Causal Discovery under Time-Varying Delays

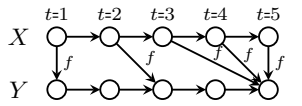
Lénaïg Cornanguer David Kaltenpoth Jilles Vreeken
CISPA Helmholtz Center for Information Security
{name.surname}@cispa.de

Abstract

Causal discovery aims to uncover the causal relationships among variables beyond correlations from observational data. On time series, a major challenge is non-stationarity, which is typically modeled as regime-based shifts or drifting causal strength. In contrast, we consider non-stationarity arising from varying cause-to-effect delay, and introduce a causal model that is identifiable with few assumptions. To discover causal relationships with varying causal delay in practice, we formalize the problem in terms of the algorithmic model of causation, and propose STRETCH to discover causal graphs. We show on synthetic data that our method correctly recovers the causal relationships, and show the relevance of this causal model on a case study on the impact of El Niño climatic phenomenon on Indian monsoon.

1 Introduction

Causal discovery aims to uncover cause-effect relationships among variables from observational data [19]. To go beyond correlations, causal discovery methods leverage conditions that hold only in the presence of a causal relationship, and are typically based on conditional independence tests [27] or restricted model classes [7]. Time series data introduce additional challenges, including identifying the delay between a cause and its effect, and managing potential nonstationarity, which might conceal the causal relationships. Usually, nonstationarity is viewed as a change in the nature or existence of a causal link, manifesting as a sudden regime shift [15, 22] or gradual functional drifts [9, 10]. However, a third, equally critical form of change often goes ignored: *the variation of the causal delay itself*. As delays dictate how information propagates through a system, accurately capturing their evolution is essential for robust causal reasoning and a deeper understanding of dynamic processes.



(a) Full causal graph. (b) Summary causal graph.

Figure 1: In our setup, nonstationarity arises from a varying cause-to-effect delay.

In this paper, we consider drifting nonstationarity arising from changes in the causal delay between two variables (Fig. 1a), while the summary causal links remain over time (Fig. 1b). Examples of where the delays between cause and effect vary include information transmission speed in networks given the medium deterioration, river discharge that is higher when the soil is already saturated, or climatic phenomena whose effect is concentrated on a specific period of time. In such settings, it is *not possible* to answer counterfactual questions ("*What would have happened if we had intervened at that specific time?*") unless we allow for varying delay: a fixed delay would provide the wrong timing and quantity for the predicted outcome. To address this setup, we propose a structural causal model for time series with a delay function instead of a fixed delay, and prove the identifiability of the causal graph and delay function.

We formalize the problem of identifying such causal relationships using the algorithmic model of causation (AMC) [12], instantiated into a score via the Minimum Description Length principle [6], and present a method (STRETCH) for discovering causal graphs in practice. In particular, we estimate

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

the delay function for a child variable and its parent using a hidden Markov model (HMM) [21] where the state at time t indicates the specific set of parent observations (e.g., in Fig. 1, the parent set of $Y_{t=3}$ is $\{X_{t-1}\}$), and the emission probabilities reflect the child data likelihood given the parent sets. This formulation provides a finite parameter set without restricting delay variations. The causal function (f) is estimated concurrently with the delay through the HMM training via weighted regression. Finally, MDL serves as a theoretically grounded proxy for Kolmogorov complexity [16], enabling the selection of true causal relationships from candidate edges.

Through extensive experiments on synthetic and real-world data, we show that STRETCH performs on par with SoTA methods when delays are static, and outperforms them for non-static delays.

In short, the key contributions of this paper are : (1) the introduction of a new identifiable causal model for nonstationary time series, (2) STRETCH, for discovering causal graphs and varying cause-to-effect delays, and (3) extensive experiments on synthetic and real-world data.

2 Preliminaries

In this section we introduce the notation and assumptions used throughout the paper. We begin by describing causal graphs for discrete-time series and introduce which deviations from stationarity we are interested in. We then introduce the algorithmic model of causation as the principle behind our score, and the Minimum Description Length principle as its computable instantiation.

2.1 Causality in Time Series

Let $\mathbf{X} = \{X^1, \dots, X^N\}$ denote a multivariate time series observed at regular discrete times $t \in \{1, \dots, T\}$, where X_t^i is the value of variable i at time t. We are interested in causal relations between variables whose effects may arrive after a delay that may change over time. We therefore give here only the graph representations needed to distinguish variable-level causal relations from their time-indexed realizations; see Assaad et al. [2] for a wider survey.

The most explicit representation is the *full (or unrolled) causal graph*, which forms a directed acyclic graph (DAG; see Fig. 1a). Its nodes are individual observations X_t^i, and its edges encode direct causal effects from a source time to a target time, with realized lag $\tau = t' - t \geq 0$.

Definition 2.1 (Full Causal Graph)**.** Let $\mathbf{X} = \{X_t^i\}$ be a multivariate time series indexed by variable $i \in 1, \dots, N$ and time $t \in \{1, ..., T\}$. The full causal graph $G = (V, E)$ consists of one node X_t^i per pair (i, t), and contains an edge $X_t^i \to X_{t'}^j$ if and only if the former directly causes the latter, with lag $\tau = t' - t \geq 0$. We assume that the graph is acyclic, so that in particular $\tau > 0$ if $i = j$.

Many time series methods assume that edge patterns are repeated over time—if $X_t^i \to X_{t+\tau}^j$ holds for any t, then it holds for all $t \in \{1, ..., T\}$. In this case, the unrolled causal graph can be reduced to a *window causal graph* of some fixed size τ_{max} that represents the causal relationships and their associated fixed delay for any t. We relax precisely this assumption: a variable-level causal relation persists, but the arrival time of each individual causal effect may vary.

In many cases, we may be more interested in the larger scale relationships between variables X^i and X^j, which is contained in the *summary causal* graph over variables (Fig. 1b). This graph records which variables can directly affect which others at any lag, and is the object used for high-level causal interpretation and graph search. The full graph, together with the delay functions introduced later, specifies when these summary-level effects arrive. Because variables can affect each other with positive lags, the summary graph itself may contain cycles.

Definition 2.2 (Summary Causal Graph)**.** For the time series $\mathbf{X} = \{X_t^i\}$, the summary causal graph $G = (V, E)$ consists of one vertex per variable X^i, and there is an edge $X^i \to X^j$ if and only if there exists some time t and lag $\tau \geq 0$ such that $X_t^i \to X_{t+\tau}^j$ is an edge in the full causal graph.

Throughout, we view these graphs as structural causal graphs for a *causally sufficient* observed system, and assume the joint distribution is *Markov* and *faithful* to the full causal graph. These standard assumptions allow us to relate these graphical structures to conditional independencies. To avoid ambiguity, we write $\mathrm{Pa}(X_t^j)$ for the time-indexed parents of an observation in the full graph and $\mathrm{Pa}(j)$ for the parent variables of X^j in the summary graph.

2.2 Algorithmic Causal Discovery

Having introduced causal model representations, we now introduce a principled criterion for choosing among candidates. The Algorithmic Model of Causality (AMC) [12] states that the true causal explanation has the shortest independent description of its mechanisms. Formally, the description length is measured by the Kolmogorov complexity [14] (K). For a distribution P, $K(P)$ is the length of the shortest program computing $P(x)$ to arbitrary precision uniformly in x. In our setting, this description includes the functional relations and the process indicating the parents at time t.

In the bivariate case where X causes Y, AMC postulates that the cause and the conditional mechanism share no algorithmic information. Equivalently, their joint description satisfies $K(P(X,Y)) \stackrel{+}{=} K(P(X)) + K(P(Y \mid X))$, up to a constant independent of the distribution P. While the reverse factorization $P(Y)P(X \mid Y)$ is always mathematically valid, it couples the parameters of the two factors and does not lead to an equally short description.

For a multivariate causal graph G^*, the same principle yields an optimal description $K(P(X_1, \ldots, X_m)) \stackrel{+}{=} \sum_j K(P(X_j \mid \mathrm{Pa}_j^*))$. For time series, this statement applies to the full graph: once the active time-indexed parents of X_t^j are known, non-descendants contribute no additional algorithmic information. Consequently, both the summary graph and the delay processes that select active parents are part of the causal explanation we later score.

The AMC gives us the conceptual target for causal discovery: prefer the summary graph, delay processes, and functional mechanisms that yield the shortest joint description of the observed time series. The remaining challenge is that the ideal Kolmogorov code is not computable [14], so we next introduce a statistically well-founded approximation.

2.3 Minimum Description Length Principle

As Kolmogorov complexity cannot be computed in general, we use the Minimum Description Length (MDL) principle [6], which compares candidate explanations by the number of bits needed to transmit both the model and the data given the model. This provides a practical bridge from AMC to a score over candidate parent sets, functional mechanisms, and delay processes [16].

Given a model class $\mathcal{H}$, a two-part MDL score describes the data by separately encoding a model $h \in \mathcal{H}$, and the data D given this model, so as to minimize the total description length,

$$L(D, \mathcal{H}) = \min_{h \in \mathcal{H}} \{ \underbrace{-\log P(D \mid h)}_{\text{data cost}} + \underbrace{L(h)}_{\text{model cost}} \},$$

where L is the number of bits required to describe the model. The data term rewards good model predictions, and the model term penalizes complexity.

If $\mathcal{H}$ contained all computable programs, the MDL score would recover Kolmogorov complexity up to an additive constant. For any restricted model class, MDL instead gives an upper bound, $K(D) \stackrel{+}{\leq} L(D, \mathcal{H})$ [14]. The usefulness of the score therefore depends on choosing an encoding for L that captures the causal structure of interest. In Sec. 4.5, this will mean encoding the selected causal parents, regression parameters, and the Markov parameters governing the delay function.

3 Theory

In this section we formalize the precise causal model we are interested in and provide conditions under which we can prove identifiability of the full causal graph.

3.1 Causal Model

In our causal model's summary graph, an edge $X^i \to X^j$ means that each observation of X^i sends a direct causal effect to some observation of X^j after some delay τ. Classical stationary time-series models assume that this delay is the same for all t. We instead associate every summary edge with its own delay function $g_{i \to j} : \mathbb{N} \to \{0, \ldots, \tau_{max}\}$, so that the effect sent by X_t^i arrives at $X^j_{t+g_{i \to j}(t)}$.

Fig. 1a illustrates why we define delays from the parent's time index. At every time t, the parent observation X_t^i emits *exactly one* causal effect to X^j, and the delay function determines when that

effect reaches the child, at $X^j_{t+g_{i\to j}(t)}$. If the delay increases, some X^j_s may receive no effects at all, if it decreases, several parent observations may arrive at the same X^j_s. Thus, varying delays change the active time-indexed parents of X^j_s while preserving the underlying summary edge $X^i \to X^j$.

Let $\mathrm{Pa}(j)$ be the parents of X^j in the summary causal graph. We model X^j_t as a first order autoregressive process whose exogenous inputs at each time step t are precisely the parent observations arriving at that time, identified by the function $c(t,\tau)$ (which translates the g into the child perspective):

$$X^j_t = a_j X^j_{t-1} + \sum_{i:X^i\in\mathrm{Pa}(j)} \sum_{\tau=0}^{\tau_{max}} c_{i\to j}(t,\tau) f_{i\to j}(X^i_{t-\tau}) + \varepsilon_{j,t} \ , \qquad c_{i\to j}(t,\tau) = \mathbb{1}[g_{i\to j}(t-\tau)=\tau] \ . \tag{1}$$

where $a_j \neq 0$ is the autoregressive coefficient for variable j, $\{\varepsilon_{j,t}\}$ are jointly independent and centered innovations, and τ_{max} is the maximum causal delay considered. We assume that $f_{i\to j}$ are non-constant functions. Additionally, $g_{i\to j}$ preserves temporal ordering, $t + g_{i\to j}(t) \leq t' + g_{i\to j}(t')\ \forall t' > t$, ruling out crossing effects where a later cause would arrive at an earlier effect.

With model and assumptions in place, we next consider whether all delay functions are identifiable.

3.2 Identifiability

Since every delay function induces specific edges in the full graph, identifying the full true graph is equivalent to identifying all delay functions for non-self summary edges.

Theorem 3.1 (Delay identifiability)**.** *Consider the multivariate SEM of Eq.* (1)*, let $G^\star = (V, E^\star)$ be the full causal graph, and assume Markov, faithfulness, and causal sufficiency. Then the graph $G^\star$ is identifiable from the observational distribution $P(X)$, and so are all delay functions $\{g_{i\to j} : i \neq j\}$.*

The proof can be found in Appendix A.1. Intuitively, since every specific delay function induces a unique set of edges in the full causal graph, identifying the true full graph is equivalent to identifying all delay functions. The unique causal structure—and thus the underlying delay function—is identifiable because any other delay configuration would violate the observed conditional independence relationships. In particular, time-varying strength models cannot be observationally equivalent.

Corollary 3.2 (Varying strength and varying delay are not equivalent)**.** *If g is not constant, then no varying-strength model with fixed delays that is Markov and faithful to its own graph G can be observationally equivalent to our model in Eq.* (1)*.*

4 Algorithm

To discover causal graphs and delays in practice, we introduce STRETCH, an algorithm to discover jointly the causal graph underlying observational data and the causal delay functions.

4.1 Overview

Given observational data, a variable X^j, and a candidate parent set $\widetilde{\mathrm{Pa}}(j)$, we fit the SEM defined in Eq. (1). To this aim, we need to learn the causal delay function $g_{i\to j}$, as well as the functional mechanism $f_{i\to j}$ for each parent variable $X^i \in \widetilde{\mathrm{Pa}}(j) \setminus \{X^j\}$. These problems are entangled as we need the true delay function to fit the functional mechanism correctly, and we need the true functional mechanism to correctly identify the delay function. We solve this joint problem using an expectation-maximization (EM) approach to learn both functions separately.

Once fitted, the candidate causal edges are evaluated using the MDL principle within a causal graph search procedure or exhaustive enumeration to discover the summary causal graph.

4.2 Regression Model

To fit the functional mechanism f, we require a regression method that can capture both linear and nonlinear relationships as our causal model, is optimizable by expectation-maximization to concurrently fit the causal mechanism and the delay function, and whose parameters can be encoded via the MDL

principle to reflect the model complexity. Polynomial regression fulfills these requirements. An observation X_t^j is estimated from its parents in the candidate full causal graph $\widetilde{\mathrm{Pa}}(X_t^j) = \{X_{t'}^i\}_{t'}$ as

$$\tilde{X}_t^j = \varphi_0 + \varphi_1 X_{t-1}^j + \sum_{X_{t'}^i \in \widetilde{\mathrm{Pa}}(X_t^j) \setminus X_{t-1}^j} \sum_{m=1}^{M} \varphi_{i,m} {X_{t'}^i}^m$$

where φ_0 and φ_1 are the intercept and autoregression parameters, respectively, and $\varphi_{i,m}$ is the coefficient for a given parent variable at a given degree $m \in M$.

Assuming Gaussian noise, the log likelihood of observations given this regression model is given by

$$\log \mathcal{L}(X^j) = \log \mathcal{N}(X^j; \tilde{X}^j, \sigma^2) = -\frac{T}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2 \ln(2)} \sum_{t=1}^{T} (X_t^j - \tilde{X}_t^j)^2 \tag{2}$$

where throughout this section, we write $\log = \log_2$, and denote $\ln$ explicitly.

4.3 Hidden Markov Model

To fit the regression model defined above, we need to know the set of parents at each time step, i.e., for each variable X^i, which observations within the past τ_{max} time steps have sent a causal effect to the child observation X_t^j. As we make the reasonable assumption that the temporal ordering of the causal contributions is preserved, the evolution of the delay function is constrained and $g(t) - g(t-1) \geq -1$. Consequently, for each parent variable, the set of observations belonging to $\mathrm{Pa}(X_t^j)$ depends on the set of observations in $\mathrm{Pa}(X_{t-1}^j)$. For each variable causing X^j in the summary graph, we model the parent set of X^j at time t as a Markov process where the states are binary vectors of size $\tau_{max} + 1$ indicating if the parent variable at a relative time lag causes the specific child observation:

$$\left[\mathbb{1}[X_{t-\tau_{max}}^i \to X_t^j], \mathbb{1}[X_{t-(\tau_{max}-1)}^i \to X_t^j], \cdots, \mathbb{1}[X_{t-0}^i \to X_t^j] \right] .$$

As an example, we illustrate in Fig. 2 a parent set and the resulting Markov process state. The parent set of $X_{t=8}^j$ is constrained by the one at the last time step in the following way: $X_{t-2}^i \in \mathrm{Pa}(X_{t=7}^j) \Rightarrow X_{t-3}^i \notin \mathrm{Pa}(X_{t=8}^j)$ (each observation sends exactly one causal effect to the child variable) and $X_{t-2}^i \in \mathrm{Pa}(X_{t=7}^j) \Rightarrow X_{t-4}^i \notin \mathrm{Pa}(X_{t=8}^j)$ (temporal ordering).

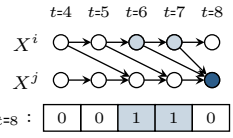


Figure 2: True HMM state ($q_{t=8}$) for $X_{t=8}^i$.

Since we do not directly observe the parent set process, and we want to recover it from the observations, we define a hidden Markov model (HMM) [21]. For a single parent variable $\widetilde{\mathrm{Pa}}(j) = \{X^i\}$, the HMM is a tuple $\langle Q, \Pi, A, B \rangle$ where $Q = \{1, u, v, ..., L\}$ is the state space and $q_t \in Q$ is the hidden state at time t, $\Pi = \{\pi_u\}$ are the initial state probabilities, $A = \{a_{uv}\}$ are the state transition probabilities, and $B = \{b_{tu}\}$ are the emission probabilities which correspond to the probability of the child observation at time t given the parent set described by state q_t and the regression model defined in Sec. 4.2.

For multiple parent variables ($|\widetilde{\mathrm{Pa}}(j)| > 1$), the HMM becomes factorial ($\langle \boldsymbol{Q}, \boldsymbol{\Pi}, \boldsymbol{A}, B \rangle$) with independent Markov processes for each delay function, coupled through the joint emission probabilities (B) when considering the child. We focus on the univariate case, as the underlying theory of the EM optimization remains consistent when evaluating the state transitions (details in App A.2).

4.4 Parameter Estimation via Expectation-Maximization

The estimation of the regression parameters (φ) requires knowing the delay function, and the estimation of the parameters of the delay function (Π and A) relies on the regression fit (via B). To solve this co-dependence, we use an EM algorithm that will update all these parameters successively and reach convergence over iterations of E- and M-steps.

E-step. During the expectation (E) step, we compute the likelihood of all HMM states over time, i.e., of the possible delay functions, given the current model parameters.

We first compute the emission probabilities B (observation likelihood) for each state and time step. Using the regression model defined in Sec. 4.2 with the parents' observation at the lag indicated by the state vector ($\mathrm{Pa}(X_t^j) = \bigcup_{\tau=0}^{\tau_{max}} \{X_{t-\tau} | q_t[\tau_{max} - \tau]\}$) and the current regression coefficients φ, we compute the expected value $\tilde{X}_t^j$. The emission probability for a state u at time t is defined as the likelihood of observing X_t^j under the regression model, assuming Gaussian residuals (Eq. (2)), $b_{tu} = p(X_t^j \mid q_t = u) = \mathcal{N}(X_t^j; \tilde{X}_t^j, \sigma^2)$.

We then run the forward-backward algorithm [21] to compute for each time step t the probability of being in a given state $q_t = u$, which depends on both transition (A) and emission (B) probabilities, as

$$\gamma_t(u) = \frac{\alpha_t(u)\beta_t(u)}{\sum_{v=1}^{L} \alpha_t(v)\beta_t(v)}$$

with $\alpha_t(u)$ the probability of ending up in any particular state given the first t observations ($\alpha_t(u) = p(X_{1:t}^j, q_t = u) = \sum_{v=1}^{L} \alpha_{t-1}(v) a_{vu} b_{tu}$) and $\beta_t(u)$ the probability of observing the remaining observations given any starting time t ($\beta_t(u) = p(X_{t+1:T}^j \mid q_t = u) = \sum_{v=1}^{L} \beta_{t+1}(v) a_{uv} b_{t+1,v}$).

Similarly, we compute the probability of transitioning from state u to state v at time t as

$$\xi_t(u, v) = p(q_t = u, q_{t+1} = v \mid X_{1:T}^j, \varphi, A, \Pi) = \frac{\alpha_t(u) a_{uv} b_{t+1,v} \beta_{t+1}(v)}{\sum_{w=1}^{L} \alpha_t(w)\beta_t(w)} \,.$$

M-step. During the maximization (M) step, we re-estimate the parameters (A, Π, φ, σ^2) by maximizing the likelihood using the computation from the E step (γ and ξ). We update the transition probabilities A and initial state probabilities Π by normalizing the expected transition counts

$$a_{u,v} = \frac{\sum_{t=1}^{T-1} \xi_t(u, v)}{\sum_{w \in Q} \sum_{t=1}^{T-1} \xi_t(u, w)} \,, \quad \pi_u = \frac{\gamma_{t=1}(u)}{\sum_{v \in Q} \gamma_{t=1}(v)} \,.$$

For the regression parameters φ, we solve the Weighted Least Squares (WLS) problem $\hat{\varphi} = (\mathbf{X}^T W \mathbf{X})^{-1} \mathbf{X}^T W Y$ where $\boldsymbol{X} = \mathrm{Pa}(X_t^j)$, $Y = X_t^j$ and with weights $W = \gamma_t(u)$. Intuitively, we fit the regression using all possible parent configurations over all time steps, but giving less importance to parent sets that are unlikely given the past child observations.

4.5 Causal Edge Discovery

Following the algorithmic model of causality (Sec. 2.2), we score a parent set $\mathrm{Pa}(j) \subseteq \{1, ..., N\}$ by approximating the Kolmogorov complexity $K(P(X^j \mid \mathrm{Pa}(j))) + K(P(\mathrm{Pa}(j)))$ using a two-part MDL score (Sec. 2.3): $L(D, h) = L(D \mid h) + L(h)$.

The cost of the data $L(D \mid h)$ corresponds to the negative log-likelihood of the child observations given the model, which consists of the autoregressive model ($\theta = \{\varphi\}$), plus the HMM model if the parent set contains other variables $i \neq j$ ($\theta = \{\varphi, A, \Pi, B\}$). In the latter case, the likelihood takes into account the HMM state sequence to reflect the probability of the delay functions on top of the regression likelihood. We compute the data cost as

$$L(D \mid h) = -\log P(X_{1:T}^j \mid \theta) = \begin{cases} -\sum_t^T \log \mathcal{N}(X_t^j; \tilde{X}_t^j, \sigma^2) & \text{for } \mathrm{Pa}(j) = \{j\}, \\ -\log \sum_u^Q \alpha_T(u) & \text{else.} \end{cases}$$

The cost of the model $L(h)$ reflects the complexity of the model in terms of the number of parent variables $n = |\mathrm{Pa}(j) \setminus j|$ and number of parameters,

$$L(h) = \underbrace{L_{\mathbb{R}}(\varphi_0) + L_{\mathbb{R}}(\varphi_1)}_{\text{intercept and autoregression}} + \underbrace{\log \binom{N-1}{n}}_{\text{parent designation}} +$$

$$\sum_{i}^{\mathrm{Pa}(j)\setminus j} \left(\underbrace{\sum_{\varphi_{i,m} \in \varphi_j} \sum L_{\mathbb{R}}(\varphi_{i,m})}_{\text{regression parameters}} + \underbrace{\sum_{u}^{Q_j \setminus u} L_{\mathbb{R}}(a_{u,v})}_{\text{HMM transition probabilities}} + \underbrace{\sum_{u}^{Q_j} L_{\mathbb{R}}(\pi_u)}_{\text{HMM initial state probabilities}} \right)$$

where $L_{\mathbb{R}}$ is an encoding for real numbers up to precision p: $L_{\mathbb{R}}(x) = L_{\mathbb{N}}(p) + L_{\mathbb{N}}(\lceil x \cdot 10^p \rceil) + 1$. We set $p = \lceil \frac{1}{2} \log_{10} T \rceil$. We show in App A.1.1 that this score is consistent.

Note that to avoid enumerating and testing all possible causal parent sets (e.g., because of a large number of variables), our causal model fitting procedure can be incorporated into a score-based DAG search algorithm such as GES [3], GLOBE [17], and TOPIC [29], as the edge fitting step.

4.6 Algorithmic Properties

We now discuss the parameter estimation and complexity of our method.

Convergence. Without assumptions on the delay process, each new observation would introduce a new delay parameter. Modeling the parent set as a finite-state Markov process instead yields a fixed-dimensional parameter set for any T, which EM estimates jointly with the regression parameters. For a fixed candidate model, EM does not decrease the likelihood and, under standard regularity conditions, converges to a stationary point. Theorem 3.1 establishes identifiability from $P(X)$, while Theorem A.1 shows that, under its assumptions and global optimization, the MDL score asymptotically selects the true graph. However, the EM stationary point may be local, and neither EM nor heuristic graph search is guaranteed to recover the identifiable true model.

Complexity. The computational complexity of the causal edge fitting procedure depends on T, τ_{max}, and $P = |\mathrm{Pa}(j)|$. There are $|Q| = 1 + \sum_{\tau=0}^{\tau_{max}} \sum_{\tau'=\tau}^{\tau_{max}} 1 = O(\tau_{max}^2)$ states in a Markov chain, representing all valid intervals of observations that preserve temporal ordering. For a single parent, the number of state transitions is upper-bounded by τ_{max}^4—however, note that in practice, the transition matrix is highly sparse because most state transitions are not valid (for the same reasons). The worst-case complexity of each iteration for a single parent is $O(T \cdot \tau_{max}^4)$. For the factorial HMM with P parents, the joint state space size is $|Q|^P$, and the complexity of each EM iteration becomes $O(T \cdot \tau_{max}^{2P})$. In practice, we can fix the parameters of previously fitted parents and only update the new parameters. This strategy reduces the joint complexity to $O(P \cdot T \cdot \tau_{max}^4)$, allowing the algorithm to scale linearly rather than exponentially with the number of parents (details in App A.2).

5 Related Work

Discovering causal relationships from observational data is a fundamental problem in science, and has therefore received a lot of attention. In the context of time series, causality is often associated with Granger causality [5] where a variable is said to Granger-cause another if its *past* helps to predict the future of the other. Unlike Pearl's causality, where the focus is on the notion of *intervention*, Granger causality is purely predictive, and cannot be used for causal and counterfactual reasoning.

The most used methods to discover causal relationships on stationary time series include the PCMCI+ algorithm [23], whose applicability to non-linear relationships depends on the chosen conditional independence test, VARLiNGAM [11], which is based on the linear non-Gaussian model [26] and has therefore strong assumptions, and DyNoTears [18], a differentiable score-based approach for linear relationships which does not come with causal guarantees.

Different kinds of nonstationarity have been studied in the causality literature. Some works consider regime-based nonstationarity where the changes are mechanism shifts only (SpaceTime [15]), or structural changes in the causal graph (RPCMCI [25], CASTOR [22]). Drifting changes have been modeled as time-varying causal strength [8, 9, 24] or mechanisms (CD-NOD [10]).

To the best of our knowledge, varying causal delays have only been studied in the predictive scope of Granger causality. Wang et al. [28] proposed to train a long short-term memory (LSTM) model with an input gate that controls which part of the past time series is used for the prediction. Closer to our work, VLTimeCausality [1] uses Dynamic Time Warping (DTW) to align a child and a parent timestamp, leading to a varying causal delay.

6 Experiments

In this section, we empirically evaluate STRETCH to check that the theory works in practice. We first provide an evaluation of STRETCH against the state-of-the-art algorithms on synthetic data. We

then present a case study on climate data modeling El Niño effects on Indian precipitation during monsoon. STRETCH's implementation and experiment code are available online[1].

6.1 Synthetic Data

Method. To evaluate STRETCH's ability to recover the causal graphs underlying the data, as well as the causal delay function used for the generation, we use synthetic data. We compare it to methods for nonstationary time series: VLTimeCausality [1], SpaceTime [15], CD-NOD [10], RPCMCI [25], CASTOR [22], and Huang et al. [8]; as well as methods assuming stationarity: PCMCI+ [23], DyNoTears [18], and VARLiNGAM [11].

We generate time series (T=1000) from random (Erdős-Rényi) DAGs with 2 or 5 variables, linear or non-linear functions (polynomial of degree 3, sinusoid, quadratic function, or (1, 10, 1) ReLU neural network), and a random walk or a discretized sinusoid as delay function (with $\tau \in [0, 6]$). We generate 50 cases for each combination of DAG size and function type.

We measure how structurally different a discovered causal graph is from the true one using the F1 score of the causal edge recovery. To assess the downstream impact of causal intervention, we use the structural interventional distance (SID) [20]. For the delay function recovery, we compute the root mean square error (RMSE). Due to varying output formats across algorithms, specific evaluation procedures are detailed in App. A.4.

Experiments were run on a Dell R6525 with two AMD EPYC 7773X CPUs and 2 TB of RAM.

Results. We first focus on the bivariate case (Table 1) to allow a comparison with VLTimeCausality, which can only handle bivariate time series. STRETCH correctly finds all linear and non-linear

Method	F1		RMSE	
	Linear	Non-linear	Linear	Non-linear
STRETCH	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.32 \pm 0.14}$	$\mathbf{0.52 \pm 0.53}$
VLTimeCausality	$\mathbf{1.00 \pm 0.00}$	0.88 ± 0.33	1.86 ± 0.67	2.03 ± 0.54
SpaceTime	0.96 ± 0.20	$\mathbf{1.00 \pm 0.00}$	1.35 ± 0.46	1.30 ± 0.54
CD-NOD	0.43 ± 0.32	0.24 ± 0.32	—	—
RPCMCI	0.73 ± 0.13	0.69 ± 0.15	1.29 ± 0.30	1.39 ± 0.63
PCMCI+	$\mathbf{1.00 \pm 0.00}$	0.96 ± 0.16	1.18 ± 0.33	1.13 ± 0.38
CASTOR	0.64 ± 0.47	0.55 ± 0.50	2.37 ± 1.27	1.31 ± 0.53
Huang et al. [8]	$\mathbf{1.00 \pm 0.00}$	0.85 ± 0.17	0.91 ± 0.40	1.34 ± 0.73
DyNoTears	0.59 ± 0.49	0.43 ± 0.50	2.12 ± 1.11	2.26 ± 1.01
VARLiNGAM	0.09 ± 0.26	0.12 ± 0.31	1.75 ± 0.40	1.31 ± 0.35

Table 1: Graph and delay function recovery on 50 bivariate cases (mean $\pm$ std).

causal relationships and estimates a delay function very close to the truth. VLTimeCausality—the only other method for varying delays—also gets good graph discovery results but fails at delay recovery. This suggests that vanilla DTW, which allows non-causal future-to-past links and violates physical constraints by permitting one-to-many parent-child alignments, is not the appropriate tool for causal delay recovery. SpaceTime seems to compensate for the varying delay with multiple regimes, maintaining good graph discovery performances, but is ultimately unable to recover the delay function. Surprisingly, PCMCI+ (stationarity assumption) outperforms RPCMCI (regime assumption) thanks to its ability to learn multiple delays for a given pair of variables. Finally, VARLiNGAM (linear function and non-Gaussian noise assumptions) and DyNoTears (linearity assumption) perform badly, which is not surprising given the gap between their assumptions and the generating setup.

On multivariate data (Table 2), we observe similar results. Our method with exhaustive search has a high runtime, but this can be reduced by using a DAG search algorithm such as GLOBE [17] as a wrapper for the edge fitting procedure (STRETCH-GLOBE) instead of enumerating all possible edges, at a small performance cost. It remains significantly lower than SpaceTime's—the second best at graph search—which compensates with many regimes and therefore takes time to converge.

[1] `http://eda.rg.cispa.io/stretch/`

Method	F1	SID	RMSE	Runtime (s)
STRETCH	$\mathbf{0.92 \pm 0.12}$	$\mathbf{0.73 \pm 1.41}$	$\mathbf{0.47 \pm 0.19}$	2623.00 ± 244.77
STRETCH-GLOBE	0.86 ± 0.14	1.72 ± 1.94	0.49 ± 0.24	213.68 ± 33.73
STRETCH-TOPIC	0.61 ± 0.20	4.00 ± 2.39	0.60 ± 0.46	216.74 ± 33.31
SpaceTime	0.80 ± 0.13	1.90 ± 2.24	1.32 ± 0.30	20090.03 ± 13600.09
CD-NOD	0.36 ± 0.16	13.41 ± 5.00	—	$\mathbf{1.39 \pm 0.02}$
RPCMCI	0.43 ± 0.12	12.02 ± 4.96	1.34 ± 0.26	1174.99 ± 143.90
PCMCI+	0.85 ± 0.13	1.85 ± 2.49	1.20 ± 0.27	58.76 ± 23.99
CASTOR	0.55 ± 0.25	5.63 ± 3.82	1.57 ± 0.77	5.09 ± 3.05
Huang et al. [8]	0.75 ± 0.20	5.10 ± 5.98	1.24 ± 0.38	151.04 ± 69.67
DyNoTears	0.29 ± 0.23	9.37 ± 5.16	1.65 ± 0.84	2.08 ± 0.06
VARLiNGAM	0.13 ± 0.16	13.88 ± 4.86	1.73 ± 0.72	1.72 ± 0.13

Table 2: Graph and delay function recovery on 50 linear and 50 non-linear cases with 5 variables.

We provide additional results on data generated with constant delay, non-Gaussian noise, and varying strength, to show that our method performs well also on less ideal setups in App. A.4.

6.2 Case Study

To assess STRETCH's ability to discover insightful causal phenomena on real data, we applied it on climate data. El Niño is a climatic phenomenon characterized by high temperature at sea surface level in the Pacific Ocean, and has an effect on meteorological conditions all around the world, and in particular in India during monsoons. We tested the relationship between *El Niño 3.4r anomaly index* (EN) and *all-India rainfall* (R) using monthly data from 1871 to 2016 downloaded from the KNMI Climate Explorer [13]. As a sanity test, we also included all-India rainfall shifted by three years in the past ($R_{\circlearrowleft}$) to check that we are not solely capturing periodic meteorological patterns.

STRETCH successfully discovers the known link from El Niño (EN) to all-India rainfall (R) (Fig. 3 (Left)). Our sanity check confirms that rainfall patterns from another year help predict monthly

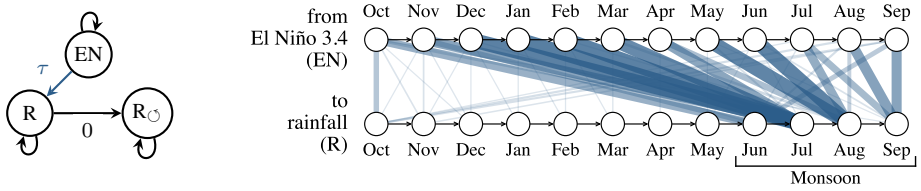


Figure 3: (Left) Learned summary causal graph. The edge labels indicate the delay. (Right) Window causal graph. The edge width and transparency linearly reflect the distribution of targeted months given the source month over the years.

rainfall (link from R to $R_{\circlearrowleft}$). However, the anomaly index only causes precipitation for the current year (R), not any year ($R_{\circlearrowleft}$). Even more interestingly, it causes Indian rainfall with a varying delay (Fig. 3 (Right)); effects originating throughout the year aggregate during the monsoon months from June to September.

As a comparison, PCMCI+ and RPCMCI return causal edges between all variables, suggesting that they see a link between El Niño and All-India rainfall, but fail to disentangle true causation from correlation due to their inability to identify the true parent sets for each month.

7 Conclusion

Causal discovery under time-varying mechanisms is challenging but crucial for counterfactual reasoning and accurate comprehension of phenomena. We proposed a causal model of nonstationary time series with varying causal delay, and proved its identifiability. Building upon this causal model,

we introduced STRETCH, a practical causal discovery method based on the algorithmic model of causality. To learn the causal delay function as well as the functional mechanism, we fit the parameters of an HMM where the states correspond to the parent set at each time step. Experiments on synthetic data showed STRETCH's ability to accurately recover the causal graph and the delay function, and a case study demonstrated its usefulness for real-world processes. In future work, we plan to study how to combine varying delay and varying strength without losing identifiability, thereby increasing the modeling possibilities.

References

[1] Chainarong Amornbunchornvej, Elena Zheleva, and Tanya Berger-Wolf. Variable-lag Granger causality and transfer entropy for time series analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(4):1–30, 2021.

[2] Charles Assaad, Emilie Devijver, and Eric Gaussier. Survey and evaluation of causal discovery methods for time series. *Journal of Artificial Intelligence Research*, 73:767–819, 02 2022. doi: 10.1613/jair.1.13428.

[3] David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3(Nov):507–554, 2002.

[4] Randal Douc, Eric Moulines, and Tobias Rydén. Asymptotic properties of the maximum likelihood estimator in autoregressive models with markov regime. *arXiv preprint arXiv:math/0503681*, 2004.

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

[6] Peter D Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

[7] Patrik Hoyer, Dominik Janzing, Joris Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008. URL `https://proceedings.neurips.cc/paper_files/paper/2008/file/f7664060cc52bc6f3d620bcedc94a4b6-Paper.pdf`.

[8] Biwei Huang, Kun Zhang, and Bernhard Schölkopf. Identification of time-dependent causal model: A Gaussian process treatment. In *IJCAI*, pages 3561–3568, 2015.

[9] Biwei Huang, Kun Zhang, Mingming Gong, and Clark Glymour. Causal discovery and forecasting in nonstationary environments with state-space models. In *International Conference on Machine Learning*, pages 2901–2910. Pmlr, 2019.

[10] Biwei Huang, Kun Zhang, Jiji Zhang, Joseph Ramsey, Ruben Sanchez-Romero, Clark Glymour, and Bernhard Schölkopf. Causal discovery from heterogeneous/nonstationary data. *Journal of Machine Learning Research*, 21(89):1–53, 2020.

[11] Aapo Hyvärinen, Kun Zhang, Shohei Shimizu, and Patrik O Hoyer. Estimation of a structural vector autoregression model using non-Gaussianity. *Journal of Machine Learning Research*, 11 (5), 2010.

[12] Dominik Janzing and Bernhard Schölkopf. Causal inference using the algorithmic Markov condition. *IEEE Transactions on Information Theory*, 56(10):5168–5194, 2010.

[13] Koninklijk Nederlands Meteorologisch Instituut (KNMI). KNMI climate explorer, 2022. URL `https://climexp.knmi.nl/start.cgi`. Accessed on 13 March 2026.

[14] Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 4th edition, 2019.

[15] Sarah Mameche, Lénaïg Cornanguer, Urmi Ninad, and Jilles Vreeken. SPACETIME: Causal discovery from non-stationary time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 19405–19413, 2025.

[16] Alexander Marx and Jilles Vreeken. Formally justifying MDL-based inference of cause and effect. *arXiv preprint arXiv:2105.01902*, 2021.

[17] Osman A Mian, Alexander Marx, and Jilles Vreeken. Discovering fully oriented causal networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8975–8982, 2021.

[18] Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Konstantinos Georgatzis, Paul Beaumont, and Bryon Aragam. DYNOTEARS: Structure learning from time-series data. In *International Conference on Artificial Intelligence and Statistics*, pages 1595–1605. PMLR, 2020.

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

[20] Jonas Peters and Peter Bühlmann. Structural intervention distance for evaluating causal graphs. *Neural Computation*, 27(3):771–799, 2015.

[21] Lawrence Rabiner and Biinghwang Juang. An introduction to hidden Markov models. *IEEE ASSP Magazine*, 3(1):4–16, 2003.

[22] Abdellah Rahmani and Pascal Frossard. Causal temporal regime structure learning. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS) 2025*, volume PMLR 258, pages 4546–4554, 2025.

[23] Jakob Runge. Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series datasets. In *Conference on uncertainty in artificial intelligence*, pages 1388–1397. Pmlr, 2020.

[24] Agathe Sadeghi, Achintya Gopal, and Mohammad Fesanghary. Causal discovery from non-stationary time series. *International Journal of Data Science and Analytics*, 19(1):33–59, 2025.

[25] Elena Saggioro, Jana de Wiljes, Marlene Kretschmer, and Jakob Runge. Reconstructing regime-dependent causal relationships from observational time series. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(11), 2020.

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

[27] Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, 2000.

[28] Yueming Wang, Kang Lin, Yu Qi, Qi Lian, Shaozhe Feng, Zhaohui Wu, and Gang Pan. Estimating brain connectivity with varying-length time lags using a recurrent neural network. *IEEE Transactions on Biomedical Engineering*, 65(9):1953–1963, 2018.

[29] Sascha Xu, Sarah Mameche, and Jilles Vreeken. Information-theoretic causal discovery in topological order. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025.

[30] Shouto Yonekura, Alexandros Beskos, and Sumeetpal S Singh. Asymptotic analysis of model selection criteria for general hidden markov models. *Stochastic Processes and their Applications*, 132:164–191, 2021.

A Appendix

A.1 Causal Model Identifiability

Theorem 3.1 (Delay identifiability)**.** *Consider the multivariate SEM of Eq.* (1)*, let $G^\star = (V, E^\star)$ be the full causal graph, and assume Markov, faithfulness, and causal sufficiency. Then the graph $G^\star$ is identifiable from the observational distribution $P(X)$, and so are all delay functions $\{g_{i\to j} : i \neq j\}$.*

Proof. By assumption, the observational distribution $P(X)$ is Markov and faithful with respect to the induced full causal graph $G^\star$, so that $P(X)$ identifies the skeleton.

Next, given the skeleton, every nonzero-lag adjacency has a forced orientation. Indeed, if X_s^i is adjacent to X_t^j with $s < t$, then the edge cannot be oriented $X_t^j \to X_s^i$, because this would violate temporal ordering. Thus all positive-lag edges in the full graph are identified.

Next, let $X_t^i - X_t^j$ be a simultaneous adjacency. There are multiple cases to consider. First, consider the case that the (unknown) summary graph contains no two-cycle over these two variables. Then either, there exist times $t < s$ such that $X_t^i \to X_s^j \in G^\star$, but no times $s' < t'$ such that $X_{s'}^j \to X_{t'}^i$, so we know that only $X^i \to X^j$ is in the summary graph, and therefore the correct edge is $X_t^i \to X_t^j$. The same applies analogously with i, j reversed. Or there exist no such times in either direction, and without loss of generality we assume that all edges $X^i \to X^j$ are simultaneous edges $X_t^i \to X_t^j$. But in this case, $X_t^i \to X_t^j \leftarrow X_{t-1}^j$ forms an unshielded collider, and is therefore identifiable. In contrast, the opposite edge would imply an unshielded collider $X_t^j \to X_t^i \leftarrow X_{t-1}^i$ instead.

Next consider the case where both $X^i \to X^j$ and $X^j \to X^i$ belong to the summary graph. Then by acyclicity and the fact that each X_t^i has precisely one child among X_s^j and vice versa, $X_t^i \to X_t^j$ if and only if there exists an outgoing edge $X_t^j \to X_{t'}^i$ with $t' > t$, and vice versa. Since these nonzero-lag adjacencies are already known, we can determine the direction of the simultaneous adjacency in this case too.

Finally, once $G^\star$ is identified, the delay functions are read off directly. For every ordered summary edge $i \to j$ and every source time s, the model contains the unique edge $X_s^i \to X_{s+g_{i\to j}(s)}^j$. Since the target time of this edge is identified from $G^\star$, we obtain

$$g_{i\to j}(s) = t - s \qquad \text{whenever } X_s^i \to X_t^j \in E^\star.$$

□

Corollary 3.2 (Varying strength and varying delay are not equivalent)**.** *If g is not constant, then no varying-strength model with fixed delays that is Markov and faithful to its own graph G can be observationally equivalent to our model in Eq.* (1)*.*

Proof. Assume for contradiction that a varying-strength model with fixed lags is observationally equivalent to our time-varying delay model. We distinguish two cases for the alternative model:

Case 1: Single Fixed Lag. Let the alternative SEM assume a single fixed lag τ,

$$Y_t = aY_{t-1} + \beta(t)X_{t-\tau} + \varepsilon_t, \qquad \beta(t) \text{ freely varying}\,.$$

This implies a graph G where the edge set is fixed for all t; specifically, the parent of Y_t in X is always $X_{t-\tau}$. However, our model in Eq. (1) defines the parent of Y_t as $X_{t-g(t)}$. If the delay function g is non-constant, there exist at least two time steps t_1 and t_2 such that the true lags are distinct: $g(t_1) = \tau_1$ and $g(t_2) = \tau_2$ with $\tau_1 \neq \tau_2$. For the alternative model to be observationally equivalent, the single fixed lag τ would need to satisfy $\tau = \tau_1$ and $\tau = \tau_2$ simultaneously. Since τ is constant by definition, this is impossible. Thus, a single fixed-lag model cannot induce the same conditional dependencies as a model with non-constant $g(t)$.

Case 2: Distributed Lag Model. Let the alternative SEM be of the form

$$Y_t = aY_{t-1} + \sum_{\tau=0}^{\tau_{\max}} \beta_\tau(t) f(X_{t-\tau}) + \varepsilon_t\,, \qquad \beta_\tau(t) \text{ freely varying}\,.$$

If the alternative model includes multiple fixed lags $\tau \in \{0, \ldots, \tau_{\max}\}$, it could theoretically emulate our model by setting $\beta_\tau(t) = 0$ whenever $g(t) \neq \tau$. However, such a model violates Causal Faithfulness, which requires that every edge in the graph skeleton $(X_{t-\tau}, Y_t)$ corresponds to a conditional dependence that does not vanish unless the edge is absent. Because the varying-delay process satisfies $Y_t \perp\!\!\!\perp X_{t-\tau} \mid X_{t-g(t)}$ for all $\tau \neq g(t)$, a fixed-lag model would require its parameters $\beta_\tau(t)$ to be exactly zero at specific times t despite the presence of the edge in the skeleton. This contradicts the assumption of faithfulness to G.

Therefore, our model is the unique minimal and faithful representation of the causal process. □

A.1.1 Score Consistency

Theorem A.1 (Score Consistency)**.** *Let $\mathcal{H}$ be a finite class of candidate summary graphs, and let $G^\star \in \mathcal{H}$ be the true graph.*

Assume that:

1. *The observed process is generated by the causal model of Eq. (1), with true graph $G^\star$, parameters $\theta^\star$, and nonzero causal contributions.*
2. *The parent-state processes induced by the delay functions follow the finite-state HMM process described in Section 4.*
3. *The scored HMM model is correctly specified, and standard regularity and identifiability conditions for likelihood-based model selection hold. In particular, the normalized optimized marginal likelihood converges to its population value, and any overfitting gain in log-likelihood is $o(\log T)$. [4, 30]*

Define $S_T(G) = \inf_{\theta \in \Theta_G} \{L_T(D \mid G, \theta) + L_T(G, \theta)\}$, where $L_T(D \mid G, \theta) = -\log P_{G,\theta}(X_{1:T})$ is the negative marginal log-likelihood and $L_T(G, \theta)$ is the model-encoding cost.

In particular, assume that $L_T(G, \theta) = \frac{d_G}{2} \log T + O(1)$, where d_G is the number of free parameters in the model associated with G. This can be achieved, for example, by encoding every parameter up to precision of order $T^{-1/2}$.

Then, the score S_T selects $G^\star$ as $T \to \infty$, i.e., $\Pr(\widehat{G}_T = G^\star) \to 1$.

Proof. Let $\theta^\star$ denote the parameters of the true causal model with graph $G^\star$, and let θ denote the parameters of any candidate model with an incorrect graph $G \neq G^\star$ satisfying Eq. (1). By identifiability (Theorem 3.1), a graph that cannot represent the true causal structure also cannot induce the same observational distribution as the true graph. The expected negative log-likelihood is uniquely minimized by the true observational distribution.

Therefore, every candidate graph that cannot represent the true observational distribution has a strictly larger asymptotic data cost: $\ell(G, \theta) > \ell(G^\star, \theta^\star)$.

More specifically, a candidate model missing a true parent cannot represent the true conditional distribution and therefore has a strictly larger asymptotic data cost. A candidate model containing a spurious parent may represent the true conditional distribution by setting that parent's causal contribution to zero, which entails equal data cost but higher free parameter cost.

For an overfitted candidate graph that can represent the true distribution, the regularity assumptions imply that its possible improvement in optimized log-likelihood over the true graph is $o(\log T)$. However, its additional parameters incur the penalty $\frac{d_G - d_{G^\star}}{2} \log T$, which diverges because $d_G > d_{G^\star}$. Therefore, model cost dominates overfitting gain.

Thus, $G^\star$ is the unique minimal graph among those attaining the minimum asymptotic data cost. Graphs that cannot represent the true distribution incur an $O(T)$ excess data cost, whereas observationally equivalent overfitted graphs are rejected by the $O(\log T)$ model cost.

Since $\mathcal{H}$ is finite, with probability tending to one every graph $G \neq G^\star$ has a larger score than $G^\star$. Therefore, $\Pr(\widehat{G}_T = G^\star) \to 1$. □

A.2 EM Optimization for Factorial HMM

A.2.1 Factorial HMM Parameters

In the multivariate parent set case with P parents, the joint hidden state of the factorial HMM is the Cartesian product of the individual parent states $Q = q^1 \times ... \times q^P$. The emission probability B is computed using the regression model in Eq. 2 by concatenating the parent observations indicated by the joint state vector. The transition matrix A decomposes into the product of independent parent transitions $A = \prod A^i$, preserving the structural logic of the univariate EM steps described in Section 4.4.

A.2.2 Sequential Optimization of Factorial HMM

As the joint optimization of multiple parents becomes intractable as the number of parents P grows, we can instead optimize the Markov chain parameters sequentially. This procedure avoids the exponential growth of the joint state space by fitting parents one at a time. We outline the procedure for two parents:

1. Fit the first parent X^1 using standard HMM-EM as described in Sec. 4.4.
2. Fix the expected state and transition probabilities for this parent (A^1 and π^1).
3. Fit a factorial HMM with parents X^1 and X^2.
   (a) During the E-step, compute the joint emission probabilities (B). Using the fixed posterior state distributions of previous parents, perform the forward-backward algorithm to update only the posterior state probabilities for the new parent.
   (b) During the M-step, update the transition matrix A^2 and initial distribution π^2 for the new parent. Update the entire regression coefficient vector φ using weighted least squares to optimize the global fit.

This procedure generalizes to P parents by iteratively adding parent X^k and optimizing A^k, π^k, and the full regression vector φ while keeping $\{A^i, \pi^i\}_{i=1}^{k-1}$ fixed.

While we could also fix the regression parameters of the previously fitted parents to reduce the computational cost even further, updating the full vector φ in each M-step allows the model to re-allocate causal contributions between the new parent and existing parents as the new parent's delay distribution is learned, preventing the greedy bias that would occur if early regression coefficients were fixed.

A.3 Limitations

Causal model. Our identifiability result relies on Markovness, faithfulness, causal sufficiency, independent innovations, bounded discrete delays, one arrival per source observation and causal relation, and preservation of arrival order. It therefore does not cover latent confounding, distributed or crossing effects, or time-varying causal strength; the latter is evaluated only empirically. For a window ending at T, identifiability over all delays up to $\tau_{\max}$ is restricted to source times $u \leq T - \tau_{\max}$, later effects are right-censored.

STRETCH algorithm. The score-consistency result additionally assumes correct model specification and global optimization. In practice, EM may converge to a local stationary point, and the graph search is not guaranteed to recover the identifiable true model. Sequential multi-parent fitting and heuristic graph searches improve tractability but may introduce approximation or ordering effects. Each parent introduces a delay-state process, so their joint state space grows exponentially with the in-degree, while increasing the total number of variables enlarges the graph-search space. STRETCH is therefore most suitable for sparse, low-in-degree systems. For higher-dimensional data, a scalable method such as PCMCI could first screen the summary graph before STRETCH refines the selected relations and estimates their time-varying delays. Finally, STRETCH uses a Gaussian emission model.

Case study. The El Niño experiment is an illustrative case study without causal ground truth and should not be interpreted as a contribution to climate science.

A.4 Experiments

Experiments were run on a Dell R6525 2x AMD Epyc 7773x with 2 CPUs and 2TB RAM available.

A.4.1 Synthetic Data Generation

For each setting, we sample a directed Erdős–Rényi graph with edge probability 0.3, conditioned on being acyclic and containing at least one edge. For every edge $X^i \to X^j$, the contribution emitted at source time u arrives at $t = \lceil u + g_{i\to j}(u) \rceil$, and the data are generated as

$$X_t^j = b_j + a_j X_{t-1}^j + \sum_{i \in \mathrm{Pa}(j)} \sum_{u:\lceil u+g_{i\to j}(u)\rceil = t} s_{i\to j}(u) f_{i\to j}(X_u^i) + \varepsilon_{j,t} \ .$$

We draw $a_j \sim \mathcal{U}(0.2, 0.5)$, $b_j \sim \mathcal{U}(0.1, 0.2)$, $\varepsilon_{j,t} \sim \mathcal{N}(0,1)$, and use constant edge strengths sampled from $\mathcal{U}(0.5, 3)$ in the main experiments. Linear mechanisms are affine, while nonlinear mechanisms are sampled uniformly from sinusoidal, cubic-polynomial, quadratic, and $(1, 10, 1)$ ReLU-network functions with randomly sampled parameters. We use either a sinusoidal delay or a bounded random walk:

$$g(u) = \lceil |m| + 1 + m \sin(au) \rceil \ , \qquad |m| \sim \mathcal{U}(0.5, 2), \qquad a \sim \mathcal{U}(0.005, 0.05) \ ,$$
$$g(u) = \mathrm{clip}\big(g(u-1) + \delta_u, \ell, h\big), \quad \Pr(\delta_u = \pm 1) = p/2, \qquad \Pr(\delta_u = 0) = 1 - p \ .$$

For the random walk, $p \sim \mathcal{U}(0.001, 0.05)$, $g(0) = \lfloor \mathcal{U}(1.5, 2.5) \rfloor$, $\ell = \lfloor \mathcal{U}(0, 3) \rfloor$, and $h = \lfloor \mathcal{U}(4, 6) \rfloor$. Both processes satisfy temporal-order preservation and have delays at most $\tau_{\max} = 6$. We discard 50 burn-in observations and min–max scale each stored series. The main experiments use $T = 1000$, $N \in \{2, 5\}$, and 50 graphs per combination of N and linearity. In the additional varying-strength experiments, $s_{i\to j}(u) = 1 + 0.5 \sin(4\pi u/T + \phi_{i\to j})$ with independently sampled $\phi_{i\to j} \sim \mathcal{U}(0, 2\pi)$; the strength-only condition fixes $g(u) = 2$, while the joint condition uses the delay processes above.

A.4.2 Method Settings

The maximum delay $\tau_{\max}$ is a user-specified hyperparameter that can be selected using domain knowledge or a preliminary lag-range analysis; all methods use $\tau_{\max} = 6$ unless stated otherwise, and unlisted arguments retain the defaults of the version-pinned implementations released with our experiment code.

- **STRETCH:** fifth-degree polynomial regression; exact-zero regression coefficients and admissible HMM transition probabilities are omitted from the encoding sums and cost no bits, while smaller nonzero values generally receive shorter codes; at most 250 EM iterations with tolerance $T \cdot 10^{-6}$; previously fitted delay-chain parameters fixed when adding a parent, while regression parameters are updated jointly; exhaustive enumeration, GLOBE, or TOPIC graph search as indicated by the method name.
- **PCMCI+:** KCI with Gaussian/RBF kernels for all variables, median-distance bandwidth, gamma approximation, $\alpha = 0.05$, and $\tau \in \{0, \ldots, 6\}$.
- **RPCMCI:** analytic ParCorr test, seed 42, seven regimes, $\tau \in \{0, \ldots, 6\}$, switch threshold 0.05, 20 optimization iterations, at most 10 annealings, PC level 0.2, and test level 0.01; the maximum number of transitions starts at $T - 1$ and is reduced by 10 until an annealing converges.
- **CD-NOD:** the sample index $0, \ldots, T-1$ is the domain variable; Fisher's z test with $\alpha = 0.05$, stable skeleton search, unshielded-collider rule 0, and priority 2.
- **DyNoTears:** lag order 6, $\lambda_W = \lambda_A = 0.1$, 100 iterations, acyclicity tolerance 10^{-8}, and fitting threshold 0.
- **VARLiNGAM:** lag order 6, BIC order criterion, and pruning enabled.
- **SpaceTime:** maximum lag 6, minimum regime duration 15, Gaussian-process score, greedy tree graph search, no-parent changepoint initialization, at most 10 graph/partition interleavings, and changepoint convergence tolerance 3.
- **VLTimeCausality:** bivariate only, cross-correlation lag window 6, and dynamic time warping with a Sakoe–Chiba window whose radius is the absolute cross-correlation lag.
- **CASTOR:** standardized inputs, maximum lag 6, and seed 0; for linear data, five outer iterations, initial window 150, loss threshold 1, edge threshold 0.4, and minimum regime mass 100; for nonlinear data, six outer iterations, initial window 200, loss threshold 0.7, edge threshold 0.3, minimum regime mass 150, and a $(10, 1)$ hidden architecture.

- **Huang et al. [8]:** delayed linear model with lags 1–6, squared-exponential GP kernel initialized with length scale and signal scale 1.1, Gaussian-noise standard deviation 0.1, and 1000 optimization steps; the optional residual-independence diagnostic is not used because it does not affect the fitted coefficients.

A.4.3 Evaluation Protocol

Due to the heterogeneous nature of the algorithms' output, we had to adapt the evaluation protocol to each algorithm, while trying to be as favorable to them as possible.

- For all methods, graph F1 is computed on the directed summary graph after removing self-links.
- CD-NOD learns a partially oriented summary graph and no delays; an undirected edge is counted in both directions and no RMSE is reported.
- VARLiNGAM, DyNoTears, and Huang et al. [8] return coefficient magnitudes without a graph threshold; we evaluate ten equally spaced thresholds over their observed magnitude range and report the one maximizing F1, so their graph results are oracle upper bounds.
- STRETCH's delay trajectory is recovered from the Viterbi-decoded HMM state sequence.
- For VARLiNGAM, DyNoTears, Huang et al., and CASTOR, the estimated delay is the lag with the strongest absolute coefficient; for CASTOR this is selected separately at each time from the active regime.
- PCMCI+ may return several lags for one summary edge; we use their rounded mean as a constant delay.
- RPCMCI's summary graph is the union of its regime graphs; within each active regime, we use the rounded mean of the returned lags for each edge.
- CASTOR's summary graph is the union over all regimes and lags, whereas SpaceTime directly learns one summary graph shared across regimes and we use its largest returned lag for each edge.
- VLTimeCausality directly returns a DTW delay trajectory; negative delays are treated as invalid and omitted from RMSE.
- Delay RMSE is computed only for correctly recovered summary edges and valid predicted time points, first over time for each edge and then averaged across edges.

A.4.4 Additional Results

We first present in Table 3 the results from Table 2 split into linear vs. non-linear cases. SpaceTime

Method	F1		SID		RMSE		Runtime (s)
	Linear	Non-linear	Linear	Non-linear	Linear	Non-linear	
STRETCH	$\mathbf{0.97 \pm 0.08}$	$\mathbf{0.87 \pm 0.13}$	$\mathbf{0.36 \pm 0.96}$	$\mathbf{1.10 \pm 1.68}$	$\mathbf{0.40 \pm 0.09}$	$\mathbf{0.55 \pm 0.23}$	2623.00 ± 244.77
STRETCH-GLOBE	0.90 ± 0.12	0.82 ± 0.15	1.38 ± 1.84	2.06 ± 1.99	0.40 ± 0.11	0.58 ± 0.29	213.68 ± 33.73
STRETCH-TOPIC	0.64 ± 0.19	0.57 ± 0.22	3.52 ± 2.31	4.48 ± 2.39	0.45 ± 0.26	0.75 ± 0.56	216.74 ± 33.31
SpaceTime	0.79 ± 0.13	0.81 ± 0.12	1.50 ± 2.12	2.26 ± 2.31	1.34 ± 0.25	1.31 ± 0.33	20090.03 ± 13600.09
CD-NOD	0.39 ± 0.13	0.34 ± 0.18	13.80 ± 4.76	13.02 ± 5.25	—	—	$\mathbf{1.39 \pm 0.02}$
RPCMCI	0.47 ± 0.12	0.39 ± 0.10	10.68 ± 4.40	13.42 ± 5.16	1.23 ± 0.21	1.46 ± 0.26	1174.99 ± 143.90
PCMCI+	0.84 ± 0.11	0.85 ± 0.14	1.30 ± 1.94	2.40 ± 2.85	1.17 ± 0.28	1.23 ± 0.27	58.76 ± 23.99
CASTOR	0.45 ± 0.24	0.64 ± 0.24	6.66 ± 3.99	4.60 ± 3.38	1.86 ± 0.92	1.30 ± 0.45	5.09 ± 3.05
Huang et al. [8]	0.91 ± 0.10	0.60 ± 0.16	1.88 ± 2.90	8.32 ± 6.54	1.07 ± 0.22	1.40 ± 0.43	151.04 ± 69.67
DyNoTears	0.42 ± 0.19	0.16 ± 0.20	7.70 ± 4.62	11.04 ± 5.17	1.58 ± 0.83	1.78 ± 0.86	2.08 ± 0.06
VARLiNGAM	0.17 ± 0.16	0.09 ± 0.15	14.80 ± 4.79	12.96 ± 4.80	1.74 ± 0.69	1.72 ± 0.79	1.72 ± 0.13

Table 3: DAG and delay function recovery on 50 cases with 5 variables over 1000 observations.

assumes a summary graph shared across regimes, matching our generating process and helping its graph recovery. In our runs, CASTOR often merged the data into too few regimes, whereas RPCMCI returned many regime-specific graphs whose errors accumulated when aggregated into a summary graph.

Varying causal strength. We evaluate 20 repetitions for each combination of $N \in \{2, 5\}$ and linear or nonlinear mechanisms, yielding 80 cases for each condition in Table 4. VLTimeCausality is omitted because it is bivariate only, and SpaceTime because the five-variable runs did not finish within the available compute budget. The results show that STRETCH remains competitive in graph

Method	F1	SHD	RMSE	Runtime (s)
Varying strength				
STRETCH	0.929 ± 0.155	0.475 ± 0.871	0.146 ± 0.164	443.22 ± 451.14
PCMCI+	0.927 ± 0.105	0.825 ± 1.271	0.254 ± 0.397	57.73 ± 56.62
Huang et al. [8]	0.832 ± 0.193	1.725 ± 2.686	0.144 ± 0.494	88.69 ± 60.50
CASTOR	0.802 ± 0.289	1.000 ± 1.243	0.119 ± 0.421	5.55 ± 2.44
RPCMCI	0.601 ± 0.192	5.700 ± 5.276	0.586 ± 0.462	216.51 ± 687.34
CD-NOD	0.382 ± 0.269	3.850 ± 3.234	—	1.15 ± 0.17
Varying strength and delay				
STRETCH	0.882 ± 0.189	0.800 ± 1.174	0.497 ± 0.211	484.36 ± 488.84
PCMCI+	0.912 ± 0.130	0.850 ± 1.244	1.001 ± 0.277	56.39 ± 52.19
Huang et al. [8]	0.827 ± 0.215	2.175 ± 4.009	1.102 ± 0.474	83.10 ± 58.44
CASTOR	0.530 ± 0.394	1.700 ± 1.626	1.579 ± 0.859	5.56 ± 2.45
RPCMCI	0.566 ± 0.177	6.400 ± 5.692	1.248 ± 0.342	114.65 ± 24.70
CD-NOD	0.398 ± 0.262	3.825 ± 3.302	—	1.13 ± 0.01

Table 4: Graph and delay recovery and runtime when causal strength varies, with either fixed or varying delays (mean $\pm$ std).

recovery and does not generally explain varying strength through strongly varying delays. CASTOR and Huang et al. obtain slightly lower conditional delay RMSE when only strength varies, whereas STRETCH has the lowest RMSE when strength and delay vary jointly.

Scalability. We run STRETCH with exhaustive graph search on 20 cases per mechanism class and configuration, varying one factor at a time (L: linear; NL: nonlinear).

T	Type	F1	SHD	RMSE	Runtime (s)
500	NL	0.797 ± 0.361	0.500 ± 0.761	1.109 ± 0.371	15.7 ± 4.1
500	L	0.867 ± 0.251	0.400 ± 0.754	1.246 ± 0.739	14.0 ± 2.1
1000	NL	0.963 ± 0.093	0.150 ± 0.366	1.360 ± 0.605	34.1 ± 8.2
1000	L	1.000	0.000	1.077 ± 0.698	27.5 ± 3.5
1500	NL	0.955 ± 0.123	0.200 ± 0.523	1.346 ± 0.797	48.0 ± 12.5
1500	L	1.000	0.000	1.134 ± 0.811	39.9 ± 8.5
2000	NL	0.970 ± 0.073	0.150 ± 0.366	1.425 ± 0.866	66.4 ± 15.1
2000	L	1.000	0.000	1.195 ± 0.869	53.5 ± 13.2
5000	NL	0.953 ± 0.099	0.200 ± 0.410	1.454 ± 0.807	194.9 ± 48.3
5000	L	1.000	0.000	1.205 ± 0.927	175.5 ± 35.5
10000	NL	0.953 ± 0.099	0.200 ± 0.410	1.447 ± 0.863	303.3 ± 75.4
10000	L	0.983 ± 0.075	0.050 ± 0.224	1.035 ± 0.760	344.8 ± 153.1

Table 5: Effect of sample length for $N = 3$ and $\tau_{\max} = 6$ (mean $\pm$ std).

N	Type	F1	SHD	RMSE	Runtime (s)
2	NL	0.950 ± 0.224	0.050 ± 0.224	1.191 ± 0.627	4.5 ± 1.6
2	L	1.000	0.000	1.234 ± 0.789	5.3 ± 1.6
3	NL	0.963 ± 0.093	0.150 ± 0.366	1.360 ± 0.605	34.1 ± 8.2
3	L	1.000	0.000	1.077 ± 0.698	27.5 ± 3.5
4	NL	0.875 ± 0.182	0.750 ± 1.020	1.309 ± 0.638	224.0 ± 44.1
4	L	1.000	0.000	1.357 ± 0.518	206.6 ± 20.5
5	NL	0.830 ± 0.170	1.500 ± 1.504	1.387 ± 0.493	1040.1 ± 353.7
5	L	0.922 ± 0.234	0.500 ± 1.235	1.328 ± 0.396	1043.4 ± 210.2

Table 6: Effect of graph size for $T = 1000$ and $\tau_{\max} = 6$ (mean $\pm$ std).

True bound	Search bound	Type	F1	SHD	RMSE	Runtime (s)
True delay range and search bound varied together						
3	3	NL	0.950 ± 0.224	0.050 ± 0.224	0.696 ± 0.433	4.8 ± 1.1
3	3	L	1.000	0.000	0.719 ± 0.502	4.8 ± 2.1
6	6	NL	0.950 ± 0.224	0.050 ± 0.224	1.191 ± 0.627	4.5 ± 1.6
6	6	L	1.000	0.000	1.234 ± 0.789	5.3 ± 1.6
9	9	NL	0.850 ± 0.366	0.150 ± 0.366	2.009 ± 1.221	7.1 ± 2.1
9	9	L	1.000	0.000	1.455 ± 0.684	7.9 ± 2.9
Same datasets, only the search bound varies						
3	3	NL	0.950 ± 0.224	0.050 ± 0.224	0.696 ± 0.433	6.1 ± 2.2
3	3	L	1.000	0.000	0.719 ± 0.502	5.1 ± 2.2
3	6	NL	0.950 ± 0.224	0.050 ± 0.224	0.744 ± 0.402	5.5 ± 2.1
3	6	L	1.000	0.000	0.772 ± 0.463	5.4 ± 2.2
3	9	NL	0.950 ± 0.224	0.050 ± 0.224	0.772 ± 0.384	6.6 ± 2.6
3	9	L	1.000	0.000	0.801 ± 0.440	7.3 ± 3.2

Table 7: Effect of the true delay bound and of overspecifying the search bound for $N = 2$ and $T = 1000$ (mean $\pm$ std).

F1 improves between $T = 500$ and $T = 1000$ and then remains stable, while runtime grows approximately linearly with T. Increasing N moderately lowers accuracy and sharply increases exhaustive-search runtime. Increasing the true delay range raises absolute RMSE, whereas increasing only the search bound from 3 to 9 leaves graph recovery unchanged and increases RMSE by less than 0.1.

In addition to the results presented in the paper, we evaluated our method's performance on data not generated under our variable-delay and Gaussian-noise assumptions.

The time series (T=1000) were generated from random (Erdős-Rényi) DAGs with 2 variables, linear or non-linear functions (polynomial of degree 3, sinusoid, quadratic function, or (1, 10, 1) ReLU neural network), **constant delay ($\tau = 0$ or $\tau = 2$), and Gaussian or Student noise (with $df = 3$)**. We generated 20 cases for each combination.

We show the results for data generated with Gaussian noise in Table 8 for the linear cases and Table 9 for non-linear cases, and with Student noise in Table 10 for the linear cases and Table 11 for non-linear cases. STRETCH's performance is not affected by a constant delay, which is consistent with our

Method	**F1**		**RMSE**	
	$\tau = 0$	$\tau = 2$	$\tau = 0$	$\tau = 2$
STRETCH	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.03 ± 0.05
VLTimeCausality	0.70 ± 0.47	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	1.11 ± 0.41
SpaceTime	0.05 ± 0.22	$\mathbf{1.00 \pm 0.00}$	$0.00 \pm$ —	$\mathbf{0.00 \pm 0.00}$
CD-NOD	0.67 ± 0.00	0.53 ± 0.27	—	—
PCMCI+	0.70 ± 0.47	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.05 ± 0.22
CASTOR	0.13 ± 0.27	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.40 ± 0.82
Huang et al. [8]	0.95 ± 0.12	$\mathbf{1.00 \pm 0.00}$	1.00 ± 0.00	$\mathbf{0.00 \pm 0.00}$
DyNoTears	0.38 ± 0.45	0.50 ± 0.48	2.33 ± 2.35	1.18 ± 1.83
VARLiNGAM	0.35 ± 0.49	0.00 ± 0.00	$\mathbf{0.00 \pm 0.00}$	—

Table 8: DAG and delay function recovery on bivariate linear cases with Gaussian noise and fixed delay τ (mean $\pm$ std).

Method	**F1**		**RMSE**	
	$\tau = 0$	$\tau = 2$	$\tau = 0$	$\tau = 2$
STRETCH	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	0.00 ± 0.01	0.10 ± 0.09
VLTimeCausality	0.65 ± 0.49	0.85 ± 0.37	0.31 ± 0.76	1.41 ± 0.14
SpaceTime	0.90 ± 0.31	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$
CD-NOD	0.53 ± 0.27	0.27 ± 0.34	—	—
PCMCI+	0.70 ± 0.47	$\mathbf{1.00 \pm 0.00}$	0.14 ± 0.36	$\mathbf{0.00 \pm 0.00}$
CASTOR	0.68 ± 0.38	0.65 ± 0.49	0.44 ± 0.51	$\mathbf{0.00 \pm 0.00}$
Huang et al. [8]	0.83 ± 0.17	0.90 ± 0.16	1.90 ± 1.59	0.35 ± 0.99
DyNoTears	0.30 ± 0.47	0.33 ± 0.47	1.00 ± 2.45	1.00 ± 1.53
VARLiNGAM	0.80 ± 0.41	0.07 ± 0.21	0.06 ± 0.25	2.50 ± 0.71

Table 9: DAG and delay function recovery on non-linear bivariate cases with Gaussian noise and fixed delay τ (mean $\pm$ std).

causal model as it can be modelled as a constant delay function, and with our algorithm which can model constant delay as well. The RMSE slightly above 0 shows that very few time steps are assigned to the wrong parent set, suggesting that the HMM model is not used for overfitting. There is a slight loss of performance for Student noise as the likelihood computation assumes Gaussian residuals, but STRETCH remains competitive. STRETCH's performances remain excellent with instantaneous effects only.

Method	F1		RMSE	
	$\tau = 0$	$\tau = 2$	$\tau = 0$	$\tau = 2$
STRETCH	0.95 ± 0.22	$\mathbf{1.00 \pm 0.00}$	0.09 ± 0.09	0.22 ± 0.07
VLTimeCausality	0.70 ± 0.47	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	1.25 ± 0.19
SpaceTime	0.15 ± 0.37	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$
CD-NOD	0.67 ± 0.00	0.43 ± 0.33	—	—
PCMCI+	0.85 ± 0.37	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$
CASTOR	0.23 ± 0.33	$\mathbf{1.00 \pm 0.00}$	0.14 ± 0.38	0.90 ± 1.02
Huang et al. [8]	$\mathbf{0.98 \pm 0.07}$	$\mathbf{1.00 \pm 0.00}$	1.00 ± 0.00	$\mathbf{0.00 \pm 0.00}$
DyNoTears	0.43 ± 0.50	0.45 ± 0.51	2.44 ± 2.83	1.33 ± 2.00
VARLiNGAM	0.07 ± 0.21	0.03 ± 0.15	2.00 ± 1.41	$3.00\pm$ —

Table 10: DAG and delay function recovery on bivariate linear cases with Student noise and fixed delay τ (mean $\pm$ std).

Method	F1		RMSE	
	$\tau = 0$	$\tau = 2$	$\tau = 0$	$\tau = 2$
STRETCH	0.90 ± 0.31	0.95 ± 0.22	0.09 ± 0.17	0.38 ± 0.36
VLTimeCausality	0.70 ± 0.47	0.95 ± 0.22	0.04 ± 0.16	1.42 ± 0.08
SpaceTime	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$
CD-NOD	0.67 ± 0.00	0.17 ± 0.30	—	—
PCMCI+	0.80 ± 0.41	0.93 ± 0.23	0.12 ± 0.34	0.05 ± 0.23
CASTOR	0.88 ± 0.31	0.90 ± 0.31	0.06 ± 0.24	$\mathbf{0.00 \pm 0.00}$
Huang et al. [8]	0.82 ± 0.17	0.92 ± 0.15	1.95 ± 1.47	0.31 ± 0.42
DyNoTears	0.25 ± 0.44	0.30 ± 0.47	2.40 ± 2.88	3.33 ± 1.63
VARLiNGAM	0.97 ± 0.10	0.08 ± 0.26	$\mathbf{0.00 \pm 0.00}$	1.50 ± 0.71

Table 11: DAG and delay function recovery on bivariate non-linear cases with Student noise and fixed delay τ (mean $\pm$ std).

With KCI, PCMCI+ performs well at nonzero fixed delays, but remains less reliable for instantaneous effects. Huang et al. recover the graphs well, but their delayed implementation only models lags 1–6 and therefore cannot recover instantaneous delays. CASTOR is accurate on nonlinear cases and nonzero linear delays, but has poor graph recovery for linear instantaneous effects. For SpaceTime on linear cases and instantaneous delay, we suspect an issue in the implementation (from the original paper's authors), as we could not find an explanation for F1 score close to 0. We explain the poor performance of VARLiNGAM on non-zero lag by the large range of delays we allow ($\tau_{max} = 6$), which dilutes the regression coefficients.

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: We introduce an identifiable causal model for varying delay under explicit assumptions, with proofs in the paper and appendix. We also provide a causal discovery algorithm, study its score consistency and complexity, and release runnable code.

   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: A dedicated Limitations section in the appendix discusses the model assumptions, practical optimization guarantees, scalability, and scope of the case study.

   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: Proofs are in appendix due to space limit and are referred to in the paper main body, along with the assumptions.

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: Section 4 describes our algorithm, and the appendix specifies the data generation, method settings, and evaluation protocol. We provide a link to the Python code to run our method and all experiments, including the baselines.

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: We provide online Python code to generate and preprocess the data, run our algorithm and the baselines, and evaluate the results, with the link included in the paper.

   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: We tried to be as exhaustive as possible regarding algorithms hyperparameters and experimental setup, either in the written appendix or in the paper main body.

   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: We report in the result tables in the standard deviation and state the number of cases in the caption.

   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: We describe the time of execution for most results (when relevant) and the type of compute workers and memory available.

   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: No item of the Code of Ethics applies.

   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: [N/A]

    Justification: The contribution is a data analysis method that does not present direct harmful applications.

    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: Does not apply.

    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: Authors of the public dataset used in the paper are properly cited with the link to the data source.

    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 released implementation is commented and documented.

    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: Does not apply.

    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: Does not apply.

    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: Does not apply.

    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.