Long Horizon Forecasting With Temporal Point Processes

Prathamesh Deshpande
IIT Bombay

Kamlesh Marathe
IIT Bombay

Abir De
IIT Bombay

Sunita Sarawagi
IIT Bombay

ABSTRACT

In recent years, marked temporal point processes (MTPPs) have emerged as a powerful modeling machinery to characterize asynchronous events in a wide variety of applications. MTPPs have demonstrated significant potential in predicting event-timings, especially for events arriving in near future. However, due to current design choices, MTPPs often show poor predictive performance at forecasting event arrivals in distant future. To ameliorate this limitation, in this paper, we design DualTPP which is specifically well-suited to long horizon event forecasting. DualTPP has two components. The first component is an intensity free MTPP model, which captures microscopic event dynamics by modeling the time of future events. The second component takes a different dual perspective of modeling aggregated counts of events in a given time-window, thus encapsulating macroscopic event dynamics. Then we develop a novel inference framework jointly over the two models by solving a sequence of constrained quadratic optimization problems. Experiments with a diverse set of real datasets show that DualTPP outperforms existing MTPP methods on long horizon forecasting by substantial margins, achieving almost an order of magnitude reduction in Wasserstein distance between actual events and forecasts.

ACM Reference Format:
Prathamesh Deshpande, Kamlesh Marathe, Abir De, and Sunita Sarawagi. 2021. Long Horizon Forecasting With Temporal Point Processes. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining (WSDM '21), March 8–12, 2021, Virtual Event, Israel.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3437963.3441740

1 INTRODUCTION

In recent years, marked temporal point processes (MTPPs) have emerged as a powerful tool in modeling asynchronous events in a diverse set of applications, such as information diffusion in social networks [8, 11, 12, 26, 27, 39, 57, 59], disease progression [37, 40, 41, 56], traffic flow [33], and financial transactions [15, 17, 19, 22, 30, 44]. MTPPs are realized using two quantities: (i) intensity functions which characterize the probabilities of arrivals of subsequent events, based on the history of previous events; and (ii) the distribution of marks which captures extra information attached with each event *e.g.*, sentiment in a Tweet, location in traffic flow, etc.

Over the myriad applications of MTPPs, we identify two modes in which MTPPs are used during prediction: (i) nowcasting, which implies prediction of only the immediate next event; and, (ii) forecasting, which requires prediction of events in a distant future. Forecasting continuous-time events with TPP models has a wide variety of use cases. For example, in emergency planning, it can assist resource allocation by anticipating demand; in transportation, it can help in congestion management; and in a social network , it can help to anticipate the rise of an orchestrated campaign. In this work, our goal is to develop a temporal point process model, which is specifically suited for accurate forecasting of arrival of events in the long term given a history of events in the past.

Limitations of prior work. Predictive models of temporal point processes have been extensively researched in recent literature [3, 7, 13, 29, 31, 34, 45, 47, 52, 55, 58]. A predominate approach is to train an intensity function for the next event conditioned on historical events, and then based on this estimated intensity function, forward sample events to predict a sequence of events in the future. While these approaches have shown promise at predicting the arrival of events in the near future, they suffer from two major limitations.

I Their modeling frameworks heavily rest on designing the intensity function— which in turn can sample only the next subsequent event. Such a design choice allows these models to be trained only for nowcasting rather than forecasting.
II Over long time horizons, the forward sampling method accumulates cascading errors as we condition on predicted events to generate the next event, whereas during training we condition on true events. Existing approaches [51, 54] of handling this mismatch via sequence-level losses provide only modest gains.

Present work. Responding to the limitations of prior approaches, we develop DualTPP, which is specifically designed to forecast events over long time horizons. The DualTPP model consists of two components. The first component encapsulates the event dynamics at a microscopic scale, whereas the second component views the same event dynamics from a different perspective and at a higher macroscopic scale. The first component is an intensity free recurrent temporal point process, which models the *time* of events conditioned on all previous events along with marks. This model has sufficient predictive ability to capture the event arrival process in the immediate future, but like existing TPPs is subject to cascading drift. The second component models the *count* of events over fixed time-intervals in the long-term future. Together, this leads to an accurate modeling of both short and long term behavior of the associated event arrival process.

WSDM '21, March 8–12, 2021, Virtual Event, Israel

ACM ISBN 978-1-4503-8297-7/21/03…$15.00
https://doi.org/10.1145/3437963.3441740

Inference in DualTPP involves forecasting events while achieving consensus across predictions from both models. This presents new algorithmic challenges. We formulate a novel joint inference objective on the two models, and show how to decompose it into a sequence of constrained concave quadratic maximization problems over continuous variables, combined with a binary search over discrete count variables. Our algorithm provides a significant departure from existing sampling-based inference that are subject to gross inaccuracies.

Our model includes both elements of multi-scale modeling like in hierarchies and multi-view learning. We show that this form of multi-view, multi-scale modeling, coupled with our joint inference algorithm, provides more accurate long-term forecasting than just multi-scale models [6, 46, 47]. We provide a comprehensive evaluation of our proposal across several real world datasets. Our experiments show that the proposed model outperforms several state-of-the-art baselines in terms of forecasting accuracy, by a substantial margin.

Summary of contributions. Summarizing, we make the following contributions in this paper.
— *Forecasting aware modeling framework:* We propose a novel forecasting aware modeling framework for temporal point process, which consists of two parts— the first part captures the low-level microscopic behavior, whereas the other part captures the high level macroscopic signals from a different perspective. These two components complement the predictive ability of each other, that helps the joint model to accurately characterize the long horizon behavior of the event dynamics.
— *Efficient inference protocol:* We devise a novel inference method to forecast the arrival of events during an arbitrary time-interval. In sharp contrast to expensive sampling procedures, the proposed inference method casts the forecasting task as a constrained quadratic optimization problem, which can be efficiently solved using standard tools.
— *Comprehensive evaluation:* Our proposal is not only theoretically principled, but also practically effective. We show superior predictive ability compared to several state-of-the-art algorithms. Our experiments are on practically motivated datasets spanning applications in social media, traffic and emergency planning. The substantial gains we obtain over existing methods establish our practical impact on these applications.

2 RELATED WORK

Our work is related to temporal point processes, long-term forecasting in time-series, and peripherally with the area of multi-view learning.

Temporal point process. Modeling continuous time event streams with temporal point processes (TPPs) follow two predominant approaches. The first approach focuses on characterizing TPPs using fixed parameterizations, by means of linear or quasi-linear forms of intensity functions [5, 20, 21, 23, 32], *e.g.*, Hawkes process, self-correcting process, etc. Such TPP models are designed to capture specific phenomena of interest. For example, Hawkes process encapsulates the self-exciting nature of information diffusion in online social networks whereas, Markov modulated point process can accurately model online check-ins. While such models provide interpretability, their fixed parameterizations often lead to model mis-specifications, limited expressiveness, which in turn constrain their predictive power. The second approach overcomes such limitations by designing deep neural TPP models, guided by a recurrent neural network which captures the dependence of previous events on the arrival of subsequent events. Du et al. [13] proposed Recurrent Marked Temporal Point Process (RMTPP), a three layer neural architecture for TPP model, which relies on a vanilla RNN to capture the dependence between inter-event arrival times. Such a design is still the workhorse of many deep recurrent TPP models. Neural Hawkes process provides a robust nonlinear TPP model, which can incorporate the effect of missing data. However, these models heavily rest on learning the arrival dynamics of one subsequent event and as a consequence, they show poor forecasting performance. Recently, a number of more powerful deep learning techniques have been borrowed to capture richer dependencies among events in TPPs. For example, Xiao et al. [54] proposes a sequence to sequence encoder-decoder model for predicting *next k* events; Xiao et al. [51] use Wasserstein GANs to generate an entire sequence of events; Vassøy et al. [47] deploy a hierarchical model; and Zuo et al. [60] apply transformer architecture to capture the dependence among events via self-attention. We compare DualTPP against these methods in Section 5 and show substantial gains.

Long-term Forecasting in Time Series. The topic of long-term forecasting has been more explored in the regular time-series setting than in the TPP setting. Existing time-series models are also auto-regressive and trained for one-step ahead prediction [16], and subject to similar phenomenon of cascading errors when used for long-range forecasting. Efforts to fix the teacher-forcing training of these one-step ahead model to adapt better to multi-step forecasting [49], have been not as effective as breaking the auto-regressive structure to directly predict for each future time-step [4, 9, 50]. Another idea is to use dilated convolutions, as successfully deployed in Wavenet [46] for audio generation, that connect each output to successively doubling hops into the past [6]. A hierarchical model that we compared with in Section 5.2 also uses dilated connections to past events. We found that this model provided much better long-range forecasts than existing TPP models, however our hybrid event-count model surpassed it consistently. A third idea is to use a loss function [28] over the entire prediction range that preserves sequence-level properties, analogous to how Wasserstein loss is used in [54] for the TPP setting.

A key difference of DualTPP compared to all previous work in both the TPP and time-series literature is that, all existing methods focus on training, and during inference continue to deploy the same one-step event generation. Our key idea is to use a second model to output properties of the aggregated set of predicted events. We then solve an efficient joint optimization problem over the predicted sequence to achieve consensus between the predicted aggregate properties and one-step generated events. This relates our approach to early work on multi-view learning in the traditional machine learning literature that we discuss next.

Multi-view Learning Models. Inference in structured prediction tasks with aggregate potentials over a large number of predicted variables was studied in tasks like image segmentation, [25, 38, 43] and information extraction [18]. In several NLP tasks too, enforcing constraints during inference via efficient optimization formulations has been found to be effective in [10, 14, 36]. In this paper we demonstrate, for the first time, the use of these ideas to TPPs, which due to their continuous nature, pose very different challenges than classical multi-view models on discrete labels.

3 MODEL FORMULATION

In this section, we formulate DualTPP, our two-component modeling framework for marked temporal point processes (MTPPs). We begin with an overview of MTPPs and then provide a detailed description of our proposed DualTPP.

3.1 Background on MTPP

An MTPP [13, 51, 60] is a stochastic process, which is realized using a series of discrete events arriving in continuous time. Given a sequence of events $\{e_1 = (m_1, t_1), e_2 = (m_2, t_2), \dots\}$ where $m_i \in [K]$ [1] indicate the discrete mark and $t_i \in \mathbb{R}^+$ indicate the arrival time of the $i-$th event, an MTPP is characterized by $H_t = \{e_i = (m_i, t_i) | t_i < t\}$ which gathers all events that arrived until time t. Equivalently, it can also be described using a counting process $N(t)$ which counts the number of events arrived until time t, *i.e.*, $N(t) = |H_t|$. The dynamics of $N(t)$ is characterized using an intensity function $\lambda^*(t)$, which specifies the likelihood of the next event, conditioned on the history of events H_t[2]. The intensity function $\lambda^*(t)$ computes the infinitesimal probability that an event will happen in the time window $(t, t + dt]$ conditioned on the history H_t as follows:

$$\mathbb{P}(dN(t) = N(t + dt) - N(t) = 1 \,|\, H_t) = \lambda^*(t)dt, \tag{1}$$

The intensity function is used to compute the expected time of the next event as:

$$\mathbb{E}[t_i \,|\, H_{t_i}] = \int_{t_{i-1}}^{\infty} t \cdot \lambda^*(t)dt \tag{2}$$

The marks are generated using some probability distribution q_m conditioned on the history of the events, *i.e.*,

$$\mathbb{P}(m_i = k | H_{t_i}) = q^*_m(k) \tag{3}$$

Given the history of events H_T observed during the time interval $(0, T]$, one typically learns the intensity function $\lambda^*(t)$ and the mark distribution q^*_m by maximizing the following likelihood function:

$$\mathcal{L}(H_T \,|\, \lambda^*, q^*_m) = \sum_{(m_i, t_i) \in H_T} \left(\log q^*_m(m_i) + \log \lambda^*(t_i)\right) + \int_0^T \lambda^*(\tau)d\tau$$

Once the intensity function $\lambda^*(t)$ and the mark distribution q^*_m are estimated, they are used to forecast events by means of thinning [32] or inverse sampling [45] mechanisms. Such mechanisms often suffer from poor time complexity. Moreover, such recursive sampling methods build up prediction error led by any model misspecification. In the following, we aim to design a temporal point process model that is able to overcome that limitation.

3.2 Design of DualTPP

We now set about to design our proposed model DualTPP. At the very outset, DualTPP has two components to model the underlying MTPP— the event model for capturing the dynamics of individual events, and the count model that provides an alternative count perspective over a set of events in the long-term future. Here we describe the model structure and training and in Section 4 describe how we combine outputs from the two models during inference.

Event model. Our event model is a generative process which draws the next event (m, t), given the history of events H_t. In several applications [24, 53] the arrival times as well as the marks of the subsequent events depend on the history of previous events. Therefore, we capture such inter-event dependencies by realizing our event model using a conditional density function $p_\theta(\bullet | H_t)$. Following several existing MTPP models [13, 24, 31, 53] we model $p_\theta(\bullet | H_t)$ by means of a recurrent neural network with parameter θ, which embeds the history of events in compact vectors $\boldsymbol{h}_\bullet$. It has three layers: (i) input layer, (ii) hidden layer and (iii) output layer. In the following, we illustrate them in detail.

— *Input layer.* Upon arrival of the i-th event $e = (m_i, t_i)$, the input layer transforms m_i into an embedding vector and compute the inter-arrival gap, which are used by next layers afterwards. More specifically, it computes

$$\overline{\boldsymbol{m}}_i = m_i \boldsymbol{w}_m + \boldsymbol{b}_m, \tag{4}$$

$$\delta_i = t_i - t_{i-1}, \tag{5}$$

— *Hidden layer.* This layer embeds the history of events in the sequence of hidden state vectors $(\boldsymbol{h}_\bullet)$ using a recurrent neural network. More specifically, it takes three signals as input: (i) the embeddings $\overline{\boldsymbol{m}}_i$ and (ii) the inter-arrival time duration δ_i, which is computed in the previous layer, as well as (iii) the hour of the event as an additional feature f_i; and then updates the hidden state $\boldsymbol{h}_\bullet$ using a gated recurrent unit as follows:

$$\boldsymbol{h}_i = \text{GRU}_{\boldsymbol{w}_h}(\boldsymbol{h}_{i-1}; \overline{\boldsymbol{m}}_i, \delta_i, f_i). \tag{6}$$

Note that $\boldsymbol{h}_i$ summarizes the history of first i events.

— *Output layer.* Finally, the output layer computes the distribution of the mark m_{i+1} and timing t_{i+1} of the next event as follows: We parameterize the distribution over marks as a softmax over the hidden states:

$$\mathbb{P}(m_{i+1} = c) = \frac{\exp(\boldsymbol{w}_{y,c}^\top \boldsymbol{h}_i + b_{y,c})}{\sum_{j=1}^K \exp(\boldsymbol{w}_{y,j}^\top \boldsymbol{h}_i + b_{y,j})} \tag{7}$$

Similar to [42], we use a Gaussian distribution to model the gap δ_i to the next event:

$$\delta_i \sim \mathcal{N}(\mu(\boldsymbol{h}_i), \sigma(\boldsymbol{h}_i)), \quad t_{i+1} = t_i + \delta_i. \tag{8}$$

Here $\mu(\boldsymbol{h}_i)$ the mean gap and its standard deviation $\sigma(\boldsymbol{h}_i)$ are computed from the hidden state as:

$$\mu(\boldsymbol{h}_i) = \text{softplus}(\boldsymbol{w}_\mu^\top \boldsymbol{h}_i + b_\mu) \tag{9}$$

$$\sigma(\boldsymbol{h}_i) = \text{softplus}(\boldsymbol{w}_\sigma^\top \boldsymbol{h}_i + b_\sigma). \tag{10}$$

The Gaussian density provided more accurate long-term forecasts than existing intensity-based approaches. Here, $\theta = \{\boldsymbol{w}_\bullet, \boldsymbol{b}_\bullet\}$ are the set of trainable parameters.

[1] In the current work, we consider discrete marks which can take K labels, however, our method can easily be extended to continuous marks.

[2] $*$ indicates the dependence on history

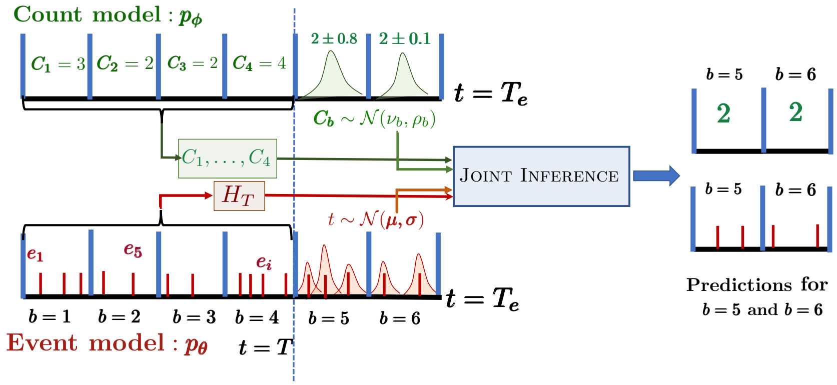

Figure 1: Overview of the inference task in DualTPP. In this example, the time horizon is split into six bins. The eleven events in the first four bins comprise the known history H_T. The event-model p_θ conditioned on H_T predicts five events until T_e spanning two bins. The count model at the top predicts two Gaussians of mean 2 each. These are combined by DualTPP's joint inference algorithm (Algorithm 1) to get the revised event predictions shown on the right. Marks are omitted for clarity.

The above event model via its auto-regressive structure is effective in capturing the arrival process of events at a microscopic level. Indeed it is sufficiently capable of accurately predicting events in the short-term future. However, its long-term predictions suffers due to cascading errors when auto-regressing on predicted events. The count model is designed to contain this drift.

Count model. Here we aim to capture the number of events arriving in a sequence of time intervals. We partition time into equal time-intervals — called as bins— of size Δ which is a hyper-parameter of our model (Figure 1 shows an example). Given a history of events H_T, we develop a simple distribution p_ϕ which generates the total count of events for subsequent n bins. Let I_s be the time-interval $[T + (s-1)\Delta, T + s\Delta)$, and C_s denote the number of events occurring within it. We factorize the distribution p_ϕ over the n future bins independently over each of the bins while conditioning on the known history H_T, and properties of the predicted bin.

$$p_\phi(C_n, C_{n-1}, \cdots, C_1 \,|\, H_T, I_n, \ldots, I_1) = \prod_{j=1}^{n} p_\phi(C_j \,|\, H_T, I_j) \quad (11)$$

This conditionally independent model provided better accuracy than an auto-regressive model that would require conditioning on future unknown counts. Similar observations have been made for time-series models in [9, 50].

Each $p_\phi(C_j \,|\, H_T, I_j)$ is modeled as a Gaussian distribution with mean $\nu_{j,\phi}$ and variance $\rho_{j,\phi}$. A feed-forward network with parameters ϕ, learns these parameters as a function of features extracted from the history H_T and current interval I_j as follows: From a time interval we extract time-features such as the hour-of-the-day in the mid-point of the bin. Then from H_T we extract the counts of events in the most recent n^- bins before T and time features from their corresponding bins.

Learning the parameters θ and ϕ. Given a stream of observed events $\{e_i^-\}$ during the time window $(0, T]$, we learn the event model θ by maximizing the following likelihood function:

$$\underset{\theta}{\text{maximize}} \sum_{e_i \in H_T} \log p_\theta(e_i^- \,|\, H_{t_i}). \quad (12)$$

In order to train the count model, we first group the events into different bins of same width Δ. Next, we sample them in different batches of $n^- + n$ bins and then learn ϕ in the following manner:

$$\underset{\phi}{\text{maximize}} \underset{H_s \sim p_{\text{Data}}}{\mathbb{E}} \sum_{j=1}^{n} \log p_\phi(C_{s+j}^- | H_s, I_{s+j}^-) \quad (13)$$

In the above H_s denotes a history of events between time $(s - n^-)\Delta$ and s, and C_{s+j}^- denotes the observed counts of events in bin I_{s+j}.

4 INFERENCE

In this section, we formulate our inference procedure over the trained models (p_θ, p_ϕ) for forecasting all events (marks and time) within a user-provided future time T_e given the history H_T of events before $T < T_e$.

Inference with Event-only Model. First, we review existing method of solving this inference task using the event-only method p_θ. Note p_θ is an auto-regressive model that provides a distribution on the next event e_{i+1} given known historical events H_T before T and predicted prior events $\hat{e}_1, \ldots \hat{e}_i$, that is $p_\theta(e_{i+1} \,|\, H_T, \hat{e}_1, \ldots \hat{e}_i)$. Let $\boldsymbol{h}_i$ denote the RNN state after input of events in the history H_T and predicted events $\hat{e}_1, \ldots \hat{e}_i$. Based on the state, we predict a distribution of the next gap via Eq. 8 and next mark using Eq. 7. The predicted times and marks of the next event are just the modes of the respective distribution as: $\hat{e}_{i+1} = (\hat{m}_{i+1} = \text{argmax}_m P(m_{i+1} = m), \hat{t}_{i+1} = \hat{t}_i + \mu(\boldsymbol{h}_i))$. The predicted event is input to the RNN to get a new state and we repeat the process until we predict an event with time $> T_e$.

As mentioned earlier, the events predicted by such forward-sampling method on p_θ alone is subject to drift particularly when T_e is far from T. We next go over how DualTPP captures the drift by generating an event sequence that jointly maximizes the probability of the event and count model.

Joint Inference Objective of DualTPP. The event model gives a distribution of the next event given all previous events whereas the count model p_ϕ imposes a distribution over the number of events that fall within the regular Δ-sized bins between time T and T_e. For simplicity of exposition we assume that T_e aligns with a bin boundary, i.e., $T_\text{e} = T + n_\text{e}\Delta$ for a positive n_e. During inference we wish to determine the sequence of events that maximizes the product of their joint probability as follows:

$$\max_{r, e_1, \ldots, e_r} \left[\sum_{i=1}^{r} \log p_\theta(e_i \mid H_T, e_1, \ldots, e_{i-1}) + \sum_{b=1}^{n_\text{e}} \log p_\phi(C_b \mid H_T, I_b) \right] \tag{14}$$

$$\text{such that,} \;\; t_r < T_\text{e}, \quad \left|\{e_i \mid t_i \in I_b\}\right| = C_b \quad \forall b \in [n_\text{e}] \tag{15}$$

Unlike the number of bins, the number of events r is unknown and part of the optimization process. The constraints ensure that the last event ends before T_e and there is consensus between the count and event model. Solving the above optimization problem exactly over all possible event sequences completing before T_e is intractable for several confounding reasons — the event model expresses the dependence of an event over *all* previous events, and that too via arbitrary non-linear functions. Also, it is not obvious how to enforce the integral constraint on the number of events in a bin as expressed in Eq. 15.

Tractable Decomposition of the Joint Objective. We propose two simplifications that allow us to decompose the above intractable objective into a sequence of optimization problems that are optimally solvable. First, we decompose the objective into n_e stages. In the b-th stage we infer the set of events whose times fall within the b-th bin I_b assuming we already predicted the set of all events before that bin. Call these $\boldsymbol{E}_b = \hat{e}_1, \ldots, \hat{e}_{r_b}$ where $r_b = |\boldsymbol{E}_b|$ denotes the number of predicted events before start of b-th bin, i.e, left of I_b. Second, we fix the RNN state $\boldsymbol{h}_i$ for all potential events in I_b to their unconstrained values as follows: Starting with the RNN state $\boldsymbol{h}_{r_b}$, we perform forward sampling as in the event-only baseline until we sample an upper limit C_max of events likely to be in I_b. We will discuss how to choose C_max later. Once the RNN state $\boldsymbol{h}_i$ is fixed, the distribution of the gap between the i-th and $(i+1)$th event is modeled as a Gaussian $\mathcal{N}(\mu(\boldsymbol{h}_i), \sigma(\boldsymbol{h}_i))$ and the predicted mark $\hat{m}_{i+1}$ is also fixed. We can then rewrite the above inference problem for the events on b-th bin as a double optimization problem as follows:

$$\max_{c \in [C_\text{max}]} \left[\max_{g_1, \ldots g_c} \sum_{i=1}^{C_\text{max}} \log \mathcal{N}(g_i; \mu(\boldsymbol{h}_{r_b+i}), \sigma(\boldsymbol{h}_{r_b+i})) + \log \mathcal{N}(c; \nu_b, \rho_b) \right]$$

$$\text{such that,} \;\; g_i \geq 0, \;\; \sum_{i=1}^{c} g_i \leq \Delta, \;\; \sum_{i=1}^{c+1} g_i > \Delta, \quad \hat{t}_{r_b} + g_1 \in I_b \tag{16}$$

In the above the constraints in the inner optimization just ensure that exactly c events are inside bin I_b. All constraints are linear in g_i unlike in Eq. 15. The optimization problem in Eq. 17 is amenable to efficient inference: For a fixed c, the inner maximization is over real-valued gap variables g_i with a concave quadratic objective and linear constraints. Thus, for a given c, the optimal gap values can be efficiently solved using any off-the-shelf QP solver. The outer maximization is over integral values of c but we use a simple binary search between the range 0 and C_max to solve the above in $\log(C_\text{max})$ time.

Let $c^*, g_1^*, \ldots, g_{c^*}^*$ denote the optimal solution. Using these we expand the predicted event sequence from r_b by c^* more events as $(\hat{m}_{r_b+1}, \hat{t}_{r_b} + g_1^*), \ldots (\hat{m}_{r_b+c^*}, \hat{t}_{r_b} + \sum_{i=1}^{c^*} g_i^*)$. We append these to $\boldsymbol{E}_b$ to get the new history of predicted events $\boldsymbol{E}_{b+1}$ conditioned on which we predict events for the $(b+1)$-th bin. The final set of predicted events are obtained after n_e stages in E_{n_e+1}

Algorithm 1: Inference of events in the $[T, T_\text{e})$

```
Input: Trained event model and trained count model $p_\theta$, $p_\phi$, event history $H_T$, end time $T_\text{e} = T + n_e\Delta$.
Output: Forecast events $\{\hat{e} \mid \hat{t} \in [T, T_\text{e})\}$
$\boldsymbol{E} \leftarrow \emptyset$ /* Predicted events so far */
for $b$ in $[n_e]$ do
    $\nu_b, \rho_b \leftarrow$ Count distribution from $p_\phi(.|H_T, I_b)$
    $\boldsymbol{h}, C_\text{max} \leftarrow$ RNNStates($p_\theta, H_T, \boldsymbol{E}, \nu_b, b$) /*set $\boldsymbol{h}_\bullet$ */
    /* Solve the optimization problem in Eq. 16*/
    $\boldsymbol{E} \leftarrow \boldsymbol{E}$ + OptimizeInBin($\boldsymbol{h}, \nu_b, \rho_b, C_\text{max}, I_b$)
end for
Return $\boldsymbol{E}$
```

Dataset	Train Size	$\mathbb{E}[t]$	$\sigma[t]$	Avg. #Events in $[T, T_e)$	Bin Size (Δ)
Elections	51859	7.0	5.8	203	7 mins.
Taxi	399433	8.0	25.8	1254	1 hour
Traffic-911	115463	778	1517	281	1 day
EMS-911	182845	492	601	275	12 hours

Table 1: Statistics of the datasets used in our experiments. Train Size denotes the number of events in the training set. $\mathbb{E}[t]$ and $\sigma[t]$ denote the mean and variance of the inter-event arrival time.

Choosing C_max . Let C_E denote the count of events in bin I_b when each gap g_i is set to its unconstrained optimum value of $\mu(.)$. We obtain this value as we perform forward sampling from RNN state $\boldsymbol{h}_{r_b}$. The optimum value of c from the count-only model is ν_b. Due to the unimodal nature of the count model p_ϕ, one can show that the optimal c^* lies between ν_b and C_E. Thus, we set the value $C_\text{max} = \max(\nu_b + 1, C_E)$. Also, to protect against degenerate event-models that do not advance time of events, we upper bound C_max to be $\nu_b + \rho_b$ since the count model is significantly more accurate, and the optimum c^* is close to its mode ν_b.

Overall Algorithm. Algorithm 1 summarizes DualTPP's inference method. An example is shown in Figure 1. To predict the events in the b-th bin, we first invoke the count model p_ϕ and get mean count ν_b, variance ρ_b. We then forward step through the event RNN p_θ after conditioning on previous events $H_T, \boldsymbol{E}$. We then continue forward sampling until bin end or $\mu_b + 1$, and return the visited RNN states, and number of steps C_max. Now, we invoke the optimization problem in Eq. 16 to get the predicted events in the bth bin which we then append to $\boldsymbol{E}$.

5 EXPERIMENTS

In this section, we evaluate our method against five state-of-the-art existing methods, on four real datasets.

5.1 Datasets

We use four real world datasets that contain diverse characteristics in terms of their application domains and temporal statistics. We also summarize the details of these datasets in Table 1.

Election. [8] This dataset contains tweets related to presidential election results in the United-States, collected from 7th April to 13th April, 2016. Here, given a tweet e, the mark m indicates the user who posted it and the time t indicates the time of the post.

Taxi. [2] This contains the pickup, drop-off timestamps and pickup, drop-off locations of taxis in New York city from 1st Jan 2019 to 28th Feb 2019. The dataset is categorized by zones. In our experiments we only consider pick up zone with zone id 237. We consider each travel as an event $e = (m, t)$, with pick up time denoted by t and drop-off zone as the marker m.

Traffic-911. [1] This dataset consists of emergency calls related to road traffic in the US, in which each event contains timestamp of the call and location of the caller, which we treat as a marker.

EMS-911. [1] This dataset consists of emergency calls related to medical services in the US, in which each event contains timestamps of the call, and location of the caller which we treat as the marker.

For all datasets, we rank markers based on their occurrence frequency and keep the top 10 markers. Rest of the markers are merged into a single mark. Hence, we have 11 markers in each dataset.

5.2 Methods Compared

We compare DualTPP against five other methods spanning a varied set of loss functions and architectures: The first two (RMTPP, THP) are trained to predict the next event via intensity functions using maximum likelihood (Sec 3.1). The next two (WGAN and Seq2Seq) are trained to predict a number of future events using a sequence-level Wasserstein loss and are better suited for long-term forecasting. The last uses a two-level hierarchy to capture long-term dynamics. We present more details below:

RMTPP. RMTPP [13] is one of the earliest neural point process model that uses a three layer recurrent neural network to model the intensity function and mark distribution of an MTPP.

Transformer Hawkes Process (THP). THP [60] is more recent and uses Transformers [48] instead of RNNs to model the intensity function of the next event. The THP leverages the positional encoding in the transformer model to encode the timestamp.

WGAN. Wasserstein TPPs [51] train a generative adversarial network to generate an event sequence. A Homogeneous Poisson process provides the input noise to the generator of future events, which by a Wasserstein discriminator loss is trained to resemble real events. Since our predicted events are conditioned on the input history, we initialize the generator by encoding known history of events using an RNN.

Seq2Seq. is a conditional generative model [54], in which, an encoder-decoder model for sequence-to-sequence learning is trained by maximizing the likelihood of the output sequence. Also added is a Wasserstein loss computed via a CNN-based discriminator.

Hierarchical Generation. We designed this method to explore if hierarchical models [6, 46, 47], could be just as effective as our count-model to capture macroscopic dynamics. We create a two-level hierarchy where the top-level events are compound events of τ consecutive events. We train a second event-only model $p_\psi(\bullet|H_t)$ over the compound events to replace the count-model. Using trained models (p_θ, p_ψ) we perform inference similar to Eq. 16. However, since compound model p_ψ imposes a distribution over every τ-th event, we solve the following optimization problem for every j-th compound event:

$$\max_{g_1 \ldots g_\tau, g_i \in \mathbb{R}^+} \Bigg[\sum_{i=1}^{\tau} \log \mathcal{N}(g_i; \mu(\boldsymbol{h}_{j\tau+i}), \sigma(\boldsymbol{h}_{j\tau+i})) \; + \\ \log \mathcal{N}(\sum_{i=1}^{\tau} g_i; \mu(\boldsymbol{h}_j^c), \sigma(\boldsymbol{h}_j^c)) \Bigg] \tag{17}$$

Similar to Eq. 16, the maximization is over positive real-valued gap variables g_i and with a concave quadratic objective. Here, the number of stages is not fixed to n_e, but we stop when the last predicted time-stamp is greater than T_e.

5.3 Evaluation protocol

We create train-validation-test splits for each dataset by selecting the first 60% time-ordered events as training set, next 20% as validation and rest 20% as test set. We chose the value of the bin-size Δ so that each bin has at least five events on average while aligning with standard time periodicity as shown in Table 1. A test 'instance' starts at a random time T_s within the test time, includes all events up to $T = T_s + 20\Delta$ as the known history H_T, and treat the interval between T and $T_\mathrm{e} = T + 3\Delta$ as the forecast horizon. The average number of events in the forecast horizon ranges between 200 and 1250 across the four datasets (shown in Table 1). For training the count model p_ϕ we created instances using the same scheme. The event model p_θ just trains for the next event using random event sub sequences of length 80.

Architectural Details. For event model, we use a single layer recurrent network with GRU cell of 32 units. We fixed the batch size to 32 and used Adam optimizer with learning rate 1e−3. The size of the embedding vector of a mark is set to 8. We train the event model for 10 epochs. We checkpoint the model at the end of each epoch and select the model that gives least validation error. The Count model is a feed-forward network with three hidden layers of 32 units, all with ReLU activation. The input layer of count model has 40 units, corresponding to counts of 20 input bins and hour-of-day at the mid-point of each bin. The output layer predicts the Gaussian parameters ν_j, ρ_j for each future bin j.

Evaluation Metrics. We use three metrics to measure performance. First, we measure the Wasserstein distance between predicted and actual event sequences to assess the microscopic dynamics between events. Given true event times H in an interval $[T_\mathrm{st}, T_\mathrm{e})$ and the corresponding predicted events $\widehat{H}$, assuming without loss of generality, $|H| < |\widehat{H}|$ we compute Wasserstein distance [51] between the two sequence of events as

$$\mathrm{WassDist}(H, \widehat{H}) = \sum_{i=1}^{|H|} |t_i^+ - \hat{t}_i| + \sum_{i=|H|+1}^{\widehat{H}} (T_\mathrm{e} - \hat{t}_i) \tag{18}$$

We randomly sample several such intervals $[T_\mathrm{st}, T_\mathrm{e})$ and report the average of WassDist of all intervals. Second, to assess the macroscopic modeling component of each method we define a CountMAE

Dataset	Model	Wass. dist	BLEU Score	Count MAE
Elections	RMTPP [13]	1231	0.684	26.7
	TransMTPP [60]	1458	0.579	31.8
	WGAN [51]	442	-	10.0
	Seq2Seq [54]	739	-	15.9
	Hierarchical	415	0.880	8.5
	DualTPP	**267**	**0.882**	**5.0**
Taxi	RMTPP [13]	9826	0.089	288
	WGAN [51]	4060	-	128
	Seq2Seq [54]	5105	-	161
	Hierarchical	8838	0.088	206
	DualTPP	**1923**	**0.090**	**39**
Traffic-911	RMTPP [13]	2406	**0.248**	41.7
	TransMTPP [60]	6096	0.081	110.0
	WGAN [54]	3892	-	69.0
	Seq2Seq [54]	4520	-	83.0
	Hierarchical	1853	0.211	33.1
	DualTPP	**1700**	0.221	**29.1**
EMS-911	RMTPP [13]	2674	0.162	20.9
	TransMTPP [60]	5792	0.070	50.0
	WGAN [54]	2432	-	19.3
	Seq2Seq [54]	9856	-	90.3
	Hierarchical	1639	**0.163**	11.8
	DualTPP	**1419**	**0.163**	**10.1**

Table 2: Comparative analysis of our method against all baselines across all datasets in terms of WassDist, BLEUScore, and CountMAE. It shows the DualTPP consistently outperforms all the baselines.

that aims to measure the relative error in predicted count in randomly sampled time interval:

$$\text{CountMAE} = \frac{1}{M} \sum_{i \in M} \frac{\left|\{e \mid t \in \mathcal{I}^{(i)}\}\right| - \left|\{\hat{e} \mid \hat{t} \in \mathcal{I}^{(i)}\}\right|}{\left|\{e \mid t \in \mathcal{I}^{(i)}\}\right|}, \qquad (19)$$

where $\mathcal{I}^{(i)}$ is randomly sampled in test-horizon and we sample M such intervals. Finally, for evaluating accuracy of the predicted discrete mark sequence, we compare our generated mark sequence with the true mark sequence (which could be of a different length) using the BLEU score popular in the NLP community [35].

5.4 Results

In this section, we first compare DualTPP against the five methods of Sec 5.2, and then analyze how accurately it can forecast events in a distant time. Next, we provide a thorough ablation study on DualTPP.

Comparative analysis. Here we compare DualTPP against five state-of-the-art methods. WGAN and Seq2Seq papers do not model marks, hence their BLEU scores are omitted. Table 2 summarizes the results, which reveals the following observations.

(1) DualTPP achieves significant accuracy gains beyond all five methods, in terms of all three metrics *i.e.*, CountMAE, WassDist and BLEUScore. For some datasets, e.g. Taxi the gains by our method are particularly striking — our error in counts is 39, and the closest alternative has almost three times higher error! Even for microscopic inter-event dynamics as measured by the Wasserstein distance we achieved a factor of two reduction. Figure 2 shows three anecdotal sequences comparing counts of events in different time-intervals of DualTPP (Blue) against actual (Black) and the RMTPP baseline (red). Notice how RMTPP drifts away whereas DualTPP tracks the actual.

(2) The Hierarchical variant of our method is the second best performer, but its performance is substantially poor compared to DualTPP, establishing that the alternative count-based perspective is as important as viewing events at different scales for accurate long-term perspective. More specifically, the Hierarchical variant considers aggregating a fixed number of events, which makes it oblivious to the prediction for heterogeneous counts in an arbitrary time interval. DualTPP aims to overcome these limitations by means of both the event and the count model, which characterize both short term and long term characteristics of an event sequence.

(3) Both RMTPP and TransMTPP are much worse than DualTPP. WGAN and Seq2Seq provide unreliable performance and show large variance across datasets.

Performance on long term forecasting. Next we analyze the performance difference further by looking at errors in different forecast time-intervals in the future in Figure 3. Here, on the X-axis each tick gives the average number of events since the known history T in gold and on the Y-axis we show the Wasserstein distance for events predicted between time of two consecutive ticks. We observe the expected pattern that events further into the future have larger error than closer events for all methods. However, DualTPP shows a modest deterioration, whereas, both RMTPP and Hierarchical show a significant deterioration. For example in the leftmost plots on the Election dataset, the Wasserstein distance for the first 68 events increases from 500 to almost 800 for RMTPP, but only from 200 to 270 for DualTPP.

We measure sensitivity of our results to bin-sizes by varying the bin-size Δ, and correspondingly the query time-range $[T, T_e = 3\Delta]$. Figure 4 shows the Wasserstein distance between true and predicted events across different bin sizes on the Taxi dataset. We find that DualTPP continues to perform better than competing methods across all bin sizes.

Ablation Study. We perform ablation study using variants of DualTPP to analyze which elements of our design contributed most to our observed gains. We evaluate these variants using the Wasserstein distance metric and summarize in Table 3

First we see the performance obtained by our Event-only model. We observe that the event-only model performs much worse than DualTPP, establishing the importance of the count model to capture its drift. We next compare with the Count-only model, where we first predict counts of event for the b-th bin (ν_b), and then randomly generate ν_b events in the b-th bin. In this case, marks are ignored. We observe that the count-only model is worse than DualTPP but it performs much better than the Event-only method.

Next, we analyze other finer characteristics of our model. In DualTPP, the event model uses the Gaussian density whereas most existing TPP models (e.g. RMTPP and THP discussed earlier) use an intensity function. We create a version of DualTPP called DualTPP-with-intensity where we model the distribution $p_\theta(\bullet|H_t)$ using

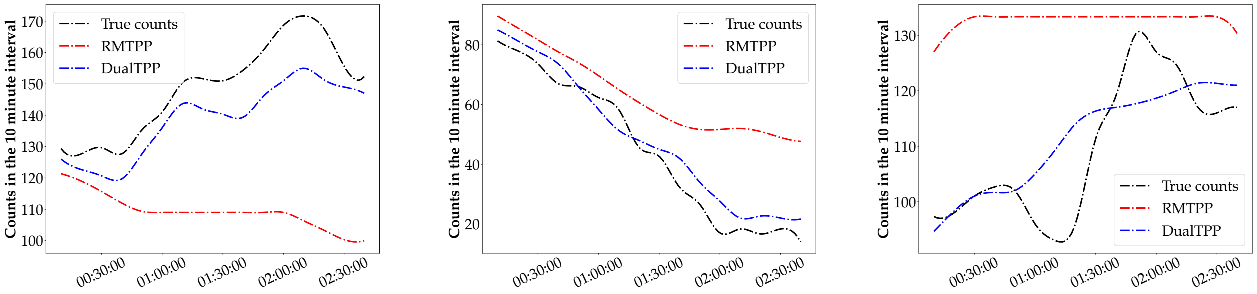

Figure 2: Anecdotal examples of variation of counts against time, collected from Taxi datasets. They show that DualTPP can mimic the high level trajectory more accurately than RMTPP. In the second example, we observe that RMTPP and DualTPP show similar nowcasting performance, whereas DualTPP shows more accurate forecasting performance than RMTPP.

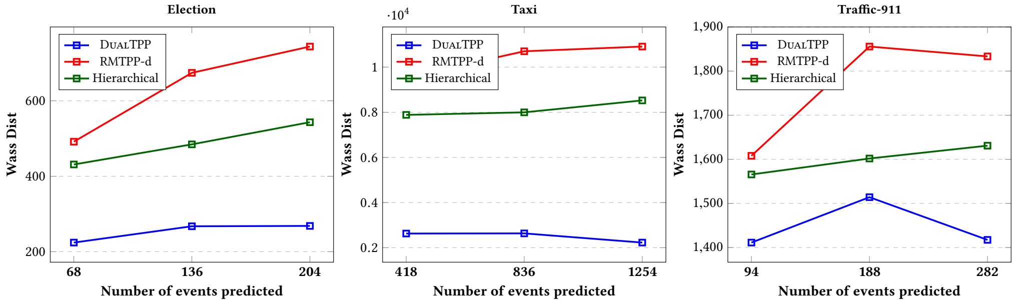

Figure 3: Long term forecasting of DualTPP, RMTPP-d and Hierarchical across three datasets in terms of WassDist. RMTPP-d is just RMTPP with Gaussian density instead of intensity. X-axis denotes the average number of events in the gold since the known history T and Y-axis denotes the Wasserstein distance between gold and predicted events.

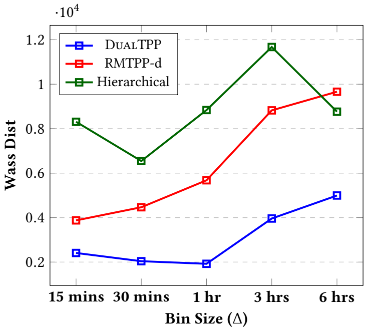

Figure 4: Long term forecasting comparison on Taxi dataset: X-axis denotes the bin size used to train the count model p_ϕ and Y-axis denotes the Wasserstein Distance between true and predicted events.

the conditional intensity of RMTPP. Comparing the two methods we observe that the choice of Gaussian density also contributes significantly to the gains observed in DualTPP.

In the DualTPP-without-count-variance model, we predict the events in the b-th bin by solving the inner optimization problem in Eq. 16 only for the mean ν_b, thereby treating p_ϕ as a point distribution. We observe a performance drop highlighting the benefit of modeling the uncertainty of the count distribution.

Dataset	Model	Wass dist
Election	DualTPP	267
	Event-only	633
	Count-only	310
	DualTPP-with-intensity	271
	DualTPP-without-count-variance	272
Taxi	DualTPP	1923
	Event-only	5679
	Count-only	1923
	DualTPP-with-intensity	1790
	DualTPP-without-count-variance	1916
Traffic-911	DualTPP	1700
	Event-only	1767
	Count-only	2098
	DualTPP-with-intensity	2211
	DualTPP-without-count-variance	1746
EMS-911	DualTPP	1419
	Event-only	1485
	Count-only	2186
	DualTPP-with-intensity	2318
	DualTPP-without-count-variance	1423

Table 3: Ablation Study: Comparison of DualTPP and its variants in terms of Wasserstein Distance between true and predicted events.

6 CONCLUSIONS

In this paper, we propose DualTPP, a novel MTPP model specifically designed for long-term forecasting of events. It consists of two components— Event-model which captures dynamics of the underlying MTPP in a microscopic scale and Count-model which captures the macrocopic dynamics. Such a model demands a fresh approach for inferring future events. We design a novel inference method that solves a sequence of efficient constrained quadratic programs to achieve consensus across the two models. Our experiments show that DualTPP achieves substantial accuracy gains beyond five competing methods in terms of all three metrics: Wasserstein distance that measures microscopic inter-event dynamics, Count-MAE that measures macroscopic count error, and BLEU score that evaluates the sequence of generated marks. Future work in the area could include capturing other richer aggregate statistics of event sequences. Another interesting area is providing inference procedures for answering aggregate queries directly.

REFERENCES

[1] 911 dataset. URL https://www.kaggle.com/mchirico/montcoalert.

[2] Taxi dataset. URL https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page.

[3] I. Apostolopoulou, S. Linderman, K. Miller, and A. Dubrawski. Mutually regressive point processes. In *NeurIPS*, pages 5115–5126, 2019.

[4] S. Ben Taieb and A. Atiya. A bias and variance analysis for multistep-ahead time series forecasting. *IEEE transactions on neural networks and learning systems*, 27(3), 2015.

[5] J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West. The markov modulated poisson process and markov poisson cascade with applications to web traffic modeling. *Bayesian Statistics*, 2003.

[6] A. Borovykh, S. Bohte, and C. W. Oosterlee. Conditional time series forecasting with convolutional neural networks, 2017.

[7] R. Cai, X. Bai, Z. Wang, Y. Shi, P. Sondhi, and H. Wang. Modeling sequential online interactive behaviors with temporal point process. In *CIKM*, pages 873–882, 2018.

[8] A. De, S. Bhattacharya, and N. Ganguly. Demarcating endogenous and exogenous opinion diffusion process on social networks. In *WWW*, pages 549–558, 2018.

[9] P. Deshpande and S. Sarawagi. Streaming adaptation of deep forecasting models using adaptive recurrent units. In *ACM SIGKDD*, 2019.

[10] D. Deutsch, S. Upadhyay, and D. Roth. A general-purpose algorithm for constrained sequential inference. In M. Bansal and A. Villavicencio, editors, *Proceedings of the 23rd Conference on Computational Natural Language Learning, CoNLL 2019, Hong Kong, China, November 3-4, 2019*, pages 482–492. Association for Computational Linguistics, 2019.

[11] N. Du, L. Song, M. Yuan, and A. J. Smola. Learning networks of heterogeneous influence. In *NeurIPS*, pages 2780–2788. Curran Associates, Inc., 2012.

[12] N. Du, L. Song, H. Woo, and H. Zha. Uncover topic-sensitive information diffusion networks. In *Artificial Intelligence and Statistics*, pages 229–237, 2013.

[13] 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*, pages 1555–1564, 2016.

[14] E. Fersini, E. Messina, G. Felici, and D. Roth. Soft-constrained inference for named entity recognition. *Inf. Process. Manag.*, 50(5):807–819, 2014.

[15] V. Filimonov and D. Sornette. Apparent criticality and calibration issues in the hawkes self-excited point process model: application to high-frequency financial data. *Quantitative Finance*, 15(8):1293–1314, 2015.

[16] V. Flunkert, D. Salinas, and J. Gasthaus. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *CoRR*, abs/1704.04110, 2017.

[17] K. Giesecke and G. Schwenkler. Filtered likelihood for point processes. *Journal of Econometrics*, 204(1):33–53, 2018.

[18] R. Gupta, S. Sarawagi, and A. A. Diwan. Collective inference for extraction mrfs coupled with symmetric clique potentials. *JMLR*, 11, Nov. 2010.

[19] B. Hambly and A. Søjmark. An spde model for systemic risk with endogenous contagion. *Finance and Stochastics*, 23(3):535–594, 2019.

[20] A. G. Hawkes. Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 33(3):438–443, 1971.

[21] A. G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

[22] A. G. Hawkes. Hawkes jump-diffusions and finance: a brief history and review. *The European Journal of Finance*, pages 1–15, 2020.

[23] V. Isham and M. Westcott. A self-correcting point process. *Stochastic processes and their applications*, 8(3):335–347, 1979.

[24] H. Jing and A. J. Smola. Neural survival recommender. In *WSDM*, pages 515–524, 2017.

[25] P. Kohli, L. Ladicky, and P. H. S. Torr. Robust higher order potentials for enforcing label consistency. *International Journal of Computer Vision*, 82(3):302–324, 2009.

[26] Q. Kong, M.-A. Rizoiu, and L. Xie. Modeling information cascades with self-exciting processes via generalized epidemic models. In *WSDM*, pages 286–294, 2020.

[27] S. Lamprier. A recurrent neural cascade-based model for continuous-time diffusion. In *ICML*, volume 97, pages 3632–3641. PMLR, 2019.

[28] V. LE GUEN and N. THOME. Shape and time distortion loss for training deep time series forecasting models. In *Advances in Neural Information Processing Systems 32*. 2019.

[29] G. Loaiza-Ganem, S. Perkins, K. Schroeder, M. Churchland, and J. P. Cunningham. Deep random splines for point process intensity estimation of neural population data. In *NeurIPS*, pages 13346–13356, 2019.

[30] M. Maciak, O. Okhrin, and M. Pešta. Infinitely stochastic micro forecasting. *arXiv*, pages arXiv–1908, 2019.

[31] H. Mei and J. M. Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. In *NeurIPS*, pages 6754–6764, 2017.

[32] Y. Ogata. Space-time point-process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics*, 50(2):379–402, 1998.

[33] M. Okawa, T. Iwata, T. Kurashima, Y. Tanaka, H. Toda, and N. Ueda. Deep mixture point processes: Spatio-temporal event prediction with rich contextual information. In *KDD*, pages 373–383, 2019.

[34] T. Omi, K. Aihara, et al. Fully neural network based model for general temporal point processes. In *NeurIPS*, pages 2122–2132, 2019.

[35] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, July 2002.

[36] V. Punyakanok, D. Roth, W. Yih, and D. Zimak. Learning and inference over constrained output. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1124–1129, 2005.

[37] Z. Qian, A. M. Alaa, A. Bellot, J. Rashbass, and M. van der Schaar. Learning dynamic and personalized comorbidity networks from event data using deep diffusion processes. *arXiv preprint arXiv:2001.02585*, 2020.

[38] S. Ramalingam, P. Kohli, K. Alahari, and P. H. S. Torr. Exact inference in multi-label crfs with higher order cliques. In *CVPR*, 2008.

[39] M.-A. Rizoiu and L. X. Xie. Online popularity under promotion: Viral potential, forecasting, and the economics of time. In *Eleventh International AAAI Conference on Web and Social Media*, 2017.

[40] M.-A. Rizoiu, S. Mishra, Q. Kong, M. Carman, and L. Xie. Sir-hawkes: linking epidemic models and hawkes processes to model diffusions in finite populations. In *WWW*, pages 419–428, 2018.

[41] A. Saichev and D. Sornette. Generating functions and stability study of multivariate self-excited epidemic processes. *The European Physical Journal B*, 83(2): 271, 2011.

[42] O. Shchur, M. Biloš, and S. Günnemann. Intensity-free learning of temporal point processes. *arXiv preprint arXiv:1909.12127*, 2019.

[43] D. Tarlow, I. Givoni, and R. Zemel. Hop-map: Efficient message passing with high order potentials. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AI-STATS)*, volume 9, pages 812–819. JMLR: W&CP, 2010.

[44] M. Trinh. *Non-stationary processes and their application to financial high-frequency data*. PhD thesis, University of Sussex, 2018.

[45] U. Upadhyay, A. De, and M. G. Rodriguez. Deep reinforcement learning of marked temporal point processes. In *NeurIPS*, pages 3168–3178, 2018.

[46] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016.

[47] B. Vassøy, M. Ruocco, E. de Souza da Silva, and E. Aune. Time is of the essence: a joint hierarchical rnn and point process model for time and item predictions. In *Web Search and Data Mining*, pages 591–599, 2019.

[48] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*. 2017.

[49] A. Venkatraman, M. Hebert, and J. Bagnell. Improving multi-step prediction of learned time series models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[50] R. Wen, K. Torkkola, and B. Narayanaswamy. A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053*, 2017.

[51] S. Xiao, M. Farajtabar, X. Ye, J. Yan, L. Song, and H. Zha. Wasserstein learning of deep generative point process models. In *Advances in neural information processing systems*, pages 3247–3257, 2017.

[52] 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. *arXiv preprint arXiv:1703.08524*, 2017.

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

[54] S. Xiao, H. Xu, J. Yan, M. Farajtabar, X. Yang, L. Song, and H. Zha. Learning conditional generative models for temporal point processes. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[55] S. Xiao, J. Yan, M. Farajtabar, L. Song, X. Yang, and H. Zha. Learning time series associated event sequences with recurrent point process networks. *IEEE transactions on neural networks and learning systems*, 30(10):3124–3136, 2019.

[56] A. S. Yang. *Modeling the Transmission Dynamics of Pertussis Using Recursive Point Process and SEIR model*. PhD thesis, UCLA, 2019.

[57] S.-H. Yang and H. Zha. Mixture of mutually exciting processes for viral diffusion. In *International Conference on Machine Learning*, pages 1–9, 2013.

[58] Y. Zhong, B. Xu, G.-T. Zhou, L. Bornn, and G. Mori. Time perception machine: Temporal point processes for the when, where and what of activity prediction. *arXiv preprint arXiv:1808.04063*, 2018.

[59] K. Zhou, H. Zha, and L. Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *Artificial Intelligence and Statistics*, pages 641–649, 2013.

[60] S. Zuo, H. Jiang, Z. Li, T. Zhao, and H. Zha. Transformer hawkes process. *arXiv preprint arXiv:2002.09291*, 2020.