FastPoint: Scalable Deep Point Processes

Ali Caner Türkmen[1,⋆] ✉, Yuyang Wang[2], and Alexander J. Smola[2]

[1] Department of Computer Engineering, Boğaziçi University, 34342 Istanbul, Turkey
[2] Amazon Research, Palo Alto CA, USA
caner.turkmen@boun.edu.tr
{yuyawang, smola}@amazon.com

Abstract. We propose FastPoint, a novel multivariate point process that enables fast and accurate learning and inference. FastPoint uses deep recurrent neural networks to capture complex temporal dependency patterns among different marks, while self-excitation dynamics within each mark are modeled with Hawkes processes. This results in substantially more efficient learning and scales to millions of correlated marks with superior predictive accuracy. Our construction also allows for efficient and parallel sequential Monte Carlo sampling for fast predictive inference. FastPoint outperforms baseline methods in prediction tasks on synthetic and real-world high-dimensional event data at a small fraction of the computational cost.

1 Introduction

Many applications produce large data sets that can be viewed as sets of events with "timestamps", occurring asynchronously. Examples abound, such as user activity on social media, earthquakes, purchases in online retail, order arrivals in a financial market, and "spiking" activity on a neuronal circuit. Modeling complex co-occurrence patterns of such events and predicting future occurrences are of practical interest in a wide range of use-cases.

Temporal point processes (TPP) are probabilistic models of such data, namely *discrete event* sets in continuous time. They have been extended widely to describe patterns through which events (*points*) interact, and to model side information available in the form of features (*marks*). However, TPPs pose two key challenges:

⋆ Work done while intern at Amazon

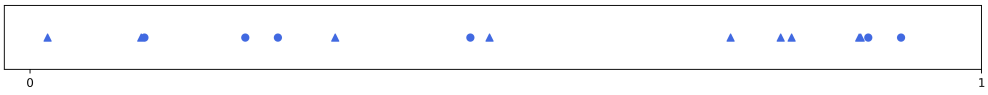


Fig. 1. Typical draw from a bivariate point process on the unit interval. Events occur in continuous time and belong to one of two types (marks) represented here as triangles and discs.

- How does one design an expressive model that can capture complex dependency patterns among events, while keeping the computational cost of learning manageable?
- How does one perform predictive inference, *i.e.,* describe distributions of how events will occur in the future, efficiently?

The first question is often addressed by a class of TPPs defined in terms of their *conditional intensity function* [3], *i.e.,* the instantaneous rate of events given previous points. A popular example is the Hawkes process [9], where the intensity is a linear function of the effects of past events. These models and variants have been explored in a range of application domains [1, 7, 8, 24]. Recently, recurrent neural networks (RNNs) have been used to approximate the conditional intensity function [6, 13, 21]. By conditioning intensity on a vector embedding of the event history, RNN-based models sidestep an important computational challenge in likelihood-based parameter estimation for TPPs. However when the point process is *multivariate*, *i.e.*, when events are identified as members of a finite set of processes such as purchases of a certain product or tweets of a certain user, the model specification must be extended to account for how these event *types* (or *marks*) interact [16, 17]. In both Hawkes processes and RNN-based approximations, computational difficulties associated with learning and inference are greatly exacerbated by *high dimensionality* – a large number of marks.

The second problem requires characterizing distributions of event patterns in a future interval, which leads to an intractable integral over all possible "point configurations". A popular alternative is Monte Carlo estimation, where forward samples from the process are taken to evaluate estimates. However, forward sampling from a point process is costly. When high-dimensionality is a concern, sampling is further complicated by drawing from the mark distribution, recomputed for each point.

In this paper, we propose a novel model, FastPoint, for efficient learning and approximate inference in multivariate TPP. We combine the expressiveness of RNNs to model mutual excitation (between marks), with well-studied Hawkes processes to capture local (within marks) temporal relationships. This results in significantly faster learning with better generalization in the high-dimensional setting. Our contributions can be summarized as follows,

- We introduce a novel multivariate TPP that uses deep RNNs as the backbone to capture mutual excitation relationships among different marks (*e.g.,* among different users on a network or different items in online retail) while using Hawkes processes to capture local dynamics. By trading off the granularity at which cross-mark dynamics are captured, FastPoint can scale to millions of correlated point processes.
- Our construction leads to favorable computational properties including reduced time complexity and enables parallel and distributed learning. Learning in high-dimensional point processes is accelerated by over an order of magnitude.

- Our model leads to a more parsimonious description of temporal dynamics and better generalization in an array of real-world problems compared to RNN-based models and Hawkes processes.
- FastPoint's unique construction can be exploited for a sequential Monte Carlo (SMC) routine that allows for substantially faster simulation and inference. This results in predictive estimates of equivalent variance for less than a percent of the computation time in comparable methods.

We introduce the required background on TPPs, neural TPP variants, and concerns in sampling in Section 2. We introduce our model and algorithm in Section 3. Section 4 presents related work, and Section 5 discusses empirical results attained on datasets from social media, user behavior in music streaming, and earthquake occurrences. Section 6 concludes the paper.

2 Background

TPPs are statistical models of *discrete (instantaneous) events* localized in *continuous time* [3]. Concretely, just as a draw from a univariate continuous probability distribution is a real number; a draw from a point process on a bounded set $(0, T]$ is a set of points $\{t_i\}_{i=1}^{N}$, $0 < t_1 < \cdots < t_N \leq T$.

Events (indexed here by $i \in \{1, \ldots, N\}$) at times t_i may be equipped with *marks*, $y_i \in \mathcal{F}$. When $\mathcal{F}$ is a finite set, indexed by $k \in \{1, \ldots, K\}$, an equivalent formalism is *multivariate* (or *multitype*) TPPs – *i.e.*, a set of K (correlated) point processes. For example, letting k index users, multivariate TPPs can be used to jointly model timestamps on their tweeting activity. Figure 1 represents a draw from a bivariate ($K = 2$) point process.

The Poisson process is the "archetypal" point process [14], and it is characterized by two main assumptions. First, one assumes that the point process is *simple*, *i.e.,* no two points coincide almost surely. Second is the assumption of independence: point occurrences on disjoint subsets of $\mathbb{R}$ are independent. While the first condition will underlie all point processes introduced here, it is this second assumption of independence that limits a realistic understanding of real-world phenomena. Many real-world events not only occur due to exogenous factors but *excite* or *inhibit* each other. For example, earthquakes excite nearby fault lines and increase the probability of "aftershocks". Social media activity elicits responses from other users. To capture such effects, one needs a richer class of TPPs than Poisson processes.

A convenient way of writing a TPP in which events depend on each other is through the *conditional intensity* function. We heuristically define the conditional intensity λ^* [3] as the probability of observing a point in the infinitesimal interval after time t, given history $\mathcal{H}_t$. Concretely,

$$\lambda^*(t) = \lim_{\delta \downarrow 0} \frac{\mathbb{P}\{N(t, t+\delta] > 0 | \mathcal{H}_t\}}{\delta},$$

where $N(a, b]$ is the random variable corresponding to the number of points in the interval $(a, b]$.

The conditional intensity uniquely determines a TPP. The log likelihood of a set of parameters of the conditional intensity Θ_{λ^*}, given realization $\{t_i\}$, can be written in terms of the conditional intensity,

$$\ell(\Theta_{\lambda^*}) = \sum_i \log \lambda^*(t_i) - \int_0^T \lambda^*(s)ds. \tag{1}$$

A concrete example of conditional intensity TPPs is the *Hawkes process* [1, 9, 10] which captures *self-excitation* behavior based on two assumptions: additivity and linearity. The conditional intensity of a (univariate) Hawkes process is

$$\lambda^*(t) = \mu + \sum_{t_j<t} \varphi(t - t_j), \tag{2}$$

where φ is a positive and causal kernel function. A common kernel is the exponential decay $\varphi(x) = \alpha\beta \exp(-\beta(x))$. The Hawkes process lends itself to interpretation as a branching (immigration-birth) process in continuous time [11]. In this sense, the *branching ratio* α corresponds to the long-run average number of "child" events a given event causes (or "excites"). $\beta \exp(-\beta(x))$, in turn, is the *delay density*, the probability density of the delay between parent and child events. For $\alpha < 1$, the process satisfies the *stationarity* condition.

The fundamental difficulty in fitting Hawkes processes, or any general point process defined via the conditional intensity, is that computing the likelihood (1) takes time quadratic in the number of events. Note (eqns. (1), (2)) that the computation of $\lambda^*(t_i)$ is a sum over all $\{t_j\}_{j<i}$, and that the likelihood requires computing intensities of all observed points. Computational issues are exacerbated by *multivariate* processes where one must account for relationships among K marks. A notable exception to quadratic-time likelihood computations is the exponential decay kernel which allows for likelihood computation in linear time (see Appendix A).

Scalability problems in parameter estimation were partially addressed by "neural point processes". Several recent contributions have proposed combining neural networks with conditional intensity TPPs. In Recurrent Marked TPP (RMTPP), Du et al. [6] propose to model a multivariate point process via an approximation to the conditional intensity function. This is achieved by an RNN, in their experiments an LSTM [12]. Effectively, the LSTM embeds the event history $\mathcal{H}_t = \{(t_i, y_i)|t_i < t\}$ to a vector, on which the conditional intensity function and the conditional distribution of the mark of the next point are calculated. Concretely, they take the conditional intensity

$$\lambda^*(t) = \exp(\mathbf{v}^\top \mathbf{h}_j + \beta(t - t_j) + b), \tag{3}$$

where β, b are scalar parameters, $\mathbf{v}$ is a vector parameter of appropriate dimension. $\mathbf{h}_j$ is the output of the LSTM for point t_j. That is, $\mathbf{h}_j = LSTM(\mathbf{h}_{j-1}, t_j, k_j)$. $j = \sup\{i \in \mathbb{N} : t_i < t\}$. Furthermore, they take, $y_{j+1} \sim \text{Categorical}(\text{softmax}(\mathbf{V}\mathbf{h}_j + \mathbf{b}))$, where $\mathbf{V}$, $\mathbf{b}$ are the weight and bias parameters of a dense neural network layer that maps LSTM outputs to the categorical likelihood.

RMTPP allows bypassing expensive optimization routines in general conditional intensity TPPs while leaving ample capacity for learning complex dependencies across time. The key observation in neural point processes is that $\mathbf{h}_j$ serves as a vector embedding for $\mathcal{H}_{t_j}$, and the intensity computation can be handled recursively. RMTPP is particularly convenient since it enables fast and easy implementation. The integral in the likelihood (the *compensator* term) can be computed exactly.

RMTPP was extended in [21], where the authors propose to parameterize the intensity function via a *continuous time* LSTM where the memory cell of the LSTM decays in time. This model, while more expressive as it captures several decaying influences, results in an intractable integral for computing the compensator.

Although TPP parameter estimation is greatly simplified by an approximation to the conditional intensity, performing predictive inference remains a significant challenge. Monte Carlo methods have emerged as the primary method for inference in TPPs, seeing as exact inference involves an intractable integral in all but the simplest models. Nevertheless, drawing exact samples from a TPP is a computationally cumbersome task: the points have to be sampled in sequence and the conditional intensity has to be re-evaluated at each point.

The traditional method for sampling from a TPP is Ogata's thinning method [22]. It is based on the observation that, conditioned on the history at a given point, the process until the next point can be cast as a non-homogeneous Poisson process. Then, if one can upper bound the intensity function, the next point can be drawn via "thinning", *i.e.*, proposing the next point from a faster homogeneous Poisson process and accepting based on the ratio of intensities. However, apart from the fact that the sampling routine has to be called in sequence, this algorithm introduces the computational cost of rejected points. We give a description of Ogata's algorithm in Appendix B.

3 FastPoint: Scalable Deep Point Process

3.1 Generative Model

Neural TPP models greatly simplify estimation in large-data (large N) regimes under a reasonable number of marks $K \approx 10^3$, while adding the ability to capture complex co-occurrence patterns. However, many real-world events have marks that are from a much larger set of possible values, *e.g.,* in events associated with millions of users or purchases from a catalog of several hundred thousand products.

The main computational difficulty arises from accounting for interactions between events from different marks in the same manner as one addresses interactions between events of the same mark. In traditional point processes, such as multivariate Hawkes processes – this leads to computing and bookkeeping for $O(K^2)$ branching parameters. In neural TPPs, training complexity reduces to $O(NK)$ which can still be prohibitively high. Furthermore, RNNs are well

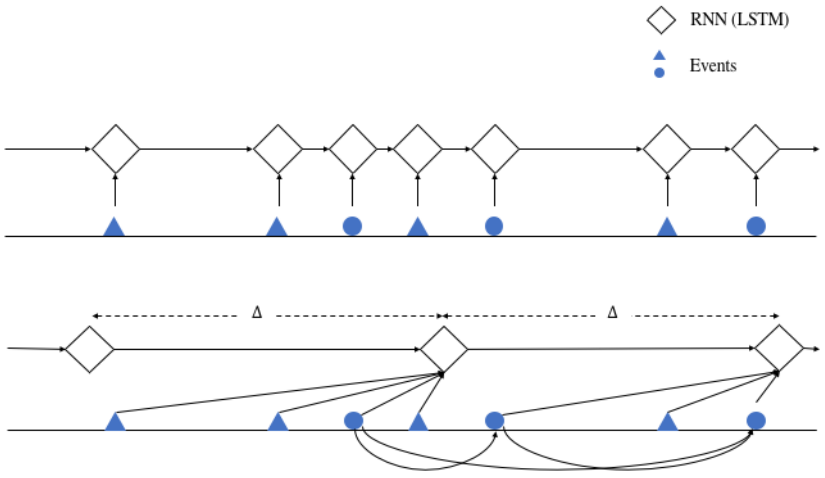


Fig. 2. Above: RMTPP computes the conditional intensity for the next point through an LSTM for each point in a sequence, Below: In FastPoint, the LSTM is conditioned on an interval of points, and computes added intensity for the next interval. The self-excitation in each mark, individually, is accounted for by a Hawkes process.

known to have difficulty in capturing long-range dependencies. This is especially important in high-dimensional temporal point processes since a large number of possibly unrelated marks are observed in a sequence before a relevant item is observed – an effect that could have easily been captured by self-exciting processes.

Our model, FastPoint, is built on a simple yet profound insight: mutual excitation dynamics can be modeled at a lower frequency than with which one accounts for self-excitation. More precisely, FastPoint addresses mutual-excitation on a fixed grid along time and through a deep neural network, while local self-exciting dynamics are captured with univariate Hawkes processes. We write the conditional intensity

$$\lambda_k^*(t) = g(\mathbf{v}_k^\top \mathbf{h}(\mathcal{H}_\tau) + b_k) + \mu_k + \sum_{\mathcal{H}_t^{(k)}} \varphi_k(t - t_i), \tag{4}$$

where g denotes the softplus function, $\tau = \sup\{\tau' \in \mathcal{G} | \tau' < t\}$, and $\mathcal{G} = \{0, \Delta, 2\Delta \dots\}$ denotes some uniformly sampled "grid". $\mathbf{v}, b_k, \mu_k$ are parameters, $\mathbf{h}(.)$ is a function implemented by an LSTM, and φ_k is the exponential decay kernel $\varphi_k(x) = \alpha_k \beta_k \exp(-\beta_k x)$.

FastPoint is composed of individual linearly self-exciting Hawkes processes to capture local effects in each process, as given by the second and third summands of (4). The first term is a non-negative added intensity contributed by an LSTM that "clocks" at set coarse intervals and "synchronizes" the processes. The inputs

Table 1. Comparison of training and sampling time complexities of multivariate temporal point processes

Process	**Conditional Intensity**	**Training Complexity**	**Sampling Complexity**
Poisson	$\lambda_k(t) = \bar{\lambda} p_k$ where $p_k = \bar{\lambda}_k / \bar{\lambda}$	$O(K)$	$O(N+K)$
Hawkes	$\lambda_k(t) = \mu_k + \sum_{\mathcal{H}_t} \varphi_k(t - t_i, y_i)$	$O(N^2 + NK)$	$O(N^2 K)$
RMTPP [6]	$\lambda_k(t) = p(k \mid h(\mathcal{H}_t)) f(t, h(\mathcal{H}_t))$	$O(NK)$	$O(NK)$
Neural Hawkes Process[21]	$\lambda_k(t) = f_k(\mathbf{w}_k^\top h(t))$ $h(t) = o_i \odot (2\sigma(2c(t)) - 1)$	$O(NK)$	$O(NK)$
FastPoint	$g(\mathbf{v}_k^\top \mathbf{h}(\mathcal{H}_\tau) + b_k) + \mu_k + \sum \varphi(t - t_i)$ $\tau = \sup\{\tau' < t \mid \tau' \in \mathcal{G}\}$ $\mathcal{G} = \{0, \Delta, 2\Delta, \dots\}$	$O(N + K\lvert\mathcal{G}\rvert)$	$O(N + K\lvert\mathcal{G}\rvert)$

of the LSTM are the interarrival times and embeddings of past marks of previous points, as in [6, 21]. We give a stylized depiction comparing FastPoint to other neural TPP models in Figure 2.

FastPoint can be interpreted as a global-local time series model [20], where the intensity processes are composed of a global component (given by the LSTM, the first term of (4)), and local components that are each a Hawkes process. While this greatly simplifies computation, it leads to a realistic-enough description of many real-world events. Our model encodes the assumption that mutual excitation often takes place with longer delays than self-excitation. For example, for limit order book analysis in finance, FastPoint models self-excitatory behavior of individual event sets (*i.e.,* assets) at ultra-high-frequency resolution, while cross-asset effects are captured at lower resolutions. Finally, note that FastPoint offers a general template for constructing multivariate TPPs. The global model (LSTM) can be changed with other deep neural network architectures such as the Transformer [25] or a simple multilayer perceptron. The local model can also be switched, *e.g.,* with a Poisson process.

FastPoint's key computational advantage is that it eliminates the need to compute K-many terms at each point for likelihood-based estimation. Intuitively, multivariate TPP likelihood computation requires a sequential pass over all points in an observation. Furthermore, the compensator term in the likelihood – *i.e.,* the probability that no points are observed in between each point has to be computed for *all* K marks, resulting in $O(NK)$ cost. FastPoint yields significant benefit in both respects. First, overall computational complexity decreases to $O(N+K|\mathcal{G}|)$, invoking both the memoryless property of exponential decays in the Hawkes process and favorable computational properties of RNN-based conditional intensity approximation. Second, the likelihood computation of individual marks can be parallelized over. In this manner, FastPoint is amenable to both massively

parallel and distributed implementations and solves a crucial scalability problem in point process estimation and simulation. See Table 1 for a comparison of computational complexities associated with different point processes. We give further details on FastPoint's construction, implementation, global and local model choices in Appendix A.

3.2 Sequential Monte Carlo Sampling

We turn to predictive inference, characterizing distributions of future event occurrences with FastPoint. Concretely, we seek to estimate expectations of the form,

$$\mathbb{E}_{\mathbb{P}}\left[\phi\left(\{(t_i, y_i)\}_{(t,t+T]}\right)|\mathcal{H}_t\right], \tag{5}$$

where $\{(t_i, y_i)\}_{(t,t+T]}$ denotes (random) realizations of an arbitrary marked point process $\mathbb{P}$ which we approximate with FastPoint, on the *forecast horizon* $(t, t+T]$. ϕ denotes some function of the data, *e.g.*, a summary statistic. For brevity, we denote $\mathit{\Pi} = \{(t_i, y_i)\}_{(t,t+T]}$, *i.e.,* the random variate corresponding to possible configurations of (a.s. finitely many) marked points on the given interval. Note that

$$\mathbb{E}\left[\phi(\mathit{\Pi})|\mathcal{H}_t\right] = \int_{\mathit{\Pi}\in\mathcal{X}} \phi(\mathit{\Pi}) f(\mathit{\Pi}|\mathcal{H}) d\mathit{\Pi}, \tag{6}$$

is a non-trivial integral over $\mathcal{X}$, the *point configuration space* [19].

We will rely on Monte Carlo methods for approximating $\mathbb{E}_{\mathbb{P}}\left[\phi(\mathit{\Pi})|\mathcal{H}_t\right]$. FastPoint already alleviates part of the computational burden associated with sampling from multivariate point processes. That is, it allows for simulating each mark individually, in parallel, between each LSTM computation. For each such interval, one could work with Ogata's thinning algorithm in parallel. This still results in the difficulty of sampling sequentially, with the added overhead of rejecting some of the points drawn. These computational issues are further complicated by the difficulty of implementing thinning in "batch" mode, in modern deep learning frameworks such as Apache MXNet.

We suggest an alternative approach hinted by the global-local assumption of FastPoint. We take sequential importance weighted samples to evaluate expectations of the form (5), by proposing from a suitably parameterized Poisson process. The sync points of the global model, on the grid $\mathcal{G}$, are natural points to serve as the epochs of a sequential Monte Carlo (SMC) algorithm [5]. Furthermore, we find that the intensity at the beginning each interval doubles as a good proposal intensity.

For short enough prediction horizons, the Poisson process constitutes an effective proposal in two regards. First, surprisingly, short enough intervals result in low-variance samples or high effective sample sizes (ESS). Furthermore, invoking homogeneity and the *thinning* property of Poisson processes [14], the times and the marks of future points can be sampled independently and in parallel. Concretely, for sampling from a multivariate homogeneous Poisson process with

Algorithm 1: Sequential Monte Carlo sampling of FastPoint

Input: T, Δ, M, c
begin
  $w_j \leftarrow 1, \forall j$
  $\tau \leftarrow t_0$
  while $\tau < T$ **do**
    for *particles* $j = 1$ ***to*** M **do in parallel**
      Compute $\bar{\lambda}_k = \lambda_k^*(\tau)$, $\forall k \in \{1, 2, \ldots, K\}$
      Draw $N_j \sim \text{Poisson}(\Delta \times \sum_k \bar{\lambda}_k)$
      Draw $\{t_i^{(j)}\}_{i=1}^{N_j} \sim \mathcal{PP}(\sum_k \bar{\lambda}_k)$
      for $i = 1$ **to** N_j **do in parallel**
        Draw $y_i^{(j)}$ s.t. $p(y_i^{(j)} = k) \propto \bar{\lambda}_k$
      endfor
      $w_j \leftarrow w_j \times \frac{p(\{(t_i^{(j)}, y_i^{(j)})\}|\mathcal{H}_\tau)}{q(\{(t_i^{(j)}, y_i^{(j)})\}|\bar{\lambda}, p^*)}$
    endfor
    $\tau \leftarrow \tau + \Delta$
    ESS $\leftarrow \|\mathbf{w}\|_1^2 / \|\mathbf{w}\|_2^2$
    if *ESS* $< c$ **then**
      Resample particles, s.t.
      $\{(t_i^{(j')}, y_i^{(j')})\}_{t \in (t_0, \tau]} = \{(t_i^{(j)}, y_i^{(j)})\}_{t \in (t_0, \tau]}$ with prob. $\propto w_j$, $\forall j'$
    end
  end
end

intensities λ_k, we can instead sample from a global Poisson process with intensity $\bar{\lambda} = \sum_k \lambda_k$. The marks can be drawn in parallel, each in constant time, with $p^*(k) = \lambda_k / \sum_k \lambda_k$.

These observations result in a straightforward SMC algorithm. Namely, we sample from Poisson processes in sequence, updating both the particle weights and Poisson process parameters. We give a concrete description of FastPoint-SMC in Algorithm 1, where we use w_j to denote the importance sampling *particle weights*, κ the resampling threshold and $\mathcal{PP}$ the Poisson process from which timestamps are drawn. Finally, $p(\{(t_i^{(j)}, y_i^{(j)})\}|\mathcal{H}_\tau), q(\{(t_i^{(j)}, y_i^{(j)})\}|\bar{\lambda}, p^*)$ denote the densities with respect to FastPoint and the Poisson process proposals respectively.

Counterintuitively, FastPoint-SMC scales well with respect to the number of marks K. To observe why, note that in practice, the number of points sampled in each interval is much smaller than K. For marks that are not drawn, the Poisson proposal density and FastPoint density are identical. In other words, marks for which no points were drawn do not contribute to the sample variance. Therefore, for short enough Δ, FastPoint effective sample sizes remain high for large K, scaling favorably to high dimensions in sampling as well as training.

4 Related Work

In [26], the authors introduce a Wasserstein Generative Adversarial Network for point processes, which leads to *likelihood-free* learning of generative models for point processes. Recently, latent variable neural network models for marked point process generation were explored in [23]. Both models use a generative deep network as the backbone of their construction and are hence easy to sample from. However, neither model is geared toward scalability in the number of marks.

Linderman et al. [18] explore a Rao-Blackwellized particle filter for inference in latent point processes, which could be used for predictive inference. However, the algorithm is only explored in the context of multivariate Hawkes processes and not extended to high-dimensional processes or neural TPPs. Somewhat similarly to our mix of discrete-time and continuous-time point process construction [27] explore a *twin* RNN architecture. However, their model employs yet another RNN to model continuous time effects, and is not amenable to high-dimensional modeling.

Scaling to high dimensions is a current and challenging problem in TPPs [1]. To our knowledge, FastPoint is the first model to consistently address very large discrete mark spaces, as well as the first to combine self-exciting processes with the neural TPP literature. Finally, ours is the first treatment of sequential Monte Carlo simulation in neural TPP, and one of the first to explore it for TPP in general.

5 Experiments

We implement RMTPP and FastPoint on Apache MXNet [2] with operators for Hawkes process likelihood and gradient computations in the MXNet backend[1]. For learning, we use MXNet Gluon's Adam optimizer. We run experiments on AWS p3 instances equipped with NVIDIA Tesla V100 GPUs.

5.1 Model Performance

We evaluate FastPoint's performance on large-scale, high-dimensional point process data. First, we compare generalization performance based on the log-likelihood of a held-out future time frame. We then compare computational performance via standardized computation times for learning.

We compare FastPoint's generalization performance to the following set of baseline models,

- Self-exciting **Hawkes process**, *i.e.,* a collection of univariate exponential-decay Hawkes processes as given in (2). Note that this baseline amounts to FastPoint with only the *local* model component.
- **RMTPP** [6], as given in Table 1.

[1] The code is made available as part of MXNet. See `https://github.com/apache/incubator-mxnet/blob/master/src/operator/contrib/hawkes_ll-inl.h`

Table 2. Negative log-likelihood loss of different point processes on a held-out sample

	HP-5K	**HP-10K**	**NCEDC**	**MemeTracker**	**LastFM-1K**
Events (millions) (N)	1	1	0.8	7.6	18
Marks (K)	5000	10000	1000	71566	105222
Hawkes	27009	30441	10346	42406	25087
RMTPP	27008	30491	14424	42507	30489
B-RMTPP	27008	30483	14393	42304	30474
FastPoint-5	26998	**30412**	10314	**41007**	25271
FastPoint-10	**26997**	30412	10287	41253	25024
FastPoint-20	26998	30412	**10261**	41398	**24500**

– **B-RMTPP**, a modified version of RMTPP given by the conditional intensity

$$\lambda_k(t) = \mu + p(k|h(\mathcal{H}_{t_j})) \exp(\mathbf{v}_\lambda^\top h(\mathcal{H}_{t_j}) + b_\lambda + \beta(t - t_j)),$$

adding a background intensity. While a seemingly simple modification, this makes RMTPP absolutely continuous with respect to the Poisson process. That is, RMTPP is a *terminating* point process, making simulation schemes such as Ogata's algorithm invalid. Apart from correcting for this theoretical issue, this formulation leads to better generalization.

We compare the predictive performance of FastPoint to baselines on several data sets,

– **HP-5K**, and **HP-10K** are synthetic data sets sampled from a multivariate Hawkes process with the number of marks (K) set to 5000 and 10000 respectively. We use `hawkeslib` [2] to generate 1 million events from Hawkes models parameterized by randomly drawn branching matrices.
– Earthquake events collected from the **NCEDC** earthquake catalog search service [4]. We collect 800K earthquake events in the Northern California area and cluster the events into marks based on their coordinates, associating them to one of 1000 "locales". The prediction task is to best represent the time and locales of earthquake occurrences.
– A subset of the **MemeTracker** data set [15], that includes timestamped records for 7.6 million social media sharing events of *memes*, belonging to one of 71566 clusters.
– The **LastFM-1K** [3] dataset, which includes 19 million records of *listening* events, belonging to one of 105222 artists.

For RMTPP and FastPoint, we set the number of hidden units to 50 in synthetic data experiments and 100 in real-data experiments respectively. We use early stopping and weight decay for regularization. To mitigate the effect of

[2] `http://github.com/canerturkmen/hawkeslib`
[3] `https://www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/lastfm-1K.html`

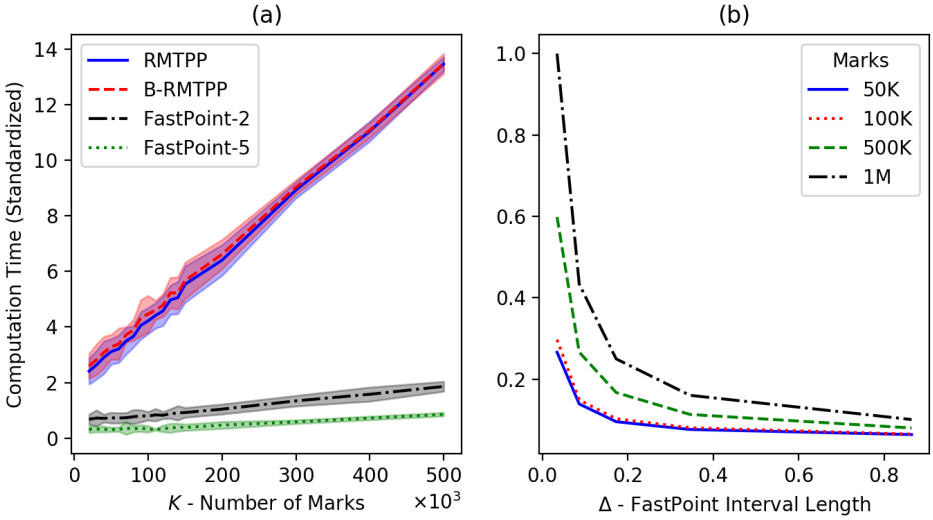


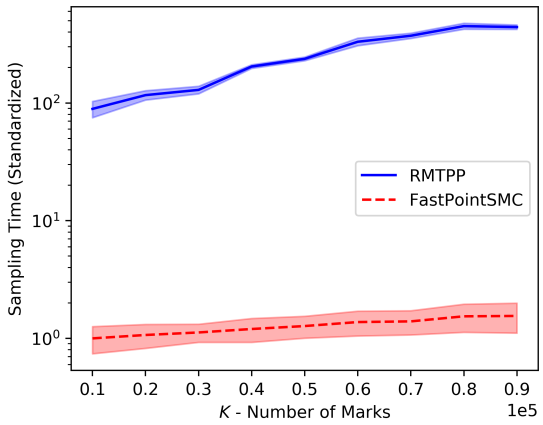
Fig. 3. (a) Training time on a single batch of 2500 events vs. the number of marks. `FastPoint-2` and `FastPoint-5` refer to the FastPoint with interval lengths set to 2 and 5 respectively. Numbers are reported as multiples of time taken by `FastPoint-2` on 10^4 marks. (b) FastPoint training time on a single batch of 25000 events as interval lengths are increased. Different lines correspond to different numbers of marks. Numbers are indexed to the time taken by for 10^6 marks with $\Delta = 0.01$
.

possible numerical issues on experimental outcomes, we normalize all data sets to the same time scale to an average intensity of 50 events per unit of time.

We compare predictive accuracy in terms of the negative log-likelihood – *i.e.,* average model loss on a held-out future interval of 5000 points. Note that a more interpretable measure of accuracy is difficult to define in point processes, and previous works have used a mix of predictive log-likelihood with other metrics such as squared error for timestamps or multiclass accuracy for marks [6]. However, we observe these metrics lead to little insight and high variance in high-dimensional processes.

We present our results in Table 2. We give outcomes for three different FastPoint alternatives, varying the LSTM interval length Δ. That is, we denote the $\Delta = 2$ case as `FastPoint-2`. We report average loss over a long held-out interval that includes at least half of the points in the full training set.

FastPoint categorically outperforms baselines in predictive accuracy. In synthetic data experiments, we observe that FastPoint leads to better generalization than both univariate Hawkes processes and neural TPP baselines. The margin of improvement widens in real-world data sets with greater K. The benefit of having a local model is especially notable in **NCEDC** and **LastFM-1K**. We



K ($\times 10^3$)	**Improvement**
10	89.1
20	109.1
30	115.1
40	169.9
50	186.3
60	240.7
70	267.2
80	292.0
90	284.7

Fig. 4. Left: Sampling times of RMTPP (Ogata's sampler) vs. FastPoint-SMC, indexed on FastPoint-SMC. **Right:** The factor of improvement in sampling times, *i.e.,* the multiple of time taken by RMTPP-Ogata relative to FastPoint-SMC. FastPoint results in gains of nearly 300x times.

also find that in practice, FastPoint is less prone to overfitting and converges reasonably quickly.

We contrast FastPoint's computational performance to other deep TPP models under increasing dimensionality. In Figure 3a, we compare computation times for processing a batch of 2500 events during training. For 500K marks, FastPoint improves on the computation time by over a factor of 20. Beyond 500K marks, the memory footprints of RMTPP and B-RMTPP grow to an unmanageable size making comparison impractical, though the trend is evident. Moreover, observe in Figure 3b that FastPoint allows trading off modeling accuracy by further decreasing granularity (increasing the LSTM interval length Δ). For example, allowing a wider interval of 20 time units, one can learn a model of a million marks in a reasonable amount of time. Combining the two effects, FastPoint can lead to faster training of over two orders of magnitude, not accounting for other side benefits such as the ability to work with larger batches of data, longer time intervals, or performing inference in parallel such as on multi-GPU architectures.

5.2 Sampling

We now present empirical findings on FastPoint's sampling performance, using the SMC sampler introduced in Section 3.2. We compare the time taken by RMTPP (using Ogata's thinning method) and the FastPoint SMC sampler to generate $N_s = 100$ samples from a learned point process, with a fixed forecast horizon of 10 time units. We present a comparison of standardized computation times in Figure 4, varying the number of marks K. FastPoint-SMC easily results in faster sampling by a factor of nearly 300x and its sampling time scales favorably with respect to the number of marks.

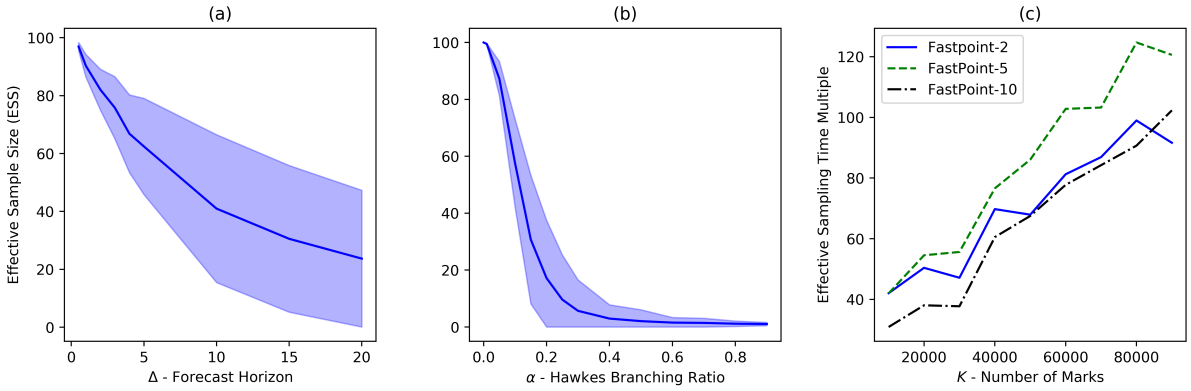


Fig. 5. (a) Effective sample sizes with increasing forecast horizon. (b) Effective sample sizes decay quickly with increasing branching ratio (c) Relative improvement in sampling time, accounting for decreases in ESS, increase with respect to K.

However, our analysis overlooks the fact that SMC generates samples that lead to higher variance estimates. Indeed, it is not apparent whether sequentially drawn importance-weighted samples from a Poisson process would lead to good estimates for FastPoint, especially in the presence of a large number of marks. Surprisingly, for small enough branching ratios α and short enough sampling intervals, the Poisson proposal leads to low variance samples. To demonstrate this, we compute the *effective sample size* (ESS, cf. Algorithm 1) for importance weighted samples of FastPoint, which corresponds to the number of exact samples that would result in equivalent variance.

In Figures 5a and 5b we present the ESS for 100 importance-weighted samples drawn on a single interval (without resampling or sequential sampling) of length Δ and for fixed branching ratios (for all marks) of α. We set the number of marks to 10^4, and the average intensity to 50 points per unit of time. We first observe that SMC produces reasonably efficient samples as the forecast horizon increases, resulting in an ESS of 60 for a horizon of 5 time units (roughly, 250 points). However, we also find that the ESS decays quickly as the branching ratio α increases beyond 0.1. In practice, however, FastPoint accounts for part of the self-excitation behavior through the global model, leading to smaller branching rations for most marks.

We find that the SMC routine performs well in terms of sampling efficiency as the dimensionsality increases. In Figure 5c, we compute the Effective Sampling Time Multiple (ESTM). Letting $T_{FP}^{(s)}, T_{O}^{(s)}$ denote the time taken to sample from FastPoint-SMC and RMTPP-Ogata respectively, we define $ESTM = \frac{T_O^{(s)}}{T_{FP}^{(s)}} \times \frac{ESS}{N_S}$. This summary metric roughly corresponds to the factor by which FastPoint accelerates sampling, accounting for the higher variance introduced by SMC. Setting $\alpha = 0.05, \Delta = 1$, we vary the number of marks to find that FastPoint

results in greater speed by a factor of over two orders of magnitude, for estimates of equivalent variance.

6 Conclusion

Multivariate point processes are natural models for many real-world data sets. However, due to the computational complexity often associated with learning and inference in TPPs, other simplified models (*e.g.*, by discretizing time or assuming independent marks) have been favored over them in many application domains. Moreover, most existing approaches do not address high-dimensional multivariate TPPs, a case that often arises in practice, with an expressive model that scales well in terms of generalization performance and computational cost. Finally, performing simulation (sampling) efficiently in general multivariate TPPs is an open problem.

FastPoint combines the interpretability and well-understood theory of Hawkes models with recurrent neural networks, addressing these long-standing challenges in point process modeling. First, it unlocks scalable estimation and simulation in millions for correlated point processes via a parsimonious global-local model. It can be used for accurate modeling of high-dimensional asynchronous event data, such as item purchases in a very large catalog, activities on a web-scale social graph, or limit order events in an order book with thousands of assets. Second, our SMC algorithm allows efficient sampling, accelerating predictive inference by over two orders of magnitude.

FastPoint's global-local point process construction is flexible. The global and local model components can be changed to other models best suitable for the task. Exploring other global-local multivariate point process constructions and better understanding their properties for learning and sampling remain exciting avenues for future research.

References

1. Bacry, E., Mastromatteo, I., Muzy, J.F.: Hawkes processes in finance. Market Microstructure and Liquidity **1**(01), 1550005 (2015)
2. Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., Xiao, T., Xu, B., Zhang, C., Zhang, Z.: MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems. arXiv preprint arXiv:1512.01274 (2015)
3. Daley, D.J., Vere-Jones, D.: An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods. Springer Science & Business Media (2007)
4. doi:10.7932/NCEDC, N.U.B.S.L.D.: Northern California earthquake data center (2014)
5. Doucet, A., Johansen, A.M.: A tutorial on particle filtering and smoothing: fifteen years later. Handbook of nonlinear filtering **12**(656-704), 3 (2009)
6. Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., Song, L.: 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. pp. 1555–1564. ACM (2016)

7. Du, N., Farajtabar, M., Ahmed, A., Smola, A.J., Song, L.: 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. pp. 219–228. ACM (2015)
8. Du, N., Song, L., Yuan, M., Smola, A.J.: Learning networks of heterogeneous influence. In: Advances in Neural Information Processing Systems. pp. 2780–2788 (2012)
9. Hawkes, A.G.: Point spectra of some mutually exciting point processes. Journal of the Royal Statistical Society. Series B (Methodological) pp. 438–443 (1971)
10. Hawkes, A.G.: Spectra of some self-exciting and mutually exciting point processes. Biometrika **58**(1), 83–90 (1971)
11. Hawkes, A.G., Oakes, D.: A cluster process representation of a self-exciting process. Journal of Applied Probability **11**(3), 493–503 (1974)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
13. Jing, H., Smola, A.J.: Neural survival recommender. In: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. pp. 515–524. ACM (2017)
14. Kingman, J.F.C.: Poisson processes, vol. 3. Clarendon Press (1992)
15. Leskovec, J., Backstrom, L., Kleinberg, J.: Meme-tracking and the dynamics of the news cycle. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 497–506. ACM (2009)
16. Linderman, S., Adams, R.: Discovering latent network structure in point process data. In: International Conference on Machine Learning. pp. 1413–1421 (2014)
17. Linderman, S.W., Adams, R.P.: Scalable Bayesian inference for excitatory point process networks. arXiv preprint arXiv:1507.03228 (2015)
18. Linderman, S.W., Wang, Y., Blei, D.M.: Bayesian inference for latent Hawkes processes. Advances in Neural Information Processing Systems (2017)
19. Liniger, T.J.: Multivariate Hawkes processes. Ph.D. thesis, ETH Zurich (2009)
20. Maddix, D.C., Wang, Y., Smola, A.: Deep factors with Gaussian processes for forecasting. arXiv preprint arXiv:1812.00098 (2018)
21. Mei, H., Eisner, J.M.: The neural Hawkes process: A neurally self-modulating multivariate point process. In: Advances in Neural Information Processing Systems. pp. 6754–6764 (2017)
22. Ogata, Y.: On Lewis' simulation method for point processes. IEEE Transactions on Information Theory **27**(1), 23–31 (1981)
23. Sharma, A., Johnson, R., Engert, F., Linderman, S.: Point process latent variable models of larval zebrafish behavior. In: Advances in Neural Information Processing Systems. pp. 10941–10952 (2018)
24. Simma, A., Jordan, M.I.: Modeling events with cascades of Poisson processes. In: Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence. pp. 546–555. AUAI Press (2010)
25. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. pp. 5998–6008 (2017)
26. Xiao, S., Farajtabar, M., Ye, X., Yan, J., Song, L., Zha, H.: Wasserstein learning of deep generative point process models. In: Advances in Neural Information Processing Systems. pp. 3247–3257 (2017)
27. Xiao, S., Yan, J., Farajtabar, M., Song, L., Yang, X., Zha, H.: Joint modeling of event sequence and time series with attentional twin recurrent neural networks. arXiv preprint arXiv:1703.08524 (2017)