Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Global Constraint-Based Causal Structure Learning with Streaming Features

Jose M. De Miguel

jose.demiguel@alumnos.upm.es

Universidad Politécnica de Madrid

Concha Bielza

Universidad Politécnica de Madrid

Pedro Larrañaga

Universidad Politécnica de Madrid

Research Article

Keywords: causal discovery, streaming features, constraint-based causal discovery, asymptotic consistency, Bayesian networks, fast causal inference (FCI)

Posted Date: October 7th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10437555/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Global Constraint-Based Causal Structure Learning with Streaming Features

José M. De Miguel 0009-0008-4238-5240[1*], Concha Bielza 0000-0001-7109-2668[1†] and Pedro Larrañaga 0000-0003-0652-9872[1†]

[1]Computational Intelligence Group, Universidad Politécnica de Madrid, Madrid, Spain.

*Corresponding author(s). E-mail(s): jose.demiguel@alumnos.upm.es;
Contributing authors: mcbielza@fi.upm.es; pedro.larranaga@fi.upm.es;
†These authors contributed equally to this work.

Abstract

With the ever-growing dimensionality of data and the rapid development of new data mining techniques, causal discovery faces both new challenges and opportunities. One such challenge is the streaming features setting, in which new features arrive over time, as in social networks incorporating real-time news. Current methods for streaming features are predominantly local and constraint-based. However, they can be computationally expensive and lack formal asymptotic consistency guarantees. We introduce FCI-SF, a global feature-wise incremental extension of Fast Causal Inference (FCI) and, to the best of our knowledge, the first constraint-based algorithm for streaming features with formal asymptotic consistency guarantees. FCI-SF updates the causal structure whenever a new feature batch arrives and exploits a new theoretical result—the *independence preservation property*—to reuse earlier CI information while preserving recovery of the correct Markov equivalence class in the large-sample limit. Synthetic experiments show that FCI-SF achieves among the highest graphical accuracies while being nearly 20 times faster than competing streaming features methods. Across nine real-world datasets without known ground truth, it obtains the best average rank in graphical self-compatibility, and its incremental analysis of the *Arabidopsis thaliana* isoprenoid network recovers structures broadly consistent with established biological pathways. We also release an open-source Python package implementing FCI-SF and the main causal discovery methods for streaming features.

Keywords: causal discovery, streaming features, constraint-based causal discovery, asymptotic consistency, Bayesian networks, fast causal inference (FCI)

1 Introduction

Discovering the causal mechanism governing an industrial or scientific process has always attracted attention due to its numerous practical applications, such as understanding whether the behavior of a component causes failure in industrial machinery or determining if a treatment causes an improvement in a patient's condition. Often, learning causal relationships through randomized controlled trials is infeasible or impossible. Consequently, under certain assumptions (Spirtes et al. 2000; Pearl 2009), causal discovery becomes an essential tool for learning cause-and-effect relationships from observational data, providing insights beyond mere correlation and having numerous applications in fields such as genomics and decision-making systems.

In recent years, the machine learning community has devoted increasing attention to settings in which the set of variables describing a system expands progressively as new measurements or feature representations become available.This scenario is commonly referred to as streaming features (SF). The SF setting arises, for example, when it is computationally expensive to generate the complete feature space, as in image processing (Ding et al. 2011; Jia et al. 2013), or when new measurements become available over time, as in social network analysis (Li et al. 2015) and long-lead flood prediction (Wang et al. 2013). Consequently, causal structure learning algorithms must update their models without waiting for the complete feature set or repeating the entire learning process whenever new features arrive.

Constraint-based methods constitute one of the principal paradigms in causal discovery and have motivated extensive research over several decades (Verma and Pearl 1990; Spirtes et al. 2000; Harris and Drton 2013; Colombo and Maathuis 2014; Li et al. 2015; Cui et al. 2016; Le et al. 2016; Sondhi and Shojaie 2019; Zarebavani et al. 2020; Giudice et al. 2022; Tsamardinos et al. 2003; Aliferis et al. 2010). These methods recover causal structure from observational data by testing conditional independence (CI) relations. They include local approaches, which recover the causal structure by learning the neighborhood of one variable at a time (Tsamardinos et al. 2003, 2006), and global approaches, which may modify adjacencies between any pair of observed features during structure learning. Among the latter, the PC algorithm (Spirtes and Glymour 1991) assumes causal sufficiency, meaning that the observed variables have no unobserved common causes. The fast causal inference (FCI) algorithm (Spirtes et al. 1995) relaxes this assumption and accommodates both latent confounding and selection effects.

Traditional constraint-based algorithms assume a fixed set of observed variables, requiring the causal structure to be relearned from scratch whenever a new batch of features arrives and thus making them unsuitable for SF scenarios. This limitation has recently prompted considerable interest in causal discovery under SF, motivating the development of several feature-wise incremental causal discovery algorithms (Yu et al. 2010, 2012; Guo and Yang 2017; Yang et al. 2018, 2023; Zhang et al. 2025). However, except for the recent method of Zhang et al. (2025), these approaches assume causal sufficiency and therefore do not explicitly account for latent confounding. This assumption is particularly restrictive in the SF setting, since a common cause that is initially unobserved may become observable only at a later stage. Moreover, to the best of our knowledge, existing approaches rely on local search strategies and lack

asymptotic consistency guarantees, leaving the recovery of the correct causal structure theoretically unresolved even under perfect CI information in the large-sample limit.

To address these limitations, we propose FCI-SF, an incremental global constraint-based causal discovery algorithm for SF. Building upon FCI, the proposed method updates the learned graph whenever a new batch of features arrives, reusing previously inferred structural information to reduce the number of CI tests, which constitute the primary computational bottleneck in constraint-based methods. We prove the asymptotic consistency of FCI-SF under standard assumptions; the complete theoretical analysis and mathematical proofs are provided in Appendix A. We also empirically demonstrate substantial improvements in execution time over existing state-of-the-art methods for causal discovery under SF. Our contributions are summarized as follows:

- We leverage the mathematical framework of CI models and maximal ancestral graphs (MAGs) (Richardson and Spirtes 2002) to model CIs in SF settings and establish the *independence preservation property*, showing that edges removed in earlier stages remain absent as new variables become observable.
- We develop FCI-SF, a feature-wise incremental global extension of the FCI algorithm for SF, and prove its asymptotic consistency.
- We conduct a systematic empirical evaluation on both synthetic and real-world datasets, showing that FCI-SF achieves substantial improvements in execution time compared to existing SF causal discovery methods.
- We release a publicly available Python package that implements FCI-SF and all existing state-of-the-art causal discovery methods designed for the SF setting, facilitating reproducibility, systematic comparison, and future research[1].

The remainder of this work is organized as follows. Section 2 reviews related work on conventional constraint-based causal discovery and causal discovery in the SF setting. Section 3 introduces the necessary terminology and formal concepts. Section 4 presents the original FCI algorithm and establishes the foundation for the proposed FCI-SF algorithm, which is formally described in Section 5. Section 6 evaluates the performance of FCI-SF on synthetic and real-world datasets. Finally, Section 7 concludes the paper. The propositions and supporting mathematical results required to establish the asymptotic consistency of FCI-SF are provided in Appendix A.

2 Related Work

2.1 Traditional Constraint-Based Causal Discovery

Within the family of local constraint-based algorithms, several strategies have been proposed to recover the causal structure by identifying candidate Markov blankets—minimal sets of variables that render a target variable CI of all remaining variables—and orienting edges among the selected variables. Some methods, such as the Grow-Shrink algorithm (Margaritis and Thrun 1999), first learn the Markov

[1]Code available at https://github.com/ComputationalIntelligenceGroup/FCI-SF.

blankets of the variables and then perform additional CI tests to determine edge orientations. In contrast, other approaches, such as the Max-Min Hill-Climbing (MMHC) algorithm (Tsamardinos et al. 2006), construct the skeleton using local structures and subsequently orient the edges through a hill-climbing procedure guided by a Bayesian Dirichlet equivalent uniform (BDeu) score. While the Grow-Shrink and MMHC algorithms rely on the causal sufficiency assumption, some local algorithms relax this assumption, such as the Markov blanket/collider set (MBCS*) algorithm (Pellet and Elisseeff 2008), or the algorithm for latent local causal discovery, LatentLCD (Ling et al. 2025).

Among global constraint-based methods, the PC algorithm (Spirtes and Glymour 1991) is one of the most prominent. Under suitable assumptions, it is asymptotically consistent, even in high-dimensional settings, meaning that the probability of recovering the correct Markov equivalence class converges to one as the sample size increases (Kalisch and Bühlman 2007). Variants such as PC-stable (Colombo and Maathuis 2014) improve its practical robustness by making skeleton discovery invariant to the ordering of the input variables. However, the PC algorithm assumes causal sufficiency and the absence of selection bias, assumptions that are often violated when unobserved common causes induce dependencies among observed variables. The FCI algorithm (Spirtes et al. 1995, 2000) addresses these limitations by accommodating latent confounding and selection effects and returning a partial ancestral graph (PAG). The FCI algorithm is reviewed in detail in Section 4.

2.2 Causal Discovery with SF

Although a growing body of work addresses incremental causal discovery in streaming settings, existing methods are fundamentally local in nature. Yu et al. (2010, 2012) introduced the symmetrical causal discovery from streaming features (S-CDFSF) algorithm, a two-phase procedure that learns the skeleton of a DAG as new features arrive. For each incoming feature, the algorithm first performs a forward phase to add candidate adjacencies to the new variable, followed by a backward phase that removes false positives. However, it only recovers the skeleton and does not subsequently orient the edges. Guo and Yang (2017), Yang et al. (2018), and Yang et al. (2023) proposed several related approaches, namely the causal structure learning based on streaming features (CSBS) algorithm, the causal structure learning with symmetrical uncertainty (CSSU) algorithm, and the partial rank casual discovery streaming feature based (PRCDSF) algorithm, respectively. These methods begin with a relevance analysis step that adds candidate adjacencies to the newly arrived feature. A redundancy analysis phase then prunes edges from the Markov blankets of the new feature and its adjacent variables. Once all features have been processed, the resulting skeleton is oriented using a greedy hill-climbing search guided by the minimum description length (MDL) score to produce a final directed acyclic graph (DAG).

However, these algorithms assume causal sufficiency and thus cannot handle latent confounding. LiNGAM-SF (Zhang et al. 2025) extends this line of work by incorporating latent variable detection through overcomplete independent component analysis of residuals (Salehkaleybar et al. 2020), incrementally refining the learned structure

under linear non-Gaussian assumptions and ultimately outputting a DAG augmented with the detected latent variables.

Despite these advances, important limitations remain. First, no existing SF algorithm provides a formal asymptotic consistency guaranty under perfect CI information, and thus it is not guaranteed to recover the correct graph equivalence class even with perfect CI tests. Second, most approaches assume causal sufficiency—an unrealistic assumption in streaming settings where new confounders may appear over time. Third, although LiNGAM-SF considers latent variables, it relies on linear non-Gaussian structural equation models, which limits its applicability. Finally, the computational advantages of current SF methods remain uncertain, as repeated local updates may approach the cost of rerunning global constraint-based algorithms (e.g., FCI) after each feature update. This motivates the need for a principled incremental strategy with both theoretical guarantees and computational efficiency. Motivated by these limitations, we introduce the FCI-SF algorithm, the first global incremental constraint-based algorithm for SF that supports latent confounding and proves asymptotic consistency (Appendix A). Table 1 compares the state-of-the-art with the proposed FCI-SF.

Table 1: Comparison of existing causal discovery algorithms for SF and the proposed method.

Algorithm	**Type**	**Latent Variables**	**Output**	**Asymptotic Consistency**
S-CDFSF (Yu et al. 2010, 2012)	Local	No	Skeleton	No
CSBS (Guo and Yang 2017)	Local	No	DAG	No
CSSU (Yang et al. 2018)	Local	No	DAG	No
PRCDSF (Yang et al. 2023)	Local	No	DAG	No
LiNGAM-SF (Zhang et al. 2025)	Local	Yes	DAG + latents	No
Proposed FCI-SF	Global	Yes	PAG	Yes

3 Formal Framework

Let $\mathcal{G} = (V, E)$ be a causal graph with vertex set V and edge set E. We consider two types of edges. A directed edge $X \to Y$ indicates that X is a direct cause of Y; in this case, X is called a parent of Y, and the set of parents of Y is denoted by $Pa(Y)$. A bidirected edge $X \leftrightarrow Y$ represents the presence of a latent common cause affecting both X and Y, implying that neither variable is a direct cause of the other. We use a circle at an edge endpoint to indicate that its orientation is undetermined. For instance, the edge $X \circ\!\!-\!\!\circ Y$ in the skeleton of a graph denotes adjacency between X and Y without specifying direction, meaning that the edge may ultimately be oriented as $X \to Y$, $Y \to X$, or $X \leftrightarrow Y$, and the edge $X \circ\!\!\to Y$ could be oriented either as $X \to Y$ or $X \leftrightarrow Y$. The set of adjacencies of a feature X is denoted as $Adj(X)$.

If there exists a directed path (containing only directed edges) of the form $X \to \cdots \to Y$, then X is called an ancestor of Y. The set of ancestors of Y is denoted

by $An(Y)$. A *triangle* is a triple (X, Y, Z), where all the vertices are adjacent (e.g., (X_2, X_3, X_4) in Fig. 1c). A triple of variables (X, Y, Z) forming a path $X - Y - Z$ is called an *unshielded triple* if X and Z are not adjacent (e.g., (X_1, X_4, X_3) in Fig. 1c). If there is a path $X - Y - Z$ where both X and Z have edges pointing into Y (e.g., $X \to Y \leftrightarrow Z$), we call the middle vertex Y a collider; unshielded triples whose middle vertex is a collider are called v-structures. An *inducing path* with respect to a feature set $\mathbf{H}$ is a path in which every feature not in $\mathbf{H}$ is a collider and an ancestor of at least one of the endpoints. If $\mathbf{H}$ is not specified, we assume $\mathbf{H} = \emptyset$. For example, in Fig. 1a the path $X_1 \leftrightarrow X_2 \leftarrow Y \to X_4$ is an inducing path with respect to $\mathbf{H} = \{Y\}$.

The causal discovery problem consists of learning the underlying causal graph from observational data. The goal is to recover either a DAG, which contains only directed edges, or a MAG, which may contain directed and bidirected edges and has no directed cycles. In a MAG, if $X \leftrightarrow Y$ is present, then neither X is an ancestor of Y nor Y is an ancestor of X. Moreover, a MAG is *maximal* by definition, which means that any two vertices connected by an inducing path with respect to $\mathbf{H} = \emptyset$ must be adjacent. A DAG is a special case of a MAG.

3.1 CI Models

This section introduces CI models as a collection of CI statements, which may come from a distribution; P, in that case, the CI model will be a distributional CI model, denoted as $\mathcal{I}(P)$. Alternatively, it may come from a causal graph, $\mathcal{G}$, in which case we call it a graphical CI model, denoted $\mathcal{I}(\mathcal{G})$. When we want to refer to a general CI model, which could be either distributional or graphical over a feature set V, we will simply denote it as $\mathcal{I}_V$. Let P denote a joint probability distribution over V. For disjoint subsets $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq V$, we write

$$\mathbf{X} \perp_P \mathbf{Y} \mid \mathbf{Z},$$

to denote that $\mathbf{X}$ and $\mathbf{Y}$ are CI given $\mathbf{Z}$ with respect to P, i.e.,

$$P(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z}) = P(\mathbf{X} \mid \mathbf{Z})\, P(\mathbf{Y} \mid \mathbf{Z}).$$

The distributional CI model of P is the collection of CI statements:

$$\mathcal{I}(P) = \{\mathbf{X} \perp_P \mathbf{Y} \mid \mathbf{Z} \text{ s.t. } \mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq V \text{ are pairwise disjoint and } \mathbf{X}, \mathbf{Y} \neq \emptyset\}. \quad (1)$$

Since the data-generating distribution P is generally unknown, the CI relations in $\mathcal{I}(P)$ must be estimated from a dataset $\mathcal{D}$ sampled from P using a CI test $\mathcal{T}_\mathcal{D}$. For any pair of variables $X, Y \in V$ and conditioning set $\mathbf{Z} \subseteq V \setminus X, Y$, let $\mathcal{T}(X, Y \mid \mathbf{Z})$ denote the p-value for the null hypothesis $X \perp_P Y \mid \mathbf{Z}$. The variables X and Y are treated as CI given $\mathbf{Z}$ whenever the null hypothesis of CI cannot be rejected at a significance level α, that is,

$$\mathcal{T}_\mathcal{D}(X, Y \mid \mathbf{Z}) > \alpha.$$

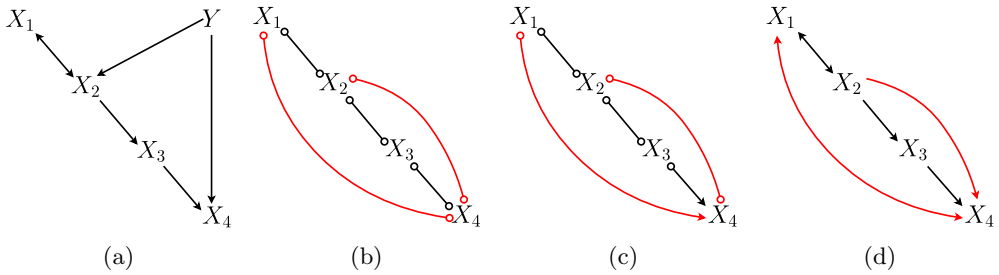


Fig. 1: Graphs (a) and (d) with directed and bidirected edges, (b) is the skeleton of (d) and (c) is the skeleton after adding the orientations of the only unshielded collider (X_1, X_4, X_3)

MAGs encode CI relations through a graphical separation criterion called *m-separation*, which generalizes the well-known notion of *d-separation* for DAGs (Pearl 2009).

Definition 1 (m-separation (Richardson and Spirtes 2002)) Let $\mathcal{G}$ be a MAG over variables V, and let $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq V$ be disjoint sets. A path between a variable $X \in \mathbf{X}$ and a variable $Y \in \mathbf{Y}$ is *blocked* by $\mathbf{Z}$ if either

- the path contains a non-collider that belongs to $\mathbf{Z}$, or
- the path contains a collider that is not an ancestor of any variable in $\mathbf{Z}$

If every path between $\mathbf{X}$ and $\mathbf{Y}$ is blocked by $\mathbf{Z}$, then $\mathbf{X}$ and $\mathbf{Y}$ are said to be *m-separated* given $\mathbf{Z}$ in $\mathcal{G}$, denoted $\mathbf{X} \perp_m \mathbf{Y} \mid \mathbf{Z}$. Otherwise, they are said to be *m-connected*. This induces the graphical CI model associated with $\mathcal{G}$:

$$\mathcal{I}(\mathcal{G}) = \{\mathbf{X} \perp_m \mathbf{Y} \mid \mathbf{Z} \text{ s.t. } \mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq V \text{ are pairwise disjoint and } \mathbf{X}, \mathbf{Y} \neq \emptyset\}. \quad (2)$$

3.2 Graphical Equivalence Classes

Multiple causal graphs may encode the same CI model. Such graphs are said to belong to the same Markov equivalence class, which we denote by $[\mathcal{G}]$. All graphs within a Markov equivalence class share the same adjacencies and differ only in the orientation of some edge endpoints. Consequently, it is possible to represent the entire equivalence class by a single graphical object. For MAGs, the Markov equivalence class is represented by a PAG, which is the graphical representation used by FCI-SF.

Definition 2 (PAG) Given the Markov equivalence class of a MAG $\mathcal{G}$, denoted $[\mathcal{G}]$, a partial ancestral graph (PAG) $\mathcal{P}$ representing $[\mathcal{G}]$ is a graphical structure whose edges may have endpoint marks in $\{\circ, -, >\}$. The five possible edge types are: $\rightarrow$, $\leftrightarrow$, $\circ\!\!-\!\!\circ$, $\circ\!\!\rightarrow$.

These satisfy:

(i) $\mathcal{P}$ has the same adjacencies as any member of $[\mathcal{G}]$; and

(ii) every non-circle mark is invariant across all MAGs in $[\mathcal{G}]$, i.e., all MAGs in $[\mathcal{G}]$ share that exact mark at the corresponding endpoint.

3.3 Latent Projection

In the scenario of SF, the full set of variables is not available from the outset. The CIs that can be observed from a subset of variables are only a projection of the CIs in the full feature space. This operation is known as *latent projection* (Verma 1993). Let V be a set of variables, and let $\mathcal{I}_V$ denote a CI model, either distributional (Eq. (1)) or graphical (Eq. (2)) over V, consisting of CIs of the form $\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}$. Let $\mathbf{O} \subseteq V$ be the set of observed variables. The latent projection of $\mathcal{I}_V$ onto $\mathbf{O}$ is the restricted independence model

$$\mathcal{I}_{\mathbf{O}} = L(\mathcal{I}_V, \mathbf{O}) = \{\, (\mathbf{X} \perp \mathbf{Y} \mid \mathbf{Z}) \in \mathcal{I}_V \text{ s.t. } \mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{O} \,\}. \tag{3}$$

That is, we only consider CIs involving variables contained in $\mathbf{O}$. Intuitively, latent projection describes the CIs that remain among the observed variables after marginalizing out the hidden variables $\mathbf{H} = V \setminus \mathbf{O}$. At the distributional level, the latent projection onto $\mathbf{O}$ is given by marginalization:

$$P(\mathbf{O}) = L(P(V), \mathbf{O}) = \sum_{\mathbf{H}} P(\mathbf{O}, \mathbf{H}).$$

At the graphical level, the latent projection $L(\mathcal{G}, \mathbf{O})$ maps a MAG $\mathcal{G}$ onto an observed feature set $\mathbf{O}$ by removing the remaining variables while preserving the CI relationships among the retained vertices. For example, projecting the graph in Fig. 1a onto $\mathbf{O} = X_1, X_2, X_3, X_4$ yields the graph shown in Fig. 1d. A key property in the SF setting is that latent projection commutes with CI model construction for both distributions (Eq. (1)) and graphs (Eq. (2)): projecting first and then deriving the CI model is equivalent to deriving the CI model first and then projecting. In particular, for a graphical CI model,

$$\mathcal{I}(L(\mathcal{G}, \mathbf{O})) = L(\mathcal{I}(\mathcal{G}), \mathbf{O}),$$

and the analogous equality holds for distributional CI models.

4 The Traditional FCI Algorithm

The FCI algorithm proceeds in five phases. The first three phases iteratively remove edges from a complete undirected graph using CI tests, producing an undirected skeleton. The last two phases orient edges to obtain a PAG. CI tests are performed between pairs of variables given conditioning sets $\text{Sep}[X, Y]$, i.e., tests of the form $X \perp Y \mid \text{Sep}[X, Y]$, where $\text{Sep}[X, Y]$ is referred to as the *separating set* between X and Y. To keep the notation simple, we will consider that separating sets are symmetric (i.e., $\text{Sep}[X, Y] = \text{Sep}[Y, X]$) for every pair of variables X and Y. Initially, separating sets are searched within the adjacency sets of each pair of variables. After this stage, partial orientations are introduced, and the algorithm performs additional CI tests where conditioning sets are drawn from the **Possible-D-SEP** set of a variable.

Formally, a variable X_k belongs to **Possible-D-SEP**(X_i) in a PAG $\mathcal{P}$ if and only if $X_k \neq X_i$ and there exists a path between X_i and X_k such that every non-endpoint

node on the path is either a collider or part of a triangle in $\mathcal{P}$. For example, in Fig. 1c, **Possible-D-SEP**$(X_1) = \{X_2, X_3, X_4\}$. Since X_2 and X_4 are adjacent to X_1, each is connected to X_1 by a path with no non-endpoint vertices and therefore satisfies the definition. Moreover, X_3 belongs to **Possible-D-SEP**(X_1) because the only non-endpoint vertex on the path $X_1 \circ\!\!\rightarrow X_4 \leftarrow\!\!\circ X_3$ is the collider X_4. By definition, $\text{Adj}(X_i) \subseteq$ **Possible-D-SEP**(X_i); therefore, to search for separating sets within **Possible-D-SEP**(X_i), FCI does not need to reconsider sets contained entirely in $\text{Adj}(X_i)$, as these have already been examined during the initial adjacency phase. Here we present an overview of the FCI algorithm, where separating sets are symmetric:
Require: Dataset $\mathcal{D}$ over feature set V and CI test $\mathcal{T}_\mathcal{D}$ with significance level α
Ensure: Maximally informative PAG $\mathcal{P}$

Phase 1: Start with the complete graph. For each pair of adjacent variables X and Y, check if there exists an $\text{Sep}[X, Y] \subseteq \mathit{Adj}(X) \setminus \{Y\}$ or $\text{Sep}[Y, X] \subseteq \mathit{Adj}(Y) \setminus \{X\}$ such that $X \perp_P Y \mid \text{Sep}[X, Y]$; then prune the edge (X, Y).
Phase 2: Orient unshielded triples as described in Chapter 6.7 of Spirtes et al. (2000).
Phase 3: For each pair of adjacent variables X and Y, check if there exists an $\text{Sep}[X, Y] \subseteq$ **Possible-D-SEP**$(X) \setminus \{Y\}$ or $\text{Sep}[Y, X] \subseteq$ **Possible-D-SEP**$(Y) \setminus \{X\}$ such that $X \perp_P Y \mid \text{Sep}[X, Y]$; then prune the edge (X, Y). Eliminate all orientations.
Phase 4: Orient unshielded triples as described in Chapter 6.7 of Spirtes et al. (2000).
Phase 5: Propagate orientation in $\mathcal{P}$ with graphical rules R1 − R10 from Zhang (2008), orienting as many edge marks as possible.

The FCI algorithm, introduced by Spirtes et al. (1995), is an asymptotically consistent constraint-based method that does not assume causal sufficiency. Because the original version failed to identify all edge marks invariant within the equivalence class, Zhang (2008) introduced additional orientation rules that made the FCI complete. Numerous FCI variants have been proposed to improve computational efficiency, robustness, and applicability. Anytime FCI (Spirtes 2001) and RFCI (Colombo et al. 2012) reduce computational costs by restricting parts of the search procedure, while FCI-Stable (Colombo and Maathuis 2014) removes order dependence during skeleton discovery. Other extensions broaden the settings in which FCI can be applied: Ψ-FCI (Jaber et al. 2020) incorporates soft interventional information, GFCI (Ogarrio et al. 2016) combines FCI with a greedy score-based search, and FCI-JCI (Mooij et al. 2020) supports causal discovery under different contexts with a static set of variables.

Nevertheless, existing approaches assume a static feature set and must relearn the causal MAG from scratch whenever new features arrive. This is computationally inefficient because many costly CI tests must be repeated. This limitation motivates FCI-SF, introduced in the next section.

5 The Proposed Method

5.1 Independence Preservation under Feature Arrival

In a SF scenario, there are two data-generating distributions: one with the new SF, P_{new}, and the previous distribution with only the old features, P_{old}. We will assume that P_{old} and P_{new} satisfy two standard assumptions used by constraint-based algorithms (Spirtes et al. 2000): both are latent projections of a third distribution P^* that satisfies the causal Markov assumption and the faithfulness assumption with respect to a MAG $\mathcal{G}^*$. The causal Markov assumption states that every separation in the true causal graph $\mathcal{G}^*$ implies a corresponding CI in the distribution P^* (i.e., $\mathcal{I}(\mathcal{G}^*) \subseteq \mathcal{I}(P^*)$). The faithfulness assumption states that every CI observed in P^* is due to a separation in the causal graph. Together, these assumptions imply that the graph captures exactly the CI relations of the distribution, that is, $\mathcal{I}(\mathcal{G}^*) = \mathcal{I}(P^*)$. As a consequence of this assumption, P_{old} is faithful to an initial MAG $\mathcal{G}_{old}$ and the introduction of new features augments the independence model of the original MAG by adding new independencies without eliminating the existing ones.

For instance, consider the graph in Fig. 2a. The simplest CIs that this MAG encodes are: $X_1 \perp_m X_3 \mid X_2$, $X_1 \perp_m X_5 \mid X_2$, $X_1 \perp_m X_4, X_2 \perp_m X_4$, $X_3 \perp_m X_5 \mid X_2$. Now, suppose additional features Y_1 and Y_2 arrive through a SF that respects the previous assumptions. The inclusion of these new features enables the us to detect new CI information involving Y_1 and Y_2, specifically: $X_1 \perp_m Y_1$, $X_1 \perp_m Y_2$, $X_2 \perp_m Y_1$, $X_2 \perp_m Y_2$, $X_3 \perp_m X_4 \mid Y_1$, $X_4 \perp_m X_5 \mid Y_2$, $Y_1 \perp_m Y_2$. The new distribution, P_{new} will remain faithful to the MAG shown in Fig. 2b. Notably, despite the addition of new features, all the nonadjacent features in the original graph in Fig. 2a remain nonadjacent in the new graph depicted in Fig. 2b. We call this property the *independence preservation property.*

Proposition 1 (Independence preservation property) *Consider $\mathcal{G}_{old}$ and $\mathcal{G}_{new}$, the two underlying causal structures in an SF scenario associated with the data generating distributions P_{old} and P_{new}, respectively, that are latent projections of a third distribution P^* that satisfies the causal faithfulness and Markov assumptions. Then any pair of nonadjacent features in $\mathcal{G}_{old}$ remain nonadjacent in $\mathcal{G}_{new}$ after the new set of previously hidden features, $\mathbf{H}_{new}$, streams in.*

Proof Let $\mathbf{O}_{old}, \mathbf{O}_{new}$ and V be the feature spaces of the distributions P_{old}, P_{new}, and P^*, respectively, such that $\mathbf{O}_{new} = \mathbf{O}_{old} \cup \mathbf{H}_{new}$. Let $\mathcal{G}^*$ be the underlying causal MAG of P^*. By construction, using Eq. (3), since $\mathbf{O}_{old} \subseteq \mathbf{O}_{new}$, then

$$\mathcal{I}(\mathcal{G}_{old}) \;=\; \mathcal{I}(L(\mathcal{G}^*, \mathbf{O}_{old})) = L(\mathcal{I}(\mathcal{G}^*), \mathbf{O}_{old}) \subseteq L(\mathcal{I}(\mathcal{G}^*), \mathbf{O}_{new}) = \mathcal{I}(\mathcal{G}_{new}). \tag{4}$$

All the CI statements in $\mathcal{G}_{old}$ are preserved in $\mathcal{G}_{new}$. In a MAG, two features are adjacent if and only if they are connected by an inducing path, which is equivalent to them being m-connected (Verma 1993). Furthermore, any pair X_i, X_j of nonadjacent features in $\mathcal{G}_{old}$ has a CI statement in $\mathcal{I}(\mathcal{G}_{old})$ that m-separates them. Eq. (4) guarantees that X_i, X_j are m-separable in $\mathcal{G}_{new}$. Consequently, X_i, X_j remain nonadjacent in the new causal graph. □

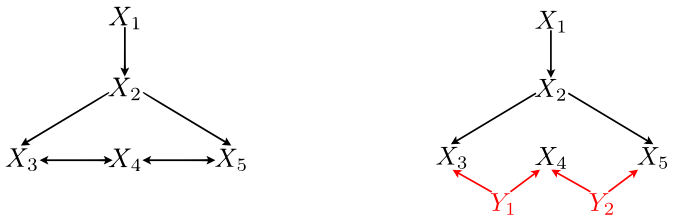


Fig. 2: In (a), the initial MAG. In (b), the new MAG (which is a DAG) results from receiving Y_1 and Y_2 as SF

5.2 The FCI-SF Algorithm

The FCI-SF algorithm has the same five phases as the original FCI algorithm (Section 4). However, FCI-SF introduces two main modifications. First, it extends the input to include the previously learned PAG $\mathcal{P}_{old}$, its separating sets $\text{Sep}_{old}[\cdot,\cdot]$, the augmented dataset $\mathcal{D}_{new}$ containing the new features $\mathbf{H}_{new}$, and a CI test $\mathcal{T}_{\mathcal{D}}$ with a significance level α. Second, it modifies the skeleton-discovery Phases 1 and 3 to reuse the structural and CI information stored in $\mathcal{P}_{old}$ and $\text{Sep}_{old}[\cdot,\cdot]$, thereby avoiding redundant CI tests and enabling incremental updates. A schematic overview of FCI-SF is given below.

Require:
Previous PAG $\mathcal{P}_{old}$
Previous separating set $\text{Sep}_{old}[\cdot,\cdot]$
Augmented dataset D_{new} that includes new features $\mathbf{H}_{new}$
CI test $\mathcal{T}_{\mathcal{D}}$ with significance level α
Ensure: New maximally informative PAG $\mathcal{P}_{new}$ based on $\mathcal{P}_{old}$

Phase 1: Departing from $\mathcal{P}_{old}$, create initial skeleton $\mathcal{P}_{new}$ using Algorithm 1 (below).
Phase 2: Orient unshielded triples as described in Chapter 6.7 of Spirtes et al. (2000).
Phase 3: Eliminate adjacencies of separable features, update $\mathcal{P}_{new}$, and eliminate all directions using Algorithm 2 (below).
Phase 4: Orient unshielded triples as described in Chapter 6.7 of Spirtes et al. (2000).
Phase 5: Propagate orientation in $\mathcal{P}$ with graphical rules R1 − R10 from Zhang (2008), orienting as many edge marks as possible.

Algorithm 1 corresponds to Phase 1 of FCI-SF. It constructs the initial skeleton by performing CI tests with conditioning sets of increasing size ℓ, removing an edge whenever a separating set is found. It differs from the adjacency phase of nonincremental constraint-based algorithms in three key respects. First, in addition to the augmented dataset, it receives the previously learned PAG $\mathcal{P}_{old}$ and separating sets $\text{Sep}_{old}[\cdot,\cdot]$, allowing previously obtained structural and CI information to be reused. Second, in lines 1–2, it initializes $\mathcal{P}_{new}$ using the independence preservation property

established in Proposition 1: the adjacencies among old features are inherited from $\mathcal{P}_{old}$ and $\text{Sep}_{new}[\cdot,\cdot]$ is initialized with $\text{Sep}_{old}[\cdot,\cdot]$ rather than as an empty map. Consequently, pairs of old features that are nonadjacent in $\mathcal{P}_{old}$ remain nonadjacent in $\mathcal{P}_{new}$, and their previously identified separating sets are retained. Furthermore, in line 1 of Algorithm 1 a complete graph is created over the new features, and every new feature is connected to every old feature. Third, line 9 imposes the restriction

$$\big(\{X_i, X_j\} \cup \mathbf{S}\big) \cap \mathbf{H}_{new} \neq \emptyset,$$

which requires every CI test to involve at least one new feature, either as an endpoint or as a member of the conditioning set. Consequently, when both X_i and X_j belong to $\mathbf{O}_{old}$, only candidate separating sets $\mathbf{S}$ containing at least one feature from $\mathbf{H}_{new}$ are considered. Tests involving only old features are skipped because they have already been performed in the previous execution. Nevertheless, an edge between two old features can still be removed when a newly observed feature yields a separating set that was previously unavailable. An illustrative execution is presented in Section 5.3.

Algorithm 1 builds the initial skeleton based on the *stable* adjacency phase of Colombo and Maathuis (2014), because it removes the order dependence of the original procedure; by storing the adjacencies of each feature level-wise (lines 5-7 of Algorithm 1) instead of doing so in the loop of line 8 in Algorithm 1, it yields the same skeleton regardless of how new incoming variables are enumerated (i.e., it does not matter the order in which CI statements are tested; the resulting graph should be the same), a property known as stability. This is crucial in our incremental framework to reduce the instability of trying CI statements with different orders in the algorithm.

After Algorithm 1 constructs the initial skeleton, Phase 2 orients the v-structures as in the standard FCI (Spirtes et al. 2000). Phase 3 then applies Algorithm 2 to remove adjacencies that can only be separated using variables outside the local adjacency sets, as identified through **Possible-D-SEP**. This phase introduces, in line 6, the same restriction used in Algorithm 2: a CI test is performed only if the endpoints or conditioning set contains at least one new feature. Tests involving exclusively old features are therefore skipped because they were already considered in the previous execution, reducing the total number of CI tests. After pruning the skeleton in lines 1–18, the algorithm replaces all remaining edge marks with circles in lines 19–21, thereby removing previous orientations. FCI-SF then orients the resulting graph by first identifying v-structures in Phase 4 and subsequently applying the graphical rules R1–R10 from Zhang (2008) in Phase 5.

Together, these modifications yield a global incremental constraint-based algorithm for the SF setting. As shown in Section 6, FCI-SF achieves substantial wall-time savings, particularly when incoming feature batches are small relative to the accumulated feature set. More importantly, FCI-SF is asymptotically consistent under the standard assumptions of constraint-based causal discovery, as established by the following proposition.

Algorithm 1 FCI-SF for the initial skeleton with SF

Require:
Previous PAG $\mathcal{P}_{old} = (\mathbf{O}_{old}, E_{old})$
Previous separating set $\mathrm{Sep}_{old}[\cdot,\cdot]$
Augmented dataset $\mathcal{D}_{new}$ over feature space $\mathbf{O}_{new} = \mathbf{O}_{old} \cup \mathbf{H}_{new}$
CI test $\mathcal{T}_{\mathcal{D}_{new}}$ with significance level α
Ensure:
New undirected skeleton $\mathcal{P}_{new}$
New separating sets $\mathrm{Sep}_{new}[\cdot,\cdot]$

$\mathcal{P}_{new} \leftarrow$ create the complete undirected graph over $\mathbf{H}_{new}$, join this graph with $\mathcal{P}_{old}$ by adding an edge between each variable in $\mathbf{O}_{old}$ and each feature in $\mathbf{H}_{new}$. Change endmarks of all the edges as $\circ\!\!-\!\!\circ$
$\mathrm{Sep}_{new}[\cdot,\cdot] \leftarrow \mathrm{Sep}_{old}[\cdot,\cdot]$
$\ell \leftarrow 0$
repeat
  for all vertices X_i in $\mathcal{P}_{new}$ **do**
    $\mathrm{a}(X_i) \leftarrow \mathrm{Adj}_{\mathcal{P}_{new}}(X_i)$
  end for
  for all ordered pairs (X_i, X_j) **s.t.** $X_j \in \mathrm{a}(X_i)$ **and** $|\mathrm{a}(X_i)\backslash\{X_j\}| \geq \ell$ **do**
    $\mathcal{S}_\ell \leftarrow$ subsets $\mathbf{S} \subseteq \mathrm{a}(X_i)\backslash\{X_j\}$ of size ℓ **s.t.** $(\{X_i, X_j\} \cup \mathbf{S}) \cap \mathbf{H}_{new} \neq \emptyset$
    repeat
      Select a new subset $\mathbf{S} \in \mathcal{S}_\ell$
      $\mathcal{S}_\ell \leftarrow \mathcal{S}_\ell \backslash \{\mathbf{S}\}$
      if $\mathcal{T}_{\mathcal{D}_{new}}(X_i, X_j \mid \mathbf{S}) > \alpha$ **then**
        Remove the edge between X_i and X_j from $\mathcal{P}_{new}$
        $\mathrm{Sep}_{new}[X_i, X_j] \leftarrow \mathbf{S}$
      end if
    until X_i and X_j are no longer adjacent **or** $\mathcal{S}_\ell = \emptyset$
  end for
  $\ell \leftarrow \ell + 1$
until all pairs of adjacent vertices (X_i, X_j) satisfy $|a(X_i)\backslash\{X_j\}| < \ell$
return $\mathcal{P}_{new}$, $\mathrm{Sep}_{new}[\cdot,\cdot]$

Proposition 2 *Let P^* be a distribution over the feature set V that is Markov and faithful with respect to a MAG $\mathcal{G}^* = (V, E)$, and let $\mathcal{D}^*$ be a dataset sampled from P^*. Assume that* FCI-SF *uses a consistent CI test. Let V be partitioned into k disjoint batches, $V = \mathbf{H}_1 \cup \mathbf{H}_2 \cup \cdots \cup \mathbf{H}_k$ which arrive sequentially, such that the previously hidden features in $\mathbf{H}_i$ become observable upon the arrival of batch i.*

Then, for each fixed batch i, in the large-sample limit, the output of FCI-SF *after sequentially processing batches $\mathbf{H}_1, \ldots, \mathbf{H}_i$ converges to the maximally informative PAG representing the latent projection of $\mathcal{G}^*$ onto the observed feature set $\mathbf{O}_i = \mathbf{H}_1 \cup \cdots \cup \mathbf{H}_i$.*

Proof This proposition follows directly from Corollary 7 in Appendix A, which establishes the result under a perfect CI oracle. For each fixed batch i, consistency of the CI test implies that, in the large-sample limit, the probability of correctly identifying all CI relations required by FCI-SF converges to one. Consequently, the algorithm asymptotically behaves as if supplied with perfect CI information and therefore converges to the maximally informative PAG characterized in Corollary 7 in Appendix A. □

Algorithm 2 FCI-SF for pruning nonadjacent separating sets for the final skeleton

Require:
Partially oriented graph $\mathcal{P}_{new} = (\mathbf{O}_{new}, E)$, with new incoming variables $\mathbf{H}_{new}$
Separating sets $\text{Sep}_{new}[\cdot, \cdot]$
Augmented dataset $\mathcal{D}_{new}$ over feature space $\mathbf{O}_{new} = \mathbf{O}_{old} \cup \mathbf{H}_{new}$
CI test $\mathcal{T}_{\mathcal{D}_{new}}$ with significance level α

Ensure:
Updated skeleton $\mathcal{P}_{new}$
Updated separating sets $\text{Sep}_{new}[\cdot, \cdot]$

for all vertices X_i in $\mathcal{P}_{new}$ **do**
  $\text{PDS}(X_i) \leftarrow$ **Possible-D-SEP**(X_i)
  for all $X_j \in \text{Adj}_{\mathcal{P}_{\text{new}}}(X_i)$ **do**
    $\ell \leftarrow 0$
    repeat
      $\mathcal{S}_\ell \leftarrow$ subsets $\mathbf{S} \subseteq \text{PDS}(X_i) \backslash \{X_j\}$ of size ℓ **s.t.** $(\{X_i, X_j\} \cup \mathbf{S}) \cap \mathbf{H}_{new} \neq \emptyset$
      repeat
        Select a new subset $\mathbf{S} \in \mathcal{S}_\ell$
        $\mathcal{S}_\ell \leftarrow \mathcal{S}_\ell \backslash \{\mathbf{S}\}$
        if $\mathcal{T}_{\mathcal{D}_{new}}(X_i, X_j \mid \mathbf{S}) > \alpha$ **then**
          Remove the edge between X_i and X_j from $\mathcal{P}_{new}$
          $\text{Sep}_{new}[X_i, X_j] \leftarrow \mathbf{S}$
        end if
      until X_i and X_j are no longer adjacent **or** $\mathcal{S}_\ell = \emptyset$
      $\ell \leftarrow \ell + 1$
    until X_i and X_j are no longer adjacent **or** $|\text{PDS}(X_i) \backslash \{X_j\}| \leq \ell$
  end for
end for
for all remaining edges between pairs of features X_i, X_j **do**
  Draw the endmarks of the edge between X_i and X_j as $X_i \circ\!\!-\!\!\circ X_j$
end for
return $\mathcal{P}_{new}$, $\text{Sep}_{new}[\cdot, \cdot]$

5.3 Example of the Execution of FCI-SF

Consider an illustrative execution of the FCI-SF algorithm. We assume that our data is ruled by the causal graph $\mathcal{G}^*$ in Fig. 3a which is defined over the full feature set $V = \{X_1, X_2, X_3, X_4, H_1, H_2, H_3\}$, but initially, only the subset $\mathbf{O}_{\text{old}} = \{X_1, X_2, X_3\}$ is observable. When only the features in $\mathbf{O}_{\text{old}}$ are available, the algorithm initially operates with an standard constraint-based algorithm (e.g., FCI) and learns the PAG shown in Fig. 3b. After the arrival of the feature H_1, we aim to update the PAG from Fig. 3b. To this end, we assume access to perfect CI information from the dataset, i.e., an "oracle" that provides exact CI statements derived from the latent projection of $\mathcal{I}(\mathcal{G}^*)$ onto $\mathbf{O}_{\text{new}} = \{X_1, X_2, X_3, X_4, H_1\}$. We assume an oracle provides the CI statements relevant to H_1: specifically, $H_1 \perp_m X_2 \mid X_3$ and $H_1 \perp_m X_4 \mid \{X_2, X_3\}$. FCI-SF will accept three inputs: (1) The previous PAG, $\mathcal{P}_{old}$, as shown in Fig. 3b; (2) The set of previously identified separating sets, Sep_{old}, for nonadjacent pairs: specifically, $\text{Sep}_{old}[X_2, X_4] = \{X_1\}$ and $\text{Sep}_{old}[X_1, X_3] = \{X_2\}$; and (3) The augmented dataset $\mathcal{D}_{new}$ incorporates the new feature H_1.

Upon execution, Algorithm 1 takes these inputs ($\mathcal{P}_{old}$, Sep_{old}, and $\mathcal{D}_{new}$). After completing line 1, $\mathcal{P}_{new}$ will match Fig. 3c and Sep_{new} will be equal to Sep_{old}. Initially, the algorithm checks separating sets of size $\ell = 0$ without removing any edges,

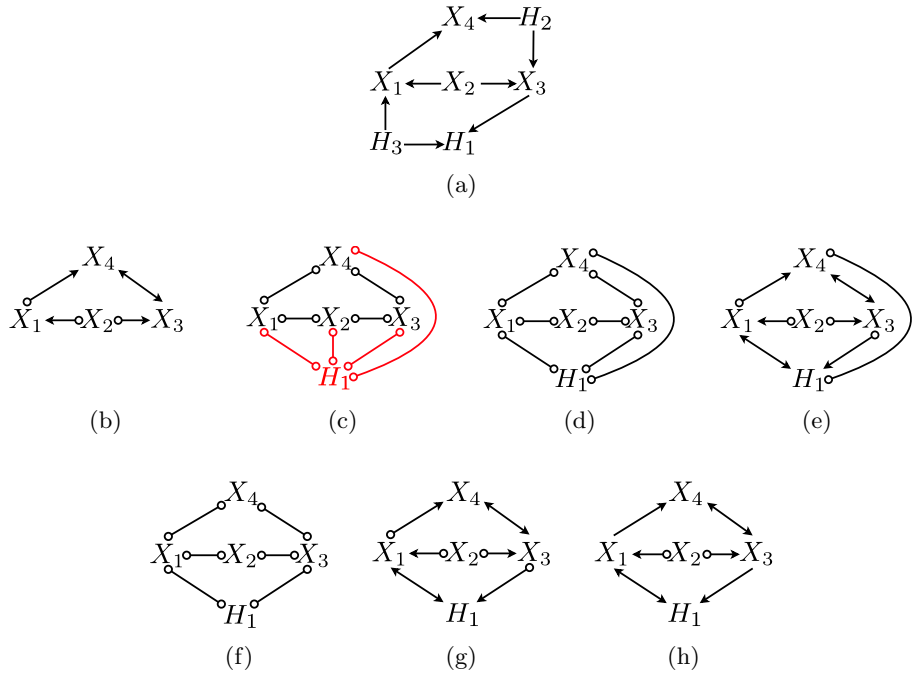


Fig. 3: Panel (a) shows the DAG $\mathcal{G}^*$ over the feature space $V = \{X_1, X_2, X_3, X_4, Y, H_2, H_3\}$; the initial PAG is in (b), where $\mathbf{O}_{\text{old}} = \{X_1, X_2, X_3, X_4\}$; and panels (c)–(h) show the step-by-step PAG updates performed by the FCI-SF algorithm upon the arrival of the new streaming feature Y

subsequently incrementing ℓ to 1. With $\ell = 1$, since X_2 and H_1 are CI given X_3, the edge between X_2 and H_1 is removed, and X_3 is recorded as $\text{Sep}_{new}[X_2, H_1] = \{X_3\}$. With $\ell = 2$, Algorithm 1 would not be able to identify any CI statement. After this step with $\ell = 3$, since all vertices have a degree of three or less, no additional separating sets can be identified, and Algorithm 1 terminates, outputting $\mathcal{P}_{new}$ (Fig. 3d) and Sep_{new} which contains the separating sets Sep_{old} and the new extra separating set $\text{Sep}_{new}[X_2, H_1] = \{X_3\}$.

Next, the FCI-SF algorithm proceeds to Phase 2 of FCI-SF to orient v-structures, which produces the updated PAG shown in Fig. 3e. Subsequently, Phase 3 of FCI-SF triggers Algorithm 2. With input $\mathcal{P}_{new}$, Sep_{new}, and $\mathcal{D}_{new}$, Algorithm 2 examines CIs for subsets with nonadjacent features. It does not detect any separating set until it reaches $\ell = 3$, when it discovers that H_1 is CI of X_4 given $\{X_1, X_2, X_3\}$ and removes the edge $H_1 \circ\!\!-\!\!\circ X_4$. This operation updates $\mathcal{P}_{new}$ as shown in Fig. 3f, with the separating set $\text{Sep}_{new}[H_1, X_4] = \{X_1, X_2, X_3\}$.

Phase 4 then identifies and orients the remaining v-structures, yielding the PAG shown in Fig. 3g. Finally, Phase 5 applies rules R1–R10 from Zhang (2008) to orient as many remaining edge marks as possible, producing the final PAG in Fig. 3h. When two variables arrive together, for example, when H_2 and H_3 arrive together in the same feature batch, FCI-SF first constructs the complete graph over the new variables (i.e., $H_2 \circ\!\!-\!\!\circ H_3$), connects each new variable to every old variable, and then performs CI tests as in the single-variable case.

6 Experiments

All experiments and causal discovery algorithms are available in the FCI-SF Python repository developed for this work. Aside from our repository, the implementation provided by Zhang et al. (2025) includes only LiNGAM-SF, CSBS, and CSS algorithms and is developed in MATLAB. In contrast, our repository is implemented in Python and is built on top of the causal-learn repository (Zheng et al. 2024), which is becoming increasingly relevant in the research community in the field of causality and facilitates integration as well as further development. To the best of our knowledge, this makes our repository the only one that offers implementations not only of FCI-SF but also of all currently available state-of-the-art causal discovery algorithms with streaming capabilities.

We evaluated the performance of all causal discovery algorithms for SF using multiple metrics on both synthetic and real-world datasets, enabling a comprehensive comparison of their behavior. The experiments were performed on a computer equipped with an Intel Core i7-6700K CPU (4 cores, 8 threads, 4.0–4.2 GHz) and 32 GB of RAM.

In the case of synthetic data, the ground-truth graph is available. Consequently, recovering the skeleton is arguably the most crucial step, as any errors made at this stage inevitably propagate to the final orientation phase. For this reason, our evaluation prioritizes metrics that assess how accurately the algorithm recovers the adjacency structure of the graph. We report the Hamming distance (*HD*), which counts the number of missing and spurious adjacencies relative to the ground-truth skeleton. We also

include metrics derived from the confusion matrix but applied them only to skeleton recovery. Namely, the $F1$ score, which balances precision and recall of predicted adjacencies, and Cohen's κ, which adjusts for agreement occurring by chance and therefore provides a more robust measure of skeleton-level accuracy. Among the metrics that assess edge endmarks, we use the structural Hamming distance with endmarks (*SHDe*), as defined by Colombo and Maathuis (2014), which penalizes incorrect orientations or endpoint marks in addition to adjacency errors. Together, these metrics provide a complete assessment of structural accuracy, from adjacencies to endmarks.

In the case of real world-data, the ground-truth graph is often not available. As a solution, self-compatibility metric (Faller et al. 2024) is inspired by the idea that causal structures learned from different subsets of variables should be mutually compatible, with any incompatibility reflecting either finite-sample effects or violations of the algorithm's assumptions. Moreover, self-compatibility can be used as a metric that provides a practical "proxy" for the *SHDe* even when the ground-truth graph is unavailable. Let the full set of variables be $\mathbf{O}$. Given a dataset $\mathcal{D}$ and an algorithm $\mathcal{A}$, we denote by $\mathcal{A}(\mathcal{D})$ the graphical model learned from the dataset, containing variables V with algorithm $\mathcal{A}$. Then, for subsets of variables of our choice $\mathbf{H}_1, \ldots, \mathbf{H}_k \subseteq V$, the *self-compatibility* metric (SC[2]) is defined as the average *SHDe* between the graph obtained by the marginalized latent projection (which plays the role of ground-truth) of $\mathcal{A}(\mathcal{D})$ onto $\mathbf{H}_i$ and applying $\mathcal{A}$ to each restricted dataset $\mathcal{D}_{\mathbf{H}_i}$,

$$SC^{\mathcal{G}}(\mathcal{A}, \mathcal{D}, \{\mathbf{H}_i\}_{i=1}^{k}) = \frac{1}{k} \sum_{i=1}^{k} SHDe(L(\mathcal{A}(\mathcal{D}), \mathbf{H}_i), \mathcal{A}(\mathcal{D}_{\mathbf{H}_i})).$$

While interventional metrics (i.e., metrics that evaluate causal models based on their ability to predict the effects of interventions on the system) would, in principle, provide a more faithful evaluation of a model's causal predictions, existing techniques—such as the structural intervention distance (Peters and Bühlmann 2015) or the PC-stable + IDA approach of Colombo and Maathuis (2014)—are limited to DAGs. Given that our work is formulated in terms of MAGs, to the best of our knowledge, no computationally tractable intervention-based evaluation is currently available in this setting. This limitation is consistent with the broader landscape of causal discovery research: only about 5% of papers use interventional metrics, as shown in the systematic review of Brouillard et al. (2025)—highlighting the need for improved interventional evaluation tools, which lies outside the scope of this work.

6.1 Synthetic Datasets

The synthetic experiments compare the accuracy and runtime of FCI-SF with those of state-of-the-art methods under controlled conditions. We aimed to cover a broad range of settings while keeping the total computational time below one week, which guided choices such as the number of variables and generated DAGs. Data were sampled from random Gaussian Bayesian networks generated using the implementation of Ankan and Textor (2024). The underlying DAGs followed an Erdős–Rényi model with 50

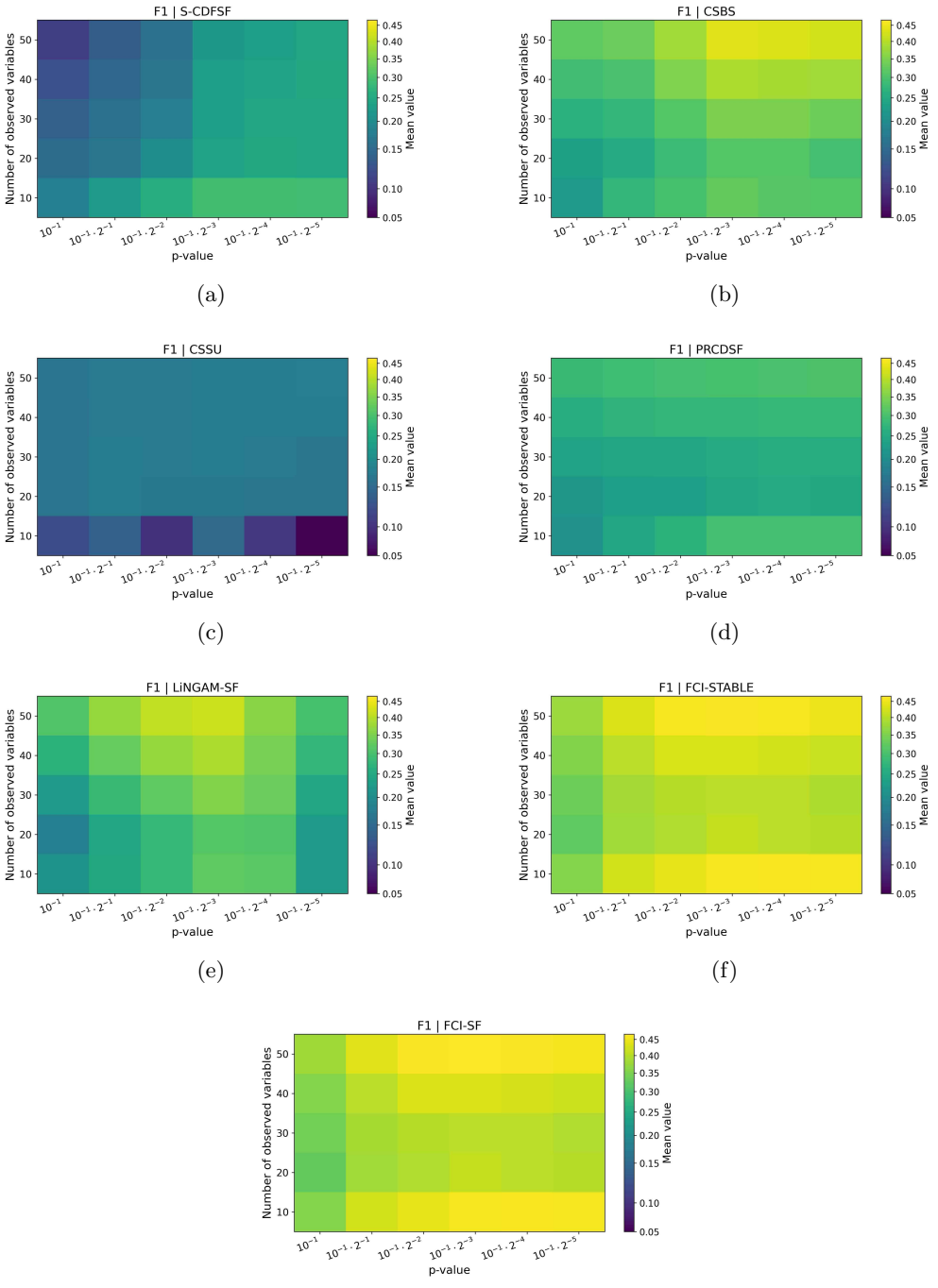
[2]Despite the author's denote it as $k^{\mathcal{G}}$ we use $SC^{\mathcal{G}}$ to differentiate it from Cohen's k and from the number of subsets of variables.

variables and an expected degree of two, consistent with the graph-generation setting considered by Colombo and Maathuis (2014).

Once the data were generated, we learned the graphs using CI tests based on partial correlations with Fisher's Z-transformation for the S-CDFSF, CSBS, PRCDSF, LiNGAM-SF, FCI, and FCI-SF algorithms, while CSSU employed Gaussian symmetrical uncertainty computed via Gaussian differential entropies. Graph learning was conducted incrementally over five iterations, starting with 10 variables and adding 10 variables at each step. While the FCI algorithm learns the graph from scratch after new variables arrive, the rest of the algorithms are adapted to SF and add the new variables to the rest of the distribution. The experimental datasets were generated from 100 distinct DAG., with one dataset produced per DAG and two different variable orderings evaluated for each DAG. The objective is to assess how each algorithm evolves as new features are introduced over time. We present three different simulation setups,

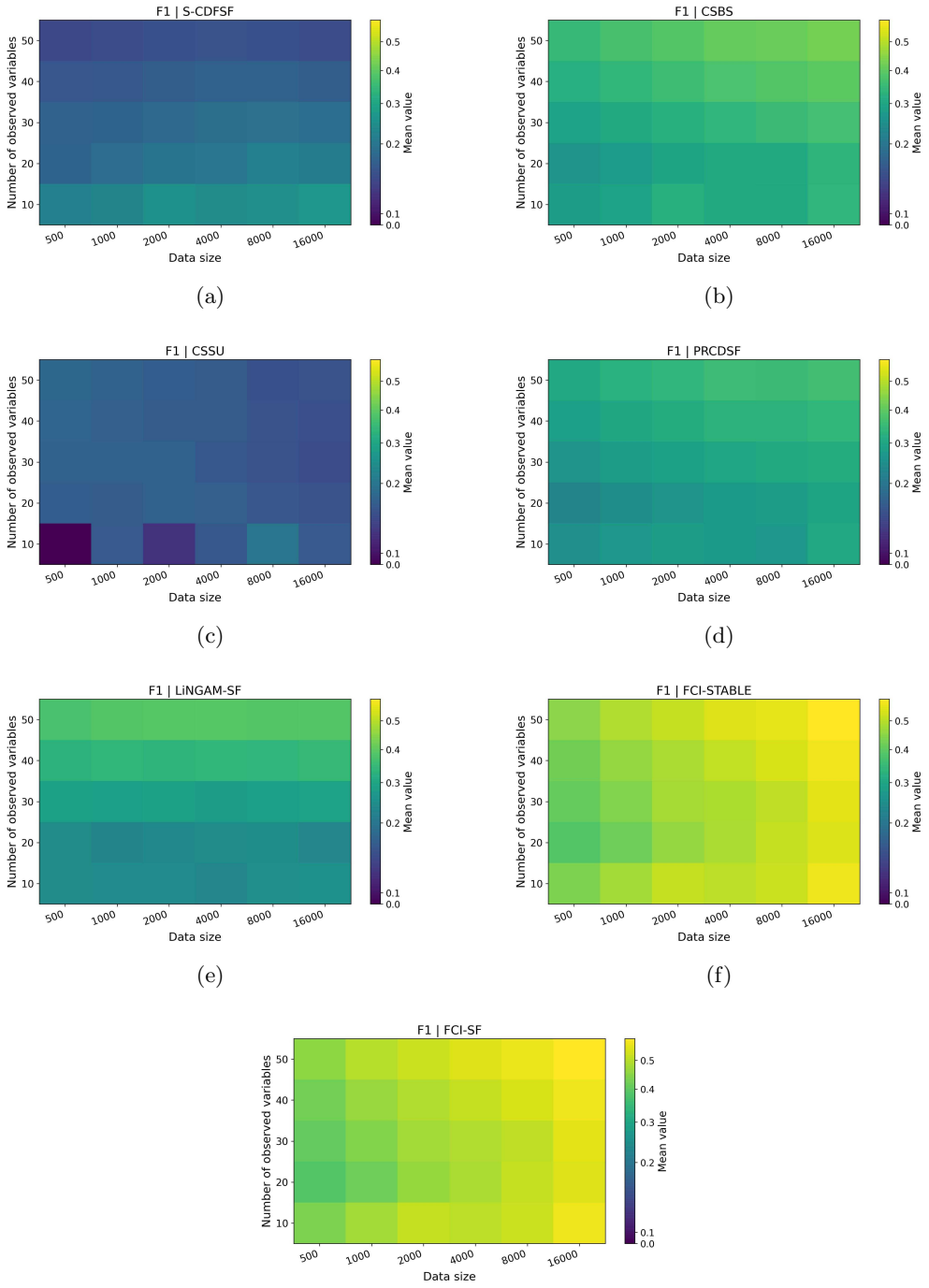
The first simulation setup tests different data sizes and p-values (significance levels). First, this simulation fixes the data size at $N = 500$ i.i.d. samples. We compared the algorithms across a tuning-parameter (p-value) grid of $\alpha \in \{0.003125, 0.00625, 0.0125, 0.025, 0.05, 0.1\}$. Although CSSU uses a different test for learning the graph, Yang et al. (2018) showed that its best performance is achieved for $\alpha \in [0.05, 0.1]$, allowing us to compare it to the other algorithms on a similar scale. Figs. 4a–4g present heatmaps of the $F1$ score for each algorithm, where lighter values indicate better performance. The complete numerical results underlying Fig. 4 are reported in Table B1 in Appendix B. Overall, FCI-SF achieves the best performance, matching that of FCI. However, while FCI-SF reuses the previously learned graph as new batches of features arrive, FCI relearns the graph from scratch each time. Lower p-values generally lead to higher $F1$ scores, except for the CSSU algorithm, which remains largely unaffected by the p-value when the number of variables exceeds 10. In general, no algorithm achieves an $F1$ score higher than 0.45. There are two main reasons for this: (1) the data are sparse (only 500 instances); and (2) we consider graphs (and their latent projections over subspaces of variables) with 50 variables and an expected edge density of two. Consequently, each graph contains approximately 50 edges out of 1225 possible edges, meaning that more than 95% of the edges are negatives, which makes detecting positive edges particularly challenging.

In the second simulation setup, we set $\alpha = 0.05$, since no algorithm reaches either their minimum or maximum. The synthetic datasets contained i.i.d. samples $N \in \{500, 1000, 2000, 4000, 8000, 16000\}$. In this setup, our goal is to evaluate the algorithms under varying data sizes and numbers of variables, using heatmaps of the $F1$ score for each algorithm, where lighter values indicate better performance. The full numerical data used to generate Fig. 5 are presented in Table B2 in Appendix B. At first glance, FCI-SF (Fig. 5g) and FCI (Fig. 5f) yield very similar results and achieve the best overall performance among all algorithms. Moreover, they are the most sensitive to the data size and exhibit the greatest improvement as the sample size increases. Although we selected a p-value for which FCI-SF does not attain its optimal performance, we observe that its $F1$ score approaches 0.6 as the data size grows, which remains a competitive result with more than 95% negative edges.



(a) (b) (c) (d) (e) (f) (g)

Fig. 4: Heatmaps of the F1 score for (a) S-CDFSF, (b) CSBS, (c) CSSU, (d) PRCDSF, (e) LiNGAM-SF, (f) FCI, and (g) FCI-SF. Lighter colors indicate higher F1 scores. The significance level is shown on the x-axis and the number of observed variables on the y-axis



(a) (b) (c) (d) (e) (f) (g)

Fig. 5: Heatmaps of the F1 score (lighter colors indicate better performance) for different causal discovery algorithms, with the data size on the x-axis and the number of variables on the y-axis

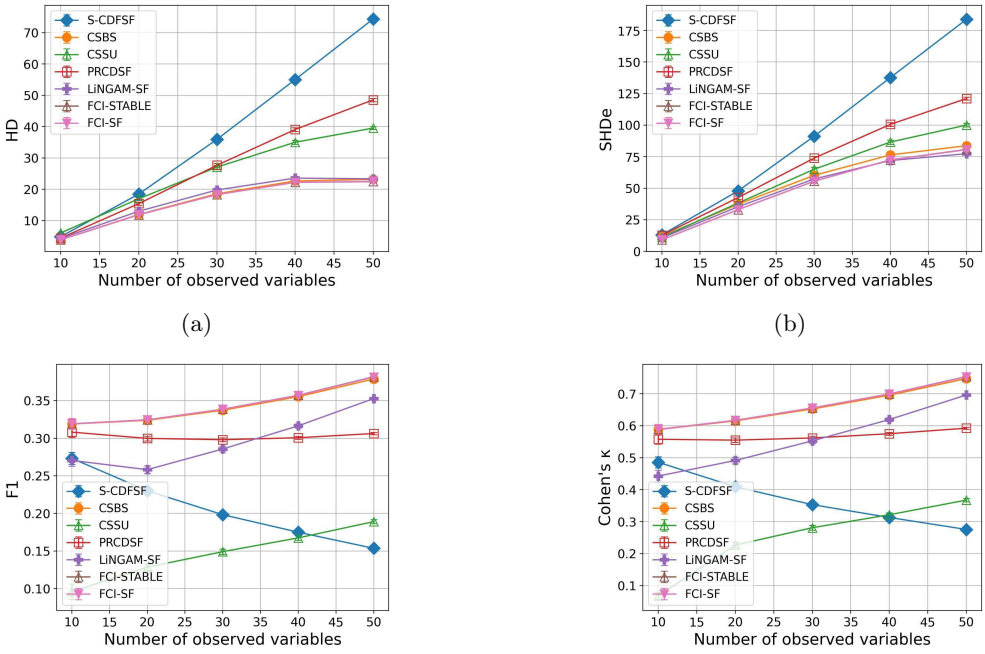


Fig. 6: Results of four metrics (y-axis) : (a) *HD*, (b) *F1*, (c) Cohen's κ, and (d) *SHDe* with varying feature space dimensions (x-axis). New features are added incrementally in batches of 10. $\alpha = 0.05$ and $N = 500$

In the third simulation setup, we study the effect of incremental learning on algorithm performance. We set the p-value as 0.05 for the same reasons explained before, and we set $N = 500$ to avoid giving FCI-SF an advantage over its competitors. Note that Figs. 6a–6b include 95% confidence interval bars; however, due to the low variability in the results of each algorithm, these intervals are barely visible. The underlying numerical results, together with the corresponding confidence interval half-widths, are reported in Table B3 in Appendix B. As the number of variables increases, the *HD* (Fig. 6a) and the *SHDe* (Fig. 6b) also increases. A naïve interpretation would suggest that graphical performance deteriorates when more variables are included. However, the growth of these metrics is to be expected since the number of edges also grows. At the same time, the F1 score (Fig. 6c) and Cohen's κ (Fig. 6d) improve with larger variable sets. In particular, for FCI-SF and CSBS, Cohen's κ reaches values between 0.6 and 0.75, which demonstrates a strong performance. This indicates that, although more structural errors occur in absolute terms when additional variables are introduced, the algorithm gains more information about the causal system. As a consequence, adding variables to the system leads to improved discrimination between true structure and random noise—hence, the accuracy becomes less reliant on chance.

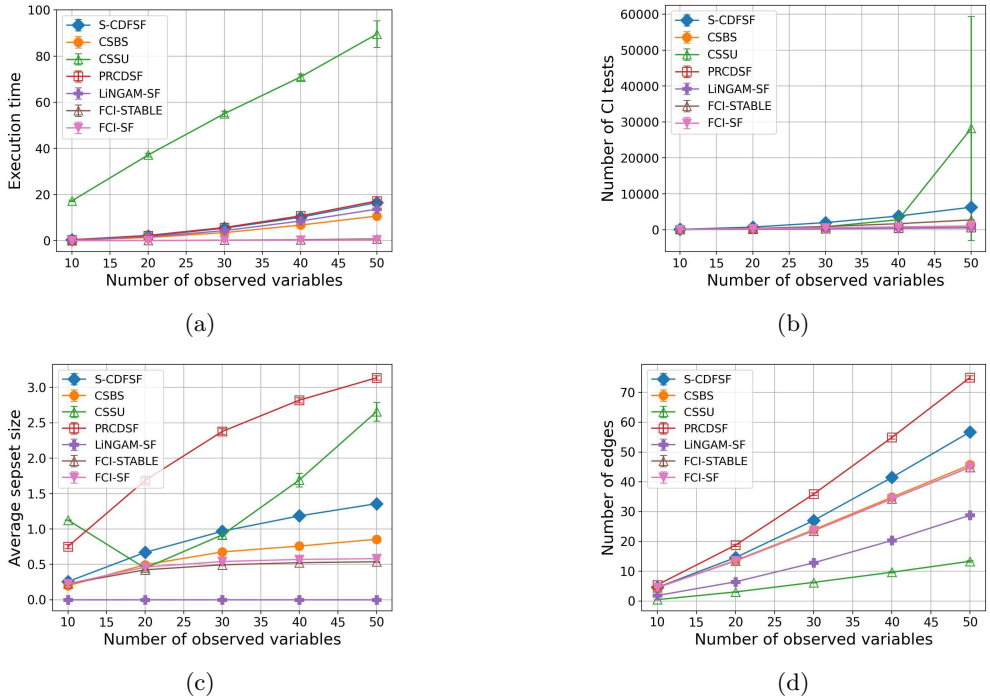


Fig. 7: Results of the computational cost (y-axis) with varying feature space dimensions (x-axis). New features are added incrementally in batches of 10. Panel (a) shows the average number of CI tests, (b) the average separating set size, (c) the number of edges in the resulting graph, and (d) the execution time in seconds. $\alpha = 0.05$ and $N = 500$

Finally, in Fig. 7 we compare the effect of incremental learning on computational gains in the third simulation setup ($\alpha = 0.05$ and $N = 500$). The numerical values are presented in Table B4 in Appendix B. We consider both execution time (Fig. 7a) and the number (Fig. 7b) and complexity of CI tests (Figures 7c and 7d). Here, FCI-SF clearly outperforms the alternatives, reducing execution time by 44% compared to rerunning FCI and being nearly 20× faster than the second-fastest SF causal discovery algorithm, CSBS. Similarly, FCI-SF halves the number of CI tests of FCI-Stable (Fig. 7b) while CSSU performs an excessive number of CI tests, and—as in the case of PRCDSF—its separating sets keep growing in size (Fig. 7c), rendering CSSU computationally intractable. Indeed, during our experiments, we observed that for variable sets larger than 50 variables, both can take several days to learn a single graph. Thus, our experimental results indicate that FCI-SF outperforms the competing algorithms in higher-dimensional settings in terms of execution time, while achieving comparable or better accuracy.

6.2 Real-World Datasets

This section presents two experiments. The first compares FCI-SF with other causal discovery algorithms designed for the SF setting when the ground-truth graph is unavailable. The comparison is based on graphical self-compatibility (Faller et al. 2024) and uses the nine real-world datasets listed in Table 2. The second experiment applies FCI-SF to the *Arabidopsis thaliana* isoprenoid gene-expression data analyzed by Wille et al. (2004) to infer and examine the corresponding causal network.

We run the first experiment using self-compatibility with *SHDe* with endmarks on nine real-world datasets (Table 2) obtained from different repositories such as NASA's open data portal and the UCI machine learning repository. All datasets underwent a data-cleaning process (Appendix C.1), and the resulting cleaned versions are available in the FCI-SF repository. For the self-compatibility analysis, we constructed five nested subsets of variables for each dataset, $\mathbf{H}_1 \subseteq \cdots \subseteq \mathbf{H}_5$, to reproduce the incremental nature of the SF setting. The initial subset $\mathbf{H}_1$ contains 20% of the features, and each subsequent subset adds another 20%, such that $\mathbf{H}_i$ contains $20i$% of the complete feature set.

Fig. 8 presents the critical-difference (CD) diagram obtained from the Friedman test followed by the Nemenyi post-hoc test. The horizontal axis shows the average ranks of the algorithms, ranging from 6 (worst) to 1 (best). FCI-SF achieves the lowest average rank (2.44), followed by CSBS (2.83), LiNGAM-SF (2.89), CSSU (3.33), S-CDFSF (3.89), and PRCDSF (5.78). Horizontal bars denote groups with no statistically significant rank differences, yielding PRCDSF as the method with the worst average rank and with two overlapping groups: {S-CDFSF, CSSU, LiNGAM-SF, CSBS}, and {CSSU, LiNGAM-SF, CSBS, FCI-SF}. Overall, FCI-SF provides the most favorable accuracy–efficiency trade-off: it achieves the best average rank, ranks first on four of the five datasets, and belongs to the top-performing group. Moreover, as shown in the synthetic experiments, FCI-SF is approximately 20 times faster than the competing algorithms.

Table 2: Summary of the cleaned real-world datasets.

Dataset	**Reference**	**# Vars.**	**Size** N
Appliances prediction	Candanedo and Feldheim (2016)	26	19735
Brightness temperature and soil moisture	Colliander et al. (2019)	11	5184
Cancer Wisconsin diagnostic	Wolberg et al. (1994)	30	569
CIPCaD-Bench	Menegozzo et al. (2022)	17	23132
Financial data for FCMs	Commodity Futures Trading Commission (2024)	16	63
Fire emissions Indonesia	NASA (2020)	16	275184
Gas turbine emissions	Kaya and Tüfekci (2019)	11	7384
Parkinson's telemonitoring	Tsanas et al. (2010)	19	5875
Wine quality	Cortez et al. (2009)	11	1599

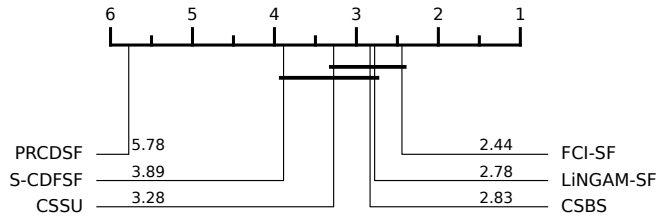


Fig. 8: CD diagram from applying Friedman's test using weighted self-compatibility with *SHDe* on the nine datasets from Table 2

The second experiment applies FCI-SF to the gene-expression dataset studied by Wille et al. (2004) to infer the network associated with isoprenoid biosynthesis in *Arabidopsis thaliana*. The dataset contains expression measurements for 39 genes across 118 instances. It includes two known pathways with established causal directions, represented by dashed edges in Figs. 9-11: the cytosolic mevalonate (MVA) pathway, comprising 15 genes, and the plastidial methylerythritol phosphate (MEP) pathway, comprising 19 genes. The remaining five genes encode proteins localized in the mitochondrion.

We learned the causal graph incrementally over four feature batches constructed according to the known causal progression of the pathways. Each new batch introduced only genes downstream of the currently observed variables in the pathway, thereby reproducing a laboratory setting in which measurements are added along a causal progression. To assess the stability of the learned graphs, we applied bootstrap resampling. Following Wille et al. (2004), an adjacency was retained in the final graph only if it appeared in at least 80% of the bootstrap replicates. Whereas Wille et al. (2004) reported only undirected adjacencies and studies such as Bühlmann et al. (2014) presented the result as a DAG, we additionally summarized the endpoint marks identified by FCI-SF. For each retained adjacency, an endpoint mark was preserved if the same mark appeared in at least 70% of the bootstrap graphs in which that adjacency was present; otherwise, the corresponding endpoint was represented by a circle. We used a lower threshold for endpoint marks because their estimation is more variable and depends on the prior recovery of the corresponding adjacency.

The first batch contains the first four MEP genes, the first three MVA genes, and the five mitochondrial proteins. Figure 9a shows the learned structure, with solid dark edges denoting inferred relationships and dashed edges the known pathway directions. Given the limited number of observed genes, the graph is plausible: many adjacencies are bidirected, $\leftrightarrow$, indicating no inferred ancestral relation and possible latent confounding by genes not yet introduced.

The second batch adds the next four genes downstream in the MEP pathway and the next five genes in the MVA pathway. The result is shown in Fig. 9b. Again, solid dark edges represent the inferred relationships and dashed edges indicate the known causal directions of the pathways. As these variables become observable, new

structural patterns emerge, including the partially oriented path DXR$\circ\!\!\rightarrow$ MCT$\circ\!\!\rightarrow$ CMK$\circ\!\!\rightarrow$ MECPS, which is consistent with the known pathway direction. The third batch introduces five additional MEP genes and three MVA genes (Fig. 10)). Most previously identified adjacencies are preserved, suggesting that these additions contribute less structural information. Finally, the fourth batch adds the six GGPPS genes associated with the MEP pathway, together with two GGPPS genes and two DDPS genes from the MVA pathway (Fig. 11). This final update produces relatively few structural changes, with most adjacencies remaining unchanged from the third batch. Overall, the learned graphs are broadly consistent with the results of Wille et al. (2004) and Bühlmann et al. (2014): many inferred edges connect genes within the same pathway, and most oriented endpoint marks follow the known upstream-to-downstream progression, typically pointing from ancestors to descendants.

7 Conclusion

This work addressed constraint-based causal discovery under incremental SF and introduced FCI-SF, a global incremental extension of FCI that avoids redundant CI tests while preserving its theoretical guarantees. To the best of our knowledge, FCI-SF is the first global constraint-based algorithm for the SF setting with formal asymptotic consistency guarantees. Our analysis also highlights the importance of accounting for hidden variables in incremental settings, where variables that are initially unobserved may become available at later stages.

The theoretical framework developed in this work extends beyond FCI-SF and may support incremental versions of other constraint-based algorithms while preserving asymptotic consistency. We also developed, to the best of our knowledge, the first Python package providing a common implementation of the main causal discovery methods proposed for the SF setting. Experiments on synthetic and real-world data showed that FCI-SF achieves substantial computational savings while maintaining graphical accuracy comparable to that of FCI learned from scratch. It also obtained the best average rank in the real-world experiments and recovered plausible causal structures in the *Arabidopsis thaliana* gene-expression study. Overall, these results establish FCI-SF as a theoretically grounded and computationally efficient method for incremental causal discovery in SF settings.

Several directions remain open for future research. Graph operations such as marginalization remain computationally demanding and would benefit from more efficient implementations. Robust incremental methods are also needed for settings involving noisy, missing, or heterogeneous observations. Finally, extending the proposed framework to dynamic and time-evolving causal systems would further broaden its applicability.

Acknowledgements

This work was supported by the Spanish Ministry of Education through General Grant No. 24AE/1405247 and Research Collaboration Grant No. 24CO1/007185. The first author would also like to thank his colleagues at the laboratory for their advice, feedback, and willingness to discuss his research ideas.

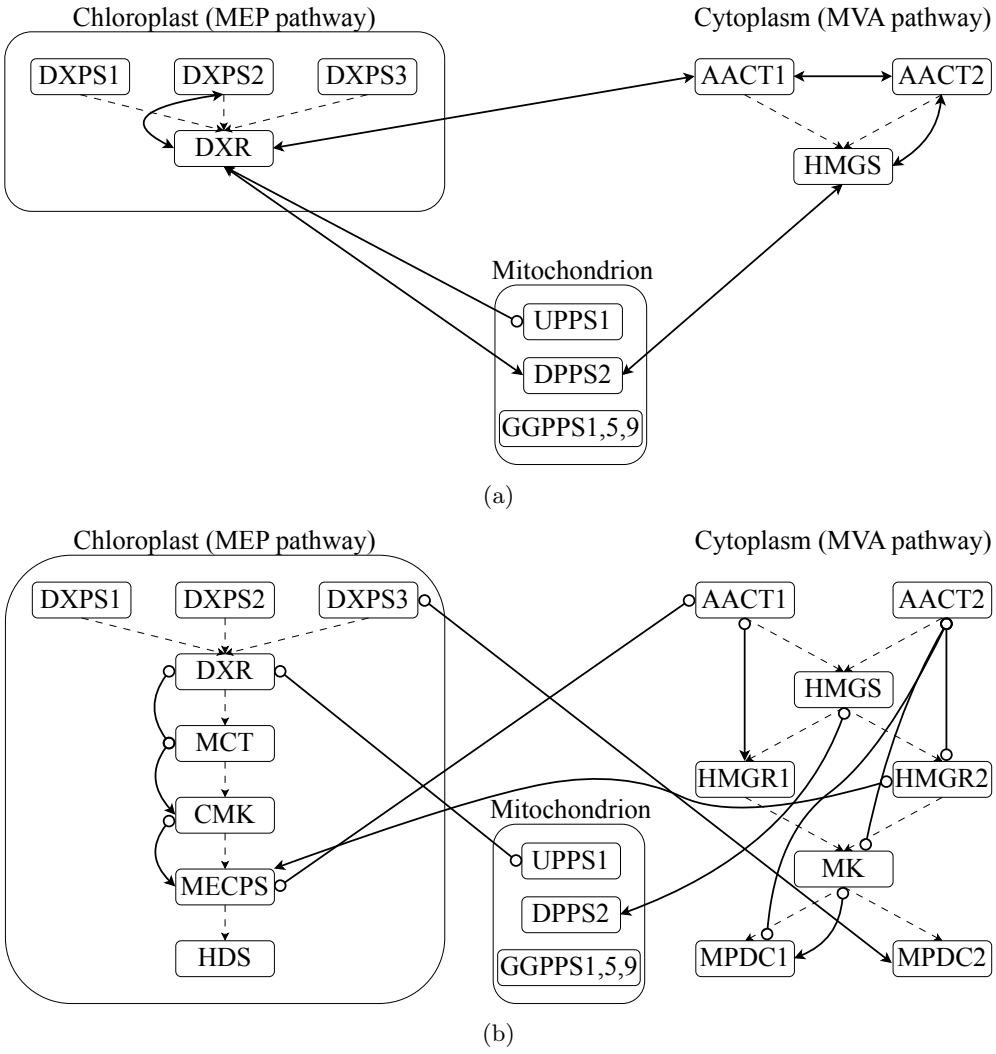


Fig. 9: Results obtained with FCI-SF. Panel (a) shows the graph after processing the first feature batch, and panel (b) shows the graph after processing the second feature batch

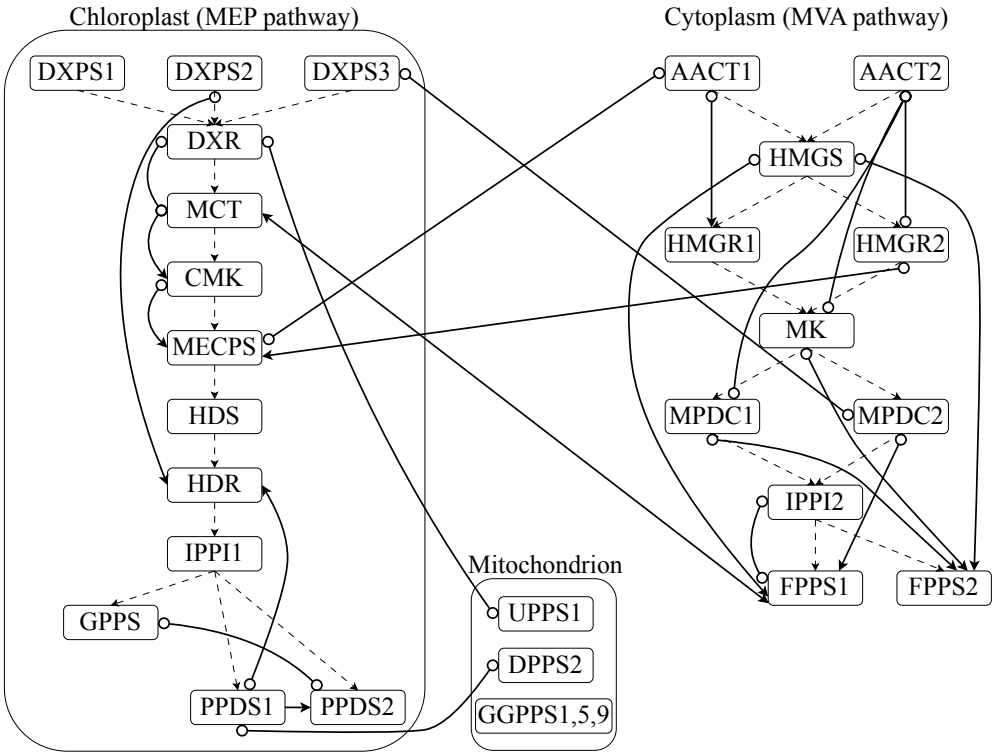


Fig. 10: Results obtained with FCI-SF after processing the third feature batch

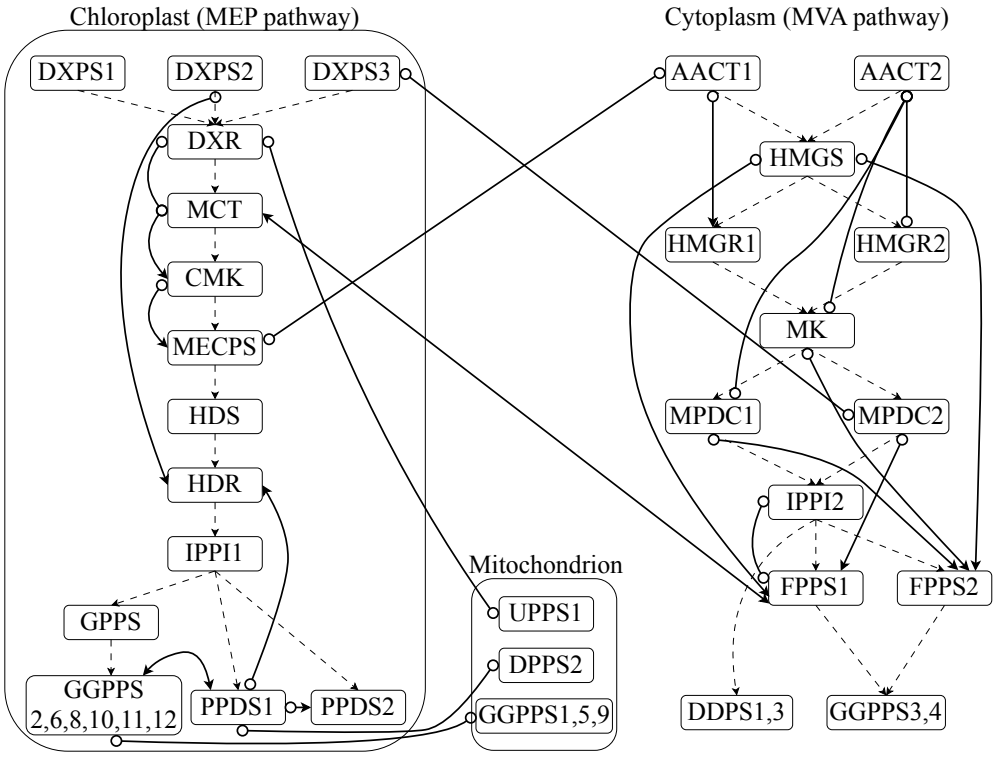


Fig. 11: Results obtained with FCI-SF after processing the fourth feature batch

Statements and Declarations

Competing interests

The authors have no competing interests to declare that are relevant to the content of this article.

Author contributions

J.M.D.M. conceived the study, developed the methodology, implemented the proposed framework, conducted the theoretical analysis and experiments, analyzed the results, and wrote the original manuscript. C.B. and P.L. supervised the work, contributed to funding acquisition, and provided critical feedback and manuscript revisions. All authors reviewed and approved the final manuscript.

Data availability

The synthetic data were generated using pseudorandom number generation to ensure the reproducibility of the experiments. All real-world datasets used in this study are publicly available. The cleaned datasets used to produce the critical-difference diagram are available in the repository accompanying this work: https://github.com/ComputationalIntelligenceGroup/FCI-SF/tree/main/clean_datasets/real-world_datasets2.

Code availability

The implementation is available at https://github.com/ComputationalIntelligenceGroup/FCI-SF.

Appendix A Correctness of the FCI-SF Algorithm

A desirable property of a causal discovery method is asymptotic consistency, meaning that the probability of recovering the correct causal representation converges to one as the sample size increases. For constraint-based methods, this property is commonly established by first proving correctness under perfect CI information obtained directly from the distributional CI model $\mathcal{I}(P)$, often referred to as the "oracle" setting. Correctness requires soundness and completeness, meaning that the procedure recovers the correct Markov equivalence class and produces a maximally informative graph in which all edge endmarks invariant across the causal graphs in that equivalence class are identified. Once correctness under oracle CI information has been established, a consistent CI test recovers the same CI relations as the oracle as the sample size tends to infinity. Consequently, the procedure is asymptotically consistent.

To the best of our knowledge, existing causal discovery algorithms for SF do not provide general guarantees of soundness and completeness and may therefore return incorrect causal relations even under oracle CI information. In contrast, FCI-SF satisfies both properties, as established in Theorem 6. Consequently, it is the first incremental FCI-style procedure whose output is provably equivalent to that obtained

by rerunning FCI from scratch after each feature arrival, as formalized in Corollary 7. Before proving the main theorem, we introduce several auxiliary results.

For a MAG $\mathcal{M}$, $\mathbf{D\text{-}SEP}(X_i, X_j)$ denotes the set of vertices $X_k \neq X_i$ for which there exists a path from X_i to X_k such that every non-endpoint vertex is both a collider on the path and an ancestor of X_i or X_j. A central property of this set is that, whenever X_i and X_j are nonadjacent and $X_i \notin \mathrm{An}_{\mathcal{M}}(X_j)$, conditioning on $\mathbf{D\text{-}SEP}(X_i, X_j)$ m-separates them (Spirtes 2001, Theorem 6.2). Since a MAG contains no directed cycles, for every nonadjacent pair (X_i, X_j), either $\mathbf{D\text{-}SEP}(X_i, X_j)$ or $\mathbf{D\text{-}SEP}(X_j, X_i)$ provides such a separating set.

The key property exploited by FCI-SF is that all vertices required to construct valid separating sets are considered. Since Algorithm 2 examines subsets of $\mathbf{Possible\text{-}D\text{-}SEP}(X_i)$ for each variable X_i, it suffices to establish that, for every other variable X_j,

$$\mathbf{D\text{-}SEP}(X_i, X_j) \subseteq \mathbf{Possible\text{-}D\text{-}SEP}(X_i),$$

as stated in the following proposition.

Proposition 3 *Let P^* be a distribution over V that is faithful to a MAG $\mathcal{G}^* = (V, E)$, and let $\mathbf{H}_{new} \subseteq V$ denote a set of hidden variables. Let $\mathcal{P}_{old}$ be the PAG representing the Markov equivalence class of the latent projection of $\mathcal{G}^*$ onto $V \setminus \mathbf{H}_{new}$. Suppose that $\mathbf{H}_{new}$ becomes observable and that a perfect CI oracle is available. Let $\mathcal{P}_{new}$ denote the PAG obtained after Phase 2 of* FCI-SF *initialized with $\mathcal{P}_{old}$ and its respective separating sets. If X_i and X_j are observable and X_i is not an ancestor of X_j in $\mathcal{G}^*$, then every vertex in $\mathbf{D\text{-}SEP}(X_i, X_j)$ belongs to $\mathbf{Possible\text{-}D\text{-}SEP}(X_i)$ in $\mathcal{P}_{new}$.*

Proof The argument is identical to that of Spirtes et al. (2000), in their Lemma 6.4.1 (p. 417), and is omitted for brevity. □

As a consequence, we have the following corollary.

Corollary 4 *Let P^* be a distribution over V that is faithful to a MAG $\mathcal{G}^* = (V, E)$, and let $\mathbf{H}_{new} \subseteq V$ denote the set of initially hidden variables. Let $\mathcal{P}_{old}$ be the PAG representing the Markov equivalence class of the latent projection of $\mathcal{G}^*$ onto $V \setminus \mathbf{H}_{new}$. Suppose that $\mathbf{H}_{new}$ becomes observable and that a perfect CI oracle is available. Let $\mathcal{P}_{new}$ denote the PAG obtained after Phase 2 of* FCI-SF*. If X_i and X_j are observable and m-separable in $\mathcal{G}^*$, then there exists a separating set $\mathbf{S}$ such that in $\mathcal{P}_{new}$ either*

$$\mathbf{S} \subseteq \mathbf{Possible\text{-}D\text{-}SEP}(X_i) \text{ or } \mathbf{S} \subseteq \mathbf{Possible\text{-}D\text{-}SEP}(X_j).$$

Proof Assume, for contradiction, that no separating set contained in $\mathbf{Possible\text{-}D\text{-}SEP}(X_i)$ or $\mathbf{Possible\text{-}D\text{-}SEP}(X_j)$ renders X_i and X_j independent. By Proposition 3, the sets $\mathbf{D\text{-}SEP}(X_i, X_j)$ and $\mathbf{D\text{-}SEP}(X_j, X_i)$ are, respectively, subsets of $\mathbf{Possible\text{-}D\text{-}SEP}(X_i)$ and $\mathbf{Possible\text{-}D\text{-}SEP}(X_j)$. Our assumption, therefore, implies that neither $\mathbf{D\text{-}SEP}(X_i, X_j)$ nor $\mathbf{D\text{-}SEP}(X_j, X_i)$ separates X_i and X_j. Consequently, X_i and X_j must be adjacent in $\mathcal{G}$, contradicting the premise that they are m-separable. Hence, a separating set $\mathbf{S}$ satisfying the stated condition must exist, completing the proof. □

To establish the correctness of the algorithm, given its incremental nature, it is necessary to explicitly reference the PAG and the separating set map that FCI-SF receives at each iteration. In practice, the algorithm always receives as input the output from the previous batch of learned features. However, for clarity in constructing an inductive argument in the proof of Theorem 6, we assume that the input PAG provided to the algorithm is correct and that the accompanying separating set map contains, for each pair of m-separable features in the latent model, at least one valid separating set; concretely, this set would be the one that rendered the pair independent in the previous execution of the FCI-SF procedure whose output is the input in the new iteration.

Proposition 5 *Let the distribution P^* of V be faithful to a MAG $\mathcal{G}^* = (V, E)$, and assume that the subset $\mathbf{H}_{old} \subseteq V$ is initially latent. Suppose that some of the previously hidden variables, $\mathbf{H}_{new} \subseteq \mathbf{H}_{old}$, become observable and that a perfect oracle of CI information is available. Then, if the* FCI-SF *algorithm is initialized with a PAG representing the equivalence class of the latent projection of $\mathcal{G}$ onto $V \setminus \mathbf{H}_{old}$ along with its corresponding separating set map, the output of the algorithm is a PAG representing the Markov equivalence class of the latent projection of $\mathcal{G}$ onto $V \setminus (\mathbf{H}_{old} \setminus \mathbf{H}_{new})$.*

Proof Since the orientation and propagation phases remain unchanged from those in Spirtes et al. (2000) and Zhang (2008), it suffices to prove the proposition only for the adjacency phase of the FCI-SF algorithm.

Let $\mathcal{P}_{old}$ and $Sep_{old}[\cdot, \cdot]$ denote the input PAG and its separating set map, respectively, and let $\mathcal{P}_{new}$ represent the output of the algorithm. After the arrival of the previously hidden variables $\mathbf{H}_{new}$, denote as $\mathbf{O}_{new} = V \setminus (\mathbf{H}_{old} \setminus \mathbf{H}_{new})$ the new set of observed variables. First, it must be established that if variables X and Y are nonadjacent in $\mathcal{P}_{new}$, they are also nonadjacent in $L(\mathcal{G}, \mathbf{O}_{new})$. In Algorithm 1, before any pruning occurs, any pair of vertices is nonadjacent if and only if they were nonadjacent in $\mathcal{P}_{old}$. The independence preservation property (Proposition 1) guarantees that pairs nonadjacent in $\mathcal{P}_{old}$ remain nonadjacent in $L(\mathcal{G}, \mathbf{O}_{new})$. Furthermore, Algorithms 1 and 2 eliminate adjacencies only if the corresponding pair of features is m-separable in $L(\mathcal{G}, \mathbf{O}_{new})$. Therefore, any features nonadjacent in $\mathcal{P}_{new}$ must also be nonadjacent in $L(\mathcal{G}, \mathbf{O}_{new})$.

Conversely, assume features X and Y are adjacent in $\mathcal{P}_{new}$. Thus, X and Y are not m-separable in $L(\mathcal{G}, \mathbf{O}_{new})$ given either **Possible-D-SEP**(X) or **Possible-D-SEP**(Y). Therefore, by Corollary 4, X and Y cannot be m-separated in $L(\mathcal{G}, \mathbf{O}_{new})$ at all, and since $L(\mathcal{G}, \mathbf{O}_{new})$ is a MAG, X and Y must be adjacent in $L(\mathcal{G}, \mathbf{O}_{new})$. □

We then have the main result.

Theorem 6 *Let the distribution P^* of V be faithful to a MAG $\mathcal{G}^* = (V, E)$, where the feature set V is partitioned into k batches as $V = \mathbf{H}_1 \cup \mathbf{H}_2 \cup \cdots \cup \mathbf{H}_k$. Suppose these batches arrive incrementally, and at each batch i, the corresponding set of features $\mathbf{H}_i$ becomes observable. Further, assume that a perfect oracle of CI information is available. Then, if the proposed* FCI-SF *algorithm is executed sequentially each time a new batch of features arrives, its final output after processing all batches is a maximally informative PAG that correctly represents the causal structure encoded by $\mathcal{G}^*$.*

Proof When the first batch arrives, the algorithm receives the empty graph with an empty separation set map and behaves merely as the FCI, so its output is the PAG representing the Markov equivalence class of $L(\mathcal{G}, \mathbf{H}_1)$ and its respective separating set map.

Assume that for batch $i < k$ the result is the PAG representing the Markov equivalence class of $L(\mathcal{G}, \bigcup_{r=1}^{i} \mathbf{H}_r)$ and its respective separating set map. For the batch $i+1$, it would receive the PAG and the separating set map of the previous batch. Therefore, by Proposition 5 with $\mathbf{H}_{old} = \bigcup_{r=i+1}^{k} \mathbf{H_r}$ and $\mathbf{H}_{new} = \mathbf{H}_{old} \setminus \mathbf{H}_{i+1}$. The output of the FCI-SF is the PAG representing the Markov equivalence class of $L(\mathcal{G}, \bigcup_{r=1}^{i+1} \mathbf{H}_r)$ and its respective separating set map. Therefore, by induction, the output after processing all the batches is the Markov equivalence class of $\mathcal{G}^*$ and its respective separating set. □

In summary, it is possible to obtain the following corollary.

Corollary 7 *Let the distribution P^* of V be faithful to a MAG $\mathcal{G}^* = (V, E)$, where the feature set V is partitioned into k batches as $V = \mathbf{H}_1 \cup \mathbf{H}_2 \cup \cdots \cup \mathbf{H}_k$. Suppose these batches arrive incrementally, and at each batch i, the set of features $\mathbf{H}_i$ becomes observable. Assume further that a perfect oracle of CI information is available.*

Then, if the proposed FCI-SF *algorithm is executed sequentially as each batch arrives, its output after processing batch i is a maximally informative PAG representing the causal structure encoded by the latent projection of $\mathcal{G}^*$ onto the observed variables $\mathbf{O}_i = \mathbf{H}_1 \cup \cdots \cup \mathbf{H}_i$.*

Appendix B Synthetic Experiments Tables

This section reports the numerical values underlying the figures from the synthetic experiments presented in Section 6.1.

Table B1: Mean F1 scores for different numbers of observed variables and significance levels. Bold values indicate the best result for each algorithm and number of observed variables. This table reports the numerical values underlying the heatmaps in Fig. 4. All values are rounded to three decimal places.

S-CDFSF

|**O**|	0.1	0.05	0.025	0.0125	0.00625	0.003125
10	0.179	0.227	0.260	0.294	**0.294**	0.292
20	0.152	0.168	0.201	0.237	**0.250**	0.248
30	0.137	0.156	0.177	0.232	0.249	**0.249**
40	0.121	0.146	0.166	0.228	0.241	**0.250**
50	0.108	0.134	0.157	0.218	0.231	**0.243**

CSBS

|**O**|	0.1	0.05	0.025	0.0125	0.00625	0.003125
10	0.223	0.274	0.299	**0.326**	0.316	0.311
20	0.234	0.254	0.286	**0.311**	0.311	0.299
30	0.265	0.276	0.314	0.349	**0.349**	0.337
40	0.291	0.303	0.352	**0.392**	0.387	0.381
50	0.322	0.332	0.382	**0.435**	0.432	0.425

CSSU

|**O**|	0.1	0.05	0.025	0.0125	0.00625	0.003125
10	0.120	0.140	0.100	**0.150**	0.103	0.085
20	0.163	**0.175**	0.169	0.169	0.167	0.166
30	0.164	0.175	0.173	**0.175**	0.173	0.167
40	0.163	0.171	0.172	**0.175**	0.174	0.174
50	0.165	0.171	0.173	0.176	0.176	**0.177**

PRCDSF

|**O**|	0.1	0.05	0.025	0.0125	0.00625	0.003125
10	0.207	0.248	0.267	**0.299**	0.297	0.296
20	0.217	0.228	0.232	0.246	**0.254**	0.251
30	0.240	0.248	0.246	0.255	**0.258**	0.257
40	0.260	0.271	0.276	0.276	0.279	**0.280**
50	0.282	0.290	0.295	0.301	0.304	**0.306**

LiNGAM-SF

|**O**|	0.1	0.05	0.025	0.0125	0.00625	0.003125
10	0.209	0.251	0.280	**0.323**	0.319	0.226
20	0.182	0.242	0.279	**0.313**	0.310	0.226
30	0.227	0.284	0.325	**0.353**	0.332	0.247
40	0.267	0.329	0.374	**0.391**	0.345	0.277
50	0.310	0.375	0.411	**0.420**	0.360	0.301

FCI-STABLE

|**O**|	0.1	0.05	0.025	0.0125	0.00625	0.003125
10	0.359	0.424	0.444	0.457	**0.459**	0.457
20	0.324	0.382	0.393	**0.413**	0.404	0.401
30	0.337	0.385	0.400	**0.405**	0.404	0.391
40	0.355	0.405	**0.430**	0.429	0.425	0.414
50	0.376	0.431	0.459	**0.462**	0.459	0.448

FCI-SF

|**O**|	0.1	0.05	0.025	0.0125	0.00625	0.003125
10	0.359	0.424	0.444	0.457	**0.459**	0.457
20	0.326	0.382	0.394	**0.412**	0.405	0.401
30	0.341	0.387	0.401	**0.406**	0.404	0.394
40	0.358	0.406	**0.432**	0.430	0.427	0.420
50	0.380	0.436	0.462	**0.467**	0.462	0.453

Table B2: Mean F1 scores for different numbers of observed variables and sample sizes. Bold values indicate the best result for each algorithm and number of observed variables. This table reports the numerical values underlying the heatmaps in Fig. 5. All values are rounded to three decimal places.

S-CDFSF

|**O**|	500	1,000	2,000	4,000	8,000	16,000
10	0.220	0.228	0.255	0.246	0.252	**0.272**
20	0.169	0.186	0.196	0.198	**0.216**	0.213
30	0.167	0.169	0.178	0.189	**0.192**	0.187
40	0.151	0.154	0.163	0.165	**0.169**	0.162
50	0.135	0.142	0.145	**0.148**	0.145	0.141

CSBS

|**O**|	500	1,000	2,000	4,000	8,000	16,000
10	0.281	0.294	0.325	0.308	0.308	**0.338**
20	0.263	0.275	0.295	0.308	0.312	**0.333**
30	0.290	0.310	0.326	0.338	0.352	**0.366**
40	0.320	0.341	0.358	0.376	0.383	**0.397**
50	0.346	0.371	0.387	0.407	0.413	**0.425**

CSSU

|**O**|	500	1,000	2,000	4,000	8,000	16,000
10	0.096	0.158	0.116	0.154	**0.198**	0.157
20	0.161	0.160	**0.166**	0.164	0.151	0.149
30	0.167	**0.168**	0.167	0.153	0.147	0.141
40	**0.171**	0.164	0.160	0.159	0.149	0.142
50	**0.174**	0.169	0.161	0.160	0.145	0.147

PRCDSF

|**O**|	500	1,000	2,000	4,000	8,000	16,000
10	0.252	0.268	0.282	0.274	0.272	**0.305**
20	0.227	0.249	0.265	0.274	0.281	**0.295**
30	0.260	0.279	0.287	0.297	0.310	**0.316**
40	0.283	0.304	0.316	0.331	0.329	**0.342**
50	0.307	0.327	0.338	0.355	0.353	**0.361**

LiNGAM-SF

|**O**|	500	1,000	2,000	4,000	8,000	16,000
10	0.244	0.242	0.240	0.231	0.253	**0.257**
20	0.247	0.228	0.238	0.245	**0.248**	0.232
30	0.288	0.285	0.288	0.282	0.283	**0.290**
40	0.329	0.340	0.334	0.341	0.338	**0.345**
50	0.372	0.387	0.389	**0.392**	0.384	0.387

FCI-STABLE

|**O**|	500	1,000	2,000	4,000	8,000	16,000
10	0.435	0.475	0.513	0.504	0.519	**0.562**
20	0.384	0.417	0.460	0.484	0.513	**0.539**
30	0.407	0.439	0.478	0.488	0.508	**0.550**
40	0.420	0.458	0.486	0.511	0.535	**0.566**
50	0.451	0.491	0.514	0.541	0.554	**0.583**

FCI-SF

|**O**|	500	1,000	2,000	4,000	8,000	16,000
10	0.435	0.475	0.513	0.504	0.519	**0.562**
20	0.384	0.417	0.461	0.485	0.515	**0.541**
30	0.405	0.441	0.477	0.490	0.508	**0.549**
40	0.422	0.464	0.489	0.513	0.535	**0.562**
50	0.452	0.495	0.520	0.542	0.558	**0.583**

Table B3: HD, SHDe, F1 score, and Cohen's κ for different numbers of observed variables. Each entry reports the mean (±confidence interval half-width). All values are rounded to three decimal places. This table reports the numerical values underlying the results in Fig. 6.

Algorithm	$\lvert\mathbf{O}\rvert$	HD	SHDe	F1	Cohen's κ
S-CDFSF	10	4.890 (±0.230)	13.088 (±0.540)	0.273 (±0.008)	0.486 (±0.018)
	20	18.489 (±0.528)	47.848 (±1.349)	0.230 (±0.004)	0.409 (±0.009)
	30	35.938 (±0.725)	91.251 (±1.943)	0.198 (±0.003)	0.353 (±0.007)
	40	55.010 (±0.800)	137.710 (±2.179)	0.175 (±0.003)	0.313 (±0.006)
	50	74.383 (±0.817)	183.889 (±2.298)	0.154 (±0.003)	0.275 (±0.005)
CSBS	10	3.895 (±0.205)	11.928 (±0.486)	0.319 (±0.007)	0.588 (±0.016)
	20	11.970 (±0.452)	37.315 (±1.121)	0.324 (±0.004)	0.615 (±0.009)
	30	18.570 (±0.534)	60.204 (±1.364)	0.337 (±0.003)	0.653 (±0.007)
	40	22.659 (±0.509)	76.411 (±1.283)	0.355 (±0.002)	0.696 (±0.005)
	50	23.220 (±0.375)	83.714 (±0.903)	0.379 (±0.002)	0.748 (±0.004)
CSSU	10	6.074 (±0.259)	11.233 (±0.594)	0.096 (±0.010)	0.068 (±0.011)
	20	16.986 (±0.536)	38.258 (±1.367)	0.128 (±0.004)	0.227 (±0.009)
	30	27.076 (±0.629)	65.059 (±1.711)	0.149 (±0.004)	0.281 (±0.007)
	40	35.014 (±0.600)	86.573 (±1.685)	0.167 (±0.003)	0.322 (±0.006)
	50	39.540 (±0.451)	100.078 (±1.376)	0.189 (±0.003)	0.367 (±0.005)
PRCDSF	10	4.420 (±0.206)	12.699 (±0.497)	0.308 (±0.007)	0.558 (±0.015)
	20	15.433 (±0.463)	42.595 (±1.168)	0.300 (±0.003)	0.555 (±0.007)
	30	27.665 (±0.603)	73.901 (±1.493)	0.298 (±0.003)	0.562 (±0.006)
	40	39.080 (±0.608)	100.629 (±1.449)	0.301 (±0.002)	0.575 (±0.004)
	50	48.568 (±0.548)	121.069 (±1.173)	0.306 (±0.002)	0.593 (±0.003)
LiNGAM-SF	10	4.323 (±0.238)	10.714 (±0.540)	0.271 (±0.008)	0.443 (±0.019)
	20	13.008 (±0.523)	35.333 (±1.261)	0.258 (±0.005)	0.491 (±0.011)
	30	19.776 (±0.605)	57.360 (±1.547)	0.286 (±0.004)	0.553 (±0.008)
	40	23.575 (±0.565)	71.954 (±1.439)	0.317 (±0.003)	0.620 (±0.006)
	50	23.371 (±0.383)	77.380 (±0.970)	0.353 (±0.002)	0.696 (±0.004)
FCI-STABLE	10	3.885 (±0.205)	9.146 (±0.507)	0.319 (±0.007)	0.589 (±0.016)
	20	11.869 (±0.453)	33.039 (±1.180)	0.324 (±0.004)	0.617 (±0.009)
	30	18.305 (±0.534)	55.615 (±1.459)	0.339 (±0.003)	0.656 (±0.007)
	40	22.216 (±0.505)	72.456 (±1.401)	0.357 (±0.002)	0.700 (±0.005)
	50	22.469 (±0.366)	80.738 (±1.129)	0.382 (±0.002)	0.754 (±0.004)
FCI-SF	10	3.885 (±0.205)	9.146 (±0.507)	0.319 (±0.007)	0.589 (±0.016)
	20	11.881 (±0.454)	33.049 (±1.179)	0.324 (±0.004)	0.617 (±0.009)
	30	18.324 (±0.535)	55.574 (±1.463)	0.339 (±0.003)	0.656 (±0.007)
	40	22.264 (±0.505)	72.480 (±1.402)	0.357 (±0.002)	0.699 (±0.005)
	50	22.566 (±0.368)	80.504 (±1.129)	0.381 (±0.002)	0.753 (±0.004)

Table B4: Execution time, number of CI tests, average separating-set size, and number of learned edges for different numbers of observed variables. Each entry reports the mean (±confidence interval half-width). Execution times are rounded to four decimal places, while the remaining values are rounded to three decimal places. This table reports the numerical values underlying the results in Fig. 7.

Algorithm	$\lvert\mathbf{O}\rvert$	Execution time	# CI tests	Avg. separating-set size	# edges
S-CDFSF	10	0.3793 (±0.0230)	109.650 (±3.745)	0.257 (±0.016)	4.599 (±0.127)
	20	2.0157 (±0.0908)	734.405 (±25.551)	0.669 (±0.022)	14.594 (±0.224)
	30	5.4057 (±0.1952)	1986.938 (±67.634)	0.968 (±0.025)	27.041 (±0.299)
	40	10.1416 (±0.3240)	3807.315 (±116.404)	1.185 (±0.024)	41.471 (±0.353)
	50	16.5145 (±0.4785)	6265.530 (±183.424)	1.358 (±0.025)	56.743 (±0.413)
CSBS	10	0.2488 (±0.0141)	71.463 (±1.617)	0.199 (±0.014)	4.526 (±0.122)
	20	1.3758 (±0.0515)	284.315 (±6.583)	0.495 (±0.022)	13.645 (±0.200)
	30	3.4921 (±0.0986)	532.181 (±13.618)	0.676 (±0.027)	23.944 (±0.252)
	40	6.7697 (±0.1377)	771.113 (±21.573)	0.759 (±0.030)	34.845 (±0.295)
	50	10.6796 (±0.1820)	1049.369 (±38.532)	0.855 (±0.035)	45.695 (±0.322)
CSSU	10	17.2754 (±0.3418)	118.200 (±0.271)	1.126 (±0.003)	0.488 (±0.049)
	20	37.1274 (±0.8360)	334.383 (±10.403)	0.447 (±0.028)	3.071 (±0.118)
	30	55.0228 (±1.1096)	845.841 (±127.488)	0.917 (±0.058)	6.280 (±0.170)
	40	70.9265 (±1.3468)	2772.791 (±856.076)	1.692 (±0.097)	9.685 (±0.208)
	50	89.4885 (±5.7702)	28196.773 (±31175.922)	2.657 (±0.131)	13.355 (±0.240)
PRCDSF	10	0.4020 (±0.0241)	49.946 (±0.397)	0.752 (±0.030)	5.434 (±0.171)
	20	2.2549 (±0.0939)	178.305 (±1.630)	1.686 (±0.040)	18.775 (±0.353)
	30	5.7070 (±0.2031)	315.971 (±2.523)	2.378 (±0.040)	35.899 (±0.467)
	40	10.7354 (±0.3392)	452.209 (±2.976)	2.820 (±0.035)	54.899 (±0.562)
	50	17.2044 (±0.5165)	591.581 (±3.593)	3.136 (±0.033)	75.013 (±0.643)
LiNGAM-SF	10	0.2552 (±0.0207)	47.603 (±0.385)	0.000 (±0.000)	1.866 (±0.090)
	20	1.6308 (±0.0817)	161.043 (±1.216)	0.000 (±0.000)	6.463 (±0.164)
	30	4.3480 (±0.1630)	279.278 (±1.884)	0.000 (±0.000)	12.843 (±0.232)
	40	8.5193 (±0.2534)	396.165 (±2.232)	0.000 (±0.000)	20.331 (±0.270)
	50	13.6684 (±0.3655)	517.801 (±2.731)	0.000 (±0.000)	28.791 (±0.322)
FCI-STABLE	10	0.0164 (±0.0005)	69.173 (±1.844)	0.226 (±0.014)	4.509 (±0.122)
	20	0.0829 (±0.0021)	356.929 (±7.717)	0.424 (±0.014)	13.464 (±0.197)
	30	0.2102 (±0.0051)	890.671 (±16.207)	0.494 (±0.012)	23.599 (±0.250)
	40	0.4226 (±0.0091)	1680.651 (±27.040)	0.524 (±0.010)	34.270 (±0.296)
	50	0.7504 (±0.0152)	2731.411 (±40.110)	0.537 (±0.008)	44.919 (±0.322)
FCI-SF	10	0.0165 (±0.0005)	69.173 (±1.844)	0.226 (±0.014)	4.509 (±0.122)
	20	0.0705 (±0.0019)	285.203 (±6.377)	0.461 (±0.015)	13.474 (±0.198)
	30	0.1489 (±0.0038)	524.216 (±10.108)	0.541 (±0.014)	23.615 (±0.250)
	40	0.2642 (±0.0061)	769.540 (±14.280)	0.570 (±0.013)	34.318 (±0.294)
	50	0.4279 (±0.0092)	1010.561 (±16.131)	0.581 (±0.011)	44.986 (±0.322)

Appendix C Real World Data

C.1 Data Cleaning

General Data-Cleaning Procedure

All datasets underwent a common data-cleaning process designed to ensure consistency across sources while preserving the underlying structure relevant for causal discovery. First, we removed non-continuous columns. Datasets with continuous data have been selected for this study. However, it is important to note that the proposed method does not rely on CI tests. Therefore, it is not restricted by the nature of the data and can be applied to both continuous and discrete datasets. The selected datasets are predominantly continuous, but in some cases they contained text fields, discrete or categorical variables, or timestamp information. These columns were excluded because we decided to use continuous CI tests to keep the analysis simple and consistent across datasets. Some exceptions were made for percentage columns expressed as integers, which were retained due to their clear numerical interpretation. Importantly, removing non-continuous variables does not introduce methodological concerns: FCI-SF explicitly allows for the presence of latent variables, meaning that any excluded variable can be treated as unobserved without violating the assumptions of the algorithm.

We also removed rows containing NaN values to avoid introducing biases or inconsistencies during independence testing. Additionally, columns with more than 40% missing values were discarded. Several datasets exhibited sparsity, and applying strict row-wise filtering would have eliminated a large portion of the data. By removing highly incomplete columns instead, we preserved as many samples as possible while still ensuring the reliability of the learned causal structures.

Dataset-Specific Adjustments

Beyond these general steps, some datasets required additional modifications. For the Gas turbine emissions dataset, we used only the 2015 subset. In the brightness temperature and soil moisture datasets, the columns `S%` and `C%` originally appeared as percentages and were rescaled to lie between 0 and 1.

References

Aliferis CF, Statnikov A, Tsamardinos I, et al (2010) Local causal and Markov blanket induction for causal discovery and feature selection for classification Part I: Algorithms and empirical evaluation. J Mach Learn Res 11(1):171–234

Ankan A, Textor J (2024) pgmpy: A Python toolkit for Bayesian networks. J Mach Learn Res 25(265):1–8

Brouillard P, Squires C, Wahl J, et al (2025) The landscape of causal discovery data: Grounding causal discovery in real-world applications. In: Proceedings of the 4th Conference on Causal Learning and Reasoning, vol 275. PMLR, pp 834–873

Bühlmann P, Peters J, Ernest J (2014) Cam: Causal additive models, high-dimensional order search and penalized regression. Ann Stat 42(6):2526–2556. https://doi.org/10.1214/14-AOS1260

Candanedo LM, Feldheim V (2016) Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models. Energy Build 112:28–39. https://doi.org/https://doi.org/10.1016/j.enbuild.2015.11.071

Colliander A, Misra S, Cosh MH (2019) SMAPVEX16 manitoba PALS brightness temperature and soil moisture data, version 1. NASA National Snow and Ice Data Center Distributed Active Archive Center, https://doi.org/https://doi.org/10.5067/WM5FXV7WPQ6F

Colombo D, Maathuis MH (2014) Order-independent constraint-based causal structure learning. J Mach Learn Res 15(1):3741–3782

Colombo D, Maathuis MH, Kalisch M, et al (2012) Learning high-dimensional directed acyclic graphs with latent and selection variables. Ann Stat pp 294–321. https://doi.org/http://dx.doi.org/10.1214/11-AOS940

Commodity Futures Trading Commission (2024) Financial data for fcms. https://www.cftc.gov/MarketReports/financialfcmdata/index.htm

Cortez P, Cerdeira A, Almeida F, et al (2009) Modeling wine preferences by data mining from physicochemical properties. Decis Support Syst 47:547–553. https://doi.org/https://doi.org/10.1016/j.dss.2009.05.016

Cui R, Groot P, Heskes T (2016) Copula PC algorithm for causal discovery from mixed data. In: Machine Learning and Knowledge Discovery in Databases. Springer, pp 377–392, https://doi.org/https://doi.org/10.1007/978-3-319-46227-1_24

Ding W, Stepinski TF, Mu Y, et al (2011) Subkilometer crater discovery with boosting and transfer learning. ACM Trans Intell Syst Technol 2(4):1–22. https://doi.org/https://doi.org/10.1145/1989734.1989743

Faller PM, Vankadara LC, Mastakouri AA, et al (2024) Self-compatibility: Evaluating causal discovery without ground truth. In: Proceedings of the 27th International Conference on Artificial Intelligence and Statistics, vol 238. PMLR, pp 4132–4140

Giudice E, Kuipers J, Moffa G (2022) The dual PC algorithm for structure learning. In: Proceedings of the 11th International Conference on Probabilistic Graphical Models, vol 186. PMLR, pp 301–312

Guo X, Yang J (2017) Causal structure learning algorithm based on streaming features. In: Proceedings of the 2017 IEEE International Conference on Big Knowledge. IEEE, pp 192–197, https://doi.org/https://doi.org/10.1109/ICBK.2017.16

Harris N, Drton M (2013) PC algorithm for nonparanormal graphical models. The J Mach Learn Res 14(1):3365–3383

Jaber A, Kocaoglu M, Shanmugam K, et al (2020) Causal discovery from soft interventions with unknown targets: Characterization and learning. In: Advances in Neural Information Processing Systems, vol 33. Curran Associates, Inc., pp 9551–9561

Jia X, Kuo BC, Crawford MM (2013) Feature mining for hyperspectral image classification. Proc IEEE 101(3):676–697. https://doi.org/https://doi.org/10.1109/JPROC.2012.2229082

Kalisch M, Bühlman P (2007) Estimating high-dimensional directed acyclic graphs with the PC-algorithm. J Mach Learn Res 8(3)

Kaya H, Tüfekci P (2019) Predicting CO and NOx emissions from gas turbines: Novel data and a benchmark PEMS. https://doi.org/https://doi.org/10.3906/elk-1807-87

Le TD, Hoang T, Li J, et al (2016) A fast PC algorithm for high-dimensional causal discovery with multi-core PCs. IEEE/ACM Trans Comput Biol Bioinform 16(5):1483–1495. https://doi.org/https://doi.org/10.1109/TCBB.2016.2591526

Li J, Hu X, Tang J, et al (2015) Unsupervised streaming feature selection in social media. In: Proceedings of the 24th ACM International Conference on Information and Knowledge Management. ACM, pp 1041–1050, https://doi.org/https://doi.org/10.1145/2806416.2806501

Ling Z, Yu J, Zhang Y, et al (2025) Local causal discovery without causal sufficiency. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 39. AAAI Press, pp 18737–18745, https://doi.org/10.1609/aaai.v39i18.34062

Margaritis D, Thrun S (1999) Bayesian network induction via local neighborhoods. In: Advances in Neural Information Processing Systems, vol 12. The MIT Press, pp 505–511

Menegozzo G, Dall'Alba D, Fiorini P (2022) CIPCaD-Bench: Continuous industrial process datasets for benchmarking causal discovery methods. In: Proceedings of the 18th IEEE International Conference on Automation Science and Engineering. IEEE, pp 2124–2131, https://doi.org/10.1109/CASE49997.2022.9926420

Mooij JM, Magliacane S, Claassen T (2020) Joint causal inference from multiple contexts. J Mach Learn Res 21(99):1–108

NASA (2020) Fire particulate emissions from combined VIIRS and AHI data for Indonesia, 2015–2020. ORNL Distributed Active Archive Center, https://doi.org/https://doi.org/10.3334/ORNLDAAC/2118

Ogarrio JM, Spirtes P, Ramsey J (2016) A hybrid causal search algorithm for latent variable models. In: Proceedings of the 8th International Conference on Probabilistic Graphical Models, vol 52. PMLR, pp 368–379

Pearl J (2009) Causality, 2nd edn. Cambridge University Press, Cambridge

Pellet JP, Elisseeff A (2008) Finding latent causes in causal networks: An efficient approach based on Markov blankets. J Mach Learn Res 9:1523–1548

Peters J, Bühlmann P (2015) Structural intervention distance for evaluating causal graphs. Neural Comput 27(3):771–799. https://doi.org/https://doi.org/10.1162/NECO_a_00708

Richardson T, Spirtes P (2002) Ancestral graph Markov models. Ann Stat 30(4):962–1030. https://doi.org/https://doi.org/10.1214/aos/1031689015

Salehkaleybar S, Ghassami A, Kiyavash N, et al (2020) Learning linear non-Gaussian causal models in the presence of latent variables. J Mach Learn Res 21(39):1–24

Sondhi A, Shojaie A (2019) The reduced PC-algorithm: Improved causal structure learning in large random networks. J Mach Learn Res 20(164):1–31

Spirtes P (2001) An anytime algorithm for causal inference. In: Proceedings of the 8th International Workshop on Artificial Intelligence and Statistics, vol R3. PMLR, pp 278–285

Spirtes P, Glymour C (1991) An algorithm for fast recovery of sparse causal graphs. Soc Sci Comput Rev 9(1):62–72. https://doi.org/https://doi.org/10.1177/089443939100900106

Spirtes P, Meek C, Richardson T (1995) Causal inference in the presence of latent variables and selection bias. In: Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann, pp 499–506, https://doi.org/https://doi.org/10.7551/mitpress/2006.003.0009

Spirtes P, Glymour CN, Scheines R (2000) Causation, Prediction, and Search, 2nd edn. The MIT Press, Cambridge, MA

Tsamardinos I, Aliferis CF, Statnikov A (2003) Time and sample efficient discovery of Markov blankets and direct causal relations. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp 673–678, https://doi.org/https://doi.org/10.1145/956750.956838

Tsamardinos I, Brown LE, Aliferis CF (2006) The max-min hill-climbing Bayesian network structure learning algorithm. Mach Learn 65(1):31–78. https://doi.org/https://doi.org/10.1007/s10994-006-6889-7

Tsanas A, Little MA, McSharry PE, et al (2010) Accurate telemonitoring of Parkinson's disease progression by noninvasive speech tests. https://doi.org/https://doi.org/10.1109/TBME.2009.2036000

Verma TS (1993) Graphical aspects of causal models. Tech. Rep. R-191, Department of Computer Science, University of California, Los Angeles

Verma TS, Pearl J (1990) Equivalence and synthesis of causal models. In: Proceedings of the 6th Annual Conference on Uncertainty in Artificial Intelligence. Elsevier Science, pp 255–270

Wang D, Ding W, Yu K, et al (2013) Towards long-lead forecasting of extreme flood events: A data mining framework for precipitation cluster precursors identification. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp 1285–1293, https://doi.org/https://doi.org/10.1145/2487575.2488220

Wille A, Zimmermann P, Vranová E, et al (2004) Sparse graphical Gaussian modeling of the isoprenoid gene network in *Arabidopsis thaliana*. Genome Biol 5(11):R92. https://doi.org/https://doi.org/10.1186/gb-2004-5-11-r92

Wolberg WH, Street WN, Mangasarian OL (1994) Machine learning techniques to diagnose breast cancer from fine-needle aspirates. https://doi.org/https://doi.org/10.1016/0304-3835(94)90099-X

Yang J, Guo X, An N, et al (2018) Streaming feature-based causal structure learning algorithm with symmetrical uncertainty. Inf Sci 467:708–724. https://doi.org/https://doi.org/10.1016/j.ins.2018.04.076

Yang J, Jiang L, Shen A, et al (2023) Online streaming features causal discovery algorithm based on partial rank correlation. IEEE Trans Artif Intell 4(1):197–208. https://doi.org/https://doi.org/10.1109/TAI.2022.3151034

Yu K, Wu X, Wang H, et al (2010) Causal discovery from streaming features. In: 2010 IEEE International Conference on Data Mining. IEEE, pp 1163–1168, https://doi.org/https://doi.org/10.1109/ICDM.2010.82

Yu K, Wu X, Ding W, et al (2012) Exploring causal relationships with streaming features. Comput J 55(9):1103–1117. https://doi.org/https://doi.org/10.1093/comjnl/bxs032

Zarebavani B, Jafarinejad F, Hashemi M, et al (2020) cuPC: CUDA-based parallel PC algorithm for causal structure learning on GPU. IEEE Trans Parallel Distrib Syst 31(3):530–542. https://doi.org/https://doi.org/10.1109/TPDS.2019.2939126

Zhang C, Yu H, Wang G, et al (2025) LiNGAM-SF: Causal structural learning method with linear non-Gaussian acyclic models for streaming features. IEEE Trans Neural Netw Learn Syst 36(6):10693–10706. https://doi.org/https://doi.org/10.1109/TNNLS.2025.3529753

Zhang J (2008) On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. Artif Intell 172(16–17):1873–1896. https://doi.org/https://doi.org/10.1016/j.artint.2008.08.001

Zheng Y, Huang B, Chen W, et al (2024) Causal-learn: Causal discovery in Python. J Mach Learn Res 25(60):1–8