Attentive Neural Point Processes for Event Forecasting

Yulong Gu
Alibaba Group
guyulongcs@gmail.com

Abstract

Event sequence, where each event is associated with a marker and a timestamp, is increasingly ubiquitous in various applications. Accordingly, event forecasting emerges to be a crucial problem, which aims to predict the next event based on the historical sequence. In this paper, we propose ANPP, an **A**ttentive **N**eural **P**oint **P**rocesses framework to solve this problem. In comparison with state-of-the-art methods like recurrent marked temporal point processes, ANPP leverages the time-aware self-attention mechanism to explicitly model the influence between every pair of historical events, resulting in more accurate predictions of events and better interpretation ability. Extensive experiments on one synthetic and four real-world datasets demonstrate that ANPP can achieve significant performance gains against state-of-the-art methods for predictions of both timings and markers. To facilitate future research, we release the codes and datasets at https://github.com/guyulongcs/AAAI2021_ANPP.

1 Introduction

Event sequence is increasingly ubiquitous in various areas, such as medical industry, finance, e-commerce and so on. Accordingly, event forecasting, which aims to predict what type of event will happen at what time, emerges to be a challenging but crucial problem. Figure 1 illustrates a real example of a customer's purchase sequence in an E-commerce site, and accurate predictions of users' purchases would be extremely important for personalized recommendation and marketing (Gu et al. 2020a,b; Liu et al. 2020; Manzoor and Akoglu 2017).

Temporal point processes (Daley and Vere-Jones 2007) present a general mathematical framework to model the sequential event data according to the prior knowledge of the application scenarios. Typical methods like Hawkes processes (Hawkes 1971) assume that the historical events have independent influence on the next event, and specify a fixed parametric conditional intensity function $\lambda^*(t)$ to formulate the influence between each historical event (e.g., the first n events in Figure 1) and the target one (e.g., the last event in the figure) (Manzoor and Akoglu 2017; Bray and Schoenberg 2013). Their specialized parametric forms of the conditional intensity functions, which might be oversimplified or even unfeasible in real-world scenarios, limit the capability of capturing complex dependencies among events.

To alleviate the over-simplification problem, recurrent marked temporal point processes (RMTPP, or Neural point processes) (Du et al. 2016; Mei and Eisner 2017) utilize recurrent neural networks (RNN) (Graves, Mohamed, and Hinton 2013) to model the dynamics in the event sequence and learn a conditional intensity function based on the hidden vector in RNN. They do not need any prior knowledge about the forms of the conditional intensity function, and have achieved state-of-the-art performance. However, they have following shortcomings: (1) They rely heavily on RNN, which are sequential structures and are hard to model the long-term dependencies in the sequence (Vaswani et al. 2017), even equipped with Long Short-Term Memory and Gated Recurrent Units. When the historical event and target event are far away in the sequence, RMTPP based approaches are hard to capture the influence between them. (2) They are hard to model the complex influence between every pair of historical events, which is crucial for event forecasting. For example, as shown in Figure 1, modeling influences between historical events can help us know that "Nail Polish" and "Oils" are highly related because they are purchased in proximate time, and accordingly a "Nail Polish" can be predicted correctly after the user's last purchase. (3) They have limited interpretation ability for their predictions.

To address these problems, we propose ANPP, an **A**ttentive **N**eural **P**oint **P**rocesses framework, which exploits time-aware self-attention layer to explicitly model the influence between each pair of historical events. Self-attention have achieved state-of-the-art performance in Natural language Processing. However, it is naturally unable to model the continuous time information in event sequence. In this work, we propose the Inter-event duration bucket embedding method to embed the time intervals information between events. Firstly, ANPP embeds markers and time intervals in the event sequence into vectors. Then it leverages multi-head time-aware self-attention to model the influence between each pair of the events as attention scores, and represent historical events as context vectors. Based on the generated context vectors, the marker of the next event can be predicted straightforwardly, and the conditional intensity function is calculated as well, which can be used to further predict the timing of the next event. Compared with

Figure 1: A real example of event sequence in Amazon

existing methods, ANPP has following advantages: (1) It can better model the long-term and complex dependencies between events, and achieves superior performance than existing methods. (2) It has better interpretation ability than RMTPP based approaches. The attention scores in ANPP can represent the degree of dependence between events, where a larger attention score between two events implies that they have stronger dependency.

We conduct extensive experiments on one synthetic and four real-world datasets across a variety of domains, which demonstrate the significant performance gains of ANPP against state-of-the-art methods: Averagely, ANPP reduces the Mean Absolute Error by 14.9% for time prediction, and achieves accuracy gains by 38.1% for marker prediction.

To sum up, the contributions of this work are as follows:

- We propose the idea of explicitly modeling the influence between every pair of historical events for event forecasting.
- We propose time-aware self-attention layers to learn the conditional intensity function, which equips conventional self-attention techniques with our proposed Inter-event duration bucket embedding method to model the timing information for accurate event predictions.
- We conduct extensive experiments on both synthetic and real-world datasets and demonstrate that ANPP has more accurate prediction power and better interpretation ability than state-of-the-art methods.

2 Related Work

2.1 Temporal Point Processes

Temporal point processes present a general mathematical framework for modeling sequential events (Daley and Vere-Jones 2007). They have been successfully applied in various applications, such as earthquake predictions (Bray and Schoenberg 2013), financial analysis (Embrechts, Liniger, and Lin 2011), health prediction (Du et al. 2016), continuous-time document streams clustering (Du et al. 2015), knowledge representation (Trivedi et al. 2017), purchase forecasting (Manzoor and Akoglu 2017), user profiling (Wang et al. 2017; Feng et al. 2018; Chen et al. 2020, 2019b; Gu et al. 2016), popularity prediction (Liao et al. 2019), online behaviors modeling (Cai et al. 2018), real-world behaviors modeling (Kurashima, Althoff, and Leskovec 2018), information diffusion (Cao et al. 2017; Kong, Rizoiu, and Xie 2020), time-aware recommendation (Bai et al. 2019; Vassøy et al. 2019; Wang et al. 2019) and so on. For example, the Hawkes process (Hawkes 1971) is one of the most widely used temporal point processes (Bray and Schoenberg 2013; Manzoor and Akoglu 2017), which captures the mutual excitation among events to make predictions. The core concepts of the temporal point processes are introduced as follows.

Conditional intensity function. The conditional intensity function $\lambda^*(t)$ formulates the influence of the past events (Daley and Vere-Jones 2007), which specifies the probability that a new event occurs within a small time window $[t, t + dt]$ given a sequence of historical events $\mathcal{H}_t$. Here the notation $*$ emphasizes that the function is conditional on the history. Formally, it can be presented as $\lambda^*(t)dt = \mathbb{P}\{event\ occurs\ in\ [t, t + dt] \mid \mathcal{H}_t\}$.

Conditional density function. The conditional density function $f^*(t)$ calculates the probability that the next event occurs at time t given all historical events $\mathcal{H}_t$. The relation between $\lambda^*(t)$ and $f^*(t)$ (Daley and Vere-Jones 2007) can be formulated as follows: $\lambda^*(t) = \frac{f^*(t)}{1-F^*(t)}$, where $F^*(t) = \int_0^t f^*(\tau)d\tau$ is the cumulative probability distribution function of $f^*(t)$. However, typical point processes assume that historical events have independent influence on the target event. They generally exploit a predefined parametric conditional intensity function $\lambda^*(t)$ to present the influence between each historical event and the target event. For example, the condition intensity function of the Hawkes process is defined as $\lambda^*(t) = \mu + \alpha \sum_{t_j<t} \exp\left(-\beta(t - t_j)\right)$, where $\mu \geq 0$ and $\alpha, \beta > 0$. In practice, their independence assumption is usually not true, and practical intensity functions might be extremely different in various real-world scenarios. Such drawbacks significantly limit the flexibility and performance of the methods.

2.2 Recurrent Marked Temporal Point Processes

To improve the flexibility of the temporal point processes, state-of-the-art methods (Du et al. 2016; Xiao et al. 2017b; Cao et al. 2017; Yang, Cai, and Reddy 2018; Qiao et al. 2018; Mei and Eisner 2017; Guo, Li, and Liu 2018; Loaiza-Ganem et al. 2019; Pan et al. 2020; Boyd et al. 2020) exploit Recurrent Neural Networks to model event sequence.

Du et al. (2016) used recurrent neural networks (RNN) to learn the conditional intensity function, which is referred to as recurrent marked temporal point processes. Furthermore, Xiao et al. (2017b) leveraged two RNNs to model the time series and the event sequence respectively, demonstrating effectiveness of their method on a machine maintenance dataset. Qiao et al. (2018) combined RNN and point process for clinical event prediction. Mei and Eisner (2017) utilized the continuous-time LSTM for learning. These approaches do not need any prior-knowledge about the parametric forms of the conditional intensity function, and have achieved state-of-the-art performance for event prediction. However, these approaches process the historical events sequentially and encode all of the historical information into one hidden state vector, which limits their expression capability. Furthermore, they cannot explicitly represent the influences between every pair of the historical events. These challenges deteriorate their prediction performance. Besides, they have limited ability for interpreting their predictions. In this paper, we focus on investigating better representations of historical behaviors for event forecasting. The most related approaches are RMTPP (Du et al. 2016) and NeuralHawkes (Mei and Eisner 2017). The pair-wise (Qiao et al. 2018), hierarchical RNN (Vassøy et al. 2019), contextual information enhanced (Okawa et al. 2019), Hazard function (Omi, Aihara et al. 2019), Wasserstein distance (Xiao et al. 2018, 2017a), Survival analysis (Jing and Smola 2017; Zhou et al. 2018; Ren et al. 2019), Adversarial Learning (Yan et al. 2018) and reinforcement learning (Li et al. 2018; Upadhyay, De, and Gomez Rodriguez 2018) based approaches are orthogonal to our method.

2.3 Self-Attention

Self-attention is an attention mechanism that learns a representation of a sequence by modeling the influences between different positions in the sequence (Vaswani et al. 2017) . It has been successfully applied into a variety of tasks, such as machine translation (Vaswani et al. 2017), speech recognition(Povey et al. 2018), recommender systems (Kang and McAuley 2018; Chen et al. 2019a; Gu et al. 2020b; Zou et al. 2020) and so on. However, it is naturally unable to predict the time of next event, and thus cannot be directly leveraged in the event forecasting problem.

3 Problem Formulation

Notations. Let $\mathcal{S} = \{S^1, S^2, \ldots, S^{|\mathcal{S}|}\}$ be a set of event sequences. The ith sequence $S^i = \langle (t_1, m_1), (t_2, m_2), \ldots, (t_{|S^i|}, m_{|S^i|}) \rangle$, where the positive real number $t_j \in \mathbb{R}^+$ presents the occurrence timestamp of the jth event, and the categorical variable $m_j \in M$ indicates the marker of the event. In particular, the markers generally provide the attribute information of the events (e.g., the type of the event). $d_{j+1} = t_{j+1} - t_j$ represents the time interval between the jth event and its successive one, which is referred to as the $(j+1)$th inter-event duration.

Definition 1 (The Event Forecasting Problem) *Given a set of historical event sequences $\mathcal{S} = \{S^1, S^2, \ldots, S^{|\mathcal{S}|}\}$, where $S^i = \langle (t_1, m_1), (t_2, m_2), \ldots, (t_{|S^i|}, m_{|S^i|}) \rangle$ is the ith sequence of events, the event forecasting problem aims to predict the next event $(t_{|S^i|+1}, m_{|S^i|+1})$ based on the historical events S^i for $i = 1, 2, \ldots, |\mathcal{S}|$.*

4 Attentive Neural Point Processes

In this paper, we propose an Attentive Neural Point Processes framework ANPP for event forecasting. The architecture of ANPP is demonstrated in Figure 2, which consists of an input layer, an embedding layer, a Time-Aware Self-Attention layer, and a prediction layer.

4.1 Input Layer

Given an arbitrary event sequence $S = \langle e_1, e_2, \ldots, e_T \rangle$, the event forecasting problem aims to predict the next event $e_T = (t_{T+1}, m_{T+1})$ in the future. As previous work did (Du et al. 2016), we add some special "padding" events at the end of the sequence if the length of the sequence is smaller than T, and choose the most recent T events if the length is larger than T. By doing this, all of the event sequences share a fixed-length T. In the training stage, the model iteratively fits the ground-truth event e_{j+1} at the j-th time step based on the previous events in the sequence for $j = 1, 2, \ldots, T-1$. Thus the input of the model is the event sequence $\langle e_1, e_2, ..., e_{T-1} \rangle$, and the expected output is a shifted version of the input sequence, i.e. $\langle e_2, e_3, \ldots, e_T \rangle$.

4.2 Embedding Layer

The embedding layer aims to represent the markers and timings of the events into dense vectors. In particular, we use an embedding matrix $\mathbf{M} \in \mathbb{R}^{|M| * d}$ to embed markers, where $|M|$ is the number of the marker types and d is the dimensionality of the embedding vectors. Besides, we consider the duration between the times of the successive events in this layer, and propose an **Inter-event duration bucket embedding** method for embedding such timing information. For a sequence of $(T+1)$ events, there are T inter-event durations. Generally in different applications, inter-event durations might vary a lot, from seconds to hours, days or even years. This method can adaptively embed inter-event durations with different scales. Specifically, we firstly discretize the durations into L buckets, where the bucket i means that the value falls in the range $[2^i, 2^{i+1})$ (unit time, e.g. seconds) for $i = 0, 1, ..., L-1$. Then we embed the discretized buckets in the sequence into an embedding matrix $\mathbf{D_b} \in \mathbb{R}^{T * d}$ using a learnable bucket embedding matrix $\mathbf{E_L} \in \mathbb{R}^{L * d}$. Then, as previous works did (Kang and McAuley 2018; Vaswani et al. 2017), we embed the event e_j into $\mathbf{e}_j = \mathbf{e}_{m_j} + \mathbf{e}_{t_j}$, the sum of embedding of the marker and embedding of time (i.e. the inter-event duration bucket embedding). Finally, the embedding of the sequence $S = \{e_1, e_2, ..., e_T\}$ can be represented as a matrix $\mathbf{E} \in \mathbb{R}^{T * d}$, where the j-th row is the embedding vector of the j-th event e_j.

In the experiments, we compare the inter-event duration bucket embedding method with two widely used time embedding methods, which are listed as follows: (1) **Positional embedding** is a common method in self-attention (Kang and

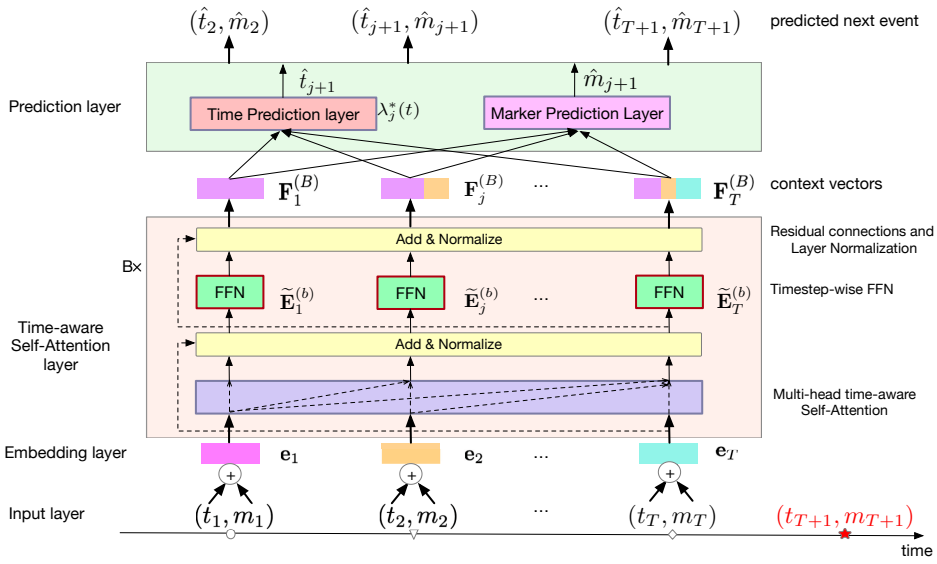


Figure 2: The Architecture of our Attentive Neural Point Processes framework (ANPP)

McAuley 2018; Vaswani et al. 2017), which directly encodes the positions $1, 2, ..., T$ of the sequence into an embedding matrix $\mathbf{D_p} \in \mathbb{R}^{T*d}$, where T is the length of the sequence and d is the dimensionality of the embedding vectors. (2) **Inter-event duration embedding** has been successfully used in the recurrent marked temporal point processes (Du et al. 2016). It utilizes a learnable matrix $\mathbf{I} \in \mathbb{R}^{1*d}$ to transform the sequence's inter-event durations vector $\mathbf{v} \in \mathbb{R}^{T*1}$ into an embedding matrix $\mathbf{D_i} = \mathbf{vI} \in \mathbb{R}^{T*d}$.

4.3 Time-Aware Self-Attention Layer

The time-aware self-attention layer is composed of a stack of B identical blocks, each consisting of a multi-head time-aware self-attention layer and a timestamp-wise fully connected feed-forward networks layer.

Multi-head time-aware self-attention layer. To capture the influence between every pair of the historical events, we use time-aware self-attention to model the embeddings of the past events, which represents both the markers and the timings information. For the b-th block, the input is a matrix $\mathbf{E}^{(b)} \in \mathbb{R}^{T\times d}$ and $\mathbf{E}^{(1)} = \mathbf{E}$, which is the embedding of the events generated from the embedding layer. In particular, $\mathbf{E}^{(b)}$ is converted into three matrices through linear projections, which are fed into an attention function for calculations. Formally, it can be formulated as follows:

$$SA(\mathbf{E}^{(b)}) = Attention(\mathbf{E}^{(b)}\mathbf{W}^Q, \mathbf{E}^{(b)}\mathbf{W}^K, \mathbf{E}^{(b)}\mathbf{W}^V) \quad (1)$$

where $\mathbf{W}^Q \in \mathbb{R}^{d\times d_k}, \mathbf{W}^K \in \mathbb{R}^{d\times d_k}, \mathbf{W}^V \in \mathbb{R}^{d\times d_v}$ are linear projection matrices, and the output $SA(\mathbf{E}^{(b)}) \in \mathbb{R}^{T\times d_v}$ is a new representation of the event sequence based on the time-aware self-attention. In Equation (1), we use the scaled dot-product attention (Vaswani et al. 2017) for implementation, which is formally represented as follows:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{QK}^T}{\sqrt{d_k}})\mathbf{V} = \mathbf{A}_s\mathbf{V} \quad (2)$$

where $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ present the queries, keys and values respectively, d_k, d_k and d_v are the dimensions of the queries, keys and values. The attention scores of the block b form a matrix $\mathbf{A}_s^{(b)}$ with the dimension $\mathbb{R}^{T\times T}$, where the element $\mathbf{A}_{s_{ij}}^{(b)}$ represents the influence of the jth event on the ith event in the bth block, and $j <= i$ holds.

We use multiple heads to calculate the input matrix $\mathbf{E}^{(b)}$ from different semantic subspaces (Vaswani et al. 2017). It firstly performs a time-aware self-attention to yield each head, which can run in parallel. Then the results are concatenated and then projected to $\widetilde{\mathbf{E}}^{(b)} \in \mathbb{R}^{T\times d}$ with the matrix $\mathbf{W}^O$. The process can be formulated as follows:

$$head_i = SA(\mathbf{E}^{(b)})_i = Attention(\mathbf{E}^{(b)}\mathbf{W}_i^{\mathbf{Q}}, \mathbf{E}^{(b)}\mathbf{W}_i^{\mathbf{K}}, \mathbf{E}^{(b)}\mathbf{W}_i^{\mathbf{V}})$$

$$Multihead(\mathbf{E}^{(b)}) = \widetilde{\mathbf{E}}^{(b)} = Concat(head_1, ..., head_h)\mathbf{W}^O$$

where $\mathbf{W}_i^Q \in \mathbb{R}^{d\times d_k}, \mathbf{W}_i^K \in \mathbb{R}^{d\times d_k}, \mathbf{W}_i^V \in \mathbb{R}^{d\times d_v}$, $\mathbf{W}^O \in \mathbb{R}^{hd_v\times d}$, h is the number of heads, and $\widetilde{\mathbf{E}}^{(b)} \in \mathbb{R}^{T\times d}$ is the new representation of the event sequence with the multi-head transformation.

Timestamp-wise fully connected feed-forward networks. To learn higher-level representations of the historical events, we leverage fully connected feed-forward networks (FFN) to process each vector in the matrix $\widetilde{\mathbf{E}}^{(b)} = [\widetilde{\mathbf{E}}_1^{(b)}; \widetilde{\mathbf{E}}_2^{(b)}; ...; \widetilde{\mathbf{E}}_T^{(b)}]$ separately and identically. As shown in Equation (3), the network consist of two linear transformations with a ReLU activation in between. They transforms $\widetilde{\mathbf{E}}^{(b)} \in \mathbb{R}^{T\times d}$ into a matrix $\mathbf{F}^{(b)} \in \mathbb{R}^{T\times d}$, which is the new representation of the historical event sequence. Formally,

$$FFN(\widetilde{\mathbf{E}}_j^{(b)}) = \mathbf{F}_j^{(b)} = ReLU(\widetilde{\mathbf{E}}_j^{(b)}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \quad (3)$$

where $\mathbf{W}_1 \in \mathbb{R}^{d*d_f}, \mathbf{W}_2 \in \mathbb{R}^{d_f*d}, \mathbf{b}_1 \in \mathbb{R}^{d_f}$ and $\mathbf{b}_2 \in \mathbb{R}^d$.

Stacking time-aware self-attention blocks. To formulate more complex influences between events, we stack the self-attention block B times (Kang and McAuley 2018; Vaswani

et al. 2017), where the b-th block transforms $\mathbf{F}^{(b-1)} \in \mathbb{R}^{T\times d}$ into $\mathbf{F}^{(b)} \in \mathbb{R}^{T\times d}$ for $b = 2, 3, \ldots, B$, and the jth row $\mathbf{F}_j^{(b)}$ calculates the influences of all previous events on event j in this block. Finally, these B stacking self-attention blocks hierarchically estimate the influence of the historical events and learn the new representation of the event sequence as a context matrix $\mathbf{F}^{(B)} \in \mathbb{R}^{T\times d}$ for event prediction.

4.4 Prediction Layer

The prediction layer is consisted of a marker prediction layer and a time prediction layer, which can predict the marker and timing of next event based on the context matrix $\mathbf{F}^{(B)} = [\mathbf{F}_1^{(B)}; \mathbf{F}_2^{(B)}; \ldots; \mathbf{F}_T^{(B)}] \in \mathbb{R}^{T\times d}$.

Marker prediction layer aims to predict the marker of the next event $\hat{m}_{j+1}$. Given the context vector $\mathbf{F}_j^{(B)}$, the probability of $\hat{m}_{j+1}$ is calculated as follows:

$$\mathbf{P}(\hat{m}_{j+1} \mid \mathbf{F}_j^{(B)}) = softmax(\mathbf{F}_j^{(B)}\mathbf{W}^m + \mathbf{b}^m) \tag{4}$$

where $\mathbf{W}^m \in \mathbb{R}^{d*|M|}$ and $\mathbf{b}^m \in \mathbb{R}^{1*|M|}$.

Time prediction layer seeks to predict the occurrence timestamp of the next event $\hat{t}_{j+1}$. Firstly, inspired by the previous work (Du et al. 2016), we calculate the condition intensity function $\lambda^*(t)$ as follows:

$$\lambda^*(t) = \exp\left(\mathbf{v}^{t^T} \cdot \mathbf{F}_j^{(B)} + \beta(t - t_j) + \lambda_0\right) \tag{5}$$

where $\mathbf{v}^t \in \mathbb{R}^{d\times 1}$ and β, λ_0 are scalars. In Equation (5), the first term $\mathbf{v}^{t^T} \cdot \mathbf{F}_j^{(B)}$ calculates the accumulative influence among the past events, the second term $\beta(t - t_j)$ indicates the evolutionary process of the intensity function with time, and the last term λ_0 represents the base intensity of the next event. The exponential function acts as a non-linear transformation, which guarantees a positive value of the intensity function. With $\lambda^*(t)$, we can further derive the conditional density function $f^*(t) = \lambda^*(t) \exp\left(-\int_{t_j}^{t} \lambda^*(\tau)d\tau\right)$ (Daley and Vere-Jones 2007). In the training stage, the probability that the next event would occur at time t_{j+1} can be calculated as $p(\hat{t}_{j+1} = t_{j+1} \mid \mathbf{F}_j^{(B)}) = f^*(t_{j+1} - t_j) = f^*(d_{j+1})$. In the test stage, the time $\hat{t}_{j+1}$ of the next event can be predicted as the expectation $\hat{t}_{j+1} = \int_{t_j}^{\infty} t \cdot f^*(t)dt$, which can be calculated with numerical integration (Daley and Vere-Jones 2007).

4.5 Parameter Learning

Let $\mathcal{S} = \{S^1, S^2, ..., S^{|\mathcal{S}|}\}$ be a set of event sequences, where $S^i = \langle (t_1, m_1), (t_2, m_2), ..., (t_{|S^i|}, m_{|S^i|})\rangle$ is the ith sequence of events. The model can be trained by maximizing the jointly weighted log-likelihood of the observations. As demonstrated in Section 4.4, at the jth step for handling S^i, the ground truth next event is (t_{j+1}, m_{j+1}), the probability of it's marker and timing are $p(\hat{m}_{j+1} = m_{j+1} \mid \mathbf{F}_j^{(B)}) = \mathbf{P}(\hat{m}_{j+1} \mid \mathbf{F}_j^{(B)})_{m_{j+1}}$ and $p(\hat{t}_{j+1} = t_{j+1} \mid \mathbf{F}_j^{(B)}) = f^*(d_{j+1})$, respectively. Then loss function of ANPP is defined as the negative jointly weighted log-likelihood:

$$L(\mathcal{S}) = -\sum_{i=1}^{|\mathcal{S}|}\sum_{j=1}^{|S_i|}\Big[\omega_m \log \mathbf{P}\left(\hat{m}_{j+1} \mid \mathbf{F}_j^{(B)}\right)_{m_{j+1}} + (1-\omega_m)\log f^*\left(d_{j+1}\right)\Big] \tag{6}$$

where ω_m balances the importance of the marker loss and the timing loss.

5 Experiments

5.1 Datasets

In the experiments, we use one synthetic dataset "Hawkes" and four real-world datasets in medical, financial and e-commerce domains respectively as benchmarks, which are widely used in literature (Du et al. 2016; Mei and Eisner 2017; Kang and McAuley 2018). Table 1 shows the numbers of the marker types $|M|$ and the total numbers of the events $|E|$ in these datasets.

Table 1: Statistics of Dataset

Data	Hawkes	Financial	MIMIC	Amazon (Beauty)	Amazon (Clothes)
$\|M\|$	1	2	75	226	482
$\|E\|$	99,968	414,800	2,419	144,219	183,637

- **Hawkes.** It is a synthetic dataset, which has been generated by a Hawkes process (Du et al. 2016).
- **Financial.** The financial transactions dataset has been collected from NYSE of the high-frequency transactions for a stock in one day (Du et al. 2016). Each transaction record logs the time and the possible action (buy or sell). The action types are treated as the markers.
- **MIMIC.** It is a medical dataset, including a collection of de-identified electronic medical records of patients between 2001 and 2008 (Du et al. 2016). Each event records the time when a patient is diagnosed with a disease, where the type of the disease is treated as the marker.
- **Amazon.** This dataset contains sequences of the product reviews from Amazon(He and McAuley 2016). We chose two widely used datasets *Beauty* and *Clothes* for our comparative study. The finest-level categories of the products are used as the markers.

5.2 Baseline Methods

- **RNN** uses the recurrent neural networks to predict the times and markers independently (Du et al. 2016; Graves, Mohamed, and Hinton 2013).
- **RMTPP** uses the recurrent neural networks to automatically learn the conditional intensity function from the event history (Du et al. 2016).
- **IRNN** is similar to RMTPP, but it uses the Gaussian distribution to calculate the likelihood of the time loss (Xiao et al. 2017b).

Table 2: Performance of ANPP, baseline methods and variants of ANPP for Time prediction

Dataset	Metrics	Baseline Methods				ANPP	Gains	*Variants of* ANPP			
		RNN	RMTPP	IRNN	NeuH			ANPP_p	ANPP_t	ANPP_G	ANPP_M
Hawkes	MAE	1.988	1.271	1.464	*1.046*	**0.885**	15.4%	0.972	0.958	0.917	0.981
Financial	MAE	0.428	*0.002*	0.429	0.352	**0.001**	50.0%	0.001	0.001	0.001	0.002
MIMIC	MAE	0.602	0.659	*0.541*	0.748	**0.514**	5.0%	0.550	0.524	0.525	0.590
Amazon(Beauty)	MAE	9.633	8.433	8.893	*8.397*	**8.182**	2.6%	8.458	8.611	9.972	9.871
Amazon(Clothes)	MAE	11.55	10.31	10.77	*10.06*	**9.884**	1.7 %	10.17	10.20	11.46	11.37

Table 3: Performance of ANPP, baseline methods and variants of ANPP for Marker prediction

Dataset	Metrics	Baseline Methods				ANPP	Gains	*Variants of* ANPP			
		RNN	RMTPP	IRNN	NeuH			ANPP_p	ANPP_t	ANPP_G	ANPP_M
Financial	ACC	0.507	*0.575*	0.507	0.507	**0.613**	6.6%	0.600	0.599	0.612	0.611
MIMIC	ACC	0.406	*0.676*	0.406	0.388	**0.847**	25.3%	0.835	0.847	0.829	0.729
	NDCG@20	0.672	*0.753*	0.672	0.654	**0.895**	18.9%	0.894	0.895	0.885	0.828
	HR@20	0.940	0.903	*0.941*	0.928	**0.953**	1.3%	0.940	0.953	0.935	0.933
Amazon (Beauty)	ACC	0.055	*0.113*	0.055	0.056	**0.161**	42.5%	0.147	0.146	0.160	0.160
	NDCG@20	0.226	*0.302*	0.225	0.226	**0.358**	18.5%	0.344	0.344	0.357	0.356
	HR@20	0.521	*0.591*	0.519	0.514	**0.658**	11.3%	0.648	0.647	0.657	0.654
Amazon (Clothes)	ACC	0.048	*0.077*	0.048	0.048	**0.137**	77.9%	0.116	0.119	0.135	0.130
	NDCG@20	0.181	*0.221*	0.181	0.185	**0.302**	36.7%	0.276	0.280	0.297	0.293
	HR@20	0.416	*0.464*	0.416	0.430	**0.554**	19.4%	0.530	0.536	0.548	0.545

- **NeuH** is Neural Hawkes model (Mei and Eisner 2017).

Variants of ANPP. We conduct an ablation study of our model by implementing several variants of ANPP. To investigate the effectiveness of our proposed Inter-event duration bucket embedding method, we design two variants: ANPP_p and ANPP_t, which use the positional embedding and the inter-event duration embedding methods (described in Section 4.2) for handling the time information, respectively. To investigate the effectiveness of the point process mechanism for time prediction, we design another two variants: ANPP_G and ANPP_M. Both of them share the same marker prediction techniques as ANPP, but they treat time prediction as a regression problem, and directly use neural networks to predict the time of the next event. In particular, ANPP_G optimizes a Gaussian-based time loss $-\log\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(\frac{\left(t_{j+1}-\hat{t}_{j+1}\right)^2}{-2\sigma^2}\right)\right]$ (Xiao et al. 2017b), and ANPP_M is based on a mean squared error-based time loss $(t_{j+1}-\hat{t}_{j+1})^2$.

Implementation details. We implement our ANPP framework with Tensorflow. The Adam optimizer is utilized for training. In the experiments, for each dataset, we randomly select 70% sequences for training, 10% sequences for validation and the rest 20% sequences for testing. The learning rate, batch size, hidden vector dimension, dropout rate are set to 0.001, 64, 64 and 0.5 respectively. For other hyperparameters that were used in previous work (e.g. maximum sequence length), we follow the settings in existing works (Du et al. 2016; Mei and Eisner 2017) to make fair comparisons. The number of blocks and heads are set to 2 and 4 using the validation set.

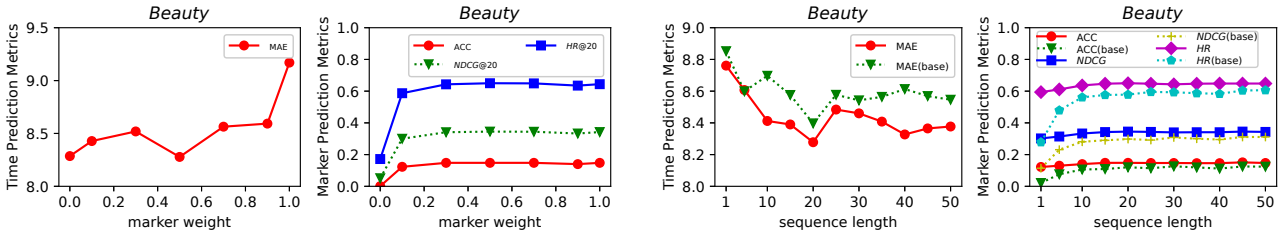
5.3 Evaluation Metrics

For time prediction, we use the **MAE** (Mean Absolute Error) metric, which measures the mean absolute error between the predicted time and the ground-truth. For marker prediction, we use three metrics for comparisons, including **ACC** (Accuracy), **NDCG@K** (Normalized Discounted Cumulative Gain), and **HR@K** (Hit Ratio) (Du et al. 2016; Kang and McAuley 2018; Valizadegan et al. 2009).

5.4 Experimental Results

Overall comparison. The experimental comparisons of our ANPP with its variants and the baselines are presented in Tables 2 and 3, where all of our gains over the best baselines are statistically significant with $p < 0.01$ in the t-test (Smucker, Allan, and Carterette 2007). As only one type of marker is involved in the Hawkes dataset, only the time of the events is needed to be predicted. Besides, since the Financial dataset only contains two types of markers, we only need the ACC metric for the marker prediction task. From the tables we can find that:

- Our method ANPP outperforms all of the baselines over all of the datasets for both tasks. Compared to the strongest baseline, ANPP averagely reduces the MAE by 14.9% for time prediction, gains 38.1% in ACC, 24.7% in NDCG@K and 10.7% in HR@K for marker prediction.
- RMTPP and NeuH generally perform better than RNN and IRNN. Furthermore, ANPP beats ANPP_G and ANPP_M for both tasks, and such improvement for the time prediction task is especially significant. These observations illustrate the effectiveness of the point processes in event forecasting.



(a) Performance with the marker weight ω_m

(b) Performance with the sequence length T

Figure 3: Performance with different marker weights and sequence lengths

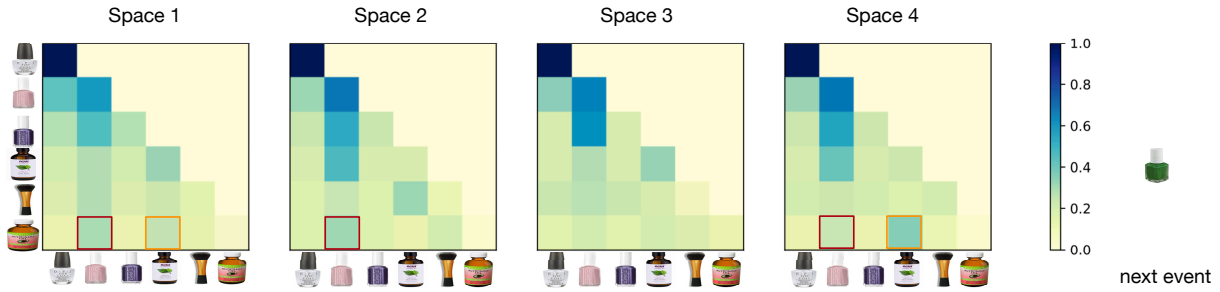


Figure 4: Case Study: Heatmap of attention scores in ANPP

- ANPP performs better than both ANPP$_p$ and ANPP$_t$ on all datasets, which demonstrates the crucial role of our inter-event duration bucket embedding method for time modeling.

Model analysis. The performance of ANPP with different parameters on the $Beauty$ dataset is shown in Figures 3(a) and 3(b). The results on other datasets are similar.

- As shown in Figure 3(a), ANPP achieves the best time predictions when ω_m is 0.5, while the performance of ANPP for marker predictions is rising as ω_m increases from 0 to 0.5, and then keeps stable when ω_m grows even larger. Such phenomena are coherent with the insight that the marker and the time information are consistent, and thus training with both could be better than with either of them alone (Du et al. 2016).
- As shown in Figure 3(b), for the two tasks, the performance of both ANPP and the strongest baseline (denoted as "base") increases when T ranges from 1 to about 20, and it is coherent with the intuition that forecasting based on longer sequences is easier because of their richer information. Their performance would keep relatively stable when T is larger than 20. Furthermore, ANPP outperforms base with all values of T, which illustrates the promising performance of ANPP for both short and long event sequences.

Interpretation ability of ANPP. Our method ANPP uses multi-head time-aware self-attention to explicitly represent the influence between each pair of the historical events. For each head, there is an attention score matrix $\mathbf{A}_s \in \mathbb{R}^{T\times T}$ that models the influence of historical events in a semantic space. Each element $\mathbf{A}_{s_{ij}}$ in row i and column j measures the influence of the j-th event on the i-th event ($j <= i$). For a consumer's purchase sequence shown in Figure 1, we visualize the attention scores among the events in different semantic spaces in Figure 4. From the figure we can find that the attention scores equip ANPP with better interpretation ability than state-of-the-art methods (Du et al. 2016; Mei and Eisner 2017).

- Higher attention scores tend to focus on some specific columns. In all of the spaces, the scores of the 2nd and 4th event are higher, but the scores of the 5th event are lower. This demonstrates that the main interests of the user are "Nail Polish" and "Oil".
- Different semantic spaces may focus on various factors. For example, in the 6th row, the attention score of the 2nd column is higher than others in spaces 1 and 2, while that of the 4th column is the highest in space 4. That might be because spaces 1 and 2 consider more about the overall importance of the event in the sequence, while space 4 appreciates more about similarity between two events.

6 Conclusion

We present ANPP, an Attentive Neural Point Processes framework for event forecasting. It can explicitly model the complex influence between every pair of the historical events by integrating the Inter-event duration bucket embedding method into self-attention for modeling the markers and time information of historical events simultaneously. Extensive experiments in various domains demonstrate that ANPP can achieve significant performance gains and better interpretation ability against state-of-the-art methods.

References

Bai, T.; Zou, L.; Zhao, W. X.; Du, P.; Liu, W.; Nie, J.-Y.; and Wen, J.-R. 2019. CTRec: A Long-Short Demands Evolution Model for Continuous-Time Recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 675–684.

Boyd, A.; Bamler, R.; Mandt, S.; and Smyth, P. 2020. User-Dependent Neural Sequence Models for Continuous-Time Event Data. *Advances in Neural Information Processing Systems* 33.

Bray, A.; and Schoenberg, F. P. 2013. Assessment of point process models for earthquake forecasting. *Statistical science* 510–520.

Cai, R.; Bai, X.; Wang, Z.; Shi, Y.; Sondhi, P.; and Wang, H. 2018. Modeling Sequential Online Interactive Behaviors with Temporal Point Process. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 873–882.

Cao, Q.; Shen, H.; Cen, K.; Ouyang, W.; and Cheng, X. 2017. Deephawkes: Bridging the gap between prediction and understanding of information cascades. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 1149–1158.

Chen, Q.; Zhao, H.; Li, W.; Huang, P.; and Ou, W. 2019a. Behavior sequence transformer for e-commerce recommendation in alibaba. In *Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data*, 1–4.

Chen, W.; Gu, Y.; Ren, Z.; He, X.; Xie, H.; Guo, T.; Yin, D.; and Zhang, Y. 2019b. Semi-supervised user profiling with heterogeneous graph attention networks. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2116–2122.

Chen, Y.; Long, C.; Cong, G.; and Li, C. 2020. Context-aware deep model for joint mobility and time prediction. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, 106–114.

Daley, D. J.; and Vere-Jones, D. 2007. *An introduction to the theory of point processes: volume II: general theory and structure*. Springer Science & Business Media.

Du, N.; Dai, H.; Trivedi, R.; Upadhyay, U.; Gomez-Rodriguez, M.; and Song, L. 2016. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1555–1564.

Du, N.; Farajtabar, M.; Ahmed, A.; Smola, A. J.; and Song, L. 2015. 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*, 219–228.

Embrechts, P.; Liniger, T.; and Lin, L. 2011. Multivariate Hawkes processes: an application to financial data. *Journal of Applied Probability* 48(A): 367–378.

Feng, J.; Li, Y.; Zhang, C.; Sun, F.; Meng, F.; Guo, A.; and Jin, D. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 World Wide Web Conference*, 1459–1468.

Graves, A.; Mohamed, A.-r.; and Hinton, G. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, 6645–6649. IEEE.

Gu, Y.; Ding, Z.; Wang, S.; and Yin, D. 2020a. Hierarchical User Profiling for E-commerce Recommender Systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, 223–231.

Gu, Y.; Ding, Z.; Wang, S.; Zou, L.; Liu, Y.; and Yin, D. 2020b. Deep Multifaceted Transformers for Multi-objective Ranking in Large-Scale E-commerce Recommender Systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2493–2500.

Gu, Y.; Song, J.; Liu, W.; and Zou, L. 2016. HLGPS: a home location global positioning system in location-based social networks. In *2016 IEEE 16th International Conference on Data Mining*, 901–906. IEEE.

Guo, R.; Li, J.; and Liu, H. 2018. INITIATOR: Noise-contrastive Estimation for Marked Temporal Point Process. In *IJCAI*, 2191–2197.

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

He, R.; and McAuley, J. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 2016 World Wide Web Conference*, 507–517.

Jing, H.; and Smola, A. J. 2017. Neural survival recommender. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 515–524.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining*, 197–206. IEEE.

Kong, Q.; Rizoiu, M.-A.; and Xie, L. 2020. Modeling Information Cascades with Self-exciting Processes via Generalized Epidemic Models. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, 286–294.

Kurashima, T.; Althoff, T.; and Leskovec, J. 2018. Modeling interdependent and periodic real-world action sequences. In *Proceedings of the 2018 World Wide Web Conference*, 803–812.

Li, S.; Xiao, S.; Zhu, S.; Du, N.; Xie, Y.; and Song, L. 2018. Learning temporal point processes via reinforcement learning. *Advances in neural information processing systems* 31: 10781–10791.

Liao, D.; Xu, J.; Li, G.; Huang, W.; Liu, W.; and Li, J. 2019. Popularity prediction on online articles with deep fusion of temporal process and content features. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, volume 33, 200–207.

Liu, Y.; Gu, Y.; Ding, Z.; Gao, J.; Guo, Z.; Bao, Y.; and Yan, W. 2020. Decoupled Graph Convolution Network for Inferring Substitutable and Complementary Items. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2621–2628.

Loaiza-Ganem, G.; Perkins, S.; Schroeder, K.; Churchland, M.; and Cunningham, J. P. 2019. Deep Random Splines for Point Process Intensity Estimation of Neural Population Data. In *Advances in Neural Information Processing Systems*, 13346–13356.

Manzoor, E.; and Akoglu, L. 2017. RUSH! Targeted Time-limited Coupons via Purchase Forecasts. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1923–1931.

Mei, H.; and Eisner, J. M. 2017. The neural hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*, 6754–6764.

Okawa, M.; Iwata, T.; Kurashima, T.; Tanaka, Y.; Toda, H.; and Ueda, N. 2019. Deep mixture point processes: Spatio-temporal event prediction with rich contextual information. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 373–383.

Omi, T.; Aihara, K.; et al. 2019. Fully neural network based model for general temporal point processes. In *Advances in Neural Information Processing Systems*, 2122–2132.

Pan, Z.; Huang, Z.; Lian, D.; and Chen, E. 2020. A Variational Point Process Model for Social Event Sequences. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, volume 34, 173–180.

Povey, D.; Hadian, H.; Ghahremani, P.; Li, K.; and Khudanpur, S. 2018. A time-restricted self-attention layer for asr. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing*, 5874–5878. IEEE.

Qiao, Z.; Zhao, S.; Xiao, C.; Li, X.; Qin, Y.; and Wang, F. 2018. Pairwise-ranking based collaborative recurrent neural networks for clinical event prediction. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 3520–3526.

Ren, K.; Qin, J.; Zheng, L.; Yang, Z.; Zhang, W.; Qiu, L.; and Yu, Y. 2019. Deep recurrent survival analysis. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, volume 33, 4798–4805.

Smucker, M. D.; Allan, J.; and Carterette, B. 2007. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 623–632.

Trivedi, R.; Dai, H.; Wang, Y.; and Song, L. 2017. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *Proceedings of the 34th International Conference on Machine Learning*, 3462–3471.

Upadhyay, U.; De, A.; and Gomez Rodriguez, M. 2018. Deep reinforcement learning of marked temporal point processes. *Advances in Neural Information Processing Systems* 31: 3168–3178.

Valizadegan, H.; Jin, R.; Zhang, R.; and Mao, J. 2009. Learning to rank by optimizing ndcg measure. In *Advances in neural information processing systems*, 1883–1891.

Vassøy, B.; Ruocco, M.; de Souza da Silva, E.; and Aune, E. 2019. Time is of the essence: a joint hierarchical rnn and point process model for time and item predictions. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, 591–599.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems* 30: 5998–6008.

Wang, C.; Zhang, M.; Ma, W.; Liu, Y.; and Ma, S. 2019. Modeling item-specific temporal dynamics of repeat consumption for recommender systems. In *Proceedings of the 2019 World Wide Web Conference*, 1977–1987.

Wang, P.; Fu, Y.; Liu, G.; Hu, W.; and Aggarwal, C. 2017. Human mobility synchronization and trip purpose detection with mixture of hawkes processes. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 495–503.

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

Xiao, S.; Xu, H.; Yan, J.; Yang, X.; Song, L.; and Zha, H. 2018. Learning conditional generative models for temporal point processes. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

Xiao, S.; Yan, J.; Yang, X.; Zha, H.; and Chu, S. M. 2017b. Modeling the intensity function of point process via recurrent neural networkss. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 1597–1603.

Yan, J.; Liu, X.; Shi, L.; Li, C.; and Zha, H. 2018. Improving Maximum Likelihood Estimation of Temporal Point Process via Discriminative and Adversarial Learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2948–2954.

Yang, G.; Cai, Y.; and Reddy, C. K. 2018. Recurrent spatio-temporal point process for check-in time prediction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2203–2211.

Zhou, T.; Qian, H.; Shen, Z.; Zhang, C.; Wang, C.; Liu, S.; and Ou, W. 2018. JUMP: a joint predictor for user click and dwell time. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 3704–3710.

Zou, L.; Xia, L.; Gu, Y.; Zhao, X.; Liu, W.; Huang, J. X.; and Yin, D. 2020. Neural Interactive Collaborative Filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 749–758.