CausalNET: Unveiling Causal Structures on Event Sequences by Topology-Informed Causal Attention

Hua Zhu[1], **Hong Huang**[1*], **Kehan Yin**[1], **Zejun Fan**[1], **Hai Jin**[1] and **Bang Liu**[2]
[1]National Engineering Research Center for Big Data Technology and System
Services Computing Technology and System Lab, Cluster and Grid Computing Lab
School of Computer Science and Technology
Huazhong University of Science and Technology, Wuhan, China
[2]DIRO, Université de Montréal & Mila & Canada CIFAR AI Chair, Canada
{huazhu, honghuang, kehanyin, zejunfan, hjin}@hust.edu.cn, bang.liu@umontreal.ca

Abstract

Causal discovery on event sequences holds a pivotal significance across domains such as healthcare, finance, and industrial systems. The crux of this endeavor lies in unraveling causal structures among event types, typically portrayed as *directed acyclic graphs* (DAGs). Nonetheless, prevailing methodologies often grapple with untenable assumptions and intricate optimization hurdles. To address these challenges, we present a novel model named CausalNET. At the heart of CausalNET is a special prediction module based on the Transformer architecture, which prognosticates forthcoming events by leveraging historical occurrences, with its predictive prowess amplified by a trainable causal graph engineered to fathom causal relationships among event types. Furthermore, to augment the predictive paradigm for real-world scenarios, we devise a causal decay matrix to encapsulate the influence of the underlying topological network on causal dependencies among events. During training, we alternatively refine the prediction module and fine-tune the causal graph. Comprehensive evaluation on a spectrum of real-world and synthetic datasets underscores the superior performance and scalability of CausalNET, marking a promising step forward in the realm of causal discovery. Code and Appendix are available at https://github.com/CGCL-codes/CausalNET.

1 Introduction

Causal discovery within event sequence data stands as a critical pursuit with far-reaching implications for a multitude of real-world applications. Its fundamental objective revolves around the discernment of causal relationships between events within sequences. Through this discernment, valuable insights into the underlying generative mechanisms of the data emerge, empowering us to make well-informed decisions. One illustrative context lies in *Artificial Intelligence for IT Operations* (AIOps), where event sequences encapsulate a diverse array of system events, ranging from historical alarm occurrences to user interactions. Grasping the intricate causal connections within these sequences serves as a linchpin for a host of advantages. For instance, by pinpointing the root cause of alarms, optimizing the allocation of resources gains precision, and the ability to anticipate potential system breakdowns becomes feasible [Gong *et al.*, 2023].

Research on causal discovery within event sequences falls into three main categories. Constraint-based methods identify causal graphs via conditional independence tests, e.g., [Spirtes and Glymour, 1991; Runge *et al.*, 2019; Bhattacharjya *et al.*, 2022]. Score-based methods search for an optimal graph using a score tester, as demonstrated in [Bhattacharjya *et al.*, 2018; Zhu *et al.*, 2019]. Granger-based methods [Xu *et al.*, 2016a; Zhang *et al.*, 2020b; Idé *et al.*, 2021] identify causality by evaluating if one event type influences another's prediction. However, these methods usually operate under the assumption that event sequences across topological nodes are independent and identically distributed (i.i.d.), leading them to address events on distinct topological nodes in isolation. Contrarily, some real-world situations often present events influenced not only by events on the same node but also by those from neighboring nodes. Addressing this, *Topological Hawkes Process* (THP) [Cai *et al.*, 2022] is proposed to learn causal relationships from non-i.i.d. event sequence data.

However, several challenges still exist for causal discovery within event sequences in real-world scenarios. Notably, state-of-the-art methods like THP require manual kernel specification for the Hawkes process [Hawkes, 1971], limiting its flexibility in modeling intricate real-world events. Moreover, many approaches aim to obtain an optimal causal graph by searching for a *Directed Acyclic Graph* (DAG) with the highest likelihood within the exponentially expanding DAG space as the number of event types increases. This expansion raises scalability concerns for search-based methods, often yielding suboptimal outcomes [Li *et al.*, 2022]. In a word, prevailing methods are entangled in unrealizable assumptions, constrained model flexibility, and scalability issues stemming from inefficient causal graph optimization.

In our pursuit to refine causal discovery for real-world

*Corresponding author

applications, we have developed the innovative CausalNET model. Positioned at the heart of CausalNET, we propose a causal-attention-based Transformer that targets event prediction. Instead of indiscriminately considering all historical events, it leverages a trainable causal graph among event types to determine events that have a direct causal relationship to the upcoming ones. Besides, we introduce a causal decay matrix to characterize the influence of the underlying topological network on causal dependencies among events, to tackle the problem of causal discovery on non-i.i.d. event sequences. Moving away from traditional search-based methods that grapple with scalability and efficiency concerns, our approach involves a gradient-based causal graph optimization via Gumbel Softmax. During training, we alternate between fine-tuning the Transformer module and the duo of the causal graph and decay matrix. Furthermore, considering causal discovery tasks usually require the final graph to be acyclic, we design a post-processing strategy to prune the causal graph learned from the above training process to be an exact DAG.

Our main contributions are listed as follows:

- We propose CausalNET, a novel model for causal discovery on event sequences with the topological network. It supports flexible event sequence modeling and addresses the constraint of the i.i.d. assumption.
- We devise a causal-based self-attention mechanism, which enables our model to gradually enhance its understanding of the causal relationships among different event types while learning to predict future events.
- We conduct experiments on both real-world datasets and synthetic datasets and demonstrate the superiority of the proposed model. Notably, our model achieves remarkable F1-score enhancements exceeding 10% over the state-of-the-art methods across both real-world datasets.

2 Related Work

Causal Discovery. Causal discovery, i.e., causal structure learning, has emerged as a crucial area of research due to its applications in various domains. General approaches for causal discovery can be grouped into three classes: constraint-based, score-based, and *functional causal model* (FCM)-based. Most constraint-based methods lie in the principle that causal relationships between variables or events can manifest as specific patterns of conditional independence, including the widely used algorithms PC [Spirtes and Glymour, 1991] and PCMCI [Runge *et al.*, 2019]. Score-based methods usually aim to search for a DAG with the best score as the causal graph. However, the search for DAGs is an NP-hard problem [Chickering, 1996] as the number of candidate DAGs increases exponentially with the number of causal graph nodes. This motivates the following works such as NOTEARS [Zheng *et al.*, 2018], DAG-GNN [Zheng *et al.*, 2018] to cast the searching problem as a continuous optimization problem. More recent works also try to solve this problem from the perspective of reinforcement learning [Zhu *et al.*, 2019] and generative flow network [Deleu *et al.*, 2022; Li *et al.*, 2022]. FCM-based methods model the data generation process using a functional causal model and identify causal relationships in data based on it. While traditional methods usually implement the FCMs as simple linear and non-linear models with specific noise such as ICALiNGAM [Shimizu *et al.*, 2006] and DirectLiNGAM [Shimizu *et al.*, 2011], recent studies have begun to use more flexible neural networks, e.g., [He *et al.*, 2021; Cheng *et al.*, 2022].

Causal Discovery on Event Sequence Data. While most of the aforementioned methods assume well-structured data such as variables or time series sampled with a regular time interval [Gong *et al.*, 2023], real-world events usually emerge irregularly and asynchronously. Therefore, these methods usually demonstrate inferior results on event sequence data. To address these challenges, specific causal discovery methods have been developed for event sequences. Among them, Granger causality-based methods are well-developed and can be categorized into two main classes. On the one line, some methods utilize the Hawkes process [Hawkes, 1971] to model the event generation process and infer causal relationships between events. Typical examples include ADM4 [Zhou *et al.*, 2013], MLE-SGL [Xu *et al.*, 2016b], NPHC [Achab *et al.*, 2017], L_0Hawkes [Idé *et al.*, 2021], THP [Cai *et al.*, 2022], and SHP [Qiao *et al.*, 2023]. The Hawkes process is a stochastic process that aims to describe the mutual excitation effects between events through a conditional intensity function [Hawkes, 1971]. Benefiting from the Hawkes process's good interpretability and compatibility with Granger causality, these methods can infer causal relationships between events from the conditional intensity function. On the other line, with the advancement of deep learning, some studies have begun to employ neural point processes to model event sequences and learn causalities [Zhang *et al.*, 2020b].

However, most of these methods still run on the independent and identically distributed assumption of event sequences. In particular, THP attempts to solve this problem by introducing a graph convolution to handle the topological information, but it suffers from inherent flexibility issues due to the manually specified kernel function and efficiency bottleneck due to the inefficient DAGs searching approach. In parallel work, [Liu *et al.*, 2024] proposes the TNPAR model to solve the issues of THP. TNPAR transforms continuous-time event sequences into discrete-time sequences and leverages a *multi-layer perceptron* (MLP) to model the event generation process. However, in practice, this simple MLP-based architecture still exhibits limitations in flexibility and scalability.

3 Problem Definition

Consider a topological network denoted as $G_N = (N, E_N)$, where N represents a collection of topological nodes, and E_N signifies the interconnections among these nodes. Additionally, let D denote the discrete space of DAGs $G_V = (V, E_V)$, where V denotes a set of event types and E_V denotes the direct edges between them. Specifically, $v_j \rightarrow v_i$ denotes a direct edge from v_j to v_i, signifying that a type-v_j event can cause the occurrence of a type-v_i event.

Given an event sequence with L events: $S = \{e_1 = (t_1, v_1, n_1), ..., e_i = (t_i, v_i, n_i), ...\}_{i=1}^{L}$, where $t_i \in T$ indicates the timestamp of the i-th event, $v_i \in V$ stands for its event type, and $n_i \in N$ represents the topological node

where it occurred. We assume that each event instance $e_i = (t_i, v_i, n_i)$ is generated from a latent process of the form:

$$e_i = f_i(parents(e_i), \epsilon) \tag{1}$$

where $parents(e_i)$ denotes the historical events that are of the types $v_j \in V$ with a direct edge $v_j \to v_i \in E_V$. The term ϵ denotes jointly independent noise variables. f_i is a function generating e_i from $parents(e_i)$ and ϵ. The goal of this paper is to learn the causal structure G_V among event types that drive this event generation process from the observable event sequence S and the underlying topological network G_N.

4 CausalNET

In this paper, we propose CausalNET to uncover causal relationships among different event types from event sequences and the underlying topological network (Figure 1). At first, we present a causal-attention-based Transformer to model the latent event generation process. To characterize how causal dependencies among events of diverse types evolve under the impact of the topological network, we introduce a causal decay matrix. Then, we implement a continuous optimization on the distribution of possible causal graphs via Gumbel Softmax. During training, we optimize two main modules alternatively in an iterative framework: the Transformer module, and the causal graph together with the decay matrix.

4.1 Causal-attention-based Transformer

Event Embedding. Given an event sequence with L events: $S = \{e_i\}_{i=1}^{L}$, where $e_i = (t_i, v_i, n_i)$. Let us denote $\hat{V} = [\vec{v_1}, \vec{v_2}, ..., \vec{v_L}] \in \mathbb{R}^{|V| \times L}$ where $\vec{v_i}$ is a one-hot vector of the event type v_i, and $\hat{N} = [\vec{n_1}, \vec{n_2}, ..., \vec{n_L}] \in \mathbb{R}^{|N| \times L}$ where $\vec{n_i}$ is a one-hot vector of the topological node n_i. Embedding representations for the events can be formulated as:

$$X = (X_t + X_v + X_n)^T \tag{2}$$

where $X_v = W_v \hat{V} \in \mathbb{R}^{d \times L}$, $X_n = W_n \hat{N} \in \mathbb{R}^{d \times L}$. Specifically, $W_v \in \mathbb{R}^{d \times |V|}$ and $W_n \in \mathbb{R}^{d \times |N|}$ are trainable embedding matrix for event types and topological nodes respectively. $X_t = [\vec{t_1}, \vec{t_2}, ..., \vec{t_L}] \in \mathbb{R}^{d \times L}$ is the temporal encoding matrix which records the timestamp of each event. In practice, we adopt the temporal encoding procedure proposed by [Zuo *et al.*, 2020], and there are alternatives such as [Zhang *et al.*, 2020a]. Finally, $X = [\vec{x_1}, \vec{x_2}, ..., \vec{x_L}] \in \mathbb{R}^{L \times d}$, and $\vec{x_i}$ represents the embedding of the i-th event in the sequence.

Self-Attention with Causal Graph. Consider a sequence of historical events up to e_i at timestamp t_i, to infer the type-v_{i+1} event's intensity at the next timestamp t_{i+1}, we need to consider the cumulative influence of all types of events before t_{i+1}. [Zuo *et al.*, 2020; Zhang *et al.*, 2020a] propose to compute the pairwise influence of each historical event on the next event via self-attention. However, none of them takes into account the latent causalities that really drive the generation process of the event sequence, and many unnecessary irrelevant events are introduced during prediction. Here, we present a causal-based self-attention mechanism with a causal graph ($M = \{m_{i,j}\}_{i,j=1}^{|V|}$) sampled from the Gumbel-Softmax distribution [Jang *et al.*, 2016] of a causal probability graph (Figure 1). This generates a hidden vector that summarizes the influence of relevant historical events up to t_i:

$$\vec{h}_{t_i} = \sum_{j=s}^{i} \hat{f}(\vec{x_i}, \vec{x_j}) \times g(\vec{x_j}) \tag{3}$$

where t_i is the timestamp of the i-th event, and $\vec{x_i}$ denotes the corresponding event embedding. We consider historical events starting from e_s instead of e_1 because considering historical events that occurred too long ago will not only introduce unnecessary noise but also increase the length of the sequence to be processed. This, in turn, would hurt the efficiency of the model. In practice, we set a hyper-parameter ξ, which means that only the historical events that occurred during $[t_{i+1} - \xi, t_{i+1})$ are considered when predicting the event at t_{i+1}. Especially, the event e_s at t_s and the event e_i at t_i are the first and last ones respectively, during the interval. $g(\cdot) = \vec{x_j} W_V \in \mathbb{R}^{d_v}$ is a linear transformation whose output is the value(**v**) in the attention terminology [Vaswani *et al.*, 2017]. $\hat{f}(\cdot, \cdot)$ is a similarity function between two events:

$$f(\vec{x_i}, \vec{x_j}) = \exp(\frac{\vec{q_i} \cdot \vec{k_j}}{\sqrt{d_k}}) / \sum_{j=s}^{i} \exp(\frac{\vec{q_i} \cdot \vec{k_j}}{\sqrt{d_k}}) \tag{4}$$

$$\hat{f}(\vec{x_i}, \vec{x_j}) = f(\vec{x_i}, \vec{x_j}) \times m_{v_j, v_{i+1}} \tag{5}$$

where $\vec{q_i} = \vec{x_i} W_Q \in \mathbb{R}^{d_k}$, $\vec{k_j} = \vec{x_j} W_K \in \mathbb{R}^{d_k}$, and $\cdot$ denotes dot product. $f(\cdot, \cdot)$ is the softmax function, whose output is a normalized attention score/weight between events. In particular, $m_{v_j, v_{i+1}}$ is used to mask irrelevant historical events for predicting the event at t_{i+1}. Specifically, only the subset of historical events that are of the types $v_j \in V$ with a direct edge $v_j \to v_{i+1}$, i.e., $m_{v_j, v_{i+1}} = 1$, are considered.

The softmax function defined in Equation (4) fully adheres to the philosophy of Transformer Decoder. Specifically, it allocates attention weights to historical events solely based on the pairwise similarity between the current event (e_i) and each historical event ($e_{j \le i}$). However, experiments show that such a softmax function may fail to help the model learn a good causal structure from the event sequence data of complicated real-world scenarios (Appendix B.3). Therefore, we further reformulate the softmax function as follows:

$$f(\vec{x_i}, \vec{x_j}) = \exp(\frac{\vec{q_i} \cdot \vec{k_j}}{\sqrt{d_k}}) / \left(\sum_{j=s}^{i} \exp(\frac{\vec{q_i} \cdot \vec{k_j}}{\sqrt{d_k}}) + \Delta_i \right) \tag{6}$$

where $\Delta_i = \exp(\frac{\vec{q_i} \cdot \vec{k}_{i+1}}{\sqrt{d_k}}) + \hat{n}$, and $\hat{n}$ is the number of the padding events[1]. The first term of Δ_i describes the similarity between the event e_i and e_{i+1}. It plays a similar role as the sampled causal graph, i.e., adjusting the attention weight to each historical event ($e_{j \le i}$) based on the relationship between each historical event and the forthcoming event e_{i+1}. The size of $\hat{n}$ reflects the number of recent historical events for e_{i+1}. When there are few historical events for it, we pad

[1]The embedding representation of a padding event is set to be $\vec{0}$. Since $\exp(\vec{q_i} \cdot \vec{0} / \sqrt{d_k}) = 1$, the sum for $\hat{n}$ padding events will be $\hat{n}$.

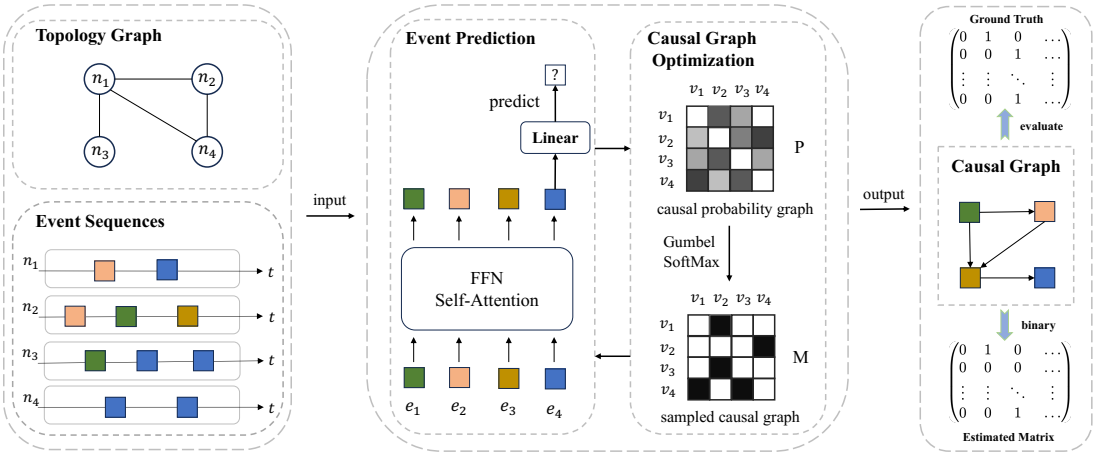


Figure 1: Model framework of CausalNET

the sequence with meaningless events and n will be larger, thus reducing the attention weight for each historical event to prevent overestimating the influence of specific events.

Incorporating the Influence of Topological Network. So far, we have incorporated a trainable causal graph among event types into the attention mechanism via a causal mask after softmax. However, in many real-world scenarios, the structure of the underlying topological network will also influence the generation process of events. [Cai *et al.*, 2022] found that ignoring the topological structure may lead to unobservable confounders and unstable results for causal discovery. Intuitively, events occurring at two topological nodes can only exert causal influence when there is at least one path between the nodes, and the influence should wane as the nodes' distance increases. Therefore, to consider the influence of the underlying topological network, we further extend such a causal attention mechanism to a topology-informed causal attention mechanism via the adjacency matrix among topological nodes and a trainable parameter matrix describing node properties. Specifically, we extend Equation (5) to:

$$\tilde{f}(\vec{x_i}, \vec{x_j}) = \sum_{k=0}^{K} \hat{f}(\vec{x_i}, \vec{x_j}) \times \hat{A}^k_{n_j, n_{i+1}} \times sig(\phi_{k,n_j,n_{i+1},v_j,v_{i+1}}) \quad (7)$$

and replace the similarity function $\hat{f}(\cdot,\cdot)$ in Equation (3) with $\tilde{f}(\cdot,\cdot)$. $\hat{A}^k$ is a binary matrix construted from the adjacency matrix $A \in \mathbb{R}^{|N|\times|N|}$, where $\hat{A}^k_{n_j,n_{i+1}} = 1$ if and only if there is at least one k-hop path between the topological node n_j and n_{i+1}. Besides, to emphasize the influence between events occurring on the same topological node, we set all the diagonal elements of $\hat{A}^k$ to be 1. $\phi \in \mathbb{R}^{K\times|N|\times|N|\times|V|\times|V|}$ is a trainable parameter matrix introduced to characterize the property of the underlying topological network. In particular, $\phi_{k,n_j,n_{i+1},v_j,v_{i+1}}$ describes the decay coefficient of the causal influence from a type-v_j event (on node n_j) to a type-v_{i+1} event (on node n_{i+1}) after propagating along a k-hop path "$n_j \to \ldots \to n_{i+1}$". Given its special positioning, we refer to this matrix as the causal decay matrix. $sig(\cdot)$ is a sigmoid function used to normalize the value of ϕ to (0,1). Intuitively, ϕ plays a similar role to the graph attention score [Veličković *et al.*, 2018] because both of them are used to aggregate the information from the topological neighbors of each node.

In addition to the self-attention module, we pass the hidden vector $\vec{h}_{t_i}$ through a position-wise feed-forward neural network, which consists of two linear transformations with a ReLU activation function in between. This generates a new hidden vector as follows:

$$\vec{h'}_{t_i} = max(0, \vec{h}_{t_i} W_1 + \vec{b}_1) W_2 + \vec{b}_2 \quad (8)$$

Auxiliary Tasks for Transformer Optimization. In order to optimize the causal-based Transformer, we employ two event prediction tasks. The first task is predicting the likelihood of the next event based on historical events. Since we have obtained the hidden representation of historical events, the remaining question is how to map it to the likelihood of the next event. To solve this, we adopt the continuous conditional intensity function proposed by [Zuo *et al.*, 2020]:

$$\lambda_{v,n}(t|H_{t_i}) = f_{v,n}(\alpha_{v,n}\frac{t-t_i}{t_i} + \vec{w}_{v,n} \cdot \vec{h'}_{t_i} + b_{v,n}) \quad (9)$$

for $t \in (t_i, t]$, where $f_{v,n}(x) = \beta_{v,n} log(1 + exp(x/\beta_{v,n}))$ is a softplus function with the softness parameter $\beta_{v,n} > 0$ to eusure a positive intensity. $\alpha_{v,n}$ is used to control the importance of the interpolation between the current timestamp t_i and the future timestamp t, and $b_{v,n}$ represents the occurrence probability of a type-v event at topological node-n without the influence of history information. Given the intensity function as $\lambda_{v,n}(t|H_{t_i})$, the probability density function of a type-v event at topological node-n can be expressed as:

$$p_{v,n}(t|H_{t_i}) = \lambda_{v,n}(t|H_{t_i}) \times exp(-\int_{t_i}^{t} \lambda_{v,n}(t|H_{t_i})\, dt) \quad (10)$$

and the log-likelihood for the whole event sequence $S = \{(t_1, v_1, n_1), ..., (t_i, v_i, n_i), ...\}_{i=1}^{L}$ can be formulated as:

$$\mathcal{L} = \sum_{i=1}^{L-1} \{\log \lambda_{v_{i+1},n_{i+1}}(t_{i+1}|H_{t_i}) - \int_{t_i}^{t_{i+1}} \lambda_{v_{i+1},n_{i+1}}(t|H_{t_i})\, dt\} \quad (11)$$

The logarithm term should be increased to explain why a type-v_{i+1} event actually occurred on topological node-n_{i+1}

at t_{i+1}, and the integral term should be decreased to account for the absence of any event between (t_i, t_{i+1}). [Rasmussen, 2018; Idé *et al.*, 2021] provide detailed mathematical derivations of the probability density and log-likelihood functions.

The second task is to predict the attributes of the next event, including its timestamp, event type, and topological node. In practice, we pass the hidden vector of the i-th event through three independent linear layers to predict the $(i+1)$-th event. This involves regression (timestamp) with squared error loss and classification (type and node) with cross-entropy losses.

4.2 Continuous Optimization of Causal Graph

Due to the large search space of DAGs and complex acyclicity constraints, most search-based methods cannot handle large-scale settings and are prone to getting stuck in local optima. Therefore, recent studies begin to recast the combinatoric search problem of DAGs into continuous optimization problems [Zheng *et al.*, 2018; Ng *et al.*, 2022]. Here, we implement a continuous optimization of the causal graph via Gumbel Softmax [Jang *et al.*, 2016], which provides a differentiable way to draw samples from a discrete distribution. This procedure includes two steps in turn. At first, we leverage the sigmoid function to transform a trainable parameter matrix ϑ into the causal probability graph $P = \{p_{i,j}\}_{i,j=1}^{|V|}$, which characterizes a discrete categorical distribution. Specifically, $p_{i,j}$ represents the probability of the causal edge $v_i \rightarrow v_j$ to be true. Then, we draw a sample of the causal graph from the distribution using Gumbel Softmax as follows:

$$m_{i,j} = \frac{exp((log(p_{i,j}) + g_{i,j})/\tau)}{exp((log(p_{i,j}) + g_{i,j})/\tau) + exp((log(1 - p_{i,j}) + g_{i,j})/\tau)} \quad (12)$$

where τ is the temperature parameter that controls the smoothness of the Gumbel Softmax. Especially, as τ approaches 0, the Gumbel Softmax distribution will be equivalent to the original categorical distribution. In addition, $\{g_{i,j}\}_{i,j=1}^{|V|}$ are i.i.d. samples from the Gumbel(0,1) distribution, where $u \sim$ Uniform(0,1) and $g = -log(-log(u))$.

4.3 Training Procedure

So far, we have introduced a causal-based Transformer and a Gumbel Softmax-based causal graph together with the causal decay matrix. Next, we will describe how these two modules interact with each other during the training process.

As shown in Figure 1, we implement an EM-style training framework like [Idé *et al.*, 2021] and [Cheng *et al.*, 2022] via plugging these two modules into a two-stage iterative framework, and optimize them alternatively. In the first stage, given a causal graph sampled from the distribution of the causal probability graph ($P = \sigma(\vartheta)$) together with the causal decay matrix between topological nodes (ϕ), we train the causal-based Transformer module ($f^{(\theta)}$) to fit the event sequence (S) by minimizing the following loss function:

$$L_{pred}(\theta; G_N, \vartheta, \phi) = -\lambda_1 \mathcal{L} + (L_t + L_v + L_n) \quad (13)$$

where $\mathcal{L}$ is the log-likelihood function defined in Equation (11), and λ_1 a hyper-parameter. L_t, L_v, and L_n are loss functions for predicting the timestamp, event type, and topological node of the next event based on historical events.

In the second stage, given a causal-based Transformer model ($f^{(\theta)}$), we optimize the causal graph (ϑ) together with the causal decay matrix (ϕ) by minimizing the loss function:

$$L_{graph}(\vartheta, \phi; G_N, \theta) = L_{pred} + \lambda_2 ||\sigma(\vartheta)||_1 + \lambda_3 h(\sigma(\vartheta)) \quad (14)$$

where $||\cdot||_1$ is the L1-norm to enforce sparse connections for the learned causal graph. $h(\cdot)$ is a differentiable characterization of graph acyclicity called DAG-ness [Zheng *et al.*, 2018], which is defined as follows:

$$h(A) = tr(e^{A \circ A} - |V|) \quad (15)$$

where $\circ$ denotes Hadamard product, and $h(A) \geq 0$. The matrix $A \in \mathbb{R}^{|V| \times |V|}$ denotes a DAG if and only if $h(A) = 0$.

The discovered causal graph $\hat{O}$ is identified based on the causal probability graph P and the causal decay matrix ϕ:

$$\hat{o}_{i,j} = \mathbb{I}(max\{p_{i,j} \times \phi_{k,m,n,i,j}\} > \varepsilon) \quad (16)$$

Specifically, if $max\{p_{i,j} \times \phi_{k,m,n,i,j}\}$ is penalized to a value below the threshold ε, we deduce that there does not exist a causal relationship $v_i \rightarrow v_j$, and set $\hat{o}_{i,j}$ to be 0.

4.4 Post-processing with Rollback Mechanism

While gradient-based causal graph optimization is more efficient than directly searching DAGs and evaluating them, the downside is that we cannot ensure the acyclicity of the causal graph. To solve this, GOLEM [Ng *et al.*, 2020] proposes to gradually increase a threshold ω and remove all the edges with absolute weights smaller than ω until the pruned graph is acyclic. However, in real-world datasets, significant noise may lead to some negative edges having higher weights than true causal edges. Experimental results show that this pruning strategy might mistakenly remove many edges that should be predicted as positive (Appendix B.2).

Therefore, we design a more flexible pruning strategy. At first, we sort the edges in $\hat{O}$ according to their weights (i.e., $max\{p_{i,j} \times \phi_{k,m,n,i,j}\}$). Then, we remove an edge with the lowest weight and compare the DAG-ness of the pruned graph with the original graph. If removing this edge reduces the graph's DAG-ness, we will keep the deletion and update the graph. Otherwise, we will roll back the operation, i.e., recover the deleted edge. One by one, we will attempt to remove each edge of $\hat{O}$ in ascending order of weight until the pruned graph's DAG-ness decreases to zero, which means that the pruned graph has become an exact DAG.

5 Experimental Setup

Real-world Datasets. We adopt two challenging real-world datasets from telecommunication networks [2]. The first is 24V_439N_Microwave (Micro-24), which has 24 alarm types, 439 topological network elements, and 64,599 alarm events in total. The second is 25V_474N_Microwave (Micro-25), which contains 25 alarm types, 474 topological network elements, and 48,573 alarm events in total. Causal discovery

[2] https://competition.huaweicloud.com/information/1000041487/dataset

on such datasets aims to uncover causal relationships among alarm types. To explore the applicability of our model in other domains (potentially those without topological network underlying event sequences), we further include the IPTV dataset [Luo *et al.*, 2015], which records the history of TV watching behavior of each user and each TV program category denotes an event type. Since the ground-truth causal graph between TV program categories is not available, we conducted qualitative analysis on this dataset (Appendix B.1).

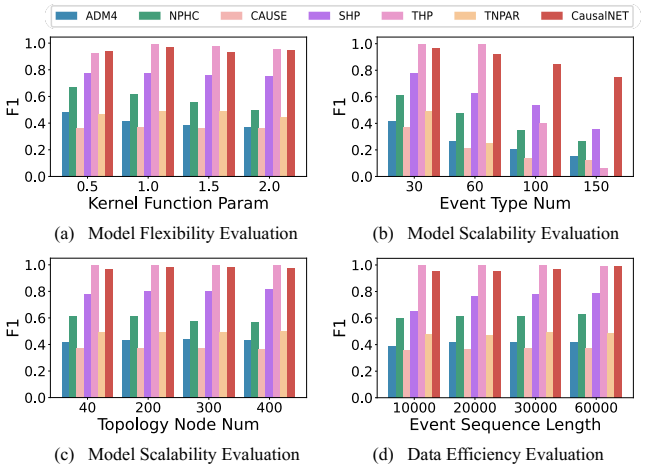
Synthetic Datasets. In addition, we also generate a range of synthetic datasets via gcatle's API [Zhang *et al.*, 2021], which simulates event sequences based on the topological Hawkes process [Cai *et al.*, 2022] engineered with a classical exponential decay kernel function. In the experiments, to generate datasets with different event interactions and temporal effects, we reformulate the original topological Hawkes process by replacing the default kernel function with some other kernel functions from the field of point process and event modeling. The distinctions among these synthetic datasets lie in the kernel function, the number of event types, the number of topological nodes, and the event sequence length (total number of events). Without further specification, each synthetic dataset utilizes an exponential decay function as the kernel, comprising 30 event types, 60 topological nodes, and an event sequence length of 30,000. Detailed statistics of the datasets are shown in Appendix B.4.

Baseline Models. We compare our model against the following 10 baselines: constraint-based methods: PC [Spirtes and Glymour, 1991] and PCMCI [Runge *et al.*, 2019]; score-based method: RL-BIC [Zhu *et al.*, 2019]; FCM-based method: ICALiNGAM [Shimizu *et al.*, 2006]; Granger-based methods: ADM4 [Zhou *et al.*, 2013], NPHC [Achab *et al.*, 2017], CAUSE [Zhang *et al.*, 2020b], SHP [Qiao *et al.*, 2023], THP [Cai *et al.*, 2022], and TNPAR [Liu *et al.*, 2024].

Evaluation Metrics. We adopt 4 widely used evaluation metrics of causal discovery as follows: *F1 Score* (F1), *True Positive Rate* (TPR), *False Positive Rate* (FPR), and *Area Under the Receiver Operating Characteristic Curve* (AUROC).

6 Experimental Results

Results on Telecommunication Network Alarm Datasets. Here we compare the performance of our proposed model CausalNET and its pruned version CausalNET* with all baseline models. Table 1 presents the experimental results for each model on the two real-world datasets. Notably, CausalNET exhibits significant advantages over all baseline algorithms across multiple metrics, with an improvement of over 10% in F1 score compared to the state-of-the-art methods. Moreover, CausalNET demonstrates a remarkable improvement in AUROC. This suggests that CausalNET excels in distinguishing between causally related and non-causally related event types. While some baselines excel in controlling FPR, they present low TPR. NPHC and CAUSE achieve high TPRs, but their FPRs are also extremely high, thus ineffective in discerning real causal relationships. Compared to other baselines, CausalNET, THP, and TNPAR outperform due to their inclusion of the underlying topological network



(a) Model Flexibility Evaluation (b) Model Scalability Evaluation

(c) Model Scalability Evaluation (d) Data Efficiency Evaluation

Figure 2: Results on a range of synthetic datasets

structure, while other baselines run on the i.i.d. assumption of event sequences. This demonstrates the importance of considering topology in causal discovery tasks. Both neural network-based methods CausalNET and TNPAR achieve much higher TPR than the parametric method THP, indicating that neural networks are more effective in capturing intricate event dependencies in real-world scenarios. In addition, while TNPAR uses a simple MLP to model event dependencies, CausalNET employs a causal-attention-based Transformer architecture. The significantly superior performance of CausalNET on both datasets demonstrates the advantages of its causal-attention-based Transformer architecture.

Results on Synthetic Datasets. Then, based on the synthetic datasets, we further evaluate the performance of CausalNET and six baseline models across three dimensions (Figure 2). (I) Flexibility: Figure 2(a) illustrates the performance of each model as the kernel function varies. In this experiment, we introduce Weibull distribution [Rinne, 2008] and modify its shape parameter to alter the kernel function (Appendix B.4). CausalNET and THP, while achieving comparable performance, significantly outperform other baselines. (II) Scalability[3] : As shown in Figure 2(b)[4] and Figure 2(c), across all event type number and topological node number settings, CausalNET exhibits excellent performance and outperforms all baselines except THP. Notably, THP performs well when there are fewer event types, but experiences a notable decline as the number of event types increases to 100. This is because THP needs to search for a DAG with the highest likelihood from the discrete DAGs space, whose size increases exponentially with the growth of event types. In contrast, CausalNET demonstrates outstanding performance on all datasets, as it has converted the NP-hard DAGs searching problem into a much more efficient gradient-based continuous optimization problem in the real-valued space. (III) Data Efficiency: Figure 2(d) indicates that

[3]When the number of events reaches 100, some algorithms fail to complete training within a reasonable timeframe. Therefore, we impose a maximum runtime of 72 hours for each algorithm.

[4]TNPAR encounters out-of-memory (OOM) issues on datasets with 100 and 150 event types on Tesla V100 GPUs (32 GB).

Dataset	Metric	PC	ICALiNGAM	RL-BIC	PCMCI	ADM4	NPHC	CAUSE	SHP	THP	TNPAR	CausalNET	CausalNET*
Micro-24	F1 ↑	0.2478	0.0940	0.2802	0.2524	0.2782	0.3282	0.3474	0.2870	0.3818	0.3459	**0.5016**	**0.4883**
	TPR ↑	0.2044	0.0803	0.2628	0.1898	0.2130	0.3139	0.5109	0.2189	0.3066	0.5240	**0.5839**	**0.5328**
	FPR ↓	0.1340	0.1959	0.1913	0.0979	0.0994	0.1868	0.4465	**0.0956**	**0.0934**	0.4696	0.2323	0.2027
	AUROC ↑	0.5327	0.4422	0.5357	0.5459	0.5555	0.5635	0.5322	0.5608	0.6066	0.5272	**0.6758**	**0.6651**
Micro-25	F1 ↑	0.2143	0.1024	0.2254	0.1576	0.2723	0.3125	0.3226	0.3035	0.3043	0.3413	**0.4742**	**0.4679**
	TPR ↑	0.1419	0.0878	0.2162	0.1081	0.2116	0.4392	0.4730	0.2297	0.2365	0.4423	**0.5270**	**0.4932**
	FPR ↓	**0.0566**	**0.0566**	0.2180	0.0818	0.1063	0.4256	0.4528	0.0880	0.0985	0.3564	0.2159	0.1908
	AUROC ↑	0.5426	0.4464	0.4991	0.5132	0.5523	0.5068	0.5101	0.5689	0.5690	0.5429	**0.6555**	**0.6512**

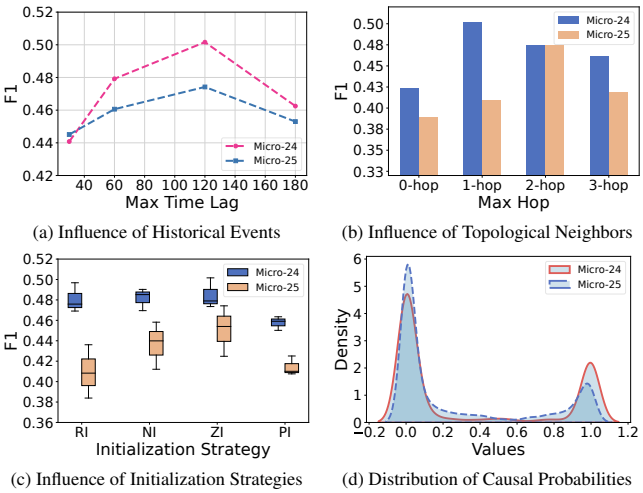
Table 1: Performance comparison on two real-world datasets

CausalNET and THP demonstrate outstanding performance right from the start. Meanwhile, some models (e.g., SHP) are sensitive to the number of events, with their performance improvement relying on an increase in the event number.

Influence of Historical Events. In this section, we study the influence of different receptive fields on historical events. Specifically, we set ξ (max time lag) to $\{30s, 60s, 120s, 180s\}$, and only allow CausalNET to use the most recent historical events within ξ to predict the next future event. As shown in Figure 3(a), on both datasets, CausalNET excels with a 120s time lag, followed by 60s and 180s, while 30s leads to the worst performance. These results emphasize the significance of dataset-specific time lag setting for optimal model performance. On the one hand, a too-short time lag limits the receptive field of self-attention, making it unable to capture long-term dependencies or causal relationships between events. On the other hand, a too-long time lag could introduce additional noise, since usually, only events occurring within a certain time interval will affect each other.

Influence of Topological Neighbors Similarly, we investigate the influence of considering different-order topological neighbors. We set the hyper-parameter k (max hop) to $\{0, 1, 2, 3\}$, where $k = 0$ corresponds to not considering any topological information. As shown in Figure 3(b), on the first real-world dataset, CausalNET performs best when $k = 1$, and closely followed by $k = 2$ and $k = 3$. In particular, $k = 0$ leads to an obvious decline in model performance. On the second dataset, CausalNET gets the best performance when $k = 2$. $k = 1$ and $k = 3$ lead to comparable worse performance. And again, $k = 0$ results in the worst results. This experiment confirms the significance of considering the appropriate topological information in causal discovery tasks.

Influence of Causal Graph Initialization. In this section, we evaluate the effects of different causal graph initialization strategies on the performance of CausalNET. Specifically, for the parameter matrix ϑ, we test four common initialization strategies. *Random Initialization* (RI) sets all parameters with random values within [-1, +1]. *Zero Initialization* (ZI) sets all parameters to be 0. This is a setting without prior assumption, where all edges are initialized to have an equal probability of being true or false. *Negative Initialization* (NI) sets all parameters to -1. This is a setting with explicit assumption, where all edges are initially predicted to have a very small probability of being true. *Positive Initialization* (PI) is opposite to NI. We repeat the experiment of each initialization strategy three times using different random seeds. As shown in Figure 3(c), ZI and NI achieve the best results across both datasets. RI exhibits slightly worse results. Besides, PI demonstrates



(a) Influence of Historical Events

(b) Influence of Topological Neighbors

(c) Influence of Initialization Strategies

(d) Distribution of Causal Probabilities

Figure 3: Results of intrinsic evaluation

notably inferior performance, a shortcoming ascribed to its imposition of an excessively strong prior. Specifically, an excessive number of edges in the causal graph are initialized as true, whereas real-world causal graphs are usually sparse.

Distribution of Causal Probability Values. We display the distribution of the learned probability value of each edge being true (i.e., $max\{p_{i,j} \times \phi_{k,m,n,i,j}\}$) for two real-world datasets in Figure 3(d). The results from both datasets indicate that the vast majority of values are concentrated around either 0 or 1. Therefore, by simply setting an appropriate threshold as indicated in Equation (16), we can exclude a majority of edges and thus obtain a sparse causal structure.

7 Conclusion

Uncovering causal relationships within event sequences in the real world plays a ubiquitous role. In this paper, we propose CausalNET, a novel approach for learning causal structure from event sequences. The core of CausalNET is a causal-attention-based Transformer that predicts future events via attention to historical events under the guidance of a causal graph. The causal graph describes causal relationships among different event types and is trained alternately with the Transformer module in an iterative framework. Extensive experiments on both real-world and synthetic datasets demonstrate that CausalNET achieves superior performance and scalability over a range of existing methods. In the future, we plan to expand this work to handle massive-scale event sequences.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62127808).

References

[Achab *et al.*, 2017] Massil Achab, Emmanuel Bacry, Stéphane Gaıffas, Iacopo Mastromatteo, and Jean-François Muzy. Uncovering causality from multivariate hawkes integrated cumulants. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1–10. PMLR, 2017.

[Bhattacharjya *et al.*, 2018] Debarun Bhattacharjya, Dharmashankar Subramanian, and Tian Gao. Proximal graphical event models. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems*, pages 8136–8145, 2018.

[Bhattacharjya *et al.*, 2022] Debarun Bhattacharjya, Karthikeyan Shanmugam, Tian Gao, and Dharmashankar Subramanian. Process independence testing in proximal graphical event models. In *Proceedings of the 3rd Conference on Causal Learning and Reasoning*, pages 144–161. PMLR, 2022.

[Cai *et al.*, 2022] Ruichu Cai, Siyu Wu, Jie Qiao, Zhifeng Hao, Keli Zhang, and Xi Zhang. Thps: Topological hawkes processes for learning causal structure on event sequences. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[Cheng *et al.*, 2022] Yuxiao Cheng, Runzhao Yang, Tingxiong Xiao, Zongren Li, Jinli Suo, Kunlun He, and Qionghai Dai. Cuts: Neural causal discovery from irregular time-series data. In *Proceedings of the 10th International Conference on Learning Representations*, 2022.

[Chickering, 1996] David Maxwell Chickering. Learning bayesian networks is np-complete. *Learning from data: Artificial intelligence and statistics V*, pages 121–130, 1996.

[Deleu *et al.*, 2022] Tristan Deleu, António Góis, Chris Emezue, Mansi Rankawat, Simon Lacoste-Julien, Stefan Bauer, and Yoshua Bengio. Bayesian structure learning with generative flow networks. In *Uncertainty in Artificial Intelligence*, pages 518–528. PMLR, 2022.

[Gong *et al.*, 2023] Chang Gong, Di Yao, Chuzhe Zhang, Wenbin Li, Jingping Bi, Lun Du, and Jin Wang. Causal discovery from temporal data. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5803–5804, 2023.

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

[He *et al.*, 2021] Yue He, Peng Cui, Zheyan Shen, Renzhe Xu, Furui Liu, and Yong Jiang. Daring: Differentiable causal discovery with residual independence. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 596–605, 2021.

[Idé *et al.*, 2021] Tsuyoshi Idé, Georgios Kollias, Dzung Phan, and Naoki Abe. Cardinality-regularized hawkes-granger model. In *Proceedings of the Thirty-fifth Annual Conference on Neural Information Processing Systems*, pages 2682–2694, 2021.

[Jang *et al.*, 2016] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *Proceedings of the 4th International Conference on Learning Representations*, 2016.

[Li *et al.*, 2022] Wenqian Li, Yinchuan Li, Shengyu Zhu, Yunfeng Shao, Jianye Hao, and Yan Pang. Gflowcausal: Generative flow networks for causal discovery. *arXiv preprint arXiv:2210.08185*, 2022.

[Liu *et al.*, 2024] Yuequn Liu, Ruichu Cai, Wei Chen, Jie Qiao, Yuguang Yan, Zijian Li, Keli Zhang, and Zhifeng Hao. Tnpar: Topological neural poisson auto-regressive model for learning granger causal structure from event sequences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 20491–20499, 2024.

[Luo *et al.*, 2015] Dixin Luo, Hongteng Xu, Yi Zhen, Xia Ning, Hongyuan Zha, Xiaokang Yang, and Wenjun Zhang. Multi-task multi-dimensional hawkes processes for modeling event sequences. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 3685–3691, 2015.

[Ng *et al.*, 2020] Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. On the role of sparsity and dag constraints for learning linear dags. In *Proceedings of the Thirty-fourth Annual Conference on Neural Information Processing Systems*, pages 17943–17954, 2020.

[Ng *et al.*, 2022] Ignavier Ng, Shengyu Zhu, Zhuangyan Fang, Haoyang Li, Zhitang Chen, and Jun Wang. Masked gradient-based causal structure learning. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pages 424–432. SIAM, 2022.

[Qiao *et al.*, 2023] Jie Qiao, Ruichu Cai, Siyu Wu, Yu Xiang, Keli Zhang, and Zhifeng Hao. Structural hawkes processes for learning causal structure from discrete-time event sequences. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*, pages 5702–5710, 2023.

[Rasmussen, 2018] Jakob Gulddahl Rasmussen. Lecture notes: Temporal point processes and the conditional intensity function. *arXiv preprint arXiv:1806.00221*, 2018.

[Rinne, 2008] Horst Rinne. *The Weibull distribution: a handbook*. CRC press, 2008.

[Runge *et al.*, 2019] Jakob Runge, Peer Nowack, Marlene Kretschmer, Seth Flaxman, and Dino Sejdinovic. Detecting and quantifying causal associations in large nonlinear time series datasets. *Science Advances*, 5(11):eaau4996, 2019.

[Shimizu *et al.*, 2006] Shohei Shimizu, Patrik O. Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(10), 2006.

[Shimizu *et al.*, 2011] Shohei Shimizu, Takanori Inazumi, Yasuhiro Sogawa, Aapo Hyvarinen, Yoshinobu Kawahara, Takashi Washio, Patrik O. Hoyer, Kenneth Bollen, and Patrik Hoyer. Directlingam: A direct method for learning a linear non-gaussian structural equation model. *Journal of Machine Learning Research*, 12(Apr):1225–1248, 2011.

[Spirtes and Glymour, 1991] Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9(1):62–72, 1991.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems*, pages 5998–6008, 2017.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proceedings of the Sixth International Conference on Learning Representations*, 2018.

[Xu *et al.*, 2016a] Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. Learning granger causality for hawkes processes. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1717–1726. PMLR, 2016.

[Xu *et al.*, 2016b] Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. Learning granger causality for hawkes processes. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1717–1726. PMLR, 2016.

[Zhang *et al.*, 2020a] Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive hawkes process. In *Proceedings of the 37th International Conference on Machine Learning*, pages 11183–11193. PMLR, 2020.

[Zhang *et al.*, 2020b] Wei Zhang, Thomas Panum, Somesh Jha, Prasad Chalasani, and David Page. Cause: Learning granger causality from event sequences using attribution methods. In *Proceedings of the 37th International Conference on Machine Learning*, pages 11235–11245. PMLR, 2020.

[Zhang *et al.*, 2021] Keli Zhang, Shengyu Zhu, Marcus Kalander, Ignavier Ng, Junjian Ye, Zhitang Chen, and Lujia Pan. gcastle: A python toolbox for causal discovery. *arXiv preprint arXiv:2111.15155*, 2021.

[Zheng *et al.*, 2018] Xun Zheng, Bryon Aragam, Pradeep K. Ravikumar, and Eric P. Xing. Dags with no tears: Continuous optimization for structure learning. In *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems*, pages 9472–9483, 2018.

[Zhou *et al.*, 2013] Ke Zhou, Hongyuan Zha, and Le Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*, pages 641–649. PMLR, 2013.

[Zhu *et al.*, 2019] Shengyu Zhu, Ignavier Ng, and Zhitang Chen. Causal discovery with reinforcement learning. In *Proceedings of the Seventh International Conference on Learning Representations*, 2019.

[Zuo *et al.*, 2020] Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *Proceedings of the 37th International Conference on Machine Learning*, pages 11692–11702. PMLR, 2020.