APPENDIX A
OPTIMIZATION OF THE LIKELIHOOD(EM)

Here we demonstrate how to learn the parameters of Topological-Multivariate Hawkes Processes through maximum likelihood learning. Our focus is on the form of the log-likelihood function and the use of an expectation-maximization (EM) framework. Given the observations set $\mathbf{X} = \{X_{n,v,t} \mid n \in \mathbf{N}, v \in \mathbf{V}, t \in \mathbf{T}\}$, the log-likelihood function of topological-multivariate Hawkes processes on the time interval $[0, T]$ is as follows:

$$\begin{aligned}
&\mathcal{L}\left(\Theta; \mathcal{G}_V, \mathbf{X}, \mathcal{G}_N\right) \\
&= \sum_{v\in\mathbf{V}} \sum_{t\in\mathbf{T}} \sum_{n\in\mathbf{N}} P\left(X_{n,v,t} \mid H_t^{\mathbf{PA}_v}\right) \\
&= \sum_{v\in\mathbf{V}} \sum_{t\in\mathbf{T}} \sum_{n\in\mathbf{N}} \log\left[\frac{e^{-\lambda_v(n,t))\Delta t}}{X_{n,v,t}!}\left(\lambda_v(n,t)\Delta t\right)^{X_{n,v,t}}\right] \\
&\geq \sum_{v\in\mathbf{V}} \sum_{t\in\mathbf{T}} \sum_{n\in\mathbf{N}} \left[-\lambda_v(n,t)\Delta t + X_{n,v,t}\log\left(\lambda_v(n,t)\right)\right],
\end{aligned} \tag{A.1}$$

where $\Theta = \left(\mu_v, \alpha_{v',v,k}\right)_{v,v'\in\mathbf{V},k=0,\ldots,K}$ is the set of parameters modeling topological-multivariate Hawkes processes. $H_t^{\mathbf{PA}_v}$ is the historical information of the cause of the event type v occurred before time t.

According to [3], [21], the log-likelihood function is typically non-differentiable and difficult to optimize. To address this issue, an EM framework [3], [29] was developed to construct a bound on the objective function, allowing for the decoupling of parameters at each iteration such that they can be solved independently. The function $Q\left(\Theta, \Theta^{(i-1)}\right)$ is defined as follows:

$$\begin{aligned}
&Q\left(\Theta, \Theta^{(i-1)}\right) \\
&= \sum_{n\in\mathbf{N}} \sum_{v\in\mathbf{V}} \sum_{t\in\mathbf{T}} \Bigg[\sum_{n'\in\mathbf{Ne}_n} \sum_{v'\in\mathbf{PA}_v} \sum_{t'\in\mathbf{T}_{t-}} \sum_{k=0}^{K} q^{\alpha}_{n,v,t}(n', v', t', k) \\
&\quad \times X_{n,v,t} \log(\alpha_{v',v,k} \hat{A}^k_{n,n'} \kappa(t-t') X_{n',v',t'})\Bigg] \\
&\quad + \sum_{n\in\mathbf{N}} \sum_{v\in\mathbf{V}} \sum_{t\in\mathbf{T}} q^{\mu}_{n,v,t} X_{n,v,t} \log \mu_v \\
&\quad - \sum_{n\in\mathbf{N}} \sum_{v\in\mathbf{V}} \sum_{t\in\mathbf{T}} \lambda_v(n,t)\Delta t + Const,
\end{aligned} \tag{A.2}$$

where $q^{\mu}_{n,v,t}$ and $q^{\alpha}_{n,v,t}\left(n', v', t', k\right)$ are auxiliary variable parameters. $q^{\mu}_{n,v,t}$ represents the probability that event v will be triggered by the base intensity in node n at time t, while $q^{\alpha}_{n,v,t}\left(n', v', t', k\right)$ is the probability that event (n, v, t) is triggered by a previous event (n', v', t') through a path of length k [3]. In the E-step, we estimate $q^{\mu}_{n,v,t}$ and $q^{\alpha}_{n,v,t}\left(n', v', t', k\right)$ as follows:

$$\begin{aligned}
q^{\mu}_{n,v,t} &= \frac{\mu_v^{(i-1)}}{\lambda_v^{(i-1)}(n,t)} \\
q^{\alpha}_{n,v,t}\left(n', v', t', k\right) &= \frac{\alpha^{(i-1)}_{v',v,k} \hat{A}^k_{n,n'} \kappa\left(t-t'\right) X_{n',v',t'}}{\lambda_v^{(i-1)}(n,t)},
\end{aligned} \tag{A.3}$$

where

$$\begin{aligned}
&\lambda_v^{(i-1)}(n,t) = \\
&\mu_v^{(i-1)} + \sum_{v'\in\mathbf{PA}_v} \sum_{n'\in\mathbf{N}} \sum_{k=0}^{K} \alpha^{(i-1)}_{v',v,k} \hat{A}^k_{n',n} \sum_{t'\in\mathbf{T}_{t-}} \kappa\left(t-t'\right) X_{n',v',t'}.
\end{aligned} \tag{A.4}$$

In the M-step, we take the partial derivative of $Q\left(\Theta^{(i)}, \Theta^{(i-1)}\right)$ with respect to $\mu_v^{(i)}$ and $\alpha^{(i)}_{v',v,k}$, and set them to zero. This allows us to simplify the parameters $\mu_v^{(i)}$ and $\alpha^{(i)}_{v',v,k}$ as follows:

$$\mu_v^{(i)} = \frac{\sum\limits_{n\in\mathbf{N}} \sum\limits_{t\in\mathbf{T}} q^{\mu}_{n,v,t} X_{n,v,t}}{|\mathbf{N}||\mathbf{T}|\ \Delta t}, \tag{A.5}$$

$$\alpha^{(i)}_{v',v,k} = \frac{\sum\limits_{n\in\mathbf{N}} \sum\limits_{t\in\mathbf{T}} \left[\sum_{n'\in\mathbf{N}} \sum_{t'\in\mathbf{T}_{t-}} q^{\alpha}_{n,v,t}(n', v', t', k)\right] X_{n,v,t}}{\sum_{n\in\mathbf{N}} \sum\limits_{t\in\mathbf{T}} \left[\sum_{n'\in\mathbf{N}} \sum_{t'\in\mathbf{T}_{t-}} \hat{A}^k_{n',n} \kappa(t-t') X_{n',v',t'}\right] \Delta t}. \tag{A.6}$$

Algorithm 1 Optimization of the Likelihood (EM)

```
Input: Causal structure $\mathcal{G}_V$, $k$-hop, $\mathbf{X}$, $\mathcal{G}_N$
Output: Likelihood $\mathcal{L}(\Theta; \mathcal{G}_V, \mathbf{X}, \mathcal{G}_N)$
  for $v \in \mathbf{V}$ do
    intensity $\mu_v$ randomly initialized
    Infectivity matrix $\alpha_{v',v,:k} \leftarrow$ initializeWeight $(\mathcal{G}_V)$
    while $\mathcal{L}(\Theta; \mathcal{G}_V, \mathbf{X}, \mathcal{G}_N)$ not convergence do
      Compute $q^{\alpha}_{n,v,t}(n', v', t', k)$ and $q_{n,v,t}$ via Equation A.3
      Updata $\alpha_{v',v,:k}$ and $\mu_v$ via Equation A.5 and A.6
    end while
  end for
  return $\mathcal{L}(\Theta; \mathcal{G}_V, \mathbf{X}, \mathcal{G}_N)$
```

APPENDIX B
DIRECTED ACYCLIC GRAPH (DAG) CONSTRAINT

The Directed Acyclic Graph (DAG) is a graph used to represent the causal relationships between variables. In this graph, edges between nodes indicate causal influences. An acyclicity constraint is a commonly used constraint that ensures the absence of cycles in the graph. This constraint guarantees that there are no paths that begin at a node and eventually loop back to that same node, following the direction of the edges. Enforcing an acyclicity constraint is crucial because the existence of cycles in the graph would imply the presence of feedback loops, which can cause instability and complicate the reasoning about the causal relationships between variables. Nowadays, this constraint is typically enforced using techniques such as those employed in CausalVAE.

APPENDIX C
THEORETICAL ANALYSIS OF ARM-HP ALGORITHM EFFICIENCY

By introducing prior knowledge and ree-step optimization strategies, ARM-HP can significantly reduce the search space. Specifically, suppose the event sequence data contains $|\mathbf{V}|$ event types, the typical method needs to search up to

$2^{|\mathbf{V}|(|\mathbf{V}|-1)}$ candidate causal graphs. The upper bound of the number of candidate graphs for ARM-HP is derived through theoretical analysis of its three optimization steps: The first step constructs the initial graph by selecting ϵ edges based on the cause score and removing redundant edges, which requires evaluating at most 2^{ϵ} candidate graphs; The third step determines the potential causal graph based on the gain score. In each iteration, neighboring causal graphs are generated by removing or adding an edge, with the total number of neighboring causal graphs being $|\mathbf{V}|(|\mathbf{V}|-1) \cdot \rho$. Iterating n_{epoch} times requires evaluating $n_{epoch} \cdot |\mathbf{V}|(|\mathbf{V}|-1) \cdot \rho$ candidate graphs. Therefore, the upper bound of the total number of candidate graphs that ARM-HP needs to evaluate is $2^{\epsilon} + n_{epoch} \cdot |\mathbf{V}|(|\mathbf{V}|-1) \cdot \rho$.

Appendix D
System

As shown in the Fig. 1, we present a typical system architecture consisting of two modules: an online process and an offline process. The online process module records relevant data in real-time and stores it, while the offline process is mainly responsible for processing time series data, extracting Granger causality, and encapsulating it as a service interface for user invocation.

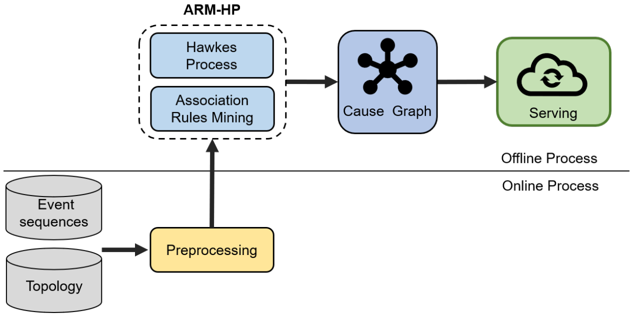

Fig. 1. Illustration of the system.