A Synthetic Data Experiments

Here we provide further details about the synthetic datasets considered for the experiments.

A.1 5 Models

We provide details about the 5 PGEMs in Figure 1. In what follows, the windows corresponding to the parents are listed in the same order as parents. We use binary vectors to indicate parental states, again in the same order as listed parents. For instance, if a node A has parents $[B, C]$ then windows $[15, 30]$ represent information that the windows from B and C to A respectively are 15 and 30. The binary parental state $[0, 1]$ implies that only C has occurred in its window, whereas $[1, 1]$ represents the case where both B and C have occurred in their respective windows.

Model 1

- parents = {'A': [], 'B': [], 'C': ['B'], 'D': ['A', 'B'], 'E': [C']}
- windows = {'A': [], 'B': [], 'C': [15], 'D': [15, 30], 'E': [15]}
- lambdas = { 'A': {[]: 0.2}, 'B': {[]: 0.05}, 'C': {[0]: 0.2, [1]: 0.3}, 'D': {[0, 0]: 0.1, [0, 1]: 0.05, [1, 0]: 0.3, [1, 1]: 0.2}, 'E': {[0]: 0.1, [1]: 0.3}, }

Model 2

- parents = {'A': ['B'], 'B': ['B'], 'C': ['B'], 'D': ['A'], 'E': ['C']}
- windows = {'A': [15], 'B': [30], 'C': [15], 'D': [30], 'E': [30]}
- lambdas = { 'A': {[0]: 0.3, [1]: 0.2}, 'B': {[0]: 0.2, [1]: 0.4}, 'C': {[0]: 0.4, [1]: 0.1}, 'D': {[0]: 0.05, [1]: 0.2}, 'E': {[0]: 0.1, [1]: 0.3} }

Model 3

- parents = {'A': ['B', 'D'], 'B': [], 'C': ['B', 'E'], 'D': ['B'], 'E': ['B']}
- windows = {'A': [15, 30], 'B': [], 'C': [15, 30], 'D': [30], 'E': [30]}
- lambdas = { 'A': {[0,0]: 0.1, [0,1]: 0.05, [1,0]: 0.3, [1,1]: 0.2}, 'B': {[]: 0.2}, 'C': {[0,0]: 0.2, [0,1]: 0.05, [1,0]: 0.4, [1,1]: 0.3}, 'D': {[0]: 0.1, [1]: 0.2}, 'E': {[0]: 0.1, [1]: 0.4} }

Model 4

- parents = {'A': ['B'], 'B': ['C'], 'C': ['A'], 'D': ['A', 'B'], 'E': ['B', 'C']}
- windows = {'A': [15], 'B': [30], 'C': [15], 'D': [15, 30], 'E': [30, 15]}
- lambdas = { 'A': {[0]: 0.05, [1]: 0.2}, 'B': {[0]: 0.1, [1]: 0.3}, 'C': {[0]: 0.4, [1]: 0.2}, 'D': {[0, 0]: 0.1, [0, 1]: 0.3, [1, 0]: 0.05, [1, 1]: 0.2}, 'E': {[0, 0]: 0.1, [0, 1]: 0.02, [1, 0]: 0.4, [1, 1]: 0.1} }

Model 5

- parents = {'A': ['A'], 'B': ['A', 'C'], 'C': ['C'], 'D': ['A', 'E'], 'E': ['C', 'D']}
- windows = {'A': [15], 'B': [30, 30], 'C': [15], 'D': [15, 30], 'E': [15, 30]}
- lambdas = { 'A': {[0]: 0.1, [1]: 0.3}, 'B': {[0,0]: 0.01, [0,1]: 0.05, [1,0]: 0.1, [1,1]: 0.5}, 'C': {[0]: 0.2, [1]: 0.4}, 'D': {[0, 0]: 0.05, [0, 1]: 0.02, [1, 0]: 0.2, [1, 1]: 0.1}, 'E': {[0, 0]: 0.1, [0, 1]: 0.01, [1, 0]: 0.3, [1, 1]: 0.1}, }

A.2 20 Randomly Generated Models

PGEMs were randomly generated similar to the approach described in the supplementary material in Bhattacharjya *et al.* [2018]. For a PGEM over label set $\mathcal{L}$, for each node, the number of its parents K are chosen uniformly from the parameters $K_{min} = 0, \cdots, K_{max} = \lfloor |\mathcal{L}|/2 \rfloor$ in integer increments. A random subset of size K from $\mathcal{L}$ is then chosen as its parent set. Windows for each edge are

generated uniformly from $w_{min} = 15$ to $w_{max} = 30$ in increments of $\Delta w = 5$. For the conditional intensity rates, we assume that each node's parent either has a multiplicative amplification or damping rate beyond a baseline rate of $r/|\mathcal{L}|$, where r is generated uniformly between $r_{min} = 0.05$ and $r_{max} = 0.2$. Nodes that always increase occurrence rate for their children are obtained by randomly choosing a subset $\mathcal{L}_A$ of size $K_A = \lfloor |\mathcal{L}|/2 \rfloor$ from $\mathcal{L}$. Nodes in the sets $\mathcal{L}_A$ and $\mathcal{L} \backslash \mathcal{L}_A$ have an amplification and damping rate of $\gamma_A = 1.5$ and $\gamma_D = 0.25$ respectively. These numbers are chosen to roughly keep the number of events N generated by each model to be commensurate with T, but this is not enforced rigorously, allowing the dataset sizes to vary across models.

A.3 Threshold Grid and Baseline Information

The PGEM BIC learner was run with window increment $\epsilon = 0.001$ for window search.

CPCIM was deployed with the following hyper-parameters. The structural prior κ was set to 0.1. For conjugate prior pseudo-count α and pseudo-duration β for each label, we used identical values for all labels. We compute ratio ρ of the total number of all arrivals over all labels to the total duration for all labels (the product of the number of labels and the horizon T under consideration) which provides an empirically based estimate of the arrival rate. We ran experiments using $\alpha = K\rho$, $\beta = K$, for various values of $K = 10, 20, \ldots$, where higher values of K increase the prior's influence. $K = 20$ was chosen. Intervals of the form $[t - t^*, t)$ are the basis functions, where we chose $t^* \in \{1, 2, 3, 4, 5, 6, 7, 15, 30, 45, 60, 75, 90, 180\}$.

Tester threshold parameters for training were chosen from:

- NI Tester: $\{0.3, 0.25, 0.2, 0.15, 0.1, 0.05, 0.03\}$
- LR Tester: $\{0.99, 0.95, 0.9, 0.8, 0.7, 0.6, 0.5\}$

B Proofs

B.1 Proof for Lemma 4

This follows from the definition of a GEM. For a PGEM, if a node X has parent nodes $\mathbf{U}$, then at any time with parent condition $\mathbf{u}$ (as determined by recent historical occurrences in the corresponding windows), the rate at which X occurs is $\lambda_{x|\mathbf{u}}$. Additional information about historical occurrences of any non-parent has no effect on the conditional intensity rate at any time, therefore process independence is true for any non-parent given the history of parent event labels.

B.2 Proof for Theorem 7

The global Markov property is satisfied when any δ^*-separation statement for valid $X, Y, \mathbf{Z}$ implies process independence $Y \not\rightarrow X|\mathbf{Z}$. The separation itself is defined based on a graph that cuts outgoing edges, except self-loops, from X. We refer to this graph as $\tilde{G}_X^D$; the superscript indicates the graph is directed. Consider an undirected graph formed from $\tilde{G}_X^D$ by taking the subgraph over ancestors of X, Y, $\mathbf{Z}$, and connecting edges between any parents with common children if they are not already connected (this operation is known as 'moralizing'). We denote this as $\tilde{G}_X^U$.

Suppose X is δ^*-separated from Y given $\mathbf{Z}$ in $\tilde{G}_X^D$. We consider a node to be a blocker in a path if it prevents a path from connecting X and Y given $\mathbf{Z}$ for the separation criterion under consideration. Note that if a path is being blocked by a non-collider in $\tilde{G}_X^D$, it will also be blocked by that non-collider in the undirected version $\tilde{G}_X^U$. Consider a path that is blocked by a collider in $\tilde{G}_X^D$. In this path, X must have an incoming edge in $\tilde{G}_X^D$ as outgoing edges have been removed. Furthermore, the collider must not be an ancestor of $\mathbf{Z}$ as it is a path blocker. There must be a path in $\tilde{G}_X^U$ from the corresponding collider node to either X or Y. It can be shown that a violation occurs for the assumption of X being δ^*-separated from Y given $\mathbf{Z}$; some other non-blocking path must exist between X and Y as this path cannot include $\mathbf{Z}$, otherwise the collider would be an ancestor of $\mathbf{Z}$, which is not possible. The original path must be blocked by the collider in $\tilde{G}_X^U$. The result follows from applying Theorem 3.4 in Didelez [2008] which uses graphical separation in $\tilde{G}_X^U$.

B.3 Proof for Theorem 9

The PC algorithm for GEMs is a variation on PC for Bayesian networks with the additional point of simplification that a step for orienting edges to adhere to acyclicity constraints is not needed. Note that the global dynamic Markov property applies to a PGEM from a prior theorem. Together with the causal dependence assumption, this implies that the independencies in the underlying marked point process are the same as those that can be determined from δ^*-separation in the graph. The argument that a perfect process independence tester with the GEMs PC algorithm results in sound and consistent learning follows the argument for the PC algorithm for Bayesian networks. The equivalence of independencies ensures that PC produces no false positives, while the causal dependence assumption ensures no false negatives.