Stationarity-Aware Causal Discovery in Time Series via Minimal Separating Sets

Shanyun Gao
Purdue University

Raghavendra Addanki
Adobe Research

Tong Yu
Adobe Research

Ryan A. Rossi
Adobe Research

Qifan Song
Purdue University

Murat Kocaoglu
Johns Hopkins University

Abstract

Discovering causal relationships from observational time series is a fundamental problem with broad applications in climate science, healthcare, and finance. Causal graphs with time-lagged structure capture the effects of underlying mechanisms over time. Under the causal stationarity assumption, these causal mechanisms remain consistent across time. Existing constraint-based methods leverage stationarity for conditional independence testing and reduce the problem to learning the parents of variables at the final time point, which can then be used to reconstruct the stationary graph. However, their separating set search strategy mimics the PC algorithm and does not take advantage of the stationary structure. We observe that the stationary graph structure and autoregressive edges impose many meaningful constraints on the separating sets between variables at different time lags. After characterizing the behavior of such separating sets, we propose a novel causal discovery algorithm that exploits this structure of minimal separating sets. Extensive evaluations on synthetic and real-world datasets demonstrate the robustness and accuracy of our method.

1 INTRODUCTION

Causal discovery from observational time series is a key challenge across domains such as climate, health, and the social sciences. See [Nowack et al., 2020, Ombadi et al., 2020, Shen et al., 2020, Vuković and Thalmann, 2022]. Pearl's structural causal modeling (SCM) framework provides a unified approach to modeling causal systems. Causal relationships are modeled by a directed acyclic graph (DAG), known as the causal graph, with edges denoting cause-and-effect relations between variables. Although often used in the IID setting [Pearl et al., 2000, Spirtes et al., 2001, Koller and Friedman, 2009], with constraint-based methods such as PC [Spirtes et al., 2000], FCI [Spirtes et al., 1995], and their extensions [Spirtes, 2001, Colombo et al., 2012, 2014, Andrews et al., 2020], causal graphs can also model systems evolving over time via so-called time-unfolded graphs. These replicate a single causal mechanism across all time points, as illustrated in Fig. 1. These time-series causal graphs can model time-lagged and autocorrelated causal relations.

A major challenge in causal discovery from time-series data is the temporal dependence between samples, which complicates statistical procedures such as conditional independence (CI) testing which is at the center of observational causal discovery. Researchers have proposed adaptations of existing algorithms in this setting. For example, PCMCI [Runge et al., 2019] addresses this challenge with a two-stage procedure: it first uses PC_1 procedure to find variables that cannot be d-separated from the target by any others, thereby forming a superset of the target variable's parents, and then applies the Momentary Conditional Independence (MCI) test to control false positives in strongly dependent time series, conditioning on supersets of the parents of both variables so the resulting residuals are IID for reliable CI testing.

While IID samples are preferred for causal discovery with CI tests, the Causal Stationarity Assumption offers a powerful alternative by ensuring consistent causal relationships over time. This makes it possible to apply

Proceedings of the 29[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

CI tests even to dependent samples, either directly or with appropriate corrections such as the MCI test. Extending constraint-based methods to time series often relies on this important assumption. Representative examples include PCMCI [Runge et al., 2019], which builds on PC [Spirtes et al., 2000], and tsFCI [Entner and Hoyer, 2010], an extension of FCI [Spirtes et al., 1995]. See Related Work for additional stationarity-based methods. However, these existing algorithms continue to adopt the same or a modified separating set search strategy of the PC algorithm. Although these methods leverage Causal Stationarity and other structural features of time series, such as autoregressive edges within each *univariate component time series* and temporal ordering, they typically use them in a limited way. Specifically, they focus on reconstructing a stationary causal graph by repeatedly enforcing time-invariant relations and orienting edges after estimating dependencies.

In contrast, our approach leverages these structural constraints *during* the discovery process itself, using them to guide and refine the identification of causal relationships. This perspective treats Causal Stationarity not merely as a background assumption but as a substantive source of structural information that prior approaches have overlooked. Specifically, we demonstrate that *certain* separating sets that d-separate a pair of variables encode additional causal relations involving variables beyond the tested pair. Yet existing constraint-based procedures typically use such separating sets only as one-off certificates that the tested variables are not adjacent. This insight motivates the novel introduction of *Lag-Anchored Separating Sets* (LASS) as the foundation of our proposed framework.

We make the following contributions:

1. We introduce the *Lag-Anchored Separating Sets*, a structured class of minimal separating sets informed by structural constraints. We provide the first formal analysis showing how these sets can act as reliable supersets of parent sets. This characterization is not limited to our proposed method and can be generalized to other causal discovery frameworks.
2. We propose a novel two-stage constraint-based causal discovery algorithm for stationary time series that explicitly encodes Lag-1 Self-Autoregressive Causality. Unlike prior methods, it exploits temporal ordering and stationarity not only for edge orientation but also to guide the discovery process itself via the *Lag-Anchored Separating Sets*. This makes our method fundamentally different from existing approaches.
3. We evaluate the proposed algorithm on a variety of synthetic datasets, benchmarks, and a real-world river flow dataset, demonstrating its accuracy and robustness. Overall, the proposed algorithm achieves the highest F_1 score and lowest structural Hamming distance (SHD), with a preference toward higher precision compared to other baselines. The code is available online.[1]

1.1 Related Work

Extensive research has explored causal discovery in multivariate time series using constraint-based, score-based, functional models, and deep learning-based approaches. Among these, the most widely adopted constraint-based methods build on PC or FCI. PCMCI in [Runge et al., 2019] is a representative method assuming causal sufficiency and no instantaneous effects. PCMCI+ in [Runge, 2020] extends this to contemporaneous causality and optimizes conditioning sets through the strongest adjacencies. LPCMCI in [Gerhardus and Runge, 2020] accounts for latent confounding. Further variants address complex settings such as regime changes, multiple datasets, or periodic causality in [Saggioro et al., 2020, Günther et al., 2023, Gao et al., 2023].

Score-based methods include DYNOTEARS in [Pamfil et al., 2020] and NTS-NOTEARS in [Sun et al., 2021], both of which extend the NOTEARS framework [Zheng et al., 2018] by formulating causal discovery as a continuous optimization problem with an acyclicity constraint. DYNOTEARS focuses on linear time-varying models, while NTS-NOTEARS handles nonlinear causal relationships using neural networks. Functional methods like VARLiNGAM in [Hyvärinen et al., 2010] and TiMINo in [Peters et al., 2013] identify causal directions by modeling each variable as a function of its causes, assuming non-Gaussian (linear) or independent (nonlinear) noise to exploit asymmetries in the data. Granger-causality-based methods include TCDF from [Nauta et al., 2019] for regularly sampled time series with convolutional neural networks, while CUTS from [Cheng et al., 2023] and CUTS+ from [Cheng et al., 2024] for irregularly sampled time series.

While both the proposed method and existing constraint-based algorithms rely on CI tests, they differ in how the separating sets are used. Existing methods use CI tests to eliminate variables from candidate parent sets based on d-separation from [Tian et al., 1998], treating the minimal separating set as a tool. In contrast, our approach focuses on the structure and content of the minimal separating set itself, independent of the specific testing variable.

[1] `https://github.com/CausalML-Lab/StationarityAwareDiscovery`.

2 STATIONARITY-AWARE CAUSAL FRAMEWORK: MINIMAL SEPARATING SETS

In this section, we introduce the important definitions, present the problem setup in Section 2.1, and outline the assumptions in Section 2.2. We also provide theoretical guarantees for the correctness of our algorithm in Section 2.3 and highlight its fundamental differences from PCMCI [Runge et al., 2019].

2.1 Preliminaries

Denote $[a] \coloneqq \{1, \cdots, a\}$ and $[a, b] \coloneqq \{a, \cdots, b\}$, where $a, b \in \mathbb{N}^+$. Sets are usually denoted by bold letters, while individual variables are denoted by non-bold letters. A multivariate time series is defined as $\mathbf{V} \coloneqq \{X_t^j\}_{j\in[n],t\in[T]}$, where X_t^j is a single variable from the j-th *univariate component time series* at time point t, and there are n *univariate component time series* of length T. The time slice at time point t is represented by $\mathbf{X}_t \coloneqq \{X_t^j\}_{j\in[n]}$, and $\mathbf{X}^j \coloneqq \{X_t^j\}_{t\in[T]}$ denotes the j-th *univariate component time series*. In summary, the superscript indexes the *univariate component time series*, and the subscript indexes the time point. For simplicity, we use s, u, v, t to denote different time points and X, Y, Z, H to denote different individual variables.

Let $\mathcal{G}(\mathbf{V}, \mathbf{E})$ denote the underlying causal graph with the observational multivariate time series $\mathbf{V}$ and edge set $\mathbf{E}$. For each variable $X \in \mathbf{V}$, the variables with directed edges pointing into X form its parent set, denoted by $\mathbf{PA}(X)$. We denote a single parent as $\mathrm{pa}(X) \in \mathbf{PA}(X)$.

For any two variables $X, Y \in \mathbf{V}$ and set $\mathbf{W} \subset \mathbf{V}$, the conditional independence relation that X is independent of Y given $\mathbf{W}$ is denoted by $X \perp\!\!\!\perp Y \,|\mathbf{W}$. Define $\mathbf{M}_{+1} = \{X_{t+1}^i \mid X_t^i \in \mathbf{M},\ i \in [n]\}$ for any set $\mathbf{M}$.

We begin by recalling two common definitions and introduce one new definition that establishes the basis for the proposed stationarity-aware causal discovery method.

Definition 2.1 (*Stationary* SCM)**.** A Stationary Structural Causal Model (SCM) is a tuple $\mathcal{M} = \langle \mathbf{V}, \mathcal{F}, \mathcal{E}, \mathbb{P} \rangle$ where there exists $\tau_{\max}^j \in \mathbb{N}^+$ for any $j \in [n]$ and a $\tau_{\max} \in \mathbb{N}^+$, defined as:

$$\tau_{\max}^j \coloneqq \max_{\tau} \{\tau : X_{t-\tau}^i \in \mathbf{PA}(X_t^j), i \in [n]\}, \quad (1)$$

$$\tau_{\max} \coloneqq \max_{j\in[n]} \tau_{\max}^j \quad (2)$$

such that with this $\tau_{\max}$, each variable $X_{t>\tau_{\max}}^j \in \mathbf{V}$ is a deterministic function of its parent set $\mathbf{PA}(X_{t>\tau_{\max}}^j) \in \mathbf{V}$ and an unobserved (exogenous) variable $\epsilon_{t>\tau_{\max}}^j \in \mathcal{E}$,

$$X_t^j = f_j(\mathbf{PA}(X_t^j), \epsilon_t^j), \quad j \in [n],\ t \in [\tau_{\max} + 1, T]. \quad (3)$$

where $f_j \in \mathcal{F}$ and $\{\epsilon_t^j\}_{t\in[T]}$ are jointly independent with probability measure $\mathbb{P}$. Thus $\tau_{\max}^j$ is the maximal lag for the jth *univariate component time series* $\mathbf{X}^j$ while $\tau_{\max}$ is the finite maximal lag for the entire causal graph $\mathcal{G}$. Note that f_j remains fixed across all time points t. See Fig. 1 for an illustration of a *Stationary* SCM.

Definition 2.2 (Minimal Separating Set)**.** A minimal separating set $\mathbf{M}$ between variables X and Y is a set that satisfies:

$$X \perp\!\!\!\perp Y \,|\mathbf{M} \quad \text{and} \quad X \not\perp\!\!\!\perp Y \,|\mathbf{M}_{\text{sub}} \quad \forall\, \mathbf{M}_{\text{sub}} \subsetneq \mathbf{M}. \quad (4)$$

Note that there may exist multiple minimal separating sets between X and Y. If X and Y are marginally independent, then the empty set is the *unique* minimal separating set.

Definition 2.3 (Lag-Anchored Separating Set)**.** Let X_s and Y_t be two variables with $s < t$. A minimal separating set $\mathbf{M}$ between X_s and Y_t is called a Lag-Anchored Separating Set if it includes Y_{t-1}, i.e., $Y_{t-1} \in \mathbf{M}$. $\mathbf{M}$ is referred to as a Lag-Anchored Separating Set of the target variable Y_t.

With this definition, we present a toy example based on Fig. 1. Let H_9 be the target variable. Two LASS examples between source variable X_4 and target variable H_9 are: $\text{LASS}_1 = \{H_8, H_7, Z_7, Z_8\}$ and $\text{LASS}_2 = \{H_8, H_7, Z_7, X_6, Y_6\}$.

Note that a *Lag-Anchored Separating Set* is defined with respect to Y_t, the specific choice of variable X_s being tested against Y_t is not essential.

2.2 Assumptions

In addition to commonly assumed conditions in constraint-based causal discovery as adopted in PCMCI [Runge, 2018], such as **A1** Causal Sufficiency, **A2** the Causal Markov Condition, **A3** the Faithfulness Condition [Pearl et al., 2000], **A4** temporal priority [Assaad et al., 2022] (i.e., no causal influence from future to past), **A5** the absence of contemporaneous (instantaneous) causal effects, and **A6** Causal Stationarity introduced in Definition 2.1, we introduce one further core assumption **A7** below. Please refer to Appendix Section B for the formal definitions of assumptions **A1-A6**.

Assumption A7 (Lag-1 Self-Autoregressive Causality)**.** For any variable X_t^j in the *univariate component time series* $\mathbf{X}^j$, its immediate predecessor X_{t-1}^j is a parent of X_t^j, that is, $X_{t-1}^j \in \mathbf{PA}(X_t^j)$.

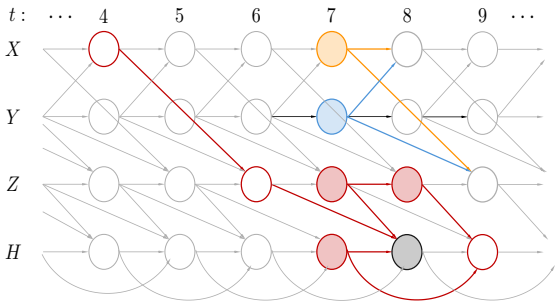


Figure 1: Illustration of a *Stationary* SCM. Assumption A7 (Lag-1 Self-Autoregressive Causality) ensures an edge exists between consecutive variables on the same *univariate component time series*. Please note that multiple self-lags of arbitrary length are allowed. The unique minimal separating set $\{X^7, Y^7\}$ between X_8 and Z_9 is highlighted in orange and blue. The red path between X_4 and H_9 provides insight into the structure of a *Lag-Anchored Separating Set* (Definition 2.3) which includes the gray node H_8 as characterized in Lemma 2.6.

This core assumption can be tested using CI tests before choosing the proposed algorithm over other constraint-based methods.

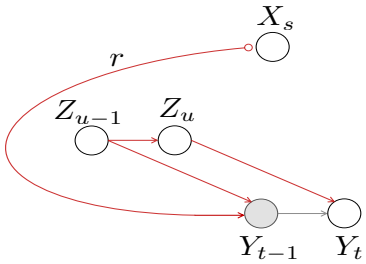
2.3 Theoretical Guarantees

For simplicity, we use X, Y, Z and H to represent variables, with subscripts indicating time points. Unless otherwise specified, these variables without superscripts may belong to the same or different *univariate component time series*. We present important results, along with proof sketches; full proofs are provided in Appendix Section C.

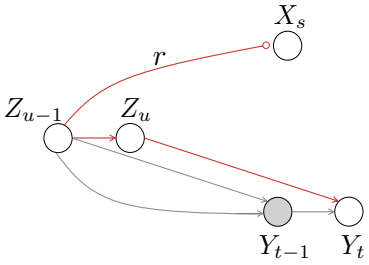
Using Lemma 2.4, we restrict the search for the *Lag-Anchored Separating Set* of Y_t to variables that occur before t, since a *Lag-Anchored Separating Set* is a minimal separating set and it does not contain any variable occurring at or after t.

Lemma 2.4 (Temporal Constraint on Minimal Separating Sets)**.** *Let X_s and Y_t be two variables with $s < t$ and let $\mathbf{M}$ be any minimal separating set (Definition 2.2) between X_s and Y_t. Then $\mathbf{M}$ does not include any variable of the form Z_u with $u \geq t$, under Assumption **A1-A7**.*

Please refer to Appendix Section C (Lemma C.4) for the proof.



(a) Case 1: Z_{u-1} is not on path r



(b) Case 2: Z_{u-1} is on path r

Figure 2: For a *Lag-Anchored Separating Set* $\mathbf{M}$ of Y_t containing Y_{t-1}, if a parent of Y_t, say $Z_u \notin \mathbf{M}$, then, $Z_{u-1} \in \mathbf{M}$. See Lemma 2.6 for additional details.

Lemma 2.5 shows that variables from $\mathbf{X}$ can be excluded from testing against the target Y_t if one of them occurring before time t is marginally independent of Y_t, since it always yields an empty minimal separating set.

Lemma 2.5 (Marginal Independence in Stationary SCM)**.** *Let X_s and Y_t be two variables with $s < t$ and suppose they are marginally independent, i.e., $X_s \perp\!\!\!\perp Y_t$. Then, the following holds for any $u < t$ under Assumption **A1-A7**:*

$$X_u \perp\!\!\!\perp Y_t, \quad \forall X_u \in \mathbf{X} \tag{5}$$

where $\mathbf{X} := \{X_v, v \in [t-1]\}$, denoting the univariate component time series that X_u comes from.

Proof sketch. Due to self-causality, X_s and Y_t cannot belong to the same *univariate component time series*; that is, $Y_t \notin \mathbf{X}$. For the sake of contradiction, we assume $X_u \not\perp\!\!\!\perp Y_t$ for some $u < t$. Therefore, there must be a d-connecting path $\tilde{p}$ between X_u and Y_t when conditioning on the empty set. We proved that for either $u < s$ or $s < u < t$, there would be a d-connecting path p between X_s and Y_t constructed on the basis of $\tilde{p}$ under **A1-A7**. Therefore, if $X_u \not\perp\!\!\!\perp Y_t$ then $X_s \not\perp\!\!\!\perp Y_t$. By contradiction, we conclude that $X_u \perp\!\!\!\perp Y_t$. Please refer to Appendix Section C (Lemma C.7) for the detailed construction of p. □

Lemma 2.6 characterizes the *Lag-Anchored Separating Set* $\mathbf{M}$ of Y_t as either containing a true parent of Y_t or

offering structural hints about this parent through a one-lag backward shift.

Lemma 2.6 (Characterization of Lag-Anchored Separating Set)**.** *Let X_s and Y_t be two variables with $s < t$ and let $\mathbf{M}$ be a Lag-Anchored Separating Set (Definition 2.3) between X_s and Y_t satisfying $Y_{t-1} \in \mathbf{M}$. Then for any $Z_u \in \boldsymbol{PA}(Y_t) \backslash \mathbf{M}$, the following holds under Assumption **A1-A7**:*

$$Z_{u-1} \in \mathbf{M} \tag{6}$$

Proof sketch. Since $Y_{t-1} \in \mathbf{M}$, removing Y_{t-1} from $\mathbf{M}$ will open a set of paths between X_s and Y_t; let us denote this collection of paths as $\mathcal{R}$. Let $v^r := \arg\max\{v \mid H_v \in \mathbf{PA}(Y_{t-1}) \cap r\}$ [2], and H_{v^r} represents the latest parent of Y_{t-1} along the path r. Given Lemma 2.4, any $r \in \mathcal{R}$ does not contain any variable with time point $\geq t$. Therefore r cannot pass through Y_{t-1} one of its children; otherwise, r would already be blocked by an unconditioned collider without removing Y_{t-1} from $\mathbf{M}$. Thus, around Y_{t-1}, the path r must take the form $\cdots H_{v^r} \to Y_{t-1} \to Y_t$ ensuring that Y_{t-1} is always entered via an incoming arrow on r.

For any $Z_u \in \mathbf{PA}(Y_t) \backslash \mathbf{M}$, we proved that $Z_{u-1} \in \mathbf{M}$ given $Y_{t-1} \in \mathbf{M}$ by considering two cases, depending on whether there exists a path $r \in \mathcal{R}$ that does not pass through Z_{u-1}. **Case 1.** There exists a path $r \in \mathcal{R}$ that does *not* include the variable Z_{u-1}. For such a path, we construct the red path shown in Fig. 2(a), where Y_{t-1} acts as a collider. Since $Y_{t-1} \in \mathbf{M}$ and the path r remains open when truncated at Y_{t-1}, we must have $Z_{u-1} \in \mathbf{M}$ to block this red path given that $Z_u \notin \mathbf{M}$. Otherwise, the red path would remain d-connecting given $\mathbf{M}$, contradicting the definition of a minimal separating set. See Fig. 1 for another example, highlighted by the red path. **Case 2.** Every path $r \in \mathcal{R}$ passes through Z_{u-1}. In this case, we construct the red path shown in Fig. 2(b), which is open since r is d-connecting and truncated at Z_{u-1}, where neither Z_{u-1} nor Z_u acts as a collider. Blocking the red path requires that $Z_{u-1} \in \mathbf{M}$ as well, given $Z_u \notin \mathbf{M}$. Please refer to Lemma C.8 for the detailed proof. □

Theorem 2.7 constructs a superset of the parent set via a *Lag-Anchored Separating Set*.

Theorem 2.7 (Construction of a Superset of the Parent Set)**.** *Let X_s and Y_t be two variables with $s < t$ and $\mathbf{M}$ be a minimal separating set between X_s and Y_t. Under Assumption **A1-A7**, we have:*

$$\textit{If } Y_{t-1} \in \mathbf{M}, \quad \textit{then } \boldsymbol{PA}(Y_t) \subset \mathbf{M} \cup \mathbf{M}_{+1}. \tag{7}$$

where $\mathbf{M}_{+1} \coloneqq \{X_{u+1} : X_u \in \mathbf{M}\}$.

[2]Here r is used as a set and denotes the set of variables along the path r. We allow this abuse of notation for simplicity.

Proof sketch. Based on Lemma 2.6, given $Y_{t-1} \in \mathbf{M}$, we have $Z_{u-1} \in \mathbf{M}$ for any $Z_u \in \mathbf{PA}(Y_t) \backslash \mathbf{M}$. Therefore for any $H_v \in \mathbf{PA}(Y_t)$, we either have $H_v \in \mathbf{M}$ or $H_{v-1} \in \mathbf{M}$, and when $H_{v-1} \in \mathbf{M}$, we have $H_v \in \mathbf{M}_{+1}$. Please refer to Theorem C.12 for the detailed proof. □

Putting Things Together. For Theorem 2.7 to be applicable, we must ensure the existence of a *Lag-Anchored Separating Set* (Definition 2.3) for Y_t that includes Y_{t-1}. We show that such a set always exists provided that the testing variable X_s^j from some *univariate component time series* $\mathbf{X}^j$ satisfies $s < t - \tau_{\max}^j$. In particular, such a variable is guaranteed to exist when $\mathbf{X}^j = \mathbf{Y}$, ensuring the existence of a *Lag-Anchored Separating Set* between Y_s and Y_t for $s < t - \tau_{\max}^j$, which is further shown to be a subset of $\mathbf{PA}(Y_t)$. See Appendix Section C for the proof of existence (Lemmas C.9 and C.10) and correctness (Theorem C.13). Fig. 1 provides an illustrative example: the orange and blue paths between X_8 and Z_9 highlight the necessity of the $\tau_{\max}$ constraint, as the unique minimal separating set $\{X_7, Y_7\}$ between X_8 and Z_9 does not qualify as a *Lag-Anchored Separating Set* for Z_9. In contrast, the pair (X_4, Z_9) admits a *Lag-Anchored Separating Set* for Z_9.

Comparison with PCMCI. The proposed Minimal-Separating-Set-based Causal Discovery (MSSD) adopts the same two-stage structure as PCMCI [Runge et al., 2019]: in stage 1, both methods identify a superset of the parent set. However, the foundations of the two methods differ fundamentally. PCMCI constructs a candidate parent set consisting of variables within a time window of length $\tau_{\max}$, and removes variables that are conditionally independent of the target variable X_t given some separating set. In this framework, variables that cannot be d-separated from a target remain as parents, while those that can be d-separated by some set are removed from the candidate parent set, without exploring the separating set itself. In contrast, MSSD centers on identifying a specific type of minimal separating set, namely, a *Lag-Anchored Separating Set*. In summary, PCMCI centers on identifying variables that cannot be d-separated from the target, whereas our approach centers on identifying the specific separating sets that d-separate some variables from the target.

3 MINIMAL-SEPARATING-SET-BASED CAUSAL DISCOVERY (MSSD)

We provide an overview of the algorithmic framework in this section. Practical considerations of the proposed

Algorithm 1 Minimal-Separating-Set-based Discovery (MSSD)

1: **Input:** $\mathbf{V} = (\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3, ..., \mathbf{X}^n)$, maximum lag upper bound τ^j_{ub}, series length T. τ^j_{ub} is assumed to be larger than $\tau^j_{\max}$.
2: **for all** $\mathbf{X}^j$ where $j \in [n]$ **do**
3: $\mathcal{C}(X^j_t) \leftarrow \emptyset$, *Dep-Time-Index* $\leftarrow \emptyset$, $\mathbf{M}^j \leftarrow \emptyset$, $D^j_{\text{pa}} \leftarrow \emptyset, \forall t \in [T]$
4: **for all** $\mathbf{X}^i$, $i \in [n], i \neq j$ **do** ▷ Stage 1: Discovery of a Superset of the Parent Set.
5: **if** $X^i_{t-1} \not\perp\!\!\!\perp X^j_t \mid \emptyset$ **then**
6: *Dep-Time-Index* $\leftarrow$ *Dep-Time-Index* $\cup \{i\}$
7: Set search space $\mathcal{C}(X^j_t) \leftarrow \{X^i_{t-\tau} : \tau \in [\tau^j_{\text{ub}}], i \in$ *Dep-Time-Index* $\cup \{j\}\}$
8: *found* $\leftarrow$ **False**
9: **for all** τ increasing from 1 to $\tau^j_{\text{ub}}+1$ and X^i_τ, $i \in$ *Dep-Time-Index* $\cup \{j\}$ **do**
10: **for all** $\mathbf{M} \subseteq \mathcal{C}(X^j_t) \setminus X^i_\tau$ with $X^j_{t-1} \in \mathbf{M}$ **do**
11: **if** $X^i_{t-\tau} \perp\!\!\!\perp X^j_t \mid \mathbf{M}$ **then**
12: $\mathbf{M}^j \leftarrow \mathbf{M} \cup \mathbf{M}_{+1}$ ▷ $\mathbf{M}_{+1} \coloneqq \{Z_{u+1} \mid Z_u \in \mathbf{M}\}$.
13: *found* $\leftarrow$ **True**
14: **break** all previous inner loops
15: **if not** *found* **then**
16: $D^j_{\text{pa}} \leftarrow D^j_{\text{pa}} \cup X^i_{t-\tau}$
17: $\widehat{PA}(X^j_t) \leftarrow \mathbf{M}^j \cup D^j_{\text{pa}}$
18: **for all** $X^k_s \in \widehat{PA}(X^j_t)$ **do** ▷ Stage 2: Recovery of the True Parent Set.
19: **if** $X^k_s \perp\!\!\!\perp X^j_t \,\big|\, \widehat{PA}(X^j_t) \setminus X^k_s$ **then**
20: Remove X^k_s from $\widehat{PA}(X^j_t)$
21: **return** $\widehat{PA}(X^j_t) \quad \forall j \in [n], t \in [T]$.

algorithm, along with its computational analysis, are also discussed.

Our proposed causal discovery algorithm (MSSD) (Algorithm 1) aims to identify the parent set of each variable in a multivariate time series $\mathbf{V}$. Under **A6** Causal Stationarity Assumption, the parent set of X^j_t for each $j \in [n]$ remains invariant over $t \in [T]$, allowing recovery at a single time point. MSSD has two stages: in Stage 1, we identify a superset of parents of X^j_t, such that $\widehat{\mathbf{PA}}(X^j_t) \supset \mathbf{PA}(X^j_t)$ by searching for a *Lag-Anchored Separating Set*, i.e., it contains X^j_{t-1}. We prove that any such set is a valid superset of the true parent set (see Theorem 2.7). In Stage 2, MSSD refines this superset to obtain the final parent set estimate. See Appendix Section A for details.

Stage 1: For each target time series $\mathbf{X}^j$, MSSD begins by testing the unconditional dependence between X^j_t and each variable X^i_{t-1} with $i \neq j$. We argue that variables from $\mathbf{X}^i$ can be excluded from the candidate set when searching for a *Lag-Anchored Separating Set* for the target X^j_t if X^i_{t-1} is marginally independent of X^j_t (using Lemma 2.5). Based on this observation, our search is initialized (Line 4–7). MSSD then searches for a *Lag-Anchored Separating Set* by iterating over each lagged variable $X^i_{t-\tau}$ for $\tau = 1, \ldots, \tau_{\text{ub}}+1$ (Line 9). If a valid set is found for some $\tau \in [\tau^j_{\text{ub}}]$, it proceeds to stage 2. If no such set is found, the testing variable $X^j_{t-\tau}$ is added to D^j_{pa} as a potential parent. We prove in Appendix Section C (Lemmas C.9 and C.10) that a *Lag-Anchored Separating Set* for X^j_t must exist when the testing variable is X^i_τ with $\tau = \tau^j_{\text{ub}} + 1$, and that such a set is a subset of $\mathbf{PA}(X^j_t)$. This justifies the design of the search space as specified in Line 7. At Line 17, the resulting set forms a superset of the parent set for X^j_t with $\mathbf{M}^j \cup D^j_{\text{pa}}$.

Stage 2: MSSD prunes $\widehat{\mathbf{PA}}(X^j_t)$ (starting at Line 18) by removing variables that are conditionally independent of X^j_t given the remaining variables in $\widehat{PA}(X^j_t)$.

Practical Considerations. We argue that with a consistent CI test and infinite samples, it suffices to test only the pair $(X^j_{t-\tau^j_{\text{ub}}-1}, X^j_t)$ since a *Lag-Anchored Separating Set* $\mathbf{M}$ between this variable pair must exist (see discussion after Theorem 2.7). However, as shown in Lemma 2.6, the accurate identification of such a set $\mathbf{M}$ depends on specific path collections $\mathcal{R}$. When the information flow along $\mathcal{R}$ is weak, discovering $\mathbf{M}$ correctly can be difficult in finite samples. Therefore, we scan over all time-lagged variables to improve robustness in practice, as shown in Line 9. We apply an early-stop criterion that terminates the scan once an $\mathbf{M}$ is identified. At most n valid $\mathbf{M}$, each obtained from a different *univariate component time series*, are then aggregated across these series. We demonstrate that the algorithm often terminates early by identifying an $\mathbf{M}$ before reaching the lag $\tau^j_{\text{ub}} + 1$, especially when τ^j_{ub} is large. See Fig. 4(b).

Computational Analysis. We examine the computational complexity for two scenarios. (i) Ideal case with consistent CI tests and infinite samples. For each $\mathbf{X}^j$, MSSD only needs to test the pair $\left(X^j_{t-\tau^j_{\text{ub}}-1}, X^j_t\right)$ to identify the *Lag-Anchored Separating Set*. Therefore, MSSD has a time complexity of $\mathcal{O}\big(n^2\tau_{\text{ub}} + n^2\tau_{\text{ub}}\big)$. The two terms correspond to the two stages of the procedure, hence we do not simplify it. (ii) Practical case with finite samples. To improve robustness, we scan variables in a window of length $\tau_{\text{ub}} + 1$ and apply an early-stopping rule. Therefore the proposed algorithm has complexity $\mathcal{O}\big(n^3\tau_{\text{ub}}(\tau_{\text{ub}}+1) + n^2\tau_{\text{ub}}\big)$, which is comparable to PCMCI. See Appendix Section D for the derivation of the computational analysis and the

runtime report. We also report experiments that use a single fixed pair (no scanning) in Appendix Section D.1.

4 EXPERIMENTS

To validate the effectiveness and robustness of our proposed algorithm, we conduct empirical evaluations in various settings. Our experiments focus on the performance of recovering the true causal graphs.

Baselines. The baselines include PCMCI [Runge et al., 2019], VARLiNGAM [Hyvärinen et al., 2010], DYNOTEARS [Pamfil et al., 2020], TCDF [Nauta et al., 2019] and tsFCI [Entner and Hoyer, 2010]. We consider two variants of MSSD. MSSD-PC combines PCMCI's Stage-1 with *Lag-Anchored Separating Set* strategy. MSSD-MCI replaces our second stage with PCMCI's MCI test. See Appendix Section A for a brief overview of PCMCI [Runge et al., 2019] and the construction of these variants.

Experimental Configurations. For all applicable methods, we set the CI-test significance level to 0.05 unless a different default is prescribed, and the maximum time lag τ_{ub} is chosen to match the true $\tau_{\max}$. All other hyperparameters follow the defaults from the original implementations. See Appendix Section E.1 for the full configuration details.

Metrics. We report standard evaluation metrics, including adjacency precision, adjacency recall, adjacency F_1 score, and structural Hamming distance (SHD), to assess the performance of each method.

4.1 Simulations with Synthetic Datasets

Here, we conduct experiments on synthetic datasets with continuous and discrete observations. Details of the data generation process are provided in Appendix Section G.

4.1.1 Linear SCMs with Gaussian Noise

Note that our method includes an additional assumption, **A7** (Lag-1 Self-Autoregressive Causality), not shared by other baselines. To ensure a fair comparison, we treat lag-1 self-causal edges as correctly identified for all baselines during evaluation. Additionally, under Assumption **A5**, contemporaneous edges detected by some baselines are not counted as false positives. As VARLiNGAM is designed for non-Gaussian noise and TCDF works for nonlinear structural causal models (SCMs), it is expected that their overall performance is less competitive in this linear-Gaussian setting.

As shown in Fig. 3(a), MSSD, MSSD-PC, and MSSD-MCI consistently outperform other baselines in terms of F_1 score and SHD, with performance improving as T increases. TCDF achieves the highest precision, while DYNOTEARS yields the highest recall, closely followed by MSSD-MCI and PCMCI. Fig. 3(b) shows similar trends: MSSD and its variants achieve the best F_1 score; PCMCI, DYNOTEARS and MSSD-MCI perform similarly in terms of recall. TCDF has the highest precision and achieves the lowest SHD followed by MSSD and VARLiNGAM. Overall, most algorithms degrade as $\tau_{\max}$ increases, whereas VARLiNGAM remains stable across $\tau_{\max}$. In Fig. 4(a), MSSD again shows stronger robustness to increasing n, especially in terms of F_1 score and SHD. In summary, MSSD achieves the strongest overall performance across both F_1 score and SHD with varying T, $\tau_{\max}$ and n. MSSD prioritizes higher precision with lower recall, whereas MSSD-MCI trades some precision for better recall. The similar performance of MSSD and MSSD-PC suggests that the *Lag-Anchored Separating Set* plays a more decisive role than other minimal separating sets, and demonstrates that the *Lag-Anchored Separating Set*-based causal discovery strategy can be incorporated by other constraint-based methods.

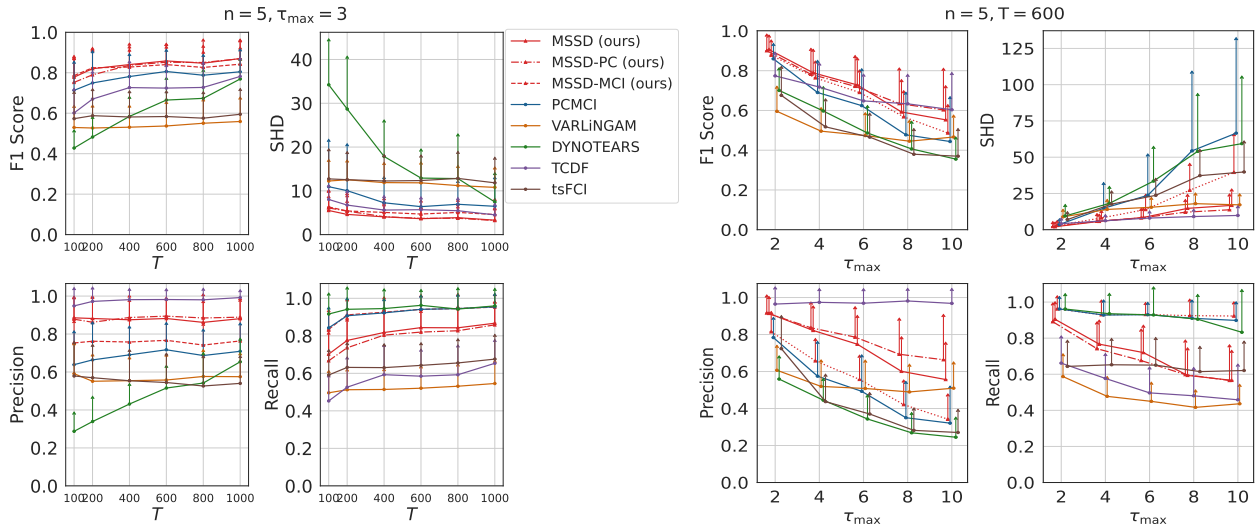
Shown in the top plot of Fig. 4(b), the average time lag at which MSSD terminates for each *univariate component time series* is approximately 4 when $\tau_{\max} = 10$, indicating that the algorithm typically identifies a *Lag-Anchored Separating Set* well before reaching $\tau_{\max}$. This advantage becomes especially beneficial when $\tau_{\text{ub}} > \tau_{\max}$. The average stopping lag is stable across different values of n.

4.1.2 Discrete Data

Baselines applicable to discrete-valued time series include PCMCI and tsFCI. As shown in Fig. 5(a), Fig. 5(b), and Fig. 11(a), MSSD-PC attains the highest F1 scores and the lowest SHD, outperforming the other methods across varying T, $\tau_{\max}$, and n.

As T increases, all methods exhibit mildly higher F1 and lower SHD. Relative to other baselines, tsFCI is notably robust across $\tau_{\max}$ in terms of F1, while MSSD-PC also shows strong robustness and accuracy over $\tau_{\max}$. When the number of variables n varies (Fig. 11(a)), MSSD-PC remains consistently strong across n, achieving the best F_1, SHD, and precision, followed by tsFCI and MSSD. By contrast, MSSD-MCI performs worst as T, $\tau_{\max}$, and n vary, primarily due to lower precision; it is also less robust, with performance fluctuating substantially across n.

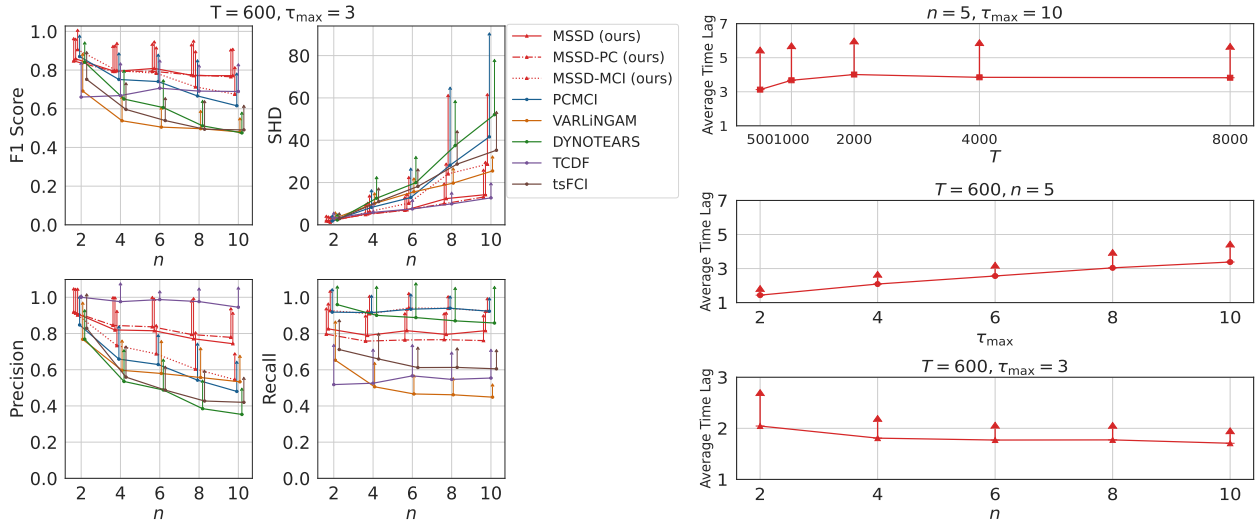
In summary, for binary-valued time series, MSSD-PC is the most accurate and stable, especially at larger $\tau_{\max}$ and T, whereas MSSD-MCI is less suitable in this setting.



(a) Performance when T varies

(b) Performance when $\tau_{\max}$ varies

Figure 3: Eight algorithms are evaluated on 5-dimensional multivariate time series with linear SCMs and Gaussian noise. Each line represents one algorithm, and each marker shows the average performance over 100 random trials in (a) and 50 random trials in (b) with error bars representing the standard deviation. (a) Adjacency F_1 score, SHD, Adjacency Precision, and Adjacency Recall are reported for varying time lengths T with fixed $\tau_{\max} = 3$. (b) The same metrics are reported for varying $\tau_{\max}$ with fixed $T = 600$.

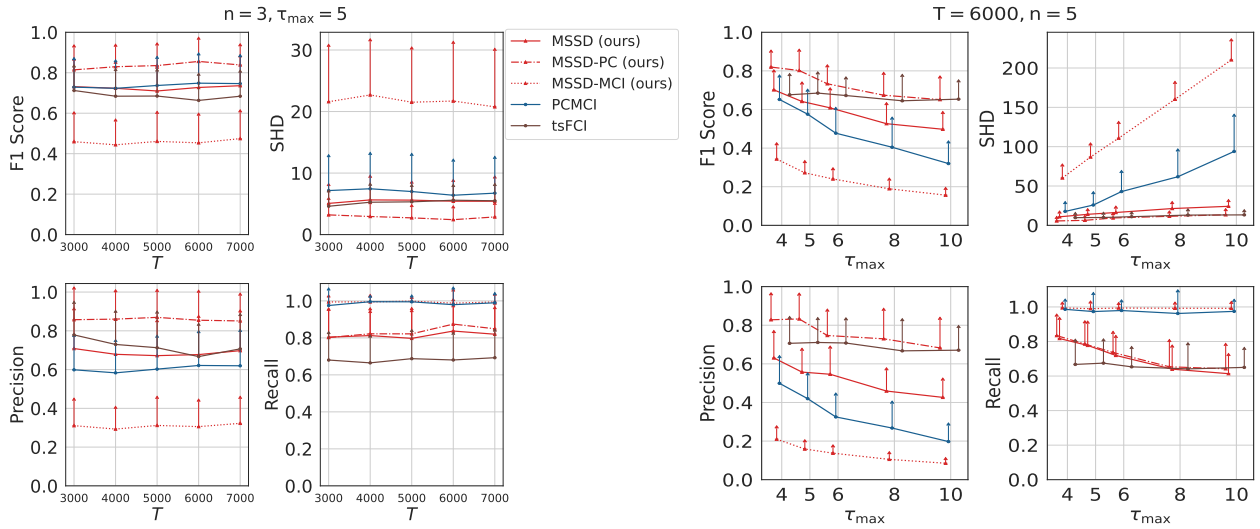


(a) Performance when n varies

(b) Average Time Lag

Figure 4: (a) Average Performance over 50 random trials under varying n with fixed $T = 600, \tau_{\max} = 3$. (b) Average time lag at which MSSD terminates for each *univariate component time series*, evaluated under varying T, $\tau_{\max}$, and n, separately.

Extended Experiments and Ablations. Please refer to Appendix Section E for extended simulations on high-dimensional datasets with $n \in [10, 100]$, other types of datasets, more benchmarks, and quantitative summary tables of the simulations. An ablation study on (i) incorrectly estimating τ_{ub} (i.e., $\tau_{\text{ub}} \neq \tau_{\max}$), (ii) violations of the Lag-1 self-autoregressive assumption (Assumption A7), and (iii) violations of A6 Causal Stationarity is presented in Appendix Section F.



(a) For binary-valued time series: performance when T varies. (b) For binary-valued time series: performance when $\tau_{\max}$ varies.

Figure 5: Five algorithms are evaluated on n-dimensional binary-valued multivariate time series. Each marker shows the average performance over 50 random trials with error bars representing the standard deviation. (a) Adjacency F_1 score, SHD, Adjacency Precision, and Adjacency Recall are reported for varying time lengths T with fixed $n = 3, \tau_{\max} = 5$. (b) The same metrics are reported for varying $\tau_{\max}$ with fixed $T = 6000, n = 5$.

Table 1: Random subgraphs with 3 nodes. Mean (Std). Best per column in **bold**.

Alg.	Prec. ↑	Rec. ↑	F1 ↑
MSSD	**0.46**(0.20)	0.73(0.32)	**0.54**(0.22)
tsFCI	0.40(0.15)	0.80(0.27)	0.52(0.15)
PCMCI	0.36(0.14)	**0.88**(0.24)	0.50(0.13)
MSSD-MCI	0.37(0.16)	**0.88**(0.24)	0.50(0.12)
MSSD-PC	0.41(0.26)	0.59(0.34)	0.46(0.26)
DYNOTEARS	0.37(0.28)	0.51(0.38)	0.39(0.27)
VARLiNGAM	0.17(0.16)	0.38(0.37)	0.24(0.21)

4.2 Real-World Dataset: River Flow

We evaluate on CausalRivers, the real-world time-series benchmark introduced by Stein et al. [2025]. The dataset contains river-discharge measurements (in m^3/s) at 15-minute resolution for eastern Germany (666 stations) and Bavaria (494 stations) from 2019–2023, together with a graph-sampling procedure that yields thousands of subgraphs spanning diverse settings.

The ground truth exploits hydrological physics: upstream discharge causally affects downstream discharge after a delay, and edges encode known flow directions along the river network. In our study, we uniformly sample 50 subgraphs with three nodes (following the benchmark's 3- and 5-node options) and set $\tau_{\max} = 3$ (as in one of the benchmark configurations). We estimate lagged graphs over a length-3 window and then evaluate on the corresponding *summary* causal graphs without explicit lags to match the provided ground truth. (TCDF produced empty graphs or incorrect edges in 45 of 50 subgraphs under these settings and is therefore omitted.)

Across these real-world subgraphs, MSSD attains the strongest precision and F_1, with tsFCI close behind, while PCMCI and MSSD-MCI achieve the highest recall, as shown in Table 1. Additional results about CausalRivers, including SHD metrics and the 5-node experiments, are provided in Appendix Section E.

5 CONCLUSION

We propose a constraint-based causal discovery method for time series under the stationarity assumption, guided by a special minimal separating set, the *Lag-Anchored Separating Set*, that forms a superset of the true parent set. Unlike conventional constraint-based causal discovery methods that search for direct parents using generic separating sets, the proposed approach leverages the causal stationarity and Lag-1 Self-Autoregressive Causality to characterize the *Lag-Anchored Separating Set*, thereby guiding the discovery process with greater accuracy and robustness. We establish the soundness of the algorithm and demonstrate its correctness and robustness across various datasets. Limitations are discussed in Appendix Section H.

Acknowledgements

We sincerely thank the reviewers for their thoughtful and insightful feedback, which helped improve the manuscript. This research has been supported in part by NSF CAREER 2239375, IIS 2348717, Amazon Research Award, Adobe Research and Intuit.

References

Bryan Andrews, Peter Spirtes, and Gregory F Cooper. On the completeness of causal discovery in the presence of latent confounding with tiered background knowledge. In *International Conference on Artificial Intelligence and Statistics*, pages 4002–4011. PMLR, 2020.

Charles K Assaad, Emilie Devijver, and Eric Gaussier. Survey and evaluation of causal discovery methods for time series. *Journal of Artificial Intelligence Research*, 73:767–819, 2022.

Yuxiao Cheng, Runzhao Yang, Tingxiong Xiao, Zongren Li, Jinli Suo, Kunlun He, and Qionghai Dai. Cuts: Neural causal discovery from irregular time-series data. *arXiv preprint arXiv:2302.07458*, 2023.

Yuxiao Cheng, Lianglong Li, Tingxiong Xiao, Zongren Li, Jinli Suo, Kunlun He, and Qionghai Dai. Cuts+: High-dimensional causal discovery from irregular time-series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11525–11533, 2024.

Diego Colombo, Marloes H Maathuis, Markus Kalisch, and Thomas S Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, pages 294–321, 2012.

Diego Colombo, Marloes H Maathuis, et al. Order-independent constraint-based causal structure learning. *J. Mach. Learn. Res.*, 15(1):3741–3782, 2014.

Doris Entner and Patrik O Hoyer. On causal discovery from time series data using fci. *Probabilistic graphical models*, 16, 2010.

Shanyun Gao, Raghavendra Addanki, Tong Yu, Ryan Rossi, and Murat Kocaoglu. Causal discovery in semi-stationary time series. *Advances in Neural Information Processing Systems*, 36:46624–46657, 2023.

Shanyun Gao, Raghavendra Addanki, Tong Yu, Ryan A Rossi, and Murat Kocaoglu. Causal discovery-driven change point detection in time series. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025.

Andreas Gerhardus and Jakob Runge. High-recall causal discovery for autocorrelated time series with latent confounders. *Advances in neural information processing systems*, 33:12615–12625, 2020.

Wiebke Günther, Urmi Ninad, and Jakob Runge. Causal discovery for time series from multiple datasets with latent contexts. In *Uncertainty in Artificial Intelligence*, pages 766–776. PMLR, 2023.

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.

Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. ISBN 0262013193.

Thuc Duy Le, Tao Hoang, Jiuyong Li, Lin Liu, Huawen Liu, and Shu Hu. A fast pc algorithm for high dimensional causal discovery with multi-core pcs. *IEEE/ACM transactions on computational biology and bioinformatics*, 16(5):1483–1495, 2016.

Meike Nauta, Doina Bucur, and Christin Seifert. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, 1(1):19, 2019.

Peer Nowack, Jakob Runge, Veronika Eyring, and Joanna D Haigh. Causal networks for climate model evaluation and constrained projections. *Nature communications*, 11(1):1415, 2020.

Mohammed Ombadi, Phu Nguyen, Soroosh Sorooshian, and Kuo-lin Hsu. Evaluation of methods for causal discovery in hydrometeorological systems. *Water Resources Research*, 56(7):e2020WR027251, 2020.

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.

Judea Pearl et al. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 19(2):3, 2000.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Causal inference on time series using restricted structural equation models. *Advances in neural information processing systems*, 26, 2013.

Jakob Runge. Causal network reconstruction from time series: From theoretical assumptions to practical estimation. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 28(7):075310, 2018.

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.

Jakob Runge, Peer Nowack, Marlene Kretschmer, Seth Flaxman, and Dino Sejdinovic. Detecting and quan-

tifying causal associations in large nonlinear time series datasets. *Science advances*, 5(11):eaau4996, 2019.

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.

Xinpeng Shen, Sisi Ma, Prashanthi Vemuri, and Gyorgy Simon. Challenges and opportunities with causal discovery algorithms: application to alzheimer's pathophysiology. *Scientific reports*, 10(1):2975, 2020.

Stephen M Smith, Karla L Miller, Gholamreza Salimi-Khorshidi, Matthew Webster, Christian F Beckmann, Thomas E Nichols, Joseph D Ramsey, and Mark W Woolrich. Network modelling methods for fmri. *Neuroimage*, 54(2):875–891, 2011.

Peter Spirtes. An anytime algorithm for causal inference. In *International Workshop on Artificial Intelligence and Statistics*, pages 278–285. PMLR, 2001.

Peter Spirtes, Christopher Meek, and Thomas Richardson. Causal inference in the presence of latent variables and selection bias. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 499–506, 1995.

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

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. The MIT Press, 01 2001. ISBN 9780262284158. doi: 10.7551/mitpress/1754.001.0001. URL `https://doi.org/10.7551/mitpress/1754.001.0001`.

Gideon Stein, Maha Shadaydeh, Jan Blunk, Niklas Penzel, and Joachim Denzler. Causalrivers – scaling up benchmarking of causal discovery for real-world time-series, 2025. URL `https://arxiv.org/abs/2503.17452`.

Xiangyu Sun, Oliver Schulte, Guiliang Liu, and Pascal Poupart. Nts-notears: Learning nonparametric dbns with prior knowledge. *arXiv preprint arXiv:2109.04286*, 2021.

Jin Tian, Azaria Paz, and Judea Pearl. *Finding minimal d-separators*. Citeseer, 1998.

Matej Vuković and Stefan Thalmann. Causal discovery in manufacturing: A structured literature review. *Journal of Manufacturing and Materials Processing*, 6(1):10, 2022.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in neural information processing systems*, 31, 2018.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes/No/Not Applicable]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes/No/Not Applicable]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes/No/Not Applicable]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes/No/Not Applicable]
   (b) Complete proofs of all theoretical results. [Yes/No/Not Applicable]
   (c) Clear explanations of any assumptions. [Yes/No/Not Applicable]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes/No/Not Applicable]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes/No/Not Applicable]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes/No/Not Applicable]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes/No/Not Applicable]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes/No/Not Applicable]
   (b) The license information of the assets, if applicable. [Yes/No/Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes/No/Not Applicable]
   (d) Information about consent from data providers/curators. [Yes/No/Not Applicable]
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Yes/No/Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Yes/No/Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes/No/Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes/No/Not Applicable]

Stationarity-Aware Causal Discovery in Time Series via Minimal Separating Sets: Supplementary Materials

Appendix

Contents

A Algorithms

A.1 Minimal-Separating-Set-based Discovery (MSSD) and Its Variants

Algorithm 1 Minimal-Separating-Set-based Discovery (MSSD)

Input: $V = (\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3, ..., \mathbf{X}^n)$, maximum lag upper bound τ^j_{ub}, series length T. τ^j_{ub} is assumed to be larger than $\tau^j_{\max}$. Maximum conditioning set size $d_{\max}$.
..... The following stage 1 could be combined with PC stage from PCMCI A2 .....
for all $\mathbf{X}^j$ where $j \in [n]$ **do**
  $\mathcal{C}(X^j_t) \leftarrow \emptyset$, $Dep\text{-}Time\text{-}Index \leftarrow \emptyset$, $M^j \leftarrow \emptyset$, $D^j_{\text{pa}} \leftarrow \emptyset, \forall t \in [T]$
  for all $\mathbf{X}^i$, $i \in [n], i \neq j$ **do**
    if $X^i_{t-1} \not\perp\!\!\!\perp X^j_t \,|\emptyset$ **then** ▷ Stage 1: Discovery of a Superset of the Parent Set.
      $Dep\text{-}Time\text{-}Index \leftarrow Dep\text{-}Time\text{-}Index \cup \{i\}$
  Set search space $\mathcal{C}(X^j_t) \leftarrow \{X^i_{t-\tau} : \tau \in [\tau^j_{\text{ub}}], i \in Dep\text{-}Time\text{-}Index \cup \{j\}\}$
  $found \leftarrow$ **False**
  for all $\mathbf{X}^i$, $i \in Dep\text{-}Time\text{-}Index \cup \{j\}$ **do**
    for all τ from 1 to $\tau^j_{\text{ub}} + 1$ **do**
      for $d = 2$ to $d_{\max}$ **do**
        for all $M \subseteq \mathcal{C}(X^j_t) \setminus X^i_{t-\tau}$ with $X^j_{t-1} \in M$ and $|M| = d$ **do**
          if $X^i_{t-\tau} \perp\!\!\!\perp X^j_t \,| M$ **then**
            $M^j \leftarrow M^j \cup M \cup M_{+1}$ ▷ $M_{+1} \coloneqq \{Z_{u+1} \mid Z_u \in M\}$.
            $found \leftarrow$ **True**
            break previous three loops
    if not $found$ **then**
      $D^j_{\text{pa}} \leftarrow D^j_{\text{pa}} \cup X^i_{t-\tau}$
  $\widehat{PA}(X^j_t) \leftarrow M^j \cup D^j_{\text{pa}}$
  ..... The following stage 2 could be replaced by MCI A3 .....
  for all $X^k_s \in \widehat{PA}(X^j_t)$ **do** ▷ Stage 2: Recovery of the True Parent Set.
    for $d = 1$ to $d_{\max}$ **do**
      if $X^k_s \perp\!\!\!\perp X^j_t \,\big|\, \widehat{PA}(X^j_t) \setminus X^k_s, \widehat{PA}(X^k_s)$ **then**
        Remove X^k_s from $\widehat{PA}(X^j_t)$
return $\widehat{PA}(X^j_t) \quad \forall j \in [n], t \in [T]$.

We introduce two variants of MSSD:

1) **MSSD-PC**. In this variant, we consider *all* minimal separating sets rather than only the Lag-Anchored Separating Sets (those that include X^j_{t-1}) in Stage 1. For each conditioning size d, we first attempt the Lag-Anchored option; if none succeeds, we then test the other minimal separating sets of the same size to determine whether the pair can be d-separated. In summary, when no Lag-Anchored set of size d is found, we fall back to the standard PC stage A2 for that variable pair—*prioritizing* the Lag-Anchored search while otherwise proceeding with the usual minimal-set exploration.

2) **MSSD-MCI**. In this variant, we replace Stage 2 with Algorithm A3. Specifically, after obtaining the superset $\widehat{\text{PA}}(X^j_t)$ from Stage 1, we run MCI tests over *all* variables in the lag window, rather than restricting the tests to those in $\widehat{\text{PA}}(X^j_t)$.

It is straightforward to construct an MSSD–PCMCI variant by prioritizing the search for Lag-Anchored Separating Sets when iterating over minimal separating sets. More broadly, the proposed approach can be readily generalized to other constraint-based algorithms.

A.2 PCMCI

PCMCI is a powerful constraint-based causal discovery algorithm for stationary time series, introduced by [Runge et al., 2019], designed to identify time-lagged causal relationships within a windowed causal graph. It is a two-stage procedure:

– $\mathrm{PC}_{q_{\max}}$ in Algorithm A2: Condition selection stage. PC_1 (by default, $q_{\max} = 1$) is a variant of the skeleton discovery phase of the PC algorithm, implemented in a more robust form known as stable-PC [Le et al., 2016]. This stage aims to identify a superset of the parents $\widehat{\mathbf{PA}}(X_t^j)$ for all variables $X_t^j \in \mathbf{X}$. The initial parent set is defined as $\widehat{\mathbf{PA}}(X_t^j) = \{X_{t-\tau}^i\}_{i\in[N],\tau\in[\tau_{\mathrm{ub}}]}$. $\widehat{\mathbf{PA}}(X_t^j)$ will remove $X_{t-\tau}^i$ if

$$X_{t-\tau}^i \perp\!\!\!\perp X_t^j \,\Big|\, \widehat{\mathbf{PA}}(X_t^j)\backslash\{X_{t-\tau}^i\} \tag{8}$$

– MCI in Algorithm A3: Momentary Conditional Independence (MCI) test causal discovery stage. In this stage, MCI tests are performed for all variable pairs $(X_{t-\tau}^i, X_t^j)$ with $i, j \in [n]$ and time delays $\tau \in [\tau_{\mathrm{ub}}]$. Remove variable $X_{t-\tau}^i$ from $\widehat{\mathbf{PA}}(X_t^j)$ if

$$X_{t-\tau}^i \perp\!\!\!\perp X_t^j \,\Big|\, \widehat{\mathbf{PA}}(X_t^j)\backslash\{X_{t-\tau}^i\}, \widehat{\mathbf{PA}}(X_{t-\tau}^i) \tag{9}$$

where $\widehat{\mathbf{PA}}(X_t^j)$ and $\widehat{\mathbf{PA}}(X_{t-\tau}^i)$ are obtained from the $\mathrm{PC}_{q_{\max}}$ stage.

Algorithm A2 $PC_{q_{\max}}$

Input: A n-variate time series $V = (\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3, ..., \mathbf{X}^n)$, selected time series $\mathbf{X}^j$, maximum time lag $\tau_{\max}$, significance threshold α_{PC}, maximum condition dimension $p_{\max}$ (default $p_{\max} = N\tau_{\max}$), maximum number of combinations $q_{\max}$ (default $q_{\max} = 1$). conditional independence test function
function $CI(X, Y, \mathbf{Z})$
    Test $X \perp\!\!\!\perp Y|\mathbf{Z}$ using test statistic measure
    return p-value, test statistic value I
Initialize preliminary set of parents $\widehat{\mathrm{PA}}(X_t^j) = \{X_{t-\tau}^i : i \in \{1, ..., n\}, \tau \in \{1, ..., \tau_{\max}\}\}$
Initialize dictionary of test statistic values $I^{\min}(X_{t-\tau}^i \to X_t^j) = \infty \; \forall X_{t-\tau}^i \in \widehat{\mathrm{PA}}(X_t^j)$
for $p = 0, 1, 2, ..., p_{\max}$ **do**
    if $|\widehat{\mathrm{PA}}(X_t^j)| - 1 < p$ **then**
        Break for-loop
    for all $X_{t-\tau}^i$ in $\widehat{\mathrm{PA}}(X_t^j)$ **do**
        $q = -1$
        for all lexicographically chosen subsets $\mathcal{S} \subseteq \widehat{\mathrm{PA}}(X_t^j) \setminus \{X_{t-\tau}^i\}$ with $|\mathcal{S}| = p$ **do**
            $q = q + 1$
            if $q \geq q_{\max}$ **then**
                Break from inner for-loop
            Run CI test to obtain (p-value, I) $\leftarrow CI(X_{t-\tau}^i, X_t^j, \mathcal{S})$
            if $|I| < I^{\min}(X_{t-\tau}^i \to X_t^j)$ **then**
                $I^{\min}(X_{t-\tau}^i \to X_t^j) = |I|$
            if p-value$> \alpha_{\mathrm{PC}}$ **then**
                Mark $X_{t-\tau}^i$ for removal from $\widehat{\mathrm{PA}}(X_t^j)$
                Break from inner for-loop
    Remove non-significant parents from $\widehat{\mathrm{PA}}(X_t^j)$
    Sort parents in $\widehat{\mathrm{PA}}(X_t^j)$ by $I^{\min}(X_{t-\tau}^i \to X_t^j)$ from largest to smallest
return $\widehat{\mathrm{PA}}(X_t^j)$

Algorithm A3 MCI

Input: A n-variate time series $V = (\mathbf{X}^1, \mathbf{X}^2, \mathbf{X}^3, ..., \mathbf{X}^n)$, sorted parents $\widehat{\mathrm{PA}}(X_t^j)$ for all variables X^j estimated with Algorithm 1, maximum time lag $\tau_{\max}$, maximum number p_X of parents of variable X^i, and conditional independence test function CI
for all $(X_{t-\tau}^i, X_t^j)$ with $i, j \in \{1, ..., n\}, \tau \in \{0, ..., \tau_{\max}\}$, excluding (X_t^j, X_t^j) **do**
    Remove $X_{t-\tau}^i$ from $\widehat{\mathrm{PA}}(X_t^j)$ if necessary
    Define $\widehat{\mathrm{PA}}_{p_X}(X_{t-\tau}^i)$ as the first p_X parents from $\widehat{\mathrm{PA}}(X_t^i)$, shifted by τ
    Run MCI test to obtain (p-value, I) $\leftarrow CI(X_{t-\tau}^i, X_t^j, \mathbf{Z} = \{\widehat{\mathrm{PA}}(X_t^j), \widehat{\mathrm{PA}}_{p_X}(X_{t-\tau}^i)\})$
Optionally adjust p-value of all links by False Discovery Rate-approach (FDR).
return p-value and MCI test statistics values

B Assumptions

For completeness, we state all required assumptions in this section.

B1. Sufficiency: There are no unobserved confounders.

B2. Causal Markov Condition: Each variable X is independent of all its non-descendants, given its parents $\mathbf{PA}(X)$ in $\mathcal{G}$.

B3. Faithfulness Condition [Pearl et al., 2000]: Let P be a probability distribution generated by $\mathcal{G}$. $\langle \mathcal{G}, P \rangle$ satisfies the Faithfulness Condition if and only if every conditional independence relation true in P is entailed by the Causal Markov Condition applied to $\mathcal{G}$.

B4. Temporal Priority: Causal relationships always flow from past to future.

B5. No Contemporaneous Causal Effects: There are no causal edges between variables within the same time step.

B6. Causal Stationarity: Causal relationships between variables remain consistent over time.

B7. Lag-1 Self-Autoregressive Causality: For any variable X_t^j in the *univariate component time series* $\mathbf{X}^j$, its immediate predecessor X_{t-1}^j is a parent of X_t^j, that is, $X_{t-1}^j \in \mathbf{PA}(X_t^j)$.

C Theoretical Guarantees

Definition C.1 (*Stationary* SCM)**.** A Stationary Structural Causal Model (SCM) is a tuple $\mathcal{M} = \langle V, \mathcal{F}, \mathcal{E}, \mathbb{P} \rangle$ where there exists $\tau^j_{\max} \in \mathbb{N}^+$ for any $j \in [n]$ and a $\tau_{\max} \in \mathbb{N}^+$, defined as:

$$\tau^j_{\max} \coloneqq \max_{\tau} \{\tau : X^i_{t-\tau} \in \mathbf{PA}(X^j_t), i \in [n]\}, \quad \tau_{\max} \coloneqq \max_{\tau} \{\tau : \tau^j_{\max}, j \in [n]\} \tag{10}$$

such that with this $\tau_{\max}$, each variable $X^j_{t>\tau_{\max}} \in \mathbf{V}$ is a deterministic function of its parent set $\mathbf{PA}(X^j_{t>\tau_{\max}}) \in \mathbf{V}$ and an unobserved (exogenous) variable $\epsilon^j_{t>\tau_{\max}} \in \mathcal{E}$,

$$X^j_t = f_j(\mathbf{PA}(X^j_t), \epsilon^j_t), \quad j \in [n],\ t \in [\tau_{\max}+1, T]. \tag{11}$$

where $f_j \in \mathcal{F}$ and $\{\epsilon^j_t\}_{t\in[T]}$ are jointly independent with probability measure $\mathbb{P}$. Thus $\tau^j_{\max}$ is the maximal lag for the jth *univariate component time series* $\mathbf{X}^j$ while $\tau_{\max}$ is the finite maximal lag for the entire causal graph $\mathcal{G}$. Note that f_j remains fixed across all time points t.

For simplicity, we sometimes use X, Y, Z and H to represent generic univariate variables, with subscripts indicating their time index. Unless otherwise specified, these variables without superscripts may belong to the same or different *univariate component time series*. In this section, $\tau_{\max}$ denotes the maximum time lag for the "target" variable Y, rather than for the entire multivariate time series V.

Definition C.2 (Minimal Separating Set)**.** A minimal separating set $\mathbf{M}$ between variables X and Y is a set that satisfies:

$$X \perp\!\!\!\perp Y \,|\mathbf{M} \quad \text{and} \quad X \not\perp\!\!\!\perp Y \,|\mathbf{M}_{\text{sub}} \quad \forall\, \mathbf{M}_{\text{sub}} \subsetneq \mathbf{M}. \tag{12}$$

Note that there may exist multiple minimal separating sets between X and Y. If X and Y are marginally independent, then the empty set is the *unique* minimal separating set.

Definition C.3 (Lag-Anchored Separating Set)**.** Let X_s and Y_t be two variables with $s < t$. A minimal separating set $\mathbf{M}$ between X_s and Y_t is called a Lag-Anchored Separating Set if it includes Y_{t-1}, i.e., $Y_{t-1} \in \mathbf{M}$. $\mathbf{M}$ is referred to as the Lag-Anchored Separating Set of the target variable Y_t.

Note that a *Lag-Anchored Separating Set* is defined with respect to Y_t, the specific choice of variable X_s being tested against Y_t is not essential.

Using Lemma C.4, we restrict the search for the *Lag-Anchored Separating Set* of Y_t to variables that occur before t, since *Lag-Anchored Separating Set* is a minimal separating set and it does not contain any variable that occurs at or after t.

Lemma C.4 (Temporal Constraint on Minimal Separating Sets)**.** *Let X_s and Y_t be two variables with $s < t$ and let $\mathbf{M}$ be any minimal separating set (Definition C.2) between X_s and Y_t. Then $\mathbf{M}$ does not include any variable of the form Z_u with $u \geq t$, under Assumption **B1-B7**.*

Proof. Let $\mathbf{U}$ denote the set of variables in $\mathbf{M}$ whose time point is equal to or greater than t.

For the sake of contradiction, suppose otherwise we have $Z_u \in \mathbf{M}$ with $u \geq t$, that is, $\mathbf{U} \neq \emptyset$.

Due to the minimality of $\mathbf{M}$, conditioning on $\mathbf{M}\backslash\mathbf{U}$, which is a strict subset of $\mathbf{M}$, should render an otherwise d-separated path d-connecting. Therefore, there must be such a path p that is d-connecting given $\mathbf{M}\backslash\mathbf{U}$ but d-separated given $\mathbf{M}$.

Let us consider two cases, depending on whether the path p between X_s and Y_t includes any node Z_u with $u \geq t$.

Case 1. The path p does not contain any nodes Z_u with $u \geq t$, except for Y_t. However, this case is impossible because adding nodes to a conditioning set that are not on p can only make an otherwise d-separated path d-connecting, not the other way around.

Case 2. The path p contains some nodes Z_u with $u \geq t$, other than Y_t.

Let $Z_{v_{\max}}$ denote the variable with the largest time point on path p. Since p contains some nodes Z_u with $u \geq t$ other than Y_t, the node $Z_{v_{\max}}$ with the largest time point must be an internal node on the path p, connected to

its two neighboring nodes along the path. The two edges pointing to $Z_{v_{\max}}$ that connect it with other variables on path p must both have arrowheads directed towards $Z_{v_{\max}}$. Otherwise, if either edge has a tail at $Z_{v_{\max}}$, the child of $Z_{v_{\max}}$ connected by that tail would have a time point greater than that of $Z_{v_{\max}}$, which contradicts the fact that $Z_{v_{\max}}$ has the largest time point on path p, given the Temporal Priority assumption B4 and the No Contemporaneous Causal Effects assumption B5. Therefore, $Z_{v_{\max}}$ must act as a collider on path p.

Since p contains some nodes Z_u where $u \geq t$ and $Z_{v_{\max}}$ has the largest time point on path p, we have $v_{\max} \geq t$. Note that $\mathbf{M} \setminus \mathbf{U}$ does not contain any nodes with time points $\geq t$, so neither $Z_{v_{\max}}$ nor any of its descendants are included in $\mathbf{M} \setminus \mathbf{U}$. Therefore, since $Z_{v_{\max}}$ is a collider, the path p must be blocked given $\mathbf{M} \setminus \mathbf{U}$. More specifically, because neither the collider $Z_{v_{\max}}$ nor its descendants are conditioned on in $\mathbf{M} \setminus \mathbf{U}$, the path p is blocked due to the presence of an unconditioned collider and all its descendants.

Therefore, any p that contains some node Z_u with $u \geq t$ is blocked given $\mathbf{M} \setminus \mathbf{U}$. This fact holds no matter p is d-connected or d-separated given $\mathbf{M}$.

Since Case 1 and Case 2 together cover all possible situations, we conclude that no such path p exists—that is, no path that is d-separated given $\mathbf{M}$ but d-connecting given $\mathbf{M} \setminus \mathbf{U}$. Therefore, $\mathbf{M} \setminus \mathbf{U}$ is a strict subset of $\mathbf{M}$ and serves as a smaller separating set. As a result, $\mathbf{M}$ cannot be a minimal separating set. By contradiction, it follows that no variable $Z_u \in \mathbf{M}, u \geq t$.

□

Lemma C.5 states that under the Causal Stationarity assumption B6, any path in the time-unrolled causal graph remains time-shifted invariant under uniform time shifts of all variables along the path. In other words, if both endpoints of a path are shifted by the same amount, all paths between them shift accordingly while preserving their structure.

Lemma C.5 (Stationarity of Paths)**.** *In a stationary SCM (Definition C.1), let* $\pi \coloneqq \{X_s, Z_{t_0}^{i_0}, Z_{t_1}^{i_1}, \cdots, Z_{t_l}^{i_l}, Y_t\}$ *be a path between* X_s *and* Y_t *with* $s < t$*. Then for any integer* δ*, the shifted path*

$$\pi^{\delta} = \{X_{s+\delta}, Z_{t_0+\delta}^{i_0}, Z_{t_1+\delta}^{i_1}, \cdots, Z_{t_l+\delta}^{i_l}, Y_{t+\delta}\} \tag{13}$$

is also a path linking variables $X_{s+\delta}$ *and* $Y_{t+\delta}$ *where all variables along the original path* π *are uniformly shifted forward by* δ *time steps with* $\delta > -\min\{s, t_0, t_1, \cdots, t_l\}$*.*

Proof. By the definition of a stationary SCM (Definition C.1), the causal mechanism of each variable is invariant under time shifts, and the causal graph structure repeats identically over time. In particular, the existence of an edge between $Z_{t_0}^{i_0}$ and $Z_{t_1}^{i_1}$ implies the existence of a corresponding edge between $Z_{t_0+\delta}^{i_0}$ and $Z_{t_1+\delta}^{i_1}$ for any integer $\delta > -\min\{t_0, t_1\}$. Given a path $\pi = \{X_s, Z_{t_0}^{i_0}, Z_{t_1}^{i_1}, \cdots, Z_{t_l}^{i_l}, Y_t\}$, there exist edges between successive pairs of variables $(X_s, Z_{t_0}^{i_0})$, $(Z_{t_0}^{i_0}, Z_{t_1}^{i_1})$, $(Z_{t_1}^{i_1}, Z_{t_2}^{i_2})$, $\cdots$, $(Z_{t_l}^{i_l}, Y_t)$. By Causal Stationarity assumption B6, this implies the existence of corresponding edges between the time-shifted variables $(X_{s+\delta}, Z_{t_0+\delta}^{i_0})$, $(Z_{t_0+\delta}^{i_0}, Z_{t_1+\delta}^{i_1})$, $(Z_{t_1+\delta}^{i_1}, Z_{t_2+\delta}^{i_2})$, $\cdots$, $(Z_{t_l+\delta}^{i_l}, Y_{t+\delta})$. Hence, the sequence $\pi^{(\delta)} = \{X_{s+\delta}, Z_{t_0+\delta}^{i_0}, Z_{t_1+\delta}^{i_1}, \cdots, Z_{t_l+\delta}^{i_l}, Y_{t+\delta}\}$ also forms a valid path. Moreover, the path $\pi^{(\delta)}$ preserves the same structure as π, maintaining both the variable ordering and edge directions, with the only difference being a uniform shift in time. □

Lemma C.6 investigates the time-shift invariance property of separating sets, including minimal separating sets, that do not satisfy the condition of being a *Lag-Anchored Separating Set*, namely, those that do not contain Y_{t-1}. Although this lemma is not directly used in the proposed algorithm, it sheds light on a distinct class of minimal separating sets and highlights the critical role of Y_{t-1}.

Lemma C.6 (Stationarity of Certain Separating Set)**.** *In a stationary SCM (Definition C.1), let* X_s *and* Y_t *be two variables with* $s < t$*. For any set* $\mathbf{M} \not\ni Y_{t-1}$*,*

if we have:

$$X_{s-1} \perp\!\!\!\perp Y_t \,|\mathbf{M}\,, \tag{14}$$

then

$$X_s \perp\!\!\!\perp Y_t \,|\mathbf{M}_{+1}\,, \tag{15}$$

where $\mathbf{M}_{+1} \coloneqq \{X^k_{u+1} : X^k_u \in \mathbf{M}\}$.

Proof. Since $Y_{t-1} \notin \mathbf{M}$, the set $\mathbf{M}$ is a separating set for the pair (X_{s-1}, Y_{t-1}). Otherwise, the path obtained by concatenating the path d-connecting X_{s-1} and Y_{t-1} with the edge $Y_{t-1} \to Y_t$ would not be d-separated given $\mathbf{M}$, implying that X_{s-1} is d-connected to Y_t given $\mathbf{M}$, which contradicts our assumption. Therefore, X_{s-1} and Y_{t-1} are d-separated by $\mathbf{M}$. According to Lemma C.5, the shifted set $\mathbf{M}_{+1}$ d-separates all paths from X_s to Y_t. □

We present two facts that will be used in the proofs of the subsequent lemmas and theorems.

Fact C.1. *Let X_s and Y_t be two variables with $s < t$ and let $\mathbf{M}$ be a minimal separating set (Definition C.2) between X_s and Y_t. If $\mathbf{M}$ is nonempty, given the definition of the minimal separating set C.2, every node in $\mathbf{M}$ must act as a non-collider variable on at least one path between X_s and Y_t.*

Proof. Otherwise, if a node appears only as a collider on all paths containing it, then removing it from $\mathbf{M}$ does not unblock any path between the two variables. Hence, it can be excluded from $\mathbf{M}$ without affecting the separating property, contradicting the definition of $\mathbf{M}$. □

Fact C.2. *Let X_s and Y_t be two variables with $s < t$ and let $\mathbf{M}$ be a minimal separating set (Definition C.2) between X_s and Y_t. If $\mathbf{M}$ is nonempty, there exists at least one path between X_s and Y_t that consists solely of non-colliders, with at least one node on this path belonging to $\mathbf{M}$.*

Proof. Otherwise, if all paths contain at least one collider, by d-separation, paths with unconditioned colliders are blocked automatically. Thus, X_s and Y_t would already be d-separated by $\emptyset$. This contradicts the assumption that $\mathbf{M}$ is nonempty. □

Lemma C.7 (Marginal Independence in Stationary SCM)**.** *Let X_s and Y_t be two variables with $s < t$ and suppose they are marginally independent, i.e., $X_s \perp\!\!\!\perp Y_t$. Then, the following holds for any $u < t$ under Assumption **B1-B7**:*

$$X_u \perp\!\!\!\perp Y_t, \quad \forall X_u \in \mathbf{X} \tag{16}$$

where $\mathbf{X} \coloneqq \{X_v, v \in [t-1]\}$ denotes the univariate component time series to which X_u belongs.

Proof. Due to self-causality, X_s and Y_t cannot belong to the same *univariate component time series*; that is, $Y_t \notin \mathbf{X}$. For the sake of contradiction, we assume $X_u \not\perp\!\!\!\perp Y_t$ for some $u < t$. Therefore, there must be a d-connecting path $\tilde{p}$ between X_u and Y_t when conditioning on the empty set. We proved that for either $u < s$ or $s < u < t$, there would be a d-connecting path p between X_s and Y_t constructed on the basis of $\tilde{p}$ under **B1-B7**.

Case 1: $s < u < t$:

For the sake of contradiction, we assume that there is a minimal separating set $\widetilde{\mathbf{M}}$ with $|\widetilde{\mathbf{M}}| \geq 1$ between X_u and Y_t with $s < u < t$.

Due to Fact C.2, there must exist a set of paths $\widetilde{\mathcal{P}}$ in which the paths only consist of non-colliders between X_u and Y_t. We can select such a path $\tilde{p}$ from $\widetilde{\mathcal{P}}$ consisting solely of non-colliders.

Now shift the path $\tilde{p}$ time steps backward by $u - s$. Let $\tilde{p}_{\text{shift}}$ denote the shifted path. $\tilde{p}_{\text{shift}}$ can be expressed as:

$$\tilde{p}_{\text{shift}} \coloneqq X_s \cdots Y_{t-(u-s)} \tag{17}$$

Based on Lemma C.5, $\tilde{p}_{\text{shift}}$ also consists of only non-colliders, it is unblocked conditioning on an empty set.

We extend $\tilde{p}_{\text{shift}}$ by linking the variable $Y_{t-(u-s)}$ and Y_t with $\{Y_{t-(u-s)+\delta}\}_{\delta \in [u-s]}$ and consecutive outgoing edges. Note that $Y_{t-(u-s)}$ must be the closest variable to Y_t that belongs to $\mathbf{Y}$ since all other variables on $\tilde{p}_{\text{shift}}$ should have a smaller time point based on Lemma C.4.

The extended path can be expressed as:

$$p \coloneqq \underbrace{X_s \cdots Y_{t-(u-s)}}_{\tilde{p}_{\text{shift}}} \to Y_{t-(u-s)+1} \to Y_{t-(u-s)+2} \to \cdots \to Y_{t-1} \to Y_t \tag{18}$$

Since all newly added variables $\{Y_{t-(u-s)+\delta}\}_{\delta \in [u-s]}$ on the path p are also non-colliders, all variables on this path are non-colliders. Therefore the path p are open conditioned on an empty set and $X_s \not\perp\!\!\!\perp Y_t \,|\emptyset$, which contradicts our assumption that $X_s \perp\!\!\!\perp Y_t \,|\emptyset$. By contradiction, $X_u \perp\!\!\!\perp Y_t \,|\emptyset$ for all $s < u < t$.

Case 2: $u < s$:

For the sake of contradiction, we assume that there is a minimal separating set $\widetilde{\mathbf{M}}$ with $|\widetilde{\mathbf{M}}| \geq 1$ between X_u and Y_t with $u < s$.

Due to Fact C.2, there must exist a set of paths only consisting of non-colliders between X_u and Y_t.

Again, we can select such a path consisting solely of non-colliders and denote it by $\tilde{p}$.

Denote by $X_{u+\delta}$ the variable closest to X_s on path $\tilde{p}$. Therefore $0 \leq \delta \leq s - u$.

In this case, we can establish a path p:

$$p \coloneqq X_s \leftarrow X_{s-1} \leftarrow \cdots \leftarrow \underbrace{X_{u+\delta} \cdots Y_t}_{p_{\text{tr}}} \tag{19}$$

where p_{tr} denotes the truncated subpath of $\tilde{p}$ starting from $X_{u+\delta}$ and ending at Y_t. If there is no other variable that belongs to $\mathbf{X}$ on path $\tilde{p}$ except for X_u, then $\delta = 0$ and $p_{\text{tr}} = \tilde{p}$. Since the path consists only of non-colliders, it remains open if none of its nodes are included in the minimal separating set, that is, $X_s \not\perp\!\!\!\perp Y_t \,|\emptyset$, thereby leading to a contradiction with the assumption that $X_s \perp\!\!\!\perp Y_t \,|\emptyset$.

Therefore by contradiction, we must have $X_u \perp\!\!\!\perp Y_t \,|\emptyset\,, \;\; \forall X_u \in \mathbf{X}$ with $0 \leq u < t$ in both Case 1 and Case 2 which cover all situations. □

By this lemma, if $X_s \perp\!\!\!\perp Y_t \,|\emptyset$ holds for some time point $s < t$, then no variable from the *univariate component time series* $\mathbf{X}$ can be a parent of Y_t. Consequently, it is unnecessary to test conditional independence between any other variables in $\mathbf{X}$ and Y_t.

Lemma C.8 (Characterization of Lag-Anchored Separating Set)**.** *Let X_s and Y_t be two variables with $s < t$ and let $\mathbf{M}$ be a Lag-Anchored Separating Set (Definition C.3) between X_s and Y_t satisfying $Y_{t-1} \in \mathbf{M}$. Then for any $Z_u \in \boldsymbol{PA}(Y_t) \backslash \mathbf{M}$, the following holds under Assumption **B1-B7**:*

$$Z_{u-1} \in \mathbf{M}. \tag{20}$$

Proof. Since $Y_{t-1} \in \mathbf{M}$, removing Y_{t-1} from $\mathbf{M}$ will open a set of paths between X_s to Y_t; let us denote this collection of paths as $\mathcal{R}$. According to Fact C.1, Y_{t-1} acts as a non-collider on the paths in $\mathcal{R}$, and there exists at least one such path. Let $v^r \coloneqq \arg\max\{v \mid H_v \in \mathbf{PA}(Y_{t-1}) \cap r\}$ [3], and H_{v^r} presents the latest parent of Y_{t-1} along the path r. Given Lemma C.4, any $r \in \mathcal{R}$ does not contain any variable with time point $\geq t$. Therefore r cannot pass through Y_{t-1} from one of its children; otherwise, r would already be blocked by an unconditioned collider without removing Y_{t-1} from $\mathbf{M}$. Thus, around Y_{t-1}, the path r must take the form $\cdots H_{v^r} \to Y_{t-1} \to Y_t$ ensuring that Y_{t-1} is always entered via an incoming arrow on r.

For any $Z_u \in \mathbf{PA}(Y_t) \backslash \mathbf{M}$, we are going to prove that $Z_{u-1} \in \mathbf{M}$ given $Y_{t-1} \in \mathbf{M}$ by considering two cases, depending on whether there exists a path $r \in \mathcal{R}$ that does not pass through Z_{u-1}.

For any $Z_u \in \mathbf{PA}(Y_t) \backslash \mathbf{M}$:

Case 1: There exists a path $r \in \mathcal{R}$ that does *not* include the variable Z_{u-1}.

Note that Z_{u-1} must be a parent of Y_{t-1} due to the Causal Stationarity assumption B6. Then, we can construct a path by concatenating the path r truncated at Y_{t-1}, named r_{tr}, with $\leftarrow Z_{u-1} \to Z_u \to Y_t$; let us call this concatenated path as r_{colli}. Then the path r_{colli} can be expressed as:

[3] Here r is used as a set and denotes the set of variables along the path r. We allow this abuse of notation for simplicity.

$$r_{\text{colli}} \coloneqq \underbrace{X_s \cdots H_{v^r}}_{r_{\text{tr}}:\ \text{open given } \mathbf{M}} \to \underbrace{Y_{t-1}}_{\in \mathbf{M}} \leftarrow Z_{u-1} \to \underbrace{Z_u}_{\notin \mathbf{M}} \to Y_t \tag{21}$$

Since r_{tr} is open given $\mathbf{M}$ by the definition of $\mathcal{R}$, Y_{t-1} acts as a collider conditioned on in $\mathbf{M}$, and Z_u is a non-collider unconditioned in $\mathbf{M}$, we have r_{colli} is open given $\mathbf{M}$ if $Z_{u-1} \notin \mathbf{M}$, which contradicts the assumption that $\mathbf{M}$ is a minimal separating set. Therefore, by contradiction, we must have that $Z_{u-1} \in \mathbf{M}$.

Case 2: Every path $r \in \mathcal{R}$ passes through Z_{u-1}. Therefore we cannot find an r that has a corresponding r_{colli} expressed in Eq. 21.

In this case, every $r \in \mathcal{R}$ can be expressed as:

$$r \coloneqq \underbrace{X_s \cdots}_{r'_{tr}:\ \text{open given } \mathbf{M}} \quad Z_{u-1} \cdots \to Y_{t-1} \to Y_t \tag{22}$$

Since r is open if removing Y_{t-1} from $\mathbf{M}$, we know that the path r truncated at Z_{u-1} is d-connected given $\mathbf{M}$ no matter Z_{u-1} itself is a collider or not. Name this truncated path as r'_{tr}. Therefore there will be a path r_{con}:

$$r_{\text{con}} \coloneqq \underbrace{X_s \cdots}_{r'_{tr}:\ \text{open given } \mathbf{M}} \quad Z_{u-1} \to \underbrace{Z_u}_{\notin \mathbf{M}} \to Y_t \tag{23}$$

Since r'_{tr} is open given $\mathbf{M}$ and Z_u serves as a non-collider unconditioned in $\mathbf{M}$. Therefore r_{con} is only d-separated if $Z_{u-1} \in \mathbf{M}$, considering that Z_u is also a non-collider on path r_{con}.

Therefore, for any $Z_u \in \mathbf{PA}(Y_t) \setminus \mathbf{M}$, we have $Z_{u-1} \in \mathbf{M}$ holds in both Case 1 and Case 2, thereby covering all possible configurations for each such Z_u. **In other words, for any $Z_u \in \mathbf{PA}(Y_t)$, Z_u and Z_{u-1} cannot be outside of a *Lag-Anchored Separating Set* $\mathbf{M}$ at the same time.** □

Lemma C.8 guarantees that, for every *Lag-Anchored Separating Set* (LASS), if a parent is not included in the LASS, its lag-1 backward-shifted counterpart must be included. With this property, a superset of parents can be obtained from LASS (by taking the union of the LASS and its lag-1 forward-shifted version), thereby facilitating the causal discovery.

To illustrate, we go back to Figure 1 in the main paper. Consider Z_7, a parent of the target variable H_9. Suppose a LASS is already identified between X_4 and H_9. Lemma 2.6 then aims to establish that:

$$\text{(S1)} \quad \text{If } Z_7 \notin \text{LASS}, \text{ then } Z_6 \in \text{LASS}.$$

Lemma C.8 shows that all paths between X_4 and H_9 via H_8 fall into two structural cases depending on whether Z_6 appears on the path. In both cases, statement (S1) must hold; otherwise, X_4 and H_9 would remain d-connected given the LASS, contradicting the fact that a LASS is a minimal separating set.

We now establish the existence of a *Lag-Anchored Separating Set*. First, Lemma C.9 shows that there exists a minimal separating set that is a subset of the parent set of the target variable Y_t. Then, Lemma C.10 demonstrates that any such minimal separating set must contain Y_{t-1}, thereby satisfying the definition of a *Lag-Anchored Separating Set*. Hence, a *Lag-Anchored Separating Set* must exist.

Lemma C.9 (Existence of a Minimal Separating Set Contained in the Parent Set)**.** *Let X_s and Y_t be two variables with $s < t$ and let $\mathbf{M}$ denote a minimal separating set (Definition C.2) between X_s and Y_t. If the minimal separating set $\mathbf{M}$ between X_s and Y_t is not an empty set, there exists a non-empty minimal separating set $\mathbf{M}$ satisfying $\mathbf{M} \subseteq \boldsymbol{PA}(X_t^j)$ under Assumption **B1-B7**.*

Proof. We know that $X_s \perp\!\!\!\perp Y_t \,|\mathbf{PA}(Y_t)$ with $s < t$ given Lemma C.4, which implies that $\mathbf{PA}(Y_t)$ is a separating set. If $\mathbf{PA}(Y_t)$ is already minimal, then we set $\mathbf{M} = \mathbf{PA}(Y_t)$. Otherwise, we iteratively remove one element at a time from $\mathbf{PA}(Y_t)$ and check whether the separation still holds. Repeating this process yields a minimal separating set $\mathbf{M} \subsetneq \mathbf{PA}(Y_t)$. □

Lemma C.10 (Existence of Lag-Anchored Separating Set)**.** *Let X_s and Y_t be two variables with $s < t - \tau_{\max}$ and let $\mathbf{M}$ denote a minimal separating set (Definition C.2) between X_s and Y_t. If the minimal separating set $\mathbf{M}$ between X_s and Y_t is not an empty set, then for any minimal separating set $\mathbf{M}$ satisfying $\mathbf{M} \subseteq \boldsymbol{PA}(X_t^j)$, under Assumption **B1-B7** we have:*

$$\{Y_{t-\tau_u}, Y_{t-1}\} \subset \mathbf{M} \tag{24}$$

where $Y_{t-\tau_u}$ denotes the oldest parent of Y_t on *univariate component time series* $\mathbf{Y}$. Note that Y_{t-1} is always a parent of Y_t on $\mathbf{Y}$ with time lag 1 given assumption B7; and $1 \leq \tau_u \leq \tau_{\max}$.

Proof. Let $\mathcal{P}$ denote the set of paths between X_s and Y_t where Y_{t-1} serves as a non-collider and no variable on the path has a time point $\geq t$, except Y_t.

This lemma can be proved once we can find a path $p \in \mathcal{P}$ that is blocked if and only if $Y_{t-1} \in \mathbf{M}$ given $\mathbf{M} \subseteq \mathbf{PA}(X_t^j)$.

In particular, it is sufficient to consider a finer path collection $\tilde{\mathcal{P}}$ where $\tilde{\mathcal{P}} \subseteq \mathcal{P}$. Besides satisfying the conditions that Y_{t-1} serves as a non-collider and no variable on the path has a time point $\geq t$, the additional condition is that $p \in \tilde{\mathcal{P}}$ does not contain any variables from $\mathbf{PA} \setminus Y_{t-1}$ and p consists of solely non-colliders. Since Y_{t-1} serves as a non-collider variable on p, all other variables in $\mathbf{M} \subseteq \mathbf{PA}(Y_t)$ are not on p and p consists of solely non-colliders, $p \in \tilde{\mathcal{P}}$ is blocked if and only if $Y_{t-1} \in \mathbf{M}$. Now we are going to prove that $\tilde{\mathcal{P}}$ is not an empty set by finding one path $p \in \tilde{\mathcal{P}}$.

Due to the Fact C.2, there must be one path between X_s and Y_t containing only non-colliders. Let e denote one of such paths.

Now shift path e with $\tau_{\max} + 1$ time steps backward. Let e_{shift} denote the shifted e, which is in the form of:

$$e_{\text{shift}} \coloneqq X_{s-\tau_{\max}-1} \cdots Y_{t-\tau_{\max}-1} \tag{25}$$

All variables on e_{shift} are at least $\tau_{\max} + 1$ time steps earlier than Y_t, and thus none can be a parent of Y_t. Moreover, since e consists of non-colliders and the path e_{shift} is formed by uniformly shifting the time indices of all variables on e backward, the collider and non-collider status of each variable is preserved by Lemma C.5, ensuring that e_{shift} also contains no colliders.

Since $Y_{t-\tau_u} \in \mathbf{PA}(Y_t)$, due to the Causal Stationarity assumption B6, we have:

$$Y_{t-\tau_u} \to Y_t \Leftrightarrow Y_{t-\tau_u-1} \to Y_{t-1} \tag{26}$$

We construct an extended path based on e_{shift}, named e_{con}:

$$\begin{aligned} e_{\text{con}} \coloneqq & X_s \leftarrow X_{s-1} \leftarrow X_{s-2} \leftarrow \cdots \leftarrow \underbrace{X_{s-\tau_{\max}-1} \cdots Y_{t-\tau_{\max}-1}}_{e_{\text{shift}}} \\ & \to Y_{t-\tau_{\max}} \to Y_{t-\tau_{\max}+1} \to \cdots \to Y_{t-\tau_u-1} \to Y_{t-1} \to Y_t \end{aligned} \tag{27}$$

Since $s < t - \tau_{\max}$, for all $v = s - \delta$ with $\delta > 0$, there is no direct edge from X_v to Y_t. As a result, any variable on e_{con} that belongs to $\mathbf{X}$ is not one of the parents of Y_t. The same for all variables on path e_{shift} due to shifting the time steps backward by $\tau_{\max} + 1$ and Lemma C.4. Furthermore, since $Y_{t-\tau_u}$ is the oldest parent of Y_t on $\mathbf{Y}$, any variable that belongs to $\{Y_{t-\delta-1}\}$ where $\delta \in [\tau_u, \tau_{\max}]$ also cannot be one of the parents of Y_t. Therefore, any variable on path e_{con} is not one of the parents of Y_t except Y_{t-1}. Note that τ_u may equal $\tau_{\max}$, in which case the path e_{con} around Y_t is in the form of $Y_{t-\tau_{\max}-1} \to Y_{t-1} \to Y_t$, directly connecting from the last node $Y_{t-\tau_{\max}-1}$ of e_{shift}.

Since e_{shift} only contains non-colliders and the newly added variables $\{X_{s-\delta}\}$ where $\delta \in [\tau_{\max}]$ and $\{Y_{t-\delta-1}\}$ where $\delta \in [\tau_u, \tau_{\max} - 1]$ are all non-colliders with outgoing edges, all variables on path e_{con} are non-colliders.

Given $\mathbf{M} \subseteq \mathbf{PA}(Y_t)$, e_{con} can only be blocked by conditioning on Y_{t-1}, since all other parents of Y_t do not lie on the path e_{con} except for Y_{t-1} and e_{con} consists of only non-colliders, rendering their consideration irrelevant. More specifically, the presence of any variable from $\mathbf{PA}(Y_t) \setminus Y_{t-1}$ in $\mathbf{M}$ does not affect the status of the path e_{con}. The path e_{con} is blocked if and only if $Y_{t-1} \in \mathbf{M}$ given $\mathbf{M} \subseteq \mathbf{PA}(Y_t)$. Therefore, Y_{t-1} must be included in $\mathbf{M}$ given $\mathbf{M} \subseteq \mathbf{PA}(Y_t)$.

Additionally, given that $Y_{t-\tau_u-1} \notin \mathbf{M}$, we must have $Y_{t-\tau_u} \in \mathbf{M}$ otherwise the following path $\tilde{e}_{\text{con}}$ is open:

$$\begin{aligned}\tilde{e}_{\text{con}} \coloneqq & X_s \leftarrow X_{s-1} \leftarrow X_{s-2} \leftarrow \cdots \leftarrow \underbrace{X_{s-\tau_{\max}-1} \cdots Y_{t-\tau_{\max}-1}}_{e_{\text{shift}}} \\ & \rightarrow Y_{t-\tau_{\max}} \rightarrow Y_{t-\tau_{\max}+1} \rightarrow \cdots \rightarrow Y_{t-\tau_u-1} \rightarrow Y_{t-\tau_u} \rightarrow Y_t\end{aligned}$$

The difference between $\tilde{e}_{\text{con}}$ and e_{con} lies in the last three variables leading to Y_t: $\tilde{e}_{\text{con}}$ contains the segment $Y_{t-\tau_u-1} \rightarrow Y_{t-\tau_u} \rightarrow Y_t$, whereas e_{con} includes $Y_{t-\tau_u-1} \rightarrow Y_{t-1} \rightarrow Y_t$, as shown in Eq. 27. The key structure used here is the Causal Stationarity B6. □

The Lemma C.11 is not directly used in the algorithm but provides insight into the causal graph structure. It states that when a minimal separating set is a subset of the parent set of the target variable, this occurs because certain variables Z_{u-1} act as "sheltered" parents, effectively blocking all paths that pass through their descendants Z_u.

Lemma C.11 (A Fun Fact)**.** *Let X_s and Y_t be two variables with $s < t - \tau_{\max}$ and let $\mathbf{M}$ denote a minimal separating set (Definition C.2) between X_s and Y_t. If $\mathbf{M}$ is nonempty and $\mathbf{M} \subsetneq \boldsymbol{PA}(Y_t)$, we have for all $Z_u \in \boldsymbol{PA}(Y_t) \setminus \mathbf{M}$:*

$$Z_{u-1} \in \boldsymbol{PA}(Y_t) \tag{28}$$

Proof. Based on Lemma C.10, $Y_{t-1} \in \mathbf{M}$ for any $\mathbf{M} \subseteq \mathbf{PA}(Y_t)$ with $s < t - \tau_{\max}$. According to Lemma C.8, given $Y_{t-1} \in \mathbf{M}$, we have $Z_{u-1} \in \mathbf{M}$ for any $Z_u \in \mathbf{PA}(Y_t) \setminus \mathbf{M}$. Therefore, $Z_{u-1} \in \mathbf{PA}(Y_t)$ for any $Z_u \in \mathbf{PA}(Y_t) \setminus \mathbf{M}$, otherwise it contradicts $\mathbf{M} \subsetneq \mathbf{PA}(Y_t)$. □

Theorem C.12 (Construction of Parent Set Superset.)**.** *Let X_s and Y_t be two variables with $s < t$ and $\mathbf{M}$ be one minimal separating set between X_s and Y_t. Under Assumption **B1-B7**, we have:*

$$\textit{If } Y_{t-1} \in \mathbf{M}, \quad \textit{then } \boldsymbol{PA}(Y_t) \subset \mathbf{M} \cup \mathbf{M}_{+1}. \tag{29}$$

where $\mathbf{M}_{+1} \coloneqq \{X_{u+1} : X_u \in \mathbf{M}\}$.

Proof. Based on Lemma C.8, given $Y_{t-1} \in \mathbf{M}$, we have $Z_{u-1} \in \mathbf{M}$ for any $Z_u \in \mathbf{PA}(Y_t) \setminus \mathbf{M}$. Therefore for any $H_v \in \mathbf{PA}(Y_t)$, we either have $H_v \in \mathbf{M}$ or $H_{v-1} \in \mathbf{M}$, and when $H_{v-1} \in \mathbf{M}$, we have $H_v \in \mathbf{M}_{+1}$. Then any variable in $\mathbf{PA}(X_t^j)$ is either in $\mathbf{M}$ or in $\mathbf{M}_{+1}$. □

Theorem C.13 (True Parent Set Discovery)**.** *Let $\widehat{\boldsymbol{PA}}(Y_t)$ be the estimated parent set of any target variable Y_t using the Algorithm MSSD. Under assumptions **B1-B7**, with an oracle (infinite sample size limit) and a consistent conditional independence test, we have that:*

$$\widehat{\boldsymbol{PA}}(Y_t) = \boldsymbol{PA}(Y_t) \tag{30}$$

Proof. By Lemma C.9 and Lemma C.10, a *Lag-Anchored Separating Set* always exists if the testing variable X_s from some *univariate component time series* $\mathbf{X}$, which is not marginally independent of Y_t, satisfies $s < t - \tau_{\max}$. In particular, when $\mathbf{X} = \mathbf{Y}$, such a variable is guaranteed to exist, ensuring the existence of a *Lag-Anchored Separating Set* between Y_s and Y_t for all $s < t - \tau_{\max}$.

Once a *Lag-Anchored Separating Set* is identified by enforcing the inclusion of Y_{t-1} during the separating set search, a superset of the parent set of Y_t can be constructed using Lemma C.12.

Starting from this superset, the algorithm iteratively removes variables by testing whether a candidate variable X_s is d-separated from the target variable Y_t conditional on the remaining variables in the superset. After this

refinement process, any remaining variable $X_s \in \widehat{\mathbf{PA}}(Y_t)$ must satisfy $X_s \in \mathbf{PA}(Y_t)$. We establish this claim through a proof by contradiction, following the same proof as in [Runge et al., 2019].

Assume, for contradiction, that $X_s \in \widehat{\mathbf{PA}}(Y_t)$ but $X_s \notin \mathbf{PA}(Y_t)$. By the contrapositive of the Faithfulness Assumption B3, $X_s \in \widehat{\mathbf{PA}}(Y_t)$ implies that $X_s \not\perp\!\!\!\perp Y_t \,\big|\, \widehat{\mathbf{PA}}(Y_t) \setminus X_s$. Let $\mathbf{W} = \widehat{\mathbf{PA}}(Y_t) \setminus \{\mathbf{PA}(Y_t) \cup X_s\}$. By the Causal Markov Condition B2, we have $\mathbf{W} \cup X_s \perp\!\!\!\perp Y_t \,|\mathbf{PA}(Y_t)$. Then, by the weak union property of conditional independence, it follows that $X_s \perp\!\!\!\perp Y_t \,|\mathbf{PA}(Y_t) \cup \mathbf{W}$, which is equivalent to $X_s \perp\!\!\!\perp Y_t \,\big|\, \widehat{\mathbf{PA}}(Y_t) \setminus X_s$, contradicting the earlier implication. Therefore, we conclude that $\widehat{\mathbf{PA}}(Y_t) = \mathbf{PA}(Y_t)$. □

D Computational Analysis and Runtime Scalability

The computational advantages of the proposed algorithm (MSSD) over PCMCI hold primarily under two conditions: (i) the conditional independence (CI) test is consistent and (ii) the sample size is effectively infinite. For the ideal case with a consistent CI test, most constraint-based algorithms, including PCMCI, must consider up to $n\tau_{\max}$ potential variable pairs per target variable for CI testing in a general n-variate time series with $\tau_{\max}$. In contrast, our method theoretically requires only **one** variable pair per target variable, since a single *Lag-Anchored Separating Set* (LASS) can recover a superset of all its parents. For practical settings with finite samples, additional steps are incorporated into MSSD to enhance robustness, which increases its complexity and makes it comparable to PCMCI. We conducted additional experiments with a simplified, efficient strategy that perform well in the ideal (asymptotic) case but are less robust than the default version with finite samples. See Subsection D.1 for details.

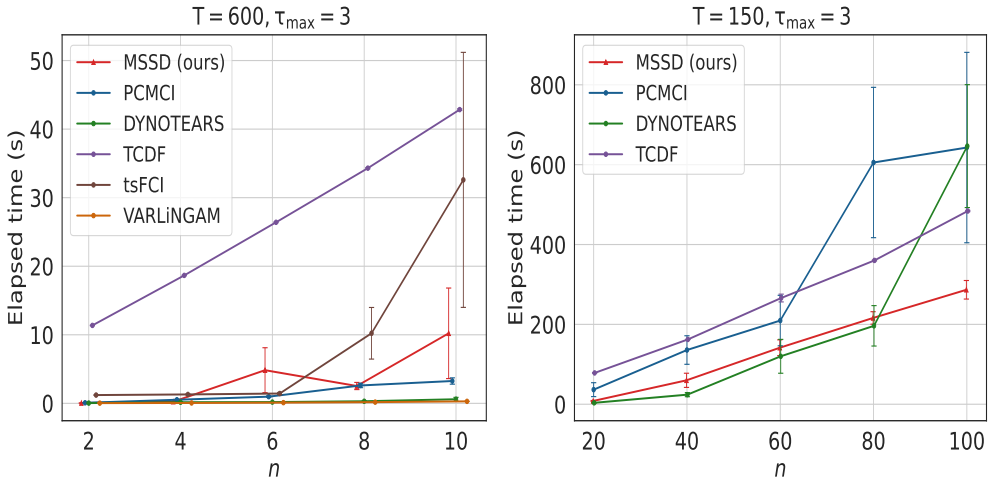
Here are the computational analyses for two scenarios.

Ideal case (consistent CI tests and infinite samples). For each univariate component time series Y, MSSD only needs to test a single pair of variables $\left(Y_{t-\tau_{\max}-1}, Y_t\right)$ to identify the *Lag-Anchored Separating Sets*, which is guaranteed to exist and can be extended to form a superset of the true parent set of Y_t. In contrast, PCMCI requires testing all $N\tau_{\max}$ pairs for each target variable iteratively. Therefore, MSSD has complexity $\mathcal{O}\left(n^2\tau_{\max} + n^2\tau_{\max}\right)$, whereas PCMCI has $\mathcal{O}\left(n^3\tau_{\max}^2 + n^2\tau_{\max}\right)$. The "+" reflects the two stages in both algorithms.

Practical case (finite samples). With limited sample sizes and imperfect CI tests, relying on a single pair to identify the Lag-Anchored Separating Set becomes unstable. To improve robustness, we iterate over variables within the window length $\tau_{\max}$, similar to PCMCI, but apply an early-stopping criterion that terminates the scan for a component time series once a Lag-Anchored Separating Set is identified. At most n Lag-Anchored Separating Sets, each obtained from a different univariate component time series, are then aggregated across these series. In this scenario, the proposed algorithm has complexity $\mathcal{O}\left(n^3\tau_{\max}(\tau_{\max}+1) + n^2\tau_{\max}\right)$, which is comparable to PCMCI's $\mathcal{O}\left(n^3\tau_{\max}^2 + n^2\tau_{\max}\right)$.

Below, we report runtimes for both small n and large n in Fig. 6(a) and Fig. 6(b). In practice, the runtime of the proposed algorithm scales approximately linearly with n and is faster than PCMCI for large n.

Please note that results for VARLiNGAM and tsFCI for large n are unavailable here. VARLiNGAM failed for $n > 20, T = 150$ with Linear SCM due to singular-matrix errors; thus, only the running time for $n = 20$ is reported as 1.183 secs $\pm$ 0.063. A single trial for tsFCI with $n = 20$ requires hours.



(a) Running Time Comparison when n varies for small n.

(b) Running Time Comparison when n varies for large n.

Figure 6: Running time comparison over 50 random trials in (a) and 50 random trials in (b) for different n.

D.1 1-pair Strategy (no scanning) with Finite Sample

While testing a single pair is theoretically more efficient, yielding *quadratic* rather than *cubic* complexity, it tends to increase variance and degrade performance in finite samples. We therefore examine the gap between the simplified strategies (testing only 1 pair or n pairs) and the default setting that scans the lag window with an early-stop criterion (i.e., stopping as soon as a Lag-Anchored Separating Set is found), and assess whether this gap narrows as the sample size grows.

We evaluate two versions of our algorithm, MSSD and MSSD-MCI, under both small sample sizes $T \in \{50, 100, 150\}$ and larger sample sizes $T \in \{800, 1000, 2000\}$. We did not evaluate MSSD-PC because PC inherently requires scanning the full lag window, making it incompatible with the simplified, efficient n-pair and 1-pair strategies.

- **Default:** MSSD scans the entire lag window to identify the *Lag-Anchored Separating Sets*, using an early-stop criterion that improves robustness when samples are limited.
- n **pairs:** For each Y_t^j, test only the pairs $\left(Y^i_{t-\tau_{\max}-1}, Y_t^j\right)$ for $i \in [n]$ to obtain the corresponding *Lag-Anchored Separating Sets*, yielding n pairs in total.
- **1 pair:** For each Y_t^j, test only a single pair, $\left(Y^j_{t-\tau_{\max}-1}, Y_t^j\right)$, producing exactly one corresponding *Lag-Anchored Separating Set*.

As computational complexity decreases from the default setting to the n pairs strategy and then to the 1 pair strategy, performance declines and variance increases as shown in Fig. 7. However, as T grows, both simplified strategies improve steadily. Therefore, with a consistent CI test and sufficient samples, testing a single pair is empirically confirmed to be consistent.

As discussed in the main paper, MSSD favors precision while MSSD-MCI favors recall. The same pattern appears here: across the three MSSD strategies, precision remains similar even with limited samples, and across all sample sizes the MSSD-MCI variants exhibit similarly high recall.

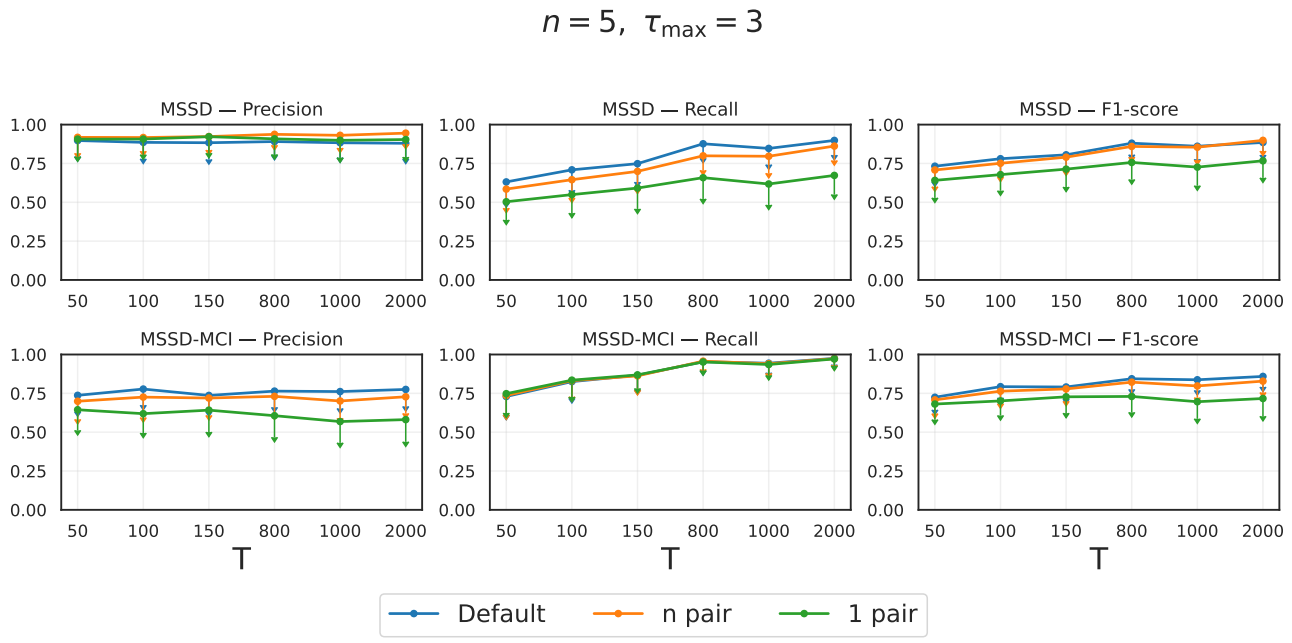


Figure 7: Simplified, efficient strategies of the proposed algorithm with finite-sample

E More Experiments

E.1 Baselines

The baselines include five representative algorithms: PCMCI [Runge et al., 2019], VARLiNGAM [Hyvärinen et al., 2010], DYNOTEARS [Pamfil et al., 2020], TCDF [Nauta et al., 2019] and tsFCI [Entner and Hoyer, 2010]. We also consider a variant of the proposed algorithm in which Stage 1 of Algorithm A.1 is combined with PC stage A2 used in PCMCI, named MSSD-PC; and another variant in which Stage 2 of Algorithm A.1 (starting at Line 20) is replaced by the MCI stage A3 used in PCMCI, named MSSD-MCI.

For all applicable methods, the time lag is set to $\tau_{\text{ub}} = \tau_{\max}$. We use a CI-test significance threshold of 0.05 for all applicable methods, unless a different default is specified by the original method.

For MSSD (the proposed algorithm) and its two variants, all CI tests are from the Tigramite library and hence the same as PCMCI: Partial Correlation test (parcorr) for linear SCMs, Conditional Mutual Information test based on a nearest-neighbor estimator (cmiknn) for nonlinear SCMs, and the G-squared test (gsquared) for categorical data.

Note: To reduce computational cost, for simulations involving nonlinear SCMs in the fMRI benchmark, we configure the cmiknn test to use the `"fixed-threshold"` significance test. Although this setting improves efficiency, it may be less powerful than the default `"shuffle-test"`.

All other hyperparameters for baseline methods are set to the default values recommended in their original implementations.

More specifically, we have:

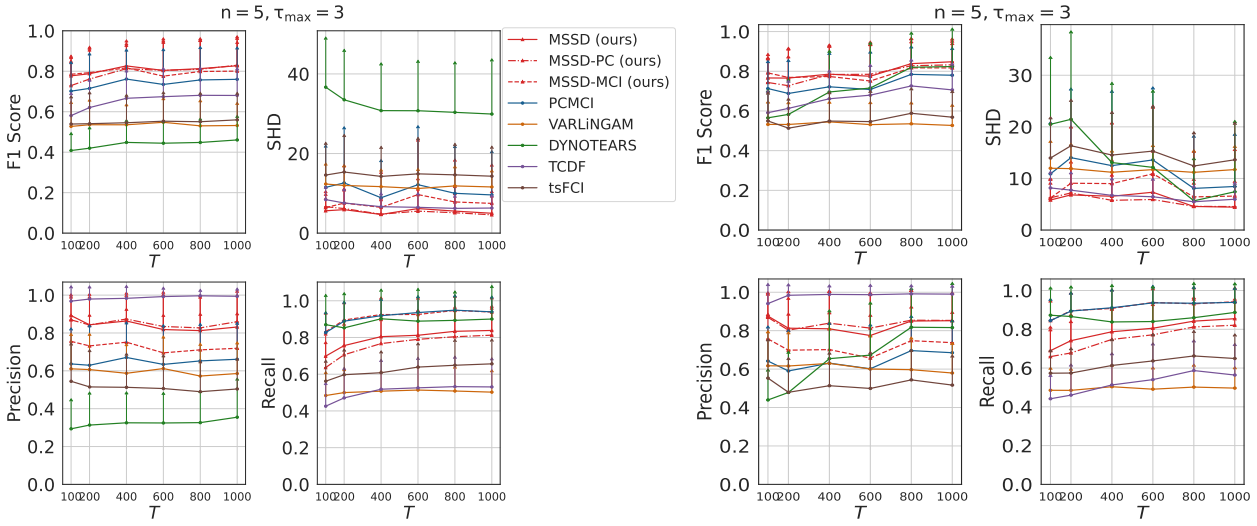
- MSSD: significance level at stage 1=0.2; significance level at stage 2=0.05; $\tau_{\text{ub}} = \tau_{\max}$.
- PCMCI: significance level at PC_1 stage=0.2; significance level at MCI stage 2=0.05; $\tau_{\text{ub}} = \tau_{\max}$. The code is available from `https://github.com/jakobrunge/tigramite/`.
- VARLiNGAM: criterion="bic"; $\tau_{\text{ub}} = \tau_{\max}$. The code is available from `https://github.com/cdt15/lingam/blob/master/lingam/var_lingam.py`.
- DYNOTEARS: $p = \tau_{\text{ub}}$, w-threshold= 0.01, $\lambda_w = 0.05$, $\lambda_a = 0.05$. The code is available from causalnex `https://github.com/mckinsey/causalnex/`.
- TCDF: epochs=2000, hidden layers=1, learning rate=0.01, significance = 0.8, kernel size=$\tau_{\text{ub}} + 1 = \tau_{\max} + 1$, dilation coefficient=$\tau_{\text{ub}} + 1 = \tau_{\max} + 1$. The code is available from `https://github.com/M-Nauta/TCDF/`.
- tsFCI: significance level=0.05; $\tau_{\tau_{\text{ub}}} = \tau_{\max}$. The code is available from Tetrad `https://www.cmu.edu/dietrich/philosophy/tetrad/use-tetrad/tetrad-in-r.html`.

E.2 More Simulations

Here, we extend the simulations from linear SCMs with Gaussian noise to linear SCMs with exponential noise, uniform noise, nonlinear SCMs and discrete-valued time series, respectively.

E.2.1 Linear SCMs with Exponential and Uniform Noise

The results in Fig. 8(a) and Fig. 8(b) demonstrate consistency with the performance observed under linear SCMs with Gaussian noise in the main paper (or previous subsection). Overall, MSSD and MSSD-PC achieve the best F_1 scores and lowest SHD for both exponential and uniform noise settings. MSSD tends to favor precision over recall, whereas MSSD-MCI exhibits the opposite trend. TCDF achieves the highest precision under both noise types, followed by MSSD. In terms of recall, DYNOTEARS, MSSD-MCI and PCMCI perform well.



(a) For Linear SCM with Exponential Noise: performance when T varies.

(b) For Linear SCM with Uniform Noise: performance when T varies.

Figure 8: Eight algorithms are evaluated on 5-dimensional multivariate time series with Linear SCMs and exponential noise in (a) and uniform noise in (b). Each line represents one algorithm, and each marker shows the average performance over 50 random trials with error bars representing the standard deviation. Adjacency F_1 score, SHD, Adjacency Precision, and Adjacency Recall are reported for varying time lengths T with fixed $\tau_{\max} = 3$.

E.2.2 Nonlinear SCMs

As shown in Fig. 9, MSSD-PC performs competitively with PCMCI and TCDF on nonlinear SCMs. However, MSSD and MSSD-MCI are overly conservative.

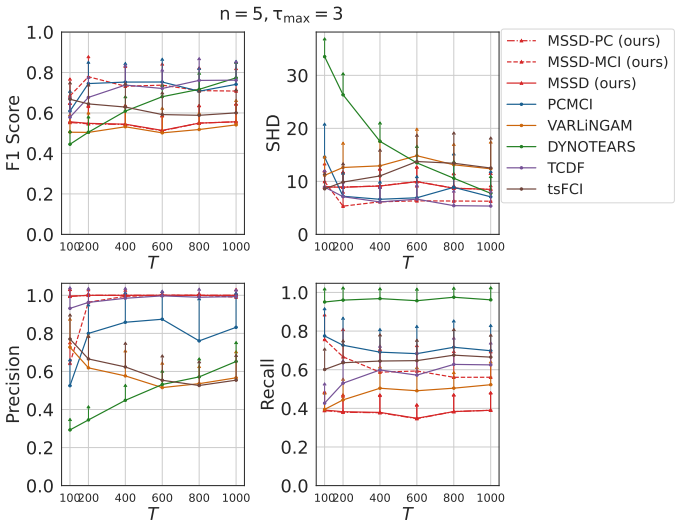


Figure 9: For Nonlinear SCM: performance when T varies. Eight algorithms are evaluated on 5-dimensional multivariate time series with nonlinear SCMs. Each line represents one algorithm, and each marker shows the average performance over 50 random trials with error bars representing the standard deviation. Adjacency F_1 score, SHD, Adjacency Precision, and Adjacency Recall are reported for varying T with fixed $\tau_{\max} = 3$.

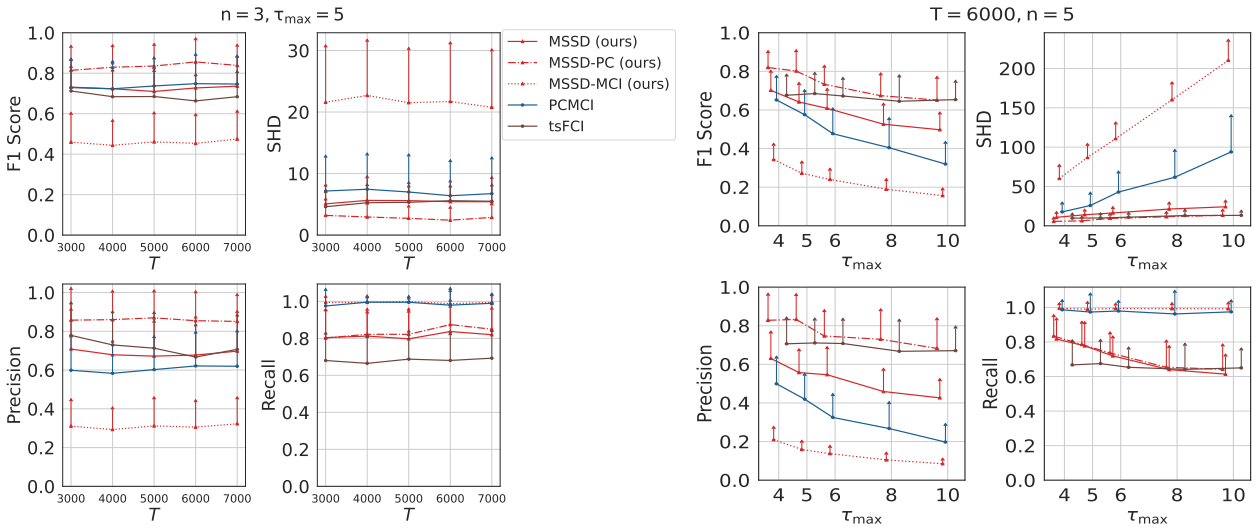
E.2.3 SCMs with Discrete Data

Baselines applicable to discrete-valued time series include PCMCI and tsFCI. As shown in Fig. 10(a), Fig. 10(b), and Fig. 11(a), MSSD-PC attains the highest F1 scores and the lowest SHD, outperforming the other methods across varying T, $\tau_{\max}$, and n.

As T increases, all methods exhibit mildly higher F1 and lower SHD. Relative to other baselines, tsFCI is notably robust across $\tau_{\max}$ in terms of F1, while MSSD-PC also shows strong robustness and accuracy over $\tau_{\max}$. When the number of variables n varies (Fig. 11(a)), MSSD-PC remains consistently strong across n, achieving the best F_1, SHD, and precision, followed by tsFCI and MSSD. By contrast, MSSD-MCI performs worst as T, $\tau_{\max}$, and n vary, primarily due to lower precision; it is also less robust, with performance fluctuating substantially across n.

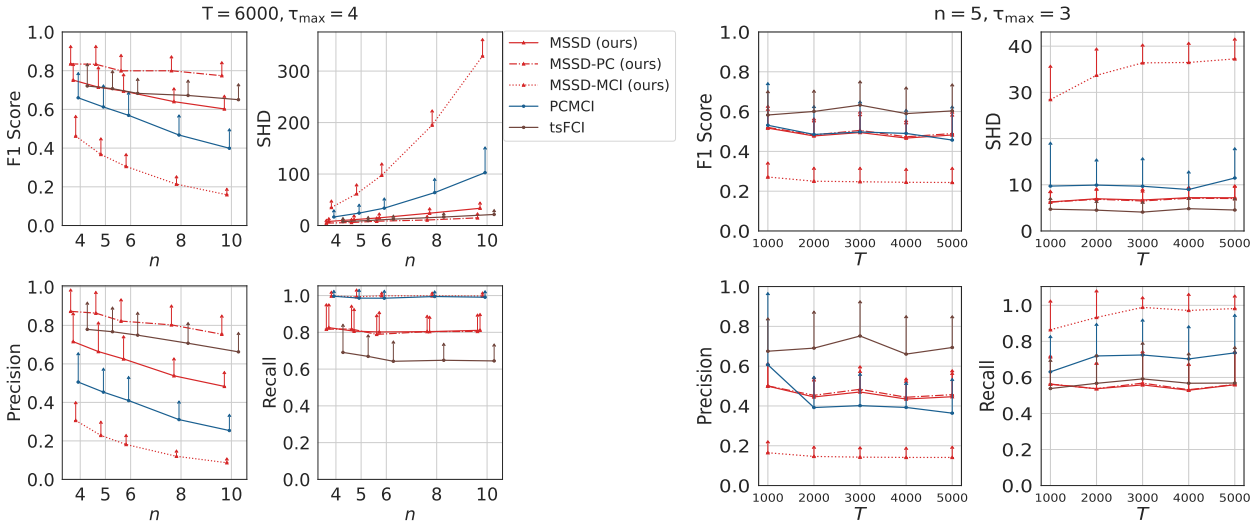
In summary, for binary-valued time series, MSSD-PC is the most accurate and stable, especially at larger $\tau_{\max}$ and T, whereas MSSD-MCI is less suitable in this setting.

We also report results for categorical time series with domain size 10 (Fig. 11(b)). In this setting, tsFCI attains the highest F_1 and the lowest SHD, while MSSD and PCMCI yield very similar F_1 scores, suggesting that the current CI test (G-squared) is not well suited to this dataset.



(a) For binary-valued time series: performance when T varies. (b) For binary-valued time series: performance when $\tau_{\max}$ varies.

Figure 10: Five algorithms are evaluated on n-dimensional binary-valued multivariate time series. Each line represents one algorithm, and each marker shows the average performance over 50 random trials in (a) and 50 random trials in (b) with error bars representing the standard deviation. (a) Adjacency F_1 score, SHD, Adjacency Precision, and Adjacency Recall are reported for varying time lengths T with fixed $n = 3, \tau_{\max} = 5$. (b) The same metrics are reported for varying $\tau_{\max}$ with fixed $T = 6000, n = 5$.



(a) For binary-valued time series: performance when n varies. (b) For categorical time series (domain size 10): performance when T varies.

Figure 11: Five algorithms are evaluated on n-dimensional binary-valued multivariate time series. Each line represents one algorithm, and each marker shows the average performance over 50 random trials in (a) and 50 random trials in (b) with error bars representing the standard deviation. (a) Adjacency F_1 score, SHD, Adjacency Precision, and Adjacency Recall are reported for varying time lengths n with fixed $T = 6000, \tau_{\max} = 4$. (b) The same metrics are evaluated on categorical data (domain size 10).

E.2.4 High-Dimensional Data

In this section, we evaluate performance on high-dimensional data with $n \in \{20, 40, 60, 80, 100\}$ and $T = 150$. The results in Fig. 12 demonstrate the robustness and accuracy of the proposed methods (MSSD and MSSD-PC) in high-dimensional settings with limited samples. TCDF also shows notable robustness. By contrast, the performance of MSSD-MCI and PCMCI is limited in this regime. PCMCI, followed by MSSD-MCI, strongly favors recall at the expense of precision, indicating that its estimated causal graphs are overly dense.

Please note that results for VARLiNGAM and tsFCI are unavailable here. VARLiNGAM failed for $n > 20$ due to singular-matrix errors; thus, only $n = 20$ is reported: Precision = 0.713 (0.113), Recall = 0.408 (0.026), F_1 = 0.516 (0.039), and SHD = 38.67 (5.44). Additionally, due to runtime constraints, tsFCI results are omitted, as a single trial with $n = 20$ requires hours.

We report runtime for small and large n in Fig. 6(a) and Fig. 6(b), highlighting the efficiency of the proposed approach, especially in high-dimensional datasets.

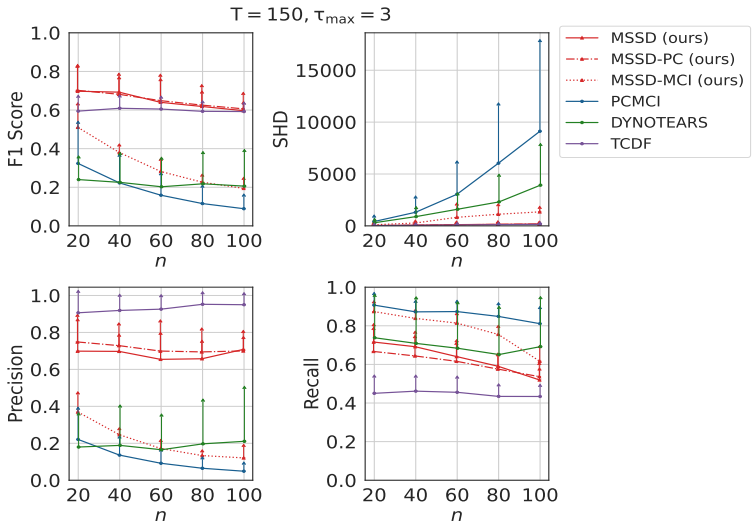


Figure 12: For High-Dimensional Data (Linear SCM with Gaussian Noise): performance when n varies. Six algorithms are evaluated on n-dimensional multivariate time series with linear SCMs and Gaussian Noise. Each line represents one algorithm, and each marker shows the average performance over 50 random trials with error bars representing the standard deviation. Adjacency F_1 score, SHD, Adjacency Precision, and Adjacency Recall are reported with fixed $T = 150, \tau_{\max} = 3$.

E.3 Real-world datasets: CausalRivers with 5-node Random Graph

Table 2: Random subgraphs with 5 nodes. Mean ± Std. Rows ordered by F_1 (descending). Best per column in **bold**, second best underlined.

Method	Precision ↑	Recall ↑	F_1 Score ↑	SHD ↓
DYNOTEARS	**0.339** ± 0.155	0.715 ± 0.283	**0.438** ± 0.162	**7.279** ± 2.877
MSSD	0.279 ± 0.085	0.719 ± 0.203	0.398 ± 0.110	8.861 ± 2.198
MSSD-PC	0.295 ± 0.099	0.636 ± 0.218	0.398 ± 0.125	7.767 ± 2.030
tsFCI	0.274 ± 0.080	0.685 ± 0.204	0.387 ± 0.105	8.745 ± 2.026
PCMCI	0.228 ± 0.048	0.899 ± 0.159	0.361 ± 0.063	13.000 ± 2.589
MSSD-MCI	0.215 ± 0.046	**0.922** ± 0.135	0.347 ± 0.046	14.026 ± 1.888
VARLiNGAM	0.106 ± 0.087	0.296 ± 0.266	0.154 ± 0.127	11.772 ± 2.010

We evaluate on CausalRivers, the real-world time-series benchmark introduced by Stein et al. [2025]. The dataset contains river-discharge measurements (in m^3/s) at 15-minute resolution for eastern Germany (666 stations) and Bavaria (494 stations) from 2019–2023, together with a graph-sampling procedure that yields thousands of subgraphs spanning diverse settings.

The ground truth exploits hydrological physics: upstream discharge causally affects downstream discharge after a delay, and edges encode known flow directions along the river network. In our study, we uniformly sample 450 subgraphs with five nodes (following the benchmark's 3- and 5-node options) and set $\tau_{\max} = 3$ (as in one of the benchmark configurations). We estimate lagged graphs over a length-3 window and then evaluate on the corresponding *summary* causal graphs without explicit lags to match the provided ground truth. (TCDF produced empty graphs or incorrect edges in 42 of 450 subgraphs under these settings and is therefore omitted.)

Across these real-world subgraphs, DYNOTEARS achieves the best Precision, the highest F_1, and the lowest SHD. MSSD ranks next overall, while MSSD-PC attains the second-highest Precision and F_1. MSSD-MCI yields the highest Recall.

E.4 Benchmark: fMRI

fMRI data from [Smith et al., 2011] is a widely used benchmark consisting of realistic simulated datasets spanning diverse brain networks. The benchmark includes 28 datasets with time series lengths ranging from 50 to 5000 and numbers of *univariate component time series* varying from 5 to 50. The number of causal relations varies from 9 to 110.

As noted in the main paper, our method imposes an additional assumption, **B7** (Lag-1 Self-Autoregressive Causality), which other baselines do not share. To ensure a fair comparison, we treat lag-1 self-causal edges as correctly identified for all baselines during evaluation.

Note that the ground truth causal graph underlying the brain networks is a summary (non-time-lagged) causal graph. Although we set the maximum time lag to 3, as in the case study from the main paper, we compress the time-lagged causal graph into a summary graph by merging any time-lagged causal effect into a corresponding instantaneous edge whenever such a relationship exists in the estimated time-lagged causal graph.

A notable case is `timeseries4`, which has length 200, 50 *univariate component time series*, and 110 causal relations—making it much denser than the others. VARLiNGAM fails on this dataset, and all algorithm performs poorly on this dataset, therefore, we report two sets of results: (1) evaluation on all 28 datasets, (2) evaluation excluding `timeseries4`.

We report the average performance metrics along with their standard errors, where the standard error is computed as the standard deviation divided by the square root of the number of datasets. The best-performing value for each metric is highlighted in bold.

From Table 3, TCDF attains the highest Precision, F_1 score, and the lowest SHD, with MSSD achieving the second-best F_1.

From Table 4 (excluding `timeseries4`), TCDF again leads in Precision, F_1, and SHD; VARLiNGAM is second in both F_1 and SHD, followed by MSSD.

Table 3: (1) Performance on 28 fMRI datasets. Mean ± Std. Rows ordered by F_1 (descending). Best per column in **bold**, second best underlined.

Method	Precision ↑	Recall ↑	F_1 Score ↑	SHD ↓
TCDF	**0.961** ± 0.068	0.498 ± 0.036	**0.655** ± 0.034	**9.074** ± 11.411
MSSD	0.697 ± 0.177	0.602 ± 0.121	0.628 ± 0.093	16.630 ± 40.273
tsFCI	0.711 ± 0.150	0.587 ± 0.121	0.627 ± 0.083	15.370 ± 35.517
MSSD-PC	0.700 ± 0.174	0.590 ± 0.115	0.623 ± 0.085	15.778 ± 35.297
MSSD-MCI	0.568 ± 0.144	0.700 ± 0.154	0.605 ± 0.103	27.037 ± 82.328
PCMCI	0.528 ± 0.144	0.739 ± 0.160	0.590 ± 0.091	28.481 ± 80.387
DYNOTEARS	0.401 ± 0.189	**0.944** ± 0.141	0.523 ± 0.141	112.407 ± 424.808

Table 4: (2) Performance on 27 selected fMRI datasets (excluding `timeseries4`). Mean ± Std. Rows ordered by F_1 (descending). Best per column in **bold**, second best underlined.

Method	Precision ↑	Recall ↑	F_1 Score ↑	SHD ↓
TCDF	**0.962** ± 0.069	0.500 ± 0.036	**0.656** ± 0.034	**7.000** ± 3.826
VARLiNGAM	0.806 ± 0.184	0.568 ± 0.094	0.649 ± 0.072	8.038 ± 4.133
MSSD	0.714 ± 0.156	0.606 ± 0.121	0.638 ± 0.075	8.923 ± 4.372
tsFCI	0.726 ± 0.129	0.588 ± 0.123	0.636 ± 0.071	8.577 ± 4.002
MSSD-PC	0.717 ± 0.156	0.594 ± 0.115	0.633 ± 0.068	9.038 ± 4.521
MSSD-MCI	0.584 ± 0.119	0.704 ± 0.155	0.618 ± 0.072	11.231 ± 5.791
PCMCI	0.542 ± 0.126	0.742 ± 0.163	0.603 ± 0.064	13.077 ± 7.563
DYNOTEARS	0.415 ± 0.178	**0.942** ± 0.144	0.539 ± 0.113	31.000 ± 39.868

Since we treat the presence of lag-1 self-autoregressive causality as correct for all baselines, we additionally

Table 5: (3) Performance on 27 selected fMRI datasets (excluding `timeseries4`) across three cases. Mean ± Std. Rows ordered by F_1 (descending). Best per column in **bold**.

Method	**C0** (F1↑)	**C1** (F1↑)	**C2** (F1↑)
MSSD	0.64±0.07	**0.65**±0.07	0.21±0.21
MSSD-PC	0.63±0.07	0.63±0.07	0.22±0.19
MSSD-MCI	0.62±0.07	0.62±0.07	**0.37**±0.16
PCMCI	0.60±0.06	0.60±0.06	0.29±0.15
VARLiNGAM	0.65±0.07	0.62±0.09	0.17±0.16
DYNOTEARS	0.54±0.11	0.54±0.11	0.33±0.14
TCDF	**0.66**±0.03	0.31±0.25	0.03±0.09
tsFCI	0.64±0.07	0.62±0.10	0.17±0.20

consider the following two settings (C1, C2), together with the results reported in Table 4 (C0), to provide a more complete assessment of performance:

(C0) each baseline is evaluated by enforcing correctness on Lag-1 Self-Autoregressive Causality,

(C1) each baseline is evaluated without enforcing correctness on Lag-1 Self-Autoregressive Causality,

(C2) predictive accuracy is evaluated only on non-self edges.

Under both C1 and C2, our algorithms achieve the highest F1 scores, as shown in Table 5, indicating that the earlier performance gap is largely attributable to the evaluation convention.

Another reason our algorithm is not the best on both the fMRI and CausalRivers (5-node) datasets is that their ground truth is a summary graph. Our method benefits from leveraging lag-specific dependencies, but this advantage diminishes when evaluation is done on a summary graph rather than a time-lagged causal graph, though it still achieves the best performance among constraint-based algorithms.

E.5 Quantitative Tables

As the crowded subplots from the paper with many lines make the improvement hard to see clearly, below we provide a quantitative table (Table 6). Each row corresponds to an experimental setting; the numbers indicate algorithm rankings (1 = best among 8 methods, 8 = worst). The total of 8 methods includes our three (MSSD, MSSD-PC, MSSD-MCI) and the five baselines introduced in Appendix Section E.1. Percentages show improvement over the best baseline.

For example, in the discrete-data setting, MSSD-PC ranks first: its average F1 across 50 random trials (aggregated over all values of T) is 13.5% higher, and its SHD is 143.3% less than the strongest baseline (PCMCI). Our methods achieve the best performance across these settings.

The + sign denotes better performance, corresponding to an increase in F_1 and a decrease in SHD.

Table 6: Quantitative Summary Tables across Various Settings

Setting	**Metric**	**MSSD**	**MSSD-PC**	**MSSD-MCI**
Linear (Gaussian Noise)	F1	**1(+7.6%)**	3(+6.2%)	2(+6.7%)
	SHD	**1(+41.8%)**	2(+36.2%)	3(+17.5%)
Linear (Exponential Noise)	F1	**1(+9.2%)**	3(+7.1%)	2(+7.7%)
	SHD	2(+26.6%)	**1(+26.9%)**	4(-8.3%)
Linear (Uniform Noise)	F1	**1(+8.7%)**	3(+6.9%)	2(+7.4%)
	SHD	2(+14.0%)	**1(+18.3%)**	4(-15.9%)
Discrete Data	F1	3(-2.8%)	**1(+13.5%)**	5(-12.7%)
	SHD	2(+22.2%)	**1(+143.3%)**	5(-67.7%)
High-Dimension	F1	2(+9.3%)	**1(+11.4%)**	4(-46.5%)
	SHD	3(-15.8%)	2(-9.1%)	4(-87.4%)
Varying Graph Complexity	F1	2(+16.0%)	3(+13.2%)	**1(+16.7%)**
	SHD	3(-18.7%)	**1(+28.3%)**	4(-26.3%)

More specifically, we also provide a quantitative table for Figure 4(a) to make the performance differences much clearer.

Table 7 is the summary table reporting performance metrics averaged over all n from Figure 4(a), assuming a linear SCM with Gaussian noise.

Table 7: Quantitative Summary Tables averaged over all n (mean ± std) from Figure 4(a)

Method	**Precision ↑**	**Recall ↑**	**F_1 ↑**	**SHD ↓**
MSSD (ours)	0.81±0.02	0.81±0.01	**0.80±0.02**	8.33±1.41
MSSD-PC (ours)	0.83±0.02	0.77±0.02	0.79±0.02	**7.50±1.04**
MSSD-MCI (ours)	0.69±0.03	**0.93±0.01**	0.77±0.02	14.49±3.14
PCMCI	0.63±0.03	**0.93±0.01**	0.72±0.02	19.03±3.93
VARLiNGAM	0.60±0.02	0.50±0.02	0.54±0.02	15.16±1.13
DYNOTEARS	0.50±0.03	0.89±0.02	0.61±0.02	25.52±2.98
TCDF	**0.98±0.01**	0.54±0.02	0.68±0.02	7.98±0.65
tsFCI	0.54±0.03	0.64±0.02	0.57±0.02	19.67±2.00

Table 8 is the table corresponding to Figure 4(a) for each value of n.

Based on the results shown in Table 7 and Table 8, our methods achieve the best overall F_1 score and SHD. The overall F_1 score exceeds that of the strongest baseline (PCMCI) by 11.1%, and the overall SHD improves upon the best baseline (TCDF) by 6.0%. For every value of n, the proposed methods achieve the highest F_1 score.

Based on Table 8, although TCDF has the lowest SHD for $n = 8$ and $n = 10$, this is primarily due to its extremely high precision paired with very low recall, indicating a conservative behavior that produces many false negatives. When considering both precision and recall jointly in terms of the F_1 score, our method consistently outperforms all baselines across different values of n, which aligns with the trends observed in Figure 4(a).

Table 8: Results with different n (mean ± std) from Figure 4(a). Best values are bolded per n within each metric block.

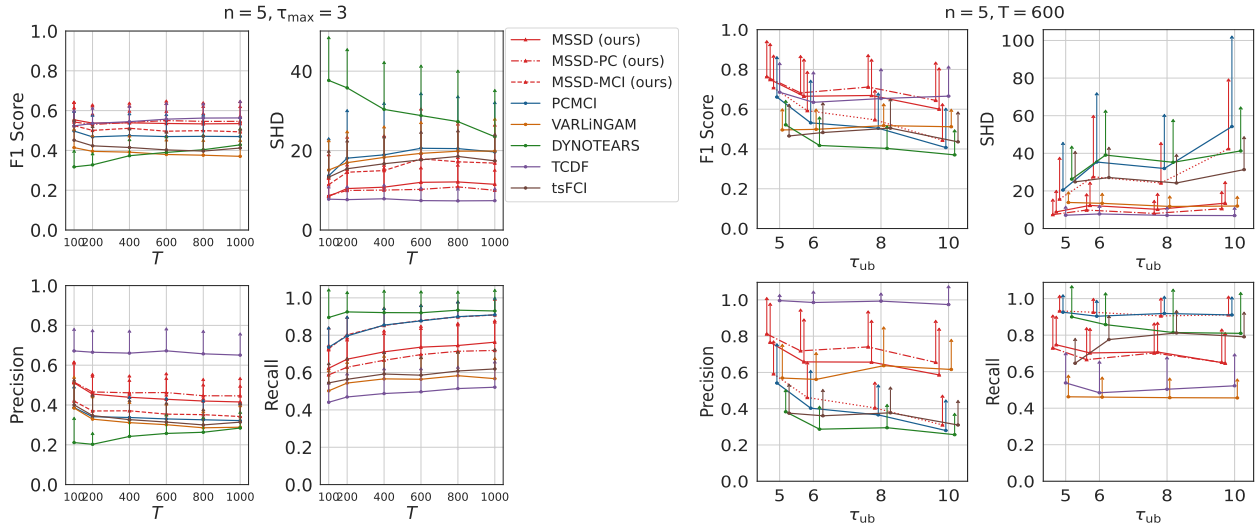
Method	$n = 2$	$n = 4$	$n = 6$	$n = 8$	$n = 10$
			F_1 ↑		
MSSD (ours)	0.86±0.03	0.79±0.03	**0.81±0.04**	**0.77±0.05**	**0.77±0.04**
MSSD-PC (ours)	0.85±0.03	0.79±0.03	0.79±0.04	**0.77±0.04**	**0.77±0.04**
MSSD-MCI (ours)	**0.91±0.03**	**0.80±0.04**	0.78±0.03	0.71±0.05	0.68±0.04
PCMCI	0.87±0.13	0.75±0.03	0.74±0.04	0.67±0.05	0.62±0.04
VARLiNGAM	0.69±0.05	0.54±0.03	0.51±0.03	0.50±0.02	0.48±0.02
DYNOTEARS	0.84±0.03	0.65±0.04	0.61±0.04	0.51±0.03	0.48±0.03
TCDF	0.66±0.05	0.67±0.04	0.71±0.04	0.69±0.03	0.69±0.04
tsFCI	0.75±0.04	0.60±0.03	0.54±0.03	0.49±0.04	0.49±0.03
			Precision ↑		
MSSD (ours)	0.91±0.04	0.82±0.05	0.82±0.05	0.77±0.06	0.74±0.04
MSSD-PC (ours)	0.92±0.04	0.84±0.04	0.84±0.04	0.79±0.05	0.78±0.04
MSSD-MCI (ours)	0.90±0.04	0.73±0.05	0.69±0.04	0.60±0.05	0.55±0.04
PCMCI	0.85±0.04	0.66±0.05	0.63±0.04	0.54±0.05	0.48±0.04
VARLiNGAM	0.77±0.06	0.60±0.04	0.58±0.05	0.56±0.04	0.53±0.04
DYNOTEARS	0.77±0.04	0.54±0.04	0.49±0.04	0.38±0.04	0.35±0.04
TCDF	**1.00±0.00**	**0.98±0.03**	**0.99±0.01**	**0.98±0.02**	**0.95±0.03**
tsFCI	0.83±0.05	0.56±0.04	0.49±0.03	0.43±0.04	0.42±0.03
			Recall ↑		
MSSD (ours)	0.83±0.04	0.79±0.03	0.82±0.03	0.80±0.03	0.82±0.03
MSSD-PC (ours)	0.80±0.04	0.76±0.04	0.76±0.04	0.77±0.04	0.76±0.04
MSSD-MCI (ours)	0.93±0.03	0.91±0.02	**0.94±0.02**	**0.94±0.02**	**0.93±0.02**
PCMCI	0.92±0.03	**0.92±0.02**	0.93±0.02	**0.94±0.02**	0.92±0.02
VARLiNGAM	0.65±0.06	0.51±0.03	0.47±0.03	0.46±0.02	0.45±0.02
DYNOTEARS	**0.96±0.03**	0.90±0.04	0.89±0.05	0.87±0.05	0.86±0.05
TCDF	0.52±0.06	0.53±0.05	0.57±0.04	0.55±0.04	0.55±0.04
tsFCI	0.71±0.05	0.66±0.04	0.61±0.03	0.61±0.03	0.61±0.03
			SHD ↓		
MSSD (ours)	1.74±0.40	5.20±1.19	7.12±1.82	12.40±4.33	14.26±4.04
MSSD-PC (ours)	1.86±0.45	**4.84±1.05**	**6.90±1.46**	10.00±2.40	13.12±3.33
MSSD-MCI (ours)	**1.30±0.45**	6.30±1.84	10.24±3.17	24.10±10.07	28.68±8.92
PCMCI	1.79±0.48	8.26±1.99	12.90±3.53	28.18±9.90	41.60±13.30
VARLiNGAM	3.21±0.52	10.12±1.08	15.60±1.50	19.74±1.67	25.48±1.64
DYNOTEARS	2.30±0.49	12.58±2.51	19.94±3.11	37.46±5.58	52.08±6.91
TCDF	3.14±0.50	5.90±0.90	7.52±0.92	**9.92±1.21**	**12.76±1.66**
tsFCI	2.88±0.50	10.98±1.47	18.16±1.99	28.72±4.06	35.26±4.76

F Ablation Study

We conduct the following ablation studies.

F.1 Violating Assumption A6: Causal Stationarity

We generate non-stationary time series in which *two* distinct causal mechanisms operate, rather than a single invariant mechanism. Figure 13(a) shows that all baselines degrade relative to the stationary case, while TCDF, MSSD, and the two MSSD variants remain comparatively robust.



(a) Ablation Study with non-stationary time series

(b) Ablation Study when $\tau_{\text{ub}} \geq \tau_{\max} = 5$

Figure 13: (a) Eight algorithms are evaluated on non-stationary time series in which *two* distinct causal mechanisms operate, rather than a single invariant mechanism. Each marker shows the average performance over 50 random trials with error bars representing the standard deviation. (b) Eight algorithms are evaluated on 5-dimensional multivariate time series with Linear SCM and Gaussian noise. While $\tau_{\max} = 5$, we set $\tau_{\text{ub}} = 5, 6, 8, 10$ over 50 independent random trials, respectively. Each marker shows the average performance over 50 random trials with error bars representing the standard deviation. Adjacency F_1 score, SHD, Adjacency Precision, and Adjacency Recall are reported with fixed $T = 600, n = 5$.

F.2 Underestimation/Overestimation of the Maximum Time Lag

In earlier experiments, we set $\tau_{\text{ub}} = \tau_{\max}$ for all methods; here we intentionally mismatch them.

Choosing an appropriate maximum time lag $\tau_{\max}$ is crucial for all window-based causal discovery methods for time series (e.g., PCMCI, tsFCI, VARLiNGAM). It is standard practice that $\tau_{\max}$ should be no smaller than the true maximum lag: if $\tau_{\max}$ is underestimated, any causal relation whose true lag exceeds the chosen value cannot be tested, leading to unavoidable false negatives for all algorithms. In contrast, a mild overestimation of $\tau_{\max}$ is typically acceptable, because all true lagged relations remain included in the search space.

To clarify this point, we have conducted experiments evaluating performance where $\tau_{\max}$ is **underestimated** when true $\tau_{\max} = 6$ as shown in Table 9. As expected, all methods, including ours, experience a substantial drop in performance, making comparisons across algorithms uninformative since every method becomes unreliable in this regime. These results indicate that the issue arises from the causal discovery setting itself, not from the algorithms.

Table 9: $\tau_{ub} \leq \tau_{\max}$ with Linear SCM and Gaussian Noise (F_1 score)

Method	$\tau_{ub} = 2$	$\tau_{ub} = 3$	$\tau_{ub} = 4$	$\tau_{ub} = 5$	$\tau_{ub} = 6^*$ (True)
MSSD	0.11±0.08	0.08±0.08	0.10±0.07	0.08±0.07	0.58±0.20
MSSD-PC	0.11±0.09	0.08±0.08	0.08±0.06	0.08±0.07	**0.61±0.21**
MSSD-MCI	0.14±0.09	0.10±0.08	0.13±0.07	0.10±0.07	0.52±0.19
PCMCI	0.15±0.09	0.12±0.08	0.13±0.07	0.13±0.08	0.45±0.20
VARLiNGAM	0.11±0.09	0.09±0.08	0.08±0.07	0.06±0.06	0.46±0.09
DYNOTEARS	**0.16±0.08**	**0.15±0.07**	**0.15±0.05**	**0.16±0.05**	0.39±0.13
TCDF	0.04±0.06	0.03±0.05	0.03±0.05	0.03±0.05	0.56±0.15
tsFCI	0.07±0.07	0.09±0.07	0.11±0.06	0.12±0.06	0.35±0.14

Now we set $\tau_{\text{ub}} \geq \tau_{\max}$ in this ablation study. Figure 13(b) demonstrates that our method retains robustness, although using an incorrect τ_{ub} generally harms performance for *all* algorithms.

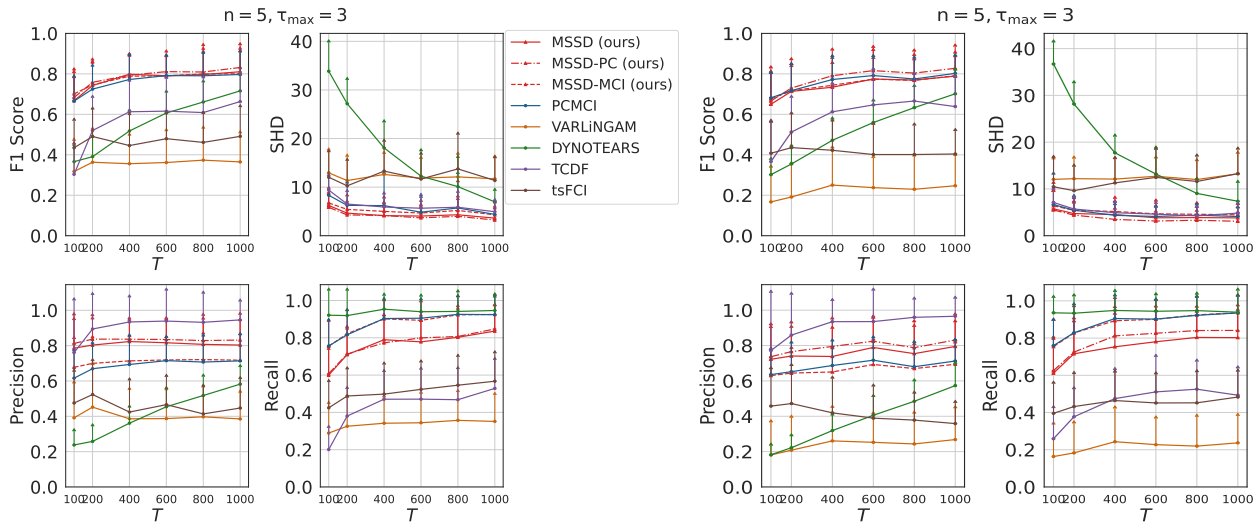
We also included a table version of Figure 13(b) where $\tau_{\max}$ is **overestimated** with true $\tau_{\max} = 5$, and our method remains robust under such misspecification, as shown in Table 10.

Table 10: $\tau_{ub} \geq \tau_{\max}$ with Linear SCM and Gaussian Noise (F_1 score)

Method	$\tau_{ub} = 5^*$ (True)	$\tau_{ub} = 6$	$\tau_{ub} = 8$	$\tau_{ub} = 10$
MSSD	0.75±0.17	0.67±0.17	0.67±0.17	0.60±0.20
MSSD-PC	**0.76±0.17**	**0.68±0.18**	**0.71±0.15**	0.64±0.18
MSSD-MCI	0.71±0.15	0.59±0.19	0.55±0.14	0.44±0.17
PCMCI	0.66±0.19	0.53±0.20	0.50±0.16	0.41±0.19
VARLiNGAM	0.50±0.09	0.50±0.09	0.52±0.10	0.51±0.08
DYNOTEARS	0.52±0.11	0.42±0.13	0.40±0.10	0.37±0.11
TCDF	0.69±0.14	0.63±0.14	0.65±0.14	**0.67±0.14**
tsFCI	0.47±0.15	0.48±0.14	0.50±0.14	0.44±0.14

F.3 Violating Assumption A7: Lag-1 Self-Autoregressive Causality

Instead of enforcing lag-1 self-causality for every *univariate component time series*, we sample its presence independently from a Bernoulli distribution: with probability 0.4 (Fig. 14(a)) and with probability 0.01 (Fig. 14(b)). In this setting, the superiority of our method over PCMCI diminishes; however, the performance of our method and its two variants remains competitive with PCMCI, indicating robustness even without assumption **A7**. Notably, autocorrelation persists in the data, which our method can still take advantage of.



(a) Ablation Study with each Lag-1 Self-Autoregressive Causality exists with probability 0.4

(b) Ablation Study with each Lag-1 Self-Autoregressive Causality exists with probability 0.01.

Figure 14: (a) Eight algorithms evaluated on time series where lag-1 self-autoregressive causality is present with probability 0.4 for each *univariate component time series* in each independent random trial. Algorithms are evaluated on 5-dimensional multivariate time series with Linear SCM and Gaussian noise. Each marker shows the average performance over 50 random trials with error bars representing the standard deviation. Adjacency F_1 score, SHD, Adjacency Precision, and Adjacency Recall are reported. (b) Same setup on time series where lag-1 self-autoregressive causality is present with probability 0.01 for each *univariate component time series* in each independent random trial.

G Data Generation Process

In this section, we outline the procedure used to generate data for the simulation experiments.

Following [Runge et al., 2019], we generate the `continuous-valued time series` in three steps:

1. Generate an n-dimensional multivariate time series $\mathbf{V}$ of length T using independent and identically distributed (IID) noise drawn from a standard Gaussian, exponential, or uniform distribution. Determine the maximum time lag $\tau_{\max}$.
2. For each univariate component time series $\mathbf{X}^j$ where $j \in [n]$, randomly generate:
   - A binary edge matrix of shape $(n, \tau_{\max})$, where a 1 indicates a causal edge and a 0 indicates no edge.
   - A coefficient matrix of the same shape, with entries sampled from a uniform distribution.

   These two matrices jointly define the causal mechanism for each $\mathbf{X}^j$. To ensure the Lag-1 Self-Autoregressive Causality assumption (Assumption B7), we explicitly enforce a lag-1 self-causality in each binary edge matrix.
3. For each time step $t > \tau_{\max}$, compute the vector $\mathbf{X}_t$ using the predefined causal mechanisms and values from past time steps. Repeat this until reaching $t = T$, ensuring that the overall process satisfies the Causal Stationarity assumption (Assumption B6).

Note that the Causal Stationarity assumption in Assumption B6 is distinct from the traditional notion of weak stationarity in time series. The generated values may grow rapidly and diverge, resulting in extreme magnitudes and numerical instability. More specifically, for a large n-variate time series with sufficiently large length T, it is possible to encounter extremely large values caused by explosive growth in the simulated series due to recursive dependencies. For instance, in one simulation with $T = 10{,}000$ and $n = 100$, we observed a maximum value of approximately 1.09×10^{73}. Therefore, we implement a safeguard: if the maximum value in the generated multivariate time series $\mathbf{V}$ exceeds 10^{20}, the data is discarded and regenerated. This filtering strategy is adapted from the procedure in [Gao et al., 2023].

We generate the `discrete-valued time series` following a similar logic. However, instead of generating coefficient matrices in the second step, we construct conditional probability tables (CPTs). The value of each target variable is then sampled according to its corresponding CPT, which is selected based on the configuration of its parent variables from previous time steps.

H Limitations and Future Work

In this section, we will discuss the limitations of the proposed algorithm along with future work.

Limitation 1: Practical Limitations of Path Sensitivity in Superset Discovery

Theoretically, under ideal conditions such as infinite samples and a consistent CI test, a superset of the parent set for each target variable Y_t can be obtained from a single CI test between the variable pair $(Y_{t-\tau_{\max}-1}, Y_t)$. This has been established by fully leveraging the Causal Stationarity structure and Lag-1 Self-Autoregressive Causality.

However, in practice, the correctness of this superset depends on the specific path collection $\mathcal{R}$ described in Lemma C.8. Identifying a *Lag-Anchored Separating Set* accurately between two d-separated variables is generally more challenging than determining whether any two variables can be d-separated by some separating set in practice.

When the information flow between two variables along $\mathcal{R}$ is weak and their time lag is large, a strict subset of the true *Lag-Anchored Separating Set* may empirically suffice to d-separate the variables with finite samples. As a result, the algorithm is modified to take the union of all *Lag-Anchored Separating Sets* obtained from testing across all candidate variables within a window of size $\tau_{\text{ub}} + 1$. This practical adjustment leads to a higher computational cost compared to the ideal case. Please refer to the Computational Analysis section D.

This is also the reason why the algorithm tends to have more false negative edges given finite samples over discrete-valued time series because the CI tests for discrete-valued time series require a large sample size to achieve satisfactory power.

Limitation 2: Assumption B7 Lag-1 Self-Autoregressive Causality

Although robustness under violations of Assumption B7 is demonstrated practically (see Figs. 14(a) and 14(b)), a formal relaxation of Assumption B7 warrants explicit discussion.

As shown in Lemmas C.6 and C.8, the properties of a minimal separating set for the target variable Y_t depend on whether it contains the key variable Y_{t-1}. We have not officially proved whether this key variable can be generalized to $Y_{t-\tau}$ if the minimum self-autoregressive time lag is τ. In other words, it remains an open question whether the Lag-1 Self-Autoregressive Causality assumption can be relaxed to a more general Lag-τ Self-Autoregressive Causality assumption.

Intuitively, this generalization seems feasible. If there is no edge from Y_{t-1} to Y_t, the *univariate component time series* $\mathbf{Y}$ can be viewed as partitioned into multiple segments with Lag-1 Self-causality. For example, under a Lag-2 Self-Autoregressive Causality assumption, where edges exist from Y_{t-2} to Y_t but not from Y_{t-1}, the series $\mathbf{Y}$ can be divided into two interleaved sub-series, each sampled at every second time point (e.g., one starting at $t = 1$ and one at $t = 2$). Each sub-series then satisfies a Lag-1 Self-Autoregressive Causality assumption. However, adopting this generalization would require all *univariate component time series* to share the same minimum self-autoregressive time lag τ, so that the resulting sub-series are of equal length.

Below we discuss two extended scenarios in which Assumption B7 is violated, along with the corresponding theoretical implications.

Scenario 1. All univariate component time series share a larger self-lag.

That is, we assume a **Lag-v Self-Autoregressive Causality** condition and no self-lag smaller than v exists. In this case, our existing theory extends naturally: the LASS of a target variable Y_t becomes any minimal separating set that contains Y_{t-v}. Under this extended definition, the key property of LASS (Lemma C.8 in Appendix or Lemma 2.6 in the main paper) becomes that either Z_δ or $Z_{\delta-v}$ must be included in each LASS, assuming that Z_δ denotes any parent of Y_t. In words, the key lag 1 is simply replaced by key lag v.

Scenario 2. Univariate component time series have different minimum self-lags. For example, the minimum self-lag for Y may be 1 while for X may be 3.

In this scenario, the path structures characterized in Lemma C.8 (Lemma 2.6) no longer exist in a consistent pattern, and thus the property of LASS breaks down. Instead of always containing either a parent or its lag-1 shift, a LASS may need to include the parent itself, or its lag-1 version, or its lag-2 version and so on, depending jointly on the minimum self-lags of the target and source variables, as well as the minimum interaction-lag between them.

Because of this variability, LASS becomes graph-dependent and no longer exhibits the unified lag-1 structure that enables our theoretical guarantees.

Future Work: Generalizing Beyond Current Assumptions

Without assuming the Sufficiency Assumption B1, a promising direction for future work is to extend the proposed algorithm to account for hidden confounders, thereby enhancing its applicability to real-world scenarios, such as [Entner and Hoyer, 2010, Gerhardus and Runge, 2020]. Additionally, as previously discussed, the Lag-1 Self-Autoregressive Causality Assumption B7 can be generalized to a lag-τ Self-Autoregressive Causality assumption, offering greater modeling flexibility. While the algorithm assumes Causal Stationarity B6, it may be extended to segmentwise stationary settings, as in [Gao et al., 2023, 2025].