Graphical Event Models

Debarun Bhattacharjya[1] Tian Gao[1]
Søren Wengel Mogensen[2] Xiao Shou[3]

[1]IBM T. J. Watson Research Center, NY, USA

[2]Lund University, Sweden

[3]Rensselaer Polytechnic Institute, NY, USA

IJCAI 2023 Tutorial, August 20, 2023

{*debarunb,tgao*}*@us.ibm.com*, *soren.wengel_mogensen@control.lth.se*, *shoux@rpi.edu*

Tutorial Link

https://sites.google.com/view/tiangao/tutorials

Introduction

Events are Everywhere!

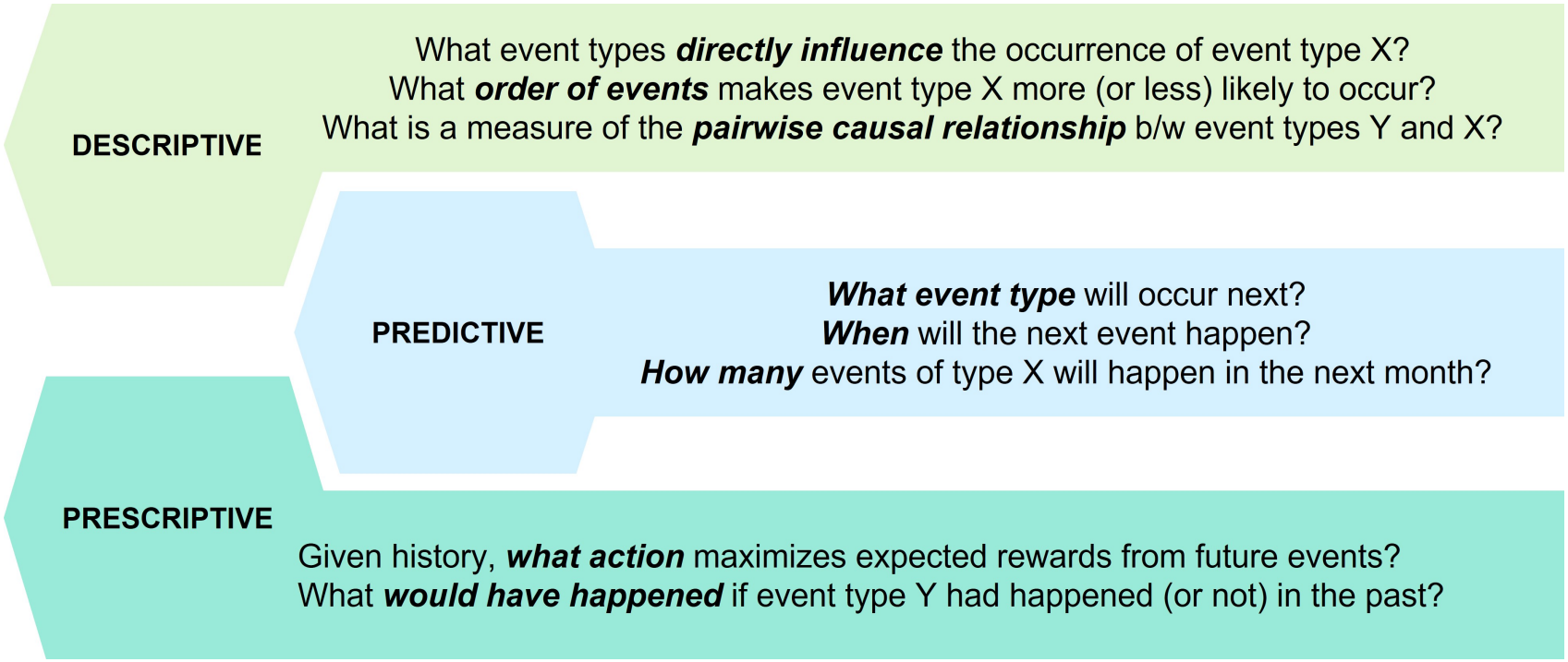
Event Datasets

- web logs
- customer transactions
- financial events
- insurance claims
- brain activity neural spikes
- social network messages
- ...

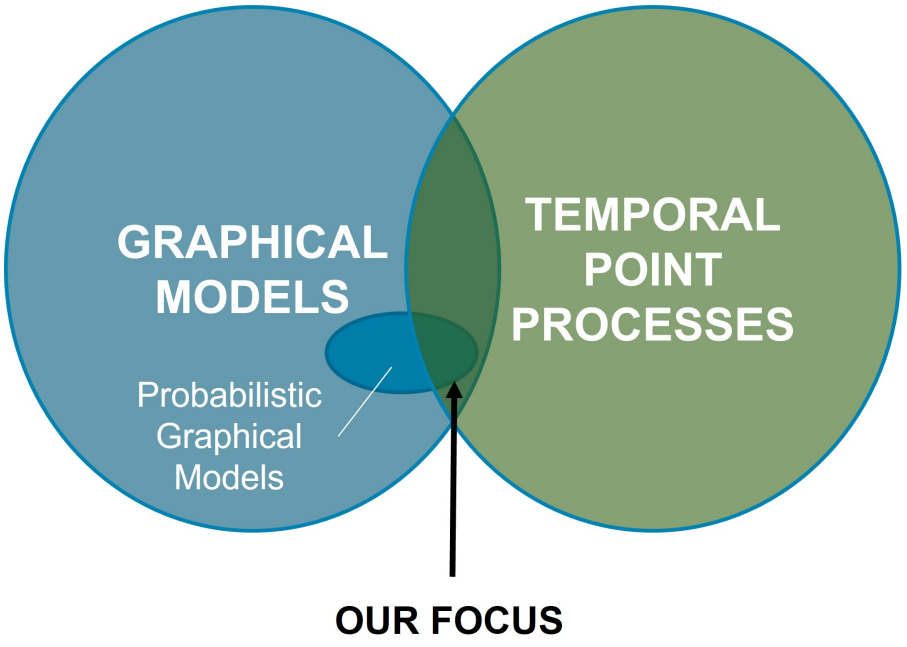
Applications

- preventive maintenance
- health outcome prediction
- scientific discovery
- knowledge discovery
- information diffusion
- recommendation systems
- ...

Motivating Analyses



Scope



What is covered

- Foundations of graphical models of TPPs
- Learning graphical models of TPPs
- …

What is not covered

- TPPs involving graph data
- Graphical models of other stochastic processes
- Continuous-time reinforcement learning
- …

Agenda

- Introduction
- Background on Temporal Point Processes
- Graphs and Temporal Point Processes
- Parametric Graphical Event Models
- —— BREAK ——
- Neural Temporal Point Processes
- Causality in Temporal Point Processes
- Conclusion
- —— DISCUSSION ——

Background on TPPs

Event Data

An *event data set* is a collection $\mathbf{D} = \{(l_k, t_k)\}_{k=1}^{N}$ where:

- t_k is the *event time* of the k^{th} event, $t_{k_0} \leq t_{k_1}$ for $k_0 \leq k_1$
- l_k is the *label* of the k^{th} event, $l_k \in \mathcal{L} = \{1, \ldots, M\}$

We write $\{(L_k, T_k)\}_{k \geq 1}$ for the corresponding random variables.

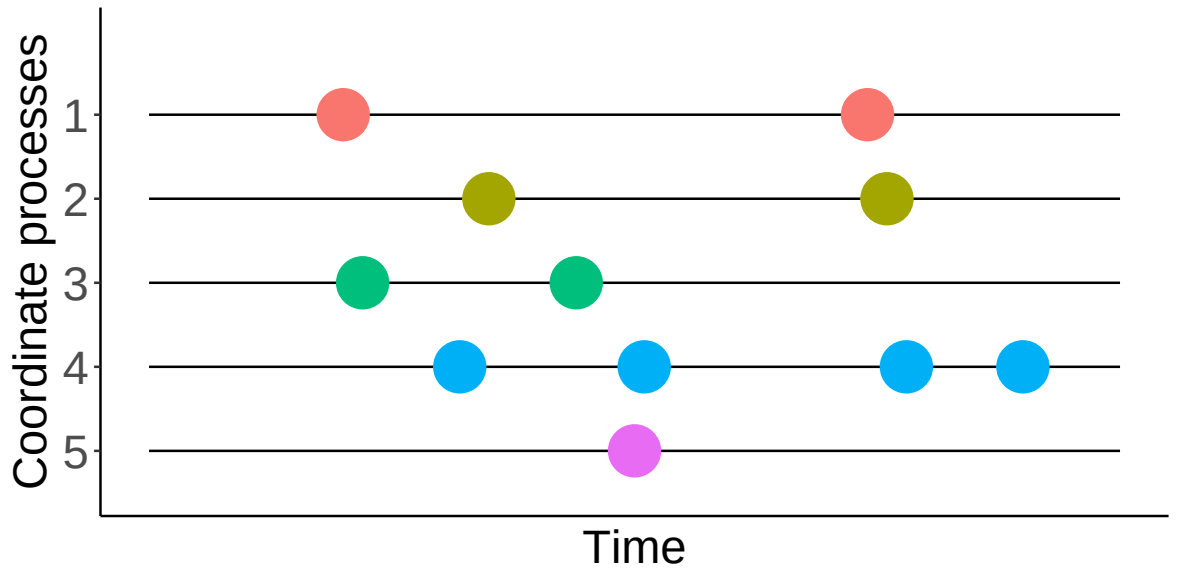
Examples

Illustration on a single time line of data set with three coordinate processes/event types ($M = 3$):

$$D = \{(A, 2), (B, 3), (C, 4), (B, 6), \ldots\}.$$

Examples

Illustration of event data set with five coordinate processes/event types ($M = 5$) with separate vertical placement:



Temporal Point Processes (TPPs)

Event data sets can be modelled using *(temporal) point processes*. A (multivariate) point process, $X_t = (X_t^1, \ldots, X_t^M)$, is a stochastic process where:

$$X_t^i = \sum_{k \geq 1, L_k = i} \delta(t - T_k).$$

We identify each $i \in \mathcal{L}$ with a *coordinate process*, X_i. One can specify a distribution of the point process using the *conditional intensities*, λ_t^i. These are themselves stochastic processes:

$$\lambda_t^i = \lim_{h \downarrow 0} P(\text{an event of type } i \text{ occurs in } (t, t+h] \mid \mathcal{H}_t)$$

where $\mathcal{H}_t$ is a σ-algebra generated by the evolution of the process until time t.

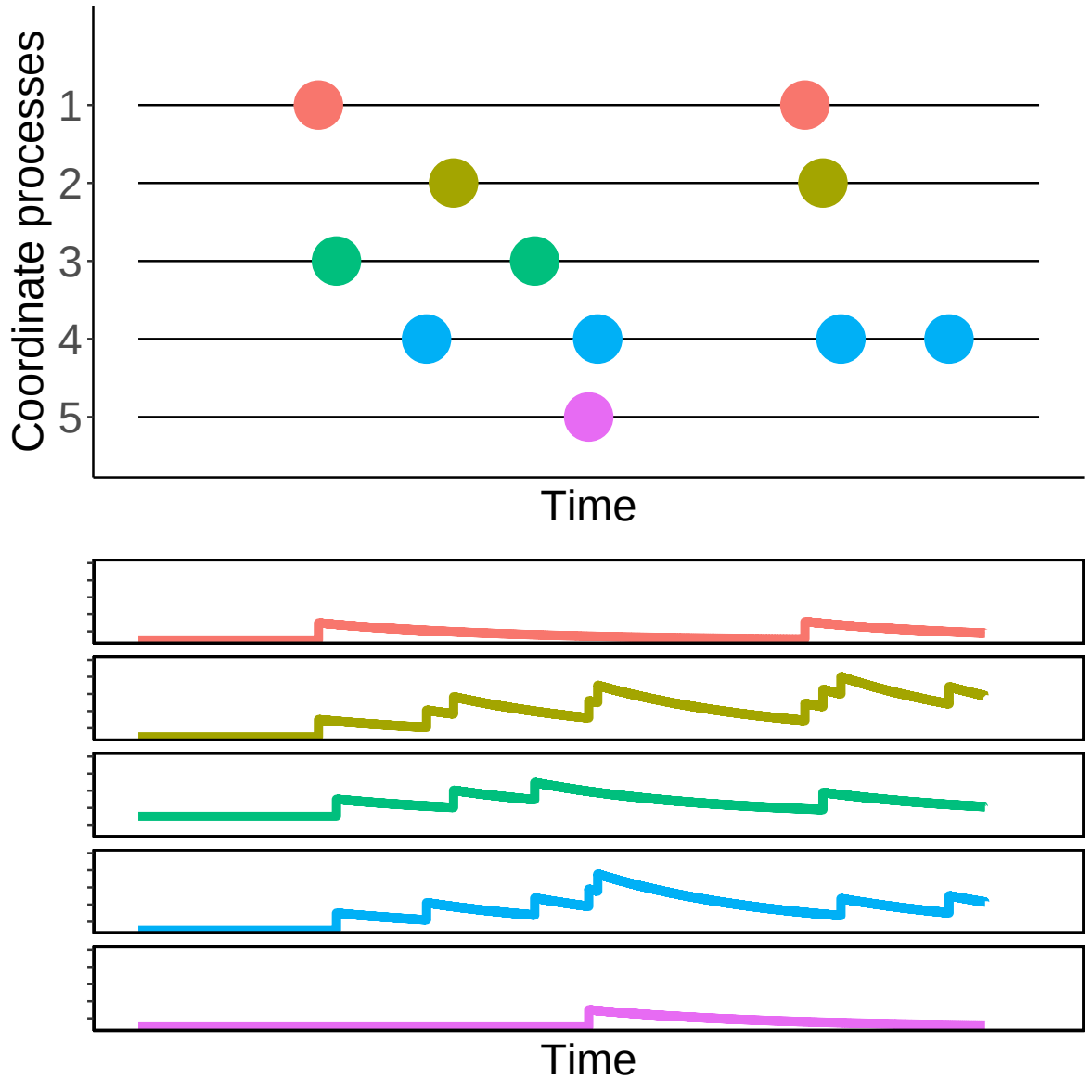
Conditional Intensities (Hawkes Process)

An example of how to specify the distribution of a point process using the conditional intensities is the *(linear) Hawkes process*:

$$\lambda_t^j = \mu_j + \sum_{i \in \mathcal{L}} \left(\sum_{\substack{k: T_k < t, \\ L_k = i}} f^{ji}(t - T_k) \right)$$

where μ_i are non-neg. constants and f^{ji} are non-neg. functions.

Conditional Intensities (Hawkes Process)

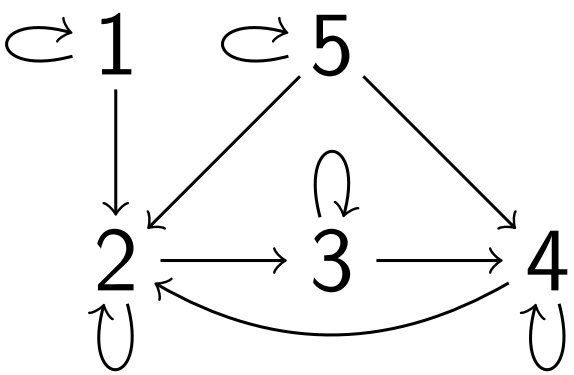


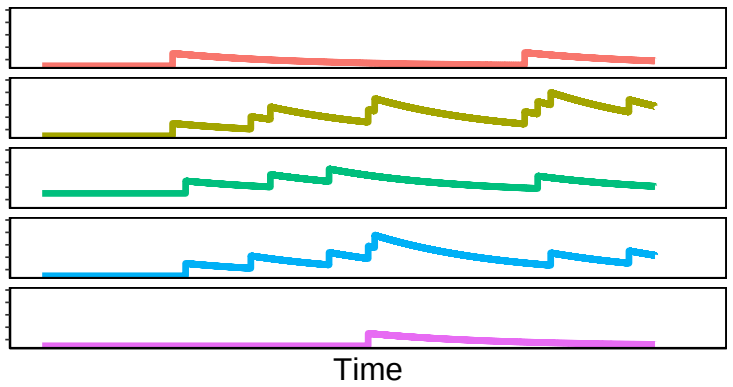
Graphs and TPPs

Graphs

We will use a *directed graph*, $\mathcal{G} = (\mathcal{L}, \mathcal{E})$, to represent sparsity in how coordinate processes influence each other.

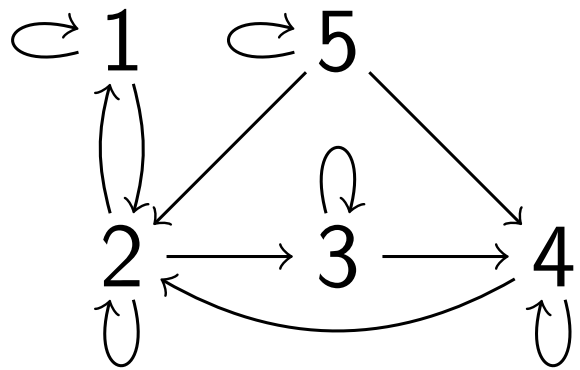
- $\mathcal{L}$ is the node set (same as the label set/index set of the coordinate processes).
- $\mathcal{E}$ is a set of edges, i.e. ordered pairs, (i, j), such that $i, j \in \mathcal{L}$.





Graphs

A *walk* is an alternating sequence of adjacent nodes and edges.
A *path* is a walk such that no node is repeated.



- $1 \leftarrow 2 \rightarrow 3 \rightarrow 3$ is a walk
- $1 \leftarrow 2 \rightarrow 3$ is a path

As there may be multiple edges between a pair of nodes, a sequence of nodes does not define unique walk in itself.

Local Independence

Definition

Let $A, B, C \subseteq \mathcal{L}$. We say that B is *locally independent* of A given C, and write $A \not\to_\lambda B \mid C$ if for all $i \in B$, $E(\lambda_t^i \mid \sigma(X_{0:t}^{A \cup C}))$ does not depend on tracks in A.

Local independence has been studied by, e.g., Schweder (1970), Aalen (1987), and Didelez (2008). One can also define local independence in other classes of processes, e.g., Commenges and Gégout-Petit (2009) and Mogensen, Malinsky, and Hansen (2018). It is similar to Granger causality in (discrete-time) time series.

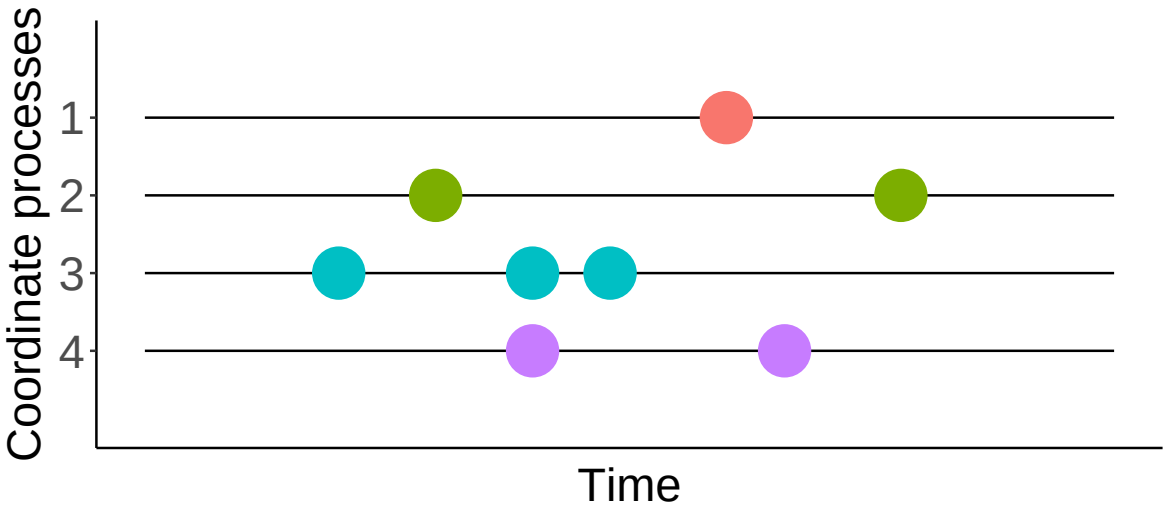
Local independence is a *ternary relation*, analogous to conditional independence. However, local independence is *asymmetric*,

$$A \not\to_\lambda B \mid C \not\Rightarrow B \not\to_\lambda A \mid C$$

Local Independence

Definition

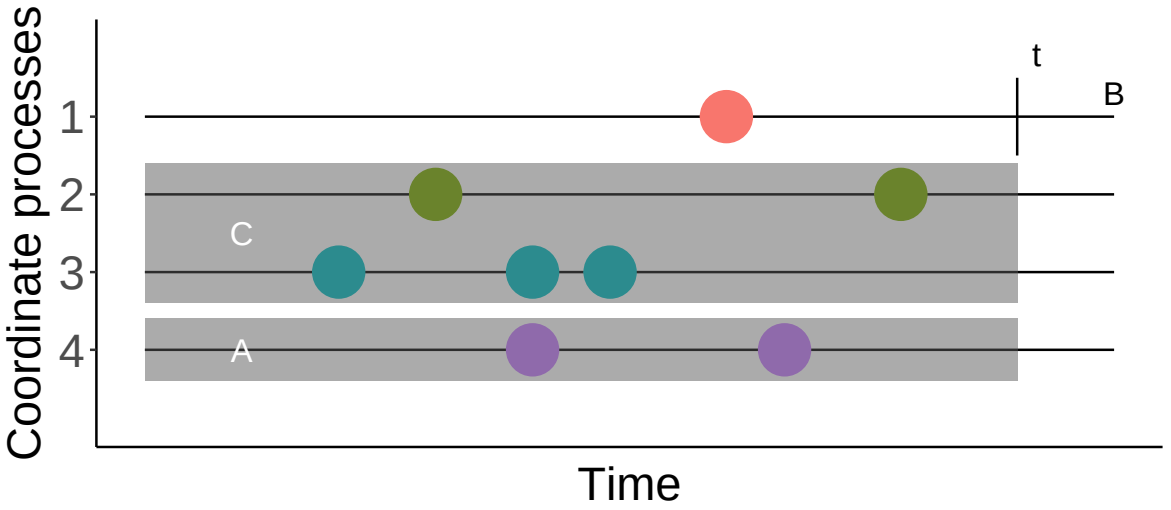
Let $A, B, C \subseteq \mathcal{L}$. We say that B is *locally independent* of A given C, and write $A \not\to_\lambda B \mid C$ if for all $i \in B$, $E(\lambda_t^i \mid \sigma(X_{0:t}^{A \cup C}))$ does not depend on tracks in A.



Local Independence

Definition

Let $A, B, C \subseteq \mathcal{L}$. We say that B is *locally independent* of A given C, and write $A \not\to_\lambda B \mid C$ if for all $i \in B$, $E(\lambda_t^i \mid \sigma(X_{0:t}^{A \cup C}))$ does not depend on tracks in A.



Local Independence Graphs / Graphical Event Models

Given a stochastic process, we define its *local independence graph* as the *directed graph* (DG), $\mathcal{G} = (\mathcal{L}, \mathcal{E})$, such that for $i, j \in \mathcal{L}$:

$$i \not\to_{\mathcal{G}} j \Leftrightarrow i \not\to_{\lambda} j \mid \mathcal{L} \setminus \{i\}$$

The implication from left to right is the *pairwise Markov property* ($i \not\to_{\mathcal{G}} j$ denotes that there is no edge from i to j in $\mathcal{G}$).

Intuitively, the edge $i \to_{\mathcal{G}} j$ is omitted if what happens at time t in process j does not depend (directly) on the past of process i.

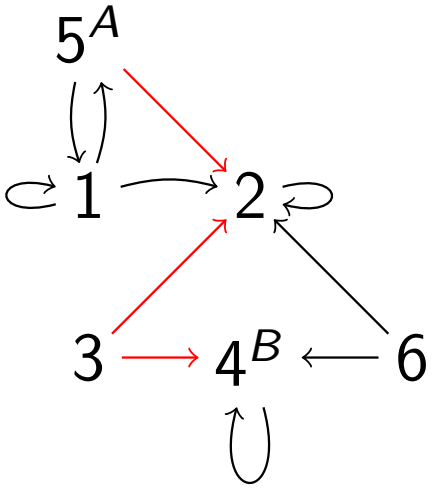
δ-separation

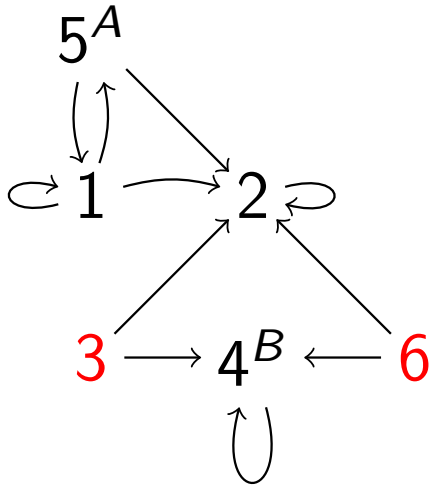
δ-separation is a graphical separation criterion, analogous to d-separation in DAGs. δ-separation from A to B given C for disjoint sets $A, B, C \subseteq \mathcal{L}$ occurs when a certain kind of walk is absent in the graph. The most important difference from d-separation is the fact that only walks with a *head* at j can be connecting from i to j given some set C.

We will just give some examples.

δ-separation

δ-separation is a graphical separation criterion, analogous to d-separation in DAGs. δ-separation from A to B given C for disjoint sets $A, B, C \subseteq \mathcal{L}$ occurs when a certain kind of walk is absent in the graph.





Left: A walk (in red) which is δ-connecting from 5 to 4 given $C = \{2\}$, and not δ-connecting given $C = \{3\}$.
Right: $A = \{4\}$ is δ-separated from $B = \{5\}$ by any C such that $\{3, 6\} \subseteq C$.

Markov Properties

Under some regularity conditions, the *global Markov property* holds (Didelez, 2008; Mogensen, Malinsky, and Hansen, 2018).
If B is δ-separated from A given C in the graph $\mathcal{G}$, then we write $A \perp_\delta B \mid C \ [\mathcal{G}]$. δ-separation is not symmetric.

Theorem (The global Markov property)

Let X be a TPP and let $\mathcal{G}$ be its local independence graph. Let $A, B, C \subseteq V$. Then

$$A \perp_\delta B \mid C \ [\mathcal{G}] \Rightarrow A \not\to_\lambda B \mid C.$$

This connect TPPs and their local independence graphs.

More General Graphs and Local Independence Testing

- *Directed mixed graphs* (include bidirected edges $\leftrightarrow$ as well as directed edges) and *μ-separation* allow graphical marginalization to model partially observed systems (Mogensen and Hansen, 2020).
  - Analogous to MAGs and ADMGs with *m*-separation in DAG-based models.
- One can learn (marginalized) local independence graphs based on tests of local independence (Meek, 2014; Mogensen, Malinsky, and Hansen, 2018; Christgau, Petersen, and Hansen, 2022).
  - Analogous to structure learning in DAG-models based on tests of conditional independence.

Parametric Graphical Event Models

Overview

(Dynamic) Graphical Models

- Discrete-time
  - Dynamic Bayes nets
  - Time series graphs
- Continuous-time
  - Continuous-time Bayes nets
  - Local independence graphs/graphical event models

Parametric (Multivariate) TPPs

The literature makes various assumptions about history dependence. Examples:

- Poisson networks (Rajaram, Graepel, and Herbrich, 2005)
- Piecewise-constant intensity models (Gunawardana, Meek, and Xu, 2011)
- Multivariate Hawkes processes (Zhou, Zha, and Song, 2013)
- Proximal GEMs (Bhattacharjya, Subramanian, and Gao, 2018).
- Ordinal GEMs (Bhattacharjya, Gao, and Subramanian, 2020; Bhattacharjya, Gao, and Subramanian, 2021).

PGEM: Preliminaries

- Event dataset $\mathbf{D} = \{(l_i, t_i)\}, i = 1, \cdots, N; l_i \in \mathcal{L}, |\mathcal{L}| = M$, where t_i are assumed temporally ordered b/w 0 and T.
- Inter-event times b/w events labels Z and X are denoted $\hat{t}_{zx}$ for $Z \neq X$; $\hat{t}_{xx}$ for $Z = X$ includes period at the end.

Example

- $M = 3$ labels; $N = 7$ events
- $\{\hat{t}_{ac}\} = \{2, 8\}$; $\{\hat{t}_{bc}\} = \{1, 7\}$; $\{\hat{t}_{bb}\} = \{3, 7, 7\}$

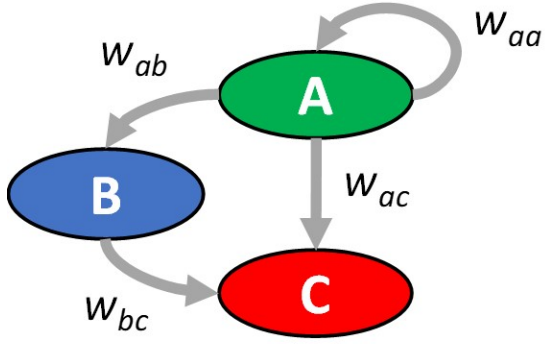
PGEM: Formulation

Definition

A proximal graphical event model includes:

- A graph $\mathcal{G}$ with a node for each label X in $\mathcal{L}$.
- A window for every edge: $\mathcal{W} = \{w_x : \forall X\} = \{w_{zx} : \forall Z \in \mathbf{U}\}$.
- An intensity parameter for every node X and instantiation $\mathbf{u}$ of its parents' occurrences, $\Lambda = \{\lambda_{x|\mathbf{u}}^{w_x} : \forall X \in \mathcal{L}\}$.

Assumption: A label's intensity depends on whether its parents have occurred at least once in their respective recent (i.e. proximal) histories.

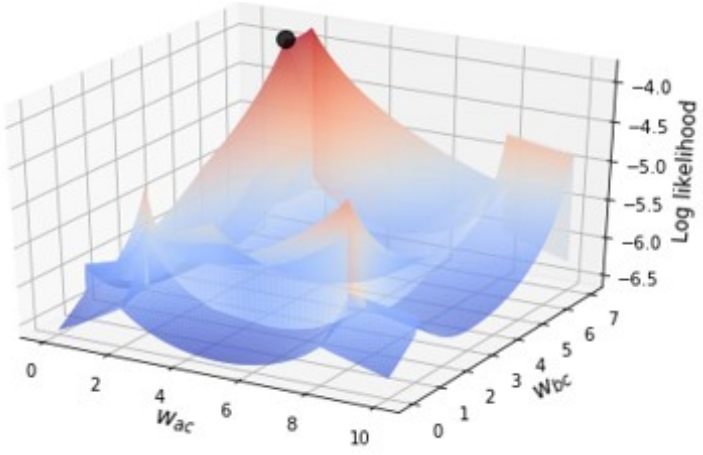


PGEM: Parameter Learning

Given graph $\mathcal{G}$ and windows $\mathcal{W}$:

- $LL = \sum_X \sum_{\mathbf{u}} \left(-D(\mathbf{u}) \lambda_{x|\mathbf{u}} + N(x, \mathbf{u}) \log \left(\lambda_{x|\mathbf{u}} \right) \right)$, where:
  - $N(x, \mathbf{u})$: # of times X occurs and condition $\mathbf{u}$ is true
  - $D(\mathbf{u})$: duration over the horizon where condition $\mathbf{u}$ is true
- $BIC = LL - \log(T) \sum_X 2^{|\mathbf{U}|}$ (Score decomposes!)
- Max. likelihood estimates: $\hat{\lambda}_{x|\mathbf{u}} = \frac{N(x,\mathbf{u})}{D(\mathbf{u})}$

Given $\mathcal{G}$, finding the optimal $\mathcal{W}$ is a combinatorial problem!



PGEM: Some Theoretical Results

Theorem

For a node X with single parent Z, the log likelihood maximizing window w_{zx} either belongs to or is a left limit of a window in the candidate set $\mathcal{W}^* = \{\hat{t}_{zx} \cup max\{\hat{t}_{xx}\}\}$.

- Intuition: the optimal is at (or limit to) points where the counts $N(x, \mathbf{u})$ change.

Theorem

Using BIC as score, if $2^{\mathbf{U}} > \frac{N(x)(1-\log N(x))}{\log T}$ for parents $\mathbf{U}$ of X then no proper superset of $\mathbf{U}$ can be X's optimal parents.

- Could help with efficient parent set, similar to Bayes nets (Campos and Ji, 2011).

PGEM: Score-based Learning

Learning Problem: Given event dataset **D**, learn PGEM $\{\mathcal{G}, \mathcal{W}, \Lambda\}$.

Outer loop performs graph search:

- Score-based forward backward search for parents **U** for every event label X

Inner loop learns the parameters:

- Compute "optimal" windows using theory-driven heuristic(s)
- Compute LL, score and MLE estimates for intensity rates

PGEM: Constraint-based Structure Learning

Recent work (Bhattacharjya et al., 2022) considers testing for process independence, analogous to methods that test for conditional independence in Bayes nets (Spirtes, Glymour, and Scheines, 2000).

Algorithm 1 PC Algorithm for Parent Discovery in GEMs

Inputs: Event label $X \in \mathcal{L}$, event dataset D (over $\mathcal{L}$), threshold parameter for tester α
Outputs: Parents $\mathbf{U}$ for X

$\mathbf{U} = \mathcal{L}$
for all Y in $\mathcal{L}$ **do**
  flag = False, $n = 0$, $\mathbf{Z}^* = \mathbf{U} \setminus Y$
  while $n \leq |\mathbf{Z}^*|$ and flag = False **do**
    for all $\mathbf{Z}$ that are subsets of size n in $\mathbf{Z}^*$ **do**
      Obtain score from a process independence test, checking if $Y \not\to X|\mathbf{Z}$
      if score $\leq \tau = f(\alpha)$ (indicating process independence) **then**
        flag = True, $\mathbf{U} = \mathbf{U} \setminus Y$, break from loop
    $n = n + 1$

This assumes we have access to a **process independence tester!**

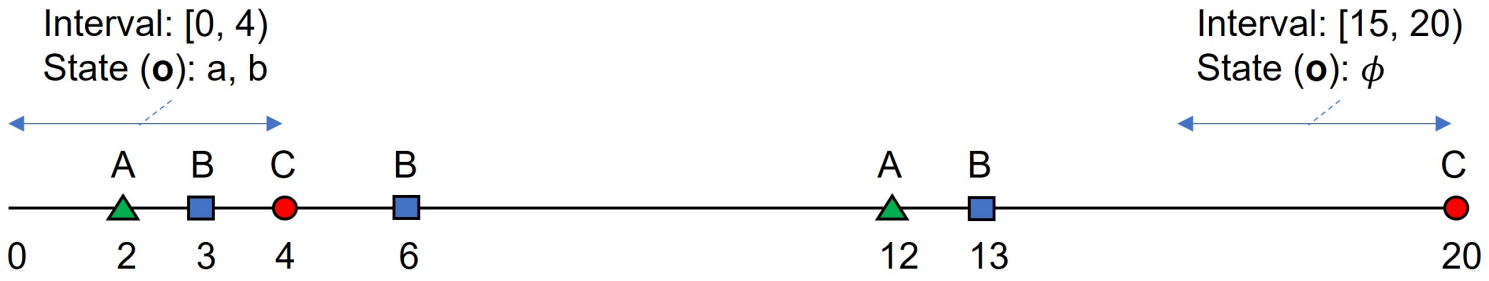
OGEM: Preliminaries

Definitions

- A masking function $\phi(\cdot)$ takes a sequence of events and returns a sub-sequence where a label is never repeated.
- An order instantiation for labels $\mathbf{Z}$ is a permutation of any subset, obtained at time t by applying $\phi(\cdot)$ to events from $\mathbf{Z}$ occurring within the interval $[\max\{t - w, 0\}, t]$.

Example

The figure below shows order instantiations at occurrences of C with respect to labels $\{A, B\}$ using window $w = 5$.



Tabular OGEM

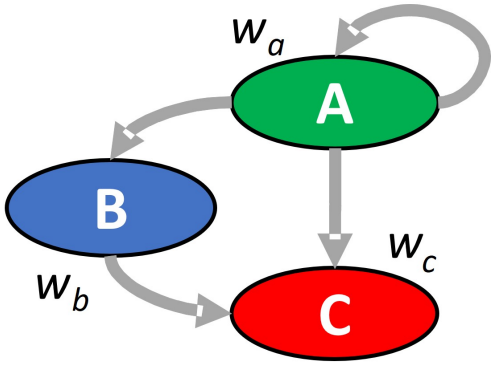
Definition

A tabular ordinal graphical event model with $\phi(\cdot)$ includes:

- A graph $\mathcal{G}$ with a node for each label X in $\mathcal{L}$.
- A window for every node: $\mathcal{W} = \{w_x : \forall X \in \mathcal{L}\}$.
- An intensity parameter for every node X and order instantiation $\mathbf{o}$ of its parents' occurrences, $\Lambda = \{\lambda_{x|\mathbf{o}}^{w_x} : \forall X\}$.

Limitations:

- # of order instantiations are super-exponential in $|\mathbf{U}|$.
- Complex models are hard to learn.
- Not all order instantiations will be observed in the data.



OGEM: Tree Representation

Basic idea: Make some order instantiations share parameters!

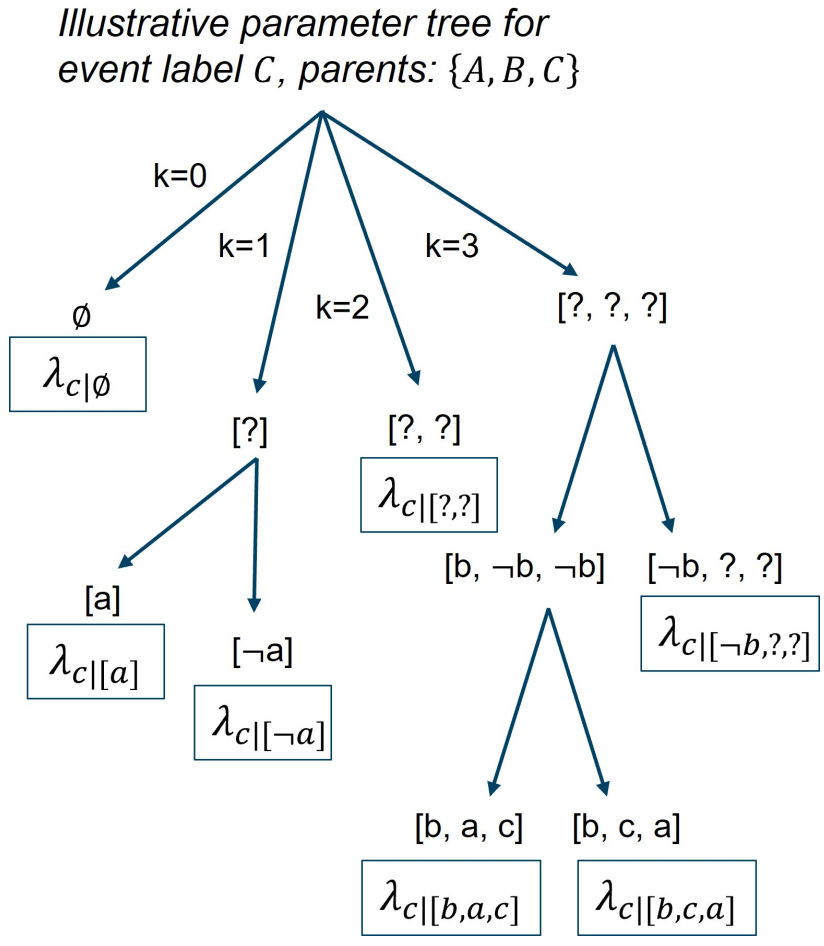
Definition

An order representation **r** of length $k < |\mathbf{U}|$ for a set of labels **Z** is a sequence of slots that are either filled with a label in **Z** or restricted by a subset of **Z**. **r** is feasible if consistent with at least one **o**.

Example

Consider $k = 2$ size orders for $\mathbf{Z} = \{A, B, C\}$.

- Ex #1: $r = [A, \neg A]$ encodes $[A, B]$ and $[A, C]$.
- Ex #2: $r = [?, ?]$ encodes all 6 permutations of pairs in $\{A, B, C\}$.



OGEM: Score-based Learning

Learning Problem: Given event dataset **D** and masking function $\phi(\cdot)$, learn $\{\mathcal{G}, \Lambda\}$ for OGEM.

- OGEM-tree: Learn OGEM tree with $\mathcal{W}$ given.
- OGEM-tree-W: As above, but also learn windows $\mathcal{W}$.

Outer loop performs graph search:

- Score-based forward backward search for parents **U** for every event label X

Inner loop learns the tree representation, given parents:

- Loop over all possible k from 0 to |**U**|
- Learn the optimal subtree for each k by splitting at each node

Empirical Investigation (Bhattacharjya, Gao, and Subramanian, 2021)

Task: To compare models around fitting event datasets.

Datasets:

- ICEWS [politics]
- IPTV [TV viewership]
- LinkedIn [employment]
- Mimic [healthcare]
- Stack Overflow [online engagement]

Dataset	N (# events)	M (# labels)	MHP	PGEM	OGEM-tab	OGEM-tree	OGEM-tree-W
ICEWS							
Argentina	3252	104	-1419	-1386	-1369	**-1366**	-1393
Brazil	4249	114	-2169	-2000	-2057	-2050	**-1993**
Colombia	841	79	-528	-534	**-518**	**-518**	-537
Mexico	1905	97	**-760**	-797	-771	-769	-766
IPTV	332980	16	**-64168**	-77009	-75114	-72696	-74491
LinkedIn	2932	10	-1593	-1462	-1478	-1418	**-1406**
Mimic	2419	75	-567	-500	-474	**-429**	-454
Stack Overflow	71254	22	-52543	-48323	-49344	-49192	**-48232**

Table 1: Log likelihood on the test sets.

Methodology:

- Metric: Log likelihood; (70/15/15)% split for train/dev/test sets
- Baselines: multivariate Hawkes process, PGEM, tabular OGEM

Results: OGEM-tree models fit data reasonably compared to baselines.

Neural TPPs

Deep Learning

Advances state-of-the-art performances in numerous tasks and applications.

- e.g., computer vision, NLP, robotics, healthcare, chemistry, astrophysics ...

Advantages:

- Universal function approximator
- High-level representation
- Scaling to billions of parameters, with modern computation tools such as GPUs

Parametric GEMs vs. Neural Point Processes

Parametric GEMs

Makes various assumptions about historical dependence + irregular time dynamics:

- Hawkes
- Proximal
- Basis functions
- ...

Neural Point Processes

Less restrictive assumptions:

- Long and complex dependency:
  - RNN
  - Transformers
- Irregular dynamics:
  - Hawkes
  - Sampling
  - Integral
  - ...

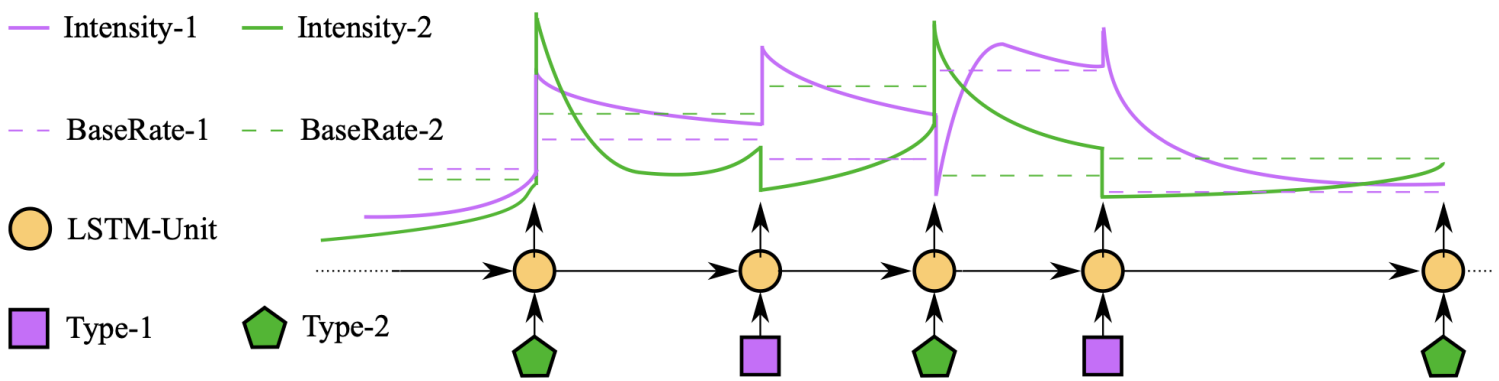
Into to Neural Point Process

Key Ideas:

- Dynamic: to use a RNN/LSTM cell to automatically learn the historical state h, with $\lambda_k(t) = f_k(\mathbf{W}_k^T \mathbf{h}(t))$.

$$\mathcal{L}(D) = \sum_i^N \log \lambda_{k_i}(t_i) - \int_{t=0}^{T} \lambda(t) dt$$

- Evolution: to let the hidden state continuous evolve (exponentially) at some rate λ_k toward a steady state value



Du et al. (2016) and Mei and Eisner (2016)

Neural Point Process with (some) Graph

How could we extract a graphical representation of (causal) relationships between different events?

Difficulties:

- neural networks: how to represent the relation graphs?

- time-dependent: how to find fixed relation graphs across time?

Neural Point Process with (some) Graph

How could we extract a graphical representation of (causal) relationships between different events?

Difficulties:

- how to represent the relation graphs?
  1. Attention mechanism
  2. A dedicated set of parameters (e.g., a binary gating matrix)
- how to find fixed relation graphs across time?
  - Aggregation function: event instance to event type relations

Neural TPPs with (some) Graph

Approach 1:
Attention mechanism is used to compute graphical relationships.

- RNN + Decay + Attention (Xiao et al., 2017)
- RNN + Sampling + Attention (Gao et al., 2020)
- Attention + Decay (Zhang et al., 2020a)
- Aggregation: mean

Approach 2:
A dedicated set of parameters is used to model the graph in neural TPPs.

Attention: A Brief Review

Original Attention

Given a current state h_i and several past states h_j

1. Alignment: compute $e_{ij} = f(h_i, h_j)$
2. Weight: $\alpha_{i,j} = softmax(e) = \frac{\exp e_{ij}}{\sum_j \exp e_{ij}}$
3. Context: $c_i = \sum_j \alpha_{ij} h_j$

Bahdanau, Cho, and Bengio (2014)

General Attention

3 components: query q, key k, and values v

1. Alignment: compute $e_{q,k_j} = f(q, k_j)$
2. Weight: $\alpha_{q,k_j} = softmax(\frac{e}{\sqrt{|k|}})$
3. Attention: $\text{Att}(q, k, v) = \sum_j \alpha_{q,k_j} v_j$

Vaswani et al. (2017)

Attention: A Brief Review

Original Attention

Given a current state h_i and several past states h_j

1. Alignment: compute $e_{ij} = f(h_i, h_j)$
2. Weight: $\alpha_{ij} = softmax(e) = \frac{\exp e_{ij}}{\sum_j \exp e_{ij}}$
3. Context: $c_i = \sum_j \alpha_{ij} h_j$

Bahdanau, Cho, and Bengio (2014)

General Attention

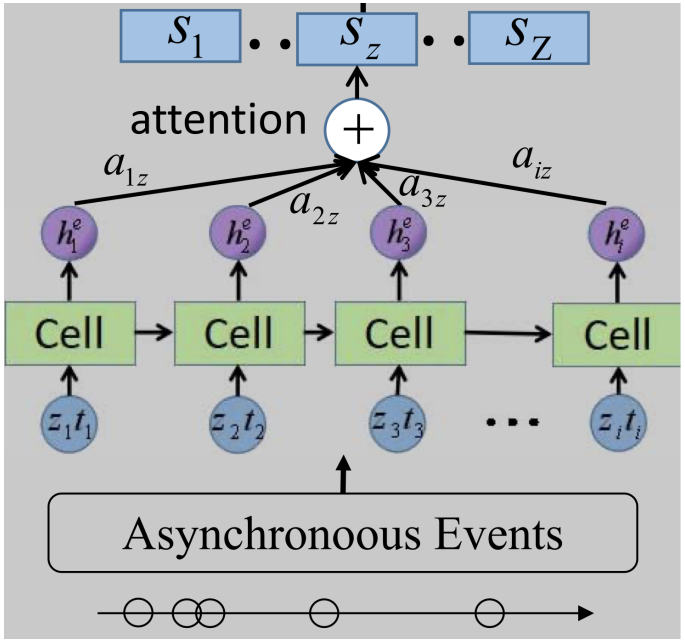
3 components: query q, key k, and value v

1. Alignment: compute $e_{q,k_j} = f(q, k_j)$
2. Weight: $\alpha_{q,k_j} = softmax(\frac{e}{\sqrt{|k|}})$
3. Attention: $\text{Att}(q, k, v) = \sum_j \alpha_{q,k_j} v_j$

Vaswani et al. (2017)

A1.1: Recurrent Point Process Network

Three parts: RNN + Dynamic Decaying + Attention



$$\alpha_{z_i,z} = \text{softmax}(f(h_i^e, u_z))$$

$$s_z(t) = \sum_i \alpha_{z_i,z} h_i^e \exp(-w(t - t_i)), \quad \lambda_z(t) = f(s_z(t))$$

Aggregation: $G_{k,k'} = \text{mean}(\alpha_{z_i,z})$

Xiao et al. (2017) and Xiao et al. (2019)

A1.2: Multi-Channel Neural GEM

Beyond exponential functions:

- Piece-wise constant function
- Time lags with delayed excitation or inhibition
- Varying time scales among events

Key ideas of MCN-GEM:

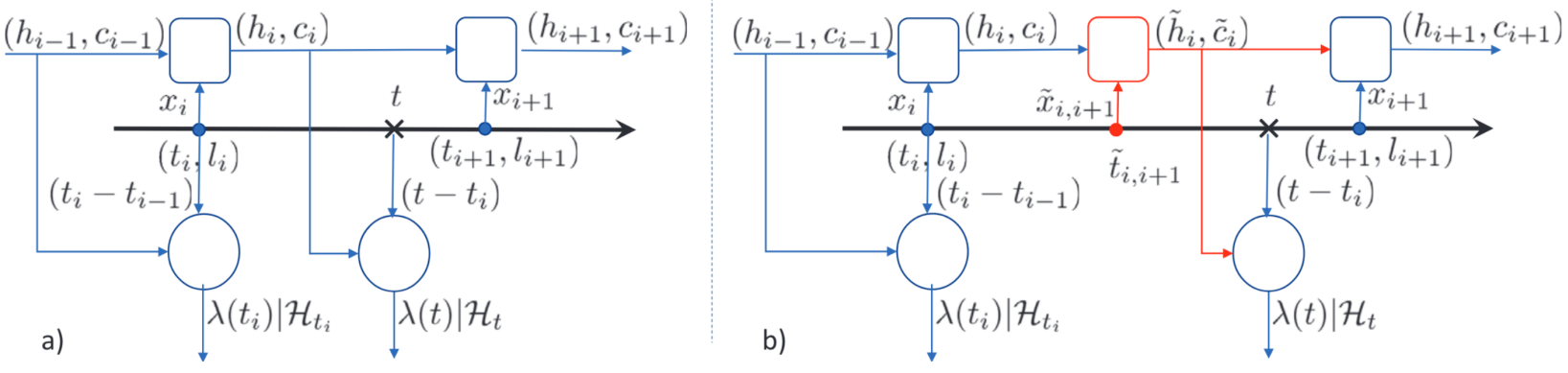
- Nonparametric: utilize time intervals between event arrivals to sample negative evidence
- Spatio-temporal attention

Gao et al. (2020)

A1.2: Multi-Channel Neural GEM

RNN + sampling negative evidence (non-event occurrences)

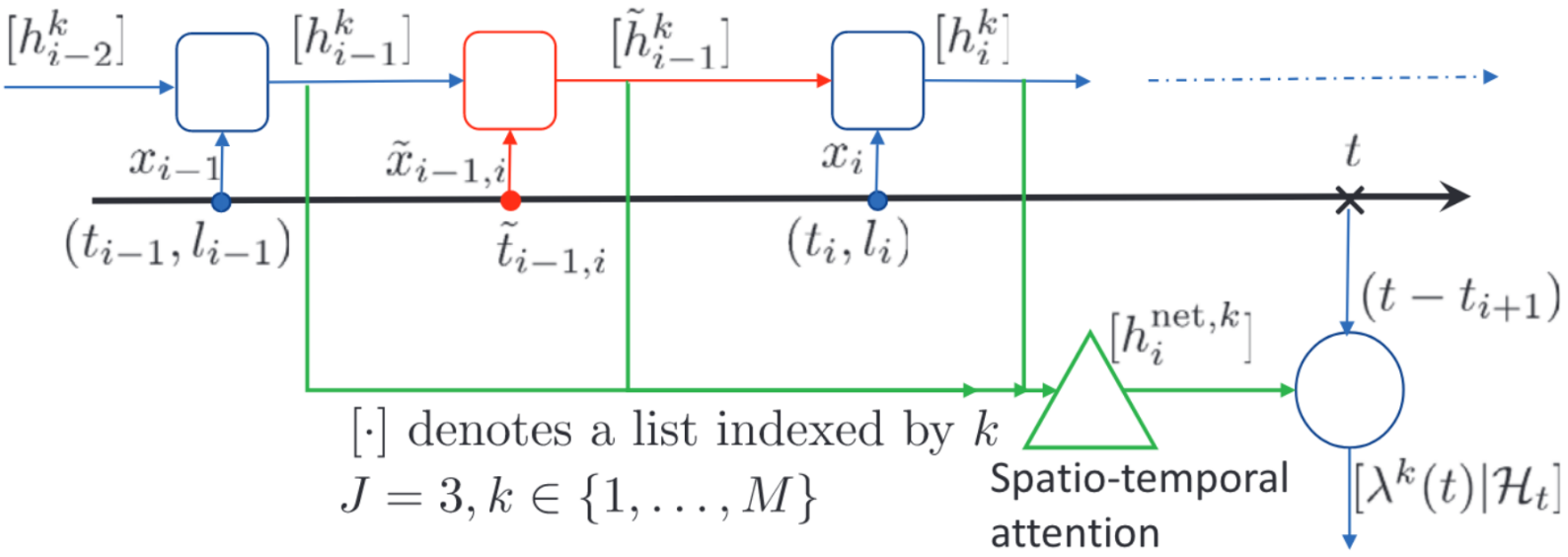
$$\mathcal{L}(D) = \sum_{i}^{N} \log \lambda_{k_i}(t_i) - \sum_{i}^{N+1} \Delta t_i \sum_{k} \lambda_{k_i}(t_i)$$



Gao et al. (2020)

A1.2: Multi-Channel Neural GEM

+ spatio-temporal attention $\alpha \in R^{J \times K} \Rightarrow G_{k,k'} = \frac{\sum_i^T \sum_j^J \alpha_{ijk'}^k}{|T||J|}$



Gao et al. (2020)

General Attention

Limitations of RNN

- Difficulty to capture the long-term and/or non-sequential dependencies.
- In-efficiency in training and hard to parallel.

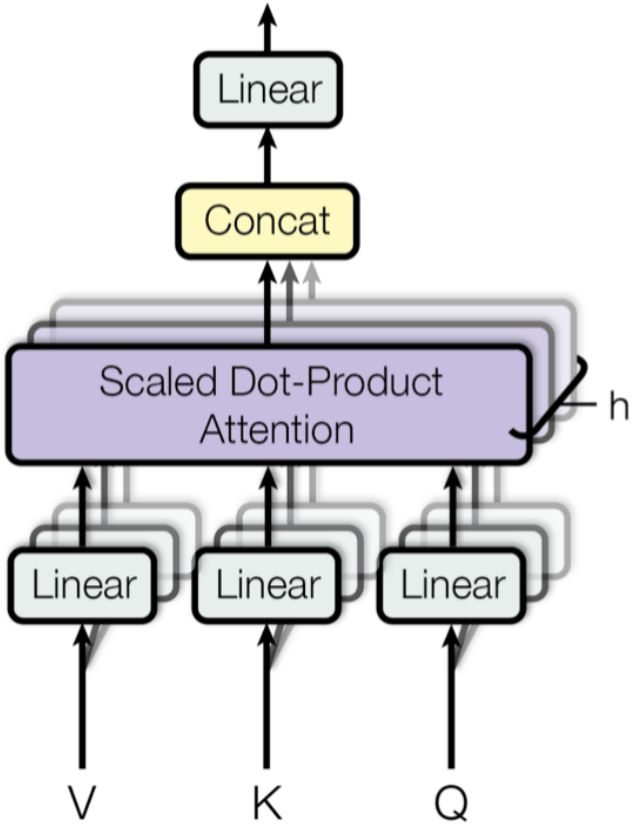
Multi-head Attention

- $\text{Att}(q, k, v) = \text{softmax}(\frac{qk^T}{\sqrt{|k|}})v$
- concatenation + combination of multiple different attentions

Transformer-based TPPs (Zhang et al., 2020a; Zuo et al., 2020; Gu, 2021; Mei, Yang, and Eisner, 2022; Shou et al., 2023a; Shou et al., 2023b)

Multi-Head Attention: A Review

Attention is all you need...

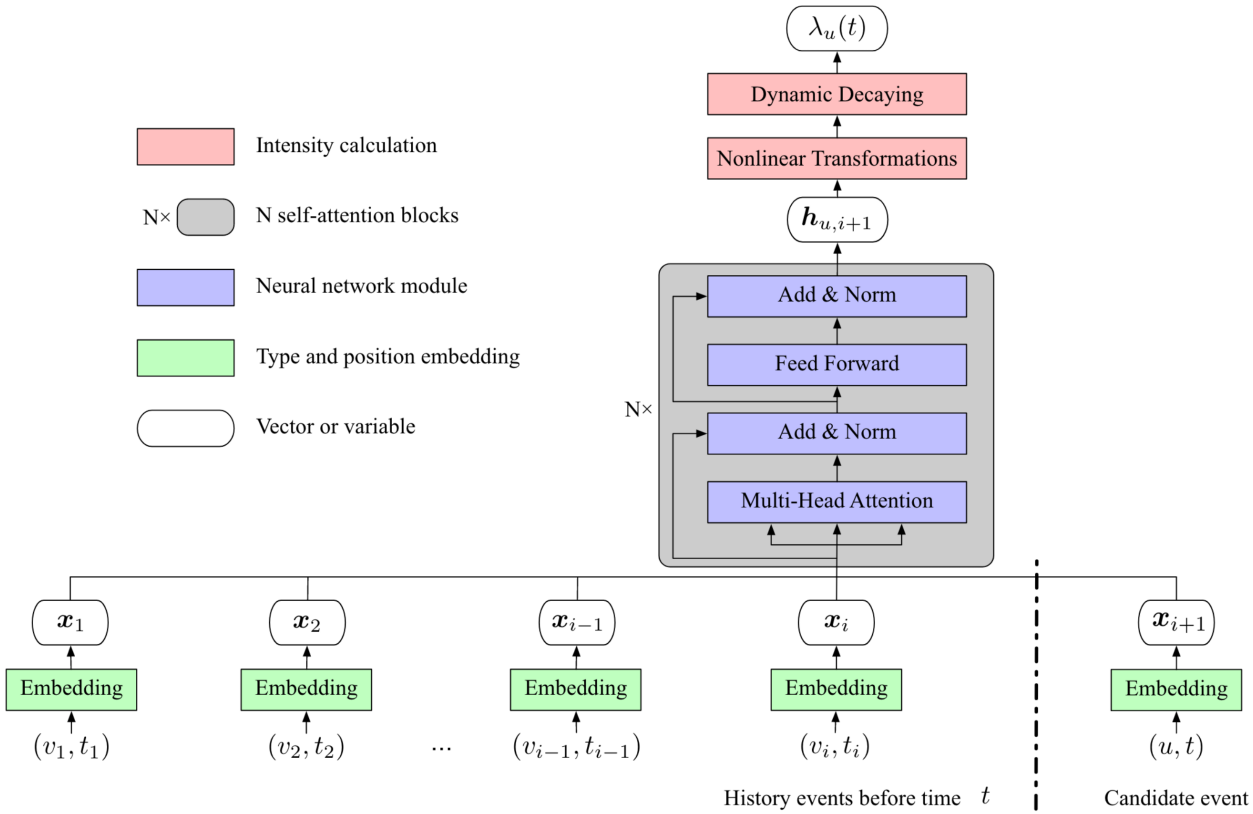


Vaswani et al. (2017)

A1.3: Self-Attentive Hawkes Models

Attention is all you need (to extract graphs)...

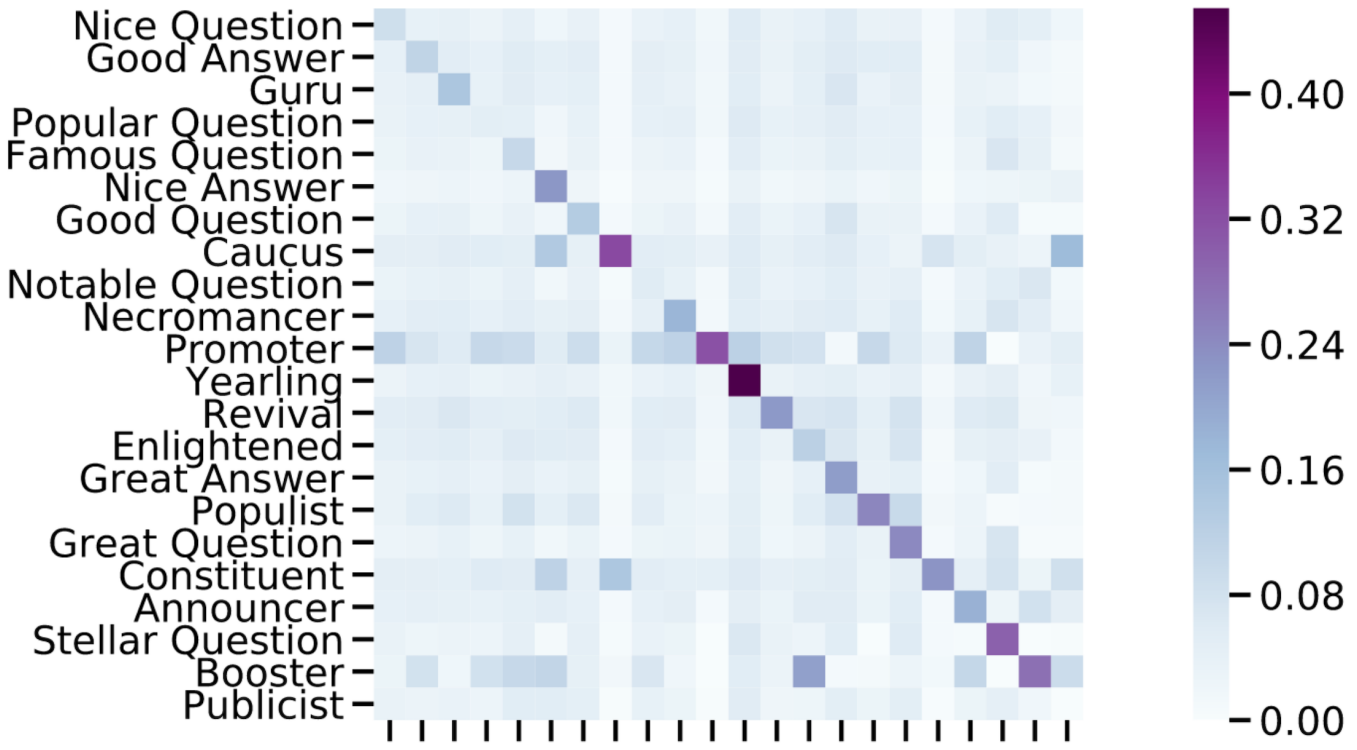
- Attentions + Dynamic Decaying



Zhang et al. (2020a)

A1.3: Self-Attentive Hawkes Models

SAHP: similar with a single attention but now with multi-head.



Zhang et al. (2020a)

Neural TPPs with Explicit Graph Modeling

Approach 1:
Attention mechanism is used to compute graphical relationships.

Approach 2:
A dedicated set of parameters is used to model the graph in neural TPPs.

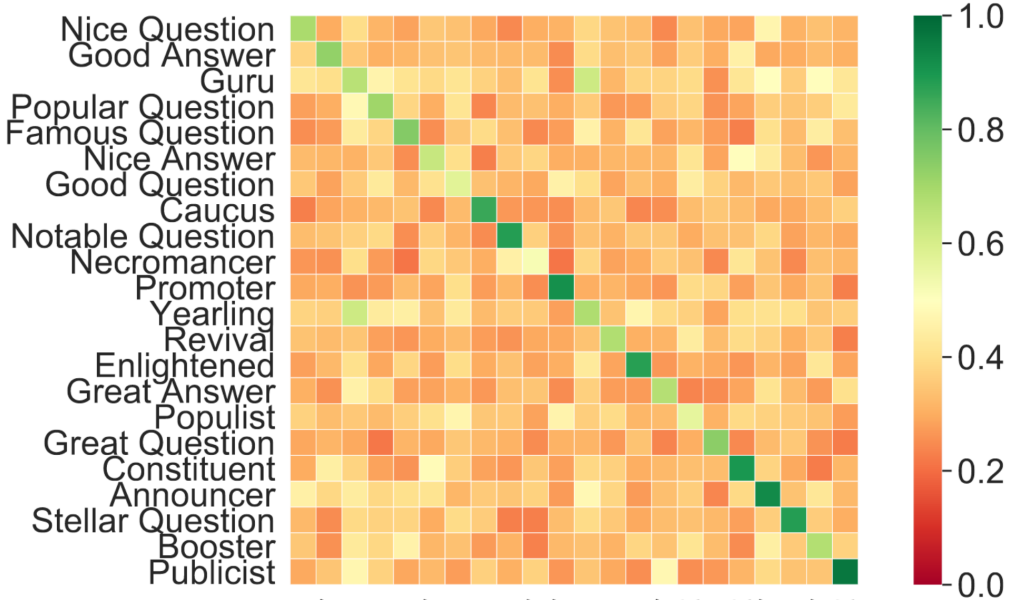
- Attention + Gating Matrix + Mean (Zhang, Lipani, and Yilmaz, 2021)
- RNN + Attributions + Mean (Zhang et al., 2020b)
- Explicit Type Attention (Shou et al., 2023b)

A2.1: Learning Neural Point Processes with Latent Graphs

Modify SAHP attention score with explicit graphs

$$\alpha(h_i, h_j) = \mathcal{G}_{(k,k')} \exp(h_i^T h_j)$$

where $G_{k,k'} \sim \text{Ber}(k, k')$



Zhang, Lipani, and Yilmaz (2021)

A2.2: Causality from Attributions on Sequence of Events

Explicit Graph Model with Attribution

Definition

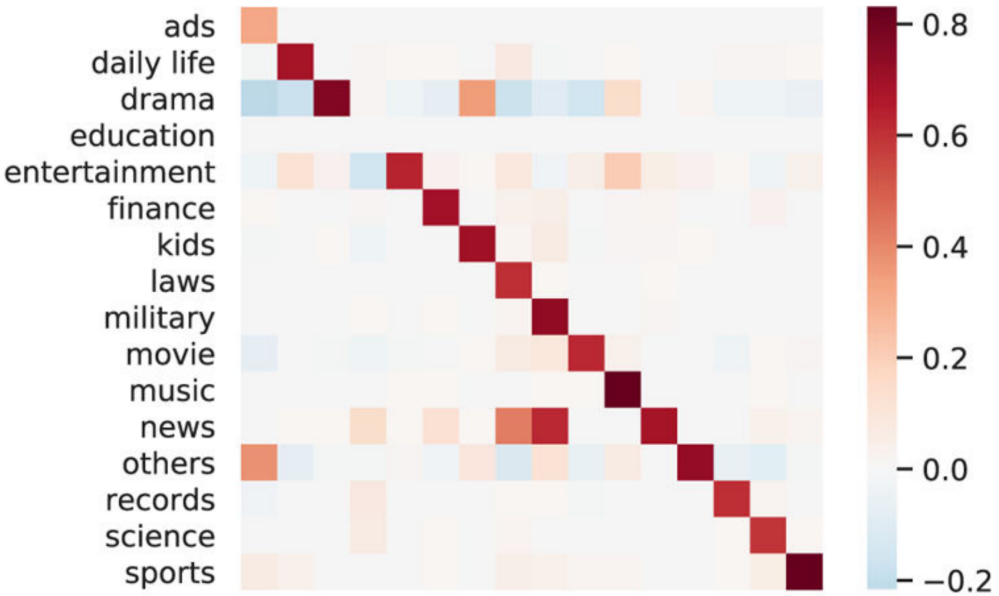
Attribution $A_j(f_k, x_i, \underline{x}_i)$ is defined as the event contribution of the j-th event to the target prediction $f_k(x_i)$ relative to the baseline $f_k(\underline{x}_i)$.

- f: cumulative intensity function
- GRU + semi-parametric weighted Gaussian Decay

Zhang et al. (2020b)

A2.2: Causality from Attributions on Sequence of Events

$$+ \quad A_{k,k'} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{i} I(k_j^s = k') A_j(f_k, x_i, \underline{x}_i)}{\sum_j^n I(k_j^s = k')}$$



Zhang et al. (2020b)

A2.3: Influence-Aware Attention for Multivariate Temporal Point Processes

Models event type influence directly

- Base dynamic model: Transformer
- Encoder: models pairwise interactions by summarizing instance-wise attention scores to type-wise probabilities

$$\mathrm{A2I}(\mathbf{S}_{enc}^{(B)}) = \mathbf{P}^{\top}\mathbf{S}_{enc}^{(B)}\mathbf{P} \tag{1}$$

  where $\mathbf{P} \in \{0,1\}^{L\times D}$ is a binary indicator matrix which specifies D type of events occurring in S for the L event instances.

- Decoder: learns a dynamical model given an influencing set for each event

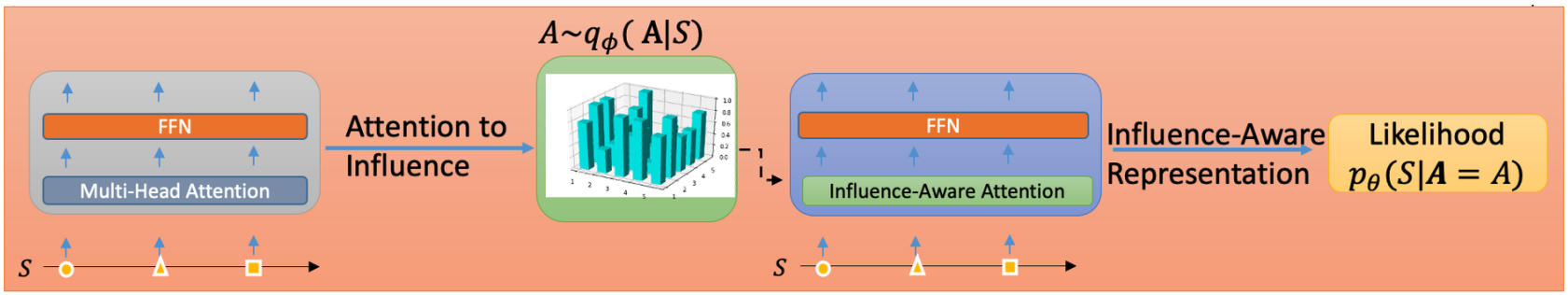
$$\tilde{\mathbf{S}}_{dec} = \mathbf{S}_{dec} \odot \mathbf{P}A\mathbf{P}^{\top} \tag{2}$$

Shou et al. (2023b)

A2.3: Influence-Aware Attention for Multivariate Temporal Point Processes

Models event type attention directly

- Base dynamic model: Transformer
- Encoder: models pairwise interactions by summarizing instance-wise attention scores to type-wise probabilities
- Decoder: learns a dynamical model given an influencing set for each event
- ELBO: $\mathcal{L}(\theta, \phi) = \mathbb{E}_{q_\phi}[\log \frac{p(\mathbf{A})}{q_\phi(\mathbf{A}|S)}] + \mathbb{E}_{q_\phi}[\log p_\theta(S|\mathbf{A} = A)]$



Shou et al. (2023b)

Summary

- Model
  - Dependency: RNNs and Transformers
  - Irregular Dynamics: parametric and non-parametric
- Graph representation + aggregation
  - Attention/multi-head attention, and explicit graph representation
- Other work: graph neural network + TPP (Zhang and Yan, 2021; Xia, Li, and Li, 2022)

Causality in TPPs

Overview

Causal Inference

- IID Observational data
  - Potential outcome framework
  - Average treatment effect (ATE)
- (Regular) Time Series
  - G-Computation
  - Marginal structural models
  - Neural methods (Melnychuk, Frauen, and Feuerriegel, 2022; Frauen et al., 2023)

Causal Inference in TPPs

The literature adapts various definition from causal inference in IID and time series setting. Examples:

- Discrete Events (Gao et al., 2021)
- Continuous-Value Events (Zhang, Cao, and Liu, 2022)
- Counterfactual TPP (Noorbakhsh and Rodriguez, 2022)

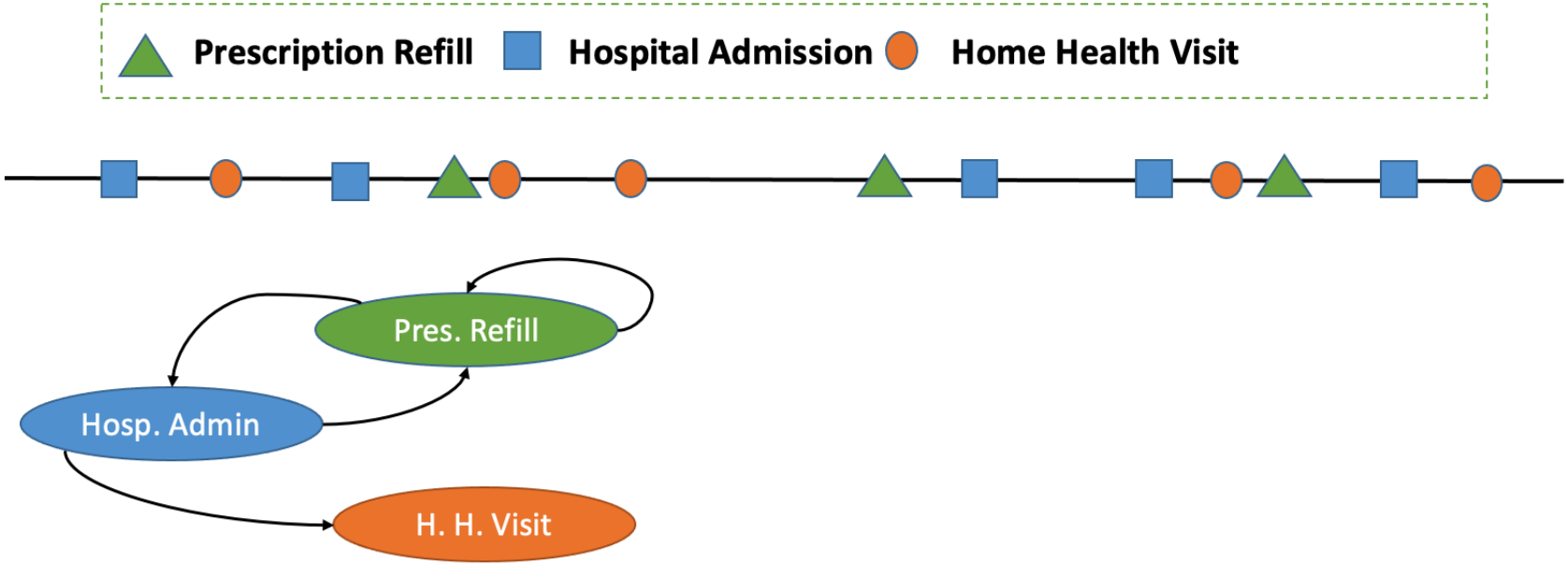
Causal Inference in TPPs

- General Theory
  - Local Independence Graphs (Didelez, 2008; Røysland, 2012; Didelez, 2015)
  - Statistical Models of Causal Effects (Lok, 2008)
- Practical Algorithms
  - Causal Event Inference: Treatment Effect
    - Discrete events (Gao et al., 2021)
    - Continuous-value events (Zhang, Cao, and Liu, 2022)
  - Counterfactual Event Generation & Inference (Noorbakhsh and Rodriguez, 2022; Schulam and Saria, 2017)

A1: Causal Inference for Events: Motivation

Motivation: Given event dataset **D**, estimate for the treatment effect for a pair of events of interest. Examples:

- exercise and blood glucose level changes of a diabetic patient
- abnormal states in web applications and other downstream faults
- food shortage and protests in socio-economics



A1: Causal Inference for Events: Definition

For a pair of events (z,y) = (Prescription Refill, Hospital Admission)

- Treatment variable Z_t at time t:
  - A function of historical prescription refills
  - Recent history formulation: whether prescription has been refilled at least once within a window w into the past from t : $[t - w, \mathrm{t})$
- Outcome variable Y_t at time t
  - A function of future hospital admissions, starting at time t
  - Consider uncertainty: the occurrence rate $\lambda_y(t)$ (and cumulative rates) of hospital admissions at time t given refilled prescription or not.
- Covariates X_t at time t:
  - A function of historical occurrences of other events, such as home health visit, urgent care visit
  - Recent history formulation: whether other event labels have occurred at least once in $[t - w, t)$

Gao et al. (2021)

A1: Causal Inference for Events: ATE Estimation

Goal: Given event dataset **D**, for a pair of event (z,y) , estimate the causal relationship of z on y via average treatment effect (ATE).

- Define ATE:

$$ATE = \mathbb{E}_{H_T}[\frac{1}{T}\int_t \lambda_y^{z_t=1}(t) - \lambda_y^{z_t=0}(t)dt] \tag{3}$$

- Define Propensity Score:

$$e_t^* = e(X_t^w) = P(Z^w = 1|X_t^w) \tag{4}$$

- For an observational event dataset, we need to adjust for observed confounding via inverse probability of treatment weighting (IPTW) to arrive at unbiased estimator of ATE.

Gao et al. (2021)

A1: Causal Inference for Events: Learning

- In practice, we estimate :

$$ATE = \mathbb{E}_{H_T}[\frac{1}{T}\int_t \frac{1}{e_t^*}\lambda_y^{z_t=1}(t) - \frac{1}{1-e_t^*}\lambda_y^{z_t=0}(t)dt] \quad (5)$$

- Parameter Learning via PGEM:
  - Window w: use prior knowledge or heuristic pairwise search
  - Propensity Score e_t^*:

$$P(Z_t^w = 1|X_t^w) = \frac{D(z_t^w = 1; X_t^w)}{D(X_t^w)} \quad (6)$$

  - $D(X_t^w)$: the duration of configuration x_t^w being true in the data, i.e.

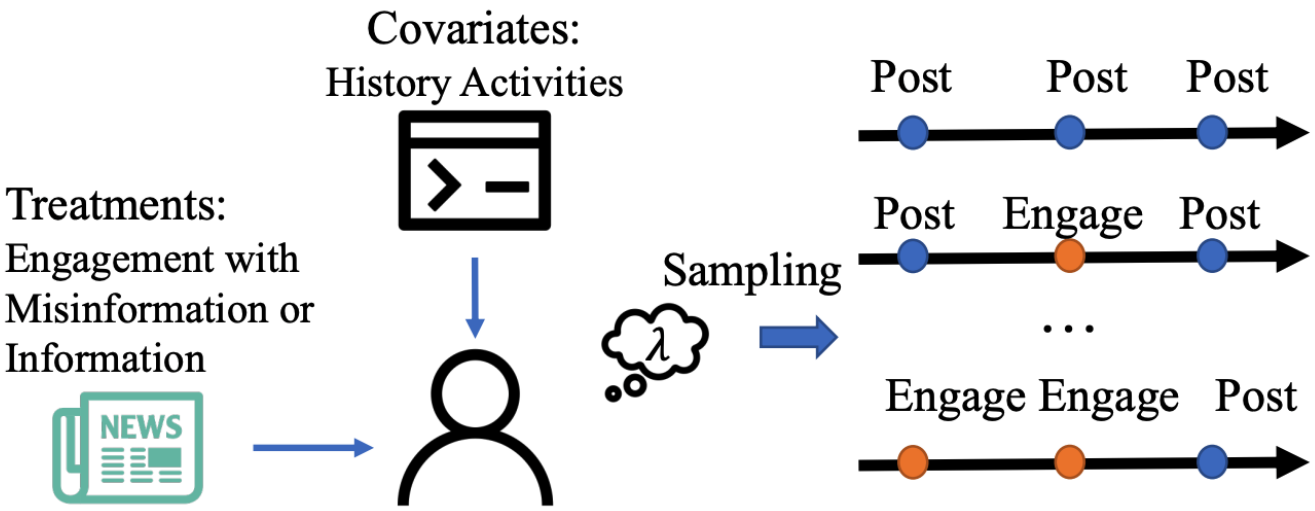
$$D(Z_t^w = 1; X_t^w) = \sum_{i=1}^{N+1}\int_{t_{i-1}}^{t_i} I_{x_t^w}(Z_t^w = 1, t)dt \quad (7)$$

  - $\lambda_y^{z_t=1}(t)$, $\lambda_y^{z_t=0}(t)$ from PGEM parental configuration $\lambda_{y|Z_t}$

Gao et al. (2021)

A2: Continuous-Valued Events: Motivation

- Extension of causal inference for event pairs framework for continuous real-value treatment (i.e. text, sentiment score, and metadata)
- Estimate the effect of misinformation post on future posts of a user in a fixed window



Zhang, Cao, and Liu (2022)

A2: Continuous-Valued Events: Definition

Dataset:

- Engaging Process: $S^{(e)} = [(f_i^{(e)}, t_i^{(e)})]_{i=1}^{n_e}$
- Generating Process: $S^{(g)} = [(f_i^{(g)}, t_i^{(g)})]_{i=1}^{n_g}$
- $f_i^{(e)}$ and $f_i^{(g)}$ are continuous real feature vectors.

Definition:

- Treatment Tr: an interaction event $(f_i^{(e)}, t_i^{(e)})$
- Covariates X: all historical events prior to $t_i^{(e)}$, $[(f_j^{(e)}, t_j^{(e)})]_{j=1}^{i-1} \cup [(f_j^{(g)}, t_j^{(g)})]_{j=i}^{n_i}$ where $t_{n_i}^{(g)} < t_i \leq t_{n_i+1}^{(g)}$
- Outcome Y: $\lambda(f, t | Tr \cup X)$ of the future generating process up to T.

Zhang, Cao, and Liu (2022)

A2: Continuous-Valued Events: ATE Estimation

- Apply a functional $\mathcal{F}$ to project the λ functions under treatment to a vector with finite dimensions: $\mathcal{F}_T(\lambda, Tr \cup X) = \frac{\phi(t,t+T,\lambda,Tr \cup X)}{\mu(t,t+T,\lambda,Tr \cup X)}$
  - $\phi(t, t+T, \lambda, Tr \cup X) = \int_{sup(f)} f df \int_t^{t+T} \lambda(f, t | Tr \cup X) dt$ (expected feature vector)
  - $\mu(t, t+T, \lambda, Tr \cup X) = \int_{sup(f)} df \int_t^{t+T} \lambda(f, t | Tr \cup X) dt$ (expected total events)
- Intuition: $\mathcal{F}$ computes expected mean feature vector of all posts generated by a user.

$$ATE = \mathbb{E}_{(X,Tr) \sim U}[\mathcal{F}_T(\lambda, Tr \cup X) - \mathcal{F}_T(\lambda, X)]$$

Zhang, Cao, and Liu (2022)

Conclusion

Conclusion

Summary

- Background in TPPs
- GEMs: Graphical Representations of TPPs
- Parametric GEMs
- Neural GEMs
- Causal inference in Event Processes

Other Directions

- Logic / Neuro-Symbolic + Event Processes
- Large Language / Foundation Models + Event Processes
- Real Benchmarks: Validation of (Causal) Graphs
- Partial Observability / Latent Events
- ...

References I

- Aalen, Odd O. (1987). "Dynamic modelling and causality". In: *Scandinavian Actuarial Journal* 1987.3-4, pp. 177–190.
- Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). "Neural machine translation by jointly learning to align and translate". In: *arXiv preprint arXiv:1409.0473*.
- Bhattacharjya, D., T. Gao, and D. Subramanian (2020). "Order-dependent event models for agent interactions". In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1977–1983.
- — (2021). "Ordinal Historical Dependence in Graphical Event Models with Tree Representations". In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 6759–6767.
- Bhattacharjya, D. et al. (2022). "Process Independence Testing in Proximal Graphical Event Models". In: *Proceedings of the 1st Conference on Causal Learning and Reasoning (CLeaR)*.

References II

- Bhattacharjya, Debarun, Dharmashankar Subramanian, and Tian Gao (2018). "Proximal graphical event models". In: *Advances in Neural Information Processing Systems*. Vol. 31.
- Campos, C. P. de and Q. Ji (2011). "Efficient structure learning of Bayesian networks using constraints". In: *Journal of Machine Learning Research* 12.Mar, pp. 663–689.
- Christgau, Alexander Mangulad, Lasse Petersen, and Niels Richard Hansen (2022). "Nonparametric Conditional Local Independence Testing". In: *arXiv preprint arXiv:2203.13559*.
- Commenges, Daniel and Anne Gégout-Petit (2009). "A General Dynamical Statistical Model with Causal Interpretation". In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 71.3, pp. 719–736.

References III

- Didelez, Vanessa (2008). "Graphical models for marked point processes based on local independence". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70.1, pp. 245–264.
- — (2015). "Causal Reasoning for Events in Continuous Time: A Decision - Theoretic Approach". In: *Proceedings of the UAI 2015 Workshop on Advances in Causal Inference*.
- Du, Nan et al. (2016). "Recurrent Marked Temporal Point Processes: Embedding Event History to Vector". In: *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 1555–1564.
- Frauen, Dennis et al. (2023). "Estimating average causal effects from patient trajectories". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 6, pp. 7586–7594.

References IV

- Gao, Tian et al. (2020). "A Multi-Channel Neural Graphical Event Model with Negative Evidence". In: *Proc. of AAAI Conference on Artificial Intelligence*. Vol. 34. 04, pp. 3946–3953.
- Gao, Tian et al. (2021). "Causal inference for event pairs in multivariate point processes". In: *Advances in Neural Information Processing Systems* 34, pp. 17311–17324.
- Gu, Yulong (2021). "Attentive Neural Point Processes for Event Forecasting". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 9, pp. 7592–7600.
- Gunawardana, Asela, Christopher Meek, and Puyang Xu (2011). "A Model for Temporal Dependencies in Event Streams". In: *Advances in Neural Information Processing Systems 24 (NIPS)*.
- Lok, Judith J (2008). "Statistical modeling of causal effects in continuous time". In.

References V

- Meek, Christopher (2014). "Toward Learning Graphical and Causal Process Models.". In: *CI at UAI*, pp. 43–48.
- Mei, Hongyuan and Jason Eisner (2016). "The Neural Hawkes Process: A Neurally Self-modulating Multivariate Point Process". In: *arXiv preprint arXiv:1612.09328*.
- Mei, Hongyuan, Chenghao Yang, and Jason Eisner (2022). "Transformer embeddings of irregularly spaced events and their participants". In: *International conference on learning representations*.
- Melnychuk, Valentyn, Dennis Frauen, and Stefan Feuerriegel (2022). "Causal transformer for estimating counterfactual outcomes". In: *International Conference on Machine Learning*. PMLR, pp. 15293–15329.
- Mogensen, Søren Wengel and Niels Richard Hansen (2020). "Markov equivalence of marginalized local independence graphs". In: *The Annals of Statistics* 48.1, pp. 539–559.

References VI

- Mogensen, Søren Wengel, Daniel Malinsky, and Niels Richard Hansen (2018). "Causal Learning for Partially Observed Stochastic Dynamical Systems". In: *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence (UAI)*.
- Noorbakhsh, Kimia and Manuel Rodriguez (2022). "Counterfactual temporal point processes". In: *Advances in Neural Information Processing Systems* 35, pp. 24810–24823.
- Rajaram, S., T. Graepel, and R. Herbrich (2005). "Poisson-networks: A model for structured point processes". In: *Proceedings of International Workshop on Artificial Intelligence and Statistics*, pp. 277–284.
- Røysland, Kjetil (2012). "Counterfactual analyses with graphical models based on local independence". In: *Annals of Statistics* 40.4, pp. 2162–2194.

References VII

- Schulam, Peter and Suchi Saria (2017). "Reliable decision support using counterfactual models". In: *Advances in neural information processing systems* 30.
- Schweder, Tore (1970). "Composable Markov Processes". In: *Journal of Applied Probability* 7.2, pp. 400–410.
- Shou, Xiao et al. (2023a). "Concurrent multi-label prediction in event streams". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. 8, pp. 9820–9828.
- Shou, Xiao et al. (2023b). "Influence-Aware Attention for Multivariate Temporal Point Processes". In: *2nd Conference on Causal Learning and Reasoning*.
- Spirtes, Peter, Clark Glymour, and Richard Scheines (2000). *Causation, Prediction, and Search*. MIT Press.
- Vaswani, Ashish et al. (2017). "Attention is all you need". In: *Advances in neural information processing systems* 30.

References VIII

- Xia, Wenwen, Yuchen Li, and Shenghong Li (2022). "Graph neural point process for temporal interaction prediction". In: *IEEE Transactions on Knowledge and Data Engineering* 35.5, pp. 4867–4879.
- Xiao, Shuai et al. (2017). "Modeling the Intensity Function of Point Process Via Recurrent Neural Networks". In: *Proc. of Conf. on Artificial Intelligence (AAAI)*, pp. 1597–1603.
- Xiao, Shuai et al. (2019). "Learning time series associated event sequences with recurrent point process networks". In: *IEEE transactions on neural networks and learning systems* 30.10, pp. 3124–3136.
- Zhang, Qiang, Aldo Lipani, and Emine Yilmaz (2021). "Learning neural point processes with latent graphs". In: *Proceedings of the Web Conference 2021*, pp. 1495–1505.

References IX

- Zhang, Qiang et al. (2020a). "Self-attentive Hawkes Process". In: *International Conference on Machine Learning*. PMLR, pp. 11183–11193.
- Zhang, Wei et al. (2020b). "Cause: Learning granger causality from event sequences using attribution methods". In: *International Conference on Machine Learning*. PMLR, pp. 11235–11245.
- Zhang, Yizhou, Defu Cao, and Yan Liu (2022). "Counterfactual neural temporal point process for estimating causal influence of misinformation on social media". In: *Advances in Neural Information Processing Systems* 35, pp. 10643–10655.
- Zhang, Yunhao and Junchi Yan (2021). "Neural Relation Inference for Multi-dimensional Temporal Point Processes via Message Passing Graph.". In: *IJCAI*, pp. 3406–3412.

References X

- Zhou, K., H. Zha, and L. Song (2013). "Learning triggering kernels for multi-dimensional Hawkes processes". In: *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 1301–1309.
- Zuo, Simiao et al. (2020). "Transformer Hawkes Process". In: *International Conference on Machine Learning*. PMLR, pp. 11692–11702.