Journal of Machine Learning Research 24 (2023) 1-31 Submitted 12/22; Revised 11/23; Published 12/23

A Unified Experiment Design Approach for Cyclic and Acyclic Causal Models

Ehsan Mokhtarian EHSAN.MOKHTARIAN@EPFL.CH
School of Computer and Communication Sciences
EPFL
1015 Lausanne, Switzerland

Saber Salehkaleybar S.SALEHKALEYBAR@LIACS.LEIDENUNIV.NL
Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
2333 CA Leiden, Netherlands

AmirEmad Ghassami GHASSAMI@BU.EDU
Department of Mathematics and Statistics,
Boston University
Boston, MA 02215 USA

Negar Kiyavash NEGAR.KIYAVASH@EPFL.CH
College of Management of Technology
EPFL
1015 Lausanne, Switzerland

Editor: Jin Tian

Abstract

We study experiment design for unique identification of the causal graph of a simple SCM, where the graph may contain cycles. The presence of cycles in the structure introduces major challenges for experiment design as, unlike acyclic graphs, learning the skeleton of causal graphs with cycles may not be possible from merely the observational distribution. Furthermore, intervening on a variable in such graphs does not necessarily lead to orienting all the edges incident to it. In this paper, we propose an experiment design approach that can learn both cyclic and acyclic graphs and hence, unifies the task of experiment design for both types of graphs. We provide a lower bound on the number of experiments required to guarantee the unique identification of the causal graph in the worst case, showing that the proposed approach is order-optimal in terms of the number of experiments up to an additive logarithmic term. Moreover, we extend our result to the setting where the size of each experiment is bounded by a constant. For this case, we show that our approach is optimal in terms of the size of the largest experiment required for uniquely identifying the causal graph in the worst case.

Keywords: experiment design, cyclic graphs, cyclic SCMs, causal structure learning

©2023 Ehsan Mokhtarian, Saber Salehkaleybar, AmirEmad Ghassami, and Negar Kiyavash.

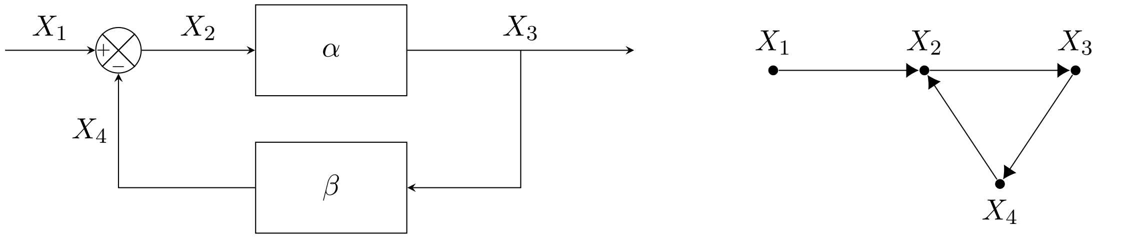
License: CC-BY 4.0, see `https://creativecommons.org/licenses/by/4.0/`. Attribution requirements are provided at `http://jmlr.org/papers/v24/22-1425.html`.

1. Introduction

One of the fundamental undertakings of empirical sciences is recovering causal relationships among variables of interest in a system (Pearl, 2009). Causal relationships are commonly represented by a directed graph (DG), referred to as the causal graph of the system. In such a representation, a directed edge from variable X to variable Y denotes that X is a direct cause of Y. In causal structure learning literature, it is predominantly assumed that the causal graph is a directed acyclic graph (DAG). However, in many real-life systems, feedback loops exist among the variables to ensure stability. Such feedback loops create cycles in the causal graph of the system when temporal dynamics are sampled at a low rate or when modeling a system's equilibrium states (Bongers et al., 2021).

As an example of a system with a feedback loop, consider the closed-loop control system in Figure 1a with four variables X_1, X_2, X_3, X_4. Figure 1b illustrates the causal graph among these variables, which is cyclic due to the feedback loop of the control system. We shall later revisit this example in more detail in Example 1. Another example appears in gene regulatory networks (GRN), where a collection of biological regulators interact with each other in order to determine the expression level of proteins. A GRN can be represented by a DG, where the vertices are the genes, and there is a directed edge from gene X to gene Y if activating gene X may directly activate or suppress gene Y. Figure 2 depicts a sub-network of Yeast's GRN (Schaffter et al., 2011), where the label of each vertex is the name of the corresponding gene. This causal graph is cyclic as it contains a directed cycle (the edges in red). Another example is in human circadian rhythms, where genes such as Bmal1, Per2, and Cry1 form feedback loops, creating cycles that control our daily biological clock (Pett et al., 2018).

Allowing cycles introduces major challenges to structure learning from observational data. For instance, for DAGs, the skeleton of the graph (i.e., the undirected graph obtained by removing the directions of the edges) can be learned from observational data (Spirtes et al., 2000; Pearl, 2009; Mokhtarian et al., 2021; Akbari et al., 2021). As we shall discuss in Section 3.1, for cyclic DGs, we can only learn a supergraph of the skeleton. Another fundamental challenge is as follows. For DAGs, if data is generated from a structural causal model (SCM), d-Markov property holds, i.e., the joint distribution over the variables contains all the conditional independencies encoded by the d-separation relations in the



(a) A closed-loop control system.

(b) Causal graph of the system.

Figure 1: An example with a feedback loop in control systems that can be modeled with a cyclic SCM (Example 1).

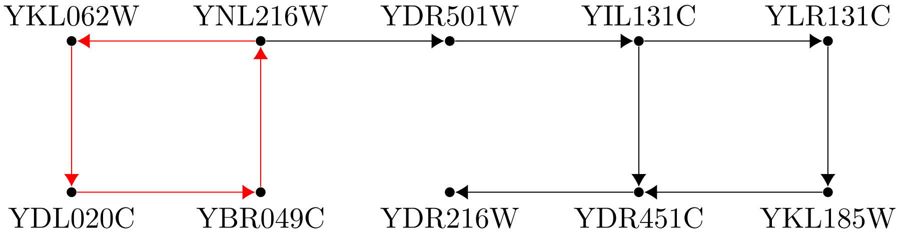


Figure 2: A sub-network of Yeast's gene regulatory network that contains a directed cycle of length 4 (the edges in red).

graph. For cyclic DGs, this property holds only in specific cases, such as linear systems with continuous variables (see Section 2.3 for a detailed discussion). In short, observational data is far less informative for structure learning in the case of cyclic graphs.

To gain more insight into the underlying causal graph, the gold standard is to perform *experiments* in the system. That is, to intervene on a subset of variables and study the effect of such intervention on the resulting interventional distribution. We refer to the problem of designing a set of experiments sufficient for learning the underlying causal graph as *experiment design problem*. As performing experiments are often costly and time-consuming, it is desirable to minimize the number of necessary experiments in the design.

Experiment design has been studied extensively for DAGs (see related work in Section 8). Unfortunately, the findings for DAGs are not directly applicable to graphs with cycles. For instance, in DAGs, an intervention on a subset of the vertices orients all the edges between the subset and the rest of the variables. In Section 3.2, we show that in cyclic DGs, performing experiments in some cases neither leads to learning the presence of edges nor orientating them. This shows that entirely new techniques are required to develop algorithms for the experiment design problem in the presence of cycles.

To the best of our knowledge, this paper proposes the first unified framework for experiment design for cyclic and acyclic graphs. Our main contributions are as follows.

- We provide a two-stage experiment design algorithm for learning DG $\mathcal{G}$, the causal graph of a *simple SCM* (Definition 5). In the first stage, we extend the so-called *separating systems* to *colored separating systems* (Definition 26), which we utilize to design a set of experiments for learning the strongly connected components (SCC) of $\mathcal{G}$ (Algorithm 1). In the second stage, we introduce the novel concept of *lifted separating systems* (Definition 35), which are defined based on the SCCs of the graph. As we mentioned before, performing an experiment does not necessarily lead to learning the presence of edges or orientating them. However, we show that by performing experiments on the elements of a lifted separating system, we can learn the set of parents of each variable and therefore, exactly recover $\mathcal{G}$ (Algorithm 2).

- We provide lower bounds on the number of experiments and the size of the largest experiment that leads to the unique identification of $\mathcal{G}$ for both adaptive and non-adaptive designs in the worst case for any fixed value of $\zeta_{\max}(\mathcal{G})$, where $\zeta_{\max}(\mathcal{G})$ denotes the size

	Max experiment size	**Number of experiments**
Unbounded-size alg.	$n-1$	$2\lceil\log_2(\chi(\mathcal{G}_r^{obs}))\rceil + \zeta_{\max}(\mathcal{G})$
Bounded-size alg.	$\zeta_{\max}(\mathcal{G}) - 1 \leq M < n$	$\lceil \frac{n}{M} \rceil \lceil \log_{\lceil \frac{n}{M} \rceil} n \rceil + \zeta_{\max}(\mathcal{G})(1 + \lfloor \frac{n-\zeta_{\max}(\mathcal{G})-1}{M-\zeta_{\max}(\mathcal{G})+2} \rfloor)$
Lower bound	$\zeta_{\max}(\mathcal{G}) - 1$	$\zeta_{\max}(\mathcal{G})$

Table 1: Main contributions of the paper. The first two rows provide the achievable bounds on the number of performed experiments for our proposed unbounded-sized (Section 5) and bounded-size (Section 6) experiment design algorithms. The last row represents our lower bounds on the number of experiments (Theorem 24) and the size of the largest experiment (Theorem 22) that lead to the unique identification of $\mathcal{G}$ in the worst case. The number of variables and the size of the largest SCC of $\mathcal{G}$ are denoted by n and $\zeta_{\max}(\mathcal{G})$, respectively. $\mathcal{G}_r^{obs}$ denotes the skeleton of a graph that can be learned from the observational distribution (Definition 15), and $\chi(\mathcal{G}_r^{obs})$ is its coloring number.

of the largest SCC of $\mathcal{G}$. Specifically, we show that in the worst case, $\mathcal{G}$ cannot be identified by performing experiments with size less than $\zeta_{\max}(\mathcal{G}) - 1$ (Theorem 22) or the number of experiments less than $\zeta_{\max}(\mathcal{G})$ (Theorem 24). Additionally, we show that the former bound is tight (Corollary 47), and the latter differs from our achievable bound (Equation 8) by an additive logarithmic term, which demonstrates the order-optimality of our proposed method. Note that in acyclic models, the lower bound on the size of the experiments is one since singleton experiments are always sufficient for learning a DAG.

- Finally, we consider a setup where the size of each designed experiment is bounded by a constant (Section 6). We provide an extension of our approach to this setting and present an upper bound on the number of designed experiments (Equation 11). In particular, we formulate the construction of bounded-size lifted separating systems as a combinatorial optimization problem and propose a greedy method for solving it (Theorem 46).

Table 1 summarizes the main contributions of the paper. The remainder of the paper is organized as follows. In Section 2, we review the preliminaries, introduce notations and assumptions, and formally describe the experiment design problem in the presence of cycles. In Section 3, we discuss two fundamental challenges of causal discovery from observation or interventional distributions in the presence of cycles. In Section 4, we present lower bounds on the size and number of experiments required for the unique identification of the causal graph. In Section 5, we propose the two stages of our experiment design algorithm. We then generalize our results in Section 6 to the setting where the size of each designed experiment is bounded by a constant. In Section 7, we provide a set of simulations over syntactic data sets to illustrate the performance of our method in practice. In Section 8, we review and discuss related work. Finally, in Section 9, we conclude the paper and discuss potential future work.

2. Preliminaries and Problem Description

Throughout the paper, we denote random variables by capital letters (e.g., X), sets of variables by bold letters (e.g., $\mathbf{X}$), and graphs by calligraphic letters (e.g., $\mathcal{G}$).

2.1 Preliminary Graph Definitions

A *directed graph* (DG) is a graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is a set of variables and $\mathbf{E}$ is a set of directed edges between the variables in $\mathbf{V}$. We denote a directed edge from X to Y by (X, Y), where X is called a *parent* of Y and Y a *child* of X. Further, *neighbors* of a variable is the union of parents and children of that variable. In this paper, we consider DGs without self-loop, i.e., $(X, X) \notin \mathbf{E}$ for all $X \in \mathbf{V}$. However, a DG can have multiple edges (at most one in each direction), i.e., it is possible that $(X, Y) \in \mathbf{E}$ and $(Y, X) \in \mathbf{E}$. Similarly, an undirected graph is a graph with undirected edges. We denote an undirected edge between two distinct variables X and Y by $\{X, Y\}$. The *skeleton* of a DG $\mathcal{G}$ is an undirected graph $(\mathbf{V}, \mathbf{E}')$, where there is an undirected edge $\{X, Y\}$ in $\mathbf{E}'$ if X and Y are neighbors, that is, either $(X, Y) \in \mathbf{E}$ or $(Y, X) \in \mathbf{E}$. A *directed acyclic graph* (DAG) is a DG with no cycles.

A *vertex coloring* for an undirected graph $\mathcal{G}$ is an assignment of colors to the vertices, such that no two adjacent vertices are of the same color. *Chromatic number* of $\mathcal{G}$, denoted by $\chi(\mathcal{G})$, is the smallest number of colors needed for a vertex coloring of $\mathcal{G}$.

Suppose $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ is a DG. A path $(X_1, X_2, \cdots, X_k)$ in $\mathcal{G}$ is called a *directed path* from X_1 to X_k if $(X_i, X_{i+1}) \in \mathbf{E}$ for all $1 \leq i < k$. Variable X is called an *ancestor* of Y and Y a *descendant* of X if there exists a directed path from X to Y in $\mathcal{G}$. Note that X is an ancestor and a descendant of itself. A non-endpoint vertex X on a path is called a *collider* if both of the edges incident to X on the path have an arrowhead at X. A variable Y is *strongly connected* to variable X if Y is both an ancestor and a descendant of X. We denote the set of parents, children, neighbors, descendants, ancestors, and strongly connected variables of X in $\mathcal{G}$ by $Pa_{\mathcal{G}}(X)$, $Ch_{\mathcal{G}}(X)$, $Ne_{\mathcal{G}}(X)$, $De_{\mathcal{G}}(X)$, $Anc_{\mathcal{G}}(X)$, and $SCC_{\mathcal{G}}(X)$, respectively. We will also apply these definitions disjunctively to sets of variables, e.g., $Pa_{\mathcal{G}}(\mathbf{X}) = \bigcup_{X \in \mathbf{X}} Pa_{\mathcal{G}}(X)$ or $Anc_{\mathcal{G}}(\mathbf{X}) = \bigcup_{X \in \mathbf{X}} Anc_{\mathcal{G}}(X)$.

Definition 1 (SCC) *Strongly connected variables of $\mathcal{G}$ partition $\mathbf{V}$ into so-called,* strongly connected components *(SCCs); two variables are strongly connected if and only if they are in the same SCC. We denote the size of the largest SCC of $\mathcal{G}$ by $\zeta_{\max}(\mathcal{G})$.*

2.2 Generative Model

Structural causal models (SCMs) are commonly used to describe the causal mechanisms of a system (Pearl, 2009).

Definition 2 (SCM) *An SCM is a tuple $\mathcal{M} = \langle \mathbf{V}, \mathbf{U}, \mathbf{F}, P(\mathbf{U}) \rangle$, where $\mathbf{V}$ is a set of endogenous variables, $\mathbf{U}$ is a set of exogenous variables with the joint distribution $P(\mathbf{U})$ where the variables in $\mathbf{U}$ are assumed to be jointly independent, and $\mathbf{F}$ is a set of functions $\{f_X\}_{X \in \mathbf{V}}$ such that $X = f_X(Pa(X), \mathbf{U}^X)$, where $Pa(X) \subseteq \mathbf{V} \setminus \{X\}$ and $\mathbf{U}^X \subseteq \mathbf{U}$.*

Let $\mathcal{M} = \langle \mathbf{V}, \mathbf{U}, \mathbf{F}, P(\mathbf{U}) \rangle$ be an SCM. The assumption of causal sufficiency holds for $\mathcal{M}$ if for any two distinct variables $X, Y \in \mathbf{V}$, $\mathbf{U}^X \cap \mathbf{U}^Y = \varnothing$. In this paper, we assume causal sufficiency. Under the causal sufficiency assumption, the causal graph of $\mathcal{M}$ is a DG over $\mathbf{V}$ with directed edges from $Pa(X)$ to X for each variable $X \in \mathbf{V}$.

Definition 3 (Acyclic SCM) *An SCM is called acyclic if the corresponding causal graph is a DAG. Acyclic SCMs are also known as recursive SEMs.*

Remark 4 *The definition of SCM does not require the causal graph to be acyclic. Acyclicity is often added (or implicitly assumed) in the literature.*

Acyclic SCMs have been widely studied in the past few decades because of their convenient properties. For instance, they always induce unique observational, interventional, and counterfactual distributions (Pearl, 2009). This is not necessarily the case for cyclic SCMs since cycles lead to various complications pertaining to solvability issues. Bongers et al. (2021) introduced *simple SCMs*, a subclass of SCMs (cyclic or acyclic), which retain most of the convenient properties of acyclic SCMs.

Definition 5 (Simple SCM) *An SCM is simple if any subset of its structural equations can be solved uniquely for its associated variables in terms of the other variables that appear in these equations.*

We refer the interested reader to Bongers et al. (2021) for a more detailed definition of simple SCMs. The following result provides a few important properties of simple SCMs.

Proposition 6 (Bongers et al. 2021) *Simple SCMs always have uniquely defined observational, interventional, and counterfactual distributions.*

Example 1 (Simple SCM) *Consider the control system shown in Figure 1 with four variables X_1, X_2, X_3, X_4. The followings are structural equations modeling the control system.*

$$X_1 = U_1, \quad X_2 = X_1 - X_4 + U_2, \quad X_3 = \alpha X_2 + U_3, \quad X_4 = \beta X_3 + U_4, \tag{1}$$

where α and β are two constants such that $\alpha\beta \neq -1$, and U_1, U_2, U_3, U_4 are independent noise variables (U_1 could be viewed as the input to the system and U_2, U_3, U_4 as the noise for each state variable in the system). This SCM is simple as any subset of the equations in (1) *can be solved uniquely for its associated variables in terms of the other variables that appear in these equations. Proposition 6 implies that observational, interventional, and counterfactual distributions all exist and are unique for this SCM. For instance, suppose we perform an intervention on variable X_4 by replacing the corresponding structural equation with $X_4 = U_4'$, where U_4' is an independent noise variable. This will remove the feedback loop and the variables in the system will be uniquely determined as follows.*

$$X_1 = U_1, \quad X_2 = U_1 - U_4' + U_2, \quad X_3 = \alpha(U_1 - U_4' + U_2) + U_3, \quad X_4 = U_4'. \tag{2}$$

2.3 From d-separation to σ-separation

For three disjoint subsets $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ of variables with the joint distribution P, conditional independence (CI) $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z})_P$ denotes that $\mathbf{X}$ and $\mathbf{Y}$ are independent conditioned on $\mathbf{Z}$, i.e., $P(\mathbf{X}, \mathbf{Y} \mid \mathbf{Z}) = P(\mathbf{X} \mid \mathbf{Z}) P(\mathbf{Y} \mid \mathbf{Z})$.

In the following, we formally define d-separation and σ-separation for DGs.

Definition 7 (d-separation) *Suppose $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ is a DG, X and Y are two distinct variables in $\mathbf{V}$, and $\mathbf{S} \subseteq \mathbf{V} \setminus \{X, Y\}$. A path $\mathcal{P} = (X, Z_1, \cdots, Z_k, Y)$ between X and Y in $\mathcal{G}$ is d-blocked by $\mathbf{S}$ if there exists $1 \leq i \leq k$ such that*

- *Z_i is a collider on $\mathcal{P}$ and $Z_i \notin Anc_{\mathcal{G}}(\mathbf{S})$, or*
- *Z_i is not a collider on $\mathcal{P}$ and $Z_i \in \mathbf{S}$.*

We say $\mathbf{S}$ d-separates X and Y in $\mathcal{G}$ and denote it by $(X \perp\!\!\!\perp_d Y|\mathbf{S})_{\mathcal{G}}$ if all the paths in $\mathcal{G}$ between X and Y are d-blocked by $\mathbf{S}$. For three disjoint subsets $\mathbf{X}, \mathbf{Y}, \mathbf{S}$ in $\mathbf{V}$, we say $\mathbf{S}$ d-separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{G}$, denoted by $(\mathbf{X} \perp\!\!\!\perp_d \mathbf{Y}|\mathbf{S})_{\mathcal{G}}$, if for any $X \in \mathbf{X}$ and $Y \in \mathbf{Y}$, $(X \perp\!\!\!\perp_d Y|\mathbf{S})_{\mathcal{G}}$.

Definition 8 (σ-separation) *Suppose $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ is a DG, X and Y are two distinct variables in $\mathbf{V}$, and $\mathbf{S} \subseteq \mathbf{V} \setminus \{X, Y\}$. A path $\mathcal{P} = (X = Z_0, Z_1, \cdots, Z_k, Z_{k+1} = Y)$ between X and Y in $\mathcal{G}$ is σ-blocked by $\mathbf{S}$ if there exists $1 \leq i \leq k$ such that*

- *Z_i is a collider on $\mathcal{P}$ and $Z_i \notin Anc_{\mathcal{G}}(\mathbf{S})$, or*
- *Z_i is not a collider on $\mathcal{P}$, $Z_i \in \mathbf{S}$, and either $Z_i \to Z_{i+1}$ and $Z_{i+1} \notin SCC_{\mathcal{G}}(Z_i)$, or $Z_{i-1} \leftarrow Z_i$ and $Z_{i-1} \notin SCC_{\mathcal{G}}(Z_i)$.*

We say $\mathbf{S}$ σ-separates X and Y in $\mathcal{G}$, denoted by $(X \perp\!\!\!\perp_\sigma Y|\mathbf{S})_{\mathcal{G}}$, if all the paths in $\mathcal{G}$ between X and Y are σ-blocked by $\mathbf{S}$. For three disjoint subsets $\mathbf{X}, \mathbf{Y}, \mathbf{S}$ in $\mathbf{V}$, we say $\mathbf{S}$ σ-separates $\mathbf{X}$ and $\mathbf{Y}$ in $\mathcal{G}$, denoted by $(\mathbf{X} \perp\!\!\!\perp_\sigma \mathbf{Y}|\mathbf{S})_{\mathcal{G}}$, if for any $X \in \mathbf{X}$ and $Y \in \mathbf{Y}$, $(X \perp\!\!\!\perp_\sigma Y|\mathbf{S})_{\mathcal{G}}$.

Remark 9 *for DAGs, σ-separation and d-separation are equivalent. That is, for three disjoint subsets $\mathbf{X}, \mathbf{Y}, \mathbf{S}$, if the σ-separation $(\mathbf{X} \perp\!\!\!\perp_\sigma \mathbf{Y}|\mathbf{Z})_{\mathcal{G}}$ holds, then the d-separation $(\mathbf{X} \perp\!\!\!\perp_d \mathbf{Y}|\mathbf{Z})_{\mathcal{G}}$ holds and visa versa. However, for cyclic DGs, the reverse direction does not necessarily hold.*

For ease of representation, we introduce letter r to stand for either d (as in d-separation) or σ (as in σ-separation). Next, we formally define r-independence model, r-Markov equivalence class, r-Markov property, and r-faithfulness.

Definition 10 ($\mathbf{IM}_r(\mathcal{G})$) *For a DG $\mathcal{G}$, the r-independence model $IM_r(\mathcal{G})$ is defined as the set of r-separations of $\mathcal{G}$. That is,*

$$IM_r(\mathcal{G}) = \{(X, Y, \mathbf{Z}) \mid X, Y \in \mathbf{V}, \mathbf{Z} \subseteq \mathbf{V} \setminus \{X, Y\}, (X \perp\!\!\!\perp_r Y|\mathbf{Z})_{\mathcal{G}}\}.$$

When $\mathcal{G}$ is a DAG, given their equivalence, we drop subscripts d and σ in d-separation and σ-separation notations, respectively, and refer to the independence model as $\mathrm{IM}(\mathcal{G})$ since $\mathrm{IM}_d(\mathcal{G}) = \mathrm{IM}_\sigma(\mathcal{G})$.

Definition 11 (r-MEC) *Two DGs with identical r-independence models are called to be r-Markov equivalent. We denote by $[\mathcal{G}]^r$ the r-Markov equivalence class (r-MEC) of $\mathcal{G}$, i.e., the set of r-Markov equivalent DGs of $\mathcal{G}$.*

Definition 12 (r-Markov property, r-faithfulness) *A distribution P satisfies r-Markov property with respect to a DG $\mathcal{G}$ if for any r-separation $(\mathbf{X} \perp\!\!\!\perp_r \mathbf{Y}|\mathbf{Z})_{\mathcal{G}}$ in $\mathcal{G}$, the CI $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})_P$ holds in P. Similarly, a distribution P satisfies r-faithfulness with respect to a DG $\mathcal{G}$ if for any CI $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})_P$ in P, the r-separation $(\mathbf{X} \perp\!\!\!\perp_r \mathbf{Y}|\mathbf{Z})_{\mathcal{G}}$ holds in $\mathcal{G}$.*

Suppose $\mathcal{M} = \langle \mathbf{V}, \mathbf{U}, \mathbf{F}, P(\mathbf{U})\rangle$ is a simple SCM with observational distribution $P^{\mathcal{M}}(\mathbf{V})$ and causal graph $\mathcal{G}$. We often drop the superscript $\mathcal{M}$ when it is clear from the context. It has been shown that P always satisfies σ-Markov property with respect to $\mathcal{G}$. However, the d-Markov property holds in specific settings, e.g., acyclic SCMs, SCMs with continuous variables and linear relations, or SCMs with discrete variables (Mooij and Claassen, 2020; Forré and Mooij, 2017). On the other hand, σ-faithfulness is a stronger assumption than d-faithfulness due to Remark 9.

2.4 Intervention and Experiment

Suppose $\mathcal{M} = \langle \mathbf{V}, \mathbf{U}, \mathbf{F}, P(\mathbf{U})\rangle$ is an SCM. A *full-support hard intervention* on a subset $\mathbf{I} \subseteq \mathbf{V}$, denoted by $do(\mathbf{I})$, converts $\mathcal{M}$ to a new SCM $\mathcal{M}_{do(\mathbf{I})} = \langle \mathbf{V}, \mathbf{U}, \mathbf{F}', P(\mathbf{U})\rangle$, where for each $X \in \mathbf{I}$, the structural assignment of X in $\mathbf{F}$ is replaced by $X = \xi_X$ in $\mathbf{F}'$, where ξ_X is a random variable whose support is the same as the support of X and is independent of all other random variables in the system. We denote the corresponding interventional distribution (i.e., the distribution of $\mathcal{M}_{do(\mathbf{I})}$) by $P_{do(\mathbf{I})}$.

Proposition 13 (Bongers et al. 2021) *If $\mathcal{M} = \langle \mathbf{V}, \mathbf{U}, \mathbf{F}, P(\mathbf{U})\rangle$ is a simple SCM, then for any $\mathbf{I} \subseteq \mathbf{V}$, SCM $\mathcal{M}_{do(\mathbf{I})}$ is also a simple SCM.*

After intervening on $\mathbf{I}$, the variables in $\mathbf{I}$ are no longer functions of other variables in $\mathbf{V}$. Hence, the corresponding causal graph of $\mathcal{M}_{do(\mathbf{I})}$ can be obtained from graph $\mathcal{G}$ by removing the incoming edges of the variables in $\mathbf{I}$. We denote the resulting graph by $\mathcal{G}_{\overline{\mathbf{I}}}$. An *experiment* on a target set $\mathbf{I}$ is the act of conducting a full-support hard intervention on $\mathbf{I}$ and obtaining the interventional distribution $P_{do(\mathbf{I})}$.

Definition 14 ($\mathcal{I}$-r-MEC) *Suppose $\mathcal{I}$ is a collection of subsets of $\mathbf{V}$ (can include the empty set). Two DGs $\mathcal{G}$ and $\mathcal{H}$ are $\mathcal{I}$-r-Markov equivalent if $IM_r(\mathcal{G}_{\overline{\mathbf{I}}}) = IM_r(\mathcal{H}_{\overline{\mathbf{I}}})$ for each $\mathbf{I} \in \mathcal{I}$. We denote by $[\mathcal{G}]^r_{\mathcal{I}}$ the $\mathcal{I}$-r-Markov equivalent class of $\mathcal{G}$, i.e., the set of $\mathcal{I}$-r-Markov equivalent DGs of $\mathcal{G}$.*

This definition implies that it is impossible to distinguish two $\mathcal{I}$-r-Markov equivalent graphs by the r-separations of the graphs resulting from experiments on the elements of $\mathcal{I}$.

2.5 Problem Description

Consider a simple SCM $\mathcal{M} = \langle \mathbf{V}, \mathbf{U}, \mathbf{F}, P(\mathbf{U})\rangle$ with observational distribution $P^{\mathcal{M}}(\mathbf{V})$ and causal graph $\mathcal{G}$. We assume causal sufficiency, in which case $\mathcal{G}$ is a DG.[1] As discussed earlier, in simple SCMs, σ-Markov property always holds (even in non-linear systems with

[1] Note that DGs cannot represent the presence of hidden confounders. Without causal sufficiency, the causal graph can be represented by a directed mixed graph (DMG) that contains bidirected edges to indicate the presence of hidden confounders.

continuous variables), while d-Markov property holds in certain settings. On the other hand, σ-faithfulness is a stronger assumption than d-faithfulness. In this paper, we consider the following two scenarios.

- **Scenario 1:** $P^{\mathcal{M}}$ satisfies d-Markov property and d-faithfulness w.r.t. $\mathcal{G}$. In this case, CI relations are equivalent to d-separations. That is, $(X \perp\!\!\!\perp_d Y|\mathbf{Z})_{\mathcal{G}} \iff (X \perp\!\!\!\perp Y|\mathbf{Z})_P$.
- **Scenario 2:** $P^{\mathcal{M}}$ satisfies σ-faithfulness w.r.t. $\mathcal{G}$. In this case, CI relations are equivalent to σ-separations. That is, $(X \perp\!\!\!\perp_\sigma Y|\mathbf{Z})_{\mathcal{G}} \iff (X \perp\!\!\!\perp Y|\mathbf{Z})_P$.

Note that if $\mathcal{G}$ is a DAG, the aforementioned scenarios are the same. However, if there are cycles in $\mathcal{G}$, the two scenarios are not necessarily equivalent.

Our goal in this paper is to design a set of experiments for learning $\mathcal{G}$ under Scenario 1 or Scenario 2. That is, to introduce a collection of subsets $\mathcal{I}$ such that $[\mathcal{G}]^r_{\mathcal{I}} = \{\mathcal{G}\}$. Additionally, as performing experiments can be costly, we aim to minimize the number of necessary experiments. Surprisingly, our approaches for both scenarios coincide, but the proof techniques differ.

3. Challenges of Experiment Design in Presence of Cycles

In this section, we discuss some of the challenges pertaining to learning DGs in the presence of cycles. In Subsection 3.1, we show that, unlike DAGs, we cannot learn the skeleton of a DG without performing experiments, i.e., from merely the observational distribution. In Subsection 3.2, we argue that even performing all size-one experiments (singleton experiments) does not suffice to learn $\mathcal{G}$ in some cases.

3.1 Skeleton of a DG is not Learnable from Observational Distribution

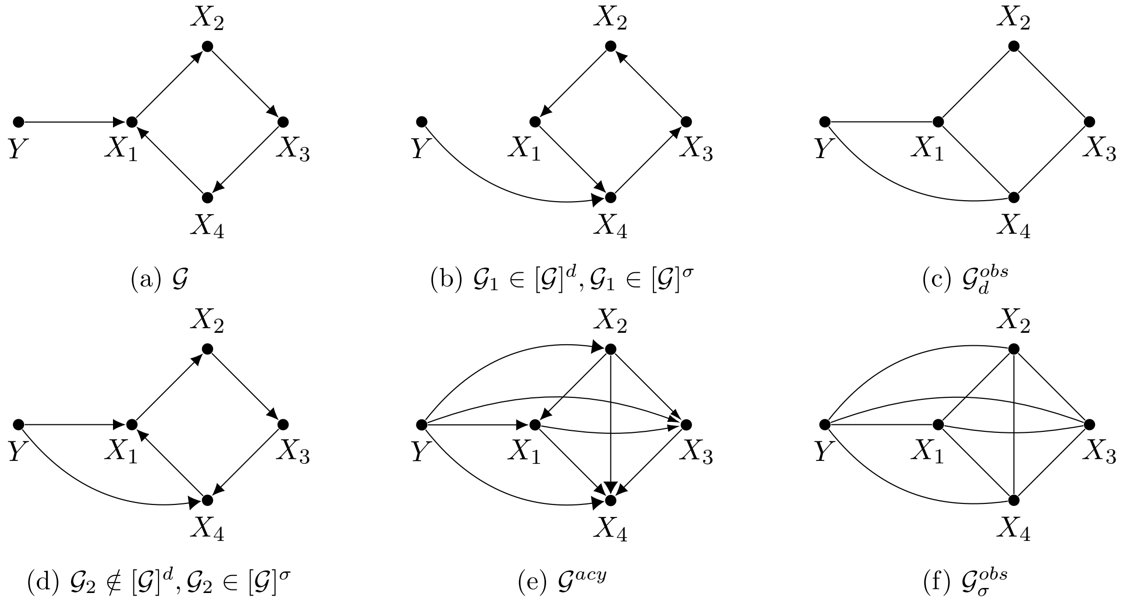
For any DAG $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, Verma and Pearl (1990) showed that non-neighbor variables are d-separable. That is, for any distinct and non-neighbor variables X and Y, there exists a subset of $\mathbf{V} \setminus \{X, Y\}$ that d-separates X and Y. This implies that the observational distribution suffices to learn the skeleton of $\mathcal{G}$. In the following, we show that this assertion is not true in cyclic graphs for either of the two scenarios introduced in 2.5. Let us begin by defining the skeleton of a graph that can be learned from the observational distribution.

Definition 15 ($\mathcal{G}^{obs}_r$) *Suppose $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ is a DG. Let $\mathcal{G}^{obs}_r$ denote the undirected graph over $\mathbf{V}$ where there is an edge between X and Y if and only if X and Y are not r-separable in $\mathcal{G}$, i.e., for any $\mathbf{S} \subseteq \mathbf{V} \setminus \{X, Y\}$ we have $(X \not\perp\!\!\!\perp_r Y|\mathbf{S})_{\mathcal{G}}$.*

Note that $\mathcal{G}^{obs}_r$ includes the skeleton of $\mathcal{G}$ but can potentially have additional edges. Next, we describe $\mathcal{G}^{obs}_d$ in Scenario 1 and $\mathcal{G}^{obs}_\sigma$ in Scenario 2.

3.1.1 Scenario 1

Example 2 (Virtual edge) *Consider DG $\mathcal{G}$ in Figure 3a. In this graph, Y and X_4 are not d-separable. Thus, there can be an edge between Y and X_4 in some of the DGs in $[\mathcal{G}]^d$, such as in DG $\mathcal{G}_1$ in Figure 3b.*



(a) $\mathcal{G}$

(b) $\mathcal{G}_1 \in [\mathcal{G}]^d, \mathcal{G}_1 \in [\mathcal{G}]^\sigma$

(c) $\mathcal{G}_d^{obs}$

(d) $\mathcal{G}_2 \notin [\mathcal{G}]^d, \mathcal{G}_2 \in [\mathcal{G}]^\sigma$

(e) $\mathcal{G}^{acy}$

(f) $\mathcal{G}_\sigma^{obs}$

Figure 3: Figures 3a, 3b, and 3c depict a cyclic DG $\mathcal{G}$, a cyclic DG in $[\mathcal{G}]^d$, and undirected graph $\mathcal{G}_d^{obs}$, respectively (Example 2). The DG in Figure 3d belongs to $[\mathcal{G}]^\sigma$ but does not belong to $[\mathcal{G}]^d$. Figures 3e and 3f depict a σ-acyclification of $\mathcal{G}$ and $\mathcal{G}_\sigma^{obs}$, respectively (Example 3).

In Example 2, a so-called *virtual edge* exists between Y and X_4 which we formally define in the following (Richardson, 1996b; Ghassami et al., 2020).

Definition 16 (Virtual edge) *There exists a virtual edge between two non-neighbor variables X and Z in a DG $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ if X and Z have a common child that is an ancestor of either X or Z, i.e., $Ch_{\mathcal{G}}(X) \cap Ch_{\mathcal{G}}(Z) \cap Anc_{\mathcal{G}}(\{X, Z\}) \neq \varnothing$.*

The following result demonstrates the importance of virtual edges.

Proposition 17 (Richardson 1996b) *Two variables are d-separable in DG $\mathcal{G}$ if and only if an edge or virtual edge does not connect them. Accordingly, $\mathcal{G}_d^{obs}$ is obtained by adding the virtual edges of $\mathcal{G}$ to the skeleton of $\mathcal{G}$.*

For DG $\mathcal{G}$ in Figure 3a (Example 2), there exists a virtual edge between Y and X_4 because $X_1 \in Ch_{\mathcal{G}}(Y) \cap Ch_{\mathcal{G}}(X_4) \cap Anc_{\mathcal{G}}(\{Y, X_4\})$. Figure 3c depicts $\mathcal{G}_d^{obs}$.

3.1.2 Scenario 2

Mooij and Claassen (2020) introduced the notion of σ-acyclification as follows.

Definition 18 (σ-acyclification) *Suppose $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ is a DG. A σ-acyclification of $\mathcal{G}$ is a DAG $\mathcal{G}' = (\mathbf{V}, \mathbf{E}')$ that satisfies the followings.*

1. *For any $X \in \mathbf{V}$ and $Y \in \mathbf{V} \setminus SCC_{\mathcal{G}}(X)$, $(X, Y) \in \mathbf{E}'$ if and only if there exists $Z \in SCC_{\mathcal{G}}(Y)$ such that $(X, Z) \in \mathbf{E}$.*
2. *For any $X \in \mathbf{V}$ and $Y \in SCC_{\mathcal{G}}(X) \setminus \{X\}$, either $(X, Y) \in \mathbf{E}'$ or $(Y, X) \in \mathbf{E}'$.*

Note that σ-acyclification is not unique since Definition 18 does not uniquely orient the edges between two variables in the same SCC.

Proposition 19 (Mooij and Claassen 2020) *There exists at least one σ-acyclification of any DG. Furthermore, if DAG $\mathcal{G}^{acy}$ is a σ-acyclification of a DG $\mathcal{G}$, then*

$$IM_{\sigma}(\mathcal{G}) = IM(\mathcal{G}^{acy}).$$

Suppose DAG $\mathcal{G}^{acy}$ is a σ-acyclification of a DG $\mathcal{G}$. Proposition 19 implies that $\mathcal{G}^{obs}_{\sigma}$ is the skeleton of $\mathcal{G}^{acy}$. Furthermore, the following corollary pertaining to the skeleton of $\mathcal{G}^{acy}$ follows from the definition of σ-acyclification.

Corollary 20 *There exists an edge between two distinct variables X and Y in $\mathcal{G}^{obs}_{\sigma}$ if and only if $Y \in SCC_{\mathcal{G}}(X)$ or there exists $Z \in SCC_{\mathcal{G}}(X)$ such that $Y \in Pa_{\mathcal{G}}(Z)$.*

This corollary describes how $\mathcal{G}^{obs}_{\sigma}$ is obtained from $\mathcal{G}$. Note that the skeleton of any DG in $[\mathcal{G}]^{\sigma}$ is a subgraph of $\mathcal{G}^{obs}_{\sigma}$.

Example 3 *Consider again the example in Figure 3. It can be shown that $\mathcal{G}$, $\mathcal{G}_1$, and $\mathcal{G}_2$ (Figure 3d) do not induce any σ-separation, i.e., $IM_{\sigma}(\mathcal{G}) = IM_{\sigma}(\mathcal{G}_1) = IM_{\sigma}(\mathcal{G}_2) = \varnothing$ and therefore, $\mathcal{G}_1, \mathcal{G}_2 \in [\mathcal{G}]^{\sigma}$. Note that $\mathcal{G}_2 \notin [\mathcal{G}]^{d}$ because $(Y \perp\!\!\!\perp_d X_2 | X_1, X_4)_{\mathcal{G}}$ but $(Y \not\!\perp\!\!\!\perp_d X_2 | X_1, X_4)_{\mathcal{G}_2}$. Surprisingly, we can construct many other DGs (more than 1000 DGs) that are in $[\mathcal{G}]^{\sigma}$ but are not in $[\mathcal{G}]^{d}$. Figures 3e and 3f depict a σ-acyclification of $\mathcal{G}$ and $\mathcal{G}^{obs}_{\sigma}$, respectively.*

To sum up this section, observational distribution does not suffice to distinguish between (i) actual edges and virtual edges in Scenario 1 and (ii) actual edges, virtual edges, and the additional edges of $\mathcal{G}^{obs}_{\sigma}$ in Scenario 2. Furthermore, $[\mathcal{G}]^{d}$ or $[\mathcal{G}]^{\sigma}$ can contain a large number of graphs with various skeletons, and it is necessary to perform experiments in order to learn the skeleton of $\mathcal{G}$.

3.2 Singleton Experiments are not Sufficient

A *singleton experiment* refers to an experiment in which the target set is comprised of a single variable. In DAGs, the children of a variable could be identified by performing a singleton experiment on it. Hence, the whole graph can be learned by performing singleton experiments on all the variables. Herein, we show that this does not hold for cyclic DGs.

Example 4 *Consider the DGs in Figure 4 and the set of singleton experiments (including the empty set) $\mathcal{I} = \{\varnothing, \{X\}, \{Y\}, \{Z\}\}$. For any DG $\mathcal{G}$ in this figure and any experiment $\mathbf{I} \in \mathcal{I}$, $IM_r(\mathcal{G}_{\overline{\mathbf{I}}}) = \varnothing$. Hence, all of the DGs in Figure 4 are $\mathcal{I}$-r-Markov equivalent. It is noteworthy that removing* any *edge in the left DG in Figure 4 results in a DG in the same $\mathcal{I}$-r-Markov equivalent class.*

This example illustrates that a DG cannot always be learned through singleton experiments, even if they are performed on all variables.

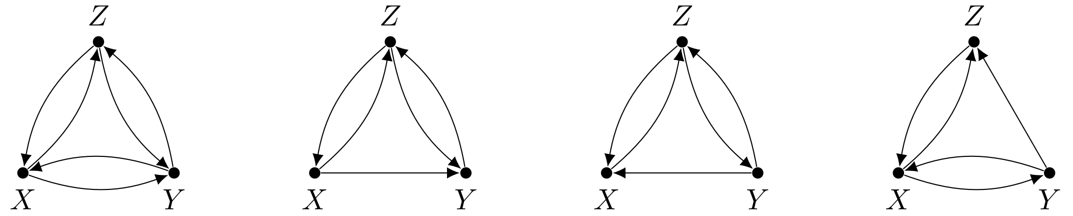


Figure 4: Four $\mathcal{I}$-r-Markov equivalent DGs, where $\mathcal{I} = \{\varnothing, \{X\}, \{Y\}, \{Z\}\}$ (Example 4).

4. Lower Bounds on Number and Size of Experiment Sets

As we discussed, performing singleton experiments does not suffice for learning a DG in some cases. In this section, we provide lower bounds on both the number and size of experiments required to learn a DG in the worst case. For any constant $c < n$, we show that among the DGs with maximum SCC size of c, there exists a DG $\mathcal{G}$ that is not uniquely identifiable by performing experiments with size less than $\zeta_{\max}(\mathcal{G}) - 1$ or conducting less than $\zeta_{\max}(\mathcal{G})$ experiments, where $\zeta_{\max}(\mathcal{G})$ denotes the size of the largest SCC of $\mathcal{G}$. To this end, we first provide the following lemma.

Lemma 21 *Consider a set of n vertices denoted by $\mathbf{V}$ and a constant $1 < c \leq n$. For an arbitrary subset $\mathbf{V}_c \subseteq \mathbf{V}$ with $|\mathbf{V}_c| = c$, let $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{E} = \{(X, Y) \mid X, Y \in \mathbf{V}_c, X \neq Y\}$. For two arbitrary and distinct variables X^* and Y^* in $\mathbf{V}_c$, let DG $\mathcal{G}' = (\mathbf{V}, \mathbf{E} \setminus \{(X^*, Y^*)\})$. For a set of experiments $\mathcal{I}$ on $\mathbf{V}$, $\mathcal{G}' \notin [\mathcal{G}]^r_{\mathcal{I}}$ if and only if*

$$\exists \mathbf{I} \in \mathcal{I} : \quad \mathbf{I} \cap \mathbf{V}_c = \mathbf{V}_c \setminus \{Y^*\}. \tag{3}$$

Proof **Sufficiency:** If there exists a subset $\mathbf{I} \in \mathcal{I}$ such that $\mathbf{I} \cap \mathbf{V}_c = \mathbf{V}_c \setminus \{Y^*\}$, after intervening on $\mathbf{I}$, X^* and Y^* are r-separable in $\mathcal{G}'$. Hence, $\mathcal{G}' \notin [\mathcal{G}]^r_{\mathcal{I}}$.

Necessity: Suppose Equation (3) does not hold. We need to show that $\mathcal{G}' \in [\mathcal{G}]^r_{\mathcal{I}}$. That is, for any $\mathbf{I} \in \mathcal{I}$ ($\mathbf{I}$ can be the empty set), we need to show that $\text{IM}_r(\mathcal{G}_{\overline{\mathbf{I}}}) = \text{IM}_r(\mathcal{G}'_{\overline{\mathbf{I}}})$. Note that $\mathcal{G}'_{\overline{\mathbf{I}}} \subseteq \mathcal{G}_{\overline{\mathbf{I}}}$ since $\mathcal{G}' \subseteq \mathcal{G}$. Hence, $\text{IM}_r(\mathcal{G}_{\overline{\mathbf{I}}}) \subseteq \text{IM}_r(\mathcal{G}'_{\overline{\mathbf{I}}})$. Let $(X, Y, \mathbf{S}) \in \text{IM}_r(\mathcal{G}'_{\overline{\mathbf{I}}})$. To complete the proof, we will show that $(X, Y, \mathbf{S}) \in \text{IM}_r(\mathcal{G}_{\overline{\mathbf{I}}})$.

If $X \in \mathbf{V} \setminus \mathbf{V}_c$ or $Y \in \mathbf{V} \setminus \mathbf{V}_c$, then $(X, Y, \mathbf{S}) \in \text{IM}_r(\mathcal{G}_{\overline{\mathbf{I}}})$ because the variables in $\mathbf{V} \setminus \mathbf{V}_c$ do not have any neighbors. Now, suppose $X, Y \in \mathbf{V}_c$. Since the variables in $\mathbf{V}_c \setminus \mathbf{I}$ are neighbors in $\mathcal{G}'$ (note that X^* and Y^* are neighbors because $(Y^*, X^*) \in \mathbf{E}$), at least one of X or Y is in $\mathbf{I}$. Without loss of generality, let us assume that $X \in \mathbf{I}$.

Next, we show that Y is also in $\mathbf{I}$. Assume by contradiction that $Y \in \mathbf{V}_c \setminus \mathbf{I}$. Since $\mathbf{S}$ r-separates X and Y in $\mathcal{G}'_{\overline{\mathbf{I}}}$, $(X, Y) \notin \mathcal{G}'$ which implies that $X = X^*$ and $Y = Y^*$. In this case, Equation (3) implies that $\mathbf{V}_c \setminus (\mathbf{I} \cup \{Y\})$ is non-empty. Let Z be a variable in $\mathbf{V}_c \setminus (\mathbf{I} \cup \{Y\})$. This implies that $Z \in Ch_{\mathcal{G}'_{\overline{\mathbf{I}}}}(X) \cap Ch_{\mathcal{G}'_{\overline{\mathbf{I}}}}(Y) \cap Pa_{\mathcal{G}'_{\overline{\mathbf{I}}}}(Y)$. Hence, there is a virtual edge between X and Y in $\mathcal{G}'_{\overline{\mathbf{I}}}$ and therefore, they are not r-separable which is a contradiction. This implies that Y is in $\mathbf{I}$.

So far we have shown that $X, Y \in \mathbf{V}_c \cap \mathbf{I}$. Due to the structure of $\mathcal{G}$, $(X, Y, \mathbf{S}) \in \text{IM}_r(\mathcal{G}_{\overline{\mathbf{I}}})$ if and only if $\mathbf{S} \cap (\mathbf{V}_c \setminus \mathbf{I}) = \varnothing$. Accordingly, to complete the proof, it suffices to show that $\mathbf{S} \cap (\mathbf{V}_c \setminus \mathbf{I}) = \varnothing$. Assume by contradiction that there exists a variable Z_1 in $\mathbf{S} \cap (\mathbf{V}_c \setminus \mathbf{I})$. In this case, $Z_1 = Y^*$ and $X^* \in \{X, Y\}$ because otherwise, $\mathbf{S}$ does not r-block path

$X \to Z_1 \leftarrow Y$ in $\mathcal{G}'_{\overline{\mathbf{I}}}$. Without loss of generality suppose $X^* = X$. Again, Equation (3) implies that there exists a variable Z_2 in $\mathbf{V}_c \backslash (\mathbf{I} \cup \{Z_1\})$. Since $\mathbf{S}$ r-blocks path $X \to Z_2 \leftarrow Y$ in $\mathcal{G}'_{\overline{\mathbf{I}}}$, then $Z_2 \notin \mathbf{S}$. In this case, $\mathbf{S}$ does not r-block path $X \to Z_2 \to Z_1 \leftarrow Y$ in $\mathcal{G}'_{\overline{\mathbf{I}}}$ which is a contradiction. This shows that $\mathbf{S} \cap (\mathbf{V}_c \setminus \mathbf{I}) = \varnothing$ and therefore, $(X, Y, \mathbf{S}) \in \mathrm{IM}_r(\mathcal{G}_{\overline{\mathbf{I}}})$, which completes the proof. ■

We now present two important consequences of Lemma 21.

Theorem 22 *Consider a set of n vertices denoted by $\mathbf{V}$ and a constant $1 < c \leq n$. There exists a DG $\mathcal{G}$ over $\mathbf{V}$ with $\zeta_{\max}(\mathcal{G}) = c$ such that for any set of experiments $\mathcal{I}$ on $\mathbf{V}$, if*

$$|\mathbf{I}| < \zeta_{\max}(\mathcal{G}) - 1, \quad \forall \mathbf{I} \in \mathcal{I}, \tag{4}$$

then $|[\mathcal{G}]^d_{\mathcal{I}}| > 1$ and $|[\mathcal{G}]^\sigma_{\mathcal{I}}| > 1$.

Proof Consider the constructed DGs $\mathcal{G}$ and $\mathcal{G}^*$ in Lemma 21. Note that $\zeta_{\max}(\mathcal{G}) = c$. If Equation (4) holds for a set of experiments $\mathcal{I}$, then Equation (3) does not hold. Therefore, Lemma 21 implies that $\mathcal{G}' \in [\mathcal{G}]^r_{\mathcal{I}}$, which completes the proof. ■

Corollary 23 *In the worst case, DG $\mathcal{G}$ cannot be learned by any algorithm (adaptive or non-adaptive) that performs experiments with size less than $\zeta_{\max}(\mathcal{G}) - 1$ for both scenarios described in Section 2.5.*

Theorem 24 *Consider a set of n vertices denoted by $\mathbf{V}$ and a constant $1 < c \leq n$. There exists a DG $\mathcal{G}$ over $\mathbf{V}$ with $\zeta_{\max}(\mathcal{G}) = c$ such that for any set of experiments $\mathcal{I}$ on $\mathbf{V}$, if $|\mathcal{I}| < \zeta_{\max}(\mathcal{G})$, then, $|[\mathcal{G}]^d_{\mathcal{I}}| > 1$ and $|[\mathcal{G}]^\sigma_{\mathcal{I}}| > 1$.*

Proof Consider the constructed DG $\mathcal{G}$ in Lemma 21. Note that $\zeta_{\max}(\mathcal{G}) = c$. Since $|\mathcal{I}| < \zeta_{\max}(\mathcal{G})$, there exists $Y^* \in \mathbf{V}_c$ such that Equation (3) does not hold. Let X^* be an arbitrary variable in $\mathbf{V}_c \setminus \{Y^*\}$, and let $\mathcal{G}'$ denote the DG obtained by removing the edge (X^*, Y^*) from $\mathcal{G}$. In this case, Lemma 21 implies that $\mathcal{G}' \in [\mathcal{G}]^r_{\mathcal{I}}$, which concludes the proof. ■

Corollary 25 *At least $\zeta_{\max}(\mathcal{G})$ experiments are required to learn $\mathcal{G}$ in the worst case.*

5. Unbounded-size Experiment Design

In this section, we propose a two-stage experiment design algorithm for learning a DG $\mathcal{G}$ (potentially cyclic) when there is no constraint on the size of the designed experiments. In the first stage, we design a set of experiments for learning the descendant sets of the variables and the strongly connected components (SCC) of $\mathcal{G}$. In the second stage and based on the findings of the first stage, we design further experiments to exactly recover $\mathcal{G}$.

5.1 Stage 1: Colored Separating System

In this section, we introduce the first stage of our approach for learning the descendant sets $\{De_{\mathcal{G}}(X)\}_{X\in\mathbf{V}}$ and the set of SCCs $\mathcal{S} = \{\mathbf{S}_1, \dots, \mathbf{S}_k\}$ of $\mathcal{G}$. This stage is based on performing experiments on certain subsets of $\mathbf{V}$ that form a *colored separating system*.

Definition 26 (Colored separating system) *Suppose* $\mathbf{V} = \{X_1, \cdots, X_n\}$ *and let* $\mathcal{C} = \{C_1, \cdots, C_n\}$ *be an arbitrary coloring for* $\mathbf{V}$*. A colored separating system* $\mathcal{I}$ *on* $(\mathbf{V}, \mathcal{C})$ *is a collection of subsets of* $\mathbf{V}$ *such that for every distinct ordered pair of variables* (X_i, X_j) *in* $\mathbf{V}$*, if* $C_i \neq C_j$*, then there exists* $\mathbf{I} \in \mathcal{I}$ *such that* $X_i \in \mathbf{I}$ *and* $X_j \notin \mathbf{I}$*.*

We note that similar definitions have been proposed in the literature. For instance, Katona (1966) introduced *separating systems*, a special case of colored separating system, where $\mathcal{C}$ must contain n different colors.

In the following, we provide an achievable bound on the cardinality of a colored separating system.

Proposition 27 *There exists a colored separating system on* $(\mathbf{V}, \mathcal{C})$ *with at most* $2\lceil\log_2(\chi)\rceil$ *elements, where* χ *is the number of colors in* $\mathcal{C}$*.*

Proof Suppose $\mathbf{V} = \{X_1, \cdots, X_n\}$ and let $l = \lceil\log_2(\chi)\rceil$. Suppose $\mathcal{C} = \{C_1, \cdots, C_n\}$, where $C_i \in \{1, \cdots, \chi\}$. For $1 \leq i \leq l$, let $\mathbf{N}_i$ be the subset of numbers in $\{1, 2, \cdots, \chi\}$ whose i-th bit in binary representation equals to 1. We now construct subsets $\mathbf{I}_i^1, \mathbf{I}_i^2 \subseteq \mathbf{V}$ for each $1 \leq i \leq l$ as follows:

$$\mathbf{I}_i^1 = \{X_j \in \mathbf{V} \mid C_j \in \mathbf{N}_i\}, \qquad \mathbf{I}_i^2 = \{X_j \in \mathbf{V} \mid C_j \notin \mathbf{N}_i\}. \tag{5}$$

Let (X_a, X_b) be an ordered pair of distinct variables in $\mathbf{V}$ such that $C_a \neq C_b$. In this case, there exists $1 \leq i \leq l$ such that the i-th bit of C_a and C_b are different in binary representation. There are two cases:

- The i-bit of C_a in binary representation is 1: In this case $X_a \in \mathbf{I}_i^1$ and $X_b \notin \mathbf{I}_i^1$.
- The i-bit of C_a in binary representation is 0: In this case $X_a \in \mathbf{I}_i^2$ and $X_b \notin \mathbf{I}_i^2$.

This shows that $\mathcal{I} = \{\mathbf{I}_i^1\}_{i=1}^l \cup \{\mathbf{I}_i^2\}_{i=1}^l$ is a colored separating set on $(\mathbf{V}, \mathcal{C})$. Note that $|\mathcal{I}| = 2l = 2\lceil\log_2(\chi)\rceil$. ■

Remark 28 *The proof of Proposition 27 is constructive. That is, with Equation* (5) *we can obtain a colored separating system on* $(\mathbf{V}, \mathcal{C})$ *with at most* $2\lceil\log_2(\chi)\rceil$ *elements.*

Equipped with Proposition 27, we present Algorithm 1 for finding the descendant sets and the set of SCCs in $\mathcal{G}$. At first, the algorithm learns $\mathcal{G}_r^{obs}$ from observational data using existing methods such as the one proposed by Ghassami et al. (2020). For more information on this step, please see Section 7.3. In lines 2 and 3, it learns a coloring of $\mathcal{G}_r^{obs}$ and subsequently, it constructs a colored separating system on $(\mathbf{V}, \mathcal{C})$ (using Proposition 27). One way to color $\mathcal{G}_r^{obs}$ is by using the *trail-path* algorithm described in Bandyopadhyay et al. (2020).

Algorithm 1: Learning descendant sets and strongly connected components

```
 1: Learn $\mathcal{G}_r^{obs}$ using observational data
 2: $\mathcal{C} \leftarrow$ A vertex coloring for $\mathcal{G}_r^{obs}$
 3: $\mathcal{I} \leftarrow$ Construct a colored separating system on $(\mathbf{V}, \mathcal{C})$
 4: for $X \in \mathbf{V}$ do
 5:     $\mathcal{I}_X \leftarrow \{\mathbf{I} \in \mathcal{I} \colon X \in \mathbf{I}\}$
 6:     Initialize $\mathbf{D}_X$ with an empty set
 7:     for $\mathbf{I} \in \mathcal{I}_X$ do
 8:         Add the elements of $\{Y \in Ne_{\mathcal{G}_r^{obs}}(X) \colon (X \not\perp\!\!\!\perp Y)_{P_{do(\mathbf{I})}}\}$ to $\mathbf{D}_X$
 9: Construct DG $\mathcal{H}$ by adding directed edges from $X$ to $\mathbf{D}_X$ for each $X \in \mathbf{V}$
10: $\{De_{\mathcal{G}}(X)\}_{X \in \mathbf{V}}, \mathcal{S} = \{\mathbf{S}_1, \dots, \mathbf{S}_k\} \leftarrow$ Compute descendant sets and SCCs of $\mathcal{H}$
11: Return $\{De_{\mathcal{G}}(X)\}_{X \in \mathbf{V}}, \mathcal{S} = \{\mathbf{S}_1, \dots, \mathbf{S}_k\}$
```

Example 5 (Colored separating system) *Consider DG $\mathcal{G}$ in Figure 5a over the set of variables $\mathbf{V} = \{X_1, X_2, X_3, X_4\} \cup \{Y_1, Y_2, Y_3, Y_4\} \cup \{Z_1, Z_2, Z_3, Z_4\}$. DG $\mathcal{G}$ is cyclic with three SCCs $\mathbf{S}_1 = \{X_1, X_2, X_3, X_4\}$, $\mathbf{S}_2 = \{Y_1, Y_2, Y_3, Y_4\}$, and $\mathbf{S}_3 = \{Z_1, Z_2, Z_3, Z_4\}$. In Scenario 1, i.e., when CIs are equivalent to d-separations, Algorithm 1 learns the undirected graph $\mathcal{G}_d^{obs}$ from observational data, which is depicted in Figure 5b. Recall that $\mathcal{G}_d^{obs}$ includes the virtual edges (red edges) and the edges of the skeleton of $\mathcal{G}$ (black edges). A coloring for $\mathcal{G}_d^{obs}$ with four colors is shown in Figure 5b. Specifically, $\{X_2, X_4, Z_2, Z_4\}$, $\{X_1, X_3, Z_1, Z_3\}$, $\{Y_1, Y_3\}$, and $\{Y_2, Y_4\}$ comprise the set of variables with the same color. Using this coloring, Proposition 27 constructs the following colored separating system of size $2\lceil \log_2(4) \rceil = 4$:*

$$\begin{aligned}\mathcal{I} = \{&\{X_1, X_3, Y_1, Y_3, Z_1, Z_3\}, \{X_2, X_4, Y_2, Y_4, Z_2, Z_4\},\\ &\{X_1, X_3, Y_2, Y_4, Z_1, Z_3\}, \{X_2, X_4, Y_1, Y_3, Z_2, Z_4\}\}.\end{aligned}$$

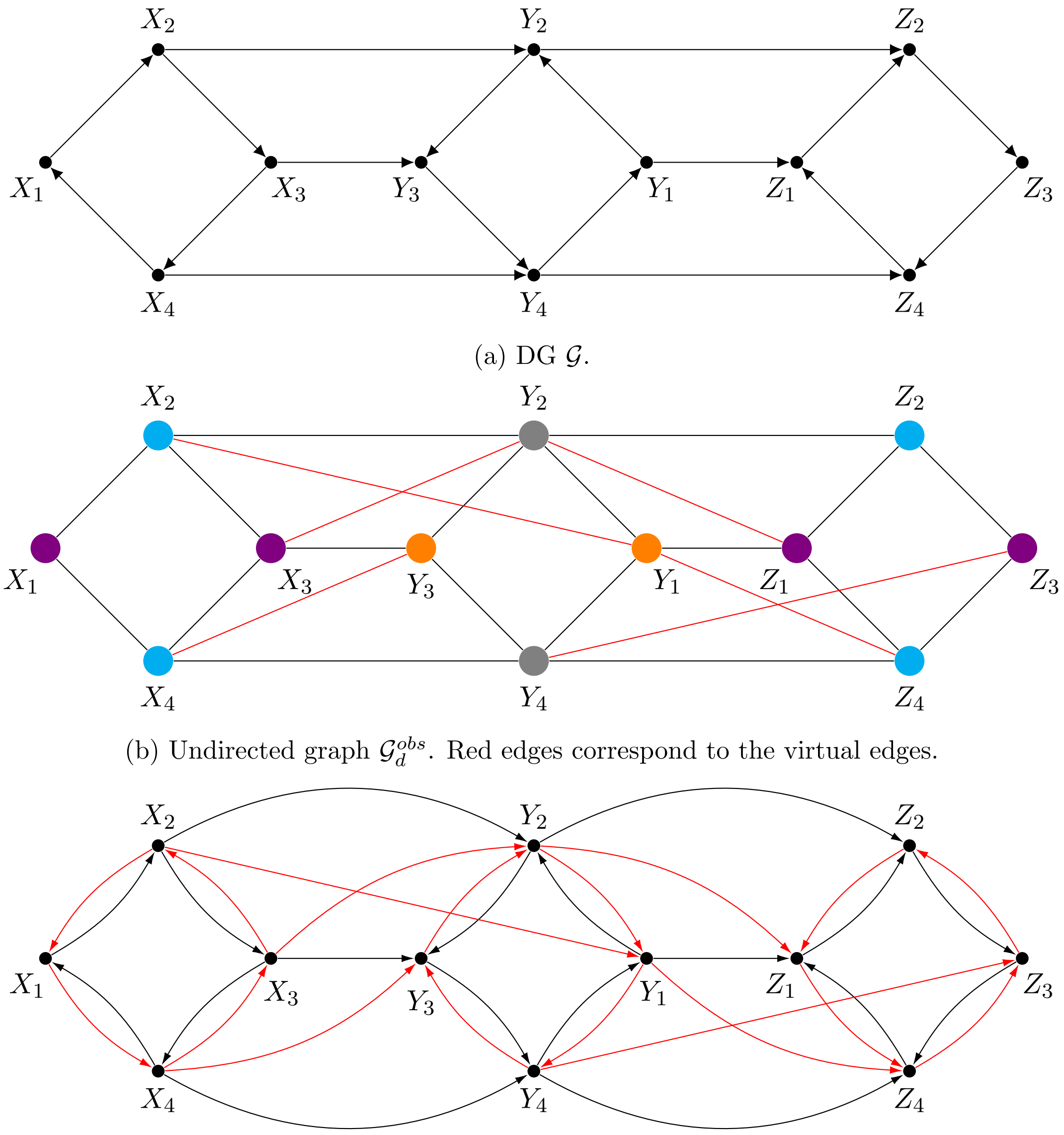
After constructing a colored separating system, Algorithm 1 constructs a set $\mathbf{D}_X$ for each $X \in \mathbf{V}$ in lines 4-8 as follows. In line 5, $\mathcal{I}_X = \{\mathbf{I} \in \mathcal{I} \colon X \in \mathbf{I}\}$ is defined and in line 6, $\mathbf{D}_X$ is initialized with an empty set. Based on the following lemma, for any set $\mathbf{I} \subseteq \mathbf{V}$ and each $X \in \mathbf{I}$, $De_{\mathcal{G}_{\overline{\mathbf{I}}}}(X)$ is learned by performing an experiment on $\mathbf{I}$.

Lemma 29 *For each $X \in \mathbf{I} \subseteq \mathbf{V}$, $De_{\mathcal{G}_{\overline{\mathbf{I}}}}(X) = \{Y \in \mathbf{V} \colon (X \not\perp\!\!\!\perp Y)_{P_{do(\mathbf{I})}}\}$.*

Proof We first show that $De_{\mathcal{G}_{\overline{\mathbf{I}}}}(X) = \{Y \in \mathbf{V} \colon (X \not\perp\!\!\!\perp_r Y)_{\mathcal{G}_{\overline{\mathbf{I}}}}\}$.

- Suppose $Y \in De_{\mathcal{G}_{\overline{\mathbf{I}}}}(X)$. In this case, there exists a directed path from X to Y in $\mathcal{G}_{\overline{\mathbf{I}}}$ and therefore, $(X \not\perp\!\!\!\perp_r Y)_{\mathcal{G}_{\overline{\mathbf{I}}}}$.
- Suppose $(X \not\perp\!\!\!\perp_r Y)_{\mathcal{G}_{\overline{\mathbf{I}}}}$. This implies that there exists a path $\mathcal{P}$ between X and Y in $\mathcal{G}_{\overline{\mathbf{I}}}$ that does not contain any colliders. Note that X does not have any parents in $\mathcal{G}_{\overline{\mathbf{I}}}$ because $X \in \mathbf{I}$. Thus, $\mathcal{P}$ must be a directed path from X to Y, which implies that $Y \in De_{\mathcal{G}_{\overline{\mathbf{I}}}}(X)$.

Since $do(\mathbf{I})$ is a full-support hard intervention, the CI assertions in $P_{do(\mathbf{I})}$ are equivalent to d-separations or σ-separations of $\mathcal{G}_{\overline{\mathbf{I}}}$ for Scenario 1 or Scenario 2, respectively. Hence, set $\{Y \in \mathbf{V} : (X \not\perp\!\!\!\perp Y)_{P_{do(\mathbf{I})}}\}$ is equal to $\{Y \in \mathbf{V} : (X \not\perp\!\!\!\perp_d Y)_{\mathcal{G}_{\overline{\mathbf{I}}}}\}$ in Scenario 1, and is



(a) DG $\mathcal{G}$.

(b) Undirected graph $\mathcal{G}_d^{obs}$. Red edges correspond to the virtual edges.

(c) DG $\mathcal{H}$ in Scenario 1. Red edges correspond to the ones that do not exist in $\mathcal{G}$.

Figure 5: A running example for our proposed approach (Examples 5, 6, and 7).

equal to $\{Y \in \mathbf{V} : (X \not\perp\!\!\!\perp_\sigma Y)_{\mathcal{G}_{\bar{\mathbf{I}}}}\}$ in Scenario 2. Therefore, under both Scenarios 1 or 2, $De_{\mathcal{G}_{\bar{\mathbf{I}}}}(X) = \{Y \in \mathbf{V}\colon (X \not\perp\!\!\!\perp Y)_{P_{do(\mathbf{I})}}\}$. ■

Applying Lemma 29, Algorithm 1 adds $De_{\mathcal{G}_{\bar{\mathbf{I}}}}(X) \cap Ne_{\mathcal{G}_r^{obs}}(X)$ to $\mathbf{D}_X$ for each $\mathbf{I} \in \mathcal{I}_X$ in line 8. Therefore, at the end of the for loop (lines 7-8), we have

$$\mathbf{D}_X = \left(\bigcup_{\mathbf{I} \in \mathcal{I}_X} De_{\mathcal{G}_{\bar{\mathbf{I}}}}(X) \right) \cap Ne_{\mathcal{G}_r^{obs}}(X). \tag{6}$$

Next, we show that $\mathbf{D}_X$ contains $Ch_{\mathcal{G}}(X)$, and it is also a subset of $De_{\mathcal{G}}(X)$.

Lemma 30 *For each $X \in \mathbf{V}$, $Ch_{\mathcal{G}}(X) \subseteq \mathbf{D}_X \subseteq De_{\mathcal{G}}(X)$, where $\mathbf{D}_X$ is defined in* (6).

Proof For any subset $\mathbf{I} \subseteq \mathbf{V}$, $De_{\mathcal{G}_{\overline{\mathbf{I}}}}(X)$ is a subset of $De_{\mathcal{G}}(X)$. Hence, $\mathbf{D}_X \subseteq De_{\mathcal{G}}(X)$.

Suppose $Y \in Ch_{\mathcal{G}}(X)$. We need to show that $Y \in \mathbf{D}_X$. Note that X and Y are neighbors in $\mathcal{G}_r^{obs}$ and, therefore, have different colors in $\mathcal{C}$. Since $\mathcal{I}$ is a colored separating system on $(\mathbf{V}, \mathcal{C})$, there exists $\mathbf{I} \in \mathcal{I}$ such that $X \in \mathbf{I}$ and $Y \notin \mathbf{I}$. In this case, $\mathbf{I} \in \mathcal{I}_X$ since $X \in \mathbf{I}$. Furthermore, $Y \in De_{\mathcal{G}_{\overline{\mathbf{I}}}}(X)$ because Y is a child of X in $\mathcal{G}_{\overline{\mathbf{I}}}$. This implies that $Y \in \mathbf{D}_X$ and therefore, $Ch_{\mathcal{G}}(X) \subseteq \mathbf{D}_X$. ■

Remark 31 *For the algorithm to successfully learn the SCCs, it is crucial that $\mathbf{D}_X$ contains all the children of X. As proven in Lemma 30, this is the case because $\mathcal{I}$ is a colored separating system on $(\mathbf{V}, \mathcal{C})$. Note that if $\mathcal{I}$ were not a colored separating system on $(\mathbf{V}, \mathcal{C})$, $\mathbf{D}_X$ would still be a subset of $De_{\mathcal{G}}(X)$, but it would not have necessarily contained all the variables in $Ch_{\mathcal{G}}(X)$.*

After learning $\mathbf{D}_X$ for all $X \in \mathbf{V}$, a DG $\mathcal{H}$ is constructed over $\mathbf{V}$ by adding directed edges from X to the variables in $\mathbf{D}_X$ for each $X \in \mathbf{V}$ (line 9).

Example 6 (DG $\mathcal{H}$) *Following Example 5, consider the graphs in Figure 5. DG $\mathcal{H}$, which is constructed by adding directed edges from X to $\mathbf{D}_X$ for each $X \in \mathbf{V}$, is depicted in Figure 5c. For instance, $\mathbf{D}_{X_2} = \{X_1, X_3, Y_1, Y_2\}$. In this figure, black edges are the edges that appear in DG $\mathcal{G}$, while red edges do not exist in $\mathcal{G}$.*

Observe that DG $\mathcal{H}$ is a super graph of $\mathcal{G}$, where the extra edges in $\mathcal{H}$ appear only from the variables to some of their descendants in $\mathcal{G}$. In fact, the following corollary of Lemma 30 holds.

Corollary 32 *In Algorithm 1, DG $\mathcal{G}$ and DG $\mathcal{H}$ (the constructed DG in line 9) have the same descendant sets, i.e., for each $X \in \mathbf{V}$, $De_{\mathcal{H}}(X) = De_{\mathcal{G}}(X)$. Accordingly, $\mathcal{G}$ and $\mathcal{H}$ have the same SCCs.*

Note that the second part of Corollary 32 is due to the fact that by definition, two variables X and Y are in the same SCC in $\mathcal{G}$ if and only if $X \in De_{\mathcal{G}}(Y)$ and $Y \in De_{\mathcal{G}}(X)$.

Given a DG with n vertices, there exist efficient depth-first search (DFS)-based algorithms, such as *Kosaraju*, for obtaining the descendant sets and the SCCs with the computational complexity of $\mathcal{O}(n^2)$ (Sharir, 1981). Applying any of these algorithms to $\mathcal{H}$, Algorithm 1 can obtain $\{De_{\mathcal{G}}(X)\}_{X \in \mathbf{V}}$ and the set of SCCs $\mathcal{S} = \{\mathbf{S}_1, \dots, \mathbf{S}_k\}$ of $\mathcal{G}$ in line 10.

Remark 33 *For the soundness of Algorithm 1, it suffices for $\mathcal{G}_r^{obs}$ to be a super graph of the skeleton of $\mathcal{G}$. For instance, if we do not have access to the observational data, Algorithm 1 can set $\mathcal{G}_r^{obs}$ to be a complete graph in line 1.*

Remark 34 *If $\mathcal{G}$ is a DAG, then $\mathcal{H} = \mathcal{G}$, and it suffices for the algorithm to return $\mathcal{H}$ at the end of this stage. To test for this, the algorithm must check whether $\mathcal{H}$ is a DAG after line 9 in Algorithm 1.*

5.2 Stage 2: Lifted Separating System

As we discussed in the previous section, the descendant sets and the set of SCCs $\mathcal{S} = \{\mathbf{S}_1, \dots, \mathbf{S}_k\}$ of $\mathcal{G}$ can be learned by performing $2\lceil \log_2(\chi(\mathcal{G}_r^{obs}))\rceil$ experiments. Herein, as the second stage of our approach, we design $\zeta_{\max}(\mathcal{G}) \coloneqq \max(|\mathbf{S}_1|, \cdots, |\mathbf{S}_k|)$ new experiments to learn $\mathcal{G}$. In this stage, we perform experiments on certain subsets of $\mathbf{V}$ that form a *lifted separating system*, formally defined in the following.

Definition 35 (Lifted separating system) *Suppose $\mathcal{S} = \{\mathbf{S}_1, \dots, \mathbf{S}_k\}$ is the set of SCCs of a DG $\mathcal{G}$ with the set of vertices $\mathbf{V}$. A lifted separating system $\mathcal{I}$ on $(\mathbf{V}, \mathcal{S})$ is a collection of subsets of $\mathbf{V}$ such that for each $i \in \{1, \cdots, k\}$ and $X \in \mathbf{S}_i$, there exists $\mathbf{I} \in \mathcal{I}$ such that $\mathbf{S}_i \setminus \{X\} \subseteq \mathbf{I}$ and $X \notin \mathbf{I}$.*

We note that, as far as we know, no similar definition exists in the literature. In the following, we provide a method for constructing a lifted separating system with at most $\zeta_{\max}(\mathcal{G})$ elements.

Proposition 36 *Suppose $\mathcal{S} = \{\mathbf{S}_1, \dots, \mathbf{S}_k\}$ is the set of SCCs of a DG $\mathcal{G} = (\mathbf{V}, \mathbf{E})$. There exists a lifted separating system on $(\mathbf{V}, \mathcal{S})$ with at most $\zeta_{\max}(\mathcal{G})$ elements.*

Proof For each $1 \leq j \leq k$, suppose $\mathbf{S}_j = \{X_1^j, \cdots X_{l_j}^j\}$, where $l_j = |\mathbf{S}_j|$. Also, let $l_{\max} = \max(l_1, \cdots, l_k) = \zeta_{\max}(\mathcal{G})$. For each $1 \leq i \leq l_{\max}$, we construct subset $\mathbf{I}_i \subseteq \mathbf{V}$ as follows. For each $1 \leq j \leq k$ such that $i \leq l_j$, we add $\mathbf{S}_j \setminus \{X_i^j\}$ to $\mathbf{I}_i$. That is,

$$\mathbf{I}_i = \bigcup_{\substack{1\leq j\leq k \\ \text{s.t. } i\leq l_j}} (\mathbf{S}_j \setminus \{X_i^j\}). \tag{7}$$

Next, we show that $\mathcal{I} = \{\mathbf{I}_1, \cdots, \mathbf{I}_{l_{\max}}\}$ is a lifted separating system on $(\mathbf{V}, \mathcal{S})$. Note that $|\mathcal{I}| = \zeta_{\max}(\mathcal{G})$. Suppose $j \in \{1, \cdots, k\}$ and $X_i^j \in \mathbf{S}_j$, where $1 \leq i \leq l_j$. We need to show that there exists $\mathbf{I} \in \mathcal{I}$ such that $\mathbf{S}_j \setminus \{X_i^j\} \subseteq \mathbf{I}$ and $X_i^j \notin \mathbf{I}$. $\mathbf{I} = \mathbf{I}_i$ satisfies this property because $\mathbf{I}_i \cap \mathbf{S}_j = \mathbf{S}_j \setminus \{X_i^j\}$. Hence, $\mathcal{I}$ is a lifted separating system on $(\mathbf{V}, \mathcal{S})$ with size $\zeta_{\max}(\mathcal{G})$. ■

Remark 37 *The proof of Proposition 36 is constructive. Given the set of SCCs, Equation (7) provides a lifted separating system on $(\mathbf{V}, \mathcal{S})$ with at most $\zeta_{\max}(\mathcal{G})$ elements.*

Example 7 (Lifted separating system) *Consider DG $\mathcal{G}$ in Figure 5 with three SCCs $\mathbf{S}_1 = \{X_1, X_2, X_3, X_4\}$, $\mathbf{S}_2 = \{Y_1, Y_2, Y_3, Y_4\}$, and $\mathbf{S}_3 = \{Z_1, Z_2, Z_3, Z_4\}$. Using Equation (7) in the proof of Proposition 36, we can construct the following lifted separating system of size $\zeta_{\max}(\mathcal{G}) = \max(|\mathbf{S}_1|, |\mathbf{S}_2|, |\mathbf{S}_3|) = 4$.*

$$\begin{aligned}\mathcal{I} = \{&\{X_2, X_3, X_4, Y_2, Y_3, Y_4, Z_2, Z_3, Z_4\}, \{X_1, X_3, X_4, Y_1, Y_3, Y_4, Z_1, Z_3, Z_4\}, \\ &\{X_1, X_2, X_4, Y_1, Y_2, Y_4, Z_1, Z_2, Z_4\}, \{X_1, X_2, X_3, Y_1, Y_2, Y_3, Z_1, Z_2, Z_3\}\}.\end{aligned}$$

Algorithm 2: Learning a DG $\mathcal{G}$

```
1: **Input:** $\{Anc_{\mathcal{G}}(X)\}_{X \in \mathbf{V}}, \mathcal{S} = \{\mathbf{S}_1, \dots, \mathbf{S}_k\}$
2: $\mathcal{I} \leftarrow$ Construct a lifted separating system on $(\mathbf{V}, \mathcal{S})$
3: Initialization: $\hat{\mathcal{G}} \leftarrow (\mathbf{V}, \hat{\mathbf{E}} = \varnothing)$
4: **for** $i$ from 1 to $k$ **do**
5:    **for** $X \in \mathbf{S}_i$ **do**
6:       $\mathbf{I} \leftarrow$ An element of $\mathcal{I}$ that contains $\mathbf{S}_i \setminus \{X\}$ but does not contain $X$
7:       **for** $Y \in \mathbf{S}_i \setminus \{X\}$ **do**
8:          Add $(Y, X)$ to $\hat{\mathbf{E}}$ if $(X \not\perp\!\!\!\perp Y)_{P_{do(\mathbf{I})}}$
9:       **for** $Y \in Anc_{\mathcal{G}}(X) \setminus \mathbf{S}_i$ **do**
10:         Add $(Y, X)$ to $\hat{\mathbf{E}}$ if $(X \not\perp\!\!\!\perp Y | Anc_{\mathcal{G}}(X) \setminus (\mathbf{S}_i \cup \{Y\}))_{P_{do(\mathbf{I})}}$
11: **Return** $\hat{\mathcal{G}}$
```

We present Algorithm 2 for learning DG $\mathcal{G}$ that takes the ancestor sets[2] $\{Anc_{\mathcal{G}}(X)\}_{X \in \mathbf{V}}$ and the set of SCCs $\mathcal{S} = \{\mathbf{S}_1, \dots, \mathbf{S}_k\}$ of $\mathcal{G}$ as inputs. The algorithm constructs a lifted separating system $\mathcal{I}$ in line 2 and initializes a DG $\hat{\mathcal{G}}$ on $\mathbf{V}$ with no edges in line 3.

Suppose X is an arbitrary variable in $\mathbf{S}_i$, where $1 \leq i \leq k$ (the for loops in lines 4 and 5). Since $\mathcal{I}$ is a lifted separating system on $(\mathbf{V}, \mathcal{S})$, there exists $\mathbf{I} \in \mathcal{I}$ that contains $\mathbf{S}_i \setminus \{X\}$ but not X (line 6). By performing an experiment on $\mathbf{I}$ and using the following two lemmas, the algorithm finds the parents of X in lines 7-10.

Lemma 38 *Suppose $Y \in \mathbf{S}_i \setminus \{X\}$ and $\mathbf{I} \subseteq \mathbf{V} \setminus \{X\}$ such that $\mathbf{S}_i \setminus \{X\} \subseteq \mathbf{I}$. Then, $Y \in Pa_{\mathcal{G}}(X)$ if and only if $(X \not\perp\!\!\!\perp Y)_{P_{do(\mathbf{I})}}$.*

Proof Recall that $\mathbf{S}_i = SCC_{\mathcal{G}}(X)$. Since $Y \in \mathbf{I}$, Lemma 29 implies that $(X \not\perp\!\!\!\perp Y)_{P_{do(\mathbf{I})}}$ if and only if $X \in De_{\mathcal{G}_{\overline{\mathbf{I}}}}(Y)$.

Sufficient part: If $Y \in Pa_{\mathcal{G}}(X)$, then $Y \in Pa_{\mathcal{G}_{\overline{\mathbf{I}}}}(X)$ since $X \notin \mathbf{I}$. Thus, $X \in De_{\mathcal{G}_{\overline{\mathbf{I}}}}(Y)$.

Necessary part: If $X \in De_{\mathcal{G}_{\overline{\mathbf{I}}}}(Y)$, then there exists a directed path from Y to X in $\mathcal{G}_{\overline{\mathbf{I}}}$. We now show that there exists no directed path from Y to X in $\mathcal{G}_{\overline{\mathbf{I}}}$ with length larger than 1. Suppose not and let $\mathcal{P} = (Y, Z_1, \cdots, Z_t, X)$ be a directed path from Y to X in $\mathcal{G}_{\overline{\mathbf{I}}}$, where $t \geq 1$. In this case, $Pa_{\mathcal{G}_{\overline{\mathbf{I}}}}(Z_1)$ is non-empty since Y is in it. Hence, $Z_1 \notin \mathbf{I}$ and $Z_1 \notin \mathbf{S}_i$ because $\mathbf{S}_i \setminus \{X\} \subseteq \mathbf{I}$. This implies that

- $Z_1 \in Anc_{\mathcal{G}}(X)$ because of the directed path $(Z_1, \cdots, Z_t, X)$, and
- $Z_1 \in De_{\mathcal{G}}(X)$ because $Y \in De_{\mathcal{G}}(X)$ and $Z_1 \in Ch_{\mathcal{G}}(Y)$.

This shows that $Z_1 \in SCC_{\mathcal{G}}(X) = \mathbf{S}_i$, which is a contradiction. Hence, there exists no directed path from Y to X in $\mathcal{G}_{\overline{\mathbf{I}}}$ with length larger than 1. Therefore, $Y \in Pa_{\mathcal{G}}(X)$. ■

Applying Lemma 38, Algorithm 2 finds the parents of X which belong to $\mathbf{S}_i \setminus \{X\}$ in lines 7-8.

Since $Pa_{\mathcal{G}}(X) \subseteq Anc_{\mathcal{G}}(X)$, the parents of X are either in $\mathbf{S}_i \setminus \{X\}$ or $Anc_{\mathcal{G}}(X) \setminus \mathbf{S}_i$. The following lemma shows how the algorithm finds the parents of X, which belong to $Anc_{\mathcal{G}}(X) \setminus \mathbf{S}_i$.

[2] Algorithm 1 returns the descendant sets which can be used to obtain the ancestor sets.

Lemma 39 *Suppose $Y \in Anc_{\mathcal{G}}(X) \setminus \mathbf{S}_i$ and $\mathbf{I} \subseteq \mathbf{V} \setminus \{X\}$ such that $\mathbf{S}_i \setminus \{X\} \subseteq \mathbf{I}$. In this case, $Y \in Pa_{\mathcal{G}}(X)$ if and only if $(X \not\perp\!\!\!\perp Y | Anc_{\mathcal{G}}(X) \setminus (\mathbf{S}_i \cup \{Y\}))_{P_{do(\mathbf{I})}}$.*

Proof Let $\mathbf{Z} = Anc_{\mathcal{G}}(X) \setminus (\mathbf{S}_i \cup \{Y\})$. We note that $(X \not\perp\!\!\!\perp Y|\mathbf{Z})_{P_{do(\mathbf{I})}}$ if and only if $(X \not\perp\!\!\!\perp_r Y|\mathbf{Z})_{\mathcal{G}_{\overline{\mathbf{I}}}}$.

Sufficient part: If $Y \in Pa_{\mathcal{G}}(X)$, then $Y \in Pa_{\mathcal{G}_{\overline{\mathbf{I}}}}(X)$ since $X \notin \mathbf{I}$. Thus, $(X \not\perp\!\!\!\perp_r Y|\mathbf{Z})_{\mathcal{G}_{\overline{\mathbf{I}}}}$.

Necessary part: Suppose $Y \notin Pa_{\mathcal{G}}(X)$. In this case, $Y \notin Ch_{\mathcal{G}}(X)$ because $Y \in Anc_{\mathcal{G}}(X) \setminus \mathbf{S}_i$. We need to show that $(X \perp\!\!\!\perp_r Y|\mathbf{Z})_{\mathcal{G}_{\overline{\mathbf{I}}}}$. Let $\mathcal{P} = (X, Z_1, \cdots, Z_t, Y)$ be a path in $\mathcal{G}_{\overline{\mathbf{I}}}$ between X and Y. Note that $t \geq 1$ because $Y \notin Pa_{\mathcal{G}}(X) \cup Ch_{\mathcal{G}}(X)$. We have the following cases:

- $X \leftarrow Z_1$ and $Z_1 \notin \mathbf{S}_i$: Then, Z_1 r-blocks $\mathcal{P}$ because $Z_1 \in Pa_{\mathcal{G}}(X) \setminus \mathbf{S}_i \subseteq \mathbf{Z}$ and $X \notin SCC_{\mathcal{G}_{\overline{\mathbf{I}}}}(Z_1)$.
- $X \leftarrow Z_1$ and $Z_1 \in \mathbf{S}_i$: Then, $Z_1 \in \mathbf{I}$ and $Pa_{\mathcal{G}_{\overline{\mathbf{I}}}}(Z_1) = \varnothing$. Hence, $t \geq 2$ and $Z_1 \rightarrow Z_2$. Note that $Y \in Anc_{\mathcal{G}}(Z_1)$ since $Y \in Anc_{\mathcal{G}}(X)$ and $Z_1 \in \mathbf{S}_i$. Moreover, $Y \notin \mathbf{S}_i = Anc_{\mathcal{G}}(Z_1) \cap De_{\mathcal{G}}(Z_1)$. Therefore, $Y \notin De_{\mathcal{G}}(Z_1)$ and $\mathcal{P}$ contains a collider. Let Z_j be the first collider on $\mathcal{P}$. Note that $j \geq 2$ and $Z_j \notin \mathbf{S}_i$ because the variables in $\mathbf{S}_i \setminus \{X\}$ do not have any parents in $\mathcal{G}_{\overline{\mathbf{I}}}$. Furthermore, $Z_j \in De_{\mathcal{G}}(Z_1) = De_{\mathcal{G}}(X)$. Hence, $Z_j \notin Anc_{\mathcal{G}_{\overline{\mathbf{I}}}}(\mathbf{Z})$ and therefore, Z_j r-blocks $\mathcal{P}$.
- $X \rightarrow Z_1$: This case is similar to the previous case. $Y \notin De_{\mathcal{G}}(X)$ because $Y \notin \mathbf{S}_i$. Hence, $\mathcal{P}$ contains a collider. Let Z_j be the first collider on $\mathcal{P}$. $Z_j \notin \mathbf{S}_i$ because the variables in $\mathbf{S}_i \setminus \{X\}$ do not have any parents in $\mathcal{G}_{\overline{\mathbf{I}}}$. Furthermore, $Z_j \in De_{\mathcal{G}}(X)$. Hence, $Z_j \notin Anc_{\mathcal{G}_{\overline{\mathbf{I}}}}(\mathbf{Z})$ and therefore, Z_j r-blocks $\mathcal{P}$.

In all of the aforementioned cases, $\mathcal{P}$ is r-blocked which shows that $(X \perp\!\!\!\perp_r Y|\mathbf{Z})_{\mathcal{G}_{\overline{\mathbf{I}}}}$. ■

Applying Lemma 39, Algorithm 2 finds the rest of the parents of X in lines 9-10. Hence, by the time the algorithm terminates, all the parents of X are added to $\hat{\mathcal{G}}$, and $\hat{\mathcal{G}}$ will equal $\mathcal{G}$.

In Section 5.1, we showed that the descendant sets and SCCs of a DG $\mathcal{G}$ can be learned by performing experiments on the elements of a colored separating system. Herein, we showed that using the information about the descendant sets and SCCs, $\mathcal{G}$ can be recovered by performing experiments on the elements of a lifted separating system. Moreover, we provided Propositions 27 and 36 for constructing separating systems and lifted separating systems, respectively, which imply the following.

Corollary 40 *Algorithms 1 and 2 together can learn a DG $\mathcal{G}$ with n vertices with at most*

$$2\lceil \log_2(\chi(\mathcal{G}_r^{obs}))\rceil + \zeta_{\max}(\mathcal{G}) \tag{8}$$

experiments. Comparing this with the lower bound in Theorem 24, the proposed approach is order-optimal in terms of the number of experiments up to an additive logarithmic term.

Remark 41 *When $\mathcal{G}$ is a DAG, the first stage of the algorithm uniquely learns $\mathcal{G}$ (Remark 34). Hence, the algorithm performs $2\lceil \log_2(\chi(\mathcal{G}_r^{obs}))\rceil$ experiments.*

6. Bounded-size Experiment Design

In the previous sections, we did not impose any constraint on the size of experiments, and our algorithm was allowed to perform experiments with arbitrary sizes. In practice, performing *large-sized* experiments may not be possible or too costly. In this section, we study the experiment design problem with a constraint on the size of the experiments. Formally, our goal is to design a collection of subsets, denoted by $\mathcal{I}$, such that $[\mathcal{G}]^r_{\mathcal{I}} = \{\mathcal{G}\}$ (i.e., $\mathcal{G}$ can be learned by performing experiments on the elements of $\mathcal{I}$), where the size of each $\mathbf{I} \in \mathcal{I}$ is upper bounded by a constant number $M < n$ (i.e., $|\mathbf{I}| \leq M$). It is noteworthy that this problem was previously studied for acyclic causal graphs (Shanmugam et al., 2015; Lindgren et al., 2018).

Remark 42 *As proved in Theorem 22, it is necessary to perform some experiments with size at least $\zeta_{\max}(\mathcal{G}) - 1$ to learn a DG $\mathcal{G}$ in the worst case. Hence, the upper bound M cannot be smaller than $\zeta_{\max}(\mathcal{G}) - 1$.*

We will modify the two stages of our proposed method (introduced in Sections 5.1 and 5.2) in order to accommodate the new constraint that the size of the experiments is bounded by a constant $M \geq \zeta_{\max}(\mathcal{G}) - 1$.

6.1 Stage 1: (n, M)-separating System

In the first stage, instead of learning $\mathcal{G}^{obs}_r$ and constructing a colored separating system, we construct an (n, M)-separating system, formally defined by Shanmugam et al. (2015) as follows.

Definition 43 ((n, M)-separating system) *An (n, M)-separating system $\mathcal{I}$ on $\mathbf{V}$ is a collection of subsets of $\mathbf{V}$ such that $|\mathbf{I}| \leq M$ for each $\mathbf{I} \in \mathcal{I}$, and for every ordered pair of distinct variables (X, Y) in $\mathbf{V}$ there exists $\mathbf{I} \in \mathcal{I}$ such that $X \in \mathbf{I}$ and $Y \notin \mathbf{I}$.*

Shanmugam et al. (2015) also provided an achievable bound on the cardinality of an (n, M)-separating system.

Proposition 44 (Shanmugam et al. 2015) *There exists an (n, M)-separating system on $\mathbf{V}$ with at most $\lceil \frac{n}{M} \rceil \lceil \log_{\lceil \frac{n}{M} \rceil} n \rceil$ elements.*

Remark 45 *Shanmugam et al. (2015) provided a constructive proof for this proposition which allows us to obtain an (n, M)-separating system on $\mathbf{V}$ with at most $\lceil \frac{n}{M} \rceil \lceil \log_{\lceil \frac{n}{M} \rceil} n \rceil$ elements.*

It suffices to modify lines 1-3 of Algorithm 1 by setting $\mathcal{I}$ to be an (n, M)-separating system on $\mathbf{V}$ and leaving the rest of the algorithm unchanged. It is straightforward to verify that the modified algorithm obtains $\{De_{\mathcal{G}}(X)\}_{X \in \mathbf{V}}$ and the set of SCCs $\mathcal{S} = \{\mathbf{S}_1, \ldots, \mathbf{S}_k\}$ of $\mathcal{G}$ by performing experiments on the elements of $\mathcal{I}$.

6.2 Stage 2: Bounded Lifted Separating System

Algorithm 2 remains unchanged for this stage except that in line 2 of Algorithm 2, we need to construct a lifted separating system on $(\mathbf{V}, \mathcal{S})$ such that the size of the elements of $\mathcal{I}$ does not exceed M.

Theorem 46 *Suppose $\zeta_{\max}(\mathcal{G}) - 1 \leq M$. There exists a lifted separating system $\mathcal{I}$ on $(\mathbf{V}, \mathcal{S})$ such that for each $\mathbf{I} \in \mathcal{I}$, $|\mathbf{I}| \leq M$, and $|\mathcal{I}| \leq \zeta_{\max}(\mathcal{G})(1 + \lfloor \frac{n-\zeta_{\max}(\mathcal{G})-1}{M-\zeta_{\max}(\mathcal{G})+2} \rfloor)$.*

Proof For each $1 \leq j \leq k$, suppose $\mathbf{S}_j = \{X_1^j, \cdots X_{l_j}^j\}$, where $l_j = |\mathbf{S}_j|$. Also, let $l_{\max} = \max(l_1, \cdots, l_k) = \zeta_{\max}(\mathcal{G})$ and $t = \lfloor \frac{n-l_{\max}-1}{M-l_{\max}+2} \rfloor$.

Let us fix an $1 \leq i \leq l_{\max}$. Consider set $\mathbf{A} = \{j \mid 1 \leq j \leq k, i \leq l_j\}$ which is the set of js that variables X_i^js are defined. Furthermore, for each $j \in \mathbf{A}$, we define $\mathbf{B}_j = \mathbf{S}_j \setminus \{X_i^j\}$ and $b_j = |\mathbf{B}_j| = l_j - 1$. Note that $\mathbf{B}_j$s are disjoint and $b_j \leq l_{\max} - 1 \leq M$. Next, we will introduce $t+1$ subsets (we call them bins) $\mathbf{I}_1, \cdots, \mathbf{I}_{t+1}$ of $\mathbf{V}$, each with size at most M, such that for each $j \in \mathbf{A}$, there exists $\mathbf{I} \in \{\mathbf{I}_1, \cdots, \mathbf{I}_{t+1}\}$ such that $\mathbf{B}_j \subseteq \mathbf{I}$ but $X_i^j \notin \mathbf{I}$. It is noteworthy that this problem is a special case of *bin-packing problem.* For simplicity, suppose $\mathbf{A} = \{j_1, \cdots, j_a\}$, where $a = |\mathbf{A}|$. We initialize the bins with empty sets. Then, we add $\mathbf{B}_j$s to them in a greedy manner such that the size of bins remains less than M. That is, we first add the variables in $\mathbf{B}_{j_1}$ to $\mathbf{I}_1$. Note that this is feasible since $|\mathbf{B}_{j_1}| \leq M$. Then, we add $\mathbf{B}_{j_2}$ to the first feasible bin, i.e., the first bin, such that its size remains less than M after adding the elements of $\mathbf{B}_{j_2}$ to it. We subsequently add the elements of $\mathbf{B}_j$s to the first feasible bin. It is left to show that there always exists a feasible bin during this process. Suppose $\mathbf{B}_{j_1}, \cdots, \mathbf{B}_{j_x}$ are already placed in the bins, where $1 \leq x < a$, and we want to find a feasible bin for $\mathbf{B}_{j_{x+1}}$. Assume by contradiction that there is no feasible bin for $\mathbf{B}_{j_{x+1}}$. This shows that adding $\mathbf{B}_{j_{x+1}}$ to any bin results in a bin with at least $M+1$ elements. Hence,

$$(t+1)(M - b_{j_{x+1}} + 1) \leq b_{j_1} + \cdots + b_{j_x}. \tag{9}$$

On the other hand, $\mathbf{B}_{j_1} \cup \cdots \cup \mathbf{B}_{j_x}$ does not intersect with $\mathbf{B}_{j_{x+1}}$ and does not include any of the variables in $\{X_i^{j_1}, \cdots, X_i^{j_{x+1}}\}$. Hence,

$$b_{j_1} + \cdots + b_{j_x} \leq n - (b_{j_{x+1}} + x + 1). \tag{10}$$

Note that $b_{j_{x+1}} \leq l_{\max} - 1$ and $x \geq 1$. Hence, Equations (9) and (10) imply that

$$\lfloor \frac{n - l_{\max} - 1}{M - l_{\max} + 2} \rfloor + 1 = t + 1 \leq \frac{n - (b_{j_{x+1}} + x + 1)}{M - b_{j_{x+1}} + 1} \leq \frac{n - l_{\max} - 1}{M - l_{\max} + 2},$$

which is a contradiction. This shows that it is feasible to add all the $\mathbf{B}_j$s to the bins in a greedy manner, and therefore, the constructed $t+1$ subsets satisfy our claim.

Finally, if we repeat the whole process for each $1 \leq i \leq l_{\max}$, the constructed subsets will form a lifted separating system. Note that the total number of subsets will equal $l_{\max}(1+t)$, which is our desired bound. ■

Equipped with Theorem 46, we can obtain a lifted separating system such that the size of its elements is bounded by M. Moreover, by setting $M = \zeta_{\max}(\mathcal{G}) - 1$ in Theorem 46, we get the following notable corollary.

Corollary 47 *DG $\mathcal{G}$ can be learned by performing experiments with size at most $\zeta_{\max}(\mathcal{G})-1$. Hence, the lower bound in Theorem 22 is tight.*

To sum up this section, our algorithms can learn a DG $\mathcal{G}$ with n vertices by performing at most

$$\lceil \frac{n}{M} \rceil \lceil \log_{\lceil \frac{n}{M} \rceil} n \rceil + \zeta_{\max}(\mathcal{G})(1 + \lfloor \frac{n-\zeta_{\max}(\mathcal{G})-1}{M-\zeta_{\max}(\mathcal{G})+2} \rfloor) \tag{11}$$

experiments with size at most M, where $\zeta_{\max}(\mathcal{G})-1 \leq M < n$.

Remark 48 *Similar to Remark 41, when $\mathcal{G}$ is a DAG, the first stage of the algorithm uniquely learns $\mathcal{G}$. Hence, the algorithm only performs $\lceil \frac{n}{M} \rceil \lceil \log_{\lceil \frac{n}{M} \rceil} n \rceil$ experiments.*

7. Simulation Results

In this section, we evaluate the performance of the proposed method over random graphs generated from a variant of stochastic block models (SBMs).[3]

7.1 Graph Generation

In an SBM(n, p, b), a graph $\mathcal{G}$ with n vertices is generated as follows: the variables are randomly partitioned into $\lceil n/b \rceil$ blocks: $\mathbf{B}_1, \cdots, \mathbf{B}_{\lceil n/b \rceil}$, where $|\mathbf{B}_i| = b$ for $1 \leq i \leq \lceil n/b \rceil - 1$. For two variables in the same block, there can exist an edge in both directions, each with probability p. For two variables in different blocks, there can be an edge between them with probability p only in one direction. That is, directed edge (X, Y) exists with probability p when $X \in \mathbf{B}_i$ and $Y \in \mathbf{B}_j$, where $1 \leq i < j \leq \lceil n/b \rceil$. This means that the variables in each SCC belong to the same block, and b is a surrogate for $\zeta_{\max}(\mathcal{G})$.

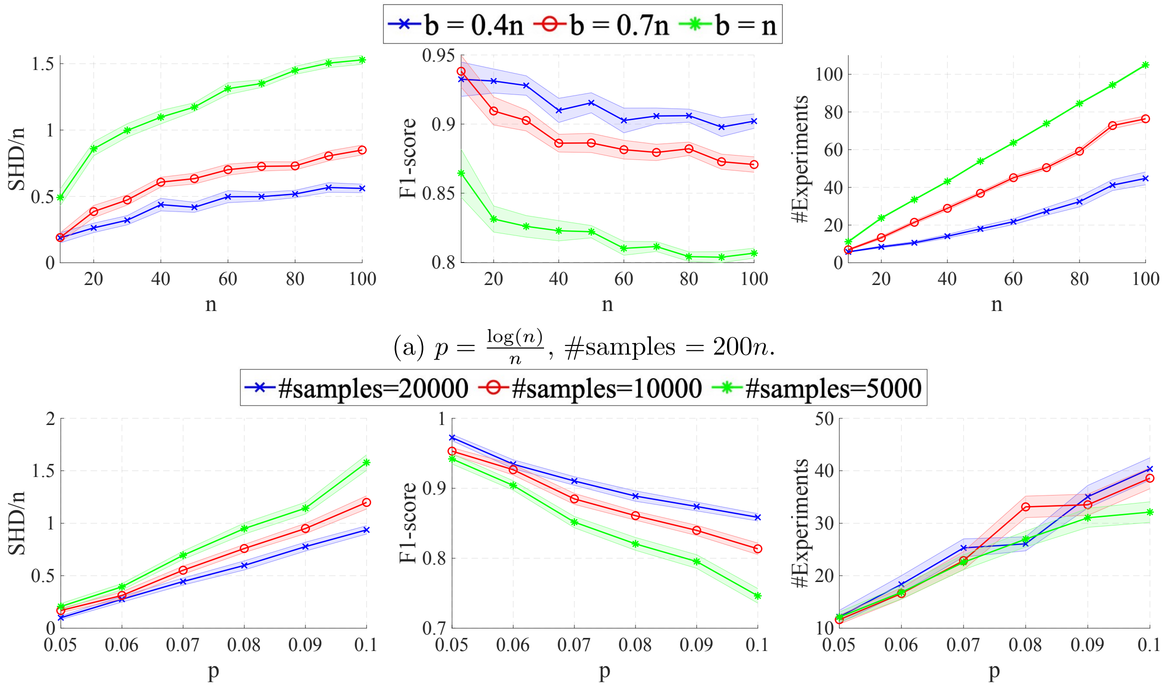
7.2 Data Generation

For each graph, synthetic data sets from observational and interventional distributions were generated with a finite number of samples and fed to our proposed algorithm. The observational samples were generated using a linear SCM where each variable X is a linear combination of its parents plus an exogenous noise variable ϵ_X; the coefficients were chosen uniformly at random from $[-1.5, -1] \cup [1, 1.5]$, and ϵ_X was generated at random according to $\mathcal{N}(0, \sigma_X^2)$, where σ_X is selected uniformly at random from $[\sqrt{0.5}, \sqrt{1.5}]$. To generate interventional samples for an experiment on a subset $\mathbf{I} \subseteq \mathbf{V}$, the equation of each variable in $\mathbf{V} \setminus \mathbf{I}$ remained unchanged, and the equation of each variable $X \in \mathbf{I}$ was replaced by $X = \epsilon_X$, where ϵ_X had the same distribution as in the original SCM.

7.3 Implementation Details

For the simulations of this section, we used the structure learning algorithm in Mokhtarian et al. (2022) to learn $\mathcal{G}_r^{obs}$, as it is scalable to large graphs. We note that due to Remark 33, the algorithm does not need to be *complete* (even for DAGs), as we just need $\mathcal{G}_r^{obs}$ to be a supergraph of the skeleton of $\mathcal{G}$. Hence, we can exploit any constraint-based causal

[3] Our codes are available at `https://github.com/Ehsan-Mokhtarian/cyclic_experiment_design`.



(a) $p = \frac{\log(n)}{n}$, #samples $= 200n$.

(b) $n = 50$, $b = 25$.

Figure 6: Performance of our approach on random graphs generated from $\mathrm{SBM}(n, p, b)$.

discovery from observational data method that is *sound* (but not necessarily *complete*) for DAGs to learn $\mathcal{G}_r^{obs}$.

To color $\mathcal{G}_r^{obs}$, we applied *trail-path* algorithm in Bandyopadhyay et al. (2020). To find the descendant sets and the strongly connected components of $\mathcal{H}$ in line 9 of Algorithm 1, we used the predefined function *conncomp* in MATLAB. Finally, we used Fisher Z-transformation with a significance level of 0.01 to perform conditional independence tests.

7.4 Results

In Figure 6, we report the number of experiments performed by our proposed method and the accuracy of the learned graphs when the underlying true graphs are generated randomly from $\mathrm{SBM}(n, p, b)$. Each point on the plots is reported as the average of 50 runs with a 90% confidence interval. We measured the accuracy of the recovered DGs by normalized structural hamming distance (SHD$/n$) and F1-scores, which we formally define in Subsection 7.5.

Figure 6a illustrates the effect of n (number of vertices) and b (the parameter that controls $\zeta_{\max}(\mathcal{G})$) when $p = \frac{\log(n)}{n}$ (graph density) and the number of samples was fixed at $200n$. As can be seen, for moderate values of b, and accordingly $\zeta_{\max}(\mathcal{G})$, the proposed algorithm achieves good accuracy in terms of F1-score and SHD. Moreover, the number of experiments scales linearly with b, which is consistent with our analysis.

In Figure 6b, the effect of graph density is studied by varying p for three different sample sizes (5000, 10000, 20000) when $n = 50$ and $b = 25$. This shows that once we reach an adequate number of samples, the sample size has a negligible effect on the number of experiments. Furthermore, we observe that the proposed approach performs better (both in terms of SHD and F1-score) on sparser graphs.

7.5 Evaluation Metrics

We measured the accuracy of our algorithm by two commonly used metrics in the literature: F1-score and normalized Structural Hamming Distance (SHD/n). Herein and similar to Mokhtarian et al. (2023), we define these measures.

Let $\mathcal{G}_1$ and $\mathcal{G}_2$ denote the true DG and the learned DG, respectively. We first define a few notations. True-positive (TP) is the number of edges that appear in both $\mathcal{G}_1$ and $\mathcal{G}_2$. False-positive (FP) is the number of edges that appear in $\mathcal{G}_2$ but do not exist in $\mathcal{G}_1$. False-negative (FN) is the number of edges in $\mathcal{G}_1$ that the algorithm failed to learn in $\mathcal{G}_2$. In this case, SHD is defined as follows.

$$\text{SHD} = \text{FP} + \text{FN}, \quad \text{SHD}/n = \frac{\text{FP} + \text{FN}}{n}.$$

SHD is a non-negative integer, and smaller numbers indicate better accuracy. F1-score is defined by *precision* and *recall* in the following.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{F1-score} = 2 \times \frac{\text{Precision} \ \times \ \text{Recall}}{\text{Precision} \ + \ \text{Recall}}.$$

Note that $0 \leq$ F1-score ≤ 1 and larger numbers indicate better accuracy.

8. Related Work

The goal of causal discovery is to learn the causal graph of a system, which represents the existence and direction of relations among the variables of the system under study. In general, the causal graph can only be identified up to the Markov equivalence class (MEC) from mere observational data. Richardson (1996b) provided necessary and sufficient conditions for the Markov equivalence of two DGs, based on which he proposed a consistent structure learning algorithm that can learn a DG up to the MEC (Richardson, 1996a). Subsequently, Mooij and Claassen (2020) showed that the Fast Causal Inference (FCI) algorithm, originally designed for learning DAGs, can also learn a cyclic DG up to the MEC. Forré and Mooij (2018) introduced σ-connection graphs (σ-CG), a new class of mixed graphs (containing undirected, bidirected, and directed edges). They proposed a causal discovery algorithm for σ-CGs, handling non-linear causal mechanisms, latent confounders, and data from multiple interventional distributions. Ghassami et al. (2020) instead focused on the notion of distribution equivalence. They provided necessary and sufficient conditions for the distribution equivalence of two DGs for linear Gaussian causal DG models and proposed a score-based method for learning the structure from observational data. Lacerda et al. (2008) focused on the case of linear models with non-Gaussian noises and generalized the ICA-based approach of Shimizu et al. (2006) to allow for cycles.

As we discussed, to uniquely identify the causal graph, the gold standard is to perform experiments, leading to the experiment design problem. To the best of our knowledge, there is no previous work on the problem of experiment design in cyclic models. In the following, we mainly review previous work in acyclic models, where it has been studied extensively (Eberhardt, 2007; Eberhardt et al., 2005; Eberhardt, 2008; He and Geng, 2008; Shanmugam et al., 2015). We review the previous work based on the following three aspects of the experiment design problem.

- **The objective of the problem:** The work on experiment design can be divided into two main categories. In the first category, the goal is to minimize the cost of experiments while it is required to learn the whole graph. This problem is referred to as the *min-cost identification* problem. The second category aims to minimize the ambiguity about the causal graph while a limited budget for performing experiments is available. This problem is referred to as *fixed budget* or *budgeted experiment design.*

- **Adaptive versus non-adaptive methods:** An alternative way to divide methods is in terms of whether the interventions are performed adaptively or non-adaptively. *Adaptive* methods sequentially perform experiments, where they exploit the results of previously performed experiments to design the latter ones. These methods are practical in cases where the experiments are not highly time-consuming. On the other hand, *non-adaptive* methods design all the experiments simultaneously and perform them in parallel.

- **Bounded-size experiments:** In several applications, it is not feasible to perform large-size experiments. In such cases, the size of the designed experiments must be bounded by a given constant. This problem is referred to as the *Bounded-size experiment design* problem.

The majority of earlier work focused on the min-cost identification problem in acyclic models. In particular, Eberhardt et al. (2005) proposed worst-case bounds on the number of required experiments where the number of intervened variables could be as large as half of the size of the graph. He and Geng (2008) proposed adaptive and non-adaptive algorithms for the case where the experiments are singleton, i.e., each experiment is comprised of a single variable. Their non-adaptive approach is brute force, and it can find the optimal solution. However, it is not scalable to large graphs as they enumerate all the DAGs in a MEC, and the number of DAGs in a MEC can grow super-exponentially with the number of variables. In the adaptive case, they presented a heuristic algorithm based on Shannon's entropy to select the intervened variable in each step. Hauser and Bühlmann (2014) proposed an optimal algorithm for minimizing the number of undirected edges in the worst case when we are allowed to perform just one intervention. They further utilized this algorithm to propose a heuristic adaptive experiment design method. Shanmugam et al. (2015) proposed a lower bound on the number of experiments for the adaptive methods based on the notion of separating systems. Kocaoglu et al. (2017b) proposed a stage-wise algorithm for the experiment design problem in the presence of unobserved variables. First, the induced subgraph between observed variables is recovered, and then, by performing some "do-see" tests, the existence and the location of latent variables are identified.

The experiment design problem has also been studied when intervention on each variable has a particular cost. In this setting, Kocaoglu et al. (2017a) proposed an optimal algorithm when there is no constraint on the number of interventions in each experiment. Greenewald et al. (2019) presented a 2-approximation adaptive algorithm for the tree causal structures.

In a follow-up, Squires et al. (2020) proposed an adaptive algorithm for a more general class of causal graphs, matching the optimal number of interventions up to a multiplicative logarithmic factor.

Ghassami et al. (2018) introduced the fixed budget formulation of the experiment design problem. They considered the average number of recovered edges (after an intervention) as the objective function and showed that a general greedy algorithm is an approximation algorithm. Moreover, to estimate the objective function, they proposed a sampler from MEC, which evaluates the objective function by a Monte Carlo scheme. Ghassami et al. (2019b) presented a uniform sampler on clique trees for accelerating the generating of random DAGs from a given MEC. Then, they utilized it as a sub-routine for designing experiments. Ghassami et al. (2019a) proposed an efficient exact algorithm for tree causal structures to minimize the number of undirected edges after performing interventions in the worst-case scenario. Later, AhmadiTeshnizi et al. (2020) proposed a method for iterating over all possible DAGs in the corresponding MEC after intervening on a variable, and introduced an exact algorithm for the fixed budget problem. The methods described above can be further reinforced by using state-of-the-art techniques for counting and sampling Markov equivalent DAGs. In particular, Wienöbst et al. (2021, 2023) show that these tasks can be performed in polynomial time.

The experiment design problem has also been studied in the Bayesian framework. For instance, Agrawal et al. (2019) proposed a tractable adaptive algorithm for the fixed budget problem with an approximation guarantee on sub-modularity. Tigas et al. (2022) proposed an adaptive experiment design method that designs not only the experiments but also the value at which each intervened variable should be set.

9. Conclusion and Future Work

Feedback cycles in causal graphs are more the norm rather than the exception. We showed that in cyclic models, observational data is far less informative for structure learning, and it is necessary to solve the experiment design problem. The presence of cycles also introduces major challenges for the experiment design. For instance, intervening on a variable may not lead to recovering the presence or the direction of the edges incident to it.

In this work, we proposed a unified experiment design framework that allows learning cyclic and acyclic graphs. We further provided a theoretical analysis to calculate the required number and size of experiments in the worst case. The analysis demonstrated that our proposed approach is order-optimal in terms of the number of experiments up to an additive logarithmic term and optimal in terms of the size of the largest experiment required for unique identification of the causal graph in the worst case.

In the following, we discuss potential future work.

- The main assumption of our proposed method is causal sufficiency. An important unsolved research problem is to relax this assumption and allow for latent confounders. We note that in the presence of latent confounders and even in acyclic models, experiment design is a challenging problem.

- Although we assumed that the generative model is a simple SCM, for the soundness of our results, we only required that the interventional distribution exists (not necessarily unique) and that the CI assertions in the observational and interventional distributions are equivalent to either d-separation or σ-separations in the causal graph. Accordingly, another direction of future work is to characterize the class of SCMs satisfying the aforementioned assumptions.

- In Scenario 2, we considered the σ-faithfulness assumption to ensure that any CI in the distribution implies σ-separation in the causal graph. As mentioned in Remark 9, σ-faithfulness assumption is stronger than d-faithfulness assumption. It could be interesting to investigate how restrictive the assumption of σ-faithfulness is.
- We assumed that intervening in a variable removes its incoming edges. This type of intervention is commonly called *hard intervention* (aka, perfect intervention). However, there are other types of interventions, such as *soft-interventions*, in which the incoming edges will not necessarily be omitted (even in some cases, new edges will be added to the causal graph). Studying the problem of experiment design and investigating the required number and size of experiments under other types of interventions remains open.
- The lower bounds presented in Theorems 22 and 24 are *worst-case* lower bounds, in the sense that for any constant c, there exists a DG $\mathcal{G}$ with $\zeta_{\max}(\mathcal{G}) = c$, which requires at least the specified number or size of interventions to be identified uniquely. A few recent works such as Choo et al. (2022) have established *instance-wise* lower bounds for experiment design on DAGs. These bounds can be used to develop instance-wise competitive guarantees for experiment design algorithms. The development of such bounds for cyclic causal models remains an important open problem.

Acknowledgments

We thank our anonymous reviewer for their valuable comments. This research was in part supported by the Swiss National Science Foundation under NCCR Automation, grant agreement 51NF40_180545 and Swiss SNF project 200021_204355 /1.

References

Raj Agrawal, Chandler Squires, Karren Yang, Karthikeyan Shanmugam, and Caroline Uhler. Abcd-strategy: Budgeted experimental design for targeted causal structure discovery. In *International Conference on Artificial Intelligence and Statistics*, pages 3400–3409. PMLR, 2019.

Ali AhmadiTeshnizi, Saber Salehkaleybar, and Negar Kiyavash. Lazyiter: a fast algorithm for counting markov equivalent dags and designing experiments. In *International Conference on Machine Learning*, pages 125–133. PMLR, 2020.

Sina Akbari, Ehsan Mokhtarian, AmirEmad Ghassami, and Negar Kiyavash. Recursive causal structure learning in the presence of latent variables and selection bias. *Advances in Neural Information Processing Systems*, 34:10119–10130, 2021.

Abhirup Bandyopadhyay, Amit kumar, and Sankar Basu. Graph coloring: a novel heuristic based on trailing path—properties, perspective and applications in structured networks. *Soft Computing*, 24(1):603–625, 2020.

Stephan Bongers, Patrick Forré, Jonas Peters, and Joris M Mooij. Foundations of structural causal models with cycles and latent variables. *The Annals of Statistics*, 49(5):2885–2915, 2021.

Davin Choo, Kirankumar Shiragur, and Arnab Bhattacharyya. Verification and search algorithms for causal dags. *Advances in Neural Information Processing Systems*, 35: 12787–12799, 2022.

Frederick Eberhardt. Causation and intervention. *Unpublished doctoral dissertation, Carnegie Mellon University*, 2007.

Frederick Eberhardt. Almost optimal intervention sets for causal discovery. In *Conference on Uncertainty in Artificial Intelligence*, 2008.

Frederick Eberhardt, Clark Glymour, and Richard Scheines. On the number of experiments sufficient and in the worst case necessary to identify all causal relations among n variables. In *Conference on Uncertainty and Artificial Intelligence*, pages 178–184, 2005.

Patrick Forré and Joris M Mooij. Markov properties for graphical models with cycles and latent variables. *arXiv preprint arXiv:1710.08775*, 2017.

Patrick Forré and Joris M Mooij. Constraint-based causal discovery for non-linear structural causal models with cycles and latent confounders. In *Conference on Uncertainty in Artificial Intelligence*, 2018.

AmirEmad Ghassami, Saber Salehkaleybar, Negar Kiyavash, and Elias Bareinboim. Budgeted experiment design for causal structure learning. In *International Conference on Machine Learning*, pages 1724–1733. PMLR, 2018.

AmirEmad Ghassami, Saber Salehkaleybar, and Negar Kiyavash. Interventional experiment design for causal structure learning. *arXiv preprint arXiv:1910.05651*, 2019a.

AmirEmad Ghassami, Saber Salehkaleybar, Negar Kiyavash, and Kun Zhang. Counting and sampling from markov equivalent dags using clique trees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3664–3671, 2019b.

AmirEmad Ghassami, Alan Yang, Negar Kiyavash, and Kun Zhang. Characterizing distribution equivalence and structure learning for cyclic and acyclic directed graphs. In *International Conference on Machine Learning*, pages 3494–3504. PMLR, 2020.

Kristjan Greenewald, Dmitriy Katz, Karthikeyan Shanmugam, Sara Magliacane, Murat Kocaoglu, Enric Boix Adsera, and Guy Bresler. Sample efficient active learning of causal trees. *Advances in Neural Information Processing Systems*, 32, 2019.

Alain Hauser and Peter Bühlmann. Two optimal strategies for active learning of causal models from interventional data. *International Journal of Approximate Reasoning*, 55 (4):926–939, 2014.

Yang-Bo He and Zhi Geng. Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research*, 9(Nov):2523–2547, 2008.

Gyula Katona. On separating systems of a finite set. *Journal of Combinatorial Theory*, 1 (2):174–194, 1966.

Murat Kocaoglu, Alex Dimakis, and Sriram Vishwanath. Cost-optimal learning of causal graphs. In *International Conference on Machine Learning*, pages 1875–1884. PMLR, 2017a.

Murat Kocaoglu, Karthikeyan Shanmugam, and Elias Bareinboim. Experimental design for learning causal graphs with latent variables. *Advances in Neural Information Processing Systems*, 30, 2017b.

Gustavo Lacerda, Peter L Spirtes, Joseph Ramsey, and Patrik O Hoyer. Discovering cyclic causal models by independent components analysis. In *Conference on Uncertainty in Artificial Intelligence*, pages 366–374, 2008.

Erik Lindgren, Murat Kocaoglu, Alexandros G Dimakis, and Sriram Vishwanath. Experimental design for cost-aware learning of causal graphs. *Advances in Neural Information Processing Systems*, 31, 2018.

Ehsan Mokhtarian, Sina Akbari, AmirEmad Ghassami, and Negar Kiyavash. A recursive markov boundary-based approach to causal structure learning. In *The KDD'21 Workshop on Causal Discovery*, pages 26–54. PMLR, 2021.

Ehsan Mokhtarian, Sina Akbari, Fateme Jamshidi, Jalal Etesami, and Negar Kiyavash. Learning bayesian networks in the presence of structural side information. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

Ehsan Mokhtarian, Mohmmadsadegh Khorasani, Jalal Etesami, and Negar Kiyavash. Novel ordering-based approaches for causal structure learning in the presence of unobserved variables. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12260–12268, 2023.

Joris M Mooij and Tom Claassen. Constraint-based causal discovery using partial ancestral graphs in the presence of cycles. In *Conference on Uncertainty in Artificial Intelligence*, pages 1159–1168. PMLR, 2020.

Judea Pearl. *Causality*. Cambridge university press, 2009.

J Patrick Pett, Matthew Kondoff, Grigory Bordyugov, Achim Kramer, and Hanspeter Herzel. Co-existing feedback loops generate tissue-specific circadian rhythms. *Life science alliance*, 1(3), 2018.

Thomas S Richardson. A discovery algorithm for directed cyclic graphs. In *Conference on Uncertainty in Artificial Intelligence*, pages 454–461, 1996a.

Thomas S Richardson. A polynomial-time algorithm for deciding Markov equivalence of directed cyclic graphical models. In *Conference on Uncertainty in Artificial Intelligence*, pages 462–469, 1996b.

Thomas Schaffter, Daniel Marbach, and Dario Floreano. Genenetweaver: in silico benchmark generation and performance profiling of network inference methods. *Bioinformatics*, 27(16):2263–2270, 2011.

Karthikeyan Shanmugam, Murat Kocaoglu, Alexandros G Dimakis, and Sriram Vishwanath. Learning causal graphs with small interventions. *Advances in Neural Information Processing Systems*, 28, 2015.

Micha Sharir. A strong-connectivity algorithm and its applications in data flow analysis. *Computers & Mathematics with Applications*, 7(1):67–72, 1981.

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

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

Chandler Squires, Sara Magliacane, Kristjan Greenewald, Dmitriy Katz, Murat Kocaoglu, and Karthikeyan Shanmugam. Active structure learning of causal dags via directed clique trees. *Advances in Neural Information Processing Systems*, 33:21500–21511, 2020.

Panagiotis Tigas, Yashas Annadani, Andrew Jesson, Bernhard Schölkopf, Yarin Gal, and Stefan Bauer. Interventions, where and how? experimental design for causal models at scale. *Advances in Neural Information Processing Systems*, 35:24130–24143, 2022.

Thomas Verma and Judea Pearl. Equivalence and synthesis of causal models. In *Conference on Uncertainty in Artificial Intelligence*, pages 255–270, 1990.

Marcel Wienöbst, Max Bannach, and Maciej Liskiewicz. Polynomial-time algorithms for counting and sampling markov equivalent dags. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12198–12206, 2021.

Marcel Wienöbst, Max Bannach, and Maciej Liśkiewicz. Polynomial-time algorithms for counting and sampling markov equivalent dags with applications. *Journal of Machine Learning Research*, 24(213):1–45, 2023.