RNN cells' parameters so as to maximize the conditional probability

$$P(\mathcal{S}^{m,n}|\mathcal{S}^{1,n}) = \prod_{i=n}^{n+m-1} P(t_{i+1}|h_i, t_1, ..., t_i), \tag{5.37}$$

what is carried through by parameter gradient updates over the combined loss Wasserstein and MLE loss. The adversarial component of the training, the discriminator $F_w^{S2S}(\cdot)$, is modeled as a residual convolutional network (see [28]) with a 1-Lipschitz constraint, which is related to the magnitude of the gradients of the discriminative

model. The full optimization problem then becomes

$$\min_{\theta}\max_{w}\underbrace{\sum_{l=1}^{M}F_w^{S2S}(\{\mathcal{S}_l^{1,n},\mathcal{S}_l^{m,n}\})-\sum_{l=1}^{M}F_w^{S2S}(\{\mathcal{S}_l^{1,n},G_\Theta(\mathcal{S}_l^{1,n})\})}_{\text{Wasserstein loss}} \tag{5.38}$$

$$-\underbrace{\gamma_{LIP}|\frac{\partial F_{w'}^{S2S}(\hat{x})}{\partial\hat{x}}-1|}_{\text{1-Lipschitz constraint}}-\underbrace{\gamma_{MLE}log(\mathbb{P}_\theta(\mathcal{S}^{m,n}|\mathcal{S}^{1,n}))}_{\text{MLE loss}} \tag{5.39}$$

Further works on RNN-based modeling of HPs can also be found in [60] and [34].

5.1. Self-Attentive and Transformer Models. Another improvement for NN-based modeling, proposed in [95], involves a so-called self-attention strategy [76] to improve the accuracy of the resulting network. The i-th event tuple (t_i, m_i) is embedded as a variable x_i:

$$\boldsymbol{z_i} = \boldsymbol{tp_m} + \boldsymbol{pe_{(m_i,t_i)}}, \tag{5.40}$$

which simultaneously encodes information about the event mark through

$$\boldsymbol{tp_m} = \boldsymbol{z_e^m}\mathbb{W}_E, \tag{5.41}$$

with $\boldsymbol{z_e^m}$ as an one-hot encoding vector of the mark and $\mathbb{W}_E$ as an embedding matrix, and information about the time interval among consecutive events through a sinusoidal-based positional encoding vector $\boldsymbol{pe_{(m_i,t_i)}}$, with its k-th entry defined as:

$$pe^k_{(m_i,t_i)} = sin(\omega_k^i \times i + \omega_k^t \times t_i) \tag{5.42}$$

From this encoded variable $\boldsymbol{x_i}$, a hidden state $\boldsymbol{h_{u,i}}$ is then defined for each category u of the marks, which captures the influence of all previous events:

$$\boldsymbol{h_{u,i+1}} = \frac{\left(\sum_{j=1}^{i} f(\boldsymbol{z_{i+1}}, \boldsymbol{z_j})\boldsymbol{g(z_j)}\right)}{\sum_{j=1}^{i} f(\boldsymbol{z_{i+1}}, \boldsymbol{z_j})} \tag{5.43}$$

Through a series of nonlinear transformations, the intensity $\lambda_u(t)$ for the u-th mark is then computed. A concurrently developed approach in [102] uses multiple Attention layers to build a so-called "Transformer Hawkes Process", which also surpasses the performance of RNN-based approaches in a series of datasets, as shown in Table 3

5.2. Graph Convolutional Networks. A further improvement of neural HP models, described in [69], involves the graph properties of multivariate HPs, which may be embedded in a NN modeling framework through the recently proposed Graph Convolutional Networks (GCN)[38].

The method is composed of a GCN module, for capturing meaningful correlation patterns in a large sets of event sequence, followed by an usual RNN module, for modeling the temporal dynamics. In short, the time sequences are modeled as a HPs, and the adjacencies among different processes are encoded as a graph. The novel (GCN+RNN) model is meant to extract significant local patterns from the graph. The output of the initial GCN network module, which is simply a matrix of the form $\boldsymbol{\chi} = [\boldsymbol{\mu}, \mathbb{A}]$, which includes the baseline vector $\boldsymbol{\mu}$ and the adjacency matrix $\mathbb{A}$, are

	Loglikelihood per Event					Time Prediction RMSE			Mark Prediction Accuracy		
Method \ Dataset	RT	MT	FIN	MIMIC-II	SO	FIN	MIMIC-II	SO	FIN	MIMIC-II	SO
RMTPP [18]	-5.99	-6.04	-3.89	-1.35	-2.60	1.56	6.12	9.78	61.95	81.2	45.9
Neural HP [52]	-5.60	-6.23	-3.60	-1.38	-2.55	1.56	6.13	9.83	62.20	83.2	46.3
TSES [84]	-	-	-	-	-	1.50	4.70	8.00	62.17	83.0	46.2
Self-Attentive HP [95]	-4.56	-	-	-0.52	-1.86	-	3.89	5.57	-	-	-
Transformer HP [102]	-2.04	0.68	-1.11	0.82	0.04	0.93	0.82	4.99	62.64	85.3	47.0

Table 3: Performance comparison of NN-based HP models, regarding: a) Loglikelihood averaged per number of events; b) RMSE of predicted time interval; and c) Accuracy of mark prediction. The performances were measured over sequences from Retweet [98], MemeTracker [42], Financial [18], Medical Records [33], and Stack Overflow [42] datasets. The TSES method is a likelihood-free model, and so its entries are not evaluated for the **Loglikelihood per Event** section of the table. Values are obtained from [102]

fed into the RNN-based module, there taken as a LSTM. Then, the output of this RNN module are input to a further module, a fully connected layer, for calculating the changes $d\mathbb{X}$ to be applied to the current parameter matrix $\mathbb{X}$. Then, after each training step T, the predicted value of this parameter matrix becomes as

$$\mathbb{X}(T) = \mathbb{X}(T-1) + d\mathbb{X}(T-1)$$

.

The work in [90] proposes a model for check-in time prediction which is composed by a LSTM-based module, in which the feature vector is composed so as to capture each relevant aspect of the problem: the event time coordinate t_i, an additional field to indicate if the check-in occurred on a weekday or during the weekend, the euclidean distance between a given check-in location and the location (l) of the previous check-in, the location type of the check-in (e.g., Hotel, Restaurant, etc.), the number of users overlapping with a given location, and the check-ins by friends of the user. This aggregates social, geographical and temporal information in a single NN-based HP-like predictive model.

All these variants of neural-network point process models allow for more flexible (nonlinear) representation of the effect of past events in the future ones, besides putting at the inference procedure's disposal a myriad of Deep Learning tools and techniques which have enjoyed a surge of popularity over the recent years.

6. Further Approaches. In this section, we briefly review some recently proposed approaches which do not fit conveniently into any of the three previously discussed subgroups, but may be considered as bridgings between usual HP-related tasks and other mathematical subfields.

6.1. Sparse Gaussian Processes. By building over the modeling in [97], which considers the triggering function of the HP as a Gaussian Process, the work in [96] proposes an approach involving Sparse Gaussian Processes for optimizing over a dataset. In this, the optimization of the likelihood is being taken not over the samples, but over a set of much fewer so-called inducing points, which are also taken as latent variables, so as to result in a final model which is both expressive enough to capture the complexity of the dataset but also tractable enough to be useful and applicable to reasonably-sized datasets.

6.2. Stochastic Differential Equation. Another way of modeling HPs is through a Stochastic Differential Equation, as proposed in [40]. In them, the decay of the triggering kernel is taken as exponential, but its amplitude is defined as an stochastic process, there proposed as being either a Geometric Brownian Motion or an exponentiated version of the Langevin dynamics.

6.3. Graph Properties. Besides some works, previously discussed, which deal with the properties of the excitation matrices of the multivariate HPs as terms to be optimized jointly with other parameters, such as the Graph Convolutional approaches and the sparsity inducing penalization terms of some parametric and non-parametric approaches, there have been several other optimization strategies taking into account other properties of these matrices.

[46] explicitly inserts considerations of the excitation matrix as a distribution over some types of randomly generated matrices into the optimization of the HP likelihood. [48] introduces a penalization term involving the proximity of the excitation matrix to a so-called connection matrix, defined to capture the underlying connectivity among the nodes of the multivariate HP, to the parameter optimization strategy. [49] introduces a weighted sum of the Wasserstein Discrepancy and the so-called Gromov-Wasserstein Discrepancy as a penalizing factor on the usual MLE procedure of HP estimation, with the intention of inducing both absolute and relational aspects among the nodes of the HP. [2] introduces, besides the sparsity-inducing term, another one related to the resulting rank of the excitation matrix, which is designed to induce resulting matrices which are composed of both few nonzero entries and also few independent rows.

6.4. Epidemic HPs. The work in [66] blends the HP excitation effect with traditional epidemic models over populations. By considering a time event as an infection, it models the diffusion of a disease by introducing a HP intensity function which is modulated by the size of the available population, as below:

$$\lambda(t) = \left(1 - \frac{N_t}{\tilde{N}}\right) \{\mu + \sum_{t_i<t} \phi(t - t_i)\}, \tag{6.1}$$

where N_t denotes the counting process associated with the HP, while $\tilde{N}$ is the total finite population size.

6.5. Popularity Prediction. The work in [54] proposes the use of HP modeling blended with other Machine Learning techniques, such as Random Forests, to obtain an associated so-called "Popularity" measure, which is defined as the total number of events the underlying process is expected to generate as $t \to \infty$. This measure is treated as an outcome derived from features associated with some entity (e.g., social network user), s.a. number of friends, total number of posted statuses and the account creation time.

In the next section, we will discuss models in which one not only wants to capture the temporal dynamics, but also wishes to influence it towards a certain goal, implicitly defined through a so-called reward function.

7. Stochastic Control and Reinforcement Learning of HPs. In this section, we briefly review some control strategies regarding HPs. In some cases, one may wish not only to be able to model the traces of some event sequences, or to capture the underlying distribution of said sequences, but also try to influence their temporal dynamics towards more advantageous ones. These cases are considered in the works on Stochastic Control and Reinforcement Learning approaches for HPs.

This concept of advantageous is explicited through the definition of a so-called Reward Function, which is defined in terms of specific , and sometimes application-specific, properties of the sequences. Most works related to the subject deal with Social Network applications, and one example of Reward can be the total time the post of a user stays at the top of the feed of his/her followers. One type of reward which is not domain-specific is the dissimilarity among two sets of sequences, computed through mappings such as "kernel mean embeddings" [57]. In the case of Imitation Learning approaches, one still focus on solely modeling the HPs, without steering it towards desirable behaviours. In these, the reward function is simply defined over how well the samples of the chosen model to be adjusted approximates the samples of the original HP.

7.1. Stochastic Optimal Control. One example of this control approach is described in [94], in which the variable to be controlled are the times to post of a given user, implicitly defined as an intensity function, so as to maximize the reward function $\boldsymbol{r}(t)$, here computed as the total time that this user's posts stay at the top of the feed of his/her followers.

This "when-to-post" problem can be formulated as:

$$\min_{u(t_0,t_f)} \mathbb{E}_{(N_i,M_i)(t_0,t_f]} \left[\Omega(\boldsymbol{r}(t_f)) + \int_{t_0}^{t_f} \boldsymbol{L}(\boldsymbol{r}(\tau), u(\tau)) d\tau \right]$$

$$\text{subject to } u(t) \geq 0, \forall t \in (t_0, t_f], \tag{7.1}$$

where:

- i is the index of the broadcaster of the posts;
- $N_i(t)$ is the Counting Process of the i-th broadcaster, with $\boldsymbol{N}(t) = \{N_i(t)\}_{i=1}^n$ being an array of Counting Processes along all the n users of the network;
- $\mathbb{A} \in \{0,1\}^{n \times n}$ is the Adjacency Matrix of the network;
- $\boldsymbol{M}_i(t) = \mathbb{A}^T \boldsymbol{N}(t) - \mathbb{A}_i N_i(t)$, which means that $M_i(t)$ is the sum of the Counting Process of all users connected to user i excluding user i him/herself;
- t_0 and t_f are, respectively, the starting and ending times of the problem horizon taken in consideration;
- $u(t) = \mu_i(t)$, the controlled variable, is the baseline intensity of user i, to be steered towards the maximization of the reward function;
- $\Omega(\boldsymbol{r}(t_f))$ is an arbitrarily defined penalty function;
- $\boldsymbol{L}(\boldsymbol{r}(\tau), u(\tau))$ is a nondecreasing convex loss function defined w.r.t. the visibility of the broadcaster's posts in each of his/her followers' feeds.

The approach used in the problem is by defining an optimal cost-to-go $J(\boldsymbol{r}(t_f), \lambda(t), t)$:

$$J(\boldsymbol{r}(t_f), \lambda(t), t) = \min_{u(t,t_f]} \mathbb{E}_{(N,M)(t,t_f]} \left[\phi(\boldsymbol{r}(t_f)) + \int_{t}^{t_f} \boldsymbol{l}(\boldsymbol{r}(\tau), u(\tau)) d\tau \right], \tag{7.2}$$

and find the optimal solution through the Bellman's Principle of Optimality:

$$J(\boldsymbol{r}(t), \lambda(t), t) = \min_{u(t,t+dt]} \{\mathbb{E}[J(\boldsymbol{r}(t+dt), \lambda(t+dt), t+dt)] + \boldsymbol{l}(\boldsymbol{r}(t), u(t)) dt\}$$

For example, in the case of a broadcaster with one follower ($\boldsymbol{r}(t) = r(t)$), if the penalty and loss functions are defined as:

$$\phi(r(t_f)) = \frac{1}{2} r^2(t_f) \tag{7.3}$$

and

$$\boldsymbol{L}(r(t), u(t)) = \frac{1}{2}s(t)r^2(t) + \frac{1}{2}qu^2(t), \tag{7.4}$$

for some positive significance function $s(t)$ and some trade-off parameter q, which calibrates the importance of both visibility and number of posts, we set the derivative of $J(r(t), \lambda(t), t)$ over $u(t)$ to 0 and solve it to get to an analytical solution

$$u^*(t) = q^{-1}[J(r(t), \lambda(t), t) - J(0, \lambda(t), t)], \tag{7.5}$$

which is thus the optimal intensity a broadcaster must adopt to maximize visibility, constrained on the cost associated to the number of posts, along this one follower's feed. Further derivations are provided to the more natural and general case, in which the broadcaster may have multiple followers.

An earlier version of this type of Stochastic Optimal Control-based approach to influence activity in social networks can be found in [93]. In this, the goal is to maximize the total number of actions (or events) in the network. Analogously to the previously discussed algorithm, one may solve the Continuous Time version of the Bellman Equation through defining a optimal cost-to-go $J(\boldsymbol{\lambda}(t), t)$, which here depends only on the intensities of the nodes and the time.

The control input vector $\boldsymbol{u}(t)$ acts on the network by increasing the original vector of uncontrolled intensities

$$\boldsymbol{\lambda}(t) = \boldsymbol{\mu}_0 + \mathbb{A}\int_0^t \kappa(t-s)d\boldsymbol{N}(s) \tag{7.6}$$

with the equivalent rates of an underlying Counting Process vector $d\boldsymbol{M}(s)$, such that the new controlled intensity vector $\boldsymbol{\lambda}^*(t)$ is now described by

$$\boldsymbol{\lambda}^*(t) = \boldsymbol{\mu}_0 + \mathbb{A}\int_0^t \kappa(t-s)d\boldsymbol{N}(s) + \mathbb{A}\int_0^t \kappa(t-s)d\boldsymbol{M}(s), \tag{7.7}$$

where $\kappa(t) = e^{-\beta t}$ in the model.

Then, in the same way, by differentiating the equivalent $J(\boldsymbol{\lambda}(t), t)$ over the control input, setting the corresponding expression to 0, then defining

$$\boldsymbol{L}(\boldsymbol{\lambda}(t), \boldsymbol{u}(t)) = -\frac{1}{2}\boldsymbol{\lambda}^T(t)\boldsymbol{Q}\boldsymbol{\lambda}(t) + \frac{1}{2}\boldsymbol{u}^T(t)\boldsymbol{S}\boldsymbol{u}(t) \tag{7.8}$$

and

$$\Omega(\boldsymbol{\lambda}(t_f)) = -\frac{1}{2}\boldsymbol{\lambda}^T(t_f)\boldsymbol{F}\boldsymbol{\lambda}(t_f), \tag{7.9}$$

with previously defined symmetric weighting matrices $\boldsymbol{Q}$, $\boldsymbol{F}$ and $\boldsymbol{S}$, we arrive to a closed-form expression for the optimal control intensity value

$$\boldsymbol{u}^*(t) = -\boldsymbol{S}^{-1}\left[\mathbb{A}^T\boldsymbol{g}(t) + \mathbb{A}^T\boldsymbol{H}(t)\boldsymbol{\lambda}(t) + \frac{1}{2}diag(\mathbb{A}^T\boldsymbol{H}(t)\mathbb{A})\right], \tag{7.10}$$

where $\boldsymbol{H}(t)$ and $\boldsymbol{g}(t)$ can be computed by solving the differential equations

$$\dot{\boldsymbol{H}}(t) = (\beta\boldsymbol{I} - \mathbb{A})^T\boldsymbol{H}(t) + \boldsymbol{H}(\beta\boldsymbol{I} - \mathbb{A}) + \boldsymbol{H}(t)\mathbb{A}\boldsymbol{S}^{-1}\mathbb{A}^T\boldsymbol{H}(t)\boldsymbol{Q} \tag{7.11}$$

$$\begin{aligned}\dot{\boldsymbol{g}}(t) = &\left[\beta\boldsymbol{I} - \mathbb{A}^T + \boldsymbol{H}(t)\mathbb{A}\boldsymbol{S}^{-1}\mathbb{A}^T\right]\boldsymbol{g}(t) - \beta\boldsymbol{H}(t)\boldsymbol{\mu_0} \\ &+ \frac{1}{2}\left[\boldsymbol{H}(t)\mathbb{A}\boldsymbol{S^{-1}} - \boldsymbol{I}\right]diag(\mathbb{A}^T\boldsymbol{H}(t)\mathbb{A}),\end{aligned} \tag{7.12}$$

with final conditions $\boldsymbol{g}(t_f) = 0$ and $\boldsymbol{H}(t_f) = -\boldsymbol{F}$. The solution is constant between two consecutive events and gets recomputed at each event arrival.

7.2. Reinforcement Learning. The described SOC-based approaches have two main drawbacks:

- The functional forms of the intensities and mark distributions are constrained to be from a very restricted class, which does not include the state-of-the-art RNN-based HP models, such as those described in Section 5;
- The objective function being optimized is also restricted to very specific classes of functions, so as to keep the tractability of the problem.

For circumventing these drawbacks, there have been some proposed approaches which combine more flexible and expressive HP models with robust stochastic optimization procedures independent from the functional form of the objective function.

On of these methods, entitled "Deep Reinforcement Learning of Marked Temporal Point Processes" [75], works by, given a set of possible actions and corresponding feedbacks, which are both expressed as temporal point processes jointly modeled by a RNN-based intensity model $\lambda^*_\theta(t)$:

$$\lambda^*_\theta(t) = exp(b_\lambda + w_t(t - t_i) + \boldsymbol{V_\lambda h_i}), \tag{7.13}$$

with

$$\boldsymbol{h_i} = tanh(\boldsymbol{W_h h_{i-1}} + \boldsymbol{W_1}\mathcal{T}_i + \boldsymbol{W_2 y_i} + \boldsymbol{W_3 z_i W_4 b_i} + \boldsymbol{b_h}),$$

where

$$\mathcal{T}_i = f_T(t_i - t_{i-1}) \text{ and } \boldsymbol{b_i} = f_b(1 - b_i, b_i)$$

$$\boldsymbol{y_i} = f_y(y_i), \text{ if } b_i = 0 \text{ and } \boldsymbol{z_i} = f_z(z_i), \text{ if } b_i = 1.$$

The term b_i is an indicator function to whether the i-th event is an action or a feedback. By taking the weight matrices and bias vectors from all the linear transformations f a parameter vector θ, what the algorithm wishes to do is to update this vector with the gradients of each parameter over an expected Reward Function $J(\theta)$:

$$\theta_{l+1} = \theta_l + \eta_l \nabla_\theta J(\theta)|_{\theta=\theta_l} \tag{7.14}$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{\mathcal{U}_T \ p^*_{\mathcal{U},\theta}(\cdot), \mathcal{F}_T \ p^*_{\mathcal{F},\phi}(\cdot)}[R^*(T)\nabla_\theta log \mathbb{P}_\theta(\mathcal{U}_T)], \tag{7.15}$$

where

$$p^*_{\mathcal{U},\theta} = (\lambda^*_\theta, m^*_\theta) \tag{7.16}$$

is the joint conditional intensity and mark distribution for action events, and

$$p^*_{\mathcal{F},\phi} = (\lambda^*_\phi, m^*_\phi) \tag{7.17}$$

is the joint conditional intensity and mark distribution for feedback events. The Reward $R(T)$ is defined over some domain-specific metric, which may involve the responsiveness of followers in a social network setting, or the effectiveness of memorization of words in a foreign language, in a spaced-repetition learning setting.

7.3. Imitation Learning. Another way of using this reinforcement learning approach is through a technique entitled "Imitation Learning" [44]. The reasoning behind it is to treat the real-data sequences as generated by an expert and then, using RNN-based sequence generation, try to make these models approximate the real-data sequence as closely as possible.

Thus, the reward function to dictate the proportion in which the gradient of the parameters over each sequence are going to be considered is equal to how much this given sequence is likely to be drawn from the underlying distribution over the real data. This similarity is computed through a Reproducing Kernel Hilbert Space (RKHS).

The theory of RKHS is very extense, and it is not our goal to give a detailed account of it here. The key idea is that, to compute similarities among items of a given space, you compute inner products between them. Taking two items, x_1 and x_2, and computing a so-called Positive Definite Kernel (PDS) $K(x_1, x_2)$ is equivalent to computing a inner product among these two items in a high-dimensional, and potentially infinite-dimensional, vector space. The PDS used in the corresponding paper is the Gaussian kernel.

The reward function is then defined as:

$$\hat{r}^*(t) \propto \frac{1}{L}\sum_{l=1}^{L}\sum_{i=1}^{N_T^{(l)}} \mathbb{K}(s_i^{(l)}, l) - \frac{1}{M}\sum_{m=1}^{M}\sum_{i=1}^{N_T^{m}} \mathbb{K}(t_i^{(m)}, t), \tag{7.18}$$

where:

- L is the number of expert trajectories;
- M is the number of trajectories generated by the model;
- $\mathbb{K}(\cdot, \cdot)$ is the reproducing kernel operator;
- $s_i^{(l)}$ is the i-th time coordinate of the l-th expert trajectory;
- $t_i^{(m)}$ is the i-th time coordinate of the m-th model-generated trajectory.

The parameters of the model are then updated through a Gradient Descent-based approach towards convergence, in which the model is expected to generate sequences indistinguishable from the real-world process.

8. Real-world Data Limitations. One key aspect of HP modeling which inevitably encompasses all the previously mentioned approaches is that of their applicability to real-world datasets. These may present a series of sistematic issues on the training and testing sequences, which may completely hinder the generalization of the models. We discuss some key issues here, along with some recently proposed methods, designed to handle each of them.

8.1. Synchronization Noise. Temporal data, specially multivariate ones, may have their event streams being extracted from distributed sensor networks. A key challenge regarding them is that of synchronization noise, i.e., when each source is subject to an unknown and random time delay.

In these cases, an inference procedure which neglects these time delays may be ignoring critical causal effects of some events over others, thus resulting in a poorly generalizing model. The work in [74] deals specifically with this aspect of real-data HP modeling, and proposes, for the exponential triggering functions HPs, including the random time shift vector (one entry for each distinct event stream) as a parameter in the HP model, which results in an inference procedure of the following form:

$$\hat{\boldsymbol{z}}, \hat{\boldsymbol{\theta}} = \underset{\boldsymbol{z}\in\mathbb{R}, \theta\geq 0}{\operatorname{argmax}} \log \mathbb{P}(\tilde{\boldsymbol{t}}|\mathcal{N}, \boldsymbol{\theta}), \tag{8.1}$$

where $\mathcal{N}$ is the random noise vector, and $\boldsymbol{\theta}$ corresponds to the parameters of the original exponential HP model.

8.2. Sequences with few events. In many domains, the availability of data is scarce, and the event streams will be composed of too few events, which results in noisiness of the likelihood and, as a consequence, unreliability of the fitted HP model, which calls for strong regularization strategies over the objective functions to be optimized.

For this type of situation, an approach, presented in [68], deals with HPs with triggering functions defined as exponentials and also as a mixture of gaussian kernels, as in [86]. Then, the parameters left to search are the background rates vector $\boldsymbol{\mu}$ and the tensor of weightings $\boldsymbol{A}$ for the excitation functions. The optimization is done through an Variational Expectation-Maximization algorithm which takes the distributions over these parameters as gaussians, and optimize, through Monte Carlo sampling, over an Evidence Lower Bound (ELBO) of their corresponding loglikelihood over a set of sequences.

8.3. Sequences with Missing data. Another issue in HP modeling revolves around learning from incomplete sequences, i.e., streams in which one or more of the events are missing. For this type of problem, two rather distinct approaches were recently proposed:

1. The first one, presented in [72], is applied to exponential and power-law HPs, and consists of a Markov Chain Monte Carlo-based inference over a joint process implicitly defined by the product of the likelihoods of the observed events and of the so-called virtual event auxiliary variables, which are candidates for unobserved events. This virtual variable is weighted through a parameter κ, which is related to the percentage of missing events with respect to the total event count;
2. The second one, introduced in [53], proposes finding the missing events over the sequences through importance weighting of candidate filling event subsequences generated by a bidirectional LSTM model built on top of the Neural Hawkes Process [52].

9. Application Examples. In this section, we use our HP modeling background obtained so far and apply it in case studies for three different domains: Retweeting behaviour in Social Networks, Earthquake aftershocks, and COVID-19 contact tracing. We hope this will encourage the reader to consider HPs as a modeling choice for a broad scope of applications.

9.1. Retweet. In [20], HP are used jointly with Latent Dirichlet Allocation (LDA) [8] models for distinguishing among genuine and fake (i.e., artificially induced) retweeting of posts among Twitter users.

Given a set of 2508 users, with each j-th user corresponding to a sequence

$$\mathbf{RT}^j = \{(t_i^j, \mathcal{W}_i^j)\}_{i=0}^{N_j}, \tag{9.1}$$

, where t_i^j corresponds to the timestamp associated with the i-th retweet from the j-th user, and $\mathcal{W}_i^j$ as the text content of the corresponding retweet.

The 2508 corresponding sequences were manually separated between *genuine* and *fake* users and labelled as such, based on a set of criteria (e.g., the content in a most of said user's retweets contains spammy links and common spam keywords, multiple retweets from a given user contain promotional/irrelevant text, the user biographical

information is fabricated or contained promotional activity, or a large number of tweets or retweets were posted within a very short time window of just a few seconds).

The two resulting disjoint sets were used for training two LDA models, LDA_f and LDA_g, for modeling the topics of fake and genuine retweeters, respectively. Given a predefined number of possible topics, here set as 10, and a given text content $\mathcal{W}_i^j$, the LDA model outputs a 10-element vector, with the probabilities of the $\mathcal{W}_i^j$ corresponding to each of the 10 possible topics.

The 10-sized vector $\mathcal{V}_f$ from LDA_f is concatenated with the 10-sized vector $\mathcal{V}_g$ from LDA_g and, together with the baseline intensity μ and the decay β of an exponential HP with $\phi(t) = \mathrm{e}^{-\beta t}$, fitted over the t_i^j's of each j-th sequence, forms a feature vector

$$\{\mathcal{V}_f^j, \mathcal{V}_g^j, \mu^j, \beta^j\}, \tag{9.2}$$

which is then fed to a clustering algorithm, which aims to correctly classify each of the 2508 retweet sequences among fake and genuine ones. The intuition behind this hybrid HP-LDA model is to use temporal features, from the HP modeling, together with context (written) ones from the users to improve the resulting detection algorithm.

9.2. Earthquake aftershocks. It is well known from the study of earthquake-related time series that a strong first seismic shock gives way to a series of weaker aftershocks, which occur in a very restricted time window [59].

For modeling this self-exciting property of aftershocks' arrivals, [59] proposes a power-law self-triggering kernel

$$\phi_{PWL}(t, \theta_{PWL}) = \frac{K}{(t+c)^p}, \tag{9.3}$$

with $\theta_{PWL} = (K, c, p) \in \mathbb{R}_+^3$ as trainable parameters. This model, together with an additional baseline rate parameter μ, is fitted over the temporal sequence $\{t_1, t_2, ..., t_n\}$ of aftershock timestamps with a MLE optimization procedure, such as the one in Equation 2.17, in which, due to the simple parametric form of the equation for the intensity, the loglikelihood can be given in closed-form.

9.3. COVID-19. In [13], a spatiotemporal HP-inspired model is used to predict the daily rates of cases and deaths associated with the Sars-CoV-2 pandemic.

Given:

- A sequence $\mathcal{S} = \{t_1, t_2, ..., t_n\} \in \mathbb{Z}_+^n$ of timestamps (or dates);
- A baseline rate μ_c;
- A probability distribution for inter-infection time, assumed to be a Weibull distribution with shape α and scale β;
- A vector $\boldsymbol{\mathcal{M}}_c^t = \{\mathcal{M}_1^t, \mathcal{M}_2^t, ...\}$ of mobility indices, measured in percentual increase/decrease with respect to standardized values for each activity category (Recreation, Groceries, Parks, etc);
- A vector $\boldsymbol{\mathcal{D}}_c^t = \{\mathcal{D}_1^t, \mathcal{D}_2^t, ...\}$ of static demographic features, such as percentage of smokers, population density, and number of ICU beds;
- A parameter Δ for capturing a potential delay between a change on mobility indices and the time t_j of a primary infection being reported.

A model of the rates of cases (or deaths) λ_c is built as

$$\lambda_c = \mu_c + \sum_{t > t_j, t_j \in \mathcal{S}} \mathcal{R}_c^{t_j}(\boldsymbol{\mathcal{D}}_c, \theta_{\mathcal{D}}) \times \mathcal{R}_c^{t_j}(\mathbf{m}_c^{t_j - \Delta}, \theta_m) w(t - t_j), \tag{9.4}$$

where $\mathcal{R}_c^{t_j}$ refers to the Reproduction Number, which is the number of people a given infected individual is expected to transmit the disease to. This $\mathcal{R}_c^{t_j}$ is model through a Poisson regression

$$\mathbb{E}\left[\mathcal{R}_c^{t_j} | \mathbf{x}_c^{t_j-\Delta}, \theta\right] = \mathrm{e}^{\theta_{\mathcal{M}}^T \mathbf{x}_c^{t_j-\Delta}} \tag{9.5}$$

with $\mathbf{x}_c^{t_j-\Delta} = \left[\boldsymbol{\mathcal{D}}_c \ \boldsymbol{\mathcal{M}}_c^{t_j-\Delta}\right]$ combining both temporal and spatial covariates into a single vector.

An Expectation-Maximization strategy, similar to that of [43], is used for fitting the model parameters. This HP-based model is shown to outperform the Susceptible-Exposed-Infected- Removed (SEIR) model, most usually associated with disease spread forecasting.

10. Comparisons with other Temporal Point Process approaches. HPs, and the simpler PPs, have been the most prevalent choice for modeling time event sequences, but some different approaches have been proposed, which occasionally surpassed the performance of HPs in some situations, such as:

- Wold Processes: These are the equivalent of a HP in which only the effect of the most recent event is considered in the computation of the intensity function. This Markovian aspect, regardless of the choice of the excitation function, has been shown in [21] to surpass the performance of several HP models for estimation of networked processes.
- Intensity-Free Learning of Inter-Event Intervals: Another approach which has been recently introduced involves ignoring the intensity function completely, and focusing on modeling the probabilistic distribution of the time intervals among consecutive events. This distribution is modeled after Normalizing Flows [64], which can be summed up as families of distributions with incremental complexities. The approach was introduced in [70, 71], and was shown to surpass state-of-the-art neural-based HP models in some large-sized datasets.
- Continuous-Time Markov Chain: In this model, the marks correspond to states which have fixed rates (intensities) associated to them. The transition time are sampled from these constant intensities. It has been used as a comparison baseline for some HP models, such as [19].

11. Current Challenges for Further Research. Regarding the challenges currently tackled by HP researchers, we could mention:

- Enriching HP variants (parametric, nonparametric, neural), or blending them with other ML approaches, so as to make them suitable for specific situations. The works with Multi-Armed Bandits [14], randomized kernels [32], graph neural networks for temporal knowledge graphs [24] and composition of HP-like Point Processes with Warping functions defined over the time event sequences [85] can be considered in this category;
- Improving the speed of inference or sampling, so as to reduce the time spent in model estimation an aspect which may be critical for some real-world applications. The works of [31] in Bayesian mitigation of spatial coarsening, [99] in multi-resolution segmentation for nonstationary Hawkes process using cumulants, [45] on thinning of event sequences for accelerating inference steps, [50] on the use of Lambert-W functions of improving sequence sampling, [11] on perfect sampling are examples of such, and [61] on recursive computation of HP moments;

- How to properly evaluate and compare HP models among them: While there has been a lot of work regarding proposing new approaches, the comparison among existing models is often biased or incomplete. The works of [77] on how to quantify the uncertainty of the obtained models, [80] on measuring goodness-of-fit, [51] on robust identification of HPs with controlled terms, and [9] on the rigorous comparison of networked point process models are among this type of work;
- Theoretical guarantees, properties and formulations of specific HP approaches, such as the works of [12] on strong mixing, [23] on the consistency of some parametric models, [15] on elementary derivations of HP momenta, and [36, 37] on field master equation formulation for HPs.

12. Conclusions. Hawkes Processes are a valuable tool for modeling a myriad of natural and social phenomena. The present work aimed to give a broad view, to a newcomer to the field, of the Inference and Modeling techniques regarding the application of Hawkes Processes in a variety of domains. The parametric, nonparametric, Deep Learning and Reinforcement Learning approaches were broadly covered, as well as the current research challenges on the topic and the real-world limitations of each approach. Illustrative application examples in the modeling of Retweeting behaviour, Earthquake aftershock occurence and COVID-19 spreading were also briefly discussed, for motivating the applicability of Hawkes Processes in both natural and social phenomena.

13. Acknowledgements. Any opinions, findings, and conclusions expressed in this work are those of the author and do not necessarily reflect the views of Samsung R&D Institute Brazil.

REFERENCES

[1] M. Achab, E. Bacry, S. Gaïffas, I. Mastromatteo, and J. Muzy, *Uncovering causality from multivariate hawkes integrated cumulants*, Journal of Machine Learning Research, 18 (2017), pp. 192:1–192:28.

[2] E. Bacry, M. Bompaire, S. Gaïffas, and J.-F. Muzy, *Sparse and low-rank multivariate hawkes processes*, 2015, https://arxiv.org/abs/1501.00725.

[3] E. Bacry, S. Delattre, M. Hoffmann, and J. Muzy, *Scaling limits for hawkes processes and application to financial statistics*, 123 (2012).

[4] E. Bacry, S. Gaïffas, I. Mastromatteo, and J. Muzy, *Mean-field inference of hawkes point processes*, CoRR, abs/1511.01512 (2015), http://arxiv.org/abs/1511.01512.

[5] E. Bacry, I. Mastromatteo, and J.-F. Muzy, *Hawkes processes in finance*, arXiv, (2015), https://arxiv.org/abs/1502.04592.

[6] E. Bacry and J. Muzy, *First- and second-order statistics characterization of hawkes processes and non-parametric estimation*, IEEE Trans. Information Theory, 62 (2016), pp. 2184–2202.

[7] T. Björk, *An introduction to point processes from a martingale point of view*. 2011.

[8] D. M. Blei, A. Y. Ng, M. I. Jordan, and J. Lafferty, *Latent dirichlet allocation*, Journal of Machine Learning Research, 3 (2003), p. 2003.

[9] G. Borges, P. O. S. V. de Melo, F. Figueiredo, , and R. Assuncao, *Networked point process models under the lens of scrutiny*, in The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), 2020.

[10] J. Chen, A. G. Hawkes, and E. Scalas, *A fractional hawkes process*, 2020, https://arxiv.org/abs/2003.01027.

[11] X. Chen and X. Wang, *Perfect sampling of multivariate hawkes process*, 2020, https://arxiv.org/abs/2007.05940.

[12] F. Cheysson and G. Lang, *Strong mixing condition for hawkes processes and application to whittle estimation from count data*, 2020, https://arxiv.org/abs/2003.04314.

[13] W.-H. Chiang, X. Liu, and G. Mohler, *Hawkes process modeling of covid-19 with mobility leading indicators and spatial covariates*, medRxiv, (2020), https://doi.org/10.1101/2020.06.06.20124149.

[14] W.-H. Chiang and G. Mohler, *Hawkes process multi-armed bandits for disaster search and rescue*, 2020, https://arxiv.org/abs/2004.01580.

[15] L. Cui, A. Hawkes, and H. Yi, *An elementary derivation of moments of hawkes processes*, Advances in Applied Probability, 52 (2020), p. 102–137, https://doi.org/10.1017/apr.2019.53.

[16] D. Daley and D. Vere-Jones, *An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods*, Springer, 2003.

[17] S. Donnet, V. Rivoirard, and J. Rousseau, *Nonparametric bayesian estimation of multivariate hawkes processes*, 2018, https://arxiv.org/abs/1802.05975.

[18] 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 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, 2016.

[19] N. Du, M. Farajtabar, A. Ahmed, A. J. Smola, and L. Song, *Dirichlet-hawkes processes with applications to clustering continuous-time document streams*, in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015, 2015, pp. 219–228.

[20] H. S. Dutta, V. R. Dutta, A. Adhikary, and T. Chakraborty, *Hawkeseye: Detecting fake retweeters using hawkes process and topic modeling*, IEEE Transactions on Information Forensics and Security, 15 (2020), pp. 2667–2678, https://doi.org/10.1109/TIFS.2020.2970601.

[21] F. Figueiredo, G. R. Borges, P. O. S. V. de Melo, and R. Assunção, *Fast estimation of causal interactions using wold processes*, in Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada, 2018, pp. 2975–2986.

[22] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, *Generative adversarial nets*, in Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, 2014, pp. 2672–2680.

[23] X. Guo, A. Hu, R. Xu, and J. Zhang, *Consistency and computation of regularized mles for multivariate hawkes processes*, 2018, https://arxiv.org/abs/1810.02955.

[24] Z. Han, Y. Ma, Y. Wang, S. Günnemann, and V. Tresp, *Graph hawkes neural network for forecasting on temporal knowledge graphs*, 2020, https://arxiv.org/abs/2003.13432.

[25] A. G. Hawkes, *Point spectra of some mutually exciting point processes*, Journal of the Royal Statistical Society. Series B (Methodological), 33 (1971), pp. 438–443, http://www.jstor.org/stable/2984686.

[26] A. G. Hawkes, *Spectra of some self-exciting and mutually exciting point processes*, Biometrika, (1971), pp. 201–213.

[27] A. G. Hawkes, *Hawkes processes and their applications to finance: a review*, Quantitative Finance, 18 (2018), pp. 193–198.

[28] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, 2016, pp. 770–778.

[29] A. Helmstetter and D. Sornette, *Diffusion of epicenters of earthquake aftershocks, omori's law, and generalized continuous-time random walk models*, Phys. Rev. E, 66 (2002), p. 061104.

[30] S. Hochreiter and J. Schmidhuber, *Long short-term memory*, Neural Computation, 9 (1997), pp. 1735–1780.

[31] A. J. Holbrook, X. Ji, and M. A. Suchard, *Bayesian mitigation of spatial coarsening for a fairly flexible spatiotemporal hawkes model*, 2020, https://arxiv.org/abs/2010.02994.

[32] F. Ilhan and S. S. Kozat, *Modeling of spatio-temporal hawkes processes with randomized kernels*, IEEE Transactions on Signal Processing, 68 (2020), p. 4946–4958, https://doi.org/10.1109/tsp.2020.3019329, http://dx.doi.org/10.1109/TSP.2020.3019329.

[33] A. E. Johnson, T. J. Pollard, L. Shen, L. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, *Mimic-iii, a freely accessible critical care database*, Scientific data, 3 (2016), p. 160035.

[34] S. Joseph, L. D. Kashyap, and S. Jain, *Shallow neural hawkes: Non-parametric kernel estimation for hawkes processes*, 2020, https://arxiv.org/abs/2006.02460.

[35] A. T. Kalai and R. Sastry, *The isotron algorithm: High-dimensional isotonic regression*,

in COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009, 2009.

[36] K. Kanazawa and D. Sornette, *Field master equation theory of the self-excited hawkes process*, Physical Review Research, 2 (2020), https://doi.org/10.1103/physrevresearch.2.033442, http://dx.doi.org/10.1103/PhysRevResearch.2.033442.

[37] K. Kanazawa and D. Sornette, *Nonuniversal power law distribution of intensities of the self-excited hawkes process: A field-theoretical approach*, Phys. Rev. Lett., 125 (2020), p. 138301, https://doi.org/10.1103/PhysRevLett.125.138301, https://link.aps.org/doi/10.1103/PhysRevLett.125.138301.

[38] T. N. Kipf and M. Welling, *Semi-supervised classification with graph convolutional networks*, CoRR, abs/1609.02907 (2016), http://arxiv.org/abs/1609.02907, https://arxiv.org/abs/1609.02907.

[39] R. Kobayashi and R. Lambiotte, *Tideh: Time-dependent hawkes process for predicting retweet dynamics*, in Proceedings of the Tenth International Conference on Web and Social Media, Cologne, Germany, May 17-20, 2016., 2016.

[40] Y. Lee, K. W. Lim, and C. S. Ong, *Hawkes processes with stochastic excitations*, in Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016, 2016, pp. 79–88.

[41] R. Lemmonier, K. Scaman, and A. Kalogeratos, *Multivariate hawkes processes for large-scale inference*, in Proceedings of the Conference on Artificial Intelligence, 2017.

[42] J. Leskovec and A. Krevl, *SNAP Datasets: Stanford large network dataset collection*, 2014, http://snap.stanford.edu/data.

[43] E. Lewis and G. Mohler, *A nonparametric em algorithm for multiscale hawkes processes*, Journal of Nonparametric Statistics, 1 (2011), pp. 1–20.

[44] S. Li, S. Xiao, S. Zhu, N. Du, Y. Xie, and L. Song, *Learning temporal point processes via reinforcement learning*, in Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada., 2018, pp. 10804–10814.

[45] T. Li and Y. Ke, *Thinning for accelerating the learning of point processes*, in Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, eds., 2019, pp. 4093–4103.

[46] S. W. Linderman and R. P. Adams, *Discovering latent network structure in point process data*, in Proceedings of the International Conference on Machine Learning, 2014, pp. 1413–1421.

[47] T. Liniger, *Multivariate Hawkes Processes*, PhD thesis, ETH Zurich, 2009.

[48] Y. Liu, T. Yan, and H. Chen, *Exploiting graph regularized multi-dimensional hawkes processes for modeling events with spatio-temporal characteristics*, in Proceedings of the 27th International Joint Conference on Artificial Intelligence, AAAI Press, 2018, p. 2475–2482.

[49] D. Luo, H. Xu, and L. Carin, *Fused gromov-wasserstein alignment for hawkes processes*, CoRR, abs/1910.02096 (2019), http://arxiv.org/abs/1910.02096.

[50] M. Magris, *On the simulation of the hawkes process via lambert-w functions*, 2019, https://arxiv.org/abs/1907.09162.

[51] M. Mark and T. A. Weber, *Robust identification of controlled hawkes processes*, Phys. Rev. E, 101 (2020), p. 043305, https://doi.org/10.1103/PhysRevE.101.043305, https://link.aps.org/doi/10.1103/PhysRevE.101.043305.

[52] H. Mei and J. Eisner, *The neural hawkes process: A neurally self-modulating multivariate point process*, in Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, 2017.

[53] H. Mei, G. Qin, and J. Eisner, in Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, vol. 97 of Proceedings of Machine Learning Research, PMLR, 2019, pp. 4475–4485.

[54] S. Mishra, M. Rizoiu, and L. Xie, *Feature driven and point process approaches for popularity prediction*, in Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016, ACM, 2016, pp. 1069–1078.

[55] G. O. Mohler, M. B. Short, P. J. Brantingham, F. P. Schoenberg, and G. E. Tita, *Self-exciting point process modelling of crime*, Journal of the American Statistical Association, 106 (2012), pp. 100–108.

[56] J. Moller and J. G. Rasmussen, *Perfect simulation of hawkes processes*, Advances in Ap-

plied Probability, 37 (2010), pp. 629–646.
[57] K. Muandet, K. Fukumizu, B. Sriperumbudur, and B. Schölkopf, *Kernel mean embedding of distributions: A review and beyond*, Foundations and Trends® in Machine Learning, 10 (2017), pp. 1–141, https://doi.org/10.1561/2200000060, http://dx.doi.org/10.1561/2200000060.
[58] Y. Ogata, *On lewis' simulation method for point processes*, IEEE Transactions on Information Theory, 27 (1981), pp. 23–31.
[59] Y. Ogata, *Seismicity analysis through point-process modelling: A review*, Pure and Applied Geophysics, 155 (1999), pp. 471–507.
[60] T. Omi, N. Ueda, and K. Aihara, *Fully neural network based model for general temporal point processes*, in Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada, H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, eds., 2019, pp. 2120–2129.
[61] N. Privault, *Recursive computation of the hawkes cumulants*, 2020, https://arxiv.org/abs/2012.07256.
[62] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*, Adaptive computation and machine learning, MIT Press, 2006.
[63] A. Reinhart, *A review of self-exciting spatio-temporal point processes and their applications*, Statistical Science, 33 (2018), p. 299–318, https://doi.org/10.1214/17-sts629, http://dx.doi.org/10.1214/17-STS629.
[64] D. J. Rezende and S. Mohamed, *Variational inference with normalizing flows*, in Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015, F. R. Bach and D. M. Blei, eds., vol. 37 of JMLR Workshop and Conference Proceedings, JMLR.org, 2015, pp. 1530–1538.
[65] M. Rizoiu, Y. Lee, and S. Mishra, *Hawkes processes for events in social media*, in Frontiers of Multimedia Research, 2018, pp. 191–218.
[66] M. Rizoiu, S. Mishra, Q. Kong, M. J. Carman, and L. Xie, *Sir-hawkes: Linking epidemic models and hawkes processes to model diffusions in finite populations*, in Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018, ACM, 2018, pp. 419–428.
[67] M. G. Rodriguez and I. Valera, *Learning with temporal point processes*. http://learning.mpi-sws.org/tpp-icml18/, 2018.
[68] F. Salehi, W. Trouleau, M. Grossglauser, and P. Thiran, *Learning hawkes processes from a handful of events*, in Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada, 2019, pp. 12694–12704.
[69] J. Shang and M. Sun, *Geometric hawkes processes with graph convolutional recurrent neural networks*, in The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, AAAI Press, 2019, pp. 4878–4885.
[70] O. Shchur, M. Bilos, and S. Günnemann, *Intensity-free learning of temporal point processes*, in 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net, 2020.
[71] O. Shchur, N. Gao, M. Bilos, and S. Günnemann, *Fast and flexible temporal point processes with triangular maps*, CoRR, abs/2006.12631 (2020), https://arxiv.org/abs/2006.12631.
[72] C. Shelton, Z. Qin, and C. Shetty, *Hawkes process inference with missing data*, 2018.
[73] I. Sutskever, O. Vinyals, and Q. V. Le, *Sequence to sequence learning with neural networks*, in Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, 2014, pp. 3104–3112.
[74] W. Trouleau, J. Etesami, M. Grossglauser, N. Kiyavash, and P. Thiran, *Learning hawkes processes under synchronization noise*, in Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, vol. 97 of Proceedings of Machine Learning Research, PMLR, 2019, pp. 6325–6334.
[75] U. Upadhyay, A. De, and M. Gomez-Rodriguez, *Deep reinforcement learning of marked temporal point processes*, CoRR, abs/1805.09360 (2018), http://arxiv.org/abs/1805.09360.
[76] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention is all you need*, in Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, eds., 2017, pp. 5998–6008.

[77] H. Wang, L. Xie, A. Cuozzo, S. Mak, and Y. Xie, *Uncertainty quantification for inferring hawkes networks*, CoRR, abs/2006.07506 (2020), https://arxiv.org/abs/2006.07506.
[78] Y. Wang, G. Williams, E. Theodorou, and L. Song, *Variational policy for guiding point processes*, in Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, 2017, pp. 3684–3693.
[79] Y. Wang, B. Xie, N. Du, and L. Song, *Isotonic hawkes processes*, in Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016, 2016, pp. 2226–2234.
[80] S. Wei, S. Zhu, M. Zhang, and Y. Xie, *Goodness-of-fit test for self-exciting processes*, CoRR, abs/2006.09439 (2020), https://arxiv.org/abs/2006.09439, https://arxiv.org/abs/2006.09439.
[81] S. Xiao, M. Farajtabar, X. Ye, J. Yan, X. Yang, L. Song, and H. Zha, *Wasserstein learning of deep generative point process models*, in Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, 2017, pp. 3250–3259.
[82] S. Xiao, H. Xu, J. Yan, M. Farajtabar, X. Yang, L. Song, and H. Zha, *Learning conditional generative models for temporal point processes*, in Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, 2018, pp. 6302–6310.
[83] S. Xiao, J. Yan, M. Farajtabar, L. Song, X. Yang, and H. Zha, *Joint modeling of event sequence and time series with attentional twin recurrent neural networks*, CoRR, abs/1703.08524 (2017).
[84] S. Xiao, J. Yan, X. Yang, H. Zha, and S. M. Chu, *Modeling the intensity function of point process via recurrent neural networks*, in Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA., 2017, pp. 1597–1603.
[85] H. Xu, L. Carin, and H. Zha, *Learning registered point processes from idiosyncratic observations*, in Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, 2018, pp. 5439–5448.
[86] H. Xu, M. Farajtabar, and H. Zha, *Learning granger causality for hawkes processes*, in Proceedings of the International Conference on Machine Learning, 2016, pp. 1717–1726.
[87] H. Xu, D. Luo, and H. Zha, *Learning hawkes processes from short doubly-censored event sequences*, in Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, 2017, pp. 3831–3840.
[88] J. Yan, *Recent advance in temporal point process: from machine learning perspective*. http://thinklab.sjtu.edu.cn/src/pp_survey.pdf, 2019.
[89] J. Yan, X. Liu, L. Shi, C. Li, and H. Zha, *Improving maximum likelihood estimation of temporal point process via discriminative and adversarial learning*, in Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden., 2018, pp. 2948–2954.
[90] G. Yang, Y. Cai, and C. K. Reddy, *Recurrent spatio-temporal point process for check-in time prediction*, in Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018, ACM, 2018, pp. 2203–2211.
[91] Y. Yang, J. Etesami, N. He, and N. Kiyavash, *Online learning for multivariate hawkes processes*, in Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, 2017, pp. 4944–4953.
[92] B. Yuan, H. Li, A. L. Bertozzi, P. J. Brantingham, and M. A. Porter, *Multivariate spatiotemporal hawkes processes and network reconstruction*, 2018, https://arxiv.org/abs/1811.06321.
[93] A. Zarezade, A. De, H. R. Rabiee, and M. Gomez-Rodriguez, *Cheshire: An online algorithm for activity maximization in social networks*, CoRR, abs/1703.02059 (2017).
[94] A. Zarezade, U. Upadhyay, H. R. Rabiee, and M. Gomez-Rodriguez, *Redqueen: An online algorithm for smart broadcasting in social networks*, in Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017, 2017, pp. 51–60.
[95] Q. Zhang, A. Lipani, Ö. Kirnap, and E. Yilmaz, *Self-attentive hawkes processes*, CoRR, abs/1907.07561 (2019).
[96] R. Zhang, C. J. Walder, and M. Rizoiu, *Sparse gaussian process modulated hawkes process*,

CoRR, abs/1905.10496 (2019).

[97] R. Zhang, C. J. Walder, M. Rizoiu, and L. Xie, *Efficient non-parametric bayesian hawkes processes*, in Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, S. Kraus, ed., ijcai.org, 2019, pp. 4299–4305.

[98] Q. Zhao, M. A. Erdogdu, H. Y. He, A. Rajaraman, and J. Leskovec, *Seismic: A self-exciting point process model for predicting tweet popularity*, in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 1513–1522.

[99] F. Zhou, Z. Li, X. Fan, Y. Wang, A. Sowmya, and F. Chen, *Fast multi-resolution segmentation for nonstationary hawkes process using cumulants*, International Journal of Data Science and Analytics, 10 (2020), https://doi.org/10.1007/s41060-020-00223-3.

[100] K. Zhou, H. Zha, and L. Song, *Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes*, in Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2013, Scottsdale, AZ, USA, April 29 - May 1, 2013, vol. 31, JMLR.org, 2013, pp. 641–649.

[101] K. Zhou, H. Zha, and L. Song, *Learning triggering kernels for multi-dimensional hawkes processes*, in Proceedings of the International Conference on Machine Learning, 2013, pp. 1301–1309.

[102] S. Zuo, H. Jiang, Z. Li, T. Zhao, and H. Zha, *Transformer hawkes process*, CoRR, abs/2002.09291 (2020), https://arxiv.org/abs/2002.09291.