MOCHA: Discovering Multi-Order Dynamic Causal Structure in Temporal Point Processes

Yunyang Cao[1] **Juekai Lin**[3] **Wenhao Li**[2] **Bo Jin**[1,2,†]
[1] Shanghai Research Institute for Intelligent Autonomous Systems, Tongji University
[2] School of Computer Science and Technology, Tongji University
[3] School of Software Technology, Zhejiang University
[†] Corresponding author: bjin@tongji.edu.cn

Abstract

Modeling event sequences requires understanding when future events will occur and how event types causally influence one another over time. Existing temporal point process (TPP) models typically assume static or first-order dependence structures, which limits their ability to capture the dynamic and multi-order causal mechanisms commonly observed in real-world systems. We propose MOCHA, a **M**ulti-**O**rder **C**ausal **H**ierarchical **A**rchitecture for multivariate TPP that jointly models time-varying causal structure and multi-hop influence propagation in continuous time. MOCHA learns latent dynamic weighted directed acyclic graphs (DAG) over event types, where acyclicity and sparsity constraints promote structurally valid causal graphs. Based on the learned graph, the model decomposes event dynamics into multi-order causal paths, and incorporates both direct influence and indirect propagation into the intensity function. The entire framework is end-to-end differentiable and optimized jointly for event prediction and causal structure discovery. Experiments on seven real-world datasets from different domains show that MOCHA achieves lower negative log-likelihood than the evaluated TPP baselines, while learning dynamic structural patterns consistent with domain knowledge. These results support MOCHA as a framework integrating dynamic causal structure learning with TPP fitting and event prediction.

1 Introduction

Event sequence data, where events occur asynchronously over time, is fundamental for modeling complex systems in domains such as clinical care [1, 2], large language model [3, 4], and embodied AI [5, 6]. Temporal point processes (TPPs) provide a principled framework for modeling such sequences by capturing temporal dependencies among events [7, 8]. Despite their empirical success, most existing TPP models remain weak in structural transparency: they can describe what is likely to happen next, but not clearly reveal the evolving causal structures underlying event interactions.

Discovering causal structure in TPPs aims to uncover how events influence one another beyond surface-level correlations. Our focus is on discovering dynamic causal structures in multivariate TPPs, rather than addressing the full scope of general temporal causal discovery. This setting is important because many real-world event streams exhibit dependencies that evolve over time and propagate through multi-step pathways. However, most existing models rely on restrictive structural assumptions. First, some Hawkes-based formulations [9, 10] estimate a fixed influence matrix that represents pairwise relationships, often resulting in a fully connected graph. Second, Granger causality models [11, 12] infer relationships based on temporal precedence and predictive utility, capturing statistical dependence rather than structural causality. Third, recent methods based on Structural Causal Models (SCMs) [13, 14] can recover Directed Acyclic Graphs (DAGs) with stronger

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

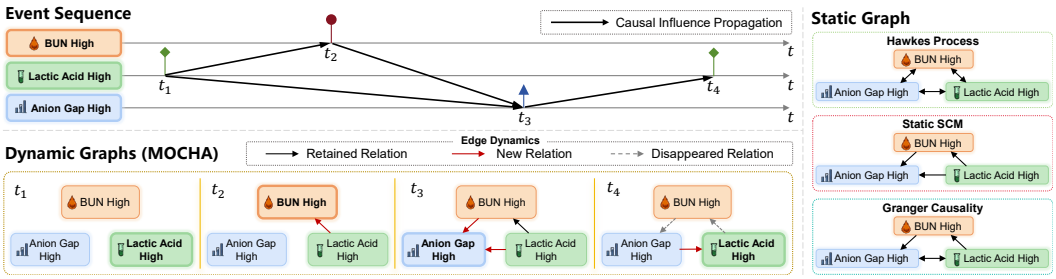

Figure 1: Illustration of an event sequence and its causal graphs. Multi-order paths capture indirect influences over time.

structural semantics, yet they still aggregate observations over time and therefore produce static causal graph. These considerations motivate integrating time-varying graph learning and multi-order influence modeling within a TPP framework.

Real-world causal mechanisms are inherently *dynamic* and *multi-order*. Consider the clinical trajectory of a patient with acute kidney injury in Figure 1, which tracks three interactions. These relationships are dynamic: initially, high Lactic Acid raises Blood Urea Nitrogen (BUN), and subsequently, their interaction drives a high Anion Gap. Later, the Anion Gap feeds back to Lactic Acid. This progression demonstrates that the underlying causal graph is inherently time-dependent. Consequently, approximating these evolving interactions with a single static graph oversimplifies the temporal mechanisms, leading to structural misspecification and suboptimal inference.

This motivates a pivotal question: *How can we simultaneously uncover time-varying causal structures and quantify multi-order influences within a unified continuous-time framework?* Addressing this presents two challenges. First, *structural non-stationarity*: the latent causal graph evolves continuously, making static global optimization ineffective. Second, *multi-order entanglement*: distinguishing direct influences from indirect propagation (e.g., $A \to B \to C$) without excessive computational overhead is non-trivial.

We propose the Multi-Order Causal Hierarchical Architecture (MOCHA), a novel framework that integrates dynamic causal structure learning with TPP fitting and event prediction. MOCHA features a *dynamic causal graph learner* that infers time-dependent weighted DAGs, constrained by acyclicity to enforce structural validity and support a well-defined multi-order decomposition under the model assumptions. Furthermore, it introduces *multi-order causal decomposition* to characterize high-order dependencies as multi-hop paths propagating through the learned dynamic graph. By training end-to-end, MOCHA jointly optimizes the likelihood of observed events and the validity of the causal structures. Our contributions are as follows: **(1) Dynamic causal structure learning:** We develop a dynamic SCM for multivariate TPPs that learns time-varying weighted DAGs from non-stationary event sequences. **(2) Multi-order causal decomposition:** We incorporate both direct effects and indirect multi-hop propagation into the intensity function. **(3) Superior performance:** We train MOCHA end-to-end for event prediction and causal structure learning, achieving dynamic causal patterns on real-world datasets.

2 Related Work

Temporal Point Processes (TPPs) provide a probabilistic framework for modeling event sequences in continuous time. Classical models such as the Poisson process [15], Hawkes process [7], and self-correcting process [16] capture triggering influences through parametric excitation functions. Neural TPPs encode historical events using RNNs [17], LSTMs [9], and attention-based mechanisms [10, 18, 19]. More recent models, such as ITHP [20], NJDTPP [21], and S2P2 [8], further enhance expressiveness by modeling richer event interactions or non-stationary dynamics. These methods improve predictive flexibility but generally do not recover dynamic causal structure over event types.

Causal structure learning in event sequences can be broadly grouped into four categories. Granger-causality-based methods [22], including Hawkes-based extensions [23], RPPN [11], and CAUSE [12], infer directed dependence from temporal precedence and predictive utility. Counterfactual meth-

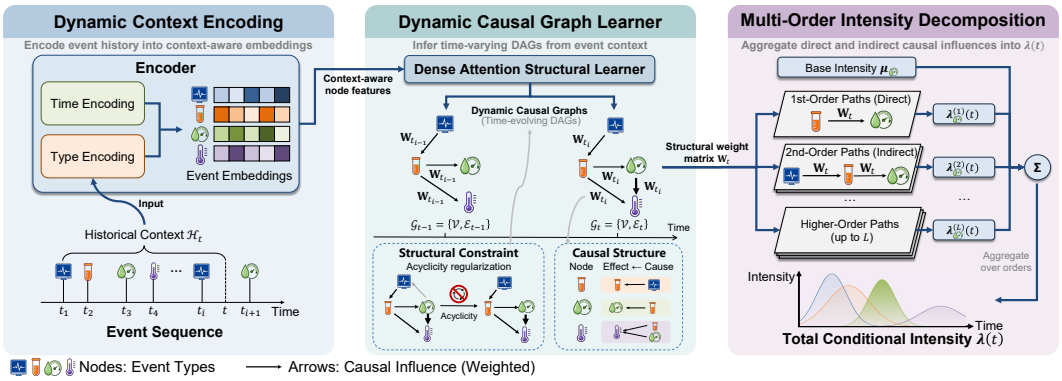

Figure 2: The overall framework of the proposed MOCHA. The architecture consists of three main components: (1) Dynamic Context Encoding, which embeds historical event sequences using time and type encodings; (2) Dynamic Causal Graph Learner, which utilizes a dense attention structural learner to construct time-evolving DAGs while enforcing acyclicity; and (3) Multi-Order Intensity Decomposition, which calculates the total conditional intensity by aggregating the base intensity with 1st-order and higher-order causal influences derived from the learned graph structure.

ods [24–26] are valuable for intervention analysis, yet they are not designed to recover a global causal graph over event types. Static structural methods, including SCM-based models [13, 14, 27] and event-level discovery frameworks [28, 29], use explicit structural assumptions but rely on time-invariant graphs. Dynamic temporal models [30–32] address temporal dependencies, while MOCHA focuses on integrating dynamic graph learning with TPP fitting and event prediction.

3 Methodology

We propose **M**ulti-**O**rder **C**ausal **H**ierarchical **A**rchitecture (MOCHA), a neural framework that integrates multi-order dynamic causal structure learning with TPP fitting and event prediction. Figure 2 shows the framework of the MOCHA.

3.1 Problem Formulation

Multivariate Temporal Point Process. We consider a Multivariate Temporal Point Process (MTPP) characterized by a sequence of discrete events $\mathcal{S} = \{(t_i, k_i)\}_{i=1}^{N}$ observed over a time horizon $[0, T)$, where $t_i \in \mathbb{R}^+$ denotes the occurrence time and $k_i \in \{1, \ldots, K\}$ denotes the event type. The counting process $N_k(t)$ records the number of type-k events up to time t. The dynamics of the process are governed by the conditional intensity function $\lambda_k(t)$, defined as the instantaneous probability of an event of type k occurring at time t given the history $\mathcal{H}_t = \{(t_j, k_j) | t_j < t\}$:

$$\lambda_k(t|\mathcal{H}_t) := \lim_{\Delta t \to 0} \frac{\mathbb{E}[N_k(t + \Delta t) - N_k(t)|\mathcal{H}_t]}{\Delta t}. \tag{1}$$

Dynamic Causal Formulation in TPP. Standard MTPPs often assume a dense interaction matrix, which ignores the causal relationships between events. Recent works incorporate static Structural Causal Models (SCMs) [33], building a static DAG to identify causal relationships. This assumption of topological invariance may be restrictive in many real-world temporal systems, such as medical care, where causal dependencies can evolve or even reverse over time. To capture these evolving mechanisms, we introduce the Dynamic SCM [34] into TPP. We formulate the problem using dynamic structural causality, where the occurrence of an event is governed exclusively by its causal ancestors.

Definition 3.1 (Dynamic Structural Causality in TPP)**.** Let $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$ be a time-varying causal graph over event types $\mathcal{V} = \{1, \ldots, K\}$. For any node $v \in \mathcal{V}$, let $\mathrm{Pa}_t(v) = \{u | (u, v) \in \mathcal{E}_t, u \neq v\}$ denote the set of its parent nodes at time t. A TPP exhibits dynamic structural causality if the conditional intensity of event type v, denoted as $\lambda_v(t)$, depends on the history $\mathcal{H}_t$ solely through the history of its parents and itself:

$$\lambda_v(t|\mathcal{H}_t) = \lambda_v(t \mid \mathcal{H}_t^{\mathrm{Pa}_t(v) \cup \{v\}}), \tag{2}$$

where $\mathcal{H}_t^{\text{Pa}_t(v)\cup\{v\}}$ denotes the sub-history of events associated with the parent set $\text{Pa}_t(v)$ and event type v itself.

Under Definition 3.1, the core challenge lies in modeling the evolution of the parent sets $\text{Pa}_t(v)$. We parameterize the inter-type causal graph $\mathcal{G}_t$ using a structural weight matrix $\mathbf{W}_t \in \mathbb{R}^{K\times K}$. The entry $[\mathbf{W}_t]_{uv}$ represents the instantaneous parent-to-child causal strength from event type u to event type v at time t. This matrix enforces the structural constraints of Definition 3.1 by ensuring that $[\mathbf{W}_t]_{uv} = 0$ whenever $u \notin \text{Pa}_t(v)$, and $[\mathbf{W}_t]_{vv} = 0$ for every v. Concretely, an edge $u \to v$ at time t means that the history of event type u contributes to the conditional intensity of type v after conditioning on the histories of the other modeled parents. The self-history of type v is modeled by a separate type-specific term in the intensity function and therefore does not induce a self-loop in the DAG over event types. MOCHA learns a time-invariant mapping function f_θ, parameterized by θ, to infer the latent causal structure: $\mathbf{W}_t = f_\theta(\mathcal{H}_t)$, subject to $\mathcal{G}_t \in$ DAGs, where $\mathbf{W}_t$ is the corresponding structural weight matrix of graph $\mathcal{G}_t$. Each $\mathcal{G}_t$ is constrained to be a DAG. Under this formulation, the intensity vector of all types depends on the learned structure and is parameterized by ϕ: $\boldsymbol{\lambda}(t) = g_\phi(\mathcal{H}_t, \mathbf{W}_t)$. MOCHA aims to jointly learn the invariant generator f_θ and the intensity dynamics g_ϕ to maximize the likelihood of the observed sequence $\mathcal{S}$, while disentangling multi-order causal paths within $\mathcal{G}_t$.

The causal interpretation of MOCHA relies on some assumptions: structural parent restriction, dynamic DAG, and observational scope. These assumptions and their scope are discussed in Appendix A.

3.2 Dynamic Causal Graph Generation

To implement the invariant mechanism f_θ, we propose a generation module that dynamically infers the causal graph $\mathcal{G}_t$ from the historical context. This process consists of four sequential steps: context encoding, structural weight learning, causal graph extraction, and structural constraints.

Context Encoding. To capture the historical context effectively, we first construct a dynamic representation that integrates both temporal recency and event-type semantics. Let K denote the total number of event types. We represent the most recent occurrence times of all event types as a vector $\mathbf{t}^{last} = (t_1^{last}, \cdots, t_K^{last})^\top$, where $t_k^{last} = \max\{\tilde{t} | (\tilde{t}, k) \in \mathcal{H}_t\}$ denotes the timestamp of the latest event of type k. Specifically, given that the latent state of a specific event type is updated upon each new occurrence, we approximate the cumulative historical influence by focusing solely on the latest event of each type. Hence, we define this condensed set of latest occurrences as $\tilde{\mathcal{H}}_t = \{(t_k^{last}, k) | k = 1, \cdots, K\}$, and hereafter, we employ $\tilde{\mathcal{H}}_t$ as a surrogate for the complete history $\mathcal{H}_t$ in this paper. At the current time t, the time interval since the last occurrence of each event type k is calculated as $\tau_k = t - t_k^{last}$, forming the interval vector $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_K)$.

To encode these continuous time intervals into a high-dimensional space, we employ a sinusoidal positional encoding function $\text{PE} : \mathbb{R} \to \mathbb{R}^{2d}$. For each dimension $i = 1, \ldots, d$, the encoding is defined as: $\text{PE}(\tau_k)[i] = \sin\left(\tau_k / 10000^{i/d}\right)$, $\text{PE}(\tau_k)[i+d] = \cos\left(\tau_k / 10000^{i/d}\right)$. This yields a temporal embedding matrix $\mathbf{E}_t^{(\text{time})} \in \mathbb{R}^{K\times 2d}$, where the k-th row corresponds to $\text{PE}(\tau_k)$. Additionally, to capture the static semantic identity of each event, we assign a learnable type embedding vector $\mathbf{e}_k^{\text{type}} \in \mathbb{R}^{2d}$ to each event type k, forming the matrix $\mathbf{E}^{(\text{type})} \in \mathbb{R}^{K\times 2d}$. The final dynamic embedding $\mathbf{H}_t$ is obtained by summing the temporal and type embeddings:

$$\mathbf{H}_t = \mathbf{E}_t^{(\text{time})} + \mathbf{E}^{(\text{type})} \in \mathbb{R}^{K\times 2d}. \tag{3}$$

Here, $\mathbf{H}_t$ serves as the input features for the subsequent structural learning.

Structural Weight Learning. Based on the dynamic embeddings $\mathbf{H}_t$, we aim to learn the structural weight matrix $\mathbf{W}_t \in \mathbb{R}^{K\times K}$, where each entry $[\mathbf{W}_t]_{uv}$ represents the inter-type causal influence strength from event type u to v at time t. We use a dense attention structural learner with fully connected asymmetric attention to model these pairwise interactions.

First, we compute the attention scores between event types to measure their dependency. We compute attention scores for all ordered pairs of the K event types. We use an asymmetric dense attention mechanism to learn directed pairwise interactions and produce structural weights between event types. For any pair of event types (u, v), the unnormalized attention score $e_{u\to v}(t)$ is computed via

an attention mechanism:

$$e_{u\to v}(t) = \sigma\left(\mathbf{a}^\top[\mathbf{W}_Q\mathbf{h}_v(t) \,\|\, \mathbf{W}_K\mathbf{h}_u(t)]\right), \tag{4}$$

where $\mathbf{h}_u(t)$ and $\mathbf{h}_v(t)$ are the rows of $\mathbf{H}_t$ corresponding to types u and v. $\mathbf{W}_Q, \mathbf{W}_K \in \mathbb{R}^{d'\times 2d}$ are learnable projection matrices, $\mathbf{a} \in \mathbb{R}^{2d'}$ is the attention vector, $\sigma(\cdot)$ is an activation function, such as LeakyReLU, and $\|$ denotes concatenation. We then normalize these scores using the softmax function to obtain the attention weights $\alpha_{u\to v}(t)$: $\alpha_{u\to v}(t) = \frac{\exp(e_{u\to v}(t))}{\sum_{u'=1}^{K}\exp(e_{u'\to v}(t))}$. Next, we aggregate the information from all source types u to the target type v to form a context vector $\tilde{\mathbf{h}}_v(t)$: $\tilde{\mathbf{h}}_v(t) = \sum_{u=1}^{K}\alpha_{u\to v}(t)\cdot\mathbf{h}_u(t)$. Finally, to obtain the structural weights, we project the context vector $\tilde{\mathbf{h}}_v(t)$ into a K-dimensional vector using a linear transformation: $\mathbf{w}_v(t) = \mathbf{W}_{\text{proj}}^{(v)}\tilde{\mathbf{h}}_v^\top(t)$, where $\mathbf{W}_{\text{proj}}^{(v)} \in \mathbb{R}^{K\times 2d}$ is a learnable parameter matrix. The attention weights aggregate event-type context, while the subsequent projection produces the structural weights. By stacking these vectors row-wise, we construct the final dynamic structural weight matrix:

$$\mathbf{W}_t = [\mathbf{w}_1(t), \cdots, \mathbf{w}_K(t)] \in \mathbb{R}^{K\times K}. \tag{5}$$

We interpret $\mathbf{W}_t$ as the parent-to-child structural matrix and enforce $\mathrm{diag}(\mathbf{W}_t) = \mathbf{0}$, so that $\mathbf{W}_t$ only encodes inter-type edges while self-history is modeled separately in the intensity function.

Causal Graph Extraction. MOCHA captures dynamic causal structures by learning time-varying DAGs. While $\mathbf{W}_t$ provides continuous weights, causal reasoning requires a definitive structure to distinguish between existence and non-existence of edges. Given the dynamic graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}_t)$ with weights $\mathbf{W}_t$, the existence of an edge $(u \to v) \in \mathcal{E}_t$ is determined by whether the transformed weight $\sigma(\mathbf{W}_t[u, v])$ exceeds a specific threshold θ. Formally, the binary adjacency matrix $\mathbf{A}_t \in \{0, 1\}^{K\times K}$ of the graph $\mathcal{G}_t$ is derived as:

$$[\mathbf{A}_t]_{uv} = \mathbb{I}(\sigma([\mathbf{W}_t]_{uv}) > \theta), \tag{6}$$

The non-linear activation function is defined as $\sigma(w) = 1 - \exp(-\beta|w|)$, where $|w|$ is the magnitude of the weight, and $\beta > 0$ and $\theta \geq 0$ are fixed hyperparameters. $\mathbb{I}(\cdot)$ is the indicator function. This extraction mechanism distills continuous dynamic weights into causal graphs, providing the structural relationships between events. Throughout the paper, $\mathbf{A}_t$ serves as the interpretable structural summary, whereas $\mathbf{W}_t$ remains the continuous quantity used in intensity computation and optimization.

Structural Constraints. To encourage acyclic and sparse learned dynamic graphs $\mathcal{G}_t$, we impose specific constraints on the structural weight matrix $\mathbf{W}_t$.

Acyclicity Constraint. Following the standard continuous characterization of acyclicity in DAG learning [27], we impose the DAG constraint on each time-indexed structural weight matrix $\mathbf{W}_t$. Specifically, define

$$h(\mathbf{W}_t) := \mathrm{Tr}\left(e^{\mathbf{W}_t\circ\mathbf{W}_t}\right) - K, \tag{7}$$

where $\circ$ denotes the Hadamard product and $\mathrm{Tr}(\cdot)$ is the trace operator. The characterization implies that $h(\mathbf{W}_t) = 0$ if and only if $\mathbf{W}_t$ represents a DAG. Thus, $h(\mathbf{W}_t) = 0$ implies the absence of cycles. For a sequence of observations $\mathcal{S} = \{(t_i, k_i)\}_{i=1}^{N}$, we minimize the following regularization term:

$$\mathcal{L}_{\text{acyclic}} = \gamma_{\text{acyclic}}\sum_{i=1}^{N} h(\mathbf{W}_{t_i}), \tag{8}$$

where γ_{acyclic} is a hyperparameter controlling the strength of the acyclicity constraint.

Sparsity Constraint. This constraint aligns with the assumption that causal influences are selective. To avoid dense graphs, we encourage sparsity in the causal structure. We apply an ℓ_1 regularization on the structural weights, which drives insignificant edge weights toward zero:

$$\mathcal{L}_{\text{sparse}} = \gamma_{\text{sparse}}\sum_{i=1}^{N} |\mathbf{W}_{t_i}|_1, \tag{9}$$

where $|\cdot|_1$ is the element-wise ℓ_1 norm and γ_{sparse} is the regularization coefficient.

3.3 Multi-Order Causal Propagation

Based on the learned dynamic causal graph structure, MOCHA formulates the event dynamics by recursively aggregating causal influences along multi-hop paths. Unlike standard multivariate Hawkes processes that aggregate historical influence into a single term, MOCHA decomposes the intensity function into hierarchical components, ranging from first-order excitation to higher-order recursive propagation restricted by the learned structure.

3.3.1 Learnable Time Decay

The influence of historical events naturally diminishes as time progresses. To capture this temporal dependency, we employ a time decay function $\kappa(\Delta t)$, where $\Delta t = t - t_j$ denotes the time elapsed since a historical event (t_j, k_j). Instead of relying on fixed parametric forms which may lack expressiveness, we introduce a learnable, context-aware decay function. We first map the scalar Δt to a high-dimensional space using sinusoidal positional encodings, $\mathrm{PE} : \mathbb{R} \to \mathbb{R}^{2d}$, to preserve temporal resolution. This embedding is then processed by a lightweight Multi-Layer Perceptron (MLP) to generate a time-varying decay factor: $\kappa(\Delta t) = \text{Sigmoid}\left(\text{MLP}\left(\mathrm{PE}(\Delta t)\right)\right)$. Here, the Sigmoid function ensures the decay factor remains within $(0, 1)$. For numerical stability, we enforce $\kappa(\Delta t) = 0$ when $\Delta t \leq 0$.

3.3.2 Multi-Order Influences

Multi-order influences include first-order and higher-order influences. We formulate the influence propagation using these two components.

First-Order Influence. The first-order influence corresponds to direct excitation from a past event type u to a current type v. This is computed as the product of the structural weight and the temporal decay:

$$\Phi^{(1)}_{u \to v}(t - t_j) = [\mathbf{W}_t]_{uv} \cdot \kappa(t - t_j), \tag{10}$$

where $[\mathbf{W}_t]_{uv}$ denotes the dynamic edge weight $w_{u \to v}(t)$ conditioned on the graph structure $\mathcal{G}_t$.

Higher-Order Influences. Higher-order influences are modeled as propagating through multi-hop paths within a DAG $\mathcal{G}_t$. We define a causal path π of length l as a sequence of event types $(k_0 \to k_1 \to \cdots \to k_l)$, where influence flows recursively. For a specific chain of events $(t_0, k_0) \to (t_1, k_1) \to \cdots \to (t_l, k_l)$ occurring in history, the total influence propagated along this path is the product of dynamic weights and decay kernels at each hop:

$$\Phi^{(l)}_{\pi}(t_l - t_0) = \prod_{r=1}^{l} \left([\mathbf{W}_{t_r}]_{k_{r-1}, k_r} \cdot \kappa(t_r - t_{r-1}) \right). \tag{11}$$

Here, intermediate events act as carriers of causal signals, enabling the transmission of influence across the sequence.

Bounded Maximum Order. Modeling these dependencies via a DAG $\mathcal{G}_t$ imposes a constraint on the maximum order. L denotes the maximum order. Proposition 3.2 describes its upper bound and Appendix B shows the proof.

Proposition 3.2 (Bounded Maximum Order)**.** *Let $\mathcal{G}_t$ be a DAG representing causal relationships among K event types. The maximum order L satisfies $L \leq K - 1$.*

3.4 Multi-Order Intensity

3.4.1 Intensity Function

The total intensity is composed of a base intensity, a self-history term, and the intensities of all orders from 1 to L. The total conditional intensity for an event type k at time t, denoted as $\lambda_k(t|\mathcal{H}_t)$, is constructed by a weighted sum of these components. We define the total intensity function as:

$$\lambda_k(t|\mathcal{H}_t) = \mu_k + \nu_k(t|\mathcal{H}_t^{\{k\}}) + \sum_{l=1}^{L} \alpha_k^{(l)} \lambda_k^{(l)}(t|\mathcal{H}_t), \tag{12}$$

where $\mu_k \in \mathbb{R}^+$ is the learnable base intensity representing the spontaneous occurrence rate of type k. We model the self-history contribution separately as $\nu_k(t|\mathcal{H}_t^{\{k\}}) = \xi_k \cdot \kappa(t - t_k^{last})$, where $\xi_k \in \mathbb{R}^+$ is a learnable type-specific self-history coefficient. $\lambda_k^{(l)}(t|\mathcal{H}_t)$ is the l-th order intensity. The term $\alpha_k^{(l)}$ is a learnable scalar weight representing the contribution of the l-th order intensity. The l-th order intensity $\lambda_k^{(l)}(t|\mathcal{H}_t)$ is composed of the influence of all valid paths of length l. It is computed by aggregating the propagated influences $\Phi_\pi^{(l)}$ defined in Eq.(11):

$$\lambda_k^{(l)}(t|\mathcal{H}_t) = \sum_{\pi \in \mathcal{P}_k^{(l)}(\mathcal{H}_t)} \Phi_\pi^{(l)}(t), \tag{13}$$

where $\mathcal{P}_k^{(l)}(\mathcal{H}_t)$ denotes the set of all realized causal paths of length l ending with event type k before time t.

Efficient Computation via Matrix Formulation. Directly computing the multi-order intensities by summing over all causal paths in Eq.(12) is computationally expensive, with a complexity of $\mathcal{O}(K^L)$ for a graph. To reduce the complexity, we propose a matrix-based iterative algorithm in Proposition 3.3 and prove it in Appendix C. This formulation focuses on the propagation of influence between the most recent events of each type, enabling rapid inference.

Proposition 3.3 (Recursive Intensity Aggregation)**.** *Let $\mathbf{C}_t^{(l)} \in \mathbb{R}^{K\times K}$ be the l-th-order influence matrix at time t. The intensity vector $\boldsymbol{\lambda}(t) = (\lambda_1(t), \cdots, \lambda_K(t))^\top \in \mathbb{R}^K$ is computed via:*

$$\boldsymbol{\lambda}(t) = \boldsymbol{\mu} + \boldsymbol{\nu}(t) + \sum_{l=1}^{L} \boldsymbol{\alpha}^{(l)} \circ \left(\left(\mathbf{C}_t^{(l)} \right)^\top \mathbf{1} \right), \tag{14}$$

where $\boldsymbol{\nu}(t) = \left(\nu_1(t|\mathcal{H}_t^{\{1\}}), \cdots, \nu_K(t|\mathcal{H}_t^{\{K\}})\right)^\top$, $\boldsymbol{\alpha}^{(l)} = (\alpha_1^{(l)}, \cdots, \alpha_K^{(l)})^\top$ denotes the l-th order weighting vector, $\circ$ denotes the Hadamard product, and $\mathbf{1} \in \mathbb{R}^K$ is a column vector of ones. The influence matrices are derived recursively as $\mathbf{C}_t^{(1)} = \mathbf{B}_t$ and $\mathbf{C}_t^{(l+1)} = \mathbf{T}_t \mathbf{C}_t^{(l)}$ for $l \geq 1$. Here, $\mathbf{B}_t$ is the base influence matrix capturing the direct decay from the most recent events to the current time t: $[\mathbf{B}_t]_{uv} = [\mathbf{W}_t]_{uv} \cdot \kappa(t - t_u^{last})$. And $\mathbf{T}_t$ is the temporal propagation matrix capturing the influence between the most recent events of different types: $[\mathbf{T}_t]_{uv} = [\mathbf{W}_t]_{uv} \cdot \kappa(t_v^{last} - t_u^{last})$.

3.4.2 Theoretical Foundation

Complexity Analysis. The recursive intensity aggregation requires $\mathcal{O}(LK^2)$ work per time step, while evaluating the acyclicity penalty requires $\mathcal{O}(K^3)$. A more detailed complexity analysis, together with comparisons to representative TPP baseline methods, is provided in Appendix J.
Gradient Boundedness. Proposition D.1 in Appendix D implies that for finite L and γ, the recursive propagation is numerically stable and does not suffer from the exploding gradient problem.
Existence and Uniqueness of Decomposition. Appendix E analyzes a restricted Hawkes-style setting to show that, under the DAG constraint, a finite and unique multi-order decomposition exists.
Smoothness, Penalty Consistency, and Relation to Training Convergence. A complete theory for the full dynamic neural TPP model needs additional assumptions. We clarify the scope of our theoretical guarantee in Appendix F.

3.5 Training and Inference

MOCHA is trained to model the conditional intensity function $\lambda_k(t)$ for each event type $k \in \{1, \ldots, K\}$, enabling likelihood-based learning of temporal point processes. Given an event sequence $\mathcal{S} = \{(t_i, k_i)\}_{i=1}^N$ observed in $[0, T]$, the negative log-likelihood (NLL) objective is:

$$\mathcal{L}_{\text{NLL}} = -\sum_{i=1}^{N} \log \lambda_{k_i}(t_i|\mathcal{H}_{t_i}) + \sum_{k=1}^{K} \int_0^T \lambda_k(t|\mathcal{H}_t)dt. \tag{15}$$

The first term encourages high intensity at the true event times, while the second term penalizes overprediction by integrating the predicted intensity over time. We compute $\lambda_k(t|\mathcal{H}_t)$ using Eq.(12). The integral is approximated using the trapezoidal rule over discretized time steps. To regularize the

Table 1: Negative Log-Likelihood (NLL) comparison across benchmarks (↓ is better).

	AKI	CAD	SE	ST	RT	TB	SO
RMTPP	5.40	3.26	2.24	2.89	4.42	1.48	2.88
RPPN	4.71	4.89	5.97	1.97	4.93	1.93	3.11
NHP	4.42	2.78	1.44	1.98	4.12	0.74	2.67
SAHP	4.55	2.33	2.63	1.04	4.16	0.76	3.51
THP	4.89	2.06	1.38	1.37	4.54	0.86	2.40
IFLTPP	3.97	2.78	3.40	5.64	3.96	1.13	2.34
AttNHP	5.07	1.81	2.21	2.08	4.27	1.19	2.65
ITHP	4.39	1.96	1.59	0.92	4.22	0.97	3.23
NJDTPP	3.92	2.95	2.38	3.02	4.09	1.39	2.39
S2P2	3.83	1.29	0.79	1.15	3.57	0.01	2.33
MOCHA	**3.80**	**0.69**	**0.70**	**0.91**	**3.51**	**-0.36**	**2.32**

learned graph structure, we incorporate additional regularization terms for acyclicity and sparsity, resulting in the full training objective:

$$\mathcal{L} = \mathcal{L}_{\text{NLL}} + \mathcal{L}_{\text{acyclic}} + \mathcal{L}_{\text{sparse}}, \tag{16}$$

where the latter two terms are defined in Eq.(8) and Eq.(9). The entire model is trained end-to-end using Adam gradient-based optimization [35]. Appendix G shows the algorithmic overview. During inference, MOCHA computes the intensity function $\lambda_k(t)$ by dynamically generating $\mathbf{W}_t$ and $\kappa(\Delta t)$ based on the event history $\mathcal{H}_t$. This enables efficient simulation, forecasting, and causal analysis of future events under complex and evolving dynamics.

4 Experiments

4.1 Experimental Setup

Datasets. Seven real-world datasets: Medical: *AKI* (Acute Kidney Injury), *CAD* (Coronary Artery Disease), *SE* (Sepsis), *ST* (Stroke) [36]; Non-medical: *RT* (Retweet) [37], *TB* (Taobao) [38], *SO* (StackOverflow) [39]. One synthetic benchmark: Appendix H.
Baselines. We compare *MOCHA* with a variety of TPP models: *RMTPP* [17], *RPPN* [11], *NHP* [9], *SAHP* [18], *THP* [10], *IFLTPP* [40], *AttNHP* [19], *ITHP* [20], *NJDTPP* [21], *S2P2* [8].
Evaluation Metrics. We evaluate MOCHA from both predictive and structural perspectives. In the main text, we report NLL, causal path matching, and qualitative graph visualizations; RMSE and type accuracy are deferred to Appendix I. Because real-world datasets lack ground-truth graphs, we further include synthetic recovery results in Appendix H, sensitivity analyses in Appendix K, and detailed settings in Appendix N. These experiments answer four research questions (RQ).

RQ1: Can Modeling Multi-Order Dynamics Improve Performance?

To evaluate the effectiveness of modeling multi-order dynamic causal dependencies, we compare MOCHA with state-of-the-art TPP models on a diverse set of clinical and social datasets. As shown in Table 1, MOCHA consistently achieves the lowest NLL across all seven datasets, indicating an improved capacity to model complex event sequences. The performance gain is especially pronounced on clinical datasets (AKI, CAD, SE, ST), where long-range dependencies and evolving interactions are common. These results suggest that MOCHA is better at leveraging rich and temporally extended medical histories to estimate the likelihood of future clinical events.

These results confirm that explicitly modeling multi-order dynamic causal structure can enhance the performance of model fitting. By jointly learning structural weights and adaptive time decay, MOCHA enhances its ability to model complex temporal dependencies.

RQ2: How Do Multi-Order and Dynamic Structures Influence Performance?

To disentangle the impact of multi-order and dynamic structures in MOCHA, we conduct an ablation study. Starting from a univariate Hawkes process (HP), we compare static first-order structure (MHP), static multi-order structure (MHP+MO), dynamic first-order structure (MHP+Dyn), and the full dynamic multi-order model (MOCHA). The results are summarized in Table 2.

Table 2: Ablation of multi-order and dynamic components on NLL (↓ is better).

Model Variant	AKI	CAD	SE	ST	RT	TB	SO
HP	5.88	2.29	3.06	3.14	4.67	1.31	5.38
MHP	5.73	2.26	2.77	2.67	4.27	0.04	3.81
MHP+MO	5.71	1.66	2.16	2.27	4.25	-0.02	3.64
MHP+Dyn	5.29	2.10	1.77	2.21	3.78	0.01	2.37
MOCHA (MHP+MO+Dyn)	**3.80**	**0.69**	**0.70**	**0.91**	**3.51**	**-0.36**	**2.32**

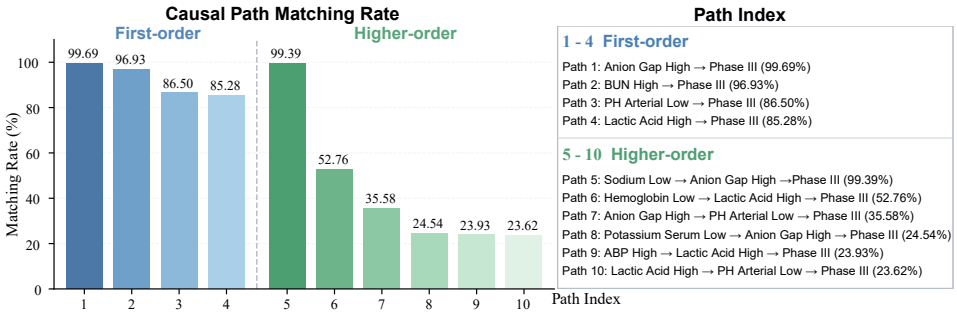

Figure 3: Matching rate of each clinical causal path.

Adding multi-order propagation to MHP lowers NLL, especially on CAD, SE, and ST, highlighting the role of higher-order dependencies. Dynamic structure learning also consistently improves MHP, particularly on AKI, SE, RT, and SO, where interactions vary over time. MOCHA achieves the best NLL on all datasets, showing that combining both components is necessary to realize the full benefit of the framework. Detailed results are provided in Appendix L.

RQ3: Are MOCHA's Learned Paths Consistent with Clinical Knowledge?

To assess the clinical consistency of the learned dynamic graphs, we compare them against 10 clinically supported AKI Phase III pathways (4 first-order and 6 higher-order). Detailed path definitions and supporting evidence are deferred to Appendix O. A path is counted as matched if all of its constituent edges appear in the inferred DAG $\mathcal{G}_t$ when Phase III occurs, and we report the corresponding matching rate. Some paths are only relevant to specific patient subgroups, so the maximum matching rate is below 100%. In general, higher matching rates indicate closer agreement with clinical knowledge.

Figure 3 visualizes the matching rates. The average matching rate for first-order causal relationships is 92.10%, reflecting the model's high agreement with the selected first-order clinical paths. More importantly, MOCHA also shows high agreement with selected higher-order clinical paths, with Sodium Low → Anion Gap High → Phase III achieving a 99.39% matching rate. These results indicate that MOCHA learns dynamic paths consistent with clinical knowledge and provides useful insights for temporal reasoning.

RQ4: Can MOCHA Suggest Clinical Hypotheses?

To assess whether MOCHA can suggest potentially relevant clinical hypotheses, we conduct a case analysis on an AKI patient, focusing on the final six time steps preceding Phase III onset. The learned causal graphs at each time step are visualized in Figure 4, illustrating how the inferred structures evolve as the patient's condition deteriorates.

One representative third-order path discovered by MOCHA is: Sodium Low → Anion Gap High → Lactic Acid High → PH Arterial Low. MOCHA infers this indirect path at multiple time steps: Lactic Acid High appears as an intermediate node in the inferred path that gradually bridges early abnormalities to the eventual drop in arterial pH, eventually leading to AKI Phase III. Recent clinical research [41] proposes this mechanism of lactic acidosis. Importantly, MOCHA identifies such multi-order causal paths without human intervention. The emergence of such a consistent, non-trivial pattern suggests that MOCHA can suggest candidate clinical hypotheses for further validation.

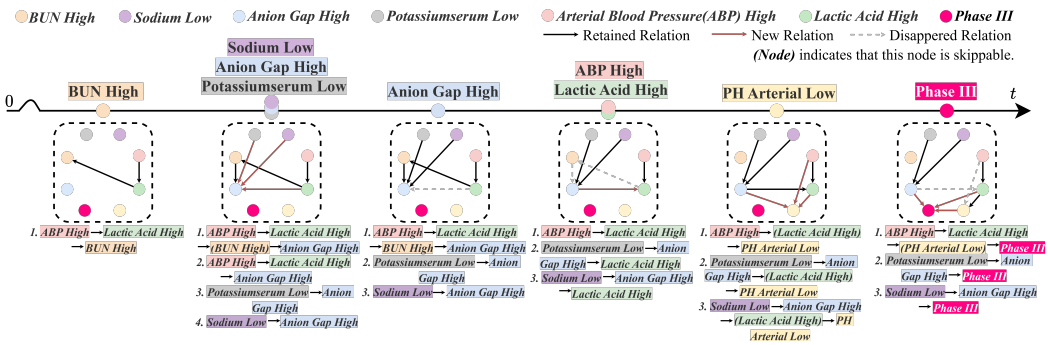

Figure 4: Dynamic causal graphs and multi-order causal paths discovered by MOCHA for an AKI patient. MOCHA identifies increasingly complex structures. In the last three time steps, the first causal path ABP High → Lactic Acid High → PH Arterial Low → Phase III shows progressive expansion from a first-order path to a third-order path, illustrating how higher-order relations emerge and stabilize as the patient's condition deteriorates.

5 Conclusion

We presented MOCHA, a neural framework integrating dynamic causal structure learning with TPP fitting and event prediction. MOCHA achieves strong predictive performance on the evaluated benchmarks and learns time-varying structural patterns consistent with clinical knowledge.

Acknowledgements

This work was supported by the National Science and Technology Major Project (2026ZD0124703), the National Natural Science Foundation of China (62406270) and the STCSM Shanghai Rising-Star Program (24YF2748800).

References

[1] Peter Schulam and Suchi Saria. Reliable decision support using counterfactual models. *Advances in Neural Information Processing Systems*, 30, 2017.

[2] Yunyang Cao, Juekai Lin, Hongye Wang, Wenhao Li, and Bo Jin. Interpretable hybrid-rule temporal point processes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 443–459. Springer, 2025.

[3] Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. Agenttuning: Enabling generalized agent abilities for LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3053–3077, 2024.

[4] Zitao Song, Chao Yang, Chaojie Wang, Bo An, and Shuang Li. Latent logic tree extraction for event sequence explanation from LLMs. In *International Conference on Machine Learning*, pages 46238–46258. PMLR, 2024.

[5] Changyeon Kim, Haeone Lee, Younggyo Seo, Kimin Lee, and Yuke Zhu. DEAS: DEtached value learning with action sequence for scalable offline RL. In *International Conference on Learning Representations*, 2026.

[6] Sandeep Routray, Hengkai Pan, Unnat Jain, Shikhar Bahl, and Deepak Pathak. ViPRA: Video prediction for robot actions. In *International Conference on Learning Representations*, 2026.

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

[8] Yuxin Chang, Alex James Boyd, Cao Xiao, Taha Kass-Hout, Parminder Bhatia, Padhraic Smyth, and Andrew Warrington. Deep continuous-time state-space models for marked event sequences. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[9] Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in Neural Information Processing Systems*, 30, 2017.

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

[11] Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen Chu. Modeling the intensity function of point process via recurrent neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

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

[13] Hua Zhu, Hong Huang, Kehan Yin, Zejun Fan, Hai Jin, and Bang Liu. CausalNET: unveiling causal structures on event sequences by topology-informed causal attention. In *International Joint Conference on Artificial Intelligence*, pages 7144–7152, 2024.

[14] Songyao Jin and Biwei Huang. Causal structure learning in hawkes processes with complex latent confounder networks. In *International Conference on Learning Representations*, 2026.

[15] Dale W Jorgenson. Multiple regression analysis of a poisson process. *Journal of the American Statistical Association*, 56(294):235–245, 1961.

[16] Valerie Isham and Mark Westcott. A self-correcting point process. *Stochastic Processes and Their Applications*, 8(3):335–347, 1979.

[17] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1555–1564, 2016.

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

[19] Chenghao Yang, Hongyuan Mei, and Jason Eisner. Transformer embeddings of irregularly spaced events and their participants. In *International Conference on Learning Representations*, 2022.

[20] Zizhuo Meng, Ke Wan, Yadong Huang, Zhidong Li, Yang Wang, and Feng Zhou. Interpretable transformer hawkes processes: Unveiling complex interactions in social networks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2200–2211, 2024.

[21] Shuai Zhang, Chuan Zhou, Yang Aron Liu, Peng Zhang, Xixun Lin, and Zhi-Ming Ma. Neural jump-diffusion temporal point processes. In *International Conference on Machine Learning*, pages 60541–60557. PMLR, 2024.

[22] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969.

[23] Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. Learning granger causality for hawkes processes. In *International Conference on Machine Learning*, pages 1717–1726. PMLR, 2016.

[24] Tian Gao, Dharmashankar Subramanian, Debarun Bhattacharjya, Xiao Shou, Nicholas Mattei, and Kristin P Bennett. Causal inference for event pairs in multivariate point processes. *Advances in Neural Information Processing Systems*, 34:17311–17324, 2021.

[25] Kimia Noorbakhsh and Manuel Rodriguez. Counterfactual temporal point processes. *Advances in Neural Information Processing Systems*, 35:24810–24823, 2022.

[26] Yizhou Zhang, Defu Cao, and Yan Liu. Counterfactual neural temporal point process for estimating causal influence of misinformation on social media. *Advances in Neural Information Processing Systems*, 35: 10643–10655, 2022.

[27] Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems*, 31, 2018.

[28] Amirkasra Jalaldoust, Kateřina Hlaváčková-Schindler, and Claudia Plant. Causal discovery in hawkes processes by minimum description length. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6978–6987, 2022.

[29] 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 *International Joint Conference on Artificial Intelligence*, pages 5702–5710, 2023.

[30] Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000.

[31] Sikun Yang and Hongyuan Zha. A variational autoencoder for neural temporal point processes with dynamic latent graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 16343–16351, 2024.

[32] Sindy Löwe, David Madras, Richard Zemel, and Max Welling. Amortized causal discovery: Learning to infer causal graphs from time-series data. In *Conference on Causal Learning and Reasoning*, pages 509–525. PMLR, 2022.

[33] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, Cambridge, UK, 2 edition, 2009.

[34] Paul K Rubenstein, Stephan Bongers, Bernhard Schölkopf, and Joris M Mooij. From deterministic odes to dynamic structural causal models. In *Conference on Uncertainty in Artificial Intelligence*, pages 114–123, 2018.

[35] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[36] Alistair E W Johnson, David J Stone, Leo A Celi, and Tom J Pollard. The MIMIC code repository: Enabling reproducibility in critical care research. *Journal of the American Medical Informatics Association*, 25(1):32–39, 2018.

[37] Ke Zhou, Hongyuan Zha, and Le Song. Learning triggering kernels for multi-dimensional hawkes processes. In *International Conference on Machine Learning*, pages 1301–1309. PMLR, 2013.

[38] Siqiao Xue, Xiaoming Shi, James Zhang, and Hongyuan Mei. Hypro: A hybridly normalized probabilistic model for long-horizon prediction of event sequences. *Advances in Neural Information Processing Systems*, 35:34641–34650, 2022.

[39] Jure Leskovec and Rok Sosič. Snap: A general-purpose network analysis and graph-mining library. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(1):1, 2016.

[40] Oleksandr Shchur, Marin Biloš, and Stephan Günnemann. Intensity-free learning of temporal point processes. In *International Conference on Learning Representations*, 2020.

[41] Jeffrey A Kraut and Nicolaos E Madias. Lactic acidosis. *New England Journal of Medicine*, 371(24): 2309–2319, 2014.

[42] Siqiao Xue, Xiaoming Shi, Zhixuan Chu, Yan Wang, Hongyan Hao, Fan Zhou, Caigao Jiang, Chen Pan, James Y. Zhang, Qingsong Wen, Jun Zhou, and Hongyuan Mei. Easytpp: Towards open benchmarking temporal point processes. In *International Conference on Learning Representations*, 2024.

[43] Arif Khwaja. KDIGO clinical practice guidelines for acute kidney injury. *Nephron Clinical Practice*, 120 (4):c179–c184, 2012.

[44] Heng-Chih Pan, Tao-Min Huang, Chiao-Yin Sun, Nai-Kuan Chou, Chun-Hao Tsao, Fang-Yu Yeh, Tai-Shuan Lai, Yung-Ming Chen, and Vin-Cent Wu. Predialysis serum lactate levels could predict dialysis withdrawal in type 1 cardiorenal syndrome patients. *EClinicalMedicine*, 44, 2022.

[45] Xi Zhou, Yanlei He, Long Hu, Qianli Zhu, Qingcheng Lin, Xia Hong, Weijian Huang, Peiren Shan, and Dongjie Liang. Lactate level and lactate clearance for acute kidney injury prediction among patients admitted with st-segment elevation myocardial infarction: a retrospective cohort study. *Frontiers in cardiovascular medicine*, 9:930202, 2022.

[46] Meng Chen and Dezhao Lv. Prognostic value of serum lactate level for mortality in patients with acute kidney injury. *European Journal of Medical Research*, 29(1):295, 2024.

Appendix Contents

- **A.** Assumptions and Scope
- **B.** Proof of Proposition 3.2 Bounded Maximum Order
- **C.** Proof of Proposition 3.3 Recursive Intensity Aggregation
- **D.** Gradient Boundedness
- **E.** Causal Decomposition in TPP
- **F.** Smoothness, Penalty Consistency, and Relation to Training Convergence
- **G.** Algorithmic Overview of MOCHA
- **H.** Experiments on Synthetic Data
- **I.** Quantitative Results on Benchmark Datasets
- **J.** Complexity Analysis and Training Runtime
- **K.** Hyperparameter Sensitivity Analysis
  - **K.1** Sensitivity to Maximum Order L
  - **K.2** Sensitivity to Sparsity Coefficient γ_{sparse}
  - **K.3** Sensitivity to Acyclicity Coefficient $\gamma_{acyclic}$
- **L.** Detailed Ablation Study
- **M.** Effect of Retaining Multiple Recent Events
- **N.** Implementation and Training Details
  - **N.1** Datasets and Preprocessing
  - **N.2** Evaluation Metrics
  - **N.3** Model Hyperparameters
- **O.** Literature Basis of AKI Causal Structures
- **P.** Use of LLMs

A Assumptions and Scope

The causal interpretation of MOCHA is tied to explicit modeling assumptions. MOCHA operates within an observational scope and does not establish intervention-level causal identification.

Assumption A.1 (Structural Parent Restriction)**.** For each event type v and time t, the conditional intensity $\lambda_v(t)$ depends on the past only through the modeled parent set $\mathrm{Pa}_t(v)$ and the self-history of type v, as stated in Definition 3.1.

Assumption A.2 (Dynamic DAG Structure)**.** For each time t, the latent structural graph $\mathcal{G}_t$ is acyclic. This assumption supports finite multi-order decomposition and yields a well-defined time-varying structural summary.

Assumption A.3 (Observational Scope)**.** The method is learned from observational event sequences. It does not model interventions, counterfactual queries, or latent confounders, and therefore its outputs should be interpreted as model-dependent dynamic structural summaries under the assumptions above, rather than uniquely identified causal graphs in arbitrary observational settings.

B Proof of Proposition 3.2 Bounded Maximum Order

Proposition B.1 (Bounded Maximum Order)**.** *Let $\mathcal{G}_t$ be a DAG representing causal relationships among K event types. The maximum order L satisfies $L \leq K - 1$.*

Proof. Let $\mathbf{A} \in \{0,1\}^{K\times K}$ be the adjacency matrix of $\mathcal{G}_t$, where $\mathbf{A}_{ij} = 1$ denotes a causal link $i \to j$. The entry $[\mathbf{A}^n]_{ij}$ represents the number of paths of length n from node i to node j.

Since $\mathcal{G}_t$ is a DAG, there exists a permutation matrix P, corresponding to a topological ordering, such that $\mathbf{U} = \mathbf{P}\mathbf{A}\mathbf{P}^\top$ is strictly upper triangular. The characteristic polynomial of a strictly upper triangular matrix with zeros on the diagonal is:

$$\det(\lambda \mathbf{I} - \mathbf{A}) = \lambda^K, \tag{17}$$

where λ is a variable, and $\mathbf{I}$ is the identity matrix of order K. By the Cayley-Hamilton theorem, $\mathbf{A}$ satisfies its own characteristic equation:

$$\mathbf{A}^K = \mathbf{0}. \tag{18}$$

Consequently, for any order $n \geq K$, the number of paths is 0. Therefore, the maximum order for a non-zero path L satisfies $L \leq K - 1$. □

C Proof of Proposition 3.3 Recursive Intensity Aggregation

Proposition C.1 (Recursive Intensity Aggregation)**.** *Let $\mathbf{C}_t^{(l)} \in \mathbb{R}^{K \times K}$ be the l-th-order influence matrix at time t. The intensity vector $\boldsymbol{\lambda}(t) = (\lambda_1(t), \cdots, \lambda_K(t))^\top \in \mathbb{R}^K$ is computed via:*

$$\boldsymbol{\lambda}(t) = \boldsymbol{\mu} + \boldsymbol{\nu}(t) + \sum_{l=1}^{L} \boldsymbol{\alpha}^{(l)} \circ \left(\left(\mathbf{C}_t^{(l)} \right)^\top \mathbf{1} \right), \tag{19}$$

where $\boldsymbol{\nu}(t) = \left(\nu_1(t|\mathcal{H}_t^{\{1\}}), \cdots, \nu_K(t|\mathcal{H}_t^{\{K\}}) \right)^\top$ collects the self-history terms, $\boldsymbol{\alpha}^{(l)}$ denotes the l-th order weighting vector, $\circ$ denotes the Hadamard product, and $\mathbf{1}$ is a column vector of ones. The influence matrices are derived recursively:

$$\mathbf{C}_t^{(1)} = \mathbf{B}_t, \tag{20}$$

$$\mathbf{C}_t^{(l+1)} = \mathbf{T}_t \mathbf{C}_t^{(l)}, \quad \textit{for } l \geq 1. \tag{21}$$

Here, $\mathbf{B}_t$ is the base influence matrix:

$$[\mathbf{B}_t]_{uv} = [\mathbf{W}_t]_{uv} \cdot \kappa(t - t_u^{last}), \tag{22}$$

and $\mathbf{T}_t$ is the temporal propagation matrix:

$$[\mathbf{T}_t]_{uv} = [\mathbf{W}_t]_{uv} \cdot \kappa(t_v^{last} - t_u^{last}). \tag{23}$$

Thus, the recursion captures only the inter-type multi-order contribution, while the self-history contribution is represented separately by $\boldsymbol{\nu}(t)$.

Proof. We aim to show that $[\mathbf{C}_t^{(l)}]_{uv}$ represents the cumulative inter-type influence from event type u to event type v via a path of length l, and that Eq. (19) correctly aggregates these influences.

Component-wise Decomposition. Consider the v-th component of the intensity vector $\lambda_v(t)$. From Eq. (19), noting that the v-th element of $(\mathbf{C}_t^{(l)})^\top \mathbf{1}$ is the sum of the v-th column of $\mathbf{C}_t^{(l)}$:

$$\lambda_v(t) = \mu_v + \nu_v(t|\mathcal{H}_t^{\{v\}}) + \sum_{l=1}^{L} \alpha_v^{(l)} \sum_{u=1}^{K} [\mathbf{C}_t^{(l)}]_{uv}. \tag{24}$$

It suffices to prove that $[\mathbf{C}_t^{(l)}]_{uv}$ captures the l-th order inter-type influence from u to v at time t by induction.

Base Case ($l = 1$). From Eq. (20) and (22):

$$[\mathbf{C}_t^{(1)}]_{uv} = [\mathbf{B}_t]_{uv} = [\mathbf{W}_t]_{uv} \cdot \kappa(t - t_u^{last}). \tag{25}$$

This term explicitly models the direct 1-hop inter-type influence from the most recent event of type u on type v at the current time t. Thus, the base case holds.

Inductive Step ($l \geq 2$). Assume $[\mathbf{C}_t^{(k)}]_{jv}$ correctly represents the cumulative influence of a k-length path from j to v ending at time t. Consider the $(k+1)$-th order matrix defined in Eq. (21):

$$\mathbf{C}_t^{(k+1)} = \mathbf{T}_t \mathbf{C}_t^{(k)}. \tag{26}$$

The element at (u, v) is given by the matrix multiplication:

$$[\mathbf{C}_t^{(k+1)}]_{uv} = \sum_{j=1}^{K} [\mathbf{T}_t]_{uj} \cdot [\mathbf{C}_t^{(k)}]_{jv}. \tag{27}$$

Substituting the definition of $\mathbf{T}_t$ from Eq. (23):

$$[\mathbf{C}_t^{(k+1)}]_{uv} = \sum_{j=1}^{K} \underbrace{\left([\mathbf{W}_t]_{uj} \cdot \kappa(t_j^{last} - t_u^{last})\right)}_{\text{Influence } u \to j \text{ (step 1)}} \cdot \underbrace{[\mathbf{C}_t^{(k)}]_{jv}}_{\text{Influence } j \to v \text{ (step } 2, \cdots, k+1)}. \tag{28}$$

The term $[\mathbf{T}_t]_{uj}$ captures inter-type influence propagation between the most recent events u and j. The product combines the transition $u \to j$ with the subsequent k-th-order path $j \to \cdots \to v$. Summing over all intermediate nodes j yields the total influence of all paths of length $(k+1)$ from u to v.

Summary. By induction, $\mathbf{C}_t^{(l)}$ captures the l-th-order inter-type influence. Substituting this back into Eq. (24) shows that $\boldsymbol{\lambda}(t)$ is the weighted sum of base intensity, self-history, and historical inter-type influences aggregated across all orders $l = 1, \cdots, L$. □

D Gradient Boundedness

Proposition D.1 (Gradient Boundedness)**.** *Assume the decay kernel satisfies $\kappa(\cdot) \in [0, 1]$ and the learned structural matrix is bounded such that $\sup_t \|\mathbf{W}_t\|_F \leq \gamma$. Furthermore, let the parameterization mapping $f_\theta : \theta \mapsto \mathbf{W}_t$ be Lipschitz continuous with constant L_f. Then, the intensity function $\boldsymbol{\lambda}(t)$ is Lipschitz continuous with respect to θ, and the gradient norm is bounded by:*

$$\|\nabla_\theta \boldsymbol{\lambda}(t)\| \leq C \cdot L_f \sum_{l=1}^{L} l \cdot \gamma^{l-1}, \tag{29}$$

where C is a constant depending on the maximum intensity coefficients.

Proof. Let the intensity function be defined as $\boldsymbol{\lambda}(t) = \boldsymbol{\mu} + \boldsymbol{\nu}(t) + \sum_{l=1}^{L} \boldsymbol{\alpha}^{(l)} \circ \left((\mathbf{C}_t^{(l)})^\top \mathbf{1}\right)$. Here θ denotes the parameters of the structural generator f_θ, so the baseline term $\boldsymbol{\mu}$, the self-history term $\boldsymbol{\nu}(t)$, and the coefficients $\boldsymbol{\alpha}^{(l)}$ are independent of θ. Hence,

$$\nabla_\theta \boldsymbol{\lambda}(t) = \sum_{l=1}^{L} \nabla_\theta \Big[\boldsymbol{\alpha}^{(l)} \circ \left((\mathbf{C}_t^{(l)})^\top \mathbf{1}\right)\Big]. \tag{30}$$

The l-th order influence matrix expands to:

$$\mathbf{C}_t^{(l)} = (\mathbf{T}_t)^{l-1} \mathbf{B}_t = (\mathbf{W}_t \circ \mathbf{K}_T)^{l-1} (\mathbf{W}_t \circ \mathbf{K}_B). \tag{31}$$

Gradient Decomposition. By the chain rule, the gradient of $\boldsymbol{\lambda}(t)$ with respect to parameters θ is bounded by:

$$\|\nabla_\theta \boldsymbol{\lambda}(t)\| \leq \sum_{l=1}^{L} \sqrt{K}\, \|\boldsymbol{\alpha}^{(l)}\|_\infty \cdot \|\nabla_{\mathbf{W}} \mathbf{C}_t^{(l)}\| \cdot \|\nabla_\theta \mathbf{W}_t\|, \tag{32}$$

where the factor $\sqrt{K}$ comes from $\|(\mathbf{C}_t^{(l)})^\top \mathbf{1}\| \leq \|\mathbf{C}_t^{(l)}\|_F \|\mathbf{1}\| = \sqrt{K}\, \|\mathbf{C}_t^{(l)}\|_F$. Since f_θ is L_f-Lipschitz, its Jacobian norm is bounded by L_f at every differentiability point, so $\|\nabla_\theta \mathbf{W}_t\| \leq L_f$ almost everywhere. Let $C = \sqrt{K} \max_{1 \leq l \leq L} \|\boldsymbol{\alpha}^{(l)}\|_\infty$.

Bound on Recursive Structure. $[\mathbf{B}_t]_{uv} = [\mathbf{W}_t]_{uv} \cdot \kappa(t - t_u^{last})$ and $[\mathbf{T}_t]_{uv} = [\mathbf{W}_t]_{uv} \cdot \kappa(t_v^{last} - t_u^{last})$. Let $\gamma = \sup_t \|\mathbf{W}_t\|_F$. Since $\kappa(\cdot) \in [0, 1]$, every entry of $\mathbf{K}_B$ and $\mathbf{K}_T$ lies in $[0, 1]$, so the Hadamard products cannot increase the Frobenius norm. Thus, $\|\mathbf{T}_t\|_F \leq \gamma$ and $\|\mathbf{B}_t\|_F \leq \gamma$. Applying the product rule to $\mathbf{C}_t^{(l)}$:

$$\nabla_{\mathbf{W}_t} \mathbf{C}_t^{(l)} = \sum_{i=0}^{l-2} \mathbf{T}_t^i (\nabla_{\mathbf{W}_t} \mathbf{T}_t) \mathbf{T}_t^{l-2-i} \mathbf{B}_t + \mathbf{T}_t^{l-1} (\nabla_{\mathbf{W}_t} \mathbf{B}_t). \tag{33}$$

Moreover, since $\mathbf{T}_t = \mathbf{W}_t \circ \mathbf{K}_T$ and $\mathbf{B}_t = \mathbf{W}_t \circ \mathbf{K}_B$, their derivatives with respect to $\mathbf{W}_t$ are Hadamard multiplications by matrices with entries in $[0, 1]$, and therefore satisfy $\|\nabla_{\mathbf{W}_t} \mathbf{T}_t\| \leq 1$ and $\|\nabla_{\mathbf{W}_t} \mathbf{B}_t\| \leq 1$. Taking norms yields:

$$\|\nabla_{\mathbf{W_t}} \mathbf{C}_t^{(l)}\| \leq \sum_{i=0}^{l-2} \|\mathbf{T}_t\|_F^i \cdot 1 \cdot \|\mathbf{T}_t\|_F^{l-2-i} \|\mathbf{B}_t\|_F + \|\mathbf{T}_t\|_F^{l-1} \cdot 1 \tag{34}$$

$$\leq (l-1)\gamma^{l-2} \cdot \gamma + \gamma^{l-1} \tag{35}$$

$$= l \cdot \gamma^{l-1}. \tag{36}$$

Combining the Bounds. Substituting the bounds back into the gradient sum:

$$\|\nabla_\theta \boldsymbol{\lambda}(t)\| \leq C \cdot L_f \sum_{l=1}^{L} l \cdot \gamma^{l-1}. \tag{37}$$

This confirms that the gradient norm is bounded by a polynomial series of the weight norm γ. Since the bound is uniform in θ, $\boldsymbol{\lambda}(t)$ is Lipschitz continuous with respect to θ. □

E Causal Decomposition in TPP

We establish the theoretical validity of decomposing the intensity into multi-order components in a restricted static linear setting. To stay consistent with the main model, we let the total kernel coefficient matrix absorb both self-excitation and DAG-structured inter-type effects, and then show that this total matrix admits a unique decomposition into a diagonal self-excitation part and a finite inter-type part. We then add a sufficient condition under which the static structural matrix is identifiable from the conditional intensity itself in the same restricted linear setting.

Proposition E.1 (Causal Decomposition in TPP)**.** *Let $\{\mathbf{N}(t)\}_{t\geq 0}$ be a K-dimensional multivariate temporal point process. Assume the conditional intensity function $\boldsymbol{\lambda}(t)$ follows a linear Hawkes model with a separable kernel structure:*

$$\boldsymbol{\lambda}(t) = \boldsymbol{\mu} + \mathbf{U} \int_0^t \kappa(t-\tau) d\mathbf{N}(\tau), \tag{38}$$

where $\boldsymbol{\mu} \in \mathbb{R}_+^K$ is a constant baseline intensity, $\kappa(t)$ is a time decay kernel function, and

$$\mathbf{U} = \mathbf{D} + \sum_{l=1}^{K-1} \mathbf{W}^l \tag{39}$$

with $\mathbf{D} \in \mathbb{R}^{K\times K}$ diagonal and $\mathbf{W} \in \mathbb{R}^{K\times K}$ an acyclic structural weight matrix satisfying $\mathrm{diag}(\mathbf{W}) = \mathbf{0}$. Then $\mathbf{U}$ admits a unique decomposition of the form $\mathbf{D} + \sum_{l=1}^{K-1} \mathbf{W}^l$, and the intensity admits the induced finite multi-order decomposition:

$$\begin{aligned} \boldsymbol{\lambda}(t) &= \boldsymbol{\mu} + \mathbf{D} \int_0^t \kappa(t-\tau) d\mathbf{N}(\tau) + \sum_{l=1}^{K-1} \mathbf{W}^l \int_0^t \kappa(t-\tau) d\mathbf{N}(\tau) \\ &= \boldsymbol{\mu} + \mathbf{D} \int_0^t \kappa(t-\tau) d\mathbf{N}(\tau) + \sum_{l=1}^{K-1} \boldsymbol{\lambda}^{(l)}(t), \end{aligned} \tag{40}$$

where $\boldsymbol{\lambda}^{(l)}(t) := \mathbf{W}^l \int_0^t \kappa(t-\tau) d\mathbf{N}(\tau)$ is the l-th order intensity.

Proof. The proof is structured into three parts. First, we establish the finite expansion property. Second, we derive the decomposition form. Finally, we prove the uniqueness of this decomposition.

Finite Expansion. The causal graph $\mathcal{G}$ is a DAG with the structural weight matrix $\mathbf{W} \in \mathbb{R}^{K\times K}$. According to Proposition 3.2, a strictly upper triangular matrix of dimension $K \times K$ is nilpotent with an index of at most K. Consequently, the powers of $\mathbf{W}$ vanish for all orders $l \geq K$:

$$\mathbf{W}^l = \mathbf{0}, \quad \forall l \geq K. \tag{41}$$

Thus, any infinite series involving powers of $\mathbf{W}$ naturally truncates to a finite sum up to order $K-1$.

Decomposition. We define the inter-type part of the total coefficient matrix by the Neumann series of the structural weight matrix $\mathbf{W}$:

$$\mathbf{U}_{\mathrm{pa}} := \sum_{l=1}^{\infty} \mathbf{W}^l. \tag{42}$$

Applying the finite expansion property from Eq.(41), we obtain

$$\mathbf{U}_{\mathrm{pa}} = \sum_{l=1}^{K-1} \mathbf{W}^l. \tag{43}$$

Hence the total matrix can be written as

$$\mathbf{U} = \mathbf{D} + \mathbf{U}_{\mathrm{pa}} = \mathbf{D} + \sum_{l=1}^{K-1} \mathbf{W}^l. \tag{44}$$

Substituting this identity into the intensity definition yields

$$\begin{aligned} \boldsymbol{\lambda}(t) &= \boldsymbol{\mu} + \mathbf{U} \int_0^t \kappa(t-\tau) d\mathbf{N}(\tau) \\ &= \boldsymbol{\mu} + \left(\mathbf{D} + \sum_{l=1}^{K-1} \mathbf{W}^l \right) \int_0^t \kappa(t-\tau) d\mathbf{N}(\tau) \\ &= \boldsymbol{\mu} + \mathbf{D} \int_0^t \kappa(t-\tau) d\mathbf{N}(\tau) + \sum_{l=1}^{K-1} \left(\mathbf{W}^l \int_0^t \kappa(t-\tau) d\mathbf{N}(\tau) \right). \end{aligned} \tag{45}$$

By defining the l-th order intensity component as $\boldsymbol{\lambda}^{(l)}(t) := \mathbf{W}^l \int_0^t \kappa(t-\tau) d\mathbf{N}(\tau)$, we arrive at the decomposed form stated in Eq.(40).

Uniqueness. We prove that given a total matrix $\mathbf{U}$, there exists a unique pair $(\mathbf{D}, \mathbf{W})$ satisfying the decomposition above.

Because $\mathbf{W}$ is acyclic, there exists a topological ordering under which $\mathbf{W}$ is strictly upper triangular. Therefore every power $\mathbf{W}^l$ is also strictly upper triangular, and hence

$$\mathrm{diag}(\mathbf{U}_{\mathrm{pa}}) = \mathbf{0}. \tag{46}$$

Since $\mathbf{D}$ is diagonal and $\mathbf{U} = \mathbf{D} + \mathbf{U}_{\mathrm{pa}}$, we must have

$$\mathbf{D} = \mathrm{diag}(\mathbf{U}), \qquad \mathbf{U}_{\mathrm{pa}} = \mathbf{U} - \mathrm{diag}(\mathbf{U}). \tag{47}$$

Thus the diagonal and inter-type parts are uniquely determined by $\mathbf{U}$.

It remains to recover $\mathbf{W}$ uniquely from $\mathbf{U}_{\mathrm{pa}}$. From the relation $\mathbf{U}_{\mathrm{pa}} = \sum_{l=1}^{\infty} \mathbf{W}^l$, we recognize the geometric series of matrices, which converges to:

$$\mathbf{U}_{\mathrm{pa}} = (\mathbf{I} - \mathbf{W})^{-1} - \mathbf{I}, \tag{48}$$

provided the spectral radius $\rho(\mathbf{W}) < 1$. Since $\mathbf{W}$ is nilpotent, all its eigenvalues are 0, satisfying $\rho(\mathbf{W}) = 0 < 1$. Thus, $(\mathbf{I} - \mathbf{W})$ is invertible. We can invert this relationship to solve for $\mathbf{W}$:

$$\mathbf{I} + \mathbf{U}_{\mathrm{pa}} = (\mathbf{I} - \mathbf{W})^{-1} \tag{49}$$

$$(\mathbf{I} + \mathbf{U}_{\mathrm{pa}})^{-1} = \mathbf{I} - \mathbf{W} \tag{50}$$

$$\mathbf{W} = \mathbf{I} - (\mathbf{I} + \mathbf{U}_{\mathrm{pa}})^{-1} \tag{51}$$

Since the matrix inverse is unique, $\mathbf{W}$ is uniquely determined by $\mathbf{U}_{\mathrm{pa}}$, hence by $\mathbf{U}$. Consequently, the powers $\mathbf{W}^l$ are unique for all l, and the corresponding intensity components $\boldsymbol{\lambda}^{(l)}(t)$ are uniquely defined. This concludes the proof. □

This proposition motivates our model design by establishing an exact decomposition in the restricted static linear setting. In this setting, it also shows that one may equivalently work with the single total

matrix $\mathbf{U}$, because under the DAG assumption $\mathbf{U}$ uniquely separates into a diagonal self-excitation part and a finite multi-order inter-type part.

A sufficient condition for causal identifiability. Proposition E.1 shows that once the total matrix $\mathbf{U}$ is known, its decomposition into self-excitation and DAG-induced inter-type components is unique. We next record a sufficient condition under which $(\boldsymbol{\mu}, \mathbf{D}, \mathbf{W})$ is identifiable from the conditional intensity in the same static linear-Hawkes-style setting.

Proposition E.2 (Structural Identifiability in the Restricted Static Setting)**.** *Consider the restricted model*

$$\boldsymbol{\lambda}(t) = \boldsymbol{\mu} + \mathbf{U}\mathbf{z}(t), \qquad \mathbf{z}(t) = \int_0^t \kappa(t-\tau)\, d\mathbf{N}(\tau), \qquad \mathbf{U} = \mathbf{D} + \sum_{l=1}^{K-1} \mathbf{W}^l, \tag{52}$$

where $\mathbf{D}$ *is diagonal,* $\mathbf{W}$ *is acyclic with* $\operatorname{diag}(\mathbf{W}) = \mathbf{0}$*,* κ *is known, and the augmented regressor*

$$\tilde{\mathbf{z}}(t) := \begin{bmatrix} 1 \\ \mathbf{z}(t) \end{bmatrix} \tag{53}$$

satisfies

$$\mathbf{Q}_T := \int_0^T \tilde{\mathbf{z}}(t)\tilde{\mathbf{z}}(t)^\top\, dt \succ 0. \tag{54}$$

Then, within this restricted static model, exact knowledge of the conditional intensity on $[0, T]$ *uniquely determines* $(\boldsymbol{\mu}, \mathbf{D}, \mathbf{W})$*. Consequently, the order-specific structures* $\mathbf{W}^l$ *are also uniquely determined.*

Proof. Suppose two parameter sets $(\boldsymbol{\mu}, \mathbf{D}, \mathbf{W})$ and $(\boldsymbol{\mu}', \mathbf{D}', \mathbf{W}')$ induce the same conditional intensity on $[0, T]$. Let

$$\mathbf{U} = \mathbf{D} + \sum_{l=1}^{K-1} \mathbf{W}^l, \qquad \mathbf{U}' = \mathbf{D}' + \sum_{l=1}^{K-1} (\mathbf{W}')^l. \tag{55}$$

Then, for all $t \in [0, T]$,

$$\boldsymbol{\mu} + \mathbf{U}\mathbf{z}(t) = \boldsymbol{\mu}' + \mathbf{U}'\mathbf{z}(t), \tag{56}$$

which implies

$$(\boldsymbol{\mu} - \boldsymbol{\mu}') + (\mathbf{U} - \mathbf{U}')\mathbf{z}(t) = \mathbf{0}. \tag{57}$$

Equivalently,

$$[\boldsymbol{\mu} - \boldsymbol{\mu}' \quad \mathbf{U} - \mathbf{U}']\, \tilde{\mathbf{z}}(t) = \mathbf{0}, \qquad \forall t \in [0, T]. \tag{58}$$

Multiplying on the right by $\tilde{\mathbf{z}}(t)^\top$ and integrating over $[0, T]$, we obtain

$$[\boldsymbol{\mu} - \boldsymbol{\mu}' \quad \mathbf{U} - \mathbf{U}'] \int_0^T \tilde{\mathbf{z}}(t)\tilde{\mathbf{z}}(t)^\top\, dt = [\boldsymbol{\mu} - \boldsymbol{\mu}' \quad \mathbf{U} - \mathbf{U}']\, \mathbf{Q}_T = \mathbf{0}. \tag{59}$$

Since $\mathbf{Q}_T \succ 0$, it is invertible. Hence,

$$\boldsymbol{\mu} = \boldsymbol{\mu}', \qquad \mathbf{U} = \mathbf{U}'. \tag{60}$$

Now Proposition E.1 implies that the decomposition of $\mathbf{U}$ into a diagonal matrix $\mathbf{D}$ and a finite DAG-induced series $\sum_{l=1}^{K-1} \mathbf{W}^l$ is unique. Therefore $\mathbf{D} = \mathbf{D}'$ and $\mathbf{W} = \mathbf{W}'$. It follows that $\mathbf{D}$ and the structural matrix $\mathbf{W}$ are identifiable, and so are the order-specific structures $\mathbf{W}^l$. □

This result is intentionally stated only for the restricted static linear setting in Appendix E. It provides a sufficient identifiability condition when the kernel is known and the augmented regressor is nondegenerate on $[0, T]$. Combined with Proposition E.1, it shows that once the total matrix $\mathbf{U}$ is identified from the conditional intensity, its self-excitation and multi-order structural components are also uniquely identified. At the population level, the observational law determines the predictable conditional intensity relative to the fixed observed-history filtration, provided an absolutely continuous compensator exists. Under correct specification, this identifies the modeled intensity up to almost-everywhere equivalence, but does not establish uniqueness of the neural parameters or structural graph. Identifiability of the dynamic structural matrix remains an open question.

F Smoothness, Penalty Consistency, and Relation to Training Convergence

This appendix clarifies the scope of our theoretical guarantee. In MOCHA, training updates the context encoder, the dynamic graph learner, and the multi-order intensity model together through end-to-end nonconvex gradient optimization. A full theorem for the convergence and feasibility of this practical training process would need extra assumptions about the optimizer, such as the step-size rule, bounded iterates, gradient noise, and how accurately each penalized subproblem is solved. These assumptions are not the main focus of this paper, and they are usually hard to verify in neural TPP training. We therefore focus on the structural part that is directly tied to our method. Specifically, we show that the continuous DAG penalty in Eq.(8) is smooth, nonnegative, and penalty-consistent, so increasing the acyclicity coefficient drives exact minimizers of the penalized objective toward the feasible DAG set. This result is meaningful for MOCHA because it gives a clear justification for how the method enforces causal validity, regardless of the specific neural form used for the temporal encoder or the graph learner. The proposition can also serve as a starting point for a stronger training analysis. Under additional standard assumptions from nonconvex penalty methods, such as Lipschitz continuity of the full gradient, bounded iterates, sufficient descent or vanishing first-order residuals, and an increasing penalty schedule, one could further show that accumulation points of the training path are stationary at each penalty level and become asymptotically feasible as the penalty grows. We therefore present the proposition as a focused structural guarantee that supports the method and also provides a basis for a future optimizer-level convergence result.

Proposition F.1 (Smoothness and Penalty Consistency of the Acyclicity Constraint)**.** *Let*

$$h(\mathbf{W}) := \operatorname{Tr}\big(e^{\mathbf{W}\circ\mathbf{W}}\big) - K, \qquad \mathbf{W} \in \mathbb{R}^{K\times K},$$

and define, for a sequence of structural matrices $\boldsymbol{\mathcal{W}} := (\mathbf{W}_{t_1}, \ldots, \mathbf{W}_{t_N})$*,*

$$H(\boldsymbol{\mathcal{W}}) := \sum_{i=1}^{N} h(\mathbf{W}_{t_i}).$$

Then the following statements hold.

(i) The map h *is real analytic on* $\mathbb{R}^{K\times K}$*; in particular,* $h \in C^\infty(\mathbb{R}^{K\times K})$*. Its gradient is*

$$\nabla h(\mathbf{W}) = 2\mathbf{W} \circ \big(e^{\mathbf{W}\circ\mathbf{W}}\big)^\top. \tag{61}$$

Consequently, ∇h *is Lipschitz continuous on every bounded subset of* $\mathbb{R}^{K\times K}$*.*

(ii) For every $\mathbf{W} \in \mathbb{R}^{K\times K}$*, one has* $h(\mathbf{W}) \geq 0$*. Moreover,* $h(\mathbf{W}) = 0$ *if and only if* $\mathbf{W}$ *represents a DAG, equivalently, if and only if* $\operatorname{Tr}(e^{\mathbf{W}\circ\mathbf{W}}) = K$*.*

(iii) Let $\ell : (\mathbb{R}^{K\times K})^N \to \mathbb{R}$ *denote the part of the training objective independent of* γ_{acyclic} *(e.g., the negative log-likelihood plus the sparsity term), and for* $\gamma > 0$ *define*

$$F_\gamma(\boldsymbol{\mathcal{W}}) := \ell(\boldsymbol{\mathcal{W}}) + \gamma H(\boldsymbol{\mathcal{W}}). \tag{62}$$

Assume that there exists a feasible point $\bar{\boldsymbol{\mathcal{W}}}$ *such that* $H(\bar{\boldsymbol{\mathcal{W}}}) = 0$*, and that for each* $\gamma > 0$*, the objective* F_γ *admits a global minimizer* $\boldsymbol{\mathcal{W}}_\gamma$*. If* ℓ *is bounded below by a constant* m *on the set* $\{\boldsymbol{\mathcal{W}}_\gamma : \gamma > 0\} \cup \{\bar{\boldsymbol{\mathcal{W}}}\}$*, then*

$$0 \leq H(\boldsymbol{\mathcal{W}}_\gamma) \leq \frac{\ell(\bar{\boldsymbol{\mathcal{W}}}) - m}{\gamma}. \tag{63}$$

In particular, $H(\boldsymbol{\mathcal{W}}_\gamma) \to 0$ *as* $\gamma \to \infty$*. Therefore, every accumulation point* $\boldsymbol{\mathcal{W}}^*$ *of the family* $\{\boldsymbol{\mathcal{W}}_\gamma\}_{\gamma>0}$ *satisfies* $H(\boldsymbol{\mathcal{W}}^*) = 0$*, and hence each matrix* $\mathbf{W}^*_{t_i}$ *is acyclic.*

Proof. For part (i), set $\mathbf{S}(\mathbf{W}) := \mathbf{W} \circ \mathbf{W}$. The map $\mathbf{W} \mapsto \mathbf{S}(\mathbf{W})$ is polynomial, the matrix exponential is real analytic, and the trace is linear. Hence $h(\mathbf{W}) = \operatorname{Tr}(e^{\mathbf{S}(\mathbf{W})}) - K$ is real analytic on $\mathbb{R}^{K\times K}$.

To compute the gradient, let $\mathbf{H} \in \mathbb{R}^{K\times K}$ be an arbitrary perturbation. Since $\mathrm{d}(\mathbf{W}\circ\mathbf{W})[\mathbf{H}] = 2\mathbf{W}\circ\mathbf{H}$ and the Fréchet derivative of $\mathbf{S} \mapsto \operatorname{Tr}(e^{\mathbf{S}})$ in direction $\mathbf{G}$ is $\operatorname{Tr}(e^{\mathbf{S}}\mathbf{G})$, we obtain

$$\mathrm{d}h(\mathbf{W})[\mathbf{H}] = \operatorname{Tr}\big(e^{\mathbf{W}\circ\mathbf{W}}(2\mathbf{W}\circ\mathbf{H})\big) \tag{64}$$

$$= \Big\langle 2\mathbf{W} \circ \big(e^{\mathbf{W}\circ\mathbf{W}}\big)^\top, \mathbf{H} \Big\rangle, \tag{65}$$

where $\langle \mathbf{A}, \mathbf{B} \rangle := \mathrm{Tr}(\mathbf{A}^\top \mathbf{B})$ denotes the Frobenius inner product. Therefore,

$$\nabla h(\mathbf{W}) = 2\mathbf{W} \circ \left(e^{\mathbf{W} \circ \mathbf{W}}\right)^\top.$$

Because h is analytic, its Hessian is continuous. On every bounded set, the Hessian is therefore bounded on the closure of that set, and the mean-value theorem implies that ∇h is Lipschitz on every bounded set.

For part (ii), write $\mathbf{M} := \mathbf{W} \circ \mathbf{W}$. Then $\mathbf{M}$ is entrywise nonnegative, and thus every diagonal entry of $\mathbf{M}^k$ is nonnegative for all $k \geq 1$. Using the power-series expansion of the matrix exponential,

$$h(\mathbf{W}) = \sum_{k=1}^{\infty} \frac{1}{k!} \mathrm{Tr}(\mathbf{M}^k) \geq 0. \tag{66}$$

By the characterization of acyclicity introduced in Section 3.2, $\mathrm{Tr}(e^{\mathbf{W} \circ \mathbf{W}}) = K$ holds if and only if $\mathbf{W}$ represents a DAG. This is equivalent to $h(\mathbf{W}) = 0$.

For part (iii), fix $\gamma > 0$. Since $\boldsymbol{\mathcal{W}}_\gamma$ is a global minimizer of F_γ and $H(\bar{\boldsymbol{\mathcal{W}}}) = 0$, we have

$$F_\gamma(\boldsymbol{\mathcal{W}}_\gamma) \leq F_\gamma(\bar{\boldsymbol{\mathcal{W}}}) = \ell(\bar{\boldsymbol{\mathcal{W}}}). \tag{67}$$

By the lower bound $\ell(\boldsymbol{\mathcal{W}}_\gamma) \geq m$ and the definition of F_γ,

$$m + \gamma H(\boldsymbol{\mathcal{W}}_\gamma) \leq \ell(\bar{\boldsymbol{\mathcal{W}}}), \tag{68}$$

which immediately gives

$$0 \leq H(\boldsymbol{\mathcal{W}}_\gamma) \leq \frac{\ell(\bar{\boldsymbol{\mathcal{W}}}) - m}{\gamma}. \tag{69}$$

Hence $H(\boldsymbol{\mathcal{W}}_\gamma) \to 0$ as $\gamma \to \infty$.

Now let $\boldsymbol{\mathcal{W}}^*$ be any accumulation point of $\{\boldsymbol{\mathcal{W}}_\gamma\}_{\gamma > 0}$. Then there exists a sequence $\gamma_n \to \infty$ such that $\boldsymbol{\mathcal{W}}_{\gamma_n} \to \boldsymbol{\mathcal{W}}^*$. Since H is continuous, we obtain

$$H(\boldsymbol{\mathcal{W}}^*) = \lim_{n \to \infty} H(\boldsymbol{\mathcal{W}}_{\gamma_n}) = 0. \tag{70}$$

Each term $h(\mathbf{W}_{t_i}^*)$ is nonnegative by part (ii), so the equality $\sum_{i=1}^{N} h(\mathbf{W}_{t_i}^*) = 0$ implies $h(\mathbf{W}_{t_i}^*) = 0$ for all i. Applying part (ii) again, every $\mathbf{W}_{t_i}^*$ represents a DAG. This proves the claim. □

G Algorithmic Overview of MOCHA

To provide a clear and reproducible understanding of our proposed method, this section presents the complete algorithmic workflow of MOCHA in the form of pseudocode. Algorithm 1 summarizes how to model intensities and infer causal graphs from event sequences. Each step corresponds to components introduced in the main text.

H Experiments on Synthetic Data

Real-world benchmarks verify predictive performance and domain plausibility, but they do not provide ground-truth causal graphs, making it impossible to fully assess graph recovery quality. To complement those results, we construct a controlled synthetic benchmark in which the underlying multi-order dynamic structure is fully observed. This section investigates three research questions:

- **RQ-S1:** Can MOCHA recover the overall causal summary graph?
- **RQ-S2:** Can MOCHA recover order-specific multi-order causal graphs?
- **RQ-S3:** How sensitive is graph extraction to the threshold θ?

Synthetic benchmark settings. The synthetic sequences are generated in a Hawkes-style manner from an intensity function composed of a base intensity, time decay, and multi-order causal propagation terms. We first sample a DAG skeleton over $K = 6$ event types with maximum causal order $L = 3$, and then let the corresponding structural weights evolve over time, which yields dynamic ground-truth graphs for each sequence. Event times are sampled with the standard thinning algorithm.

Algorithm 1 MOCHA: Multi-Order Causal Hierarchical Architecture

Require: Event sequence $\mathcal{S} = \{(t_i, k_i)\}_{i=1}^{N}$, maximum order L, threshold θ, scale β
Ensure: Intensities $\boldsymbol{\lambda}(t)$ and causal graphs $\mathcal{G}_t$
Initialize total loss $\mathcal{L} \leftarrow 0$
for each event (t_i, k_i) **do**
// 1. Causal graph learning phase
    Compute embeddings $\mathbf{H}_{t_i}$ using context encoding (Eq.(3))
    Compute structural weights $\mathbf{W}_{t_i}$ using the dense attention structural learner (Eq.(5))
    Compute binary adjacency matrix $\mathbf{A}_{t_i}$ and construct causal graph $\mathcal{G}_{t_i}$ (Eq.(6))
// 2. Intensity calculation phase
    Compute event intensity $\boldsymbol{\lambda}(t_i)$ using Eq.(14)
    Apply activation $\boldsymbol{\lambda}(t_i) \leftarrow \mathrm{Softplus}(\boldsymbol{\lambda}(t_i))$
// 3. Loss update phase
    Accumulate the contribution of (t_i, k_i) to the NLL objective in Eq.(15)
    Update loss $\mathcal{L} \leftarrow \mathcal{L} + \ell_{\mathrm{NLL},i} + \mathcal{L}_{\mathrm{acyclic}}(\mathbf{W}_{t_i}) + \mathcal{L}_{\mathrm{sparse}}(\mathbf{W}_{t_i})$
end for
Update parameters via gradient descent

The resulting benchmark contains 640/160/160 train/dev/test sequences with an average length of 32. Unless otherwise specified, all results are averaged over five random seeds and reported as mean $\pm$ standard deviation.

For evaluation, we binarize learned weighted graphs using the same extraction rule as in Section 3.2. We report *Edge-Accuracy*, which measures entry-wise agreement between recovered and ground-truth binary adjacency matrices, and *Edge-F1*, which emphasizes the recovery of positive edges in sparse graphs. For methods that output dynamic graphs, we form the summary graph by taking the union of recovered edges over time; for order-wise evaluation, we aggregate recovered edges separately for each causal order.

RQ-S1: Can MOCHA Recover the Overall Causal Summary Graph?

We compare four representative methods: *Hawkes*, which provides a non-structural excitation baseline; *CAUSE* [12], a Granger-causality model; *Static SCM*, which learns a time-invariant DAG; and *MOCHA*, which learns dynamic multi-order DAGs. Table 3 shows that MOCHA achieves the best performance on both metrics. The gain is especially large on Edge-F1, indicating that MOCHA more accurately recovers the sparse set of true causal edges instead of overproducing spurious connections. This controlled result complements the real-data experiments by showing that the model not only fits sequences well, but also reconstructs the underlying causal summary graph when the ground truth is known.

Table 3: Causal summary-graph recovery on the synthetic benchmark (mean $\pm$ standard deviation over five runs; $\uparrow$ is better).

Model	Edge-Accuracy	Edge-F1
Hawkes	0.8208±0.0000	0.0000±0.0000
CAUSE	0.8426±0.0104	0.1672±0.1615
Static SCM	0.9032±0.0318	0.6896±0.1452
MOCHA	**0.9802±0.0207**	**0.9382±0.0665**

RQ-S2: Can MOCHA Recover Order-Specific Multi-Order Causal Graphs?

To determine whether the model learns genuine multi-order structure rather than only an aggregated summary graph, we separately evaluate the recovered first-, second-, and third-order graphs. We compare three structural variants: a *First-order Static SCM* that only models direct effects, a *Multi-order Static SCM* that models orders up to $L = 3$ but assumes a time-invariant graph, and *MOCHA*. As shown in Table 4, the first-order static model cannot recover higher-order structure by construction. Allowing static multi-order edges substantially improves second- and third-order recovery, but MOCHA remains clearly superior across all orders, especially in second- and third-order F1. The third-order graph is the sparsest component of the synthetic generator, which explains the relatively

larger variance and the lower absolute F1 even for the strongest model. Overall, these results show that MOCHA can learn the multi-order causal graph itself.

Table 4: Order-wise recovery of learned multi-order causal graphs on the synthetic benchmark (mean $\pm$ standard deviation over five runs; $\uparrow$ is better).

Model	1st-order Accuracy	2nd-order Accuracy	3rd-order Accuracy	1st-order F1	2nd-order F1	3rd-order F1
First-order Static SCM	0.9238±0.0267	0.0000±0.0000	0.0000±0.0000	0.7859±0.1029	0.0000±0.0000	0.0000±0.0000
Multi-order Static SCM	0.9238±0.0267	0.9418±0.0152	0.9825±0.0045	0.7859±0.1029	0.6323±0.1893	0.3071±0.4205
MOCHA	**0.9901±0.0136**	**0.9871±0.0189**	**0.9957±0.0092**	**0.9703±0.0407**	**0.9100±0.1359**	**0.7962±0.4452**

RQ-S3: How Sensitive Is Graph Extraction to the Threshold θ?

We examine the threshold θ used in Eq.(6) to extract binary graphs from continuous structural weights. Table 5 shows that the best structural recovery is obtained at $\theta = 0.5$. This behavior reflects the expected trade-off: overly small thresholds retain weak edges and increase false positives, whereas overly large thresholds remove useful edges and increase false negatives. In practice, we treat θ as a tunable hyperparameter and select it by grid search on the validation set: we first retain thresholds that achieve near-best validation NLL, and then choose the sparsest graph among them. This rule yields a principled balance between predictive fit and structural sparsity.

Table 5: Sensitivity of MOCHA to the graph extraction threshold θ on the synthetic benchmark (mean $\pm$ standard deviation over five runs; $\uparrow$ is better).

θ	Edge-Accuracy	Edge-F1
0.3	0.9315±0.0427	0.7399±0.2018
0.4	0.9410±0.0434	0.7795±0.2029
0.5	**0.9802±0.0207**	**0.9382±0.0665**
0.6	0.9526±0.0394	0.8318±0.1577
0.7	0.9310±0.0479	0.7328±0.2187
1	0.8208±0.0000	0.0000±0.0000

I Quantitative Results on Benchmark Datasets

To comprehensively assess the temporal and semantic modeling capability of MOCHA, we conduct quantitative evaluations using two standard metrics: Root Mean Squared Error (RMSE) for timestamp prediction and Accuracy (Acc) for event type prediction. IFLTPP does not provide methods for calculating RMSE and Acc.

Time prediction is a key aspect of TPP modeling, as it directly reflects the model's understanding of temporal dependencies and event dynamics. A lower RMSE indicates better alignment between predicted and true timestamps. We evaluate RMSE by measuring the squared difference between the predicted and actual timestamps of future events across all sequences, and take the square root of the average. The results are shown in Table 6.

Table 6: RMSE comparison across benchmarks ($\downarrow$ is better).

	AKI	CAD	Sepsis	Stroke	Retweet	Taobao	StackOverflow
RMTPP	20.72	2.71	1.66	1.10	20.44	0.26	1.33
RPPN	25.15	2.10	2.17	2.34	24.98	0.52	3.24
NHP	17.55	2.11	1.51	1.14	22.65	0.82	1.40
SAHP	15.88	2.09	2.04	1.47	21.80	0.62	1.28
THP	14.87	2.20	1.53	1.10	23.00	0.29	1.39
AttNHP	14.52	2.16	1.49	1.03	20.24	0.32	1.38
ITHP	14.32	2.13	2.74	1.46	21.18	0.43	4.38
NJDTPP	12.61	2.48	1.37	1.24	16.10	0.25	1.13
S2P2	8.92	1.73	1.02	0.77	15.33	0.13	1.14
MOCHA	**8.03**	**1.64**	**0.93**	**0.60**	**14.49**	**0.12**	**0.52**

As shown in Table 6, MOCHA achieves the lowest RMSE across all seven datasets. On clinical datasets like AKI and Sepsis, MOCHA reduces RMSE by up to 36.3% and 32.1% respectively

compared to strong baselines. These gains reflect the model's advantage in modeling indirect causal chains and adapting to evolving event structures. Similar improvements are observed on non-medical datasets such as Retweet and Taobao, confirming the generalizability of the approach. By integrating multi-order causal reasoning and time-varying DAG structures into the intensity modeling process, MOCHA significantly enhances timestamp prediction accuracy across domains.

Accurately predicting the type of the next event is crucial for understanding and modeling the semantic structure of event sequences. Event type prediction reflects the model's ability to capture complex interdependencies between different types of events. In clinical decision support, anticipating the kind of future complication or intervention is as important as knowing when it will occur.

To evaluate this, we report event type prediction Accuracy (Acc), calculated as the proportion of correctly predicted event types among all test events. For each event (t_i, k_i), the model produces a probability distribution over event types, and the predicted type is selected as the one with the highest probability. Accuracy is then averaged across the entire test set. The results are shown in Table 7.

Table 7: Accuracy comparison across benchmarks (↑ is better).

	AKI	CAD	Sepsis	Stroke	Retweet	Taobao	StackOverflow	Average Rank
RMTPP	0.2001	0.1872	0.5114	0.2743	0.5459	0.4360	0.4250	9.00
RPPN	0.2136	0.5652	0.4489	0.4199	0.5793	0.5260	0.4097	7.00
NHP	0.1919	0.4877	0.7001	0.4368	0.5988	0.5940	0.4450	3.43
SAHP	0.1786	0.5523	0.6403	0.2851	0.5921	0.5560	0.4204	6.57
THP	0.2677	0.4824	0.7083	0.4197	0.5847	0.5880	0.4314	4.57
AttNHP	0.2459	0.5240	0.5198	0.3969	0.5277	0.4920	0.4342	6.86
ITHP	0.2376	0.5049	0.6357	0.4212	0.5911	0.5920	0.4429	4.71
NJDTPP	0.2408	0.3808	0.5180	0.2095	0.5982	0.4960	0.4438	6.57
S2P2	0.2647	0.5794	0.6838	0.4076	0.5947	0.5720	0.4601	3.29
MOCHA	0.2556	0.5889	0.6976	0.4280	0.5862	0.5740	0.4458	**3.00**

As shown in Table 7, MOCHA achieves the best overall accuracy with average rank of 3.00 across all seven datasets. On complex datasets such as CAD, it achieves the highest accuracy of 58.89%, outperforming all baselines. Despite not being optimal on every dataset, MOCHA remains highly competitive across both medical and non-medical domains, indicating robust generalization in event type modeling. These results highlight the model's ability to capture not only temporal structure but also fine-grained categorical dependencies.

By incorporating multi-order causal structures and dynamic dependency modeling into intensity computation, MOCHA improves the precision for next-event types, offering a balanced improvement in both timing and type prediction.

J Complexity Analysis and Training Runtime

Complexity Analysis. We summarize the computational complexity of MOCHA from three main components. First, the complexity of preprocessing is $\mathcal{O}(N)$, where N denotes the max sequence length. We maintain the latest occurrence time for each event time point, so the historical input is compressed into a latest-event representation. Second, the complexity of multi-order intensity computation is $\mathcal{O}(LK^2)$ by using the recursive aggregation strategy in Proposition 3.3, where L is the maximum causal order and K is the number of event types. Third, the complexity of acyclicity constraint computation is $\mathcal{O}(K^3)$, since evaluating the matrix exponential in the acyclicity regularizer operates on a $K \times K$ structural matrix. Hence, the overall complexity of MOCHA is $\mathcal{O}(\max(K^3, N))$.

The proposed matrix formulation in Proposition 3.3 significantly reduces computational complexity in three aspects. (1) *State-Space Construction:* Computing $\mathbf{B}_t$ and $\mathbf{T}_t$ involves pairwise operations on the most recent event times, with complexity $\mathcal{O}(K^2)$. (2) *Recursive Aggregation:* The core operation is the matrix-vector multiplication $(\mathbf{C}_t^{(l+1)})^\top \mathbf{1} = (\mathbf{T}_t^l \mathbf{B}_t)^\top \mathbf{1} = \mathbf{B}_t^\top (\mathbf{T}_t^l)^\top \mathbf{1} = \mathbf{B}_t^\top (\mathbf{T}_t^\top)^l \mathbf{1}$, which has a complexity of $\mathcal{O}(K^2)$. (3) *Multi-order Intensity Computation:* For a model considering up to L orders, the computational complexity per time step is $\mathcal{O}(LK^2)$.

In contrast, the complexity of most existing efficient neural TPPs, such as RMTPP [17], THP [10], AttNHP [19] and S2P2 [8], is $\mathcal{O}(NK)$, growing with the sequence length N. The complexity of

NJDTPP [21] is $\mathcal{O}(NSK)$, where S is the number of subintervals between events. Furthermore, since $L < K < N$ is typically small and matrix operations are highly parallelizable on GPUs, MOCHA scales efficiently to large graphs.

Training Runtime. Beyond the asymptotic analysis, we also report the actual training time of MOCHA and representative TPP baselines in Table 8. All runtime measurements, including preprocessing and training, were obtained on a single NVIDIA A100 GPU. The preprocessing step is lightweight and takes less than 0.1 hours. NJDTPP adopts the Euler-Maruyama scheme for numerical approximation, which leads to the largest training runtime. S2P2 uses a state space model, allowing it to remain computationally efficient even on long sequences. Although MOCHA includes the additional DAG regularization term, its runtime remains practically competitive, especially on long-sequence datasets such as CAD, SE, and ST. In our implementation, the structural constraint terms are batched across time steps on GPU, which reduces the wall-clock time in practice, although its asymptotic cost remains cubic in K.

Table 8: Training runtime comparison across TPP methods (hours↓).

Method	AKI	CAD	Sepsis	Stroke	Retweet	Taobao	StackOverflow
RMTPP	0.06	1.03	1.88	1.33	0.14	0.07	0.03
THP	0.78	14.08	40.07	5.81	0.60	0.12	0.38
AttNHP	0.71	23.51	30.10	6.92	0.58	4.62	0.91
NJDTPP	4.18	41.12	87.49	48.93	25.62	4.55	6.74
S2P2	1.91	1.27	4.24	2.33	3.26	1.01	0.94
MOCHA	0.83	2.26	8.23	5.06	0.90	0.49	0.53

K Hyperparameter Sensitivity Analysis

We analyze the sensitivity of MOCHA to key hyperparameters on two representative datasets: *Stroke (ST)*, which represents the medical setting, and *Taobao (TB)*, which represents the non-medical setting. We focus on the maximum causal order L, the sparsity coefficient γ_{sparse}, and the acyclicity coefficient $\gamma_{acyclic}$, as these hyperparameters directly control the expressiveness and structural regularity of the learned dynamic causal graphs. Unless otherwise stated, all remaining hyperparameters follow the final dataset-specific settings reported in Appendix N.3.

K.1 Sensitivity to Maximum Order L

We first examine how the maximum order L affects predictive performance. Intuitively, increasing L allows the model to capture longer multi-hop causal propagation, but a very large value may introduce unnecessary complexity, making it harder for the model to converge quickly. As shown in Figure 5, MOCHA already achieves strong performance with a moderate order, and the best results on both representative datasets are obtained at $L = 10$, which is also the setting adopted for Stroke and Taobao in Appendix N.3. The relatively high NLL on Taobao at $L = 15$ may appear counterintuitive, but it suggests that simply increasing the maximum order does not necessarily help: once the effective dependency range is exceeded, additional high-order paths can introduce noisy interactions, which may weaken generalization. This result indicates that MOCHA does not require a large maximum order to model meaningful higher-order dependencies, and that the computational complexity can be effectively controlled with a relatively small L.

K.2 Sensitivity to Sparsity Coefficient γ_{sparse}

We next study the influence of the sparsity regularizer, which encourages parsimonious causal graphs by suppressing weak edges. In this experiment, we fix $\gamma_{acyclic} = 10^{-2}$ and vary γ_{sparse}. As shown in Figure 6, MOCHA remains stable over a moderate range of sparsity coefficients, while $\gamma_{sparse} = 10^{-5}$ yields the best NLL on both datasets. When γ_{sparse} is too large, the model tends to over-penalize useful causal connections; when it is too small, the regularization effect becomes weaker and may permit redundant edges. Overall, these results support using 10^{-5} as a robust default choice, while the final per-dataset settings used in the paper remain those listed in Appendix N.3.

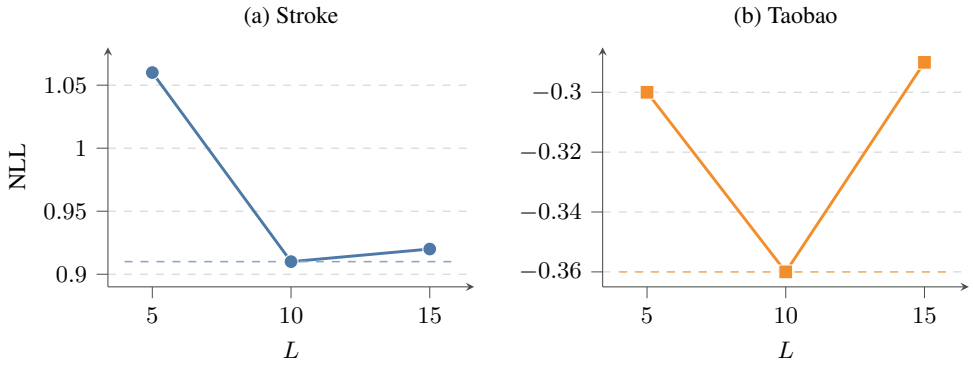

Figure 5: Sensitivity of NLL to the maximum order L on the Stroke and Taobao datasets.

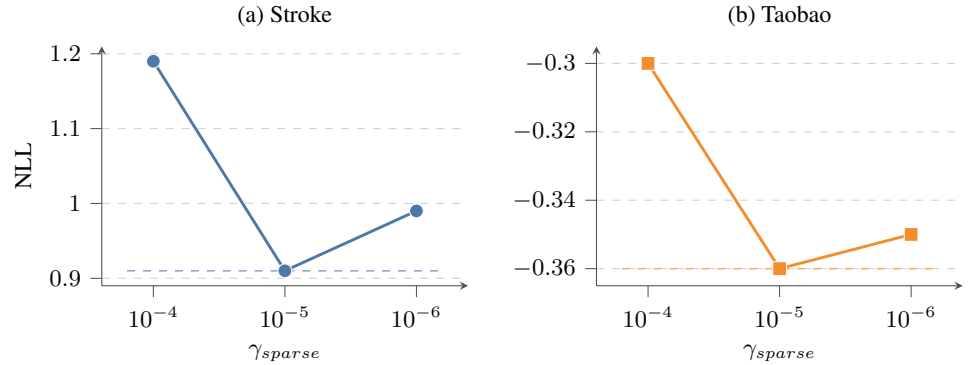

Figure 6: Sensitivity of NLL to the sparsity coefficient γ_{sparse} on the Stroke and Taobao datasets, with $\gamma_{acyclic} = 10^{-2}$.

K.3 Sensitivity to Acyclicity Coefficient $\gamma_{acyclic}$

Finally, we evaluate the coefficient that enforces the acyclicity constraint. Here we fix $\gamma_{sparse} = 10^{-5}$ and vary $\gamma_{acyclic}$. As shown in Figure 7, the performance is highly stable over 10^{-1} to 10^{-3} on both representative datasets, indicating that MOCHA is not overly sensitive to the precise strength of the acyclicity penalty once it lies in a reasonable range. A very large coefficient ($\gamma_{acyclic} = 1$) leads to a slight degradation, suggesting that excessively strong structural enforcement may limit the flexibility of early-stage optimization. In summary, the sensitivity analysis confirms that MOCHA exhibits robust behavior under moderate hyperparameter choices, with $\gamma_{sparse} = 10^{-5}$ and $\gamma_{acyclic}$ in the range of 10^{-1} to 10^{-3} offering consistently strong performance. The final settings are listed in Appendix N.3.

L Detailed Ablation Study

We conduct an ablation study to quantify the contribution of each key design component in MOCHA. Specifically, we compare five model variants that progressively introduce cross-type dependence, temporal adaptivity, and multi-order causal propagation: the univariate Hawkes process (HP) as a basic reference, the static first-order multivariate Hawkes process (MHP), the static multi-order variant (MHP+MO), the dynamic first-order variant (MHP+Dyn), and the full dynamic multi-order model (MOCHA). This comparison isolates the effect of dynamic structure learning and multi-order modeling on predictive performance. In this ablation study, we report both the average negative log-likelihood (NLL) and the corresponding standard deviation across five runs, as shown in Table 9. The NLL serves as a primary measure of model fit, reflecting how well each variant captures the observed event distributions. NLL can be negative because it is the negative logarithm of predicted probabilities. Due to model formulations involving intensities rather than probabilities, the log term can be positive. To further assess the robustness of each configuration, we additionally compute the

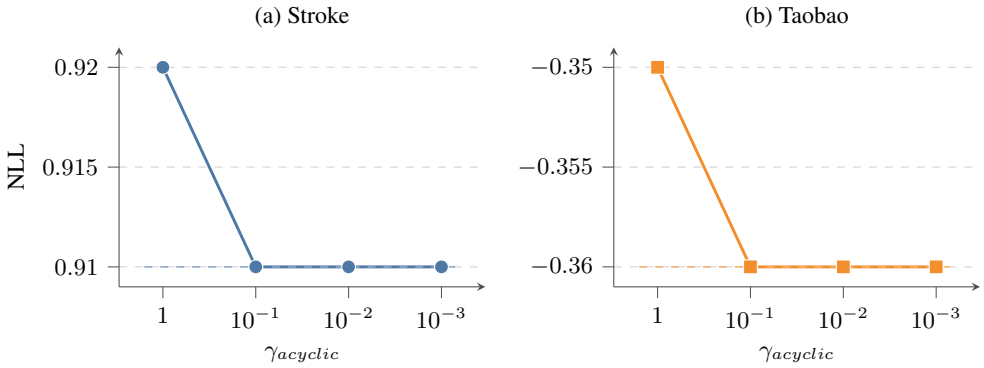

Figure 7: Sensitivity of NLL to the acyclicity coefficient $\gamma_{acyclic}$ on the Stroke and Taobao datasets, with $\gamma_{sparse} = 10^{-5}$.

standard deviation of NLL across multiple random seeds. This allows us to quantify the stability and consistency of the model under varying initializations and stochastic factors. We find that the full MOCHA model with dynamic graphs consistently achieves the best average NLL across most datasets.

Table 9: Ablation study of MOCHA NLL across seven datasets (mean ± standard deviation over five runs; ↓ is better).

Model Variant	AKI	CAD	Sepsis	Stroke	Retweet	Taobao	StackOverflow
HP	5.88±0.00	2.29±0.00	3.06±0.00	3.14±0.00	4.67±0.00	1.31±0.00	5.38±0.00
MHP	5.73±0.03	2.26±0.01	2.77±0.10	2.67±0.01	4.27±0.06	0.04±0.02	3.81±0.00
MHP+MO	5.71±0.05	1.66±0.06	2.16±0.07	2.27±0.01	4.25±0.05	-0.02±0.01	3.64±0.01
MHP+Dyn	5.29±0.17	2.10±0.27	1.77±0.08	2.21±0.37	3.78±0.25	0.01±0.01	2.37±0.01
MOCHA (MHP+MO+Dyn)	**3.80±0.10**	**0.69±0.22**	**0.70±0.08**	**0.91±0.07**	**3.51±0.06**	**-0.36±0.02**	**2.32±0.21**

M Effect of Retaining Multiple Recent Events

The matrix-based formulation in Section 3.4.1 uses the most recent event of each type to summarize historical influence. This design is intended to balance modeling expressiveness with computational efficiency, but it also introduces an approximation because earlier events of the same type are not explicitly retained in the recursive propagation. To assess the empirical impact of this design choice, we conduct an additional study in which the number of retained recent events per type is varied from 1 to 3.

We perform this analysis on two representative datasets: *Stroke (ST)*, which reflects a medical event-stream setting, and *Taobao (TB)*, which reflects a non-medical user-behavior setting. All other hyperparameters are kept the same as those used in the main experiments. Table 10 reports the resulting NLL values.

Table 10: Effect of the number of recent events retained per type on NLL (↓)

# Recent Events	Stroke	Taobao
1	0.91	**-0.36**
2	**0.87**	-0.01
3	0.88	-0.35

The results show that retaining more than the latest event does not consistently improve performance. On Stroke, using two recent events yields a small gain over the default setting, while using three events does not provide further benefit. On Taobao, the best result is already achieved by retaining only the latest event, and expanding the retained history slightly degrades performance. Overall, these findings suggest that the latest-event representation is a reasonable approximation in practice.

We attribute this behavior to two factors. (1) The influence of distant history is typically weaker than that of recent events in continuous-time event streams, so explicitly retaining additional past events of the same type offers limited incremental information. (2) In domains such as clinical monitoring, newer observations often supersede earlier states and therefore provide a more informative summary of the current system dynamics. Hence, these results support the use of the latest-event approximation as an effective design choice that preserves efficiency without materially sacrificing predictive accuracy.

N Implementation and Training Details

We provide comprehensive implementation details to ensure reproducibility and clarity of our approach. This includes descriptions of the datasets and preprocessing pipelines, evaluation metrics, and hyperparameter configurations used during training. All experiments are conducted on a single NVIDIA A100 GPU. The implementation also refers to EasyTPP [42].

N.1 Datasets and Preprocessing

To comprehensively evaluate the effectiveness and generalizability of our proposed model, we conduct experiments on seven real-world event datasets, covering both clinical and non-clinical domains. Specifically, we use four large-scale medical datasets derived from MIMIC-IV [36], each targeting a distinct clinical condition (AKI, Stroke, Sepsis, CAD), and three commonly used benchmarks for event modeling (Retweet [37], Taobao [38], and StackOverflow [39]). The medical datasets are carefully curated to reflect the irregularity, sparsity, and domain-specific causal patterns of ICU time series, while the non-medical datasets serve to validate the scalability and robustness of our model across different domains. For all datasets, we retain event timestamp information and apply minimal preprocessing to preserve the temporal dynamics essential for causal discovery.

Table 11: Statistics of benchmark datasets.

Dataset	# Types	# Sequences			Sequence Length			# Events		
		Train	Dev	Test	Min	Mean	Max	Train	Dev	Test
AKI	19	2305	326	665	2	8	58	18,270	2,468	5,408
CAD	17	847	121	242	2	295	2967	247,331	43,590	66,048
Sepsis	18	11741	1680	3354	2	367	701	428,144	61,075	122,298
Stroke	18	3194	457	915	2	82	1703	262,978	33,767	79,492
Retweet	3	9000	1535	1520	10	41	97	369,731	62,823	61,154
Taobao	17	1300	200	500	32	58	64	75,205	11,737	28,455
StackOverflow	22	1401	401	401	41	65	101	90,497	25,762	26,518

N.1.1 Medical Datasets

MIMIC-IV is a large, publicly available electronic health record database developed jointly by MIT and Beth Israel Deaconess Medical Center. It contains detailed clinical data from over 40,000 adult ICU patients treated between 2008 and 2019. The database includes multiple data types such as demographics, clinical notes, nursing records, lab tests, vital signs, medications, and diagnosis codes. In this study, four out of our seven datasets, AKI, Stroke, Sepsis, and CAD, are constructed from the MIMIC-IV database. Each is tailored to capture disease-specific physiological and biochemical patterns relevant to downstream temporal causal discovery tasks. Here are key tables used from MIMIC-IV:
patients: Contains demographic information for each patient, including age, gender, and death dates.
admissions: Records hospitalization details such as admission type, admission and discharge times.
icustays: Captures ICU stay episodes, including ICU admission and discharge times and length of stay.
chartevents: Logs bedside vital signs and nursing charted observations with timestamps.
labevents: Contains laboratory test results with numeric values and timestamps.
diagnoses_icd: Lists patient diagnoses coded by ICD standards during hospital stays.
d_items: Provides metadata and descriptions for clinical observation items recorded in other tables.
outputevents: Records output events such as fluid outputs and ventilator settings during ICU stays.

AKI [36]. The AKI dataset is constructed to analyze renal deterioration trajectories in ICU patients. We extract patient records associated with acute kidney injury (AKI) and align their ICU events leading to the onset of Phase III. Each clinical event includes one or more abnormal values from a curated set of renal- and metabolism-related variables: **Phase III, BUN, Potassiumserum, Sodiumserum, Albumin, Hemoglobin, Lactic Acid, Anion Gap, PH Arterial, Arterial CO_2 Pressure**. These indicators cover renal function, electrolyte balance, acid-base status, and metabolic activity, forming the basis for downstream temporal causal modeling. Event timestamps are preserved from the original MIMIC-IV records, and no artificial resampling is performed.

Stroke [36]. The Stroke dataset focuses on neurological deterioration and systemic response in cerebrovascular patients. We extract sequences for patients diagnosed with stroke and monitor their ICU physiological patterns across a representative variable set: **Highly Unconscious, Arterial Blood Pressure, Heart Rate, INR, Hemoglobin, Respiratory Rate, Glucose, BUN, O_2 Saturation, Temperature**. These variables span cardiovascular, hematological, respiratory, and metabolic functions, providing a rich landscape for causal pattern discovery related to stroke progression and severity. Events are recorded at irregular time points as observed in clinical practice.

Sepsis [36]. The Sepsis dataset targets patients experiencing systemic inflammatory response and multi-organ dysfunction. ICU events are sampled with a focus on inflammation, metabolic imbalance, and organ markers, using the following variables: **Low Urine, WBC, Heart Rate, Lactic Acid, Temperature, Respiratory Rate, C-Reactive Protein, Creatinine, INR, Albumin**. This dataset captures both immune and vital system dynamics, enabling the study of high-order dependencies across inflammatory and metabolic processes. All temporal irregularities are retained without imputation to reflect authentic ICU conditions.

CAD [36]. The CAD dataset is designed to model cardiovascular deterioration and myocardial stress patterns. We collect clinical event sequences from patients with coronary artery disease (CAD), using ten key physiological and biochemical variables: **Dead, Arterial Blood Pressure, Heart Rate, C-Reactive Protein, Cholesterol, Creatinine, Glucose, Potassium, Sodium, Albumin**.

These variables include inflammation, metabolic markers, electrolyte signals, and mortality endpoints, allowing our model to learn dynamic causal relationships relevant to cardiovascular decompensation. All values are timestamped from original MIMIC-IV logs.

N.1.2 Non-Medical Datasets

We evaluate MOCHA on three widely-used non-medical benchmarks to assess its generalizability in modeling diverse event domains. These datasets are commonly adopted in the temporal point process literature (e.g., EasyTPP [42]), each representing distinct real-world asynchronous event generation processes.

Retweet [37]. This dataset comprises timestamped retweet events on Twitter. Each event belongs to one of three user-size categories: small (followers<120), medium (120-1363), and large (>1363). We operate on a subset of the 5,200 most active users, each with an average sequence length of approximately 70 retweet events. This benchmark tests MOCHA's ability to capture information diffusion dynamics and cross-user cascade patterns.

Taobao [38]. This dataset records timestamped user click events on the Taobao e-commerce platform over a one-week period (November 25 - December 3, 2017). Events correspond to item clicks categorized into 20 types: the top 19 categories by frequency plus an aggregated "other" category. The processed subset contains the 4,800 most active users, each with an average sequence length of 150 events. It evaluates MOCHA on user behavior dynamics and category-level event prediction in large-scale retail settings.

StackOverflow [39]. This dataset tracks timestamped user badge awarding events on the StackOverflow platform over two years. There are 22 distinct badge types. It provides a testbed for modeling user reputation dynamics and platform-driven event prediction.

N.2 Evaluation Metrics

We provide detailed definitions of the evaluation metrics used in our experiments for point process modeling. Specifically, we introduce two fundamental metrics: Root Mean Square Error (RMSE) for timestamp prediction performance and Accuracy (Acc) for event type performance.

N.2.1 RMSE for Timestamp Prediction

In point process modeling, a model can be used to predict the timing of future events based on the learned intensity function. Given a history $\mathcal{H}_t$ up to time t, the next event time $\hat{t}$ is sampled from the conditional density function:

$$p(\hat{t}|\mathcal{H}_t) = \lambda(\hat{t}|\mathcal{H}_t) \exp\left(-\int_t^{\hat{t}} \lambda(s|\mathcal{H}_s)ds\right). \tag{71}$$

By predicting the timing of the next event, the model can capture the evolving dynamics of the sequence, offering valuable insights and data-driven support for medical diagnosis. RMSE serves as a core evaluation metric for regression tasks, quantifying the accuracy of model predictions for event occurrence times. For a given test sequence, RMSE is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{t}_i - t_i)^2} \tag{72}$$

where N denotes the total number of test events, t_i denotes the true occurrence time of the i-th event, and $\hat{t}_i$ is the corresponding time predicted by the model. This metric directly reflects the root mean square deviation between predicted and actual times, with lower values indicating higher temporal prediction accuracy.

N.2.2 Accuracy for Type Prediction

The accuracy metric primarily evaluates the performance of models in event type prediction tasks. In our implementation, the predicted type $\hat{k}_i$ is obtained by taking the $\arg\max$ over the model's output intensity vector:

$$\hat{k}_i = \arg\max_k \lambda_k(t_i) \tag{73}$$

where $\lambda_k(t_i)$ is the predicted intensity of type k at the occurrence time t_i. This reflects the model's most likely choice of event type under its learned intensity function. The accuracy is then computed over all predicted types using a binary mask to ignore padding or invalid positions in batched sequences. In multivariate TPP scenarios, accuracy is calculated as:

$$\text{Acc} = \frac{1}{N}\sum_{i=1}^{N} I[\hat{k}_i = k_i] \tag{74}$$

where $k_i \in 1, 2, \ldots, K$ denotes the ground truth type label of the i-th event, $\hat{k}_i$ denotes the predicted type, and $I[\cdot]$ is the indicator function that returns 1 if the prediction is correct, and 0 otherwise. This metric directly reflects the overall correctness of the model's event type prediction, and is widely adopted for evaluating performance in neural TPPs.

N.3 Model Hyperparameters

We report the dataset-specific hyperparameters used to train MOCHA across all experiments. The main hyperparameters include the maximum order L, the coefficient of acyclicity $\gamma_{acyclic}$, the coefficient of sparsity γ_{sparse}, the scale parameter of causal graph learning β, shown in Table 12. The threshold parameter θ for the AKI dataset is 0.2.

O Literature Basis of AKI Causal Structures

We summarize ten clinically grounded causal pathways leading to AKI Phase III, identified from the dynamic causal graphs learned by our model. These pathways are derived with reference to authoritative guidelines [43] and consolidated evidence from recent literature [44–46]. We annotate each pathway with its physiological rationale and support level.

Path 1: Lactic Acid High → Phase III
Lactic acidosis is a known biomarker of tissue hypoxia and metabolic stress, and its persistent elevation is strongly associated with AKI onset and multi-organ failure [43, 45, 46].

Table 12: Hyperparameter settings for different datasets.

Dataset	L	$\gamma_{acyclic}$	γ_{sparse}	β
AKI	18	10^{-1}	10^{-6}	10
CAD	10	10^{-1}	10^{-6}	10
Sepsis	10	10^{-1}	10^{-5}	10
Stroke	10	10^{-1}	10^{-5}	10
Retweet	2	10^{-2}	10^{-5}	10
Taobao	10	10^{-2}	10^{-5}	10
StackOverflow	20	10^{-2}	10^{-6}	10

Path 2: PH Arterial Low → Phase III
Arterial pH decline indicates worsening metabolic or respiratory acidosis, which can exacerbate renal dysfunction and correlate with poor AKI outcomes [43, 44].

Path 3: Anion Gap High → Phase III
Increased anion gap reflects unmeasured acids in the blood, often co-occurring with lactic acidosis and renal insufficiency, serving as a predictive feature for AKI progression [45].

Path 4: BUN High → Phase III
BUN is a direct measure of renal excretory function; its elevation is a hallmark of AKI and used as a primary diagnostic criterion [43, 44].

Path 5: Lactic Acid High → PH Arterial Low → Phase III
Lactate accumulation leads to metabolic acidosis, contributing to arterial pH decline. This acid-base imbalance is known to precede renal failure [45, 46].

Path 6: Anion Gap High → PH Arterial Low → Phase III
Elevated anion gap due to organic acid buildup (e.g., lactate, ketones) can reduce blood pH and impair renal perfusion, exacerbating AKI risks [43, 44].

Path 7: Arterial CO_2 Pressure High → Lactic Acid High → Phase III
Elevated CO_2 may indicate hypoventilation or impaired clearance, which can worsen tissue hypoxia, leading to lactate accumulation and renal stress [45, 46].

Path 8: Hemoglobin Low → Lactic Acid High → Phase III
Anemia reduces oxygen-carrying capacity, promoting anaerobic metabolism and subsequent lactate elevation, often observed in AKI development [44, 45].

Path 9: Sodium Low → Anion Gap High → Phase III
Hyponatremia may contribute to dilutional acidosis and disrupt fluid balance, indirectly increasing acid load and stressing renal function [43, 46].

Path 10: Potassiumserum Low → Anion Gap High → Phase III
Hypokalemia may reflect secondary responses to diuretics or volume depletion, which can shift acid-base equilibrium and precipitate acidosis [44, 46].

P Use of LLMs

We use LLMs to polish the writing. Specifically, LLMs assist in refining the grammar, clarity, and overall presentation of the paper, ensuring that the text is clear and professionally written. No experimental results or core content were generated by LLMs.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the core contributions, including dynamic causal graph learning, multi-order causal propagation, and improved predictive performance on the evaluated benchmarks and structural recovery on synthetic data. These claims are developed in Section 3 and supported by the experiments in Section 4 and Appendices H, L, and O.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses modeling scope and assumptions in Appendix A, notes that the full training theory needs additional assumptions in Appendix F, and analyzes the latest-event approximation in Appendix M. It also discusses computational trade-offs in Appendix J and sensitivity to key hyperparameters in Appendix K.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The modeling assumptions are summarized in Appendix A. Proofs for the main propositions appear in Appendices B and C, with additional theoretical results proved in Appendices D, E, and F.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The supplementary material provides the code, and the appendix together with the supplementary `README.md` describes the datasets, implementation details, and reproduction procedure. Section 4, Appendix N, and Appendix G further document the experimental setup and algorithmic workflow.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The relevant code, instructions, and supporting materials are provided in the supplementary material. Together with Section 4 and Appendix N.1, these materials provide access information and guidance for reproducing the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental setup is described in Section 4 and Appendix N, including dataset splits/statistics, evaluation metrics, key hyperparameters. Hyperparameter sensitivity and threshold selection are further discussed in Appendices H and K.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Experiments report mean $\pm$ standard deviation over five runs in Appendices H and L.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix J and supplementary material state that the experiments were run on a single NVIDIA A100 GPU 40G and reports training runtimes.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We are not aware of any aspect of the work that conflicts with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The experimental analysis discusses the potential positive impact of the method in the medical domain, where improved event prediction and causal structure discovery can support clinical understanding and decision-making. The theoretical assumptions section clarifies the model's limitations and scope boundaries. Exceeding these boundaries may lead to negative impacts.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper poses no such risks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper and the released code comply with the licenses of the assets used in this work.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The code is provided in the supplementary material in accordance with the submission requirements.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or direct interaction with human participants. Experiments are conducted on existing clinical and public event-sequence datasets.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The work does not report new studies with recruited human participants or crowdsourced workers.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: Section P states that LLMs were used only to polish writing and did not affect the core methods or experimental results. Under the checklist definition, this does not require a method-level LLM declaration.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.