Influence-Aware Attention for Multivariate Temporal Point Processes

Xiao Shou SHOUX@RPI.EDU
Rensselaer Polytechnic Institute, Troy, NY, USA

Tian Gao TGAO@US.IBM.COM
Dharmashankar Subramanian DHARMASH@US.IBM.COM
Debarun Bhattacharjya DEBARUNB@US.IBM.COM
IBM Research, Thomas J. Watson Research Center, Yorktown Heights, NY, USA

Kristin P. Bennett BENNEK@RPI.EDU
Rensselaer Polytechnic Institute, Troy, NY, USA

Editors: Mihaela van der Schaar, Dominik Janzing and Cheng Zhang

Abstract

Identifying the subset of events that influence events of interest from continuous time datasets is of great interest in various applications. Existing methods however often fail to produce accurate and interpretable results in a time-efficient manner. In this paper, we propose a neural model – Influence-Aware Attention for Multivariate Temporal Point Processes (IAA-MTPPs) – which leverages the powerful attention mechanism in transformers to capture temporal dynamics between event types, which is different from existing instance-to-instance attentions, using variational inference while maintaining interpretability. Given event sequences and a prior influence matrix, IAA-MTPP efficiently learns an approximate posterior by an Attention-to-Influence mechanism, and subsequently models the conditional likelihood of the sequences given a sampled influence through an Influence-to-Attention formulation. Both steps are completed efficiently inside a B-block multi-head self-attention layer, thus our end-to-end training with parallelizable transformer architecture enables faster training compared to sequential models such as RNNs. We demonstrate strong empirical performance compared to existing baselines on multiple synthetic and real benchmarks, including qualitative analysis for an application in decentralized finance.

Keywords: Graphical Event Model, Multivariate Temporal Point Process, Variational Inference, Transformer, Attention Mechanism

1. Introduction

Many real world phenomena and human activities consist of sequences of events where events belonging to some discrete set happen irregularly in continuous time. A multivariate temporal point process (MTPP) (Daley and Jones, 2003) provides an elegant mathematical tool for modeling event sequences. For example, MTPPs are frequently used to model neural spike training in neuroscience and user activities in social networks. A classical approach to model event sequences as an MTPP is through a Hawkes process (Hawkes, 1971) where a particular parametric form is given to capture the dynamics of the events and interactions among different event types. The past few years have witnessed the rise of neural MTPP models and their state-of-the-art performance on standard benchmarks for predictive tasks (Du et al., 2016; Mei and Eisner, 2016; Xiao et al., 2017b; Omi et al., 2019; Shchur et al., 2019; Zuo et al., 2020).

In this work, we focus on identifying influencing events for MTPP data because knowledge about such interaction among different types of events has a wide range of applications. Our work is closely related to and inspired by graphical models for MTPPs that capture process (in)dependence between each individual event process – these are known as graphical event models (GEMs) or local independence graphs (Didelez, 2008; Gunawardana and Meek, 2016; Bhattacharjya et al., 2018, 2023). The parents of an event type in an underlying GEM graph provides its direct influencing events. Such a model has numerous real-world applications. For instance, accurate identification of how a group of users interact with each other in a social network can help a platform design socially enhanced applications to improve security and performance for email, web browsing and overlay routing (Wilson et al., 2009), or counter coordinated activities from malicious accounts that manipulate public opinion (Sharma et al., 2020). Another motivating example is learning how different types of actions that users make around cryptocurrency transactions influence each other; in particular, what previous actions lead to liquidation is of interest to researchers studying decentralized finance.

Multivariate Hawkes processes (MHP) naturally embed how one event influences another through its infectivity matrix (Zhou et al., 2013; Linderman and Adams, 2014; Eichler et al., 2017; Liu et al., 2018). The drawback of MHPs for real applications is that they assume certain parametric form which may be inadequate or unsuitable for capturing the dynamics and interaction among different events. Various types of neural TPP models have thus been proposed to learn more complex dynamics over the years by using a recurrent neural network (RNN), or its variants (Du et al., 2016; Mei and Eisner, 2016; Omi et al., 2019; Xiao et al., 2017b; Shchur et al., 2019). Zhang and Yan (2021) propose a neural relation inference model namely NRI-TPP for event sequences by using message passing graph and RNN. Zhang et al. (2020b) propose CAUSE model for learning Granger causality between event types by attribution methods with RNN-based neural point process models. However, RNNs often fail to capture the long-term, nonsequential dependencies of contexts. In addition, they are inefficient to train on long sequences as training cannot be parallelized. Advances in neural machine translation (Vaswani et al., 2017) further introduce transformers for modeling MTPPs and they have shown state-of-the-art performance on prediction tasks (Zhang et al., 2020a; Zuo et al., 2020; Gu, 2021). Yet transformers have not been directly applied to solve type-to-type inference problems because they only capture instance-to-instance interaction, which motivates our use of transformer-based attention models for discovering influencing event types for a given event of interest in multivariate event sequences. To fill the gap, we propose influence-aware attention for MTPPs to identify the influencing event types, which is a faster and more accurate approach as compared to existing RNN-based approaches. To the best of our knowledge, ours is the first non-trivial implementation for determining type-wise inference. Our contributions are as follows:

- We design a novel paradigm which connects a variational inference framework with an attention mechanism for learning and identifying influencing events in MTPPs.
- We propose a concise formulation for modeling event type relations into attention mechanisms, which integrates the probabilistic aspect into attention through attention-to-influence and influence-to-attention transformations within the transformer architecture.
- We show our model is more efficient and accurate than previous work and state-of-the-art performance is demonstrated through extensive experiments on simulated data and two real world applications.

2. Background

2.1. Multivariate Temporal Point Process

Multivariate temporal point processes are often used to model event streams in which discrete events take place in the continuous time domain (Daley and Jones, 2003). A D-variate temporal point process is a stochastic process that generates a sequence S of timestamps and labels namely $\{t_i, y_i\}_{i=1}^{n}$ where t_i is the time of occurrence of i^{th} event and y_i is the label of the event which belongs to a label set $\mathbb{L}$ whose cardinality is D. Strictly temporally ordered timestamps are usually assumed in a given window of observation $[0, T]$, i.e. $t_i < t_j$ for $i < j$ where $t_i \in [0, T]$ for all $i \in [1, 2, ..., n]$. A general temporal point process can be characterized by its conditional intensity function(CIF): $\lambda(t) = \lim_{\Delta t \to 0} \frac{\mathbb{E}(\mathcal{N}(t+\Delta t|H_t) - \mathcal{N}(t|H_t))}{\Delta t}$ where $\mathcal{N}(t|H_t)$ counts the number of events prior to history H_t until time t. MTPP on the other hand models CIF for each event type $\lambda_e(t)$. Classical MTPP models such as Hawkes process assume some parametric form of the conditional intensity function while neural MTPPs are more flexible to capture the underlying dynamics in a data-driven manner. Both approaches are commonly trained to minimize the negative log-likelihood. The log-likelihood of observing a sequence S is the sum of log-likelihood of events and non-events and can be expressed as the following (assuming the starting time is t_0):

$$\log p(S) = \sum_{i=1}^{n} \sum_{e=1}^{D} \log\lambda_e(t_i) - \sum_{i=0}^{n} \int_{t_i}^{t} \sum_{e=1}^{D} \lambda_e(t) dt \tag{1}$$

It is worth noting that usually analytical form in the second term in equation 1 is not available. Other alternatives are proposed: Wasserstein distance (Xiao et al., 2017a), adversarial losses (Yan et al., 2018) and reward function in inverse reinforcement learning (Li et al., 2018) are proposed and shown to achieve good results on benchmark datasets.

2.2. Multivariate Hawkes Process and Infectivity Matrix

Most classical approaches for relational inference on marked event sequences are through the learning of the infectivity matrix of an MHP, which embodies trigger coefficients among different event types. The conditional intensity function of the i^{th} dimension of a D-dimensional MHP has the following parametric form:

$$\lambda_i(t) = \mu_i + \sum_{j=1}^{D} \sum_{k:t_{j,k}<t} \mathbf{W}_{ij} g(t - t_{j,k}) \tag{2}$$

where $\mu_i \in \mathbb{R}_+$ is base intensity and $g(t) \in \mathbb{R}_+$ is a kernel function representing the extent of influence of an event. Exponential and powerlaw kernel are usually specified to model the decay of influence from an event instance on an event type over time. The matrix $\mathbf{W} \in \mathbb{R}_+^{D \times D}$ is the infectivity matrix and its entry $\mathbf{W}_{ij}$ signifies the magnitude (or nonexistence if $\mathbf{W}_{ij} = 0$) of how event type j triggers event type i. The infectivity matrix has been studied extensively for learning Granger-causality (Xu et al., 2016; Eichler et al., 2017). For example, Eichler et al. (2017) prove that the absence or existence of a causal influence from process j on process i is equivalent to the entry $\mathbf{W}_{ij}$ being 0 or not. Common learning paradigms of MHPs utilize the Expectation-Maximization (EM) framework. Zhou et al. (2013) introduce nuclear and $l1$ norm to regularize the likelihood function to learn a sparse and low-rank infectivity matrix. They leverage a combination of alternating direction method of multipliers and majorization minimization (ADM4) for optimization. Linderman and Adams (2014) combine Hawkes processes and random network models by masking the infectivity matrix with a binary adjacency matrix to learn the latent relational graph. Other related work focuses on applying appropriate constraints. Liu et al. (2018) impose graph regularization to the objective by leveraging spatial information for spatio-temporal event data. Salehi et al. (2019) take a variational approach to tackle the problem through variational EM.

2.3. Neural MTPP for Learning Influencing/Infectivity Matrix

Zhang and Yan (2021) propose a relational inference model namely NRI-TPP in a recent study. Although their work provides the first neural probabilistic relation mining for MTPP data, the message passing graph is not efficient since messages are passed stepwise for each timestamp. Furthermore, the use of RNNs to update latent states (nodes) of the graph fails to capture long-term dependencies because the interactions between events far away in time are always weakly linked within any recurrent structure (Hochreiter et al., 2001). Another recent model, CAUSE by Zhang et al. (2020b) for event data although enjoys the elegant theoretical property from attribution method, is not end-to-end: it heavily relies on the learning of a neural point processes model (with RNN) and granger-causality then is derived from such process.

2.4. Transformers on Event Data

Attention and transformer models have been used to model event data in recent years (Xiao et al., 2019; Zhang et al., 2020a; Zuo et al., 2020; Gu, 2021; Shou et al., 2023). The self-attention mechanism, in our context, relates different event instances of a single sequence in order to compute a representation of the sequence. The architecture of transformers for MTPPs consists of an embedding layer and a self-attention layer. In Transformer Hawkes Processes (THP) (Zuo et al., 2020), for example, time embedding is through

$$[z(t_j)]_i = \begin{cases} \cos(t_j/10000^{\frac{i-1}{M}}) & \text{if } i \text{ is odd} \\ \sin(t_j/10000^{\frac{i}{M}}) & \text{if } i \text{ is even} \end{cases} \tag{3}$$

where t_j is a timestamp and M is the dimension of encoding. Time embedding and one-hot encoded types are combined to form the embedded input $\mathbf{X}$. For sequence $S = \{t_i, y_i\}_{i=1}^{L}$, time embedding $\mathbf{z_i}$ for each instance is specified in Equation 3 and for the entire sequence with length L, the embedding is $\mathbf{Z} \in R^{M \times L}$. Type embedding are through the product of a trainable embedding matrix $\mathbf{U} \in \mathbb{R}^{M \times K}$ and one hot encoded vectors $\mathbf{y_i}$'s for all type instances, i.e. $\mathbf{X} = (\mathbf{UY} + \mathbf{Z})^T$ where $\mathbf{Y} = [\mathbf{y_1}, \mathbf{y_2}, ..., \mathbf{y_L}]$. $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ are query, key and value matrix; they are linear transformations of $\mathbf{X}$, i.e. $\mathbf{Q} = \mathbf{XW}^Q$, $\mathbf{K} = \mathbf{XW}^K$, $\mathbf{V} = \mathbf{XW}^V$ where $\mathbf{W}^Q$, $\mathbf{W}^K$, $\mathbf{W}^V$ are trainable weights.

Attention output $\mathbf{C}$ is computed by the following:

$$\mathbf{C} = \text{softmax}(\frac{\mathbf{QK}^T}{\sqrt{M_k}})\mathbf{V} = \mathbf{A}_s\mathbf{V} \tag{4}$$

where $\mathbf{A}_s$ denotes attention score matrix. The output $\mathbf{C}$ is then fed into a pointwise feed forward neural network (FFN) (commonly with residual connection) to learn a high level representation of the sequence for modeling the conditional intensity function.

2.5. Further Remarks

Most attention and transformer based TPP models achieve state-of-the-art performance for predictive tasks, which demonstrates the effectiveness of these models on capturing complex dependencies among different events. To address the vague interpretation of attention (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019), we propose a model namely Influence-Aware Attention for Temporal Point Process (IAA-MTPP) which inherits the interpretability of neural probabilistic relation mining for MTPP and the power and efficiency of self-attention mechanism in transformers to model the dynamics. To our best knowledge, we are the first to bridge neural probabilistic modeling with transformer models for event data.

3. Model Formulation: IAA-MTPP

We propose a probabilistic attention model that helps identify influencing events among different event types. Our IAA-MTPP model consists of two parts: an attention encoder and a influence-aware attention decoder as shown in Figure 1. The former leverages attention-mechanism in transformer architecture to model the interaction of event types and the latter learns a dynamical model given an influence matrix. Let $\mathbf{A}$ be a binary random matrix whose entries encode how one event type influences another. By leveraging variational inference (Zhang et al., 2018), our model seeks to find a posterior distribution of influencing matrix $\mathbf{A}$ given sequence S, $p(\mathbf{A}|S)$ which can be approximated by a global variational distribution $q_\phi(\mathbf{A}|S)$ (parametrized by attention encoder with ϕ). For ease of computation, we use a prior which is of entry-wise product, i.e. $p(\mathbf{A}) = \prod_{i,j} p(\mathbf{A}_{ij} = 1)$. Consider a realization of $\mathbf{A}$, $A \in \{0,1\}^{D\times D}$; each entry (i,j) of A indicates the existence of influence from type j to i. Naturally, the evidence lower bound (ELBO) (Hoffman et al., 2013) can be expressed as:

$$\mathcal{L}(\theta,\phi;S) = \mathbb{E}_{q_\phi}[\log\frac{p(\mathbf{A})}{q_\phi(\mathbf{A}|S)}] + \mathbb{E}_{q_\phi}[\log p_\theta(S|\mathbf{A}=A)] \tag{5}$$

which is the sum of negative KL-divergence between $q_\phi(\mathbf{A}|S)$ and $p(\mathbf{A})$ and the expected conditional likelihood of observing the sequence for a given relation parametrized by attention decoder with ϕ. Different from neural relation inference literature (Kipf et al., 2018; Zhang and Yan, 2021), our model is capable of modeling the diagonal components A_{ii}'s since future event instances i can also attend to past instances of i. This "self-relation" turns out to be important in graphical event models (GEMs) (Gunawardana and Meek, 2016; Bhattacharjya et al., 2018; Gao et al., 2020), as it describes how past events of a particular type influence the occurrence of events of the same type in the future.

3.1. Encoder

The encoder is designed to infer the influencing events given a sequence S with length L. We follow a similar procedure as described in Transformer Hawkes Processes (Zuo et al., 2020) to apply B blocks of multi-head self-attention to obtain a high level representation of the sequence. The attention encoder then outputs an attention score matrix from B^{th} block namely $\mathbf{S}_{enc}^{(B)}$. Each entry (i,j) of $\mathbf{S}_{enc}^{(B)} \in R_{+}^{L\times L}$ signifies past influence of event instance j on event instance i through attention mechanism as captured similarly to $\mathbf{A}_s$ in Equation 4. Our approach in fact naturally embeds the following:

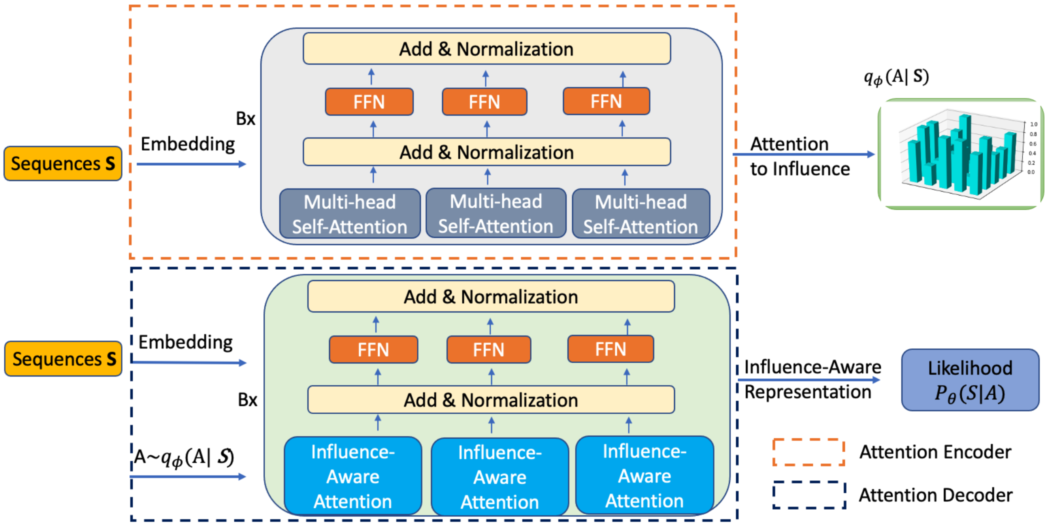

Figure 1: Neural architecture of IAA-MTPP. The attention encoder takes an event sequence S and outputs an approximate posterior $q_\phi(\mathbf{A}|S)$ of latent relation; the attention decoder takes the same sequence and a sample A from $q_\phi(\mathbf{A}|S)$ and outputs a influence-aware high level representation to be used for modeling the likelihood given influencing matrix A.

Theorem 1 *Any event instance i in a sequence S only attends to past instance j where $j \in \mathbb{L}$. j and i can be of the same type.*

Proof sketch: Consider imposing a strictly lower triangular attention mask on $\mathbf{A}_s$ in Equation 4, i.e. $\mathbf{A}_s \odot \mathbf{I}_l$ (entry-wise product), the attention score of any event in the past does not contribute to overall result in the matrix multiplication, and thus does not help future prediction in the FFN.

Attention-to-Influence. We summarize how event type j influences event type i based on our proposed Attention-to-Influence (A2I) formulation in the following to obtain an unscaled score matrix:

$$\text{A2I}(\mathbf{S}_{enc}^{(B)}) = \mathbf{P}^\intercal \mathbf{S}_{enc}^{(B)} \mathbf{P} \tag{6}$$

where $\mathbf{P} \in \{0,1\}^{L \times D}$ is a binary indicator matrix which specifies the type of events occurring in S for the L event instances [1]. In particular, each row of $\mathbf{P}$ is an D-dimensional one-hot vector. We emphasize the above equation captures the transformation from instance-instance interaction to type-type interaction in a concise and computationally efficient form. Furthermore, to be more general, we model the relation by a modified weighted score matrix:

$$\tilde{\mathbf{S}}_{enc}^{(B)} = \mathbf{S}_{enc}^{(B)} \odot \exp^{-\gamma \Delta T} \tag{7}$$

where ΔT is the inter-event times for events in sequence S, γ is a hyper-parameter which controls the decay of influence of event j on i. The Attention-to-Influence output A2I $(\tilde{\mathbf{S}}_{enc}^{(B)}) \in R_{+}^{D \times D}$ does not immediately specify a proper probability distribution as some entries may be well beyond

[1] In practice, we add an extra dimension 0 for padded events.

unity. We search for a class of functions that projects each nonnegative entry in the matrix to an element in the interval $[0, 1]$, i.e. f: $R_+ \rightarrow [0, 1]$ while maintaining the relative order. Many candidate functions are possible, however we select a shifted version of sigmoid function as it is most commonly used as non-linear activation function in neural network. Thus the variational distribution $q_\phi(\mathbf{A}|S)$ can be component-wise expressed as

$$q_\phi(\mathbf{A}_{ij} = 1|S) := 2\sigma(\text{A2I}(\tilde{\mathbf{S}}^{(B)}_{enc,ij})) - 1 \tag{8}$$

where σ specifies the sigmoid function: $\sigma(x) = \frac{1}{1+e^{-x}}$. Sampling from $q_\phi(\mathbf{A}|S)$ is straightforward, however since the distribution is discrete, Gumbel-Softmax (Maddison et al., 2017; Jang et al., 2016) is used to provide differentiable samples for back-propagation in our neural network.

3.2. Decoder

The goal of the attention decoder is to model the dynamics by leveraging sampled influence. In particular, we incorporate influence into the learning of the dynamics. We make the following simplifying assumptions which describe the relation between attention and influence matrix.

Assumption 1 *For any pair of events (i, j) in a sequence S, event instance i attends to event instance j if and only if event type j directly influences event type i, i.e. their influence value is nonzero. Event type j* ***directly*** *influences event type i if j is a parental process of i.*

The above assumption can be viewed as analogous to the one in GEMs (Didelez, 2008; Bhattacharjya et al., 2018; Yu et al., 2020): only parent processes j's can affect child process i; and other processes are considered to be locally independent from the child process.

Influence-to-Attention. We propose the learning of influence-aware attention in the following. For each event instance in the sequence S, we check the event type it influences given a sampled influence A, and construct influence indicator $\mathbf{I} \in \{0, 1\}^{L \times L}$:

$$\mathbf{I} = \mathbf{P}A\mathbf{P}^\intercal \tag{9}$$

where $\mathbf{P}$ is the binary indicator matrix same as in Equation 6. We combine an attention score matrix in the decoder $\mathbf{S}_{dec}$ with relation indicator $\mathbf{I}$ to derive the influence-aware attention score:

$$\tilde{\mathbf{S}}_{dec} = \mathbf{S}_{dec} \odot \mathbf{I} \tag{10}$$

A representation from the attention module is obtained directly by modifying Equation 4:

$$\tilde{\mathbf{S}} = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{M_k}})\mathbf{V} = \tilde{\mathbf{S}}_{dec}\mathbf{V} \tag{11}$$

For an h-head influence-aware attention module in the i^{th} block, we concatenate the above representation from each head and multiplied by a trainable weight matrix $\mathbf{W}^O$ to form a new representation: $\tilde{\mathbf{S}}^{(i)} = [\tilde{\mathbf{S}}^{(i)}_1, \tilde{\mathbf{S}}^{(i)}_2, ..., \tilde{\mathbf{S}}^{(i)}_h]\mathbf{W}^O$ In the B-block multi-head self-attention layer, the i^{th} block outputs a high level influence-aware representation $\tilde{\mathbf{H}}^{(i)}$ from FFN with trainable weights and biases $\mathbf{W}^{FC}_i$'s and $\mathbf{b}_i$'s:

$$\tilde{\mathbf{H}}^{(i)} = \text{ReLU}(\tilde{\mathbf{S}}^{(i)}\mathbf{W}^{FC}_1 + \mathbf{b}_1)\mathbf{W}^{FC}_2 + \mathbf{b}_2 \tag{12}$$

Each $\tilde{\mathbf{H}}^{(i)}$ is then sequentially fed into $i + 1^{th}$ block until the B^{th} block is reached.

Likelihood. The modeling of conditional intensity is similar to THP (Zuo et al., 2020), except the high-level event representation in our case is the B^{th} block influence-aware representation vector $\tilde{\mathbf{H}}^{(B)}$. The conditional intensity for type-k event takes the following form:

$$\lambda_k(t|H_t) = f_k(\alpha_k \frac{t - t_j}{t_j} + \mathbf{w_k}^T \tilde{\mathbf{H}}^{(B)}(j,:) + b_k) \tag{13}$$

where $H_t = \{(t_j, y_j) : t_j < t\}$ is the history up to t. α_k is a parameter modulates the importance of current influence, W_k and b_k are the weights and bias that characterize the historical and base influence. $f_k(x)$ is the softplus function. The total conditional intensity is modeled as the sum of conditional intensity in each dimension. Conditional log-likelihood $p_\theta(S|A)$ is computed based on Equation 1 for a given relation matrix. The computation of non-event likelihood is through Monte Carlo approximation (Robert and Casella, 2013). Our proposed attention-to-influence and influence-to-attention formulation captures the conversion between instance-wise interaction to type-wise influence in very concise forms. The power of this key insight makes transformer models applicable to many other structural/relational/causal inference problems in sequential data. Previous work has only used instance-wise attention. Using Theorem 1 and Assumption 1, we state the following result. (Please see the Appendix for proof sketches). It is worthy noting that our approach of learning a causal graph is implicit; yet the posterior quantifies the extent of influence which is similar to methods of using an infectivity matrix for Granger causality.

Theorem 2 *A learned $q_\phi(\mathbf{A}|S)$ from 1-block 1-head transformer architecture without residual connection, for a given threshold τ, encodes the graph structure of events where each nonzero entry is a parent process.*

3.3. Training

The variational training is performed jointly on the encoder and decoder by mini-batch stochastic gradient descent given N sequences $\{S_i\}_{i=1}^N$. The procedure is fully described by Algorithm 1. Note the computation of the ELBO $\mathcal{L}$ is straightforward, as the negative KL term for discrete distribution can be computed in closed-form and conditional likelihood is estimated as the average from a total of F samples:

$$\mathcal{L}(\theta, \phi) = \sum q_\phi(\mathbf{A}) \text{log} \frac{p(\mathbf{A})}{q_\phi(\mathbf{A}|S)} + \frac{1}{F} \sum_{f=1}^{F} \text{log} p_\theta(S|A_f) \tag{14}$$

4. Empirical Experiments

We follow standard implementation of transformer architecture for our model and give more details in Appendix. In addition, we use uniform ($p(\mathbf{A}_{ij} = 1) = 0.5$) and sparse ($p(\mathbf{A}_{ij} = 1) = 0.2$) prior namely IAA-MTPP-unif and IAA-MTPP-sparse in our experiments.

Baselines. We compare our models against state-of-the-art models for learning Granger-causal infectivity matrix. Training of baselines follows the recommended setting.

Hawkes-based models: Hawkes process with exponential kernel (Hawkes-exp) and ADM4. Standard implementation of Hawkes-exp and ADM4 are available from the tick module [2].

[2] https://x-datainitiative.github.io/tick/modules/hawkes.html

Algorithm 1 Training Procedure for IAA-MTPP

```
Input: Given sequences $S = \{S_i\}_{i=1}^N$, sample size $F$, prior $p(\mathbf{A})$, mini-batch size $b$, training epochs $K$
Output: Approximate posterior $q_\phi(\mathbf{A}|S)$
for $epoch \leftarrow 1$ to $K$ do
    for $iteration \leftarrow 1$ to $\lceil \frac{M}{b} \rceil$ do
        Sample a batch of sequences $S'$ from $S$
        Compute $q_\phi(\mathbf{A}|S)$ via Attention-to-Influence
        Sample $A, ..., A_F \sim q_\phi(\mathbf{A}|S)$
        Compute log$p_\theta(S'|A_f)$ via Influence-to-Attention
        Compute ELBO $\mathcal{L}$: $\sum q_\phi(\mathbf{A}|S')\log\frac{p(\mathbf{A})}{q_\phi(\mathbf{A}|S')} + \frac{1}{F}\sum_{f=1}^F \log p_\theta(S'|A_f)$
        Back-propagate with gradient $\nabla_{\theta,\phi}\mathcal{L}$
        Update parameters of network $\theta, \phi$
    end
end
Return: $q_\phi(\mathbf{A}|S)$
```

RNN-based models: CAUSE and NRI-TPP with sparse prior. The implementation for CAUSE is publicly available [3]. While NRI-TPP codes are not provided, we implement according to Zhang and Yan (2021).

Transformer-based models: Self-attentive Hawkes process (ATTN-SAHP) and transformer Hawkes Process (ATTN-THP). Both implementations are available [4] [5]. The inferred infectivity matrix is computed through Attention-to-Influence transformation for transformer-based baselines as post-processing.

Evaluation Metrics. We threshold at each percentile of entries in the learned influence matrix and compare against the binarized ground truth to compute the corresponding F1-scores and select the maximum for final comparison.

4.1. Synthetic Datasets

We perform two sets of synthetic experiments based on Hawkes process and graphical event models; each set contains 5 simulations. More details around data generation can be found in Appendix A.

MHP-Exp. We generate synthetic datasets of 100 dimensions from Hawkes processes according to previous studies (Zhang and Yan, 2021; Zhang et al., 2020b). The exponential kernel $g(t) = \beta\exp(-\beta t)$ is used to generate 2,000 sequences on the time interval [0, 20], with $\beta = 2.5$. A binary ground-truth causality matrix was constructed from the infectivity matrix.

PGEM. We use a proximal graphical event model (PGEM) (Bhattacharjya et al., 2018) generator to generate 10 sequences in time horizon [0, 1000]; please see the Appendix for details about the ground truth model (Bhattacharjya et al., 2022). Influencing sets for each event type are its parental nodes in the underlying graph, which provide a binary ground-truth matrix.

[3] https://github.com/razhangwei/CAUSE

[4] https://github.com/QiangAIResearcher/sahp_repo

[5] https://github.com/SimiaoZuo/Transformer-Hawkes-Process

Results. Table 1 shows that our model IAA-MTPP-sparse achieves the best results among all baselines in recovering the latent graph as indicated by high F1 scores for both experiments. Specifically, sparse prior performs better than uniform prior; this echos the sparsity of ground truth matrix. The superior performance is partially due to the use of multi-head self-attention and the modeling of the conditional intensity with Equation 13 being effective for capturing the dynamics. While CAUSE is a close competing model in the synthetic experiments, it is much worse in learning influencing (sub)sets from real benchmarks as we show in the following.

Table 1: F1 scores and their standard deviation on two datasets. Best results are shown in bold. Second best ones are in italics.

Model/Dataset	MHP-Exp	PGEM
Hawkes-exp	0.76(0.01)	0.51(0.05)
ADM4	0.85(0.00)	0.49(0.09)
ATTN-SAHP	0.69(0.01)	0.49(0.10)
ATTN-THP	0.82(0.00)	0.50(0.06)
NRI-TPP-sparse	0.89(0.00)	*0.52(0.03)*
CAUSE	**0.99(0.00)**	0.48(0.00)
IAA-MTPP-unif	0.89(0.00)	**0.53(0.02)**
IAA-MTPP-sparse	*0.90(0.00)*	*0.52(0.05)*

4.2. Real Applications

We consider 2 real world event datasets in the domains of healthcare and decentralized finance. A brief description of the datasets follows:

Diabetes contains daily events for meal intake, exercise activity, insulin dosage and changes in blood glucose measurements for 67 diabetes patients [6]. A partial relation is given by domain experts (Acharya, 2014).

DEFI contains user trading events from Aave website[7]. We filter out irrelevant features so that our data only consists of timestamp, (trans)action types for each event.

While our focus is on accurately learning the latent influence matrix, ground truth from real data is often inaccessible. We select parts of the learned influence matrix that corresponds to the partially known structure on **Diabetes** and evaluate by F1-score, following previous studies (Bhattacharjya et al., 2021; Gao et al., 2021). Furthermore, we provide a qualitative interpretation of the learned interactions on **DEFI**.

Results on Diabetes. IAA-MTPP-unif achieves highest F1-score as shown in Table 2. A sparse prior in this dataset may not be appropriate because relations among events in health and disease system are more complex in nature and are more likely to result in a non-sparse interaction network; and certain deep interactions may escape domain experts (our models are trained to learn all interactions, but only partially are evaluated). ATTN-SAHP can be considered to use a uniform prior since it incorporates no prior knowledge of the relation in the learning. Then noticeably, uniform priors are much better than their sparse counterparts. It is worth noting that neural models are much

[6] https://archive.ics.uci.edu/ml/datasets/diabetes

[7] aave.com

better at learning the latent relation than classical approaches on this real dataset. Additionally, even with sparse prior, IAA-MTPP still roughly improves performance by 10% over its close competitor NRI-TPP.

Table 2: F1 score on Diabetes. Best Result is in bold.Second best is in italics.

Hawkes-exp	ADM4	ATTN-SAHP	ATTN-THP	NRI-TPP-sparse	CAUSE	IAA-MTPP-unif	IAA-MTPP-sparse
0.200	0.286	*0.389*	0.345	0.345	0.272	**0.419**	0.375

Results on DEFI. Figure 2 shows the extracted interaction for DEFI. Each square represents an entry (i, j) in the binarized relation matrix and indicates whether event j influences event i. Beige indicates existence of such influence while black indicates nonexistence. Our model captures key interactions within DEFI lending. Aave is over-collateralized, so users must ”deposit” coins before they can ”borrow”. We see ”deposits” influence ”redeems” of borrowed coins. To ”repay” a loan, users must ”deposit” and ”borrow” so both influence ”repays” of loans. If users no longer have sufficient deposits for collateral (e.g from ”redeem” of the loan), the loan goes into ”liquidation” and the collateral is used to payback the principal. ”Swap” changes the cryptocurrency used in deposits. The fact that ”swap” influences ”redeems”, ”deposits”, ”repays” and ”borrows” provides insights into how users use ”swaps”, an intriguing finding since there is no analog to swaps in traditional banking. IAA-MTPP-sparse is able to unravel influences and patterns in complex DEFI transaction protocols without any prior knowledge. Thus our proposed models could be very valuable to understand usage and predict transactions in the rapidly evolving space of both existing and future DEFI protocols. Future work is needed to enhance IAA-MTPP models to incorporate additional information that influence transactions such as coin-types, transaction amounts, transaction fees, interest rates, and transaction prices.

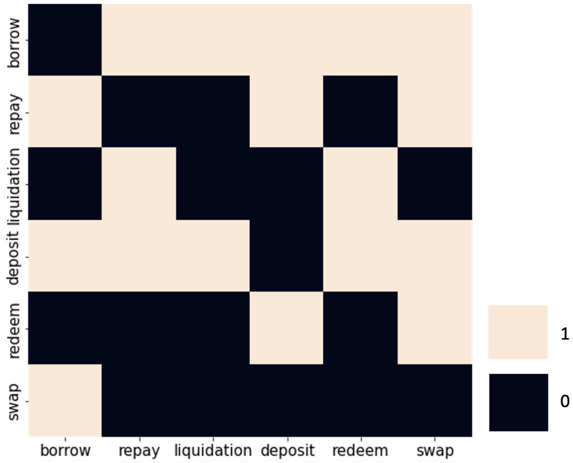

Figure 2: Possible interactions among 6 event types in DEFI inferred by IAA-MTPP-sparse.

4.3. Ablation Studies

4.3.1. Ablation I: Efficiency vs. NRI-TPP

We generate higher dimension version of **MHP-Exp** datasets accordingly, and test the efficiency of two models: IAA-MTPP and NRI-TPP with sparse prior. Training is on Google Colab with GPU and high-RAM, more specifically with Tesla P100-PCIE-16GB and 52GB RAM. Both models are trained with same magnitude of parameters for 10 epochs for a fair comparison. We evaluate the accuracy of relation inference in higher dimensions by F1-score. As shown in Table 3, while the accuracy of our attention model is higher for all dimensions compared to NRI-TPP-sparse, our model is at least 10 times more efficient. This agrees with the general trend of using parallelizable structure rather than recurrent model for faster and potentially more accurate learning. The possible reason for fast convergence of our model is that transformer component in IAA-MTPP is efficient in both model size and training speed as shown Appendix A.1 in Zuo et al. (2020). Furthermore, in our study we used a very small model for all experiments (precisely 50210 parameters, only half of the smallest setting of THP model on twitter dataset).

Table 3: Comparing IAA-MTPP and NRI-TPP. IAA-MTPP is faster and more accurate.

	Accuracy (F1)		Efficiency (seconds)	
Dim	NRI-TPP	IAA-MTPP	NRI-TPP	IAA-MTPP
100	0.879	**0.893**	1576	**105**
200	0.781	**0.783**	5914	**208**
300	0.774	**0.777**	15267	**383**
400	0.651	**0.687**	22636	**560**

4.3.2. ABLATION II: EFFECT OF PRIOR VS. SAMPLE SIZE

We generate **MHP-Exp** data of differing sample sizes according to the procedure described in section 4.1 and test the effect of sample size on the performance of IAA-MTPP models with the two priors. Figure 3 depicts the general trend: the more samples used for training, the more accurate the recovered relation is as indicated by higher F1 scores. With fewer samples, IAA-MTPP-sparse outperforms IAA-MTPP-unif; with higher samples, both models become equally good. This result is more-or-less consistent with general understanding of the role of priors in maximum-a-posteriori inference: it becomes less crucial as sample size gets larger, and data eventually overwhelms the prior (Murphy, 2012).

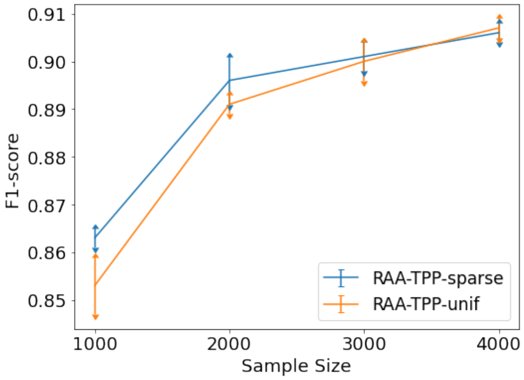

Figure 3: Structure discovery with two different priors and varying sample sizes. Larger samples boost performance.

5. Conclusion

We have proposed a contemporary neural relational inference model for multivariate temporal point processes and demonstrated its value for various applications including diabetes and cryptocurrency (DEFI) applications. To the best of our knowledge, our model reflects a novel integration of a probabilistic aspect into a typical deterministic attention mechanism, resulting a type-to-type attention mechanism (different from existing instance-to-instance attentions). Our work enjoys the efficiency and accuracy that attention models provide as well as the interpretability that graphical models offer. Importantly, the proposed model is potentially capable of capturing Granger-causal relations among events due to the underlying assumptions in our framework; this aspect is also relevant for causal inference between event pairs from event datasets (Gao et al., 2021). Furthermore, the use of attention mechanism in our model is quite general which makes it very flexible to be adapted to other types of cutting-edge attention models. For example, one can equip a memory-efficient form of transformer (Beltagy et al., 2020; Kitaev et al., 2019) for estimating influences in future work while retaining or even improving upon the accuracy and time-efficiency of the proposed model.

6. Acknowledgments

This work was supported by the Rensselaer-IBM AI Research Collaboration (http://airc.rpi.edu), part of the IBM AI Horizons Network (http://ibm.biz/AIHorizons)

We would like to thank four anonymous reviewers for helpful feedback.

References

Saurav Acharya. *Causal Modeling and Prediction over Event Streams*. The University of Vermont and State Agricultural College, 2014.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Debarun Bhattacharjya, Dharmashankar Subramanian, and Tian Gao. Proximal graphical event models. *Advances in Neural Information Processing Systems*, 31, 2018.

Debarun Bhattacharjya, Tian Gao, Nicholas Mattei, and Dharmashankar Subramanian. Cause-effect association between event pairs in event datasets. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 1202–1208, 2021.

Debarun Bhattacharjya, Karthikeyan Shanmugam, Tian Gao, and Dharmashankar Subramanian. Process independence testing in proximal graphical event models. In *Proceedings of the Conference on Causal Learning and Reasoning*, volume 177, pages 144–161. PMLR, 2022.

Debarun Bhattacharjya, Tian Gao, Shankar Subramaniam, and Xiao Shou. Score-based learning of graphical event models with background knowledge augmentation. In *AAAI Conference on Artificial Intelligence*, 2023.

Daryl J Daley and D Vere Jones. *An Introduction to the Theory of Point Processes: Elementary Theory of Point Processes*. Springer, 2003.

Vanessa Didelez. Graphical models for marked point processes based on local independence. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):245–264, 2008.

Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le 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.

Michael Eichler, Rainer Dahlhaus, and Johannes Dueck. Graphical modeling for multivariate Hawkes processes with nonparametric link functions. *Journal of Time Series Analysis*, 38(2): 225–242, 2017.

Tian Gao, Dharmashankar Subramanian, Karthikeyan Shanmugam, Debarun Bhattacharjya, and Nicholas Mattei. A multi-channel neural graphical event model with negative evidence. In *AAAI Conference on Artificial Intelligence*, volume 34, pages 3946–3953, 2020.

Tian Gao, Dharmashankar Subramanian, Debarun Bhattacharjya, Xiao Shou, Nicholas Mattei, and Kristin Bennett. Causal inference for event pairs in multivariate point processes. *Advances in Neural Information Processing Systems*, 34, 2021.

Yulong Gu. Attentive neural point processes for event forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7592–7600, 2021.

Asela Gunawardana and Chris Meek. Universal models of multivariate temporal point processes. In *Artificial Intelligence and Statistics*, pages 556–563. PMLR, 2016.

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

Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 2013.

Sarthak Jain and Byron C Wallace. Attention is not explanation. *arXiv preprint arXiv:1902.10186*, 2019.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, pages 2688–2697. PMLR, 2018.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2019.

Shuang Li, Shuai Xiao, Shixiang Zhu, Nan Du, Yao Xie, and Le Song. Learning temporal point processes via reinforcement learning. *arXiv preprint arXiv:1811.05016*, 2018.

Scott Linderman and Ryan Adams. Discovering latent network structure in point process data. In *International conference on machine learning*, pages 1413–1421. PMLR, 2014.

Yanchi Liu, Tan Yan, and Haifeng Chen. Exploiting graph regularized multi-dimensional Hawkes processes for modeling events with spatio-temporal characteristics. In *International Joint Conference on Artificial Intelligence*, pages 2475–2482, 2018.

Chris Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *Proceedings of the international conference on learning Representations*. International Conference on Learning Representations, 2017.

Hongyuan Mei and Jason Eisner. The neural Hawkes process: A neurally self-modulating multivariate point process. *arXiv preprint arXiv:1612.09328*, 2016.

Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

Takahiro Omi, Naonori Ueda, and Kazuyuki Aihara. Fully neural network based model for general temporal point processes. *arXiv preprint arXiv:1905.09690*, 2019.

Christian Robert and George Casella. *Monte Carlo Statistical Methods*. Springer, New York, 2013.

Farnood Salehi, William Trouleau, Matthias Grossglauser, and Patrick Thiran. Learning Hawkes processes from a handful of events. *Advances in Neural Information Processing Systems*, 32, 2019.

Karishma Sharma, Yizhou Zhang, Emilio Ferrara, and Yan Liu. Identifying coordinated accounts on social media through hidden influence and group behaviours. *arXiv preprint arXiv:2008.11308*, 2020.

Oleksandr Shchur, Marin Biloš, and Stephan Günnemann. Intensity-free learning of temporal point processes. *arXiv preprint arXiv:1909.12127*, 2019.

Xiao Shou, Tian Gao, Shankar Subramaniam, Debarun Bhattacharjya, and Kristin Bennett. Concurrent multi-label prediction in event streams. In *AAAI Conference on Artificial Intelligence*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. *arXiv preprint arXiv:1908.04626*, 2019.

Christo Wilson, Bryce Boe, Alessandra Sala, Krishna PN Puttaswamy, and Ben Y Zhao. User interactions in social networks and their implications. In *Proceedings of the 4th ACM European Conference on Computer Systems*, pages 205–218, 2009.

Shuai Xiao, Mehrdad Farajtabar, Xiaojing Ye, Junchi Yan, Le Song, and Hongyuan Zha. Wasserstein learning of deep generative point process models. *arXiv preprint arXiv:1705.08051*, 2017a.

Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen M Chu. Modeling the intensity function of point process via recurrent neural networks. In *AAAI Conference on Artificial Intelligence*, pages 1597–1603, 2017b.

Shuai Xiao, Junchi Yan, Mehrdad Farajtabar, Le Song, Xiaokang Yang, and Hongyuan 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.

Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. Learning Granger causality for Hawkes processes. In *International Conference on Machine Learning*, pages 1717–1726. PMLR, 2016.

Junchi Yan, Xin Liu, Liangliang Shi, Changsheng Li, and Hongyuan Zha. Improving maximum likelihood estimation of temporal point process via discriminative and adversarial learning. In *International Joint Conference on Artificial Intelligence*, pages 2948–2954, 2018.

Xiufan Yu, Karthikeyan Shanmugam, Debarun Bhattacharjya, Tian Gao, Dharmashankar Subramanian, and Lingzhou Xue. Hawkesian graphical event models. In *International Conference on Probabilistic Graphical Models*, pages 569–580. PMLR, 2020.

Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. Advances in variational inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):2008–2026, 2018.

Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive Hawkes process. In *International Conference on Machine Learning*, pages 11183–11193. PMLR, 2020a.

Wei Zhang, Thomas Panum, Somesh Jha, Prasad Chalasani, and David Page. CAUSE: Learning Granger causality from event sequences using attribution methods. In *International Conference on Machine Learning*, pages 11235–11245. PMLR, 2020b.

Yunhao Zhang and Junchi Yan. Neural relation inference for multi-dimensional temporal point processes via message passing graph. In *International Joint Conference on Artificial Intelligence*, pages 3406–3412, 2021.

Ke Zhou, Hongyuan Zha, and Le Song. Learning social infectivity in sparse low-rank networks using multi-dimensional Hawkes processes. In *Artificial Intelligence and Statistics*, pages 641–649. PMLR, 2013.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer Hawkes process. In *International Conference on Machine Learning*, pages 11692–11702. PMLR, 2020.

Appendix A. Synthetic Data Generation

Hawkes-exp. We set the true infectivity matrix $\mathbf{W} = \mathbf{U}\mathbf{V}^T$, where $\mathbf{U}$ and $\mathbf{V}$ are initialized as zero matrices of dimension 100×9. We sample from a uniform distribution for certain entries, i.e. $\mathbf{U}_{10(i-1)+1:10(i+1),i} \sim \text{Uniform}(0.1, 0.2)$ for $i = 1, ..., 9$; same sampling is performed on $\mathbf{V}$. The spectral radius of $\mathbf{W}$ is scaled to 0.8. Baseline intensities μ_i's are also drawn from uniform distribution, $\mu_i \sim \text{Uniform}(0, 0.02)$, for $i = 1, ..., 100$. $\mathbf{W}$, μ and the following specified kernels are used to generate synthetic datasets. The generated events in the dataset are 186700. To mimic real world data, We add noise to every timestamp t_i, i.e. $\tilde{t_i} = t_i + \epsilon_i \sim \mathbf{N}(0, 0.01)$. We accept it if $\tilde{t_i} > 0$; otherwise we repeat sampling ϵ_i until $\tilde{t_i} > 0$. All timestamps are rescaled to $[0, 1]$.

PGEM. We describe the windows and conditional intensity parameters for the PGEM generator. They are listed in the following format: windows corresponding to the parents are listed in the same order as parents, and binary vectors are used to indicate parental states in the same order as listed parents as well. Figure 4 shows the graph structure of the ground truth of event dependence.

- Parents = {A: [A], B: [A, C], C: [C], D: [A, E], E: [C, D]}
- Windows = {A: [15], B: [30, 30], C: [15], D: [15, 30], E: [15, 30]}
- Lambdas = { A: {[0]: 0.1, [1]: 0.3}, B: {[0,0]: 0.01, [0,1]: 0.05, [1,0]: 0.1, [1,1]: 0.5}, C: {[0]: 0.2, [1]: 0.4}, D: {[0, 0]: 0.05, [0, 1]: 0.02, [1, 0]: 0.2, [1, 1]: 0.1}, E: {[0, 0]: 0.1, [0, 1]: 0.01, [1, 0]: 0.3, [1, 1]: 0.1} }

Appendix B. Real Datasets

We introduce 3 more datasets for evaluating model fitting by log-likelihood in the following. A summary of datasets are described in Table 4.

MIMIC-II includes patient-level electronic health records with clinical visits in Intensive Care Unit for 7 years. Each patient has a sequence of hospital visit events, and each event records its time stamp and diagnosis, which serves as the event label.

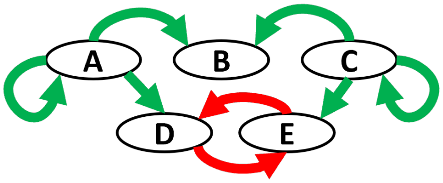

Figure 4: Graph of events from PGEM. Green (red) arcs indicate amplification (inhibition) effects, i.e. when a parent increases (decreases) a child's conditional intensity rate.

Stack Overflow is a question-answering website where users get engaged in the online community. Each user receives a sequence of badges over a two-year period.

Cosmetics is from Kaggle's e-Commerce Events History in Cosmetics Shop [8]. The original file consists of 20M users and their 5-month shopping behavior data from a medium cosmetics online store from October 2019 to February 2020. For each transaction related event, there are 4 categories $view$, $cart$, $remove\ from\ cart$ and $purchase$. We filter out sequences with missing data or length shorter than 10 events.

Table 4: A summary of datasets used in our experiments. Train seq: number of sequences in training; test seq: number of sequences in test data; train events: number of events in training data; test events: number of events in test data. A Github repository with DEFI data with full description will be released on publication.

Data	train seqs	test seqs	train events	test events	classes
MIMIC II	520	130	1915	504	75
DEFI	4626	1156	159319	38046	6
Cosmetics	3659	914	198977	50963	4
Stack Overflow	5307	1326	386316	94098	22
Diabetes	54	13	20582	5974	12

Table 5: Evaluation on four real datasets by LL/event. RAA-TPP-sparse achieves the best results on DEFI, Cosmetics, and Stack Overflow; RAA-TPP-unif outperforms others on MIMIC II. * indicates ADM4 fails in completing the run.

Model/Data	Hawkes-exp	ADM4	ATTN-SAHP	NRI-TPP-sparse	RAA-TPP-unif	RAA-TPP-sparse
MIMIC II	0.004	N/A*	0.740	0.693	**0.749**	0.744
DEFI	0.072	0.000	2.345	3.018	5.327	**5.328**
Cosmetics	2.160	0.000	0.156	3.488	3.851	**3.852**
Stack Overflow	0.006	0.000	2.975	2.333	4.482	**4.491**

[8] https://www.kaggle.com/mkechinov/ecommerce-events-history-in-cosmetics-shop

Appendix C. Implementation Details

We train IAA-MTPP with standard Pytorch implementation of multi-head attention module in the encoder and our influence-aware attention in the decoder. The attention module in the encoder and decoder is adapted from Transformer Hawkes Processes [9], and we modify accordingly by designing Influence-to-Attention and Attention-to-Influence components to make a unified variational inference model. The experiments are mostly performed on a Rensselaer IDEA Cluster Node 2 [10] and we train with CPUs; except in ablation study (**Efficiency vs. NRI-TPP**) we run on Google Colab with High RAM setting.

Parameter tuning. The parameters are selected based on the overall ELBO value in equation 14 from the training subset. Empirically we find the following neural architecture results in the best performing model and used in all experiments: number of heads = 6, number of layers = 4, d_{model} (the dimensionality of the representations used as input to the multi-head attention) = 30, d_{inner} (the dimensionality of the hidden layer of the feed forward neural network) = 16, d_v(the dimensionality of the linearly projected values) = 6, d_k (the dimensionality of the linearly projected keys) = 6, learning rate = 0.002, and number of samples = 2, decay rate (γ) = 35 (in equation 7).

Appendix D. Sketch Proof of Theorem 2

Proof sketch: For a simple transformer architecture, i.e., 1-block 1-head without residual connection, an attention score matrix $\mathbf{A}_s$ in equation 4, by Assumption 1 and 2, contains nonzero entries wherever a parental event for a specific event occurs in the past history, and zero entries for non parental events. Entries of non-parental events remain the same order through sigmoidal transformation in equation 8, since (shifted) sigmoid is monotonically increasing. Our Attention-to-Influence procedure is to aggregate such entries by summation. There must exist a threshold τ that separates aggregated scores of nonparental events (effectively zeros) and those of parental events, because positive numbers are dense. Hence $q_\phi(A_{ij}|S)$ effectively learns decomposed parents for each event as we assume pairwise influence.

[9] https://github.com/SimiaoZuo/Transformer-Hawkes-Process

[10] https://idea.rpi.edu/IDEA_Cluster_Access