CAUSAL DISCOVERY BEYOND MARKOV EQUIVALENCE

BY

AMIREMAD GHASSAMI

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2020

Urbana, Illinois

Doctoral Committee:

Adjunct Associate Professor Negar Kiyavash, Chair
Assistant Professor Sanmi Koyejo
Associate Professor Maxim Raginsky
Professor Rayadurgam Srikant
Associate Professor Kun Zhang, Carnegie Mellon University

ABSTRACT

The focus of the dissertation is on learning causal diagrams beyond Markov equivalence. The baseline assumptions in causal structure learning are the acyclicity of the underlying structure and causal sufficiency, which requires that there are no unobserved confounder variables in the system. Under these assumptions, conditional independence relationships contain all the information in the distribution that can be used for structure learning. Therefore, the causal diagram can be identified only up to Markov equivalence, which is the set of structures reflecting the same conditional independence relationships. Hence, for many ground truth structures, the direction of a large portion of the edges will remain unidentified. Hence, in order to learn the structure beyond Markov equivalence, generating or having access to extra joint distributions from the perturbed causal system is required. There are two main scenarios for acquiring the extra joint distributions. The first and main scenario is when an experimenter is directly performing a sequence of interventions on subsets of the variables of the system to generate interventional distributions. We refer to the task of causal discovery from such interventional data as *interventional causal structure learning*. In this setting, the key question is determining which variables should be intervened on to gain the most information. This is the first focus of this dissertation. The second scenario for acquiring the extra joint distributions is when a subset of causal mechanisms, and consequently the joint distribution of the system, have varied or evolved due to reasons beyond the control of the experimenter. In this case, it is not even a priori known to the experimenter which causal mechanisms have varied. We refer to the task of causal discovery from such multi-domain data as *multi-domain causal structure learning*. In this setup the main question is how one can take the most advantage of the changes across domains for the task of causal discovery. This is the second focus of this dissertation.

Next, we consider cases under which conditional independency may not reflect all the information in the distribution that can be used to identify the underlying structure. One such case is when cycles are allowed in the underlying structure. Unfortunately, a suitable characterization for equivalence for the case of cyclic directed graphs has been unknown so

far. The third focus of this dissertation is on bridging the gap between cyclic and acyclic directed graphs by introducing a general approach for equivalence characterization and structure learning. Another case in which conditional independency may not reflect all the information in the distribution is when there are extra assumptions on the generating causal modules. A seminal result in this direction is that a linear model with non-Gaussian exogenous variables is uniquely identifiable. As the forth focus of this dissertation, we consider this setup, yet go one step further and allow for violation of causal sufficiency, and investigate how this generalization affects the identifiability.

To my parents and my sister, for their love and support.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my adviser, Prof. Negar Kiyavash. I would like to thank her for all her guidance while carrying out this work, for the freedom she gave me in choosing my research style, and her unequivocal support in all academic aspects. She was always available to give me advice and profoundly shaped my way of thinking about research, academia and life. I do not have words to express my gratitude to her.

No less am I indebted to Prof. Kun Zhang. During two visits that I had to Carnegie Mellon University, he shaped my thoughts about the field of causal inference and discovery. While working on research problems together, he pointed me in directions and led the project in a way that I do not think I would have reached on my own. Also, as anyone how knows Kun will agree, he is one of the nicest people one can ever meet.

I would like to thank my doctoral committee, Profs. Negar Kiyavash, Sanmi Koyejo, Maxim Raginsky, Rayadurgam Srikant, and Kun Zhang, as well as my great collaborators and friends Prof. Saber Salehkaleybar, Prof. Elias Bareinboim, Alan Yang, and Biwei Huang for their help and contributions to this dissertation.

I am in debt to the wonderful teachers that I had at UIUC, especially the following: Prof. Bruce Hajek, who formed the foundation of my knowledge of probability theory; Prof. Yihong Wu, who taught me how I should think about information theory; Prof. Rayadurgam Srikant, who taught me several topics such as optimization, game theory, and queueing theory; Prof. Xiaochun Li, who taught me real analysis; Prof. Jozsef Balogh, who taught me combinatorial mathematics and in general combinatorial thinking; and Prof. Maxim Raginsky, who taught me statistical learning theory.

I would also like to thank my wonderful friends at UIUC, especially Corinne Soutar, James Schmidt, Massi Amrouche, and Erman Gungor, who made my stay in Champaign-Urbana such a memorable and enjoyable experience.

I would also like to thank the great staff and managers of Café Kopi in downtown Champaign. A big part of developing the ideas and preparation of this dissertation was done in the calm environment of this amazing coffee shop.

Last but certainly not least, I would like to thank my parents and my sister for their love and support.

TABLE OF CONTENTS

PUBLICATIONS ON WHICH THE DISSERTATION IS BASED

- A. Ghassami, S. Salehkaleybar, and N. Kiyavash, "Interventional Experiment Design for Causal Structure Learning," arXiv preprint arXiv:1910.05651.
- A. Ghassami, S. Salehkaleybar, B. Huang, N. Kiyavash, and K. Zhang, "Multi-Domain Causal Structure Learning," under preparation.
- A. Ghassami, A.Yang, N. Kiyavash, and K. Zhang, "Characterizing Distribution Equivalence for Cyclic and Acyclic Directed Graphs," Proceedings of the International Conference on Machine Learning (ICML), 2020.
- S. Salehkaleybar, A. Ghassami, N. Kiyavash, and K. Zhang, "Learning Linear Non-Gaussian Causal Models in the Presence of Latent Variables," Journal of Machine Learning Research (JMLR), 2020.
- A. Ghassami, S. Salehkaleybar, N. Kiyavash, and K. Zhang, "Counting and Sampling from Markov Equivalent DAGs Using Clique Trees," Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI), 2019.
- A. Ghassami, N. Kiyavash, B. Huang, and K. Zhang, "Multi-Domain Causal Structure Learning in Linear Systems," Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), 2018.
- A. Ghassami, S. Salehkaleybar, N. Kiyavash, and E. Bareinboim, "Budgeted Experiment Design for Causal Structure Learning," Proceedings of the International Conference on Machine Learning (ICML), 2018.
- A. Ghassami, S. Salehkaleybar, N. Kiyavash, and K. Zhang. "Learning Causal Structures Using Regression Invariance," Proceedings of the Advances in Neural Information Processing Systems (NIPS), 2017.

CHAPTER 1

INTRODUCTION

Since the dawn of modern science, a predominant theme in scientific research in various areas such as biology, sociology, medicine, etc., has been centered on discovering and understanding the causal relationships among components of the systems under study. This great interest in learning the causal relations is primarily in pursuit of two fundamental goals:

First and foremost, a natural desire of human beings is to explain different phenomena around them. Whether it is a question regarding why the sun rises at a certain time, or what leads to happiness in life, the curious mind is constantly searching for satisfactory answers. Yet, it is of course not possible to find such answers and fully comprehend a phenomenon without knowing the direct causes leading to that. The knowledge regarding the causal relations provides us with the ability to explain and describe different phenomena regarding the system and to provide answers to *"why"* and *"how"* questions.

The second aim is to acquire the ability to estimate properties related to an unseen distribution. The standard statistical inference discusses the task of estimating features of the distribution from which the data is gathered. Causal inference goes one step further and aims to estimate features of the distribution after changes. It provides us with the ability to predict the consequences of changes and interventions in a system. We may have never observed any samples from the intervened system; however, the knowledge regarding causal modules enables us to predict the output of the system after any variations to the causes. Examples of this ability include predicting the future state of the market based on the changes and events that occurred today in the economic world, and predicting the consequences of enforcing a law by the government. This ultimately enables us to design actions and intervene in the system in a specific way to drive it to a desired state (i.e., to control it). Causal inference also provides us with the ability to provide answers to *"what if"* questions, which is known as counterfactual reasoning. This is again beyond the realm of the abilities of the common current AI, as it requires reasoning regarding a setup from which the machine has no observations. This ability is the driving force for creativity, innovation and invention and is the resource that enables us to imagine unseen events and objects such

as mythical creatures.

One of the most prominent approaches for modeling and representing causal relationships among variables in a system is to use a *structural causal model*. For a given set of *endogenous* variables $V = \{X_1, ..., X_p\}$, a structural causal model consists of a set of equations of the form $X_i = f_i(Pa(X_i), N_i)$, $1 \leq i \leq p$, where $Pa(X_i) \subseteq V \setminus \{X_i\}$ denotes the set of direct causes of variable X_i with respect to V, and N_i is the *exogenous* variable corresponding to X_i, representing noise or disturbance. Note that the equations in a structural causal model should be understood as generating mechanisms. A structural causal model induces a distribution P_V on the endogenous variables. Consider the directed graph generated by drawing a directed edge from each element of $Pa(X_i)$ to X_i, for all $i \in [p]$. The resulting directed graph G is called the *causal diagram*. The task of learning the causal diagram from data is referred to as *causal structure learning*, or *causal discovery*.

One of the main classes of the methods for causal structure learning is constraint-based methods. These methods are based on using the conditional independencies in the data distribution: The causal digram G represents certain disconnectivities among the variables of the system, called d-separations (defined in **Chapter 2**), and the distribution P_V represents certain conditional independencies among the variables. Under some assumptions (explained in detail in **Chapter 2**), there is a one-to-one correspondence between the d-separations and conditional independencies. That is, a set of variables X_S d-separates variables X_1 and X_2 in the causal digram if and only if X_1 is independent of X_2 conditioned on variables X_S. This is one of the main links connecting the graphical world and statistical world, which enables us to learn the causal graph over the variables from data gathered from those variables. However, for a set of d-separation relationships, the directed graph representing those d-separations is not in general unique. For instance, all three graphs $X_1 \to X_2 \to X_3$, $X_1 \leftarrow X_2 \to X_3$ and $X_1 \leftarrow X_2 \leftarrow X_3$ indicate that X_2 d-separates X_1 from X_3. Such diagrams that represent the same d-separation relations are called *Markov equivalent*, and the set of Markov equivalent graphs form a *Markov equivalence class*. There are several constraint-based algorithms for different assumptions on the system such as whether or not the underlying causal structure is acyclic, or whether or not there are any latent variables in the system [1, 2].

Another main class of methods for causal structure learning is the score-based methods (e.g., [3, 4]). Score-based methods focus on finding a structure which maximizes a score function, which mainly includes the likelihood function and a regularization term. There are also methods called hybrid methods (such as [5]) which aim to combine the advantages of the constraint-based and score-based methods.

There is another point of view towards causal structure learning which is focused on studying constraints on the generating causal modules, i.e., the f_i functions in the structural causal model, which lead to identifiability. For instance, it is shown that under assumptions on the underlying data generating processes, such as considering linear models with non-Gaussian exogenous variables [6, 7], or assuming specific types of non-linearity on the causal modules [8, 9, 10], in the population dataset, along with some extra assumptions, the underlying causal diagram can be identified uniquely.

1.1 Causal Discovery Beyond Markov Equivalence

First, we consider the case that the causal diagram is acyclic, i.e., it is a directed acyclic graph (DAG), and we have the causal sufficiency assumption which implies that there are no latent confounders (latent variables with more than two observed effect variables) in the system. These are usually the baseline assumptions used for causal structure learning. In this case, conditional independence relationships contain all the information in the distribution that can be used for structure learning. Therefore, the causal diagram can be identified only up to Markov equivalence, and hence, for many ground truth structures, the direction of a large portion of the edges will remain unidentified. Hence, in order to learn the structure beyond Markov equivalence, generating or having access to extra joint distributions from the perturbed causal system is required. There are two main scenarios for acquiring the extra joint distributions:

- The first and main scenario is when an experimenter is directly performing a sequence of interventions on subsets of the variables of the system to generate *interventional* distributions. We refer to the task of causal discovery from such interventional data as the interventional causal structure learning. In this setting, the key question is determining which variables should be intervened on to gain the most information. This is our focus in **Chapter 3**. We address this question by casting the problem of finding the best intervention target set as an optimization problem which seeks to maximize the number of causal relationships identified as a result of the interventions. We consider the problem in both the worst-case and the average-case settings. We demonstrate that the problem of intervention design can be cast as a sub-modular set function optimization problem and hence is efficiently solvable in its most general form.

- Although performing interventional experiments is the gold standard for causal dis-

covery, in many applications, intervening on certain variables in the system may be expensive, unethical, impossible, or even undefined. The second scenario for acquiring the extra joint distributions is when a subset of causal mechanisms, and consequently the joint distribution of the system have varied or evolved due to reasons beyond the control of the experimenter. In many real-life systems, the data generating distribution may vary over time, or the dataset could come from different domains and hence, not follow a single distribution. In this case, the experimenter is usually not even aware of which causal mechanisms have varied. While such data is usually problematic in statistical analysis, this property can be leveraged for the purpose of causal discovery. We refer to the task of causal discovery from such multi-domain data as the multi-domain causal structure learning. In this setup the main question is how can one take the most advantage of the changes across domains for the task of causal discovery. This is our focus in **Chapter 4**. We propose several efficient algorithms for this task.

Next, we consider cases under which conditional independency may not reflect all the information in the distribution that can be used to identify the underlying structure. The dominant cases are when (1) cycles are allowed in the structure, (2) causal sufficiency is violated, (3) there are extra assumptions on the generating causal modules.

Consider the case that the causal diagram may contain cycles. Most real-life causal systems contain feedback loops, since feedback is generally required to stabilize the system and improve performance in the presence of noise. Hence, the causal directed graph corresponding to such systems will be cyclic. In this case, in general, Markov equivalence is not the extent of identifiability: When cycles are allowed, conditional independence may not be a suitable notion for equivalence of two structures, as it does not reflect all the information in the distribution that is useful for identification of the underlying structure. That is, it is possible that two graphs can be distinguishable from observational data even though they are in the same Markov equivalence class. Unfortunately, a suitable characterization for equivalence for the case of cyclic directed graphs has been unknown so far. With the goal of bridging the gap between cyclic and acyclic directed graphs, in **Chapter 5**, we introduce a general approach for equivalence characterization and structure learning, capable of dealing with cycles in a causal model. We present a general, unified notion of equivalence based on the set of distributions that the directed graphs are able to generate. We propose an algebraic and graphical characterization of the equivalence of two directed graphs, be they cyclic or acyclic. Furthermore we propose a score-based method for structure learning from observational data with local search, where we show that the proposed score asymptotically achieves the extent

of identifiability.

As mentioned earlier, another point of view towards causal structure learning is focusing on constraints on the generating causal modules. A seminal result in this direction is that under the assumptions of acyclicity and no latent confounders, a linear model with non-Gaussian exogenous variables is uniquely identifiable [6]. In **Chapter 6**, we focus on the same setup, yet go one step further and relax the assumption of no latent confounders. One of the main challenges in causal discovery is accounting for variables in the system from which we have no observations. For instance, latent confounders can lead to estimating *spurious causal relations*. It is often explicitly assumed that there are no latent confounders in the systems to avoid complications caused by those variables. Unfortunately, this assumption does not hold true in many settings. We consider learning causal diagrams from observational data generated by linear non-Gaussian causal models with latent variables. Despite the fact that the causal structure in general is not fully identifiable in the presence of latent variables, we show that the causal order among the observed variables is still identifiable. We also provide necessary and sufficient graphical conditions under which the number of latent variables is uniquely identifiable.

Table 1.1 summarizes how each chapter of this dissertation addresses the problem of causal discovery beyond Markov equivalence.

Table 1.1: How each chapter goes beyond Markov equivalence.

	Use more than one distribution	Allow for cycles	Allow for latent confounders	Use causal module identifiability constraints
Chapter 3	✓			
Chapter 4	✓			
Chapter 5		✓		
Chapter 6			✓	✓

CHAPTER 2

PRELIMINARIES

In this chapter we briefly review concepts and classical results from the fields of graph theory, graphical models and causal structure learning, needed in the rest of the dissertation. For the definitions in this chapter, we mainly follow [11], [1], and [12].

2.1 Graphical Notation and Terminology

A graph G is a pair $G = (V(G), E(G))$, where $V(G)$ is a finite set of vertices and $E(G)$, the set of edges, is a subset of $(V \times V) \setminus \{(a, a) : a \in V\}$. If for an edge $(a, b) \in E(G)$ its opposite edge, i.e., (b, a), also belongs to $E(G)$ then this edges is called an *undirected edge*, and we write $a - b \in G$. If for an edge $(a, b) \in E(G)$, we have $(b, a) \notin E(G)$, then this edge is called a *directed edge*, and we write $a \to b \in G$. In this case, vertex a is called a *parent* of vertex b and b is called a *child* of a. The set of parents and children of vertex a are denoted by $Pa(a)$ and $Ch(a)$, respectively. For vertex a, the set of vertices b such that $(a, b) \in E(G)$ or $(b, a) \in E(G)$ is called the set of *neighbors* of a, and is denoted by $N(a)$. A graph is called directed if all of its edges are directed, and is called undirected if all of its edges are undirected. A vertex is called *root* if all of its neighbors are its children, and is called *sink* if all of its neighbors are its parents. An undirected graph G^s, for which $V(G^s) = V(G)$ and $E(G^s) = E(G) \cup \{(a, b) : (b, a) \in E(G)\}$ is called the *skeleton* of G. For a subset of vertices $A \subseteq V(G)$ the *induced subgraph* of G on A is the graph $G[A] \coloneqq (A, E[A])$, where $E[A] \coloneqq E(G) \cap (A \times A)$.

A sequence of distinct vertices $(a_1, a_2, ..., a_m)$ is called a *path* from a_1 to a_m if for $1 \leq i \leq m - 1$, $(a_i, a_{i+1}) \in E(G)$, and is called a *quasi-path* from a_1 to a_m if for $1 \leq i \leq m - 1$, $(a_i, a_{i+1}) \in E(G)$ or $(a_{i+1}, a_i) \in E(G)$. A sequence of vertices $(a_1, a_2, ..., a_m = a_1)$, in which all vertices except the first vertex are distinct, is called a *cycle* if for $1 \leq i \leq m - 1$, $(a_i, a_{i+1}) \in E(G)$. If all the edges on a path or cycle are directed, then it is called a directed path or cycle. If at least one directed and one undirected edge belongs to a path or cycle,

then it is called partially directed. Vertices which have a directed path from (to) vertex a are called the *descendants* (*ancestors*) of a, denoted by $Des(a)$ ($Anc(a)$). Any vertex is assumed to be an ancestor and descendant of itself. A directed acyclic graph (DAG) is a directed graph with no directed cycles. A chord of a cycle is an edge not in the cycle whose endpoints are in the cycle. A hole in a graph is a cycle of length at least 4 having no chords. A graph is *chordal* if it has no holes. A graph is called a *chain graph* if it contains no directed or partially directed cycles. After removing all directed edges of a chain graph, the components of the remaining undirected graph are called the *chain components* of the chain graph.

2.2 Causal Bayesian Networks

A Bayesian network is a probabilistic graphical model representing statistical independencies among a set of variables via a DAG. This type of graphical model is of particular interest in many applications, such as pattern recognition, epidemiology, and econometrics, due to its power in facilitating efficient statistical inference. A Bayesian network is formally defined as follows:

Definition 1 (Bayesian Network)**.** *Let $G = (V, E)$ be a DAG on a set of random variables $V = \{X_1, ..., X_p\}$, and P_V be the joint distribution of V.*[1] *The pair (G, P_V) is called a Bayesian network if each variable in G is independent of its non-descendants given its parents according to P_V (referred to as local Markov property).*

Based on Definition 1, in a Bayesian network (G, P_V), the joint distribution P_V can be factorized as follows:

$$P_V = \prod_{X_i \in V} P_{X_i|Pa(X_i)},$$

where $Pa(X)$ denotes the set of the parents of variable X in G.

Definition 2 (d-separation)**.** *In a DAG G, a quasi-path is said to be blocked by a subset of vertices X_S, $S \subseteq [p]$, if*

1. *the quasi-path contains an induced subgraph of form $X_a \to X_c \to X_b$ or $X_a \leftarrow X_c \to X_b$ such that X_c is in X_S, or*

2. *the quasi-path contains an induced subgraph of form $X_a \to X_c \leftarrow X_b$ such that X_c is not in X_S and no descendant of X_c is in X_S.*

[1] In the sequel, we will refer to variables and their corresponding vertices in the graph interchangeably.

For any two variables X_i and X_j and a subset of variables X_S, we say X_S d-separates X_i from X_j, denoted by $(X_i$ d-sep $X_j|X_S)$, if X_S blocks every quasi-path from X_i to X_j on G.

Consider Bayesian network (G, P_V). Let $\mathcal{I}(P_V)$ represent the set of all conditional independence relationships in P_V, and $\mathcal{I}(G)$ represent the set of all d-separations in G. By definition, distribution P_V satisfies the local Markov property with respect to G. As shown in [13], this implies that every conditional dependency in P_V is reflected in d-separations in G, referred to as *Global Markov property*. However, there may be conditional independencies in P_V which are not reflected in G. If there is a one-to-one correspondence between the element of $\mathcal{I}(G)$ and $\mathcal{I}(P_V)$, then G is called a perfect I-map for distribution P_V. Therefore, the following extra condition is needed:

Definition 3 (Faithfulness condition)**.** *The distribution P_V is* faithful *to structure G if for any two variables X_i, X_j, and any subset of variables $X_S \subseteq V$, we have*

$$(X_i \text{ d-sep } X_j|X_S) \in \mathcal{I}(G) \text{ if } (X_i \perp X_j|X_S) \in \mathcal{I}(P_V).$$

For the task of learning a Bayesian network representing a given distribution, it is common in the literature to assume the given distribution satisfies Markov and faithfulness conditions with respect to a DAG [14], as in this case, data can be used to learn a DAG reflecting precisely the conditional independencies in the data.

The directed edges in a perfect I-map does not necessarily imply causation. For instance, for a joint distribution P_V on variables $V = \{X_1, X_2, X_3\}$, such that $\mathcal{I}(P_V) = \{(X_1 \perp X_3|X_2)\}$, all three DAGs $G_1 : X_1 \to X_2 \to X_3$, $G_2 : X_1 \leftarrow X_2 \to X_3$, and $G_3 : X_1 \leftarrow X_2 \leftarrow X_3$ are perfect I-maps. Nevertheless, the ubiquity of DAG models in statistical applications stems primarily from their causal interpretation [11]. The goal in the field of *causal structure learning* (also known as *causal discovery*) is to learn a directed graph over the variables in the system, V, in which a directed edge $X_i \to X_j$ implies that X_i is a direct cause of X_j with respect to the set V. We use the language of structural causal models proposed in [11] to formalize this notion.

For a given set of *endogenous* variables $V = \{X_1, ..., X_p\}$, a structural causal model consists of a set of equations of the form

$$X_i = f_i(Pa(X_i), N_i), \qquad 1 \leq i \leq p, \tag{2.1}$$

where $Pa(X_i) \subseteq V \setminus \{X_i\}$ denotes the set of direct causes of variable X_i, and N_i is the *exogenous* variable corresponding to X_i, representing noise or disturbance. The equation in

(2.1) should be understood as a generating mechanism, and sometimes the notation $X_i \leftarrow f_i(Pa(X_i), N_i)$ is used.

Consider the directed graph generated by drawing a directed edge from each element of $Pa(X_i)$ to X_i, for all $i \in [p]$. The resulting directed graph G is called the *causal diagram*. If the causal diagram is acyclic and the exogenous variables are jointly independent, then the model induces a distribution P_V on the endogenous variables that satisfies the local Markov property with respect to G [15]. Therefore, the pair (G, P_V) is a Bayesian network referred to as *causal Bayesian network*. In Chapters 3, 4 and 6, we assume that the causal diagram is always a DAG. We extend our models to allow for cycles in Chapter 5.

2.2.1 Linear Structural Causal Model

In Chapters 4, 5, and 6, we consider a linear structural causal model over p endogenous variables $V = \{X_1, ..., X_p\}$, with Gaussian exogenous variables in Chapters 4 and 5, and with non-Gaussian exogenous variables in Chapter 6. For $i \in [p]$, variable X_i is generated by the following mechanism:

$$X_i = \sum_{j=1}^{p} B_{j,i} X_j + N_i.$$

Non-zero entries $B_{j,i}$ correspond to direct causes of X_i. Let $X \coloneqq [X_1 \cdots X_p]^\top$. The model can be represented in matrix form as

$$X = B^\top X + N, \tag{2.2}$$

where, B is a $p \times p$ weighted adjacency matrix of G, with $B_{j,i}$ as its (j, i)-th entry, and $N = [N_1 \cdots N_p]^\top$. If the underlying structure is a DAG, rows and columns of B can be permuted to make it a strictly upper triangular matrix. Also, if the system is causally sufficient, that is, the exogenous variables do not have latent confounders (common causes), the elements of N are jointly independent. Since we can always center the data, without loss of generality, we assume that N, and hence, X is zero-mean. Hence, for a causally sufficient system with Gaussian exogenous variables, the noise vector N is distributed according to the normal distribution $\mathcal{N}(0, \Omega)$, where Ω is a $p \times p$ diagonal matrix with $\Omega_{i,i} = \sigma_i^2 = Var(N_i)$. Therefore, the system can be fully described by parameters in B and Ω. This model induces a distribution P_V on the endogenous variables. The model in (2.2) could be also represented

as

$$X = A^\top N, \tag{2.3}$$

where $A = (I - B)^{-1}$. This implies that each variable $X_i \in V$ can be written as a linear combination of the exogenous noises in the system.

Note that the linear Gaussian model is one of the most problematic models in the literature of causal discovery, due to the symmetries in this model. In fact, in a structural causal model with additive noise of form $X_i = f(X_j) + N_i$, if the noise variable N_i is non-Gaussian [6], or if the generating function is non-linear (with some mild conditions) [8], the direction of causal influence can be identified from a single observational distribution.

2.2.2 Markov Equivalence

Definition 4. *Directed graphs G_1 and G_2 are independence equivalent (I-equivalent), also known as Markov equivalent, if $\mathcal{I}(G_1) = \mathcal{I}(G_2)$.*

The notion of Markov equivalence is not restricted to acyclic graphs; however, almost all the literature focuses on the case of DAGs. Below is some of the main concepts and results related to Markov equivalent DAGs.

The authors of [16] proposed a graphical test for Markov equivalence among DAGs: Define a v-structure of graph G as a triple of vertices (a, b, c), with induced subgraph $a \to c \leftarrow b$. Markov equivalence can be tested as follows:

Lemma 1 (Verma and Pearl [16])**.** *Two DAGs are Markov equivalent if and only if they have the same skeleton and v-structures.*

For a given DAG G, the *Markov equivalence class* (MEC) of G is defined as

$$\mathit{MEC}(G) = \{G' : G' \textit{ is DAG, and } \mathcal{I}(G') = \mathcal{I}(G)\}.$$

That is, the set of all DAGs, which are Markov equivalent with G. $\mathit{MEC}(G)$ can be uniquely represented by a graph $\tilde{G} = (V(\tilde{G}), E(\tilde{G}))$, called the *essential graph* corresponding to $\mathit{MEC}(G)$, for which $V(\tilde{G}) = V(G)$, and

$$E(\tilde{G}) = \bigcup_{G' \in \mathit{MEC}(G)} E(G').$$

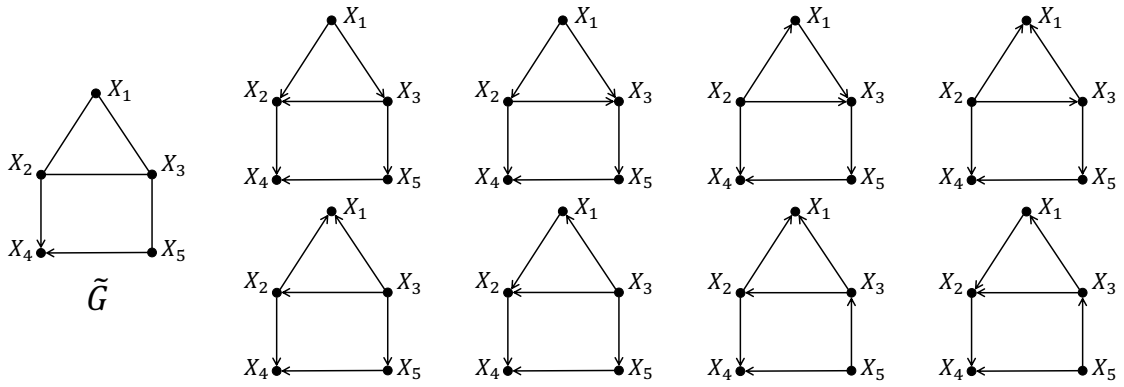

Figure 2.1: Example of the members and the essential graph corresponding to a MEC.

In other words, an essential graph has the same vertices and skeleton as its members of the corresponding MEC, the directed edges are those that have the same direction in all members of the class [12]. See Figure 2.1 for an example of all the elements of a MEC and the essential graph corresponding to the MEC. With a slight abuse of notation, we denote the MEC corresponding to essential graph $\tilde{G}$ by $MEC(\tilde{G})$. Essential graphs are also referred to as completed partially directed acyclic graphs (CPDAGs) [4], and maximally oriented graphs [17]. [12] proposed a graphical criterion for characterizing an essential graph. They showed that an essential graph is a chain graph in which every chain component is chordal. As a corollary of Lemma 1, for an essential graph G, no DAG in $MEC(G)$ can contain a v-structure in the subgraphs corresponding to chain components of G. In order to obtain the essential graph from observational data, one can first learn the skeleton and v-structures of the underlying DAG using conditional independence tests, and then apply the Meek rules [17] to learn the direction of the rest of the directed edges of the essential graph in polynomial time. The Markov and faithfulness assumptions guarantee that the essential graph can be learned from the population dataset.

The authors of [18] observed that the orientation for one chain component does not affect the orientations for other components. Therefore, each chain component can be considered as an essential graph independent of the other components. We call such an essential graph an undirected connected essential graph (UCEG). Note that a UCEG $\tilde{G}$ is chordal and no DAGs in its corresponding equivalence class $MEC(\tilde{G})$ is allowed to have any v-structures. Each DAG in $MEC(\tilde{G})$ has exactly one root variable:

Lemma 2. *Any v-structure-free connected DAG has exactly one root variable.*

See [19] for a proof.

Suppose a joint distribution satisfying Markov and faithfulness conditions to the ground truth causal DAG G^* is given, and we have no latent variables. Without any assumptions on the type of the functions or the distribution of the exogenous variables in the underlying structural causal model in (2.1), the ground truth causal DAG can be identified only up to its Markov equivalence [1, 11]. Hence, the direction of all the edges in the chain components of the essential graph corresponding to $MEC(G^*)$ will remain unresolved. In order to go beyond Markov equivalence and differentiate among the causal structures within a MEC, we focus on two scenarios: Performing interventional experiments, and having access to more than one joint distribution generated from the causal system, which are the focus of Chapters 3 and 4, respectively.

CHAPTER 3

INTERVENTIONAL CAUSAL STRUCTURE LEARNING

Performing interventions is the gold standard for causal structure learning. In interventional causal structure learning, a set of interventions is performed, each on a subset of the variables of the system, and subsequently data is collected from the intervened system. An intervention on a variable X varies the conditional distribution of X given its direct causes. It can also make variable X completely independent of its causes. The information obtained from an intervention depends on the type of the performed intervention, as well as the size of the intervention (i.e., the number of the target variables), and the location of the targets of the intervention in the underlying causal DAG. An interventional experiment is comprised of a sequence of interventions with different target sets. It can be adaptive, in which each intervention in the sequence is designed based on the information obtained from previous interventions, or non-adaptive, in which all the interventions in the sequence are designed before any data is collected. There are two main questions regarding the design of interventional experiments for structure learning:

1. What is the smallest required number of interventions in order to fully learn the underlying causal graph?

2. For a fixed number of interventions (budget), what portion of the causal graph is learnable?

The first problem has been addressed in the literature under different assumptions [20, 21, 18, 22]. Specifically, [20] provided the worst case bounds on the number of required interventions for different types of interventions. The second question mentioned above has received less attention and we address this question herein. We consider a setup in which given a budget k, we design k interventions non-adaptively. The setup that we present here can be interpreted as an extension of the adaptive experiment design, in which interventions are designed in batches of size k; i.e., setting $k = 1$, reduces the setup to the standard adaptive experiment design. Our main contributions are summarized as follows:

- We cast the problem of finding the best intervention target set as an optimization problem which aims to maximize the experiment gain. The gain is defined as the number of edges whose directions are identified due to the performed interventions. We consider the optimization of the worst-case gain, as well as the average gain.

- We start the investigation of the optimization problems by considering the case that the underlying causal structure is a tree. For this case, we present an efficient exact algorithm for the worst-case gain setup, as well as an approximate algorithm for the average gain setup. The latter is based on proving that the objective function for the average gain setup is a monotonically increasing and submodular set function.

- We extend the approximate algorithm to the case of general causal DAGs. In this case, besides the design of interventions, calculating the objective function is also challenging. We propose an efficient exact calculator as well as an unbiased and a fast heuristic estimator for this task. Convergence analysis is provided for the unbiased estimator.

The material in this chapter is taken from [23, 24, 25].

Interventional Structure Learning

We assume that the causal diagram is a DAG and there are no latent confounders in the system. As mentioned in Chapter 2, in general, from a single joint distribution over a set of variables, the ground truth causal structure can be identified up to Markov equivalence. An interventional experiment is the process of perturbing the causal system to generate extra joint distributions over the variables to enable the experimenter to improve the identifiability either merely from the new interventional distributions, or from comparing the original and interventional distributions.

Interventions are generally divided into two types of hard interventions and soft interventions. In a hard intervention on a variable X, all the influences on X are removed and a new value or distribution is forced on X, while in a soft intervention on X, this variable will still be influenced by its original causes after the intervention. Below, we provide a formal definition of an intervention, in which we mainly follow [20].

Consider a causal Bayesian network (G, P_V) on a set of variables $V = \{X_1, ..., X_p\}$ with observational joint distribution P_V. Let X_T be the subset of V that are subject to intervention, called the intervention target set, and for $i \in T$, let W_i be the *intervention variable*

corresponding to X_i. Intervention variables are jointly independent, are not influenced by any of the variables in the system, and for all $i \in T$, W_i directly influences only X_i. A passive observation is considered to be an intervention with empty target set.

Definition 5 (Hard Intervention). *A hard intervention $I = (X_T, W_T)$ on X_T, for all $i \in T$ breaks the causal influence from $Pa(X_i)$ to X_i, i.e., makes X_i independent of $Pa(X_i)$, and sets the intervention variable W_i as the only direct cause of X_i. For all $i \in T$, W_i determines the distribution of X_i, that is, in the factorized joint distribution, replaces the term $P_{X_i|Pa(X_i)}$ with $P^{(I)}_{X_i}$. In the language of structural causal model, $I = (X_T, W_T)$ replaces $X_i = f_i(Pa(X_i), N_i)$ with $X_i = f_i^{(I)}(W_i, N_i)$, for all $i \in T$. Graphically, for all $i \in T$, it removes the directed edges from $Pa(X_i)$ to X_i, and sets the intervention variable W_i as the only parent of X_i to form the interventional graph $G^{(I)}$.*

Intervention I changes the joint distribution of X_T and all variables in the system for which an element of X_T is a direct or indirect cause, and results in an interventional joint distribution $P^{(I)}_V$. The resulting interventional joint distribution can be factorized as follows:

$$P^{(I)}_V = \prod_{X_i \in X_T} P^{(I)}_{X_i} \prod_{X_i \in V \setminus X_T} P_{X_i|Pa(X_i)}.$$

As a specific example of a hard intervention, one can choose W_i to have the same support as the support of X_i, and forces random values of W_i to X_i via $X_i = W_i$. Hard intervention or its variations are also referred to as surgical interventions [11], ideal interventions [1], independent interventions [26], and structural interventions [20] in the literature.

Definition 6 (Soft Intervention). *A soft intervention $I = (X_T, W_T)$ on X_T, for all $i \in T$ adds the intervention variable W_i as an extra direct cause to X_i. For all $i \in T$, W_i directly influences the distribution of X_i, that is, in the factorized joint distribution, replaces the term $P_{X_i|Pa(X_i)}$ with $P^{(I)}_{X_i|Pa(X_i)}$, where $P_{X_i|Pa(X_i)} \neq P^{(I)}_{X_i|Pa(X_i)}$. In the language of structural causal model, $I = (X_T, W_T)$ replaces $X_i = f_i(Pa(X_i), N_i)$ with $X_i = f_i^{(I)}(Pa(X_i), W_i, N_i)$, for all $i \in T$. Graphically, for all $i \in T$, it adds the intervention variable W_i as a parent of X_i to form the interventional graph $G^{(I)}$.*

The resulting interventional joint distribution can be factorized as follows:

$$P^{(I)}_V = \prod_{X_i \in X_T} P^{(I)}_{X_i|Pa(X_i)} \prod_{X_i \in V \setminus X_T} P_{X_i|Pa(X_i)}.$$

Soft intervention or its variations are also referred to as dependent interventions [26], and parametric interventions [20] in the literature.

In his dissertation, Eberhardt provided a more general definition of intervention than what we presented here [20]. Compared to Eberhardt's definition, we do not allow the intervention variables to be confounded by the variables in the system. Also, we do not allow one intervention variable to influence more than one variable of the system, i.e., in our setup simultaneous intervention on two variables require two independent intervention variables.

Neither hard nor soft intervention can be considered as the more general notion of intervention, and either of them can be more practical depending on the application. For instance, in a medical study on the effect of alcohol on blood pressure, if the target variable is the amount of alcohol consumption, it is often feasible to assign a certain value to this variable regardless of other factors which may influence it. However, if the target is the blood pressure, it is not feasible to remove all the other causes of this target variable, yet the value of one of the known causes can be perturbed. In fact, performing a soft intervention is often more challenging [20]. This is due to the fact that any change in the system may lead to removing a subset of the other causes of the target variable.

For an intervention I, the cardinality of the intervention target set, i.e., $|T|$, is referred to as the size of the intervention I. An intervention is called *singleton* if it has size equal to one. We define an *experiment* of size k as a sequence of k interventions $\mathcal{E} = \{I_1, ..., I_k\}$. An experiment is called *adaptive* if in the sequence of interventions, the information obtained from the previous interventions is used to design the next one, otherwise it is called *non-adaptive*, in which the intervention sequence is determined before any interventional data is collected. A non-adaptive experiment gives the experimenter the ability to perform the interventions in parallel without the need to wait for the result of one intervention to choose the next one. For example, in the study of gene regulatory networks (GRNs), when the GRN of all cells are the same, interventions can be performed simultaneously on different cells. Furthermore, as observed in [27], in the worst case, no adaptive experiment design can reduce the number of interventions required for structure learning.

The authors of [28] and [29] extended the notion of Markov equivalence to the interventional case. For an experiment $\mathcal{E}$, DAGs G_1 and G_2 are interventional Markov equivalent if $G_1^{(I)}$ and $G_2^{(I)}$ are Markov equivalent for all $I \in \mathcal{E}$. Based on this notion of equivalence, interventional Markov equivalence class and interventional essential graph are defined similar to the observational case.

The rest of the chapter is organized as follows: After a brief review of related works in Section 3.1, a formal description of the problem setup is presented in Section 3.2. The proposed experiment design approach for tree causal structures and general causal structures are presented in Sections 3.3 and 3.4, respectively. A variation of the general greedy algorithm through lazy evaluations is presented in Section 3.5. Using synthetic and real data, the proposed methods are evaluated in Section 3.6; and finally, our concluding remarks are presented in Section 3.7.

3.1 Related Works

A formal definition and the details of the utilization of interventions for the task of causal discovery is provided by [11] and [1]. Especially, [11] used the concept of atomic intervention, in which the intervened variable is forced to one particular value rather than a non-degenerate distribution, that is, $X_i = x_i$, for some value x_i in the support of random variable X_i. Works including [20, 21, 18, 22] address the problem of finding the smallest number of interventions required for fully identifying the causal structure. [20] provided the worst case bounds on the number of required interventions for different types of interventions. [30] drew connections between causality and known separating system constructions. [21] conjectured regarding the number of interventions with targets of unbounded size sufficient and in the worst case necessary for fully identifying a causal model. The conjecture was proved in [31] where the authors provided an algorithm that finds such a set of interventions in polynomial time. The problem of intervention design with interventions of unbounded size is also addressed in the case that each variable has a certain cost to intervene on [32, 33].

Note that the aforementioned works mostly assume that the cardinality of the interventions could be as large as half of the order of the graph, which may render the applicability of the results infeasible for some applications. [22] considered the problem of learning a causal graph when intervention sizes are bounded by some parameter and provided a lower bound on the number of required interventions for adaptive algorithms. We focus on a setup with singleton interventions, i.e., interventions of size 1. As will be explained in Section 3.2, this setup is suitable for the applications that certain variables cannot be randomized simultaneously, and also maximizes the gain obtained from the performed randomizations. There are other works focused on singleton interventions as well [34, 18, 31]. [34] showed that $N-1$ experiments suffice to determine the causal relations among $N > 2$ variables when each experiment randomizes at most one variable. [18] proposed an adaptive algorithm to

minimize the uncertainty of candidate structures based on the minimax and the maximum entropy criteria. [31] provided a greedy adaptive approach that maximizes the number of orientable edges based on a minimax optimization.

The problem of interventional causal structure learning is also considered in the causally insufficient systems (i.e., with latent confounders) [35]. There also exist works that consider the problem of adaptive intervention design using a Bayesian framework, in which a distribution over possible structures and their associated parameters is maintained [36, 37]. Recently, [38] proposed a method based on optimal Bayesian experimental design in which the expected of a utility function is maximized in each round of experiments according to the current belief and they provided a tractable solution with an approximation guarantee based on sub-modularity.

One less usual connection to the problem of interventional structure learning when we are limited to a budget of k vertices to intervene on, is with the literature concerned with the influence maximization problem. The goal in the influence maximization problem is to find k vertices (seeds) in a given network such that under a specified influence model, the expected number of vertices influenced by the seeds is maximized [39, 40, 41]. Besides the interpretative differences, an important distinction between the two problems is that in the influence maximization problem, the goal is to spread the influence to the vertices of the graph, while in budgeted experiment design problem, the goal is to pick the initial k vertices in a way that leads to discovering the orientation of as many edges as possible. Therefore, the optimal solution to these two problems for a given graph can be quite different (see Appendix A for an example).

3.2 Problem Description

We study the problem of causal structure learning over a set of p endogenous variables $V = \{X_1, ..., X_p\}$, with ground truth causal structure G^* using interventions. We assume that the causal diagram is acyclic, i.e., it is a DAG, and we have the causal sufficiency assumption which implies that there are no latent confounders in the system. Similar to [18], [22], and [32], we consider the case that observational data is available and hence, the interventions can be designed based on the output of an initial passive observational stage. This implies that on the population dataset, we design the interventions with side information about the MEC of the ground truth causal structure.

We consider a setup in which we are given a budget of k interventions, and we design

the interventions with the goal of discovering the direction of as many edges as possible in the causal graph. Interventions are designed non-adaptively, that is, each intervention is performed regardless of the information gained from the other interventions. Note that an adaptive experiment design is a special case of our problem: In an adaptive setup, given the information deduced from the collected data, the next intervention is designed. Therefore, this setup is equivalent to ours when $k = 1$. Equivalently, our setup could be considered as an extension of adaptive experiment design when the interventions are design in batches of size k.

After performing each intervention I_i, data is collected from interventional joint distribution $P_V^{(I_i)}$. Eventually, the observational data and the data gathered from interventions is used for the final output of the procedure. We use the GIES algorithm [28] for this final step.

We assume that all the interventions should be singleton, i.e., each intervention should have size equal to one. This is beneficial since in some applications, the experimenter may not be able to randomize certain variables simultaneously. Note that most of the literature assume that the size of each intervention is larger than one, in some cases going as high as half of the number of variables [27, 21, 31, 32]. Therefore, the set of k variables $\mathcal{I} = \{X_{I_1}, ..., X_{I_k}\}$ contains all the information to describe the targets in the experiment, where X_{I_i} is the single variable intervened on in intervention I_i. We call the set $\mathcal{I}$ the target set of the experiment. We denote the interventional MEC containing DAG G by $\mathcal{I}$-$MEC(G)$. Note that the passive observational experiment is contained in the experiment set, i.e., $\mathcal{I}$-$MEC(G)$ contains all graphs G', such that G' is Markov equivalent to G and $G'^{(I_i)}$ is Markov equivalent to $G^{(I_i)}$, for all singleton interventions I_i, $1 \leq i \leq k$. We have the following assumptions in this work:

Assumption 1. *The ground truth causal structure G^* is a DAG and exogenous variables in the structural causal model are jointly independent.*

Assumption 2. *The observational and interventional joint distributions satisfy Markov and faithfulness conditions with respect to their corresponding observational and interventional DAGs.*

Assumption 3. *The correct essential graph $\tilde{G}^*$ can be learned from the initial observational dataset.*

Under Assumptions 1-3, we have the following result regarding the effect of a singleton intervention.

Lemma 3. *Having the observational essential graph $\tilde{G}^*$, a singleton intervention (hard or soft) on variable X_i identifies the direction of all edges incident with X_i.*

[27] and [18] provided the same result as in Lemma 3 with different proofs. Also, [27] observed that given the essential graph resulted from the passive observational stage, a hard intervention I allows orientating the undirected edge $X_i - X_j$ if only one of X_i and X_j is in the target set of I. If both X_i and X_j are targeted in the intervention, this intervention is called a *zero-information* intervention for the pair $\{X_i, X_j\}$. Our setup in which $|I_i| = 1$, for all $i \in \{1, ..., k\}$, avoids such zero-information experiments. Therefore, another advantage of forcing singleton interventions is that there will be no zero-information interventions in the experiment and hence, we gain the most from each randomization. We note that a zero-information intervention does not happen for the case of soft interventions:

Lemma 4. *A sequence of k singleton soft interventions is equivalent to one soft intervention of size k on the same targets.*

Lemma 4 is a corollary of Theorem 2 of [42]. By Lemma 4, if the performed interventions are soft, they can be done simultaneously as one soft intervention of size k, i.e., we can have $|\mathcal{E}| = 1$, and $|I_1| = k$. Nevertheless, as mentioned earlier, soft interventions in general can be more challenging to perform.

By Assumption 3, we assume that $MEC(G^*)$, and hence, its corresponding essential graph $\tilde{G}^*$ is attainable from the observational data. Let $G_i \in MEC(G^*)$, and for experiment with target set $\mathcal{I}$, denote the interventional Markov equivalence class containing G_i and its corresponding interventional essential graph by $\mathcal{I}$-$MEC(G_i)$ and $\tilde{G}_i^{(\mathcal{I})}$, respectively. Define $R(\mathcal{I}, G_i)$ as the set of edges directed in $\tilde{G}_i^{(\mathcal{I})}$ but not directed in $\tilde{G}^*$, i.e., the set of edges whose directions can be learned due to the experiment with target set $\mathcal{I}$, if the ground truth DAG were G_i. Note that $R(\mathcal{I}, G)$ is the same for all $G \in \mathcal{I}$-$MEC(G_i)$. $R(\mathcal{I}, G_i)$ can be obtained as follows: As seen in Lemma 3, from an experiment with target set $\mathcal{I}$, one learns the direction of all the edges incident with the vertices in $\mathcal{I}$. Denote these directed edges by $A(\mathcal{I}, G_i)$. (Clearly, the orientation of these edges depends on the ground truth DAG G_i, and hence G_i is an input argument.) Meek rules [17] can then be applied to $A(\mathcal{I}, G_i)$ to obtain extra edges oriented in $\tilde{G}_i^{(\mathcal{I})}$ compared to $\tilde{G}^*$ in polynomial time.

Define the *gain* of an experiment with target set $\mathcal{I}$ on ground truth structure G_i as $D(\mathcal{I}, G_i) = |R(\mathcal{I}, G_i)|$, that is, the number of edges whose direction is discovered due to the experiment, if the ground truth DAG were G_i. Since the ground truth DAG is initially known only up to the elements of $MEC(G^*)$, and since there is no preference between the

members of $MEC(G^*)$, G^* is equally likely to be any of the DAGs in the class. Hence, the expected number of the edges recovered through the experiment with target set $\mathcal{I}$ is

$$\mathcal{D}(\mathcal{I}) := \frac{1}{|MEC(G^*)|} \sum_{G_i \in MEC(G^*)} D(\mathcal{I}, G_i). \tag{3.1}$$

We refer to $\mathcal{D}(\mathcal{I})$ as the average gain of the experiment with target set $\mathcal{I}$. Thus, our problem of interest can be formulated as finding intervention target set $\mathcal{I} \subseteq V$ of cardinality k that maximizes $\mathcal{D}(\mathcal{I})$:

$$\max_{\mathcal{I}:\mathcal{I}\subseteq V} \mathcal{D}(\mathcal{I}) \quad \text{s.t.} \quad |\mathcal{I}| = k. \tag{3.2}$$

We refer to (3.2) as the *average gain* optimization problem. Optimization problem (3.2) is challenging for two reasons: First, finding an optimal $\mathcal{I}$ requires a combinatorial search. Second, even for a given set $\mathcal{I}$, computing $\mathcal{D}(\mathcal{I})$ when the value of k or the cardinality of the Markov equivalence class is large, can be computationally intractable. Note that the cardinality of a MEC can be super-exponential in the number of vertices [43].

Alternatively, one can consider a minimax setup, and design the experiment for the worst-case member of the equivalence class:

$$\max_{\mathcal{I}:\mathcal{I}\subseteq V} \min_{G_i \in MEC(G^*)} D(\mathcal{I}, G_i) \quad \text{s.t.} \quad |\mathcal{I}| = k. \tag{3.3}$$

We refer to (3.3) as the *worst-case gain* optimization problem. Optimization problem (3.3) is studied by [31] for the case of $k = 1$. Here, we consider the challenges raised when k is larger than 1 and a brute force search over all subsets of V of size k is not computationally feasible. [18] have also considered a similar setup with singleton interventions with $k = 1$. But their objective functions are different and they perform a brute force search to find the optimum target.

In Section 3.3 we study optimization problems (3.2) and (3.3) for the case that the underlying causal structure is a tree, and we consider the general case in Section 3.4.

3.3 Experiment Design for Tree Structures

We start the investigation of optimization problems (3.2) and (3.3) by considering the case that the underlying causal structure is a tree. For the obtained essential graph from the observational stage, Let $\tilde{T}_1, ..., \tilde{T}_R$ denote the induced subgraphs of the essential graph on

the non-trivial chain components. Note that by definition, each $\tilde{T}_r$ is a UCEG. As mentioned in Chapter 2, orientations for one chain component of an essential graph does not affect the orientations for the other components. Thus, for a given number of interventions assigned to one UCEG, the task of experiment design in that UCEG becomes independent from other UCEGs.

Recall from Lemma 2 that for a given UCEG $\tilde{G}$, each DAG in $MEC(\tilde{G})$ has a unique root variable. Here, since the DAG is a tree and should be v-structure-free, knowing the root variable identifies the orientation of all the edges:

Lemma 5. *For a tree UCEG $\tilde{T}$, no two DAGs in $MEC(\tilde{T})$ have the same root variable, that is, the location of the root variable identifies the direction of all the edges.*

For a tree UCEG $\tilde{T}_r, 1 \leq r \leq R$, and any variable $X \in V(\tilde{T}_r)$, let T_r^X be the unique directed tree in $MEC(\tilde{T}_r)$ with root variable X. Based on Lemmas 2 and 5, $MEC(\tilde{T}_r) = \{T_r^X : X \in V(\tilde{T}_r)\}$. Therefore, optimization problem (3.2) can be written as

$$\max_{\mathcal{I}:\mathcal{I}\subseteq V} \frac{1}{p_u} \sum_{r=1}^{R} \sum_{X\in V(\tilde{T}_r)} D(\mathcal{I}_r, T_r^X), \quad \text{s.t.} \quad \sum_{r=1}^{R} |\mathcal{I}_r| = k, \tag{3.4}$$

where $p_u \coloneqq \sum_{r=1}^{R} |V(\tilde{T}_r)|$, and $\mathcal{I}_r$ is the set of intervened variables in chain component $\tilde{T}_r$, i.e., $\mathcal{I}_r \coloneqq \mathcal{I} \cap V(\tilde{T}_r)$. Furthermore, the optimization problem (3.3) can be written as

$$\begin{aligned} &\max_{\mathcal{I}:\mathcal{I}\subseteq V} \min_{\{X_{i_1},\cdots,X_{i_R}\}\subseteq V} \sum_{r=1}^{R} D(\mathcal{I}_r, T_r^{X_{i_r}}) \quad \text{s.t.} \quad \sum_{r=1}^{R} |\mathcal{I}_r| = k \\ &\equiv \max_{\mathcal{I}:\mathcal{I}\subseteq V} \sum_{r=1}^{R} \min_{X\in V(\tilde{T}_r)} D(\mathcal{I}_r, T_r^X) \quad \text{s.t.} \quad \sum_{r=1}^{R} |\mathcal{I}_r| = k, \end{aligned} \tag{3.5}$$

where the two optimization problems are equivalent due to the fact that orienting edges in one UCEG does not affect orientations of the edges in other UCEGs, and hence, minimization on the root of UCEGs can be done separately.

Let $\{C_1(\mathcal{I}_r), ..., C_{J(\mathcal{I}_r)}(\mathcal{I}_r)\}$ be the set of components of $\tilde{T}_r \setminus \mathcal{I}_r$, i.e., the components resulting from removing vertices $\mathcal{I}_r$ and edges incident to them from $\tilde{T}_r$, where $J(\mathcal{I}_r)$ is the number of the resulted components. We have the following result regarding the calculation of the gain $D(\mathcal{I}_r, T_r^X)$.

Lemma 6. *For any $X \in V(\tilde{T}_r)$ and experiment target set $\mathcal{I}_r \subseteq V(\tilde{T}_r)$, the gain $D(\mathcal{I}_r, T_r^X)$*

can be calculated as follows:

$$D(\mathcal{I}_r, T_r^X) = \begin{cases} |\tilde{T}_r| - 1 & X \in \mathcal{I}_r, \\ |\tilde{T}_r| - |C_j(\mathcal{I}_r)| & X \in C_j(\mathcal{I}_r), \end{cases}$$

where $|G|$ denotes the order (number of vertices) of G.

Using Lemma 6, the average gain of an experiment target set $\mathcal{I}$ can be calculated by the following proposition:

Proposition 1. *The average gain of an experiment target set $\mathcal{I} \subseteq V$ is given as follows:*

$$\mathcal{D}(\mathcal{I}) = \frac{1}{p_u} \sum_{r=1}^{R} |\tilde{T}_r|^2 - \frac{k}{p_u} - \frac{1}{p_u} \sum_{r=1}^{R} \sum_{j=1}^{J(\mathcal{I}_r)} |C_j(\mathcal{I}_r)|^2. \tag{3.6}$$

Based on Lemma 6 and Proposition 1, the optimizer of the optimization problem (3.4) can be found by solving

$$\min_{\mathcal{I}:\mathcal{I}\subseteq V} \sum_{r=1}^{R} \sum_{j=1}^{J(\mathcal{I}_r)} |C_j(I_r)|^2, \quad \text{s.t.} \quad \sum_{r=1}^{R} |\mathcal{I}_r| = k. \tag{3.7}$$

Also, we have

$$\begin{aligned}
&\arg\max_{\mathcal{I}:\mathcal{I}\subseteq V} \sum_{r=1}^{R} \min_{X\in V(\tilde{T}_r)} D(\mathcal{I}, T_r^X), \quad \text{s.t.} \quad \sum_{r=1}^{R} |\mathcal{I}_r| = k \\
&= \arg\max_{\mathcal{I}:\mathcal{I}\subseteq V} \sum_{r=1}^{R} \min_{1\le j\le J(\mathcal{I}_r)} |\tilde{T}_r| - |C_j(\mathcal{I}_r)|, \quad \text{s.t.} \quad \sum_{r=1}^{R} |\mathcal{I}_r| = k \\
&= \arg\max_{\mathcal{I}:\mathcal{I}\subseteq V} \sum_{r=1}^{R} |\tilde{T}_r| - \max_{1\le j\le J(\mathcal{I}_r)} |C_j(\mathcal{I}_r)|, \quad \text{s.t.} \quad \sum_{r=1}^{R} |\mathcal{I}_r| = k \\
&= \arg\max_{\mathcal{I}:\mathcal{I}\subseteq V} \sum_{r=1}^{R} - \max_{1\le j\le J(\mathcal{I}_r)} |C_j(\mathcal{I}_r)|, \quad \text{s.t.} \quad \sum_{r=1}^{R} |\mathcal{I}_r| = k \\
&= \arg\min_{\mathcal{I}:\mathcal{I}\subseteq V} \sum_{r=1}^{R} \max_{1\le j\le J(\mathcal{I}_r)} |C_j(\mathcal{I}_r)|, \quad \text{s.t.} \quad \sum_{r=1}^{R} |\mathcal{I}_r| = k.
\end{aligned}$$

Hence, the optimizer of the optimization problem (3.5) can be found by solving

$$\min_{\mathcal{I}:\mathcal{I}\subseteq V} \sum_{r=1}^{R} \max_{1\leq j\leq J(\mathcal{I}_r)} |C_j(\mathcal{I}_r)|, \quad \text{s.t.} \quad \sum_{r=1}^{R} |\mathcal{I}_r| = k. \tag{3.8}$$

Clearly, the optimization problems in (3.7) and (3.8) can be solved via a brute-force search over all $\binom{p}{k}$ target sets, which can be computationally intensive. In Subsections 3.3.1 and 3.3.2, we will introduce efficient algorithms to address these optimization problems.

3.3.1 Optimizing the Worst-Case Gain in Tree Structures

We start with the optimization problem in (3.8). As mentioned before, for a fixed number of intervention in UCEG $\tilde{T}_r$, the task of experiment design in that UCEG becomes independent of other UCEGs. Thus, we can formulate the optimization problem in (3.8) as follows:

$$\begin{aligned}
&\min_{(\mathcal{I}_1,\ldots,\mathcal{I}_R):\sum_{r=1}^{R}|\mathcal{I}_r|=k} \sum_{r=1}^{R} \max_{1\leq j\leq J(\mathcal{I}_r)} |C_j(\mathcal{I}_r)| \\
&\equiv \min_{(\mathcal{I}_1,\ldots,\mathcal{I}_R):|\mathcal{I}_r|=k_r,\sum_{r=1}^{R}k_r=k} \sum_{r=1}^{R} \max_{1\leq j\leq J(\mathcal{I}_r)} |C_j(\mathcal{I}_r)| \\
&\equiv \min_{(k_1,\ldots,k_R):\sum_{r=1}^{R}k_r=k} \sum_{r=1}^{R} \min_{\mathcal{I}_r:|\mathcal{I}_r|=k_r} \max_{1\leq j\leq J(\mathcal{I}_r)} |C_j(\mathcal{I}_r)|.
\end{aligned} \tag{3.9}$$

Herein, we first propose Algorithm 1 that solves for the minimax problem in the summation in expression (3.9) for each given UCEG $\tilde{T}_r$. That is, Algorithm 1 finds a set $\mathcal{I}_r$ in $\tilde{T}_r$ of size k_r such that after removing the variables in $\mathcal{I}_r$, the maximum size of the remaining components is minimized. Next, we will show that how Algorithm 1 can be utilized to obtain an optimum solution of the problem in (3.9).

Algorithm 1 takes a UCEG $\tilde{T}_r$ and budget of intervention k_r as inputs and returns the set $\hat{\mathcal{I}}_r$ that is a solution of the following minimax problem:

$$\min_{\mathcal{I}_r:\mathcal{I}_r\subseteq V(\tilde{T}_r)} \max_{1\leq j\leq J(\mathcal{I}_r)} |C_j(\mathcal{I}_r)|, \quad \text{s.t.} \quad |\mathcal{I}_r| = k_r. \tag{3.10}$$

In the main loop of Algorithm 1, each variable $X_i \in V(\tilde{T}_r)$ is set as the starting point for performing Depth-First Search (DFS) on $\tilde{T}_r$. For a given threshold value mid, $1 \leq mid \leq |\tilde{T}_r|$, the algorithm does the following. On the traversal of DFS, whenever all the descendants

Algorithm 1 Minimax Experiment Design for a UCEG

1: **input:** $\tilde{T}_r$, k_r.
2: **for** $X_i \in V(\tilde{T}_r)$ **do**
3: $L = 1$, $H = |\tilde{T}_r|$, $T = \tilde{T}_r$
4: **while** $\lfloor H \rfloor \neq \lfloor L \rfloor$ **do**
5: $\mathcal{I} = \emptyset$, $mid = (L + H)/2$
6: Perform DFS on T starting from X_i.
7: **for** $X_j \in V$, when all variables in $Desc(X_j)$ w.r.t. $T_r^{X_i}$ are visited in DFS traversal, **do**
8: **if** $|Desc(X_j)| > mid$ **then**
9: $\mathcal{I} = \mathcal{I} \cup \{X_j\}$
10: $T = T \setminus Desc(X_j)$
11: **end if**
12: **end for**
13: **if** $|\mathcal{I}| \leq k_r$ **then**
14: $mid(X_i) = mid$, $\mathcal{I}(X_i) = \mathcal{I}$
15: $H = mid$
16: **else**
17: $L = mid$
18: **end if**
19: **end while**
20: **end for**
21: $\hat{\mathcal{I}}_r = \mathcal{I}(\arg\min_{X_i} mid(X_i))$
22: **output:** $\hat{\mathcal{I}}_r$

of a variable X_j are visited, it decides to remove X_j and adds it to the set $\mathcal{I}$ (which is the set of variables on which we will intervene), if not doing so results in having a component with size larger than mid in the subtree rooted at X_j (lines 8-9). Note that after removing X_j, for the rest of variables in the traversal, we do not consider the disconnected vertices anymore. After checking all the variables in DFS, we see if our budget of intervention, i.e., k_r, is enough for performing $|\mathcal{I}|$ interventions (line 13). We update the value for mid in each loop using a binary search to find the minimum threshold that can be satisfied by the budget. More specifically, if the number of interventions is less than the budget k_r for a value of mid, we narrow down our search space to $[L, mid]$ (lines 13-15). Otherwise, we consider the region $[mid, H]$ (line 17). This procedure will be repeated for all possible choices of the starting point of DFS and we choose the best $\mathcal{I}(X)$ as the output of the algorithm (line 21).

Theorem 1. *Algorithm 1 returns the optimal solution of the optimization problem in* (3.10).

Establishing an algorithm for solving the minimax problem in (3.10), we can utilize it to

solve the main optimization problem in (3.9). To this end, we show that the main problem can be formulated as a multi-choice knapsack problem [44], and hence, it can be solved efficiently by existing algorithms [44] proposed for the multi-choice knapsack problem.

In order to find an optimal solution of (3.9), using Algorithm 1, we first obtain the optimal value of objective function in (3.10) for every UCEG $\tilde{T}_r$ and any assigned budget $k_r = j$, where $0 \leq j \leq k$, and denote the optimum value by $D_{r,j}$. Also, for each UCEG $\tilde{T}_r$ and budget j, we define binary indicator variable $x_{r,j}$, where $x_{r,j} = 1$ if the budget assigned to $\tilde{T}_r$ is equal to j, otherwise, $x_{r,j} = 0$. Hence, optimization problem (3.9) can be reformulated as follows:

$$\begin{aligned} \min & \sum_{r=1}^{R} \sum_{j=0}^{k} D_{r,j} x_{r,j} \\ \text{s.t.} \quad & \sum_{r=1}^{R} \sum_{j=0}^{k} j x_{r,j} \leq k, \\ & \sum_{j=0}^{k} x_{r,j} = 1, \\ & x_{r,j} \in \{0,1\}, \quad \text{for all } 1 \leq r \leq R, \text{for all } 0 \leq j \leq k. \end{aligned} \tag{3.11}$$

The first condition ensures that the total number of interventions performed in all UCEGs is less than or equal to budget k and the second condition specifies the number of interventions assigned to each UCEG $\tilde{T}_r$. Moreover, the sum $\sum_{j=0}^{k} D_{r,j} x_{r,j}$ in the objective function is equal to $D_{r,j}$ if $x_{r,j} = 1$. In other words, this sum is equal to the optimal value of objective function in (3.10) if $k_r = j$. Thus, the objective function in (3.11) is equal to the one in (3.9).

Regarding the time complexity of the proposed approach, we first run Algorithm 1 on each UCEG for any budget in the range $\{0, ..., k\}$. The time complexity of Algorithm 1 is in the order of $\mathcal{O}(p^2 \log p)$. This is due to the fact that DFS runs in time $\mathcal{O}(p)$ for a tree of order p and for a fixed value of parameter H, the while loop in Algorithm 1 will run for $\log_2(H)$ times, which can be at most $\log p$. Therefore, the time complexity of obtaining the optimal value of objective function in (3.10) for all $1 \leq r \leq R$ and $1 \leq k_r \leq k$, is in the order of $\mathcal{O}(p^3 k \log p)$. Moreover, the time complexity of solving the multi-choice knapsack problem is in the order of $\mathcal{O}(pk^2)$. Hence, the total time complexity of the proposed approach would be in the order of $\mathcal{O}(p^3 k \log p)$.

3.3.2 Optimizing the Average Gain in Tree Structures

We now move to the problem of experiment design on tree structures for maximizing the average gain presented in expression (3.6). Unlike the minimax case, in the case of maximizing the average gain, the objective function depends on both the maximum order of the components, as well as how uniform the order of the components are. This fact makes the design of the experiment target set more challenging in the average case. Unfortunately, we do not have an efficient exact algorithm for this case; however, we show that due to submodularity of the objective function, an efficient approximation algorithm for this case can be obtained. We start by reviewing monotonicity and submodularity properties for a set function.

Definition 7. *A set function $f : 2^V \to \mathbb{R}$ is monotonically increasing if for all sets $\mathcal{I}_1 \subseteq \mathcal{I}_2 \subseteq V$, we have*

$$f(\mathcal{I}_1) \leq f(\mathcal{I}_2).$$

Definition 8. *A set function $f : 2^V \to \mathbb{R}$ is submodular if for all subsets $\mathcal{I}_1 \subseteq \mathcal{I}_2 \subseteq V$ and all $X \in V \setminus \mathcal{I}_2$,*[1]

$$f(\mathcal{I}_1 \cup \{X\}) - f(\mathcal{I}_1) \geq f(\mathcal{I}_2 \cup \{X\}) - f(\mathcal{I}_2).$$

[45] showed that if f is a submodular and monotonically increasing set function with $f(\emptyset) = 0$, then the set $\hat{\mathcal{I}}$ with $|\hat{\mathcal{I}}| = k$ found by a greedy algorithm satisfies

$$f(\hat{\mathcal{I}}) \geq (1 - \frac{1}{e}) \max_{\mathcal{I}:|\mathcal{I}|=k} f(\mathcal{I}),$$

that is, the greedy algorithm is a $(1-\frac{1}{e})$-approximation algorithm. In the following, we show that the set function $\mathcal{D}$ defined in (3.6) is monotonically increasing and submodular, and hence, since $\mathcal{D}(\emptyset) = 0$, the greedy algorithm is a $(1 - \frac{1}{e})$-approximation algorithm for the maximization problem (3.2).

Proposition 2. *For tree structures, the set function $\mathcal{D}$ defined in* (3.6) *is monotonically increasing and submodular.*

Our general greedy algorithm is presented in Algorithm 2. We define the *marginal gain* of variable X when the previous chosen set is $\mathcal{I}$ as

$$\Delta_X(\mathcal{I}) = \mathcal{D}(\mathcal{I} \cup \{X\}) - \mathcal{D}(\mathcal{I}). \tag{3.12}$$

[1] If f is monotonically increasing, $X \in V \setminus \mathcal{I}_2$ relaxes to $X \in V$.

Algorithm 2 General Greedy Algorithm

input: Essential graph from the observational stage, budget k.
initialize: $\mathcal{I}_0 = \emptyset$
for $i = 1$ to k **do**
  $X_i = \arg\max_{X \in V \setminus \mathcal{I}_{i-1}} \mathcal{D}(\mathcal{I}_{i-1} \cup \{X\}) - \mathcal{D}(\mathcal{I}_{i-1})$
  $\mathcal{I}_i = \mathcal{I}_{i-1} \cup \{X_i\}$
end for
output: $\hat{\mathcal{I}} = \mathcal{I}_k$

The greedy algorithm iteratively adds a variable which has the largest marginal gain to the target set until it runs out of budget. For any input set $\mathcal{I}$, in order to calculate the value of $\mathcal{D}(\mathcal{I})$, we use the equation in (3.6). Note that $\mathcal{D}(\mathcal{I})$ can be computed efficiently from (3.6) as it is just needed to obtain the size of resulted components after removing variables in $\mathcal{I}$. To do so, we can run DFS algorithm on each component. In each DFS call, the size of a component is obtained by visiting the variables in it. Then, we will call DFS on the next unvisited component until there is no unvisited variable in the essential graph. Therefore, $\mathcal{D}(\mathcal{I})$ can be computed in $\mathcal{O}(p)$ since the total number of edges in all components is in the order of $\mathcal{O}(p)$.

3.4 Experiment Design for General Structures

In this section we consider experiment design for the case of general structures, formulated in optimization problem (3.2). We first generalize Proposition 2 by showing that the function $\mathcal{D}$ defined in (3.1) is monotonically increasing and submodular.

Proposition 3. *The set function $\mathcal{D}$ defined in* (3.1) *is monotonically increasing.*

We use the following lemma in the proof of submodularity of the function $\mathcal{D}$.

Lemma 7. *Let $\mathcal{I}_1$ and $\mathcal{I}_2$ be arbitrary subsets of variables of a DAG G. We have*

$$R(\mathcal{I}_1 \cup \mathcal{I}_2, G) = R(\mathcal{I}_1, G) \cup R(\mathcal{I}_2, G).$$

As mentioned in Section 3.2, from an experiment with target set $\mathcal{I}$, one learns the direction of all the edges incident with the vertices in $\mathcal{I}$, denoted by $A(\mathcal{I}, G)$, and then the extra edges in the interventional essential graph can be obtained by, say, using the Meek rules starting from $A(\mathcal{I}, G)$. Lemma 7 implies that the set of resolved edges in the essential graph starting

from $A(\mathcal{I}_1 \cup \mathcal{I}_2, G)$ is the same as the set of edges whose direction is resolved either in the essential graph starting from $A(\mathcal{I}_1, G)$ or in the essential graph starting from $A(\mathcal{I}_2, G)$.

Theorem 2. *The set function $\mathcal{D}$ defined in* (3.1) *is a submodular function.*

Equipped with Proposition 3 and Theorem 2, we can again use Algorithm 2, to obtain an $(1 - \frac{1}{e})$-approximation of the optimal solution of optimization problem (3.2). However, as mentioned in Section 3.2, another challenge regarding solving the optimization problem (3.2) is the computational aspect of calculating $\mathcal{D}(\mathcal{I})$ for a given experiment target set $\mathcal{I}$. In Section 3.3, for the case of tree structures, for a given set $\mathcal{I}$, we calculated the value of $\mathcal{D}(\mathcal{I})$ efficiently by applying DFS algorithm; yet this approach cannot be extended to the case of general structures. In the following subsections, we propose efficient methods for exact calculation and estimation of $\mathcal{D}(\mathcal{I})$ for general structures.

Remark 1. *As seen in the proof of Theorem 2, for DAG G and $\mathcal{I} \subseteq V(G)$, the set function $D(\mathcal{I}, G)$ is submodular. However, the minimum of submodular functions is not necessarily submodular. Hence, Algorithm 2, is not necessarily a $(1 - \frac{1}{e})$-approximation algorithm for the case of worst-case gain in optimization problem* (3.3).

3.4.1 Exact Calculation of $\mathcal{D}(\mathcal{I})$

In this section, we show that a method for counting the number of elements in a MEC can be used for calculating $\mathcal{D}(\mathcal{I})$. For an essential graph $\tilde{G}$, we define the size of its corresponding MEC as the number of DAGs in the class and denote it by $Size(\tilde{G})$. Let $\{\tilde{G}_1, ..., \tilde{G}_R\}$ be the chain components of $\tilde{G}$. $Size(\tilde{G})$ can be calculated from the size of chain components using the following equation [46, 18]:

$$Size(\tilde{G}) = \prod_{r=1}^{R} Size(\tilde{G}_r). \tag{3.13}$$

Therefore, it suffices to calculate the size of UCEGs $\tilde{G}_1, ..., \tilde{G}_R$.

Definition 9. *Let $\tilde{G}_r$ be a UCEG. The X-rooted subclass of MEC($\tilde{G}_r$) is the set of all X-rooted DAGs in MEC($\tilde{G}_r$). This subclass can be represented by the X-rooted graph $\tilde{G}_r^X = (V(\tilde{G}_r^X), E(\tilde{G}_r^X))$, called the X-rooted essential graph, where $V(\tilde{G}_r^X) = V(\tilde{G}_r)$, and $E(\tilde{G}_r^X) = \bigcup\{E(G) : G \in X$-rooted subclass of MEC($\tilde{G}_r$)$\}$.*

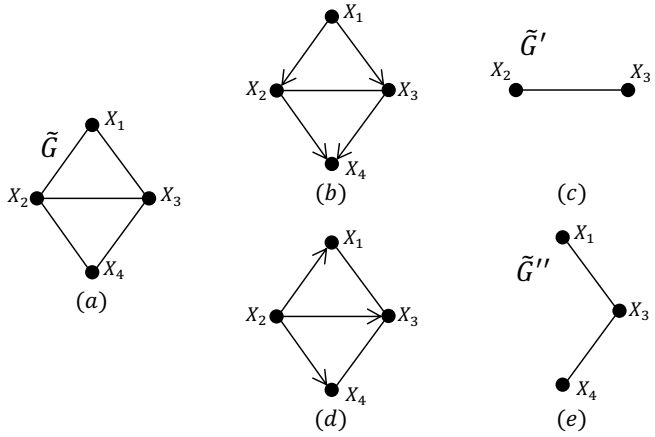

Figure 3.1: Example of the counting and sampling approach.

For instance, for UCEG $\tilde{G}$ in Figure 3.1(a), $\tilde{G}^{X_1}$ and $\tilde{G}^{X_2}$ are depicted in Figures 3.1(b) and 3.1(d), respectively.

Lemma 8 (He et al. [43])**.** *Let $\tilde{G}_r$ be a UCEG. For any $X \in V(\tilde{G}_r)$, the X-rooted subclass is not empty and the set of all X-rooted subclasses partitions MEC($\tilde{G}_r$).*

From Lemma 8 we have

$$Size(\tilde{G}_r) = \sum_{X \in V(\tilde{G}_r)} Size(\tilde{G}_r^X). \tag{3.14}$$

Therefore, using equations (3.13) and (3.14), we have

$$Size(\tilde{G}) = \prod_{r=1}^{R} \sum_{X \in V(\tilde{G}_r)} Size(\tilde{G}_r^X). \tag{3.15}$$

$\tilde{G}_r^X$ could be viewed as an essential graph, as it can be considered as an interventional essential graph with target variable X, for which in the underlying DAG all of edges incident to X are outgoing edges. Hence, the number of DAGs in its corresponding X-rooted subclass can be calculated via equation (3.13). Therefore, using equation (3.15), $Size(\tilde{G})$ can be obtained recursively: In each chain component, each variable is set as the root variable, and in each chain component of each of the resulting rooted essential graphs, each variable is set as the root, and this procedure is repeated until the resulting essential graph is a directed graph and has no chain components.

Note that in this procedure, after setting each variable as the root, we observe the directions that the edges in the rooted essential graph acquire. That is, it has the property that

we explicitly monitor the performed orientations in the given essential graph. The approach that we present in the following for calculating and estimating the value of $\mathcal{D}(\mathcal{I})$ requires this property. Therefore, methods for calculation of the size of the MEC which are based on explicit functions of the parameters of the structure cannot be used in our approach. For instance, [43] showed that there are five types of MECs whose sizes can be formulated as functions of the number of vertices; e.g., for a tree UCEG of order p, the size of the MEC is $p-1$. We have proposed an efficient counting approach with our desired property of monitoring the performed orientations in [24], where the counting is performed based on the clique tree representation of the essential graph.

Example 1. *Assume the UCEG in Figure 3.1(a) is the given essential graph.*

Setting vertex X_1 as the root of $\tilde{G}$ (by symmetry, X_4 is similar), in the rooted essential graph $\tilde{G}^{X_1}$, the directed edges are $X_1 \to X_2$, $X_1 \to X_3$, $X_2 \to X_4$, and $X_3 \to X_4$. This rooted essential graph is shown in Figure 3.1(b), which has a single chain component $\tilde{G}'$, (Figure 3.1(c)). Setting vertex X_2 as the root of $\tilde{G}'$ (by symmetry, X_3 is similar), in the rooted essential graph $\tilde{G}'^{X_2}$, the directed edge is $X_2 \to X_3$. This results in a directed graph, thus, $Size(\tilde{G}'^{X_2}) = 1$. Similarly, $Size(\tilde{G}'^{X_3}) = 1$. Therefore, using equation (3.14), we have $Size(\tilde{G}^{X_1}) = Size(\tilde{G}'^{X_2}) + Size(\tilde{G}'^{X_3}) = 2$. Similarly, we have $Size(\tilde{G}^{X_4}) = 2$.

Setting vertex X_2 as the root of $\tilde{G}$ (by symmetry, X_3 is similar), in the rooted essential graph $\tilde{G}^{X_2}$, the directed edges are $X_2 \to X_1$, $X_2 \to X_3$, and $X_2 \to X_4$. This rooted essential graph is shown in Figure 3.1(d), which has a single chain component $\tilde{G}''$, (Figure 3.1(e)). Setting vertex X_1 as the root of $\tilde{G}''$, in the rooted essential graph $\tilde{G}''^{X_1}$, the directed edges are $X_1 \to X_3$ and $X_3 \to X_4$. This results in a directed graph, thus, $Size(\tilde{G}''^{X_1}) = 1$. Similarly, $Size(\tilde{G}''^{X_3}) = 1$ and $Size(\tilde{G}''^{X_4}) = 1$. Therefore, using equation (3.14), we have $Size(\tilde{G}^{X_2}) = Size(\tilde{G}''^{X_1}) + Size(\tilde{G}''^{X_3}) + Size(\tilde{G}''^{X_4}) = 3$. Similarly, we have $Size(\tilde{G}^{X_3}) = 3$.

Finally, using equation (3.14), we obtain that $Size(\tilde{G}) = \sum_i Size(\tilde{G}^{X_i}) = 10$.

Now consider the task of counting the number of elements of a $MEC(\tilde{G})$ in the presence of prior knowledge regarding the direction of a subset of the undirected edges of the essential graph. We present the available prior knowledge in the form of a hypothesis graph $H = (V(H), E(H))$, which is the same as $\tilde{G}$, yet the orientation of the edges corresponding to the prior knowledge are determined as well. For essential graph $\tilde{G}$, let $Size_H(\tilde{G})$ denote the number of the elements of $MEC(\tilde{G})$, which are consistent with hypothesis H, i.e., $Size_H(\tilde{G}) = |\{G : G \in MEC(\tilde{G}), E(G) \subseteq E(H)\}|$. Similar to equation (3.13), we have $Size_H(\tilde{G}) = \prod_{r=1}^{R} Size_H(\tilde{G}_r)$. Also, akin to equation (3.14), for chain component of $\tilde{G}$, we

Algorithm 3 Counting with Prior Knowledge

input Essential graph $\tilde{G}$, Hypothesis graph H.
output $\textsc{Counter}(\tilde{G}, H)$

---

function $\textsc{Counter}(\tilde{G}, H)$:
if $\tilde{G}$ is a directed graph **then return** 1.
else
  for each chain component $\tilde{G}_r$ of $\tilde{G}$ **do**
    for $X \in V(\tilde{G}_r)$ **do**
      if $E(\tilde{G}_r^X) \subseteq E(H)$ **then** $Size(\tilde{G}_r^X) = \textsc{Counter}(\tilde{G}_r^X, H)$ **else** $Size(\tilde{G}_r^X) = 0$ **end if**
    end for
    $Size(\tilde{G}_r) = \sum_X Size(\tilde{G}_r^X)$
  end forreturn $\prod_r Size(\tilde{G}_r)$
end if

---

have $Size_H(\tilde{G}_r) = \sum_{X \in V(\tilde{G}_r)} Size_H(\tilde{G}_r^X)$. Therefore, in order to extend the counting approach to the case of having prior knowledge, every time that a variable is chosen as the root of a UCEG, we check if the resulting oriented edges belong to $E(H)$. If this is not the case, for X-rooted essential graph $\tilde{G}^X$, we return $Size(\tilde{G}^X) = 0$. This guarantees that any DAG considered in the counting will be consistent with the hypothesis H. See Algorithm 3 for a pseudo-code of the proposed counting approach with prior knowledge. If $H = \tilde{G}$ it implies that we have no prior knowledge, and the algorithm outputs $Size(\tilde{G})$. Note that the ability of checking the consistency of the oriented edges with the hypothesis is the reason that we stated earlier that the property of monitoring the performed orientations in the given essential graph is required in our approach.

Proposition 4. *For a given essential graph with maximum vertex degree Δ, the computational complexity of Algorithm 3 is $\mathcal{O}(p^{\Delta+2})$.*

We now demonstrate how the approach of counting with prior knowledge can be utilized for the task of calculating $D(\mathcal{I})$. Recall that for an experiment target set $\mathcal{I}$ and DAG $G_i \in MEC(G^*)$, the set $R(\mathcal{I}, G_i)$, i.e., the set of edges directed in $\tilde{G}_i^{(\mathcal{I})}$ but not directed in $\tilde{G}^*$, only depends on the $\mathcal{I}$-MEC that G_i belongs to. Also, recall that the $\mathcal{I}$-MEC that G_i belongs to only depends on $A(\mathcal{I}, G_i)$, which is the directed edges in G_i incident to vertices in $\mathcal{I}$. Therefore, all DAGs $G \in MEC(G^*)$ that have the same set $A(\mathcal{I}, G)$ lead to the same value for $D(\mathcal{I}, G)$. Therefore, one can partition the members of $MEC(G^*)$ with respect to their set $A(\mathcal{I}, G)$, and then, consider the set $A(\mathcal{I}, G)$ as prior knowledge and use the aforementioned

counting approach to count the number of DAGs in each partition of $MEC(G^*)$.

Formally, let $\mathcal{H}$ be the set of hypothesis graphs, in which each element H has a distinct configuration for $A(\mathcal{I}, G)$. If the maximum degree of the graph is Δ, cardinality of $\mathcal{H}$ is at most $2^{k\Delta}$, and hence, it does not grow with p. For a given hypothesis graph H, let $\tilde{G}_H = \{G : G \in MEC(G^*), E(G) \subseteq E(H)\}$ denote the set of members of the $MEC(G^*)$, which are consistent with hypothesis H. Note that this set is in fact an interventional MEC. Using the set $\mathcal{H}$, we can write the expression of $\mathcal{D}(\mathcal{I})$ as follows.

$$\begin{aligned}\mathcal{D}(\mathcal{I}) &= \frac{1}{Size(\tilde{G}^*)} \sum_{G_i \in MEC(G^*)} D(\mathcal{I}, G_i) \\ &= \frac{1}{Size(\tilde{G}^*)} \sum_{H \in \mathcal{H}} \sum_{G_i \in \tilde{G}_H} D(\mathcal{I}, G_i) \\ &= \sum_{H \in \mathcal{H}} \frac{Size_H(\tilde{G}^*)}{Size(\tilde{G}^*)} D(\mathcal{I}, G_i),\end{aligned} \tag{3.16}$$

where in the last summation, $G_i \in \tilde{G}_H$. Therefore, we only need to calculate at most $2^{k\Delta}$ values instead of considering all elements of $MEC(G^*)$, which reduces the complexity from super-exponential to constant in p.

Eventually, in order to design the experiment, we use the proposed calculator of $\mathcal{D}$ in a greedy algorithm. We term this approach the *Greedy Intervention Design* (GrID).

3.4.2 Unbiased $\mathcal{D}(\mathcal{I})$ Estimator

The computational complexity of the approach presented in Subsection 3.4.1 for exact calculation of $\mathcal{D}(\mathcal{I})$ is exponential in the intervention budget k. Hence, it may not be computationally tractable for large values of k. For this scenario, we propose running Monte Carlo simulations of the intervention model for sufficiently large number of times to obtain an accurate estimation of $\mathcal{D}(\mathcal{I})$. To this end, we need a uniform sampler for generating random DAGs from $MEC(G^*)$. We present such a sampler, which is based on the counting method presented in Subsection 3.4.1. The main idea is that in a UCEG, we choose a vertex as the root according to the portion of members of the corresponding MEC which have that vertex as the root, i.e., in UCEG $\tilde{G}$, vertex X should be picked as the root with probability $Size(\tilde{G}^X)/Size(\tilde{G})$. The pseudo-code of the proposed sampler is presented in function UnifSamp in Algorithm 4, in which we use function Counter from Algorithm 3.

Algorithm 4 Unbiased $\mathcal{D}(\mathcal{I})$ Estimator

input: Essential graph $\tilde{G}$ with chain components $\{\tilde{G}_1, ..., \tilde{G}_R\}$, target set $\mathcal{I}$, and N.
initialize: $\widehat{MEC} = \emptyset$
for $i = 1$ to N **do**
  Generate sample DAG $G_i = \textsc{UnifSamp}(\tilde{G})$
  $\widehat{MEC} = \widehat{MEC} \uplus G_i$
end for
output: $\hat{\mathcal{D}}(\mathcal{I}) = \frac{1}{N} \sum_{G_i \in \widehat{MEC}} D(\mathcal{I}, G_i)$

---

function $\textsc{UnifSamp}(\tilde{G})$
initialize: $\mathcal{G} = \{\tilde{G}_1, ..., \tilde{G}_R\}$
while $\mathcal{G} \neq \emptyset$ **do**
  Pick an element $\tilde{G}_r \in \mathcal{G}$, and update $\mathcal{G} = \mathcal{G} \setminus \tilde{G}_r$.
  Set $X \in V(\tilde{G}_r)$ as the root with probability $\frac{\textsc{Counter}(\tilde{G}_r^X, \tilde{G}_r^X)}{\textsc{Counter}(\tilde{G}_r, \tilde{G}_r)}$.
  Add the directed edges of $\tilde{G}_r^X$ to $\tilde{G}$
  $\mathcal{G} = \mathcal{G} \cup \{\text{chain components of } \tilde{G}_r^X\}$
end whilereturn $\tilde{G}$.

Example 2. *For the UCEG in Figure 3.1(a), as observed in Example 1,* $Size(\tilde{G}^{X_1}) = Size(\tilde{G}^{X_4}) = 2$, $Size(\tilde{G}^{X_2}) = Size(\tilde{G}^{X_3}) = 3$, *and hence,* $Size(\tilde{G}) = 10$. *Therefore, we set vertices* X_1, X_2, X_3, *and* X_4 *as the root with probabilities* 2/10, 3/10, 3/10, *and* 2/10, *respectively. Suppose* X_2 *is chosen as the root. Then as seen in Example 1,* $Size(G''^{X_1}) = Size(G''^{X_3}) = Size(G''^{X_4}) = 1$. *Therefore, in* G'', *we set either of the vertices as the root with equal probability to obtain the final DAG.*

Theorem 3. *The sampler in Algorithm 4 is uniform.*

As a corollary of Proposition 4, for bounded degree graphs, the proposed sampler runs in polynomial time.

Corollary 1. *For a given essential graph with maximum vertex degree* Δ, *the computational complexity of the uniform sampler in Algorithm 4 is* $\mathcal{O}(p^{\Delta+2})$.

Equipped with the uniform sampler in Algorithm 4, in order to estimate the value of $\mathcal{D}(\mathcal{I})$, we generate N DAGs from $MEC(G^*)$. The generated DAGs are kept in a multiset $\widehat{MEC}$, in which repetition is allowed. Finally, we calculate the estimated value $\hat{\mathcal{D}}(\mathcal{I})$ on $\widehat{MEC}$ instead of $MEC(G^*)$ as follows.

$$\hat{\mathcal{D}}(\mathcal{I}) = \frac{1}{|\widehat{MEC}|} \sum_{G_i \in \widehat{MEC}} D(\mathcal{I}, G_i).$$

The pseudo-code of our estimator is presented in Algorithm 4. In the pseudo-code, operator $\uplus$ indicates the multiset addition.

The estimation obtained from the aforementioned approach is an unbiased estimation of $\mathcal{D}(\mathcal{I})$, i.e., $\mathbb{E}[\hat{\mathcal{D}}(\mathcal{I})] = \mathcal{D}(\mathcal{I})$. To show the unbiasedness, suppose G_i is a random generated DAG in the uniform sampler. We have

$$\begin{aligned}\mathbb{E}[\hat{\mathcal{D}}(\mathcal{I})] &= \frac{1}{N}\sum_{G_i \in \widehat{MEC}} \mathbb{E}[D(\mathcal{I}, G_i)] \\ &= \frac{1}{N}\cdot N \sum_{G_i' \in MEC(G^*)} P(G_i = G_i')D(\mathcal{I}, G_i') \\ &= \frac{1}{|MEC(G^*)|}\sum_{G_i' \in MEC(G^*)} D(\mathcal{I}, G_i') = \mathcal{D}(\mathcal{I}).\end{aligned}$$

Eventually, in order to design the experiment, we use the estimator $\hat{\mathcal{D}}$ in a greedy algorithm. We term this approach the *Random Greedy Intervention Design* (Ran-GrID).

We generated 100 random UCEGs of order $p \in \{10, 20, 30\}$, with $r \times \binom{p}{2}$ edges, where parameter $0 \leq r \leq 1$ controls the graph density. For this experiment, we picked $r = 0.2$. In each graph, we selected two variables randomly to intervene on. We obtained the exact $\mathcal{D}(\mathcal{I})$ using equation (3.16). Furthermore, for a given sample size N, we estimated $\mathcal{D}(\mathcal{I})$ using Algorithm 4 and obtained empirical standard deviation of the normalized error (SDNE) over all graphs with the same size, defined as $SD(|\mathcal{D}(\mathcal{I}) - \hat{\mathcal{D}}(\mathcal{I})|/\mathcal{D}(\mathcal{I}))$. Figure 3.2 depicts SDNE versus the number of samples. As can be seen, SDNE becomes fairly low for sample sizes greater than 40. Next, we formalize our observation regarding convergence and consider the required cardinality of the set $\widehat{MEC}$ to obtain a desired accuracy in estimating $\mathcal{D}(\mathcal{I})$. We use Chernoff bound for this purpose.

Theorem 4. *Let $\bar{A}(\tilde{G})$ denote the set of undirected edges of $\tilde{G}$. For the estimator in Algorithm 4, given experiment target set $\mathcal{I}$ and $\epsilon, \delta > 0$, if $N = |\widehat{MEC}| > \frac{|\bar{A}(\tilde{G})|(2+\epsilon)}{\epsilon^2}\ln(\frac{2}{\delta})$, then*

$$\mathcal{D}(\mathcal{I})(1-\epsilon) < \hat{\mathcal{D}}(\mathcal{I}) < \mathcal{D}(\mathcal{I})(1+\epsilon),$$

with probability larger than $1-\delta$.

For any $\epsilon' > 0$, for sufficiently large sample size, the Ran-GrID method provides us with a $(1 - \frac{1}{e} - \epsilon')$-approximation of the optimal value with high probability, as formalized in the following theorem.

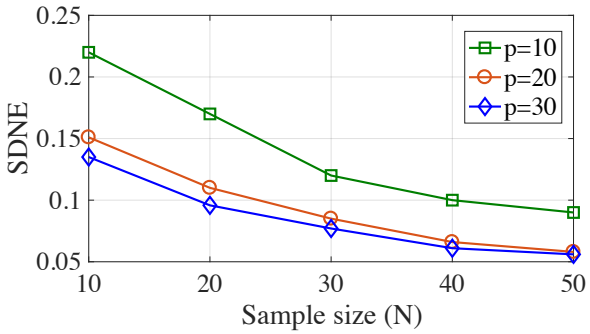

Figure 3.2: Standard deviation of the normalized error versus the sample size.

Theorem 5. *For any $\epsilon', \delta' > 0$, let $\epsilon = \frac{\epsilon'}{4k}$ and $\delta = \frac{\delta'}{4k^2}$. If for any experiment target set $\mathcal{I}$, $\mathcal{D}(\mathcal{I})(1-\epsilon) < \hat{\mathcal{D}}(\mathcal{I}) < \mathcal{D}(\mathcal{I})(1+\epsilon)$ with probability larger than $1-\delta$, then Algorithm 2 is a $(1-\frac{1}{e}-\epsilon')$-approximation algorithm with probability larger than $1-\delta'$.*

3.4.3 Fast $\mathcal{D}(\mathcal{I})$ Estimator

Recall that the computational complexity of the uniform sampler in Algorithm 4 is $\mathcal{O}(p^{\Delta+2})$, which will be intractable when the input graph has many vertices with large degrees. In this subsection, we propose another sampler, which is more suitable for graphs with large maximum degree. Although this sampler is not uniform, our extensive experimental results confirm that its sampling distribution is very close to uniform. We use this sampler in an estimator for $\mathcal{D}(\mathcal{I})$ similar to the one in Algorithm 4.

The pseudo-code of the proposed estimator is presented in Algorithm 5. In this estimator, for the given essential graph $\tilde{G}$, we generate N DAGs from the MEC of G^* as follows: We consider all subsets of size 3 from $V(\tilde{G})$ in a uniformly random order (achieved by uniformly shuffling the labels of elements of V). For each subset $\{X_i, X_j, X_k\}$, we orient the undirected edges among $\{X_i, X_j, X_k\}$ independently according to a Bernoulli(1/2) distribution. If the resulting orientation on the induced subgraph on $\{X_i, X_j, X_k\}$ is a directed cycle or a new v-structure, which was not in $\tilde{G}$, we redo the orienting. We keep checking all the subsets of size 3 until the induced subgraph on all of them are directed and none of them is a new v-structure, which did not exist in $\tilde{G}$, or a directed cycle.

Proposition 5. *Each generated DAG G_i in the sampler* FastSamp *in Algorithm 5 belongs to the Markov equivalence class of G^*.*

We generated 100 random UCEGs of order $p \in \{20, 30, ..., 60\}$ with $r \times \binom{p}{2}$ edges, where

Algorithm 5 Fast $\mathcal{D}(\mathcal{I})$ Estimator

input: Essential graph $\tilde{G}$ with chain components $\{\tilde{G}_1, ..., \tilde{G}_R\}$, target set $\mathcal{I}$, and N.
initialize: $\widehat{MEC} = \emptyset$
for $i = 1$ to N **do**
  Generate sample DAG $G_i = \textsc{FastSamp}(\tilde{G})$
  $\widehat{MEC} = \widehat{MEC} \uplus G_i$
end for
output: $\hat{\mathcal{D}}(\mathcal{I}) = \frac{1}{N} \sum_{G_i \in \widehat{MEC}} D(\mathcal{I}, G_i)$

function $\textsc{FastSamp}(\tilde{G})$
Uniformly shuffle the order of the elements of $V(\tilde{G})$.
while the induced subgraph on any subset of size 3 of the variables is not directed, or a directed cycle, or a v-structure which was not in $\tilde{G}$ **do**
  for all $\{X_i, X_j, X_k\} \subseteq V(\tilde{G})$ **do**
    Orient the undirected edges among $\{X_i, X_j, X_k\}$ independently according to $Bern(\frac{1}{2})$ until it becomes a directed structure which is not a directed cycle or a v-structure which was not in $\tilde{G}$.
  end for
end whilereturn $\tilde{G}$.

parameter $0 \leq r \leq 1$ controls the graph density. Table 3.1 shows a comparison between the run time of the fast sampler in Algorithm 5, denoted by T_f, compared to the run time of the uniform sampler in Algorithm 4, denoted by T_u, for random essential graphs with different orders. As can be seen, the run time ratio T_u/T_f increases as the order of the graphs increases.

3.5 Improved Greedy Algorithm

We exploit the submodularity of function $\mathcal{D}$ to implement an accelerated variant of the General Greedy Algorithm through *lazy* evaluations, originally proposed by [47].[2] In each round of the General Greedy Algorithm, we check the marginal gain $\Delta_X(\mathcal{I})$ for all remaining vertices in $V \backslash \mathcal{I}$. Note that as a consequence of submodularity of function $\mathcal{D}$, the set function Δ_X is monotonically decreasing. The main idea of the Improved Greedy Algorithm is to take advantage of this property to avoid checking all the variables in each round of the algorithm. More specifically, suppose for vertices X_1 and X_2, in the i-th round of the algorithm we

[2]There are improved versions of this algorithm in the literature [48].

Table 3.1: Average run time (in seconds) for the uniform sampler and the fast sampler.

		$p:$	20	30	40	50	60
$r = 0.2$	T_u		0.50	2.26	6.65	19.55	55.59
	T_f		0.018	0.055	0.163	0.3	0.63
	T_u/T_f		28.41	41.09	40.67	65.17	88.24
$r = 0.25$	T_u		0.51	2.27	7.56	25.46	59.21
	T_f		0.0218	0.06	0.1686	0.35	0.66
	T_u/T_f		23.40	37.83	44.84	72.74	89.71

have obtained marginal gains $\Delta_{X_1}(\mathcal{I}_i) > \Delta_{X_2}(\mathcal{I}_i)$. If in the $(i+1)$-th round, we calculate $\Delta_{X_1}(\mathcal{I}_{i+1})$ and observe that $\Delta_{X_1}(\mathcal{I}_{i+1}) > \Delta_{X_2}(\mathcal{I}_i)$, from monotonic decreasing property of function Δ_X, we can conclude that $\Delta_{X_1}(\mathcal{I}_{i+1}) > \Delta_{X_2}(\mathcal{I}_{i+1})$, and hence, there is no need to calculate $\Delta_{X_2}(\mathcal{I}_{i+1})$.

Improved Greedy Algorithm is presented in Algorithm 6. The idea can be formalized as follows: We define a profit parameter pro_X for each variable X and initialize the value for all variables with ∞. Moreover, we define an update flag upd_X for all variables, which will be set to `false` at the beginning of every round of the algorithm, and will be switched to `true` if we update pro_X with the value of the marginal gain of vertex X. In each round, the algorithm picks vertex $X \in V \backslash \mathcal{I}$ with the largest profit, updates its profit with the value of the marginal gain of X, and sets upd_X to `true`. This process is repeated until the vertex with the largest profit is already updated, i.e., its update flag is `true`. Then we add this vertex to $\mathcal{I}$ and end the round. For example, if in a round, the vertex X has the highest profit and after updating the profit of this vertex, pro_X is still larger than all the other profits, we do not need to evaluate the marginal gain of any other vertex and we add X to $\mathcal{I}$.

The correctness of the Improved Greedy Algorithm follows directly from submodularity of function $\mathcal{D}$. Theorem 5 holds for Algorithm 6 as well, that is, for any $\epsilon' > 0$, Improved Greedy Algorithm provides us with a $(1 - \frac{1}{e} - \epsilon')$-approximation of the optimal value. This algorithm can lead to orders of magnitude performance speedup, as shown by [40].

Algorithm 6 Improved Greedy Algorithm

input: Essential graph from the observational stage, budget k.
initialize: $\mathcal{I}_0 = \emptyset$, and $pro_X = \infty$, $\forall X \in V$.
for $i = 1$ to k **do**
  $upd_X =$ `false`, $\forall X \in V \backslash \mathcal{I}_{i-1}$
  while `true` **do**
    $X^* = \arg\max_{X \in V \backslash \mathcal{I}_{i-1}} pro_X$
    if upd_{X^*} **then**
      $\mathcal{I}_i = \mathcal{I}_{i-1} \cup \{X^*\}$
      break;
    else
      $pro_{X^*} = \mathcal{D}(\mathcal{I}_{i-1} \cup \{X^*\}) - \mathcal{D}(\mathcal{I}_{i-1})$
      $upd_{X^*} =$ `true`
    end if
  end while
end for
output: $\hat{\mathcal{I}} = \mathcal{I}_k$

3.6 Evaluation Results

3.6.1 Tree Structures

We evaluated the performance of Algorithm 1 and the Ran-GrID approach on synthetic tree structures. As shown in Section 3.3, Algorithm 1 is optimum for the worst-case gain optimization problem. We observed that this algorithm also has a good performance on the average gain optimization problem. To see this, we generated random trees based on Barabási-Albert model [49, 50], and bounded degree model created according to Galton-Watson branching process [50]. For both models we considered uniform distribution for the location of the root of the tree. Each generated tree was considered as a UCEG.

We considered an oracle experimental settings in evaluating the algorithms which can be seen as infinite sample case, in the absence of estimation errors. In particular, we assumed that the true essential graph is available as the input. Moreover, each intervention on a variable reveals the orientations of edges incident with that variable. As the performance measure, we consider the ratio of the number of edges whose directions are discovered as the result of interventions.

We generated 100 instances of random trees based on Barabási-Albert model and bounded degree model. Figure 3.3 depicts the average discovered edge ratio of Algorithm 1, Ran-GrID,

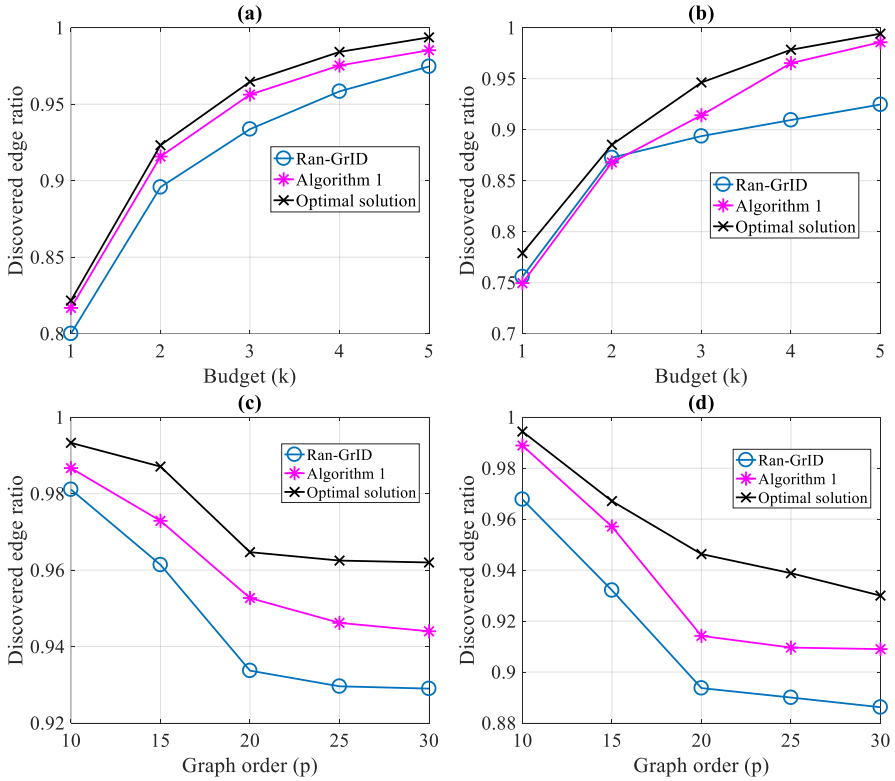

Figure 3.3: The discovered edge ratio of Algorithm 1, Ran-GrID, and the optimal solution with respect to the intervention budget with $p = 20$ (first row) and with respect to the order of the tree with $k = 3$ (second row). In the first column (parts (a) and (c)), the trees are generated based on Barabási-Albert model and in the second column (parts (b) and (d)), the trees are constructed according to the bounded degree model.

and the optimal solution for the average gain case versus budget and graph order. As can be seen, in both models, the performance of the proposed algorithm is close to the optimal solution.

3.6.2 General Structures

We evaluated the performance of the Ran-GrID algorithm for the case of general structures on synthetic and real graphs. We compared the performance of Ran-GrID with two naive approaches: 1. Rand: Selecting experiment target set randomly, 2. MaxDeg: Sorting the list of variables based on the number of undirected edges connected to them in descending order and picking the first k variables from the sorted list as the experiment target set. We studied

the performance of the algorithms on two models of random graphs, namely, Erdös-Rényi graphs and random chordal graphs, described below:

- Erdös-Rényi graphs: In this model, we first generate the skeleton of the graph by drawing an edge between any pair of vertices with a predefined probability. Then, we construct a DAG over this skeleton based on a random permutation of vertices.

- Random chordal graphs: The essential graphs of DAGs constructed from Erdös-Rényi graphs might not have large chain components. Thus, we generate random chordal graphs and consider them as a UCEG. To do so, we use randomly chosen perfect elimination ordering (PEO)[3] of the vertices to generate our underlying chordal graphs [31, 22]. For each graph, we pick a random ordering of the vertices. Starting from the vertex X with the highest order, we connect all the vertices with lower order to X with probability inversely proportional to the order of X. Then, we connect all the parents of X with directed edges, where each directed edge is oriented from the parent with the lower order to the parent with the higher order. In order to make sure that the generated graph will be connected, if vertex X is not connected to any of the vertices with the lower order, we pick one of them uniformly at random and set it as the parent of X.

We considered two experimental settings in evaluating the algorithms which we call *oracle case* and *sample case*. In the oracle case, which can be seen as infinite sample case, we execute algorithms in the absence of estimation errors. In particular, we assume that the true essential graph is available as the input. Moreover, each intervention on a variable reveals the orientations of edges incident with that variable. In the sample case, data is drawn based on a linear structural causal model with Gaussian exogenous variables. In this model, it is just needed to specify the weight of directed edges and variance of exogenous variables. Here, we drew edge weights from a uniform distribution in the range $[-1.5, -0.5] \cup [0.5, 1.5]$ and exogenous variable variances from a uniform distribution in the range $[0.01, 0.2]$. By intervening on a variable, we removed incoming edges to it and drew the samples of its exogenous variable from normal distribution $\mathcal{N}(2, 0.2)$.

Oracle case: In the oracle case, as a performance measure, we consider the ratio of the number of edges whose directions are discovered merely as a result of interventions, i.e., $D(\mathcal{I}, G^*)$ to the number of edges whose directions were not resolved from the observational

[3]A perfect elimination ordering $\{X_1, X_2, ..., X_p\}$ on the vertices of an undirected chordal graph is such that for all i, the induced neighborhood of X_i on the subgraph formed by $\{X_1, X_2, ..., X_{i-1}\}$ is a clique.

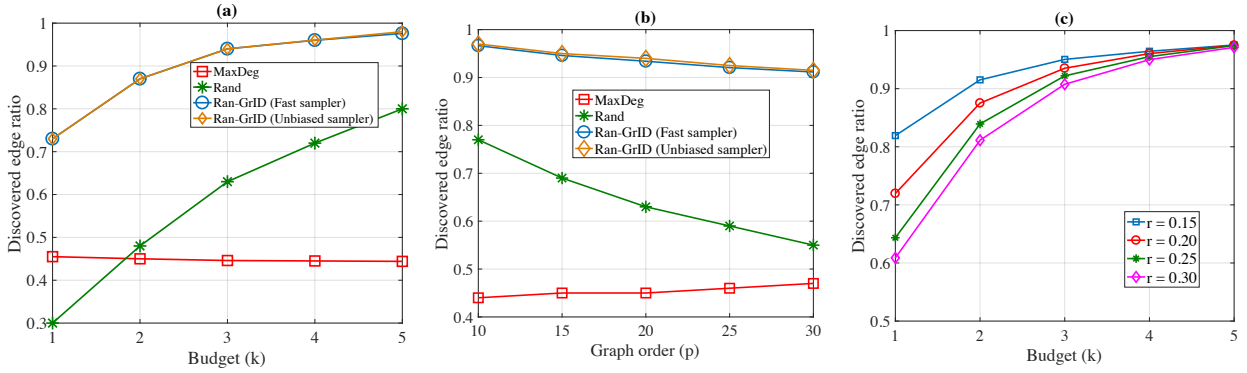

Figure 3.4: Discovered edge ratio versus (a) budget for $p = 20$, (b) graph orders for $k = 3$, (c) budget for $p = 20$ and different densities in the random chordal graphs.

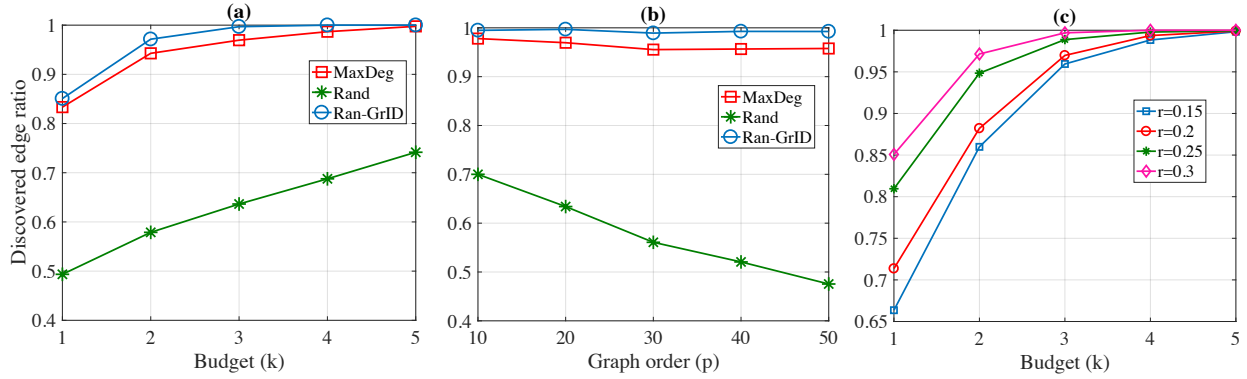

Figure 3.5: Discovered edge ratio versus (a) budget for $p = 20$, (b) graph orders for $k = 3$, (c) budget for $p = 20$ and different densities in Erdös-Rényi graphs.

data. Note that due to our specific graph generating approach in random chordal graphs, the orientation of none of the edges is learned from the observational data.

We generated 100 instances of chordal DAGs of order $p = 20$ and considered both the fast sampler and the unbiased sampler for Ran-GrID algorithm. Figure 3.4(a) depicts the discovered edge ratio with respect to the budget k. As seen in this figure, three interventions suffices to discover the direction of more than 90% of the edges. Further, to investigate the effect of the order of the graph on the performance of the proposed algorithm and two naive approaches, we evaluated the discovered edge ratio for budget $k = 3$ on graphs with order $p \in \{10, 15, 20, 25, 30\}$ in Figure 3.4(b). As can be seen in the figure, the discovered edge ratio for the proposed approach is greater than 91% for all orders. The performance of Rand approach degrades dramatically as p increases. Moreover, MaxDeg approach has even lower performance than Rand approach. Furthermore, from Figure 3.4(a-b), Ran-GrID with fast sampler has the similar performance to the one with unbiased sampler. Thus, in the rest of this section, we consider only Ran-GrID with fast sampler. We also studied the effect of

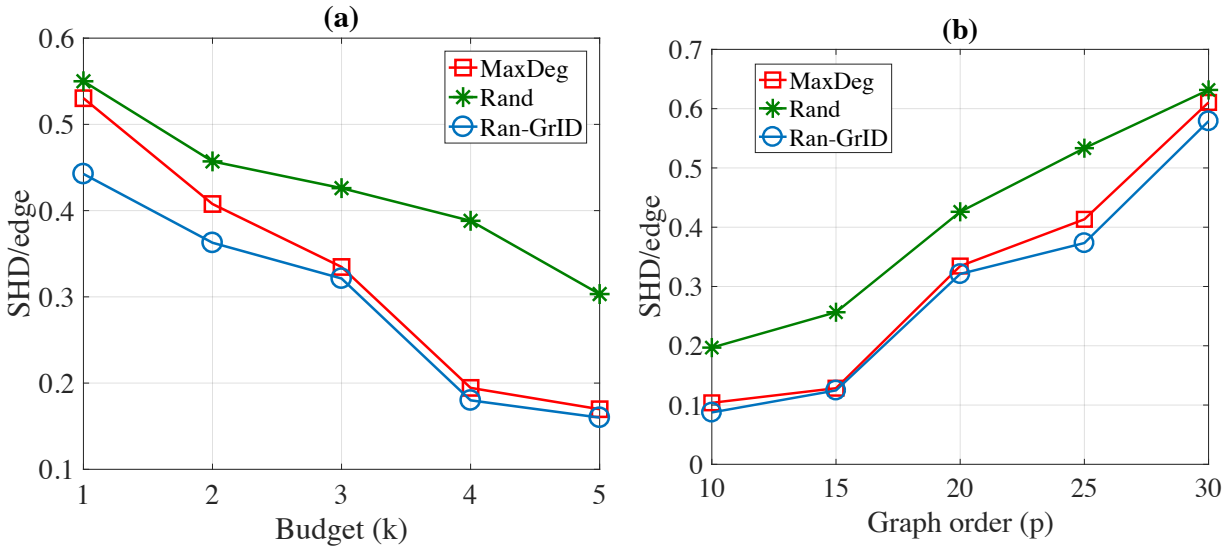

Figure 3.6: SHD per edge of true graph versus (a) budget for $p = 20$ and (b) graph orders for $k = 3$ in random chordal graphs.

graph density on the performance of proposed algorithm. Let parameter r be the ratio of average number of edges to $\binom{p}{2}$. The discovered edge ratio for chordal DAGs of order 20 versus budget for different densities is depicted in Figure 3.4(c).

Next, we generated 100 instances of Erdös-Rényi graphs and repeated the same experiments explained above. Note that in this case, the direction of some of the edges may be discovered in the observational essential graph. Experiment results are given in Figure 3.5. As can be seen, Ran-GrID approach has the best performance and MaxDeg is close to it. Moreover, the discovered edge ratio is higher for denser graphs.

Furthermore, to compare the performance of the proposed algorithm with the optimal solution, we generated 100 instances of chordal DAGs of order $p = 10$ and performed a brute force search to find the optimal solution for budget $k = 2$. The discovered edge ratio was 0.9 and 0.916 for our proposed algorithm and the optimal solution, respectively. For the aforementioned setting, the running time of the proposed approach on a machine with Intel Core i7 processor and 16 GB of RAM was 216 seconds while the one of the brute force approach was greater than 6000 seconds.

Sample case: In this part, we first generated 10^4 samples of observational data and fed them as the input to the GES algorithm [4] to obtain an estimation of the essential graph. It is noteworthy that the essential graph might be different from the true essential graph due to finite samples. Then, we generated 10^4 samples of interventional data for each experiment and gave the collection of all observational and interventional data to GIES algorithm [28] to

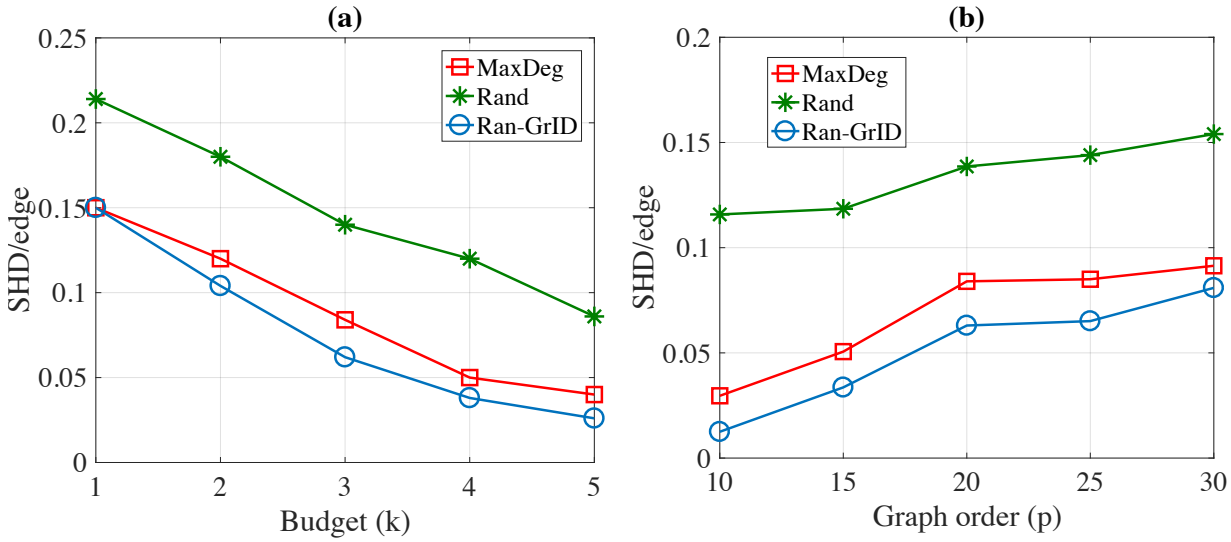

Figure 3.7: SHD per edge of true graph versus (a) budget for $p = 20$ and (b) graph orders for $k = 3$ in Erdös-Rényi graphs.

get the final output. We considered structural Hamming distance (SHD) as the performance metric, which measures the differences of the output graph and the true causal graph. Let B and $\hat{B}$ be the binary adjacency matrices of the ground truth causal DAG and the output of an algorithm, respectively. SHD is defined as follows:

$$SHD(B, \hat{B}) \coloneqq \sum_{1 \leq i < j \leq p} \mathbb{1}[(B_{ij} \neq \hat{B}_{ij}) \vee (B_{ji} \neq \hat{B}_{ji})],$$

where $\mathbb{1}[\cdot]$ is the indicator function. If the output of GES and the output of GIES after performing experiments are too different, one might exclude these instances in computing SHD since the essential graph obtained from observational data has too many errors.

In Figure 3.6(a), SHD per edges of true graph is illustrated versus the budget for $p = 20$. As can be seen, Ran-GRID outperforms other methods and it can fairly learn the true causal graph after five interventions. In Figure 3.6(b), SHD per edges of true graph is depicted versus the graph order for $k = 3$. Again, Ran-GRID has the best performance and SHD per edge increases by increasing the graph order. Next, we performed the same experiment for Erdös-Rényi graphs where the average degree of vertices is set to 3. The results are given in Figure 3.7. It can be seen that Ran-GRID performs better than other methods for any budget or graph order.

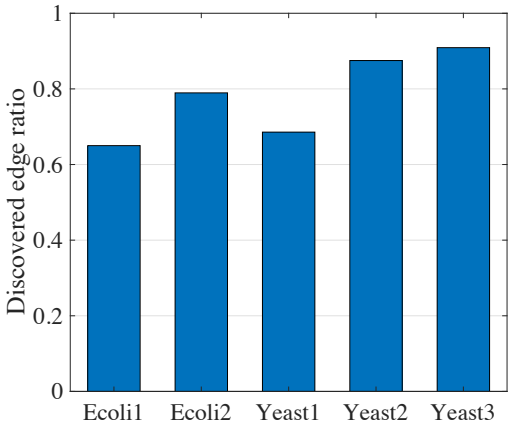

Figure 3.8: Discovered edge ratio in five GRNs from DREAM 3 challenge.

Real Graphs

We evaluated the performance of Ran-GrID in gene regulatory networks (GRN). GRN is a collection of biological regulators that interact with each other. In GRN, the transcription factors are the main players to activate genes. The interactions between transcription factors and regulated genes in a species genome can be presented by a directed graph. In this graph, links are drawn whenever a transcription factor regulates a gene's expression. Moreover, some of vertices have both functions, i.e., are both transcription factor and regulated gene.

We considered GRNs in "DREAM 3 In Silico Network" challenge, conducted in 2008 [51]. The networks in this challenge were extracted from known biological interaction networks. Since we know the true causal structures in these GRNs, we can obtain $Ess(G^*)$ and give it as an input to the proposed algorithm. Figure 3.8 depicts the discovered edge ratio in five networks extracted from GRNs of E-coli and yeast bacteria with budget $k = 5$. The order of each network is 100. As can be seen, the discovered edge ratio is at least 0.65 in all GRNs.

3.7 Conclusion

Without any assumptions on the causal modules, from observational data, a causal DAG can be learned only up to its Markov equivalence class, and hence, the direction of a large portion of the edges may be remained unidentified. In this case, it is common to perform interventions on a subset of the variables and use the resulting interventional distributions to improve the identifiability. Here, a natural question is on which variables one should

perform the intervention to gain the most from that intervention. We considered a setup in which the experimenter is limited to a budget k for the number of interventions and the interventions should be designed non-adaptively. This setup can be considered as an extension to the customary adaptive design, in which only one intervention is designed at a time. For large values of k a brute force search may not be feasible and efficient strategies for designing the interventions are required. We cast the problem as an optimization problem which aims to maximize the number of edges whose directions are identified due to the performed interventions. Here, both worst-case gain and average gain optimization can be considered. We first focused on the case that the underlying causal structure is a tree. For this case, we proposed an efficient exact algorithm for the worst-case gain setup, and an approximate algorithm for the average gain setup. The proposed approach for the average gain setup was based on our result that the objective function of the optimization in this case is monotonically increasing and submodular. In our synthetic simulations on different tree generation models, we observed that the proposed optimal algorithm for the worst-case gain also had a very high performance for the average gain. We then showed that the proposed approach for the average gain setup can be extended to the case of general causal structures. However, in this case, besides the design of interventions, calculating the objective function of the optimization problem is also challenging. This is due to the fact that the number of the members of a Markov equivalence class can potentially be super exponential in the number of the variables. We propose an efficient exact calculator for the objective function as well as two estimators. All these methods are based on a proposed method for counting and uniform sampling from the members of a Markov equivalence class. We evaluate the proposed methods using synthetic as well as real data.

Providing an exact algorithm for the average gain setup, designing interventions for the worst-case gain setup for general causal structures, and considering the problem when the variables of the system can have latent confounders are among the directions that can be considered as future work.

CHAPTER 4

MULTI-DOMAIN CAUSAL STRUCTURE LEARNING

Although interventional experiments are the gold standard for causal discovery, in many applications, intervening on certain variables in the system may be expensive, unethical, impossible, or even undefined. For example, changing the course of the planets to study the tides is impossible, forcing people to smoke to study the influence of smoking on health is unethical, modifying the placement of ads on web pages to optimize revenue may be expensive.[1] However, in many real life systems, the data generating distribution may vary over time, or the dataset may be gathered from different domains and hence not follow a single distribution [52, 53, 54, 55, 56]. While such data is usually problematic in statistical analysis and causes restrictions on the learning power, this property can be leveraged for the purpose of causal discovery, which is our focus herein. This is because of the coupling relationship between causal modeling and distribution change, i.e., the causal model constrains how the data distribution may change. Therefore, changes in the distribution help us distinguish the causal modules in the model. We refer to the task of causal discovery from such multi-domain data as the multi-domain causal structure learning. Note that in this setting, we do not intervene in or perturb the system and merely utilize the observational data gathered from different domains. In this setup the main question is how to take the most advantage of the changes across domains for the task of causal discovery.

Unlike the case of interventional causal structure learning, in multi-domain causal structure learning, the experimenter is usually not aware of the location of the changes. Also, the experimenter does not have access to the source of randomization (intervention variable). Therefore, the causal discovery cannot be done by performing conditional independence tests which directly involve the randomization sources. For instance, if in causal structure $X \to Y$, variables X and Y both vary, leading to interventional graph $W_X \to X \to Y \leftarrow W_Y$, we cannot perform conditional independence tests including W_X and W_Y. In this case, either a surrogate variable, representing change of the domain, should be used (albeit if several

[1] Examples are borrowed from the introduction of NIPS 2013 Workshop on Causality.

domains are available), or the structure learning should be done based on comparing the distributions in different domains. Furthermore, the changes in the multi-domain setup usually do not completely make the manipulated variable independent of its original parents, i.e., they are not equivalent to hard interventions defined in Definition 5.

There are relatively few works on multi-domain causal structure learning. The authors of [53] introduced mechanism change at a focal variable X as the change of the conditional distribution of X, and assumed that the marginals of all descendants of the focal variable vary. Based on this assumption, they proposed an algorithm that given a sequence of mechanism changes, finds a causal order consistent with changes in the marginals. Naturally this approach requires access to enough samples to test each variable for marginal distribution change. Invariant prediction method [54] is another approach for utilizing multi-domain data, which utilizes different domains to estimate the set of predictors of a target variable. In that work, it is assumed that the exogenous noise of the target variable does not vary across the domains. In fact, the method crucially relies on this assumption as it adds variables to the estimated predictors set only if they are necessary to keep the distribution of the target variable's noise fixed. This framework may output a set which does not contain all the parents of the target variable. Additionally, the optimal predictor set (output of the algorithm) is not necessarily unique. The authors of [55] used surrogate variables to represent the domain, and using this extra variable, proposed a two-step method to first learn the skeleton and then the direction of some edges in the structure using conditional independence tests. They proposed a constraint-based procedure to detect variables whose local mechanisms change and recover the skeleton of the causal structure over observed variables. They presented a method to determine causal orientations by making use of independent changes in the data distribution implied by the underlying causal model, benefiting from information carried by changing distributions. Due to the generality of the model, this method may require a high number of samples.

We focus on multi-domain causal structure learning in a linear Gaussian structural causal model. As mentioned in Section 2.2.1, this setup can be represented by the matrix equation $X = B^\top X + N$, where B is the weighted adjacency matrix, and the noise vector N is distributed according to the normal distribution $\mathcal{N}(0, \Omega)$. Therefore, the system can be fully described by parameters in B and Ω, which can vary across domains. We study this setup in two cases:

- **Case 1.** Only Ω varies across domains.
- **Case 2.** Both B and Ω can vary across domains.

We present efficient approaches to exploit changes across domains for causal structure learning. The proposed methods are based on the principle of independent changes (Definition 11), which states that although the cause and the effect variables are dependent, under causal sufficiency, the mechanism that generates the cause variable changes independently of the mechanism that generates the effect variable across domains. The same principle was used in [55] for utilizing non-stationary or heterogeneous data for causal structure learning. However, since that work considers a non-parametric approach, it is restricted to general independence tests among distributions, which may not have high efficiency.

For Case 1, we first propose a regression-based causal structure learning approach called Reg-MD in Section 4.2. This method directly utilizes the invariance of the functional relations of the variables to their direct causes across a set of domains. We show that Reg-MD is a sound and complete structure learning method and has the capability of learning the structure to the same extent as if the location of the changes across the domains were known and the changes were performed by the experimenter. In Section 4.3, we discuss the connection between the setup in Case 1 and the LiNGAM method, which is a well-known causal structure learning method in the literature [6]. We propose the LiNGAM-MD method which uses the multi-domain data to form a linear non-Gaussian model over variables to render using the LiNGAM method possible.

For Case 2, we propose the Gen-MD method in Section 4.4, which directly uses the principle of independent changes. Gen-MD is a score-based approach which aims to minimize the dependency among the estimated causal modules in the system. We present a polynomial algorithm for implementing the Gen-MD method. We note that invariance is a special case of the condition of independent changes, as a constant is independent of any variable. Therefore, the idea of Gen-MD can be applied to the case of the existence of invariant parameters across domains. We propose a score-based method called MC-MD for this goal in Section 4.5, and provide an efficient polynomial implementation for that. MC-MD is capable of identifying causal directions from as few as two domains. We evaluate our four proposed methods in Section 4.6 on synthetic and real datasets.

The material in this chapter is taken from [57, 58].

4.1 Problem Description

We consider a linear structural causal model over p endogenous variables $V = \{X_1, ..., X_p\}$, with Gaussian exogenous variables defined in Section 2.2.1. We assume that the correspond-

ing causal diagram G is a DAG. Therefore, as mentioned in Section 2.2.1, the weighted adjacency matrix of G, denoted by B, can be assumed to be a strictly upper triangular matrix. Since the underlying structure is a DAG, rows and columns of B can be permuted for this condition to be satisfied. Also, we assume that the system is causally sufficient, that is, the exogenous variables do not have latent confounders (common causes). This implies that the elements of N are jointly independent. Since we can always center the data, without loss of generality, we assume that N, and hence, X is zero-mean. Hence, the noise vector N is distributed according to the normal distribution $\mathcal{N}(0, \Omega)$, where Ω is a $p \times p$ diagonal matrix with $\Omega_{i,i} = \sigma_i^2 = Var(N_i)$. Therefore, the system can be fully described by parameters in B and Ω. This model induces a distribution P_V on the endogenous variables.

We consider a multi-domain setup in which observational data from variables in d domains $\mathcal{D} = \{D^{(1)}, ..., D^{(d)}\}$ is given. For any of the parameters and variables we use the superscript (i) to denote that parameter or variable in domain $D^{(i)}$. Matrices B and Ω may vary across any two domains.

Consider an ordering (i.e., a permutation) of a set of variables. An ordering on a set of variables and a DAG on those variables are *consistent* if in the ordering, every variable appears after its parents. Note that given the skeleton of a DAG, a consistent ordering determines the direction of all the edges of the DAG uniquely; however, there may be more than one ordering consistent with a given DAG. For example, for the DAG $W \to X \to Y \leftarrow Z$, orderings (W, Z, X, Y), (W, X, Z, Y), and (Z, W, X, Y) are consistent.

Definition 10 (Causal Order)**.** *An ordering on the variables is called causal if it is consistent with the ground truth causal DAG.*

Since the skeleton of the causal DAG can be identified from observational data from a single domain, the main challenge in causal structure learning is to find a causal order.

4.1.1 Principle of Independent Changes

In a structural causal model, each variable X_i is generated by its corresponding *causal module*, which is comprised of the function f_i and the exogenous variable N_i, takes the direct causes of X_i as the input, and outputs X_i.[2] Let Γ_i be the set of parameters (possibly infinite) describing the function f_i and the distribution of the exogenous variable N_i corresponding

[2]In the probabilistic formulation, the causal module corresponding to X_i is defined as the conditional distribution $P_{X_i|Pa(X_i)}$.

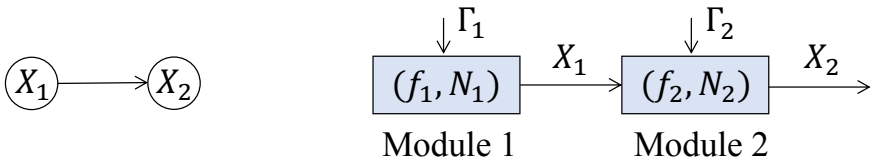

Figure 4.1: Example of causal modules.

to X_i. For instance, Figure 4.1 demonstrates a system comprised of only two variables X_1 and X_2, where X_1 is the direct cause of X_2. If the system is linear, $\Gamma_1 = \{\sigma_1^2\}$, and $\Gamma_2 = \{B_{1,2}, \sigma_2^2\}$.

As mentioned earlier, we assume that the system is causally sufficient, that is, the endogenous variables do not have latent confounders. This implies that the causal modules should change independently across the domains: When the joint distribution of a causally sufficient system changes, that is the sets Γ_i are changing, they should change independently. If Γ_i and Γ_j, are changing dependently, by Reichenbach's common cause principle [59], it implies that they have a latent common cause U. In this case, U will also be a latent common cause of X_i and X_j which violates causal sufficiency. We formalize this characteristic as follows.

Definition 11 (Principle of Independent Changes (PIC))**.** *In a causally sufficient system, the causal modules, as well as their included parameters, change independently across domains.*

The principle of independent changes can be viewed as a realization of the modularity property of causal systems [11], and as the dynamic counterpart of the principle of independent mechanisms, which states that causal modules are algorithmically independent [60, 61], or the exogeneity property of the causal system [62].

4.2 Regression-Based Multi-Domain Causal Structure Learning

We start the investigation of the problem of multi-domain causal structure learning under the assumption that the functional relationships of the variables to their direct causes across the domains are invariant. This implies the invariance of coefficients in the special case of linear structural causal model. This assumption is formally stated in the following.

Assumption 4. *The causal coefficients, i.e., the matrix B is invariant across the domains.*

The motivation behind this assumption is the belief that variation of the functional part in a causal generating mechanism should be rarer than variation of a noise variable in the

(a) (b)

Figure 4.2: Simple examples of identifiable structures using the proposed approach.

system. This assumption is in line with the setup of covariate shift which is a standard assumption in the field of transfer learning [63]. Despite the invariance of the matrix B, the matrix Ω may vary across any two domains. Note that by PIC, the variances of the exogenous noises, i.e., σ_i^2's, change independently across domains. We denote the set of variables whose corresponding exogenous variable have varied across domains $D^{(i)}, D^{(j)} \in \mathcal{D}$ by Δ_{ij}, and call it the target set across domains $D^{(i)}$ and $D^{(j)}$. The set Δ_{ij} can contain all the variables in V.

We present a regression-based causal structure learning approach for utilizing the invariances of the causal coefficients across the domains. The main idea in our proposed approach is to utilize the change of the regression coefficients, resulting from the changes across the domains, to distinguish causal directions. Using regression-based methods for structure learning is not new in the literature [64, 65, 66]. Regression-based methods have seen to be in general more robust and lead to lower estimation errors. We have the following extra assumption required for our approach:

Assumption 5. *We have one domain as the base domain, from which we learn the essential graph over the variables. We assume that the distribution in this domain satisfies Markov and faithfulness assumptions with respect to the underlying causal DAG G^*, and the correct essential graph $\tilde{G}^*$ can be learned from the base domain.*

To illustrate the idea of our regression-based approach, we use two simple examples shown in Figure 4.2. We consider having two domains and in the figure, change of an exogenous variable across the two domains is denoted by a flash sign over its corresponding endogenous variable.

Example 3. *Consider the structure in Figure 4.2(a), with structural equations $X_1 = N_1$, and $X_2 = aX_1 + N_2$, where $N_1 \sim \mathcal{N}(0, \sigma_1^2)$ and $N_2 \sim \mathcal{N}(0, \sigma_2^2)$ are independent zero-mean Gaussian exogenous variables. The exogenous variable of both X_1 and X_2 are varied*

across the domains, i.e., $\Delta_{12} = \{X_1, X_2\}$. *Denoting the regression coefficient resulting from regressing* X_i *on* X_j *by* $\beta_{i|j}$, *we have*

$$\beta_{2|1} = \frac{Cov(X_1, X_2)}{Cov(X_1)} = a,$$

and

$$\beta_{1|2} = \frac{Cov(X_1, X_2)}{Cov(X_2)} = \frac{a\sigma_1^2}{a^2\sigma_1^2 + \sigma_2^2}.$$

Therefore, $\beta_{2|1}$ *will be the same in both domains, while* $\beta_{1|2}^{(1)} = \beta_{1|2}^{(2)}$ *if* $(\sigma_1^2)^{(1)}/(\sigma_2^2)^{(1)} = (\sigma_1^2)^{(2)}/(\sigma_2^2)^{(2)}$. *Therefore, based on PIC,* $\beta_{1|2}$ *remains unvaried across domains with Lebesgue measure zero. Hence, the regression coefficient resulting from regressing the cause variable on the effect variable varies across the two domains, while the regression coefficient from regressing the effect variable on the cause variable remains the same. Therefore, the cause is distinguishable from the effect. Note that structures* $X_1 \to X_2$ *and* $X_2 \to X_1$ *are in the same Markov equivalence class and hence, not distinguishable using only one distribution.*

Remark 2. *In Example 3, note that if* σ_1^2 *and* σ_2^2 *change dependently, yet* σ_1^2/σ_2^2 *in two domains are not equal, we still can identify the causal direction. Hence, PIC is in general stronger than what we actually require for the identification approach presented in this Section.*

Example 4. *As another example, consider the structure in Figure 4.2(b). Suppose the exogenous variable of* X_1 *is varied across the two domains, i.e.,* $\Delta_{12} = \{X_1\}$. *Similar to Example 3, it can be shown that* $\beta_{1|2}$ *varies across the two domains with probability one, while* $\beta_{2|1}$ *remains the same. This implies that the edge between* X_1 *and* X_2 *is from the former to the later. Similarly,* $\beta_{2|3}$ *varies across the two domains with probability one, while* $\beta_{3|2}$ *remains the same. This implies that* X_2 *is the parent of* X_3. *Therefore, the structure in Figure 4.2(b) is distinguishable from the other structures in its Markov equivalence class.*

We now present the Regression-based Multi-Domain causal structure learning method (Reg-MD). Reg-MD takes an essential graph over the set of variables $V = \{X_1, ..., X_p\}$, and observational data from domains $\mathcal{D} = \{D^{(1)}, ..., D^{(d)}\}$ as the input and returns a graph which is the same as the input essential graph with extra identified edge directions added to it. For every pair of domains $\{D^{(i)}, D^{(j)}\}$, Reg-MD performs three steps:

S1. Find the change locations (targets), i.e., Δ_{ij}.

S2. Learn the direction of all edges incident to targets and add them to the input essential graph.

S3. Apply the Meek rules to the resulted graph from Step 2.

We first define our required notation and then explain each step in detail.

Definition 12. *For variable X_k and subset of variables X_S, $\beta_{k|S}$ denotes the regression coefficient vector resulting from regressing X_k on X_S. and $\sigma^2_{k|S} = Var(X_k - \beta_{k|S}^\top X_S)$, i.e., the variance of the residual of regressing X_k on X_S.*

Step 1. In order to implement Step 1 of Reg-MD, we need a method to find the targets of the changes. We have the following result for this aim.

Theorem 6. *For a pair of domains $(D^{(i)}, D^{(j)})$, variable $X_k \in V$ is a change target across the two domains almost surely if and only if*

$$(\sigma^2_{k|S})^{(i)} \neq (\sigma^2_{k|S})^{(j)} \qquad \forall X_S \subseteq N(X_k).$$

Based on Theorem 6, for any variable X_k, we search for a set $X_S \subseteq N(X_k)$ for which the variance of $X_k - \beta_{k|S}^\top X_S$ remains fixed across domains $D^{(i)}$ and $D^{(j)}$ by testing the following null hypothesis:

$$H^{ij}_{0,k,S} : \mathbb{E}[(X_k^{(i)} - (\beta^{(i)}_{k|S})^\top X_S^{(i)})^2] = \mathbb{E}[(X_k^{(j)} - (\beta^{(j)}_{k|S})^\top X_S^{(j)})^2].$$

In order to test the above null hypothesis, we can compute the variance of $X_k^{(i)} - (\beta^{(i)}_{k|S})^\top X_S^{(i)}$ in $D^{(i)}$ and $X_k^{(j)} - (\beta^{(j)}_{k|S})^\top X_S^{(j)}$ in $D^{(j)}$ and test whether these variances are equal using an F-test. If the p-value of the test for the set X_S is less than $\alpha/(p \times 2^\Delta)$, then we will reject the null hypothesis $H^{ij}_{0,k,S}$, where Δ is the maximum degree of the causal graph. If we reject all hypothesis tests $H^{ij}_{0,k,S}$ for all $X_S \subseteq N(X_k)$, then we will add X_k to set Δ_{ij}. Since we are performing at most $p \times 2^\Delta$ (for each variable, at most 2^Δ tests), we can obtain the set Δ_{ij} with total probability of false-rejection less than α.

Step 2. In order to implement Step 2 of Reg-MD, we need a method to learn the direction of all the edges incident to each of the targets. We have the following result for this aim.

Theorem 7. *For a pair of domains $(D^{(i)}, D^{(j)})$ with target set Δ_{ij}, for every target variable $X_k \in \Delta_{ij}$, $Pa(X_k)$ is almost surely the maximal set $X_S \subseteq N(X_k)$, for which $\beta^{(i)}_{k|S} = \beta^{(j)}_{k|S}$.*

Algorithm 7 Reg-MD

input: Essential graph $\tilde{G}$, observational data over V in domains $\mathcal{D} = \{D^{(1)}, ..., D^{(d)}\}$
for each pair of domains $(D^{(i)}, D^{(j)})$ **do**
    Obtain Δ_{ij} (Theorem 6)
    Orient all edges incident to variables in Δ_{ij} in graph $\tilde{G}$ (Theorem 7)
end for
Apply the Meek rules to $\tilde{G}$
output: $\tilde{G}$

Based on Theorem 7, for any variable X_k whose exogenous variable has changed across domains $D^{(i)}$ and $D^{(j)}$, we find the maximal subset of its neighbors X_S for which the regression coefficient of regressing X_k on X_S is the same in two domains. We orient the edges from X_S to X_k towards X_k and for the rest of the incident edges incident to X_k, we orient them outward from X_k.

Step 3. After identifying the direction of the edges incident to the targets, one can apply the Meek rules [17] to potentially learn the direction of some extra edges. These are the edges which if directed in the other direction, will create a cycle or a v-structure which does not exist in the observational essential graph.

The pseudo-code of the Reg-MD algorithm is presented in Algorithm 7. In this algorithm, for each pair of domains, we first find the target set using Theorem 6, and then orient all the edges incident to the variables in the target set using Theorem 7. At the end, Meek rules are applied to the partially directed graph.

Remark 3. *In interventional causal structure learning approaches, the experimenter intervenes on a subset of variables. Under some conditions on the type of the interventions, she can learn the direction of the edges incident to the targets of the interventions, and then she can apply the Meek rules. Our results in Theorems 6 and 7 show that what we learn via Reg-MD is the same as the identification level resulted from an interventional causal structure learning approach. This is despite the constraint that here we do not utilize any concepts surrogating the values of the interventions to enable us performing statistical tests (such as conditional independence test) on those values.*

4.2.1 Completeness of Reg-MD

Based on Theorems 6 and 7, the proposed Reg-MD approach is sound, but is it complete? That is, is Reg-MD capable of extracting all the information in the domains related to the

task of structure learning? For instance, one may wonder that one can get more information about the ground truth structure by investigating the changes in regression coefficients of all variables (i.e., also considering the ones which are not a target of change) on all possible subsets of the rest of the variables. To answer this question, we first define the alteration DAG.

Definition 13. *The alteration DAG corresponding to the domain set* $\mathcal{D} = \{D^{(1)}, ..., D^{(d)}\}$ *is the same DAG as the ground truth DAG with alteration variable* A_k *augmented to it as a parent of* X_k*, for all* $X_k \in \Delta_{ij}$*, for all* $i, j \in [d]$*.*

The information in the set of domains can be interpreted as observational data coming from one domain generated by the alteration graph. This can be realized by interpreting each alteration variable as a switch which determines specific values for the parameters of the causal module of its corresponding endogenous variable. Therefore, we have one observational domain from the alteration DAG, in which we know the direction of edges incident to the alteration variables.

Consider the hypothetical scenario in which the alteration DAG is the ground truth structure and alteration variables are normal endogenous variables in the system from which we have observational data similar to the rest of the endogenous variables, and we know the direction of edges incident to the alteration variables. It is known that Markov equivalence is the extent of identifiability for DAGs from observational linear Gaussian data. That is, using any other type of statistical tests besides conditional independence tests will not improve identifiability (see Chapter 5). Therefore, the essential graph corresponding to the alteration graph is the extent of learnability. Reg-MD already identifies the DAG to the same level as this hypothetical scenario and hence, it extracts all the available information related to structure learning. The above argument concludes in the following result.

Theorem 8. *The Reg-MD algorithm is complete.*

Example 5. *Suppose we have observational data from the DAG in Figure 4.3(a) in two domains* $D^{(1)}$ *and* $D^{(2)}$*, where across the domains we have* $\Delta_{12} = \{X_1\}$*. The corresponding alteration DAG is depicted in Figure 4.3(b) in which alteration variable* A_1 *is added to the original DAG as a parent of* X_1*. Figure 4.3(c) shows the identifiable structure from this multi-domain data, which is also the output of the Reg-MD algorithm.*

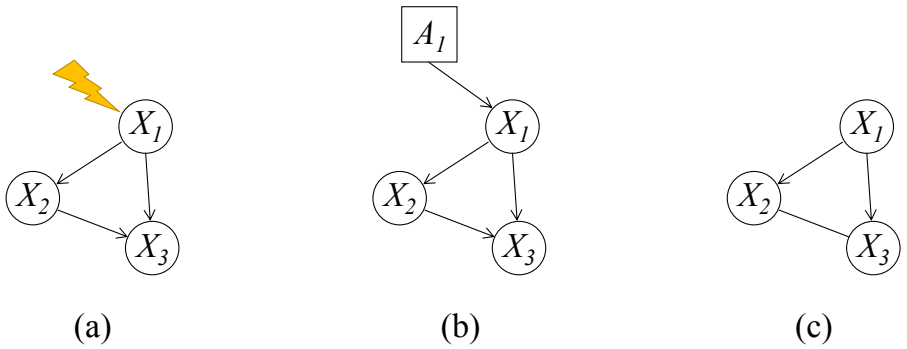

Figure 4.3: Graphs related to Example 5.

4.3 LiNGAM-Based Multi-Domain Causal Structure Learning

Under Assumption 4, one can also utilize the LiNGAM approach for the task of multi-domain causal structure learning: [6] proposed a non-Gaussian version of the linear structural causal model in expression (2.2), known as linear non-Gaussian acyclic model (LiNGAM), in which exogenous variables N_i are assumed to be non-Gaussian. LiNGAM is proven to be uniquely identifiable, i.e., the weighted adjacency matrix B can be uniquely identified based on a single observational distribution. The identifiability of LiNGAM is based on the use of the concept of independence component analysis (ICA), which is a non-Gaussian variant of factor analysis [67, 68].

4.3.1 ICA and Identifiability of LiNGAM

For observational variables $\{X_1, ..., X_p\}$, the ICA model is defined as

$$X = A^\top S, \tag{4.1}$$

where $S := [S_1 \cdots S_p]^\top$ is a vector of jointly independent non-Gaussian component (also called source) variables, and the $p \times p$ matrix A is called the mixing matrix. That is, each observation variable X_i is obtained by a linear mixture of the component variables. The mixing matrix is assumed to be of full column rank.

The main result in ICA is that the mixing matrix can be identified up to permutation, scaling, and sign of the rows. Thus, the mixing matrix identified by ICA, denoted by A_{ICA}

satisfies

$$A_{ICA} = DPA,$$

where, P is an unknown permutation matrix, and D is an unknown scaling and sign matrix.

The most common method in ICA approach is to estimate the matrix $W \coloneqq A^{-1}$, known as the separating matrix, via minimizing the dependencies among the estimated components $\hat{S} = W_{ICA}^{\top} X$, where W_{ICA} is the estimation of W. The separating matrix is identifiable up to permutation, scaling, and sign of the columns, that is,

$$W_{ICA} = WDP.$$

As seen in expression (2.3), a linear structural causal model can be represented in the same form as the ICA model in expression (4.1). Therefore, if the elements of the vector N in (2.3) are non-Gaussian, this equation is exactly representing the ICA model. Therefore, considering $(I - B)$ serving as the separating matrix, as explained above, ICA approaches are capable of identifying $(I - B)$ up to permutation, scaling, and sign. However, LiNGAM enjoys another condition that B is a strictly upper triangular matrix. This property can be utilized for unique identifiability of the model.

As mentioned above, ICA can return $W_{ICA} = (I - B)DP$. Here, unlike the general case, the correct permutation matrix P can be determined since $(I - B)D$ contains no zeros on the diagonal, and the correct scale and sign matrix D can be determined due to the unity on the diagonal of $(I - B)$. A possible estimator would be as follows [6], which we explain assuming having access to population data:

1. Apply an ICA algorithm to the data and estimate the separating matrix W_{ICA}.
2. Permute the columns of W_{ICA} and obtain the unique matrix $W_{ICA}P^{-1}$, in which the diagonal elements are non-zero.
3. Divide each row of $W_{ICA}P^{-1}$ by its corresponding diagonal element to obtain $W_{ICA}P^{-1}D^{-1}$ with all ones on the diagonal.
4. $B_{LiNGAM} = I - W_{ICA}P^{-1}D^{-1}$.
5. Output a causal order from B_{LiNGAM}.

We refer the reader to [6, 69] for detail about the statistical and computational concerns regarding dealing with finite data.

4.3.2 Applying the LiNGAM method to Multi-Domain Data

We now present the LiNGAM-based Multi-Domain causal structure learning method (LiNGAM-MD). Suppose a causal order on the variables, denoted by π, is given. We denote the estimated weighted adjacency matrix and exogenous variance matrix corresponding to the given ordering π by $\hat{B}_\pi$ and $\hat{\Omega}_\pi$, respectively. For each domain, we estimate the regression coefficients and the variance of the exogenous variable of each variable X_i on all the variables X_j with $\pi^{-1}(X_j) < \pi^{-1}(X_i)$, i.e., all the variables, which precede X_i in the given order. More formally, for any domain $D^{(i)}$,

$$(\hat{B}_\pi)^{(i)}_{j,k} = \text{the entry in } \beta^{(i)}_{\pi(k)|\{\pi(1),\dots,\pi(k-1)\}} \text{ corresponding to } \pi(j), \tag{4.2}$$

and

$$(\hat{\Omega}_\pi)^{(i)}_{k,k} = (\sigma^2_{\pi(k)|\{\pi(1),\dots,\pi(k-1)\}})^{(i)}, \tag{4.3}$$

for all $k \in [p]$, and $j \in [k-1]$, where, $\sigma^2_{i|S} = \mathrm{Var}(X_i - \beta^\top_{i|S} X_S)$.[3] Therefore, in a linear structural causal model, given the causal order on the variables, the structure (more specifically, the weighted adjacency matrix) can be estimated. Therefore, it remain to estimate a causal order. In the following, we show that having data from sufficiently many domains, we can form a LiNGAM from the multi-domain data and hence estimate a causal order.

For every index $i \in [p]$, we denote the variance of the endogenous variable X_i by ψ_i^2, and the variance of the exogenous variable N_i by σ_i^2. In equation (2.3) denote the i-th column of the matrix A by α_i. We have

$$\begin{aligned}
\psi_i^2 &= Var(\alpha_i^\top N) \\
&= \alpha_i^\top \Omega \alpha_i \\
&= (\alpha_i \odot \alpha_i)^\top \sigma^2 = (\alpha_i^{\odot 2})^\top \sigma,
\end{aligned}$$

where, the operator $\odot$ denotes the Hadamard product and σ is a column vector of size p with σ_i^2 as the i-th entry. Therefore, we have

$$\psi = (A^{\odot 2})^\top \sigma, \tag{4.4}$$

where, ψ is a column vector of size p with ψ_i^2 as the i-th entry.

[3]Note that the estimated matrices $\hat{B}$ for all causal orders are equal up to permutation. Same for estimated matrices $\hat{\Omega}$.

Algorithm 8 LiNGAM-MD

input: Observational data over V in domains $\mathcal{D} = \{D^{(1)}, ..., D^{(d)}\}$
Estimate $(\psi_j^2)^{(i)}$ for all $j \in [p]$, $i \in [d]$
Give $\psi^{(i)}$, $i \in [d]$ as the input to LiNGAM algorithm to obtain a causal order $\hat{\pi}_c$ over $\{X_1, ..., X_p\}$
Estimate $(\hat{B}_{\hat{\pi}_c})_{j,k}^{(i)}$ by expression (4.2), $k \in [p]$, $j \in [k-1]$, $i \in [d]$
$B_{j,k} = \frac{1}{d} \sum_{i=1}^{d} (\hat{B}_{\hat{\pi}_c})_{j,k}^{(i)}$, $j \in [p]$, $k \in [p]$
output: B

Now, consider a multi-domain setup with d domains and consider the values of variances $\sigma^{(i)}$, $1 \leq i \leq d$ as d samples from the random vector σ in (4.4). Due to PIC, the entries of σ are jointly independent random variables. Also, since they are the values of variances, they are non-Gaussian. Hence, equation (4.4) satisfies the requirements of the ICA model. Moreover, the matrix $A^{\odot 2}$ is upper triangular and has all ones on the main diagonal. Therefore, the LiNGAM method can be used to learn a causal order over the variables. Hence, we use the following approach to identify the weighted adjacency matrix B: We first estimate the variances of the variables in all domains. Then we use the LiNGAM method to estimate a causal order $\hat{\pi}_c$ over the variables. Then in each domain $D^{(i)}$, we regress each variable X_k on all variables before it in the causal order according to expression (4.2) to estimate coefficients $(\hat{B}_{\hat{\pi}_c})_{j,k}^{(i)}$, $1 \leq j < k$. Finally, we estimate $B_{j,k} = \frac{1}{d} \sum_{i=1}^{d} (\hat{B}_{\hat{\pi}_c})_{j,k}^{(i)}$, $j \in [p]$, $k \in [p]$. The pseudo-code of the LiNGAM-MD algorithm is presented in Algorithm 8.

Remark 4. *An alternative way to use LiNGAM is to simply apply it to the pooled data from all the domains: Since matrix B is assumed to be invariant, the pooled data will be a linear model with the same adjacency matrix as matrix B. In this model, the distribution of an exogenous variable of a targeted variable is the distribution of pooled data from more than one Gaussian distribution, and hence will not be Gaussian anymore. The issue with this approach compared to LiNGAM-MD is that if the variances of the exogenous variable of a targeted variable in the domains are close to each other, the resulting pooled distribution will still be approximately Gaussian. We have provided a comparison of this baseline use of LiNGAM versus our proposed LiNGAM-MD method in Section 5.5.*

Remark 5 (Comparison of LiNGAM-MD and Reg-MD)**.** *As mentioned above, in the LiNGAM-MD method, the estimations of the parameters in domains serve as samples from their corresponding random variables. Therefore, LiNGAM-MD requires several domains to have an acceptable performance, while ReG-MD can learn the causal relation between two variables*

with as few as two domains. Also, LiNGAM-MD requires PIC to hold, while as mentioned in Remark 2, there are cases in which Reg-MD works under conditions weaker than PIC. On the other hand, LiNGAM-MD does not require Assumption 5, i.e., it does not require the estimation of the essential graph, and it does not require the faithfulness assumption.

4.4 General Multi-Domain Causal Structure Learning

In this section we relax Assumption 4 and consider the problem of multi-domain causal structure learning when all entries of matrices B and Ω can change across the domains. To introduce our methodology, we consider a causally sufficient system comprised of two dependent variables X_1 and X_2. Observational data for variables X_1 and X_2, or in the asymptotic case, the joint distributions of X_1 and X_2, in d domains $\mathcal{D} = \{D^{(1)}, \cdots, D^{(d)}\}$ is given. The goal is to discover the causal direction between X_1 and X_2. We denote the ground truth cause variable by $X_C \in \{X_1, X_2\}$ and the ground truth effect variable by $X_E \in \{X_1, X_2\} \setminus \{X_C\}$. The relationship between X_C and X_E in domain $D^{(i)} \in \mathcal{D}$ is denoted as follows.

$$\text{domain } D^{(i)}: \qquad X_C = N_C^{(i)}, \qquad X_E = a^{(i)} X_C + N_E^{(i)},$$

where $N_C^{(i)}$ and $N_E^{(i)}$ are independent exogenous variables with variances $(\sigma_C^2)^{(i)}$ and $(\sigma_E^2)^{(i)}$, respectively. In general, all three parameters of the model, i.e., the variances of the exogenous variables and the causal coefficient can vary across the domains. For our parametric model of interest, σ_C^2 corresponds to the causal module corresponding to the cause variable, while a and σ_E^2 correspond to the causal module corresponding to the effect variables. Therefore, PIC implies that σ_C^2 changes independently of the pair (a, σ_E^2) across the domains. Note that in general, σ_E^2 need not be independent of a, as they both correspond to the mechanism generating the effect.

Recall that $\beta_{2|1}$ denotes the linear regression coefficient obtained from regressing X_2 on X_1, and $\sigma_{2|1}^2 = \text{Var}(X_2 - \beta_{2|1}^\top X_1)$, i.e., the variance of the residual of regressing X_2 on X_1. For the causal direction, we have

$$\sigma_{C|\emptyset}^2 = \sigma_C^2, \quad \beta_{E|C} = \frac{Cov(X_C, X_E)}{Cov(X_C)} = a, \quad \sigma_{E|C}^2 = \sigma_E^2. \tag{4.5}$$

For the reverse direction, we have

$$\begin{aligned}\sigma^2_{E|\emptyset} &= a^2\sigma^2_C + \sigma^2_E, \qquad \beta_{C|E} = \frac{Cov(X_C, X_E)}{Cov(X_E)} = \frac{a\sigma^2_C}{a^2\sigma^2_C + \sigma^2_E},\\ \sigma^2_{C|E} &= \mathrm{Var}(N_C - \frac{a\sigma^2_C}{a^2\sigma^2_C + \sigma^2_E}(aN_C + N_E)) = \frac{\sigma^2_C\sigma^2_E}{a^2\sigma^2_C + \sigma^2_E}.\end{aligned} \tag{4.6}$$

For any parameter $\gamma \in \{\sigma^2_{C|\emptyset}, \beta_{E|C}, \sigma^2_{E|C}, \sigma^2_{E|\emptyset}, \beta_{C|E}, \sigma^2_{C|E}\}$, let $\gamma^{(i)}$ denote the value of this parameter in domain $D^{(i)}$, $1 \leq i \leq d$. Consider $\{\gamma^{(1)}, \cdots, \gamma^{(d)}\}$ as samples from random variable γ. As stated earlier, according to PIC, $\sigma^2_{C|\emptyset} = \sigma^2_C$ is independent form $(\beta_{E|C}, \sigma^2_{E|C}) = (a, \sigma^2_E)$, while as we can see from the expressions in (4.6), such independence does not hold in general in the reverse direction. For instance, if a and σ^2_E are both fixed, an increase in $\sigma^2_{E|\emptyset}$ always leads to an increase in $\beta_{C|E}$ and $\sigma^2_{C|E}$. Therefore, we propose our causal discovery method as follows:

To test whether X_1 is the cause of X_2, we test the independence between $\sigma^2_{1|\emptyset}$ and $(\beta_{2|1}, \sigma^2_{2|1})$. If $\sigma^2_{1|\emptyset}$ and $(\beta_{2|1}, \sigma^2_{2|1})$ are independent but the counterpart in the reverse direction is not, X_1 is considered as the cause variable and X_2 the effect variable. More specifically, for order $\pi = (i, j) \in \{(1, 2), (2, 1)\}$, let $\Gamma_{\pi(2)} = \{|\beta_{j|i}|, \sigma^2_{j|i}\}$, and define the causal order indicator

$$\mathcal{T}_\pi(\mathcal{D}) \coloneqq \sum_{\gamma \in \Gamma_{\pi(2)}} \mathcal{I}(\gamma, \sigma^2_{i|\emptyset}),$$

where any standard non-parametric measure of dependence $\mathcal{I}(\cdot, \cdot)$, such as mutual information, can be used (alternatively, one can use a test of statistical independence, such as the kernel-based method in [70]). Therefore, for inferring the causal relation between X_1 and X_2, we calculate $\mathcal{T}_{(1,2)}(\mathcal{D})$ and $\mathcal{T}_{(2,1)}(\mathcal{D})$ and pick the direction which has the smaller value, i.e.,

$$\hat{\pi}_c = \arg \min_{\pi \in \{(1,2),(2,1)\}} \mathcal{T}_\pi(\mathcal{D}).$$

Although checking for independence is sufficient for discovering causal relation, in general performing a non-parametric independence test may not be efficient. This may be specially problematic as in many applications the number of domains is small. In [58], we showed that the parametric structure of our model can be exploited to devise an efficient independence test, which only performs first-order statistical test (i.e., regarding the mean) on the boundaries of the support of the variables.

Algorithm 9 Gen-MD

input: Observational data over V in domains $\mathcal{D} = \{D^{(1)}, ..., D^{(d)}\}$, initial order π_{init} over V
initiation: $\hat{\pi}_c = \emptyset$.
while $|\pi_{init}| \neq 0$ **do**
  for $X \in \pi_{init}$ **do**
    Form $\pi_{X,-1}$
    Estimate the elements in $\Gamma_{\pi_{X,-1}(k)}$ defined in (4.7) for all $1 \leq k \leq |\pi_{init}|$
    $Q(X) = Q_{\pi_{X,-1}(k)}$ defined in (4.8) for $k = |\pi_{init}|$
  end for
  $X_{last} = \arg\min_{X \in \pi_{init}} Q(X)$
  $\hat{\pi}_c = \text{concatenate}(X_{last}, \hat{\pi}_c)$, remove X_{last} from π_{init}
end while
Estimate $(\hat{B}_{\hat{\pi}_c})^{(i)}_{j,k}$ by expression (4.2), $k \in [p]$, $j \in [k-1]$, $i \in [d]$
output: $(\hat{B}_{\hat{\pi}_c})^{(i)}$, $i \in [d]$

4.4.1 Causal Discovery for More than Two Variables

In this subsection, we present the General Multi-Domain causal structure learning method (Gen-MD), which extends the proposed method to the case of having more than two variables. As stated in Section 4.3, in a linear structural causal model, given the causal order on the variables, the structure (more specifically, the weighted adjacency matrix) can be easily estimated by regressing each variable on variables that precede it in the order. Therefore, it remain to estimate a causal order. In the following we present our approach for estimating a causal order on the variables.

According to PIC, elements in each column of $B + \Omega$ should be jointly independent of elements in any other column, as they correspond to distinct causal modules. Therefore, we can set a metric for measuring dependencies, and orders that obtain the minimum value are causal orders. More specifically, for a given order π on variables, let $\hat{B}_\pi$ and $\hat{\Omega}_\pi$ be the outputs of regression, defined in expressions (4.2) and (4.3). We define

$$\Gamma_{\pi(k)} := \{|(\hat{B}_\pi)_{j,k}|, (\hat{\Omega}_\pi)_{k,k}; 1 \leq j \leq k-1\}, \qquad 1 \leq k \leq p. \tag{4.7}$$

Also, we define

$$Q_{\pi(k)} := \sum_{\gamma \in \Gamma_{\pi(k)}} \sum_{l=1}^{k-1} \sum_{\tilde{\gamma} \in \Gamma_{\pi(l)}} \mathcal{I}(\gamma, \tilde{\gamma}), \qquad 1 \leq k \leq p, \tag{4.8}$$

where $\mathcal{I}(\cdot, \cdot)$ is again any standard measure for dependence. We define the causal order

indicator as

$$\mathcal{T}_\pi(\mathcal{D}) \coloneqq \sum_{n=2}^{p} Q_{\pi(n)}.$$

Hence, one can estimate the causal order as follows.

$$\hat{\pi}_c = \arg\min_\pi \mathcal{T}_\pi(\mathcal{D}).$$

Therefore, in low dimensions, the causal order can be found by exhaustive search over all orders. However, this is infeasible for large dimensions, as the number of orders increases super-exponentially with the number of variables. Therefore, in the following we propose an alternative efficient method for implementing Gen-MD.

The pseudo-code of the proposed approach is presented in Algorithm 9. The main idea is that in each round, we find one variable which is the last in the causal order and remove it from the list, until all the variables are ordered. The algorithm starts with a random initial order π_{init} on all variables. In each round, for each variable $X \in \pi_{init}$, it forms the order $\pi_{X,-1}$, which is the same as π_{init} with X being moved to the end of the order, and calculates the quantity $Q(X)$, which shows the amount of dependency between parameters of the causal module of X and all the other estimated parameters when X is moved to the last position in π_{init}. After calculating the quantity $Q(X)$ for all variables in π_{init}, the variable X_{last} that has the lowest value for this quantity is concatenated to the left side of our estimated order $\hat{\pi}_c$, and is removed from π_{init}. This procedure is continued until all the variables are moved to $\hat{\pi}_c$.

4.5 Minimal Change Multi-Domain Causal Structure Learning

Invariance is a special case of the condition of independent changes, as a constant is independent of any variable. Therefore, the idea of the Gen-MD method can be applied to the case of existence of invariant parameters across domains. The advantage is that in this case, fewer domains are required to identify the causal directions. There are few other works exploiting invariance for the sake of causal discovery as well. Specifically, [54] assumes that the exogenous variable for a specific target variable in the system does not vary across the domains, and [66] consider the case that when learning the causal direction between two variables, the variance of the exogenous variable of at least one of them is invariant. In this section, we give a unification and generalization of the perspectives of those previous works,

which also generalizes Assumption 4 in our Reg-MD and LiNGAM-MD methods.

To introduce our methodology, we again consider the system in Section 4.4 comprised of two dependent variables X_1 and X_2. We show that in this system, two domains are generally sufficient to identify the causal direction. We require the following assumption on the invariant parameters.

Assumption 6. *For any pair of domains $D^{(i)}$ and $D^{(j)}$, if any of the parameters in the reverse direction presented in (4.6) (i.e., $\sigma^2_{E|\emptyset}$, $\beta_{C|E}$, or $\sigma^2_{C|E}$) are invariant across the domains, then the value of all the parameters of the system involved in their expressions (i.e., σ^2_C, a, and σ^2_E) are equal in the two domains.*

Assumption 6 is mild in the sense that it only rules out a 2-dimensional subspace of a 3-dimensional space. Therefore, considering Lebesgue measure on the 3-dimensional space, we are only ruling out a measure-zero subset. This assumptions can be seen as particular realizations of the faithfulness assumption [1].

Since invariance is a special case of independent changes, based on PIC, change in one causal module does not force any changes in another causal module, i.e., a change in, say, $\sigma^2_{C|\emptyset}$, will not enforce any changes on $\beta_{E|C}$ or $\sigma^2_{E|C}$. However, in the reverse direction, as it can be seen from equations in (4.5) and (4.6), if any of the variables a, σ^2_C, and σ^2_E varies across two domains, by Assumption 6, all three variables $\sigma^2_{E|\emptyset}$, $\beta_{C|E}$, and $\sigma^2_{C|E}$ will change. Therefore, under Assumption 6, compared to the direction from effect to cause, fewer or an equal number of changes are required in the causal direction to explain the variation in the joint distribution. Therefore, we propose our causal discovery method as follows.

For order $\pi = (i, j) \in \{(1, 2), (2, 1)\}$, let $\Gamma^{MC}_\pi = \{\sigma^2_{i|\emptyset}, |\beta_{j|i}|, \sigma^2_{j|i}\}$. For any pair of domains $\{D^{(i)}, D^{(j)}\}$, let $V^{(i,j)}_\pi \coloneqq \sum_{\gamma \in \Gamma^{MC}_\pi} \mathbb{1}[\log \gamma^{(i)} \neq \log \gamma^{(j)}]$. This quantity counts the number of members of Γ^{MC}_π that vary across domains $D^{(i)}$ and $D^{(j)}$. We define the causal direction indicator

$$\mathcal{T}^{MC}_\pi(\mathcal{D}) \coloneqq \sum_{1 \leq i < j \leq d} \mathbb{1}[\pi \notin \arg \min_{\pi' \in \{(1,2),(2,1)\}} V^{(i,j)}_{\pi'}],$$

where MC stands for minimal changes. $\mathcal{T}^{MC}_\pi(\mathcal{D})$ indicates in how many of the domain pairs, π has not been the order that requires minimum number of changes to explain the variation in the joint distribution. Under Assumption 6, we have the following result.

Theorem 9. *For a given dataset $\mathcal{D}$, we have $\mathcal{T}^{MC}_{(C,E)}(\mathcal{D}) \leq \mathcal{T}^{MC}_{(E,C)}(\mathcal{D})$. The inequality is strict if there exists a pair of domains across which at least one and at most two of the parameters σ^2_C, a, and σ^2_E varies.*

Using Theorem 9, for inferring the causal relation between X_1 and X_2, we calculate $\mathcal{T}^{MC}_{(1,2)}(\mathcal{D})$ and $\mathcal{T}^{MC}_{(2,1)}(\mathcal{D})$ and pick the direction which has the smaller value, i.e.,

$$\hat{\pi}_c \in \arg\min_{\pi\in\{(1,2),(2,1)\}} \mathcal{T}^{MC}_{\pi}(\mathcal{D}).$$

4.5.1 Causal Discovery for More than Two Variables

In this subsection, we present the Minimal Change Multi-Domain causal structure learning method (MC-MD), which extends the proposed method to the case of having more than two variables. As stated in Sections 4.3 and 4.4, in order to learn the causal structure, we only need to estimate a causal order over the variables. We will present our approach for this goal in the following.

In order to generalize the method, we need the following assumption, which is the extension of Assumption 6 for the case of having more than two variables.

Assumption 7. *For any pair of domains $D^{(i)}$, $D^{(j)}$, for any variable X and set X_S, if $(\sigma^2_{X|S})^{(i)} = (\sigma^2_{X|S})^{(j)}$ (or for an entry k, $[(\beta_{X|S})^{(i)}]_k = [(\beta_{X|S})^{(j)}]_k$), then the value of all the parameters of the system involved in the expression of $\sigma^2_{X|S}$ (or $[(\beta_{X|S})]_k$) are equal in the two domains.*

Roughly speaking, Assumption 7 for the linear structural causal model states that the parameters of the model should not have been designed in a way that they cancel each other out on correlations. Assumption 7 leads to the following principle, which is the counterpart of PIC for the case of invariance.

Definition 14 (Principle of Minimal Changes (PMC))**.** *Suppose Assumption 7 holds. Among all orders, in a causal order, fewer or an equal number of parameter changes are required to explain the variation in the joint distribution.*

Therefore, we propose our causal discovery method as follows. For a given order π on variables, let $\hat{B}_\pi$ and $\hat{\Omega}_\pi$ be the outputs of regression, defined in expressions (4.2) and (4.3). We define

$$\Gamma^{MC}_{\pi} \coloneqq \{|(\hat{B}_\pi)_{j,k}|, (\hat{\Omega}_\pi)_{k,k}; 1 \leq j \leq k \leq p\}. \tag{4.9}$$

For any pair of domains $\{D^{(i)}, D^{(j)}\}$, let

$$V^{(i,j)}_{\pi} \coloneqq \sum_{\gamma\in\Gamma^{MC}_{\pi}} \mathbb{1}[\log\gamma^{(i)} \neq \log\gamma^{(j)}]. \tag{4.10}$$

Algorithm 10 MC-MD

input: Observational data over V in domains $\mathcal{D} = \{D^{(1)}, ..., D^{(d)}\}$, initial order π_{init} over V

initiation: $\hat{\pi}_c = \emptyset$.

while $|\pi_{init}| \neq 0$ **do**

  for $X \in \pi_{init}$ **do**

    Form $\Pi_X = \{\pi_{init}, \pi_{X,-1}, \pi_{X,-2}\}$.

    Estimate the elements in Γ_π^{MC} defined in (4.9) for $p = |\pi_{init}|$, $\pi \in \Pi_X$

    Obtain $V_\pi^{(i,j)}$ defined in (4.10) for $1 \leq i < j \leq d$, $\pi \in \Pi_X$

    Obtain $\mathcal{T}_\pi^{MC}(\mathcal{D})$ defined in (4.11) for $\pi \in \Pi_X$

    Update $\pi_{init} = \arg\min_{\pi \in \Pi_X} \mathcal{T}_\pi^{MC}(\mathcal{D})$

  end for

  $\hat{\pi}_c = \text{concatenate}(\pi_{init}(-1), \hat{\pi}_c)$, remove $\pi_{init}(-1)$ from π_{init}

end while

Estimate $(\hat{B}_{\hat{\pi}_c})_{j,k}^{(i)}$ by expression (4.2), $k \in [p]$, $j \in [k-1]$, $i \in [d]$

output: $(\hat{B}_{\hat{\pi}_c})^{(i)}$, $i \in [d]$

We define the MC causal order indicator as

$$\mathcal{T}_\pi^{MC}(\mathcal{D}) := \sum_{1 \leq i < j \leq d} \mathbb{1}[\pi \notin \arg\min_{\pi'} V_{\pi'}^{(i,j)}]. \tag{4.11}$$

We have the following result similar to Theorem 9.

Theorem 10. *Let π_c be a causal and π' be a non-causal order. For a given dataset $\mathcal{D}$, we have $\mathcal{T}_{\pi_c}^{MC}(\mathcal{D}) \leq \mathcal{T}_{\pi'}^{MC}(\mathcal{D})$. Also, there exist two parameters of the system γ_1 and γ_2 such that if there exist two domains $D^{(i)}$, $D^{(j)}$ with $\gamma_1^{(i)} = \gamma_1^{(j)}$ and $\gamma_2^{(i)} \neq \gamma_2^{(j)}$, then $\mathcal{T}_{\pi_c}^{MC}(\mathcal{D}) < \mathcal{T}_{\pi'}^{MC}(\mathcal{D})$.*

Using Theorem 10, one can estimate the causal order as follows.

$$\hat{\pi}_c \in \arg\min_{\pi} \mathcal{T}_\pi^{MC}(\mathcal{D}).$$

Therefore, in low dimensions, the causal order can be found by exhaustive search over all orders. However, this is infeasible for large dimensions. Therefore, in the following we propose an alternative efficient method for implementing MC-MD.

The pseudo-code of the proposed approach is presented in Algorithm 10. The main idea is that in each round, we find one variable which is the last in the causal order and remove it from the list, until all the variables are ordered. The algorithm starts with a random initial order π_{init} on all variables. In each round, for each variable $X \in \pi_{init}$, it forms 3 orders in set

Π_X: π_{init} which is the initial order, $\pi_{X,-1}$ which is the same as π_{init} with X being moved to the last position, and $\pi_{X,-2}$ which is the same as π_{init} with X being moved to one before the last position in the order.[4] The algorithm then calculates the quantity $\mathcal{T}_{\pi}^{MC}(\mathcal{D})$ for each of the three orders in Π_X, and updates π_{init} to the element of Π_X that has the minimum value for this quantity. In the case of tie, we prioritize the orders as follows: $\pi_{init} > \pi_{X,-2} > \pi_{X,-1}$. This prioritization guarantees that after performing the aforementioned update of π_{init} for all variables, the last variable in π_{init}, i.e., $\pi_{init}(-1)$, will be a sink variable, in the subgraph induced on variables in π_{init}. We concatenate $\pi_{init}(-1)$ to the left side of our estimated order $\hat{\pi}_c$ and remove it from π_t. This procedure is continued until all the variables are moved to $\hat{\pi}_c$.

Theorem 11. *In each round of Algorithm 10, if X_s is a sink variable, then for all $\pi \in \Pi_{X_s}$, $\mathcal{T}_{\pi_{X_s,-1}}^{MC}(\mathcal{D}) \leq \mathcal{T}_{\pi}^{MC}(\mathcal{D})$. Also, for any of X_s's parents, X_v, if there exists a pair of domains across which at least one and at most two of variables $Var(X_v)$, $B_{v,s}$, σ_s^2 varies, then at the end of round, $\pi_{init}(-1)$ will be a sink variable.*

Remark 6. *Finding independence in Algorithm 9 and invariance in in Algorithm 10 can also be done from top to bottom of the causal order similar to the approach used in [64]. That is, in each round we can also find a variable with highest causal order as well.*

4.6 Evaluation Results

4.6.1 Reg-MD

We generated 100 random chordal graphs of order $p = 10$. We generated data from a linear Gaussian structural causal model with coefficients drawn uniformly at random from $[-1.5, -0.5] \cup [0.5, 1.5]$, and the variance of each exogenous variable was drawn uniformly at random from $[1, 2]$. For each variable of each structure, 10^5 samples were generated. First, we considered a scenario in which we have two domains, where in the second domain, the exogenous variable of $|\Delta_{12}|$ variables were varied. The perturbed variables were chosen uniformly at random.

Let B and $\hat{B}$ be the binary adjacency matrices of the ground truth causal DAG and the output of an algorithm, respectively. We define the structural hamming distance (SHD) as

[4] We have provided an example in the Appendix B to demonstrate why it is required to consider both orders $\pi_{X,-1}$ and $\pi_{X,-2}$.

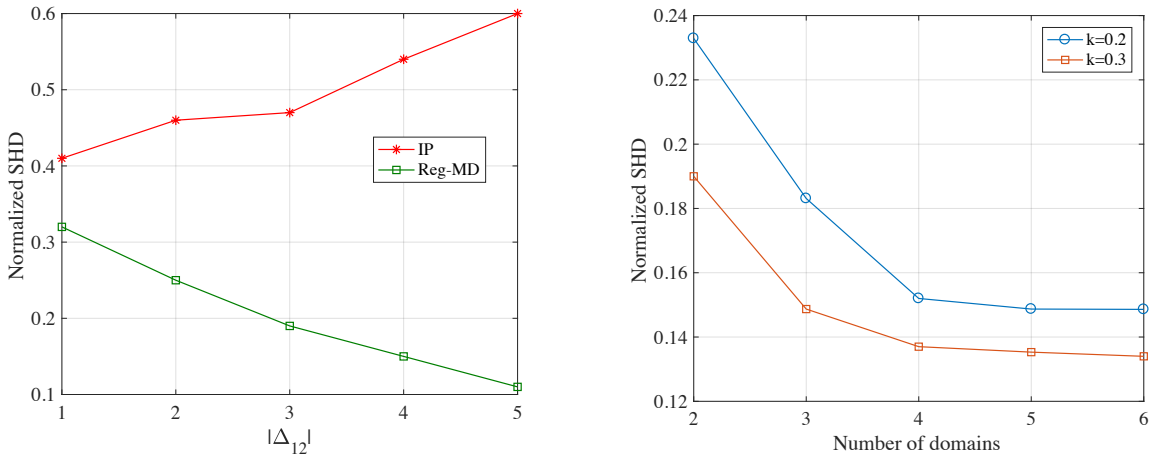

Figure 4.4: Left: Comparison of the normalized SHD of Reg-MD algorithm and IP algorithm. Right: The effect of the number of domains on the identifiability power of Reg-MD.

follows:

$$SHD(B, \hat{B}) \coloneqq \sum_{1 \leq i < j \leq p} \mathbb{1}[(B_{ij} \neq \hat{B}_{ij}) \vee (B_{ji} \neq \hat{B}_{ji})],$$

where $\mathbb{1}[\cdot]$ is the indicator function. We also the define the normalized SHD as SHD divided by $\binom{p}{2}$.

We compared the normalized SHD of Reg-MD algorithm with the invariant prediction (IP) [54] in Figure 4.4. As can be seen, the normalized SHD of IP increases as the cardinality of Δ_{12} increases. This is mainly due to the fact that in the IP approach, it is assumed that the distribution of exogenous variable of the target variable should not change, which may be violated by increasing $|\Delta_{12}|$. On the other hand, the normalized SHD of Reg-MD decreases as the cardinality of Δ_{12} increases. This shows that the proposed algorithm can correctly find the locations of changes across the two domains and use them to orient more edges for larger $|\Delta_{12}|$.

Next we considered the effect of the number of domains on the identifiability power of Reg-MD. The result is shown in Figure 4.4, where the parameter k is the fraction of variables which are varied between two consecutive domains.

4.6.2 LiNGAM-MD

We generated a random DAG of order $p = 20$ by first selecting a causal order for variables and then connecting each pair of variables with probability 0.15. We generated data from a

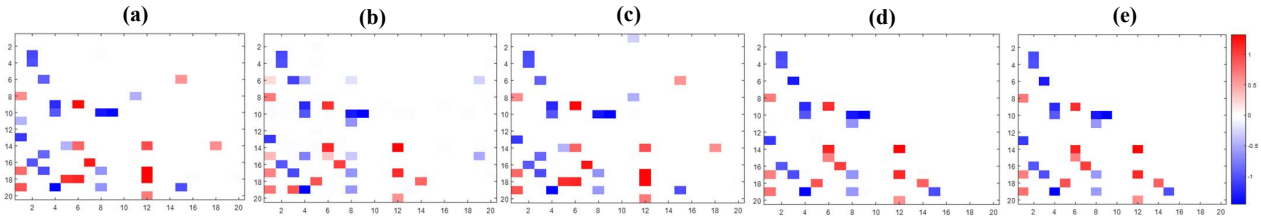

Figure 4.5: Visualization of the estimated weighted adjacency matrix with respect to the number of domains: (a) $d = p + 10$, (b) $d = p + 20$, (c) $d = p + 30$, (d) $d = p + 40$, and (e) the ground truth.

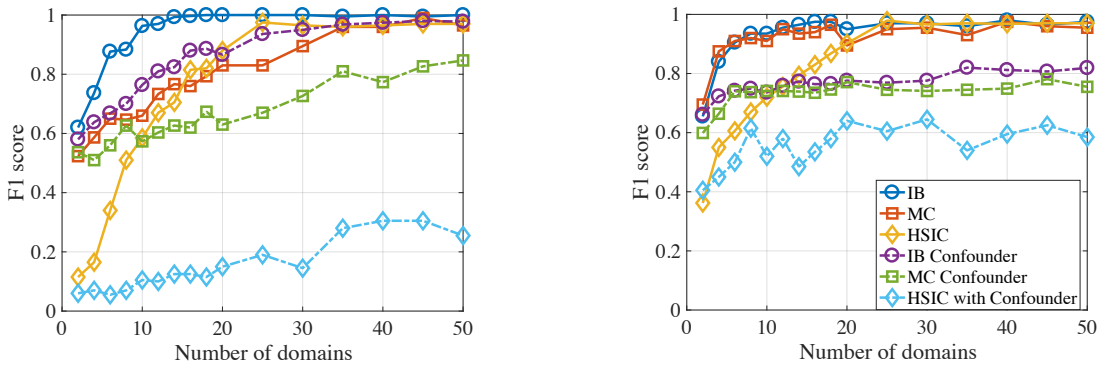

Figure 4.6: F1 score versus number of domains for model 1 on the left and model 2 on the right.

linear Gaussian structural causal model with coefficients drawn uniformly at random from $[-1.5, -0.5] \cup [0.5, 1.5]$ and fixed in all domains, and variance of each exogenous variable in each domain was drawn uniformly at random from $[1, 3]$. For each variable, 10^5 samples were generated in each domain. We estimated the weighted adjacency matrix based on the proposed LiNGAM-MD approach, where we used DirectLiNGAM algorithm for finding a causal order. The heat map of the resulting weighted adjacency matrix for different number of domains is depicted in Figure 4.5. As can be seen, the recovered matrix converges to the ground truth as the number of domains increases.

4.6.3 Gen-MD and MC-MD

We consider two models for generating the parameters of the system. In the first model, the variances of the noises and the causal coefficients follow the distributions *Unif*([1,3]) and *Unif*([-3,-0.5] ∪ [0.5,3]), respectively. In the second model, with equal probability, they

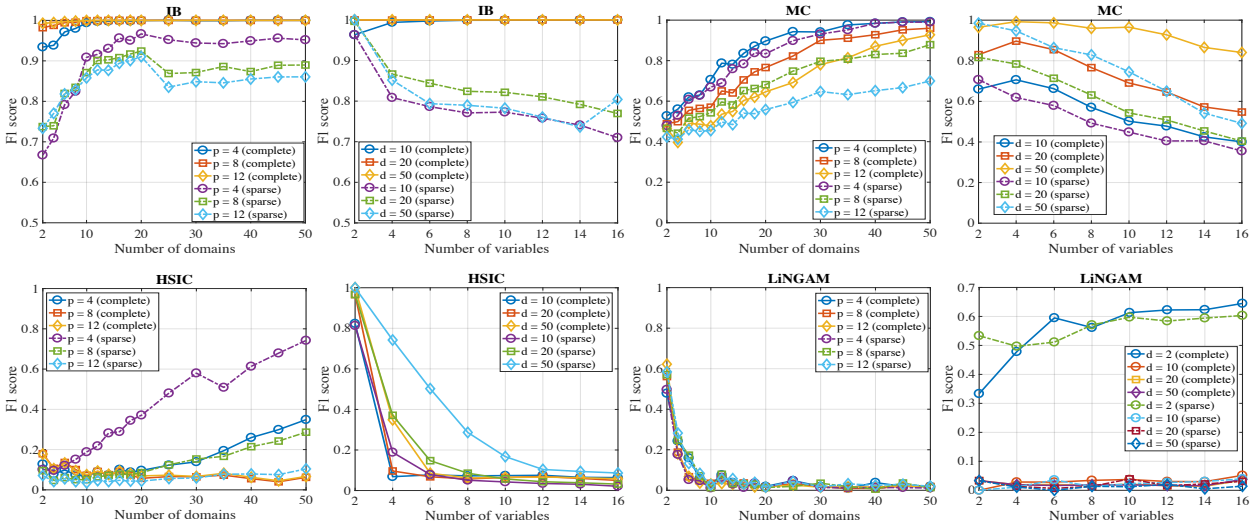

Figure 4.7: F1 score versus number of domains and number of variables.

either follow the aforementioned distributions, or are equal a fixed value. The number of samples in each domain is 10^3. We have used the state-of-the-art HISC test [70] as our non-parametric independence test. The performance of the proposed methods are depicted in Figure 4.6. We have depicted the F1 score for each case. The IB method in this figure is a parametric test for independence that we proposed in [58] and can be substituted for the non-parametric HSIC test. As seen from the F1 score, the IB method performs better than the MC method in the first model. However, if in an application we know that the parameters are not likely to change much (as in model 2), the MC method also has high performance. We also tested our proposed methods when a latent confounder was present in the system.

More than Two Variables

We considered model 1 for generating the parameters of the system, with the number of generated samples in each domain equal to 10^3. After identifying the causal ordering, we then estimate the causal coefficients B on each domain separately. We set a threshold $\alpha = 0.1$ on B from each session; if $|B_{i,j}|$ is larger than α, then there is an edge from X_i to X_j. Then if an edge appears in more than 80% of all sessions, we take this edge in the final graph. The results are shown in Figure 4.7. All experiments are performed either on complete graphs or on sparse graph generated from Erdos-Renyi model with parameter 0.3. In general, we observed better performance on denser graphs. This is expected as having more parameters

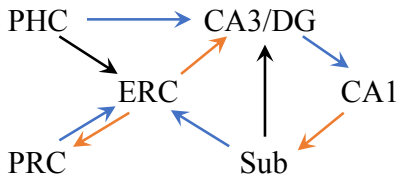

Figure 4.8: Learned structure on fMRI hippocampus data.

helps us in predicting the order. The IB and MC methods both showed high performance in our simulations. We also compared the performance with LiNGAM Algorithm [6]. To do so, we applied LiNGAM algorithm to the pooled data of all domains. As explained in [55], LiNAGM failed to perform well on our multi-domain data.

fMRI Hippocampus Data

We applied our methods to fMRI hippocampus dataset [71], which contains signals from six separate brain regions: perirhinal cortex (PRC), parahippocampal cortex (PHC), entorhinal cortex (ERC), subiculum (Sub), CA1, and CA3/Dentate Gyrus (CA3) in the resting states on the same person in 84 successive days. We used the anatomical connections [72, 55] as a reference. We applied both MC and IB on this dataset. We investigated all possible causal orders and found the one that minimizes the causal order indicator for MC and IB. After identifying the causal ordering, we estimated the causal coefficients B on each session separately with threshold $\alpha = 0.1$, and if an edge appears in more than 60% of all sessions, we took this edge in the final graph. The recovered causal graph between the six regions is shown in Figure 4.8. The black edges indicate edges, which are identified by both MC and IB methods. The blue edges are only identified by the MC method, and the orange edges are only identified by the IB method. The edges in the anatomical ground truth are as follows: PHC $\rightarrow$ ERC, PRC $\rightarrow$ ERC, ERC $\rightarrow$ CA3/DG, CA3/DG $\rightarrow$ CA1, CA1 $\rightarrow$ Sub, Sub $\rightarrow$ ERC, and ERC $\rightarrow$ CA1.

4.7 Conclusion

Under the acyclicity and causal sufficiency assumptions, more than one distribution is needed to learn the causal diagram beyond Markov equivalence. Although performing interventions in the system is the main source to obtain extra distributions, intervening on certain variables

in the system may be expensive, unethical, or even undefined. Nevertheless, in many setups, the data generating distribution may vary over time, or the dataset may be gathered from different domains and hence, not follow a single distribution. We focused on causal structure learning from such multi-domain observational data.

We proposed methods based on the principle that in a causally sufficient system, the causal modules, as well as their included parameters, change independently across domains. We study the problem in two cases:

- **Case 1.** Only Ω varies across domains.
- **Case 2.** Both B and Ω can vary across domains.

For Case 1: (1) We proposed a regression-based causal structure learning approach called Reg-MD. This method directly utilizes the invariance of the functional relations of the variables to their direct causes across a set of domains. (2) We discussed the connection between the setup in Case 1 and the LiNGAM method, and proposed the LiNGAM-MD method which uses the multi-domain data to form a linear non-Gaussian model over variables to render using the LiNGAM method possible.

For Case 2: (1) We proposed the Gen-MD method, which directly uses the principle of independent changes. We presented a polynomial algorithm for implementing the Gen-MD method. (2) Using the fact that invariance is a special case of the condition of independent changes, we applied the idea of Gen-MD to the case of the existence of invariant parameters across domains. We proposed a score-based method called MC-MD for this goal in, and provided an efficient polynomial implementation for that. MC-MD is capable of identifying causal directions from as few as two domains. See Table 4.1 for a comparison of the four proposed methods.

Table 4.1: Comparison of the four proposed multi-domain causal structure learning methods.

	Needs the essential graph	Needs several domains	Needs Assumption 4	Needs Assumption 7
Reg-MD	✓		✓	
LiNGAM-MD		✓	✓	
Gen-MD		✓		
MC-MD				✓

As future work, we consider devising regression-based methods that do not need the essential graph, consider the case that the causal directions can flip across the domains, aim for devising reliable statistical (conditional) independence tests for the setup, and consider the case that latent confounders exist in the system.

CHAPTER 5

CYCLIC CAUSAL DIAGRAMS

Most real-life causal systems contain feedback loops, since feedback is generally required to stabilize the system and improve performance in the presence of noise. Hence, the causal directed graph (DG) corresponding to such systems will be cyclic [73, 74]. However, there are relatively few works on learning structures that contain cycles. In many state-of-the-art causal models, not only is feedback ignored, it is also explicitly assumed that there are no cycles passing information among the considered quantities. Note that ignoring cycles in structure learning can be very consequential. For instance, in Figure 5.1, if one uses a conditional independence-based learning method designed for DAGs such as the PC algorithm [1], in the absence of the dashed feedback loop the skeleton will be estimated correctly on the population dataset and the directions for all edges into X_S can be determined. However, in the presence of the feedback loop, the output is a complete directed graph since no two variables will be independent conditioned on any subset of the rest of the variables.

The discrimination against cyclic structures in the literature is primarily due to the simplicity of working with acyclic models (see [73]) and the fact that in contrast to DAGs, there exists no generally accepted characterization of statistical equivalence among cyclic structures in the literature. The main method for defining equivalence among DAGs is based on the conditional independence (CI) relationships in the distributions that they imply. That is, two DAGs are equivalent if and only if they imply the same CI relations. CI relationships can be seen from statistical data, and the CI-based equivalence characterization for DAGs is attractive because CI relationships contain all the information in the distribution that can be used for structure learning under the assumption of causal sufficiency. However, when causal sufficiency is violated or cycles are allowed in the structure, conditional independency may not reflect all the information in the distribution that can be used to identify the underlying structure. That is, the joint distribution may contain information that can be used to distinguish among the members of a CI-based equivalence class, which is also known as a Markov equivalence class. This means that it is possible for two graphs to be distinguishable from observational data even though they are in the same Markov equivalence class. For

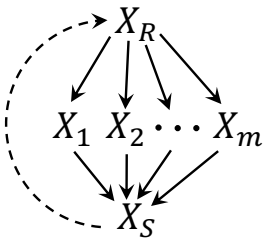

Figure 5.1: If we perform a conditional independence-based test designed for DAGs, in the presence of the feedback loop, the output will be a complete directed graph.

more details, see [7] for the case of the violation of acyclicity and [75, 76] for the case of the violation of causal sufficiency.

With the goal of bridging the gap between cyclic and acyclic DGs, in this chapter we present a general characterization of equivalence for linear Gaussian DGs.[1] In the case of DAGs, our approach provides a novel alternative to the customary tests for Markov equivalence. The proposed distribution equivalence characterization (Theorems 12 and 13) not only is capable of characterizing equivalence beyond conditional independencies, but also provides a simpler and more concise evaluation approach compared to [79]. We summarize our contributions as follows.

- We present a general, unified notion of equivalence based on the set of distributions that the directed graphs are able to generate (Section 5.1). In our proposed definition of equivalence, two structures are equivalent if they can generate the same *set* of data distributions.

- We propose an algebraic and graphical characterization of the equivalence of two DGs, be they cyclic or acyclic, based on the so-called Givens rotations (Sections 5.2 and 5.3).

- We also propose a weaker notion of equivalence called quasi-equivalence, which we show is the extent of identifiability from observational data (Section 5.4).

- We propose a score-based method for structure learning from observational data with local search. We show that our score asymptotically achieves the extent of identifiability (Section 5.4). To the best of our knowledge, this is the first local search method capable of learning structures with cycles.

[1] Note that for non-linear cyclic SEMs, even the Markov property does not necessarily hold [73, 77, 78], and hence, it is not clear if one can make general statements about the equivalence of structures regardless of the involved equations.

The material in this chapter is taken from [80].

Related Work. [2, 79] proposed graphical constraints necessary and sufficient for Markov equivalence for general cyclic DGs and proposed a constraint-based algorithm for learning cyclic DGs. That algorithm was later extended to handle latent confounders and selection bias [81]. [82, 83] also focused on structure learning based on CI relationships for possibly cyclic and causally insufficient data gathered from multiple domains that may contain conflicting CI information. They proposed an approach based on an SAT or ASP solver. Due to generality of their setup, the run time of this approach can be restricting. A similar approach was proposed in [84] for the case of nonlinear functional relationships with an extended notion of graphical separation called σ-separation. Also, [74] provided an algorithm for learning linear models with cycles and confounders that deals with perfect interventions. As mentioned earlier, having the assumption of non-Gaussian exogenous noises and specific types of non-linearity may lead to unique identifiability in DAGs. This idea was also investigated for cyclic DGs. [7] proposed a method for learning DGs based on the ICA approach for linear systems with non-Gaussian exogenous noises, and [10] investigated the case of nonlinear causal mechanisms with additive noise.

To the best of our knowledge, there exists no work on learning cyclic linear Gaussian models which utilizes the observational joint distribution itself rather than CI relationships in the distribution.

5.1 Distribution Equivalence

We consider a linear structural causal model, explained in Section 2.2.1, over p observable variables $\{X_i\}_{i=1}^p$. We assume that $B_{i,i} = 0$, for all $i \in [p]$, and elements of N are assumed to be jointly Gaussian and independent. Since we can always center the data, without loss of generality, we assume that N, and hence, X is zero-mean. Therefore, $X \sim \mathcal{N}(0, \Sigma)$, where Σ is the covariance matrix of the joint Gaussian distribution on X, and suffices to describe the distribution of X. We assume that Σ is always invertible (the Lebesgue measure of non-invertible matrices is zero). Therefore, equivalently the precision matrix $\Theta = \Sigma^{-1}$ contains all the information regarding the distribution of X. Θ can be written as

$$\Theta = (I - B)\Omega^{-1}(I - B)^\top, \tag{5.1}$$

where Ω is a $p \times p$ diagonal matrix with $\Omega_{i,i} = \sigma_i^2 = Var(N_i)$. In the sequel, we use the terms precision matrix and distribution interchangeably.

The most common notion of equivalence for DGs in the literature is independence equivalence, also called Markov equivalence. This notion is defined in Section 2.2.2. When cycles are permitted, defining equivalence of DGs based on CI relations that they represent is not suitable, as CI relations do not reflect all the information in the distribution that can be used for identification of the underlying structure; e.g., see [7]. That is, there exist DGs which can be distinguished using observational data with probability one despite representing the same CI relations.

Definition 15 (Distribution Set)**.** *The distribution set of structure G, denoted by $\Theta(G)$, is defined as*

$$\Theta(G) \coloneqq \{\Theta : \Theta = (I - B)\Omega^{-1}(I - B)^\top, \text{ for any } (B, \Omega) \\ \text{s.t. } \Omega \in diag^+ \text{ and } supp(B) \subseteq supp(B_G)\},$$

where $diag^+$ is the set of diagonal matrices with positive diagonal entries, B_G is the binary adjacency matrix of G, and $supp(B) = \{(i, j) : B_{ij} \neq 0\}$.

$\Theta(G)$ is the set of all precision matrices (equivalently, distributions) that can be generated by G for different choices of exogenous noise variances and edge weights in G.

Definition 16 (Distribution Equivalence)**.** *DGs G_1 and G_2 are distribution equivalent, or for short, equivalent, denoted by $G_1 \equiv G_2$, if $\Theta(G_1) = \Theta(G_2)$.*

It is important to note that for DG G and distribution Θ, having $\Theta \in \Theta(G)$ does not imply that all the constraints of Θ, such as its conditional independencies, can be read off of G. For instance, a complete DAG does not represent any conditional d-separations, yet all distributions are contained in its distribution set. This is due to the fact that the parameters in B can be designed to represent certain extra constraints in the generated distribution.

As mentioned earlier, we can have a pair of DGs which are distinguishable using observational data despite having the same conditional d-separations. This is not the case for DAGs. In fact, restricting the space of DGs to DAGs, Definitions 16 and 4 are equivalent.

Proposition 6. *Two DAGs G_1 and G_2 are equivalent if and only if they are I-equivalent.*

Therefore, one does not lose any information by caring only about I-equivalence when dealing with acyclic structures. All proofs are provided in the Appendix C.

For general DGs, the graphical test for I-equivalence is known to be significantly more complex [79] than the test for DAGs [16]. There are currently no known graphical conditions for distribution equivalence. This is the goal of Section 5.3.

5.2 Characterizing Equivalence

In order to determine whether DGs G_1 and G_2 are equivalent, a baseline equivalence test is as follows: We consider a distribution $\Theta \in \Theta(G_1)$ which results from a certain choice of parameters of G_1 in expression (5.1), i.e., a certain choice of exogenous noise variances and edge weights. We then check whether there exists a choice of parameters for which G_2 generates Θ. We then repeat the same procedure for G_1, considering G_2 as the original generator. More specifically, for DG G_i, let $Q_i = (I - B)\Omega^{-\frac{1}{2}}$ for any choice of B such that $\text{supp}(B) \subseteq \text{supp}(B_{G_i})$ for $i \in \{1, 2\}$. For any choice of parameters of G_1 that results in distribution $\Theta = Q_1 Q_1^\top$, we check if $Q_2 Q_2^\top = \Theta$ has real-valued solution, and vice versa. Although this baseline equivalence test provides a systematic approach, it is tedious in many cases to check for the existence of a solution. In the following, we propose an alternative equivalence test based on rotations of Q.

Let v_i be the i-th row of matrix Q. Therefore, $\Theta = QQ^\top$ is the Gramian matrix of the set of vectors $\{v_1, \cdots v_p\}$. The set of generating vectors of a Gramian matrix can be determined up to isometry. That is, given $Q_1 Q_1^\top = \Theta$, we have $Q_2 Q_2^\top = \Theta$ if and only if $Q_2 = Q_1 U$ for some orthogonal transformation U. Therefore, Q_1 should be transformable to Q_2 by a rotation or an improper rotation (a rotation followed by a reflection).

In our problem of interest, for *any* parameterization of Q_1 (resp. Q_2) it is necessary to check if there exists an orthogonal transformation of Q_1 (resp. Q_2) which can be generated for *some* parameterization of Q_2 (resp. Q_1). Therefore, only the support of the matrix before and after the orthogonal transformation matters. Hence, we only need to consider rotation transformations. This can be formalized as follows: Let Q_G be B_G with 1s on its diagonal, i.e. $Q_G := I + B_G$. This is the binary matrix that for all choices of parameters B and Ω, $supp(Q) \subseteq supp(Q_G)$.

Proposition 7. *$G_1 \equiv G_2$ if and only if for any choice of Q_1, there exists rotation $U^{(1)}$ such that $supp(Q_1 U^{(1)}) \subseteq supp(Q_{G_2})$, and for any choice of Q_2, there exists rotation $U^{(2)}$ such that $supp(Q_2 U^{(2)}) \subseteq supp(Q_{G_1})$.*

To test the existence of a rotation required in Proposition 7, we propose utilizing a sequence of a special type of planar rotations called *Givens rotations* [85].

Definition 17 (Givens rotation)**.** *A Givens rotation is a rotation in the plane spanned by two coordinate axes. For a θ-radian rotation in the (j, k) plane, the entries of the Givens rotation matrix $G(j, k, \theta) = [g]_{p\times p}$ in $\mathbb{R}^p$ are $g_{i,i} = 1$ for $i \notin \{j, k\}$, $g_{i,i} = \cos(\theta)$ for $i \in \{j, k\}$, and $g_{k,j} = -g_{j,k} = -\sin(\theta)$, and the rest of the entries are zero.*

Any rotation in $\mathbb{R}^p$ can be decomposed into a sequence of Givens rotations. Hence, in Proposition 7, we need to find a sequence of Givens matrices and define U to be their product. The advantage of this approach is that the effect of a Givens rotation is easy to track: The effect of $G(j,k,\theta)$ on a row vector v is as follows.

$$\begin{aligned}[v_1 \ \cdots \ v_j \ \cdots \ v_k \cdots \ v_p]G(j,k,\theta) = \\ [v_1 \ \cdots \ \cos(\theta)v_j + \sin(\theta)v_k \ \cdots \ -\sin(\theta)v_j + \cos(\theta)v_k \ \cdots \ v_p].\end{aligned} \tag{5.2}$$

5.2.1 Support Rotation

As previously mentioned, since all choices of parameters in the structure need to be considered, it is necessary to determine the existence of a rotation that maps one support to another. We define support matrix and support rotation as follows.

Definition 18 (Support matrix)**.** *For any matrix Q, its support matrix is a binary matrix ξ of the same size with entries in $\{0, \times\}$, where $\xi_{i,j} = \times$ if $Q_{i,j} \neq 0$ and $\xi_{i,j} = 0$ otherwise. For directed graph G, we define its support matrix as support matrix of Q_G.*

Givens rotations can be used to introduce zeros in a matrix, and hence, change its support. Consider input matrix Q. Using expression (5.2), for any $i, j \in [p]$, $Q_{i,j}$ can be set to zero using a Givens rotation in the (j,k) plane with angle $\theta = \tan^{-1}(-Q_{i,j}/Q_{i,k})$. When zeroing $Q_{i,j}$, there may exist an index l such that $Q_{l,j}$ or $Q_{l,k}$ will also become zero. However, since we consider all parameterizations of Q, we cannot take advantage of such accidental zeroings.

Definition 19 (Support Rotation)**.** *The support rotation $A(i,j,k)$ is a transformation that takes a support matrix ξ as the input and sets $\xi_{i,j}$ to zero using a Givens rotation in the (j,k) plane. The output is the support matrix of $QG(j,k,\tan^{-1}(-Q_{i,j}/Q_{i,k}))$, where $Q \in \arg\max_{Q'} |supp(Q'G(j,k,\tan^{-1}(-Q'_{i,j}/Q'_{i,k})))|$ such that the support matrix of Q' is ξ. Note that $G(j,k,\tan^{-1}(-Q'_{ij}/Q'_{i,k}))$ is the Givens rotation in the (j,k) plane which zeros $Q'_{i,j}$.*

Note that due to (5.2), $A(i,j,k)$ only affects the j-th and k-th columns of the input. The general effect of support rotation $A(i,j,k)$ is described in the following proposition.

Proposition 8. *Support rotation $A(i,j,k)$ can have three possible effects on support matrix ξ:*

1. *If $\xi_{i,j} = 0$, $A(i,j,k)$ has no effect.*

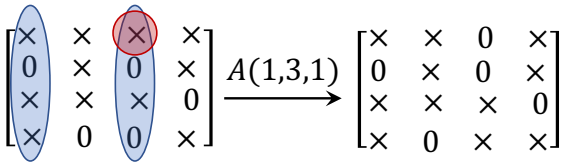

Figure 5.2: An example of support rotation (Case 2, Prop. 8). Element $\xi_{i,j}$ is in red, and columns j and k are in blue.

2. *If $\xi_{i,j} = \times$ and $\xi_{i,k} = \times$, $A(i,j,k)$ makes $\xi_{i,j} = 0$, and for any $l \in [p] \setminus \{i\}$ such that at least one of $\xi_{l,j}$ and $\xi_{l,k}$ is $\times$, $A(i,j,k)$ makes $\xi_{l,j} = \times$ and $\xi_{l,k} = \times$. This is obtained by an acute rotation.*

3. *If $\xi_{i,j} = \times$ and $\xi_{i,k} = 0$, $A(i,j,k)$ switches columns j and k of ξ. This is obtained by a $\pi/2$ rotation.*

Figure 5.2 visualizes an example of a support rotation. Observe that the following four cases partition all the effects that can be obtained from a support rotation $A(i,j,k)$.

- **Reduction.** If $\xi_{i,j} = \xi_{i,k} = \times$ and $\xi_{l,j} = \xi_{l,k}$ for all $l \in [p] \setminus \{i\}$, then only $\xi_{i,j}$ becomes zero.

- **Reversible acute rotation.** If $\xi_{i,j} = \xi_{i,k} = \times$ and there exists a row i' such that the j-th and k-th columns differ only in that row, then $\xi_{i,j}$ becomes zero and both $\xi_{i',j}$ and $\xi_{i',k}$ become $\times$.

- **Irreversible acute rotation.** If $\xi_{i,j} = \xi_{i,k} = \times$ and the j-th and k-th columns differ in at least two rows, then $\xi_{i,j}$ becomes zero and all entries on the j-th and k-th columns become $\times$ on the rows on which they differed.

- **Column swap.** If $\xi_{i,j} = \times$ and $\xi_{i,k} = 0$, then columns j and k are swapped.

Note that if ξ is transformed to ξ' via a reversible acute rotation $A(i,j,k)$, and $\xi_{i',j} = 0$, then ξ' can be mapped back to ξ via $A(i',j,k)$, hence the name reversible.

5.2.2 Characterizing Equivalence via Support Rotations

We give the following necessary and sufficient condition for distribution equivalence of two structures using the introduced support operations. We show that irreversible acute rotations

are not needed for checking equivalence. Here, for two support matrices ξ and ξ', we say $\xi \subseteq \xi'$ if $\text{supp}(\xi) \subseteq \text{supp}(\xi')$.

Theorem 12. *Let ξ_1 and ξ_2 be the support matrices of DGs G_1 and G_2, respectively. G_1 is distribution equivalent to G_2 if and only if there exists a sequence of reductions, reversible acute rotations, and column swaps that maps ξ_1 to a subset of ξ_2, and a sequence that maps ξ_2 to a subset of ξ_1.*

Theorem 12 converts the problem of determining the equivalence of two structures into a search problem for two sequences of support rotations. We propose to use a depth-first search algorithm that performs all column swaps at the end of the sequences. Due to space constraints, the pseudo-code is presented in the Appendix C.

The following result is a nontrivial application of Theorem 12 regarding reversing cycles in DGs.

Proposition 9 (Direction of Cycles)**.** *Suppose structure G_1 contains a directed cycle C. Let G_2 be a structure that differs from G_1 in two ways. (1) The direction of cycle C is reversed and (2) any variable pointing to $X_i \in C$ in G_1 via an edge which is not part of C is, in G_2, pointing to the preceder of X_i in C in G_1. In this case, G_1 is distribution equivalent to G_2. (See Figure 5.3 for an example.)*

[79] presented a result similar to Proposition 9 for the case of using CI relationships in the data and concluded that "it is impossible to orient a cycle merely using CI information." Proposition 9 extends that result by concluding that it is impossible to orient a cycle merely using observational data. The following proposition provides a necessary and sufficient condition for equivalence.

Proposition 10. *Consider DGs G_1 and G_2 with support matrices ξ_1 and ξ_2, respectively. If every pair of columns of ξ_1 differ in more than one entry, then $G_1 \equiv G_2$ if and only if the columns of ξ_2 are a permutation of columns of ξ_1.*

Example 6. *In Figure 5.4, (a) $G_1 \equiv G_2$, (b) $G_1 \not\equiv G_3$, and (c) $G_1 \equiv G_4$.*
(a) shows that unlike DAGs, equivalent DGs do not need to have the same skeleton or the same v-structures. To see $G_1 \equiv G_2$, we note that

$$\xi_1 = \begin{bmatrix} \times & \times & \times \\ 0 & \times & 0 \\ 0 & \times & \times \end{bmatrix} \xrightarrow{A(1,3,1)} \begin{bmatrix} \times & \times & 0 \\ 0 & \times & 0 \\ \times & \times & \times \end{bmatrix} \xrightarrow{A(3,1,2)} \begin{bmatrix} \times & \times & 0 \\ \times & \times & 0 \\ 0 & \times & \times \end{bmatrix} \subseteq \xi_2.$$

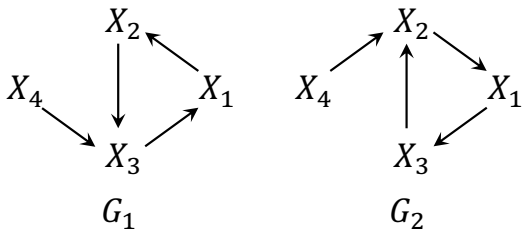

Figure 5.3: Example related to Proposition 9.

$$\xi_2 = \begin{bmatrix} \times & \times & 0 \\ \times & \times & 0 \\ 0 & \times & \times \end{bmatrix} \xrightarrow{A(2,1,2)} \begin{bmatrix} \times & \times & 0 \\ 0 & \times & 0 \\ \times & \times & \times \end{bmatrix} \xrightarrow{A(3,1,3)} \begin{bmatrix} \times & \times & \times \\ 0 & \times & 0 \\ 0 & \times & \times \end{bmatrix} \subseteq \xi_1.$$

(*b*) *follows from Proposition 10 since each pair of columns of* ξ_3 *differ in more than one entry. For* (*c*)*, we already have* $\xi_1 \subseteq \xi_4$*. For the other direction,*

$$\xi_4 = \begin{bmatrix} \times & \times & \times \\ \times & \times & 0 \\ 0 & \times & \times \end{bmatrix} \xrightarrow{A(2,1,2)} \begin{bmatrix} \times & \times & \times \\ 0 & \times & 0 \\ \times & \times & \times \end{bmatrix} \xrightarrow{A(3,1,3)} \begin{bmatrix} \times & \times & \times \\ 0 & \times & 0 \\ 0 & \times & \times \end{bmatrix} \subseteq \xi_1.$$

As seen in Example 8, structures G_1 and G_4 in Figure 5.4 are distribution equivalent. Therefore, the extra edge $X_2 \to X_1$ in G_4 does not enable this structure to generate any additional distributions. In this case, we say structure G_4 is reducible. This idea is formalized as follows.

Definition 20 (Reducibility)**.** *DG* G *is reducible if there exists* G' *such that* $G \equiv G'$ *and* $E(G') \subset E(G)$*. In this case, we say edges in* $E(G) \setminus E(G')$ *are reducible, and* G *is reducible to* G'*.*

Proposition 11. *DG* G *with support matrix* ξ *is reducible if and only if there exists a sequence of reversible acute rotations that enables us to apply a reduction to* ξ*.*

Proposition 11 implies the following necessary condition for reducibility.

Proposition 12. *A DG with no 2-cycles is irreducible.*

A 2-cycle is a cycle over only two variables, such as the cycle over X_1 and X_2 in G_2 in Figure 5.4. Propositions 11 and 12 lead to the following corollary regarding equivalence for DAGs, which bridges our proposed approach with the classic characterization for equivalence of DAGs.

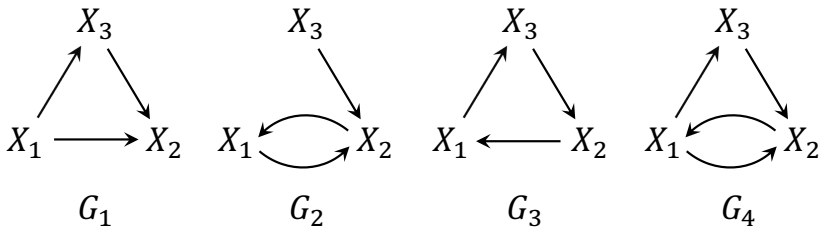

Figure 5.4: DGs related to Example 6.

Corollary 2. *DAGs G_1 and G_2 with support matrices ξ_1 and ξ_2 are equivalent if and only if there exists a sequence of reversible acute rotations and column swaps that maps ξ_1 to a subset of ξ_2, and one that maps ξ_2 to a subset of ξ_1.*

5.3 Graphical Characterization of Equivalence

In this section, we present a graphical counterpart to Theorem 12 by providing graphical counterparts to the rotations required by that Theorem.

Definition 21. *For vertices X_1 and X_2, let $P_1 := Pa(X_1) \cup \{X_1\}$ and $P_2 := Pa(X_2) \cup \{X_2\}$, where $Pa(X)$ denotes the set of parents of vertex X. X_1 and X_2 are parent reducible if $P_1 = P_2$ and parent exchangeable if $|P_1 \triangle P_2| = 1$, where $\triangle$ is the symmetric difference operator, which identifies elements which are only in one of the sets.*

The three rotations in Theorem 12 lead to the following graphical operations:

- **Parent reduction.** If X_j and X_k are parent reducible, any support rotation on columns $\xi_{\cdot,j}$ and $\xi_{\cdot,k}$ which zeros a non-zero entry on those columns except $\xi_{j,j}$ and $\xi_{k,k}$ removes the parent from X_j or X_k corresponding to the zeroed entry. We call this edge removal a parent reduction. The support rotation in this case is of reduction rotation type.

- **Parent exchange.** If X_j and X_k are parent exchangeable, by definition there exists X_i such that $P_j \triangle P_k = \{X_i\}$. In this case, any support rotation on columns $\xi_{\cdot,j}$ and $\xi_{\cdot,k}$ which zeros a non-zero entry on those columns except $\xi_{j,j}$ and $\xi_{k,k}$ removes the parent from X_j or X_k corresponding to the zeroed entry. Additionally, the missing edge from X_i to X_j or X_k is added. We call this a parent exchange. The support rotation in this case is of column swap or reversible acute rotation type.

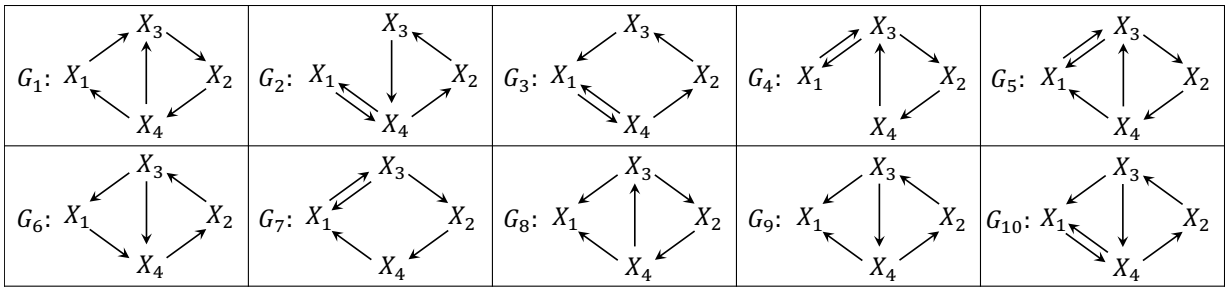

Figure 5.5: Elements of a distribution equivalence class.

- **Cycle reversion.** A cycle reversion swaps the column of each member of a cycle C with the column corresponding to its preceder in the cycle. This reverses the direction of the cycle C and changes any edge outside of C connecting to an $X_i \in C$ in the original DG to point instead to the preceder of X_i in C.

Note that in the graphical operations above, we exclude support rotations that lead to zeroing a diagonal entry, since they do not have a graphical representation (by Def. 18).

Equipped with the graphical operations, we present a graphical counterpart to Theorem 12.

Theorem 13. *G_1 is distribution equivalent to G_2 if and only if there exists a sequence of parent reductions, parent exchanges, and cycle reversions that maps G_1 to a subgraph of G_2, and a sequence that maps G_2 to a subgraph of G_1.*

Example 7. *Figure 5.5 shows the elements of a distribution equivalence class. Suppose G_1 is the original structure. Cycle reversion on the cycle (X_2, X_4, X_3, X_2) results in G_2, cycle reversion on the cycle $(X_1, X_3, X_2, X_4, X_1)$ results in G_3, parent exchange $A(4, 1, 3)$ results in G_4, and parent exchange $A(1, 3, 1)$ results in G_8.*

Remark 7. *Given observational data from any of the structures in Figure 5.5, CI-based structure learning methods such as CCD [2] may output a structure (for example G_1 without edges $X_4 \to X_1$) which is not distribution equivalent to the ground truth. This can be prevented by leveraging other statistical information in the distribution beyond CI relationships.*

We have the following corollary regarding equivalence for DAGs. The reasoning is the same as in Corollary 2.

Corollary 3. *DAGs G_1 and G_2 are equivalent if and only if there exists a sequence of parent exchanges that maps G_1 to G_2, and one that maps G_2 to G_1.*

5.4 Learning Directed Graphs from Data

Structure G imposes constraints on the entries of precision matrix Θ. We will refer to such constraints as the *distributional constraints* of G. Every distribution in $\Theta(G)$ should satisfy the distributional constraints of G. Clearly, two DGs are distribution equivalent if and only if they have the same distributional constraints. We call a distributional constraint a *hard constraint* if the set of the values satisfying that constraint is Lebesgue measure zero over the space of the parameters involved in the constraint. For instance in DAGs, if X_i and X_j are non-adjacent and have no common children, we have the hard constraint $\Theta_{i,j} = 0$. We denote the set of hard constraints of a DG G by $H(G)$.

Recall that distribution equivalence of two structures G_1 and G_2 implies that any distribution that can be generated by G_1 can also be generated by G_2, and vice versa. Therefore, no distribution can help us distinguish between G_1 and G_2. However, in practice we usually have access to only one distribution which is generated from a ground truth structure, and it may be the case that this distribution can be generated by another structure which is not equivalent to the ground truth. Therefore, finding the distribution equivalence class of the ground truth structure from one distribution is in general not possible, and extra considerations are required for the problem to be well defined. Below we will accordingly provide a weaker notion of equivalence and show that the ground truth can be recovered up to this equivalence.

The aforementioned issue also arises when learning DAGs and considering I-equivalence. The most common approach to dealing with this issue in the literature is to assume that the distribution is *faithful* to the ground truth structure. This requires a one-to-one correspondence between the conditional d-separations of the ground truth structure and the CI relationships in the distribution [1]. This is a sensible assumption from the perspective that the Lebesgue measure of the parameters which lead to extra CIs in the generated distribution is zero [86].

The case of general DGs is more complex since they can require other distributional constraints besides CIs. In particular, we may have distributional constraints other than hard constraints due to cycles. Hence, in this case the Lebesgue measure of the parameters which lead to extra distributional constraints in the generated distribution is not necessarily zero. This motivates the following weaker notion of equivalence for structure learning from observational data.

Definition 22 (Quasi Equivalence)**.** *Let θ_G be the set of linearly independent parameters*

needed to parameterize any distribution $\Theta \in \Theta(G)$. *For two DGs* G_1 *and* G_2, *let* μ *be the Lebesgue measure defined over* $\theta_{G_1} \cup \theta_{G_2}$. G_1 *and* G_2 *are quasi equivalent, denoted by* $G_1 \cong G_2$, *if* $\mu(\theta_{G_1} \cap \theta_{G_2}) \neq 0$.

Roughly speaking, two DGs are quasi equivalent if the set of distributions that they can both generate has a non-zero Lebesgue measure. Note that Definition 22 implies that if DGs G_1 and G_2 are quasi equivalent they share the same hard constraints. We have the following assumption for structure learning, which is a generalization of faithfulness:

Definition 23 (Generalized faithfulness)**.** *A distribution* Θ *is generalized faithful (Gen-faithful) to structure* G *if* Θ *satisfies a hard constraint* κ *if and only if* $\kappa \in H(G)$.

Assumption 8. *The generated distribution is Gen-faithful to the ground truth structure* G^*, *and for irreducible DG* G^*, *if there exists a DG* G *such that* $H(G) \subseteq H(G^*)$ *and* $|E(G)| \leq |E(G^*)|$, *then* $H(G) = H(G^*)$.

The following justifies the first part of Assumption 8:

Proposition 13. *With respect to Lebesgue measure over* θ_G, *the set of distributions not Gen-faithful to* G *is measure zero.*

The second part of Assumption 8 requires that if the ground truth structure G^* has no reducible edges and there exists another DG G that has only relaxed some of the hard constraints of G^*, then G must have more edges than G^*. This is clearly the case for DAGs.

Proposition 14. *Under Assumption 8, quasi equivalence is the extent of identifiability from observational data.*

5.4.1 Score-Based Structure Learning

We propose a score-based method for structure learning based on local search. Score-based methods are well-established in the literature for learning DAGs. The predominant approach is to maximize the regularized likelihood of the data by performing a greedy search over all DAGs [3], equivalence classes of DAGs [4], or permutations of the variables [87, 88]. Also, works such as [89, 90, 91, 92, 93] specifically consider the problem of learning a linear Gaussian acyclic model via penalized parameter estimation.

To the best of our knowledge, there are no existing score-based structure learning approaches for the cyclic linear Gaussian model. In light of our theory, we propose to use the

ℓ_0-regularized negative log likelihood function as the score, which is a standard choice of the score in the literature of learning DAGs, and show that it is able to recover the quasi equivalence class of the underlying DG. Let $\mathbf{X}$ be the $n \times p$ data matrix. The ℓ_0-regularized ML estimator solves the following unconstrained optimization problem:

$$\min_{G} \min_{(B,\Omega):\mathrm{supp}(B)\subseteq\mathrm{supp}(B_G)} \mathcal{L}(\mathbf{X} : B, \Omega) + \lambda \|B\|_0, \tag{5.3}$$

where

$$\mathcal{L}(\mathbf{X} : B, \Omega) = -n \log(\det(I - B)) + \sum_{i=1}^{p} \frac{n}{2} \log(\sigma_i^2) + \frac{1}{2\sigma_i^2} \|\mathbf{X}_{\cdot,i} - \mathbf{X} B_{\cdot,i}\|_2^2$$

is the negative log-likelihood of the data, $\|B\|_0 := \sum_{i,j} \mathbb{1}_{x \neq 0}(B_{i,j})$, and similar to the BIC score, we set $\lambda = 0.5 \log n$.

Remark 8. *The estimator in* (5.3) *will never output a reducible DG, since removing redundant edges improves the score. This is in line with the minimality assumption in the literature for DAGs [94, 92].*

Theorem 14. *Under Assumption 8, the global minimizer of* (5.3) *with* $\lambda = 0.5 \log n$ *outputs* $\hat{G} \cong G^*$ *asymptotically.*

Hence, by Prop. 14 and Theorem 14, the score (5.3) is consistent, i.e., it asymptotically achieves the extent of identifiability.

Structure Search

We solve the outer optimization problem in (5.3) via local search over the structures. We choose the search space to contain all DGs and use the standard operators (i.e., local changes) of edge addition, deletion, and reversal. See [14] for a discussion regarding the necessity of these operators. Two main issues arise when cycles are allowed in the structure:

Virtual edges. There exists a virtual edge between non-adjacent vertices X_i and X_j if they have a common child X_k which is an ancestor of X_i or X_j [79]. If a greedy search algorithm does not find X_k and X_i (or X_j) to be on a cycle, it can significantly increase the likelihood by adding an edge at the location of the virtual edge. The algorithm would therefore be trapped in a local optimum with one more edge than the ground truth. To resolve this issue, we propose adding the following fourth search operator: Suppose we have a triangle over three variables X_i, X_j and X_k, and there exists an additional sequence of

edges connecting X_j and X_k. In one atomic move, we perform a series of edge reversals to form a cycle containing $X_j \to X_k$ along the sequence, delete the edge connecting X_i to X_j, and orient the edge $X_i \to X_k$. If the likelihood is unchanged, the edge deletion improves the score. In the case that the oriented cycle is of length two, additional considerations are needed; see Appendix C.16 for details as well as simulations justifying this fourth operator.

Score decomposability. When the DG is acyclic, the distribution generated by a linear Gaussian structural equation model satisfies the local Markov property. This implies that the joint distribution can be factorized into the product of the distributions of the variables conditioned on their parents. The benefit of this factorization is that the computational complexity of evaluating the effect of operators can be dramatically reduced since a local change in the structure does not change the score of other parts of the DAG. In contrast, for the case of cyclic DGs the distribution does not necessarily satisfy the local Markov property. However, the distribution still satisfies the global Markov property [73]. Therefore, our search procedure factorizes the joint distribution into the product of conditional distributions. Each of these distributions is over the variables in a maximal strongly connected subgraph (MSCS), conditioned on their parents outside of the MSCS. After applying an operation, the likelihoods of all involved MSCSs are updated; see the Appendix C for additional details.

5.5 Experiments

We generated 100 random ground truth DGs of orders $p \in \{5, 20, 50\}$, all with maximum degree 4. The DGs are constrained to have maximum cycle lengths 5, 5, and 10, respectively. For each structure, we sampled the edge weights uniformly from $B_{i,j} \in [-0.8, -0.2] \cup [0.2, 0.8]$ and the exogenous noise variances uniformly from $\sigma_i^2 \in [1, 3]$ to generate the data matrix $\mathbf{X}$ of size $10^4 \times p$. We constrained the ground truth B matrices to be stable via an accept-reject approach; the modulus of all eigenvalues of B should be strictly less than one. The stability of a model guarantees that the effects of one-time noise dissipate. Our search algorithms were also constrained to only output stable structures. We used the following standard local search methods: 1. Hill climbing 2. Tabu search [14].

Evaluating the performance of a learning approach is not trivial for the case of general DGs. As seen before, equivalent cyclic DGs may have very different skeletons. Hence, conventional evaluation metrics such as structural Hamming distance (SHD) with the ground truth DG or comparison of the learned and ground truth adjacency matrices cannot be used. We propose the following evaluation methods:

1. SHD Evaluation. We enumerate the set of all DGs equivalent to the ground truth DG using Algorithm 1 in the Appendix C to form the distribution equivalence class of the ground truth. We then compute the smallest SHD between the algorithm's output DG and the members of the equivalence class as a measure of the performance.

2. Multi-Domain Evaluation. Suppose the input data is sampled from a distribution Θ generated by ground truth DG G^*, and let $\hat{G}$ denote an algorithm's output structure. Due to finite sample size and the possible violation of Assumption 8, $\hat{G}$ may be able to maximize the likelihood yet not be (quasi) equivalent to G^*. In general, we expect such an output to be compatible with only the given data and not with data sampled from other distributions generated by G^*. We therefore propose the following evaluation approach.

1. For ground truth structure G^*, generate d distributions $\{\Theta_1, ..., \Theta_d\}$ by sampling edge weights and variances.
2. For each Θ_i, run the algorithm to obtain $\hat{G}_i$.
3. For each $\hat{G}_i$, optimize its edge weights and variances to generate distributions $\{\hat{\Theta}_{i,1}, ..., \hat{\Theta}_{i,d}\}$ such that $\hat{\Theta}_{i,j}$ minimizes the KL-divergence to $\Theta_j \in \{\Theta_1, ..., \Theta_d\}$.
4. The success rate of $\hat{G}_i$ is the percentage of domains for which the minimizing KL-divergence computed in step 3 is below a threshold η.

Since domain distributions are generated randomly, if the success rate of output $\hat{G}_i$ is large, there is a non-negligible subset of the distribution set of G^* that $\hat{G}_i$ can generate as well. Hence, $\hat{G}_i$ is quasi equivalent to G^*. In our evaluations, we used $d = 50$ and $\eta = p \times 10^{-3}$. We emphasize that multi-domain data is *only* used for evaluation. In the learning stage, only one distribution is used.

We cannot compare the performance of our approach with the performance of methods based on CI relationships (such as CCD), since those approaches return a PAG representing all I-equivalent DGs, which usually represents a much larger set of DGs than the distribution equivalence class. We therefore only compared our approach with an ℓ_1-regularized maximum likelihood estimator which directly solves the optimization problem $\min_{B,\Omega} \mathcal{L}(\mathbf{X} : B, \Omega) + \lambda\|B\|_1$, which does not need a separate structure search. The results are given in Figure 5.6. The figure shows that our proposed approach successfully finds DGs capable of generating distributions generated by the ground truth structure. While the SHD evaluation shows that the outputs are not always distribution equivalent, the multi-domain evaluation provides

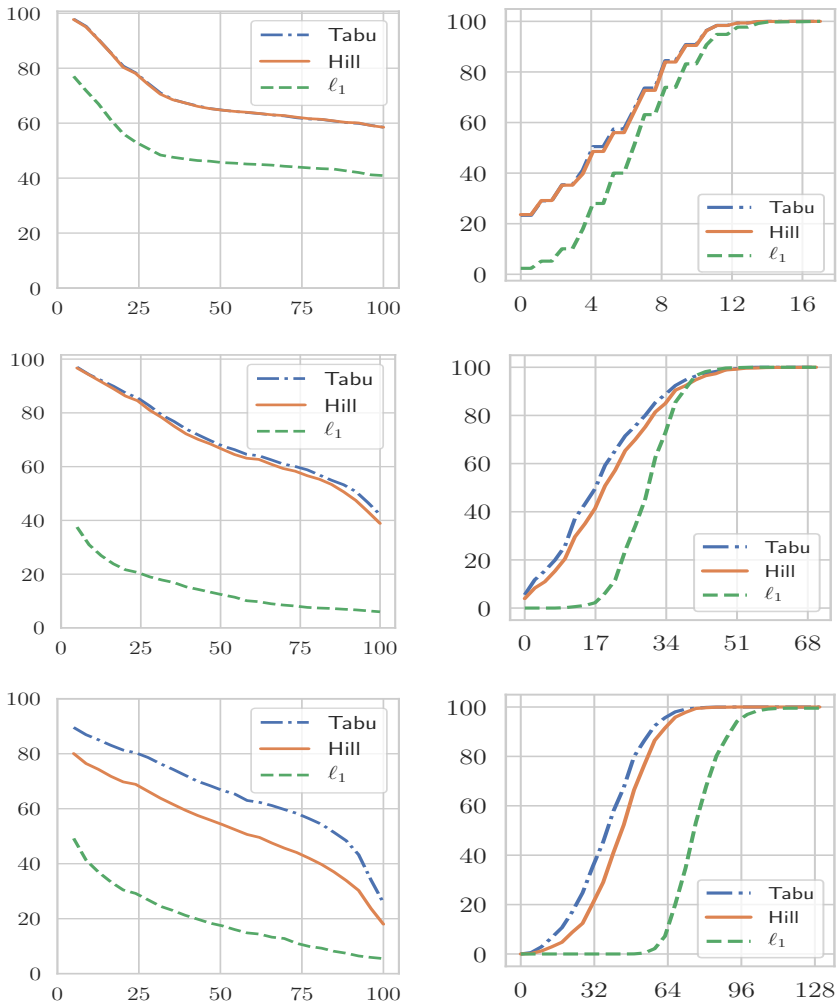

Figure 5.6: Results for $p = 5, 20, 50$, top to bottom. **Left column:** multi-domain evaluation. The percentage of outputs with success rate larger than a certain value is plotted vs. success percentages; e.g., for $p = 20$, 80% of the outputs could generate more than 25% of the distributions generated by their corresponding ground truth. **Right column:** SHD evaluation. The percentage of outputs with SHD less than or equal to a certain value is plotted vs. SHD.

evidence that many are quasi equivalent to the ground truth. We also evaluated the effect of sample size on the performance in Appendix C.

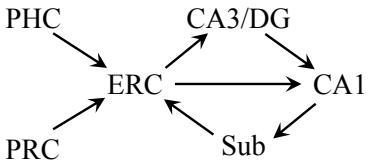

Figure 5.7: Ground truth structure for the fMRI hippocampus dataset.

5.5.1 fMRI hippocampus data

We considered the fMRI hippocampus dataset [71], which contains signals from six separate brain regions: perirhinal cortex (PRC), parahippocampal cortex (PHC), entorhinal cortex (ERC), subiculum (Sub), CA1, and CA3/Dentate Gyrus (CA3) in the resting state. We used the anatomical connections [72, 55] as the ground truth (Figure 5.7). We applied our proposed method on one of the domains in the dataset and found that two out of eight structures equivalent to the ground truth were (local) optima for the score even though there is no evidence that the data are linear Gaussian.

5.6 Conclusion

We presented a general, unified notion of equivalence for linear Gaussian DGs and proposed methods for characterizing the equivalence of two structures. We also proposed a score-based structure learning approach that asymptotically achieves the extent of identifiability. Our results are instrumental to the fields of causality and graphical models. From the causality perspective, consider for example Figure 5.5. Our results guarantee a direct causal effect between X_2 and X_4 and show that a direct causal effect does not necessarily exist between X_3 and X_4. From the graphical models perspective, our results provide the tools to handle distributions that lack a DAG representation but can be modeled by a cyclic DG. We hope that this work spurs further research in the study of directed graphs.

CHAPTER 6

LINEAR NON-GAUSSIAN CAUSAL MODELS IN THE PRESENCE OF LATENT CONFOUNDERS

As mentioned in the Introduction, if we have background knowledge about the data-generating mechanism, we may learn the underlying structure from the observed data beyond Markov equivalence [54, 23, 95, 55, 65, 9, 96, 8, 97]. For instance, [6] proposed a linear non-Gaussian acyclic model (LiNGAM) discovery algorithm that can identify causal structure uniquely, thanks to the assumption of non-Gaussian distributions for the exogenous noises in the linear structural equation model (SCM). However, LiNGAM algorithm and its regression-based variant (DirectLiNGAM) [64] rely on the causal sufficiency assumption, i.e., no unobserved common causes exist for any pair of variables that are under consideration in the model.

In the presence of latent variables, [98] showed that linear SCM can be converted to a canonical form where each latent variable has at least two children and no parents. Such latent variables are commonly called "latent confounders". Furthermore, they proposed a solution which casts the problem of identifying causal effects among observed variables into an overcomplete independent component analysis (ICA) problem [99] and returns multiple causal structures that are observationally equivalent. The time complexity of searching such structures can be as high as $\binom{p}{p_o}$ where p_o and p are the number of observed and total variables in the system, respectively. [100] proposed a method that identifies a partial causal structure among the observed variables by recovering all the unconfounded sets[1] and then learning the causal effects for each pair of variables in the set. However, their method may return an empty unconfounded set if latent confounders are the cause of most of observed variables in the system such as the simple example of Figure 6.1. [101] showed that a causal order and causal effects among observed variables can be identified if the latent confounders have Gaussian distribution and exogenous noises of observed variables are simultaneously super-Gaussian or sub-Gaussian. In [102], the ideas in DirectLiNGAM was extended to the case where latent confounders exist in the system. The proposed solution first tries to find

[1] A set of variables is called unconfounded if there is no variable outside the set which is confounder of some variables in the set. In Figure 6.1, variable V_3 is a confounder of variables V_1 and V_2 but it is not observable. Thus, the set of variables V_1 and V_2 is not unconfounded.

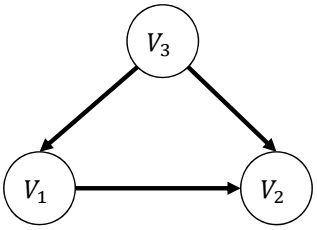

Figure 6.1: An example of causal graphs: V_1 and V_2 are observed variables while V_3 is latent.

a root variable (a variable with no parents). Then, the effect of such variable is removed by regressing it out. This procedure continues until any variable and its residual becomes dependent. Subsequently, a similar iterative procedure is used to find a sink variable and remove its effect from other variables. However, this solution may not recover causal order in some causal graphs such as the one in Figure 6.1.[2] [103] proposed a Bayesian approach for estimating the causal direction between two observed variables when the sum of non-Gaussian independent latent confounders has a multivariate t-distribution. They compute log-marginal likelihoods to infer causal directions.

There are reports in the literature of attempts to recover causal structure among observed variables in the presence of latent variables for the settings other than linear non-Gaussian model. In general cases, [1] proposed Fast Causal Inference (FCI) algorithm that can identify some causal paths in the presence of latent variables by performing conditional independence test without assuming constraints on the causal mechanism (e.g., linearity). However, it cannot guarantee the existence of causal paths in some cases such as the one where a pair of observed variables has a direct causal influence from one to the other and there is also a confounder for them. [104] proposed a method to learn Bayesian networks with latent variables based on information bottleneck concept. In the proposed method, the structure of network is learnt for a given number of hidden variables by a scored based approach with a structural expectation maximization approach. In the literature of exploratory factor analysis, there is work such as [105], which proposed a bi-factor analysis for the case with at most two latent variables in the system. In the field of Markov random model, [106] considered Gaussian Markov random field model with latent variables and tried to identify conditional independences between observed variables given all variables in the system by considering a sparsity assumption on the conditional graphical model between the observed variables. [107] utilized

[2]In Figure 6.1, the root variable (V_3) is latent and the regressor of sink variable V_2 and the residual are not independent without considering the latent variable V_3 in the set of regressors. Thus, no root or sink variable can be identified in the system.

an extension of "Verma constraints" to learn causal structures in nested Markov models with latent variables. [108] proposed a method to learn causal structure by examining the rank of submatrices of correlation matrix for the specific class of measurement model where each observed variable has exactly one latent parent.

Rather surprisingly, although the causal structure is in general not fully identifiable in the presence of latent variables, we will show that the causal order among the observed variables is still identifiable under the faithfulness assumption. In order to obtain a causal order, we first check whether there exists a causal path between any two observed variables. Subsequently, from this information, we obtain a causal order among them. Having established a causal order, we aim to figure out whether the causal effects are uniquely identifiable from observational data. We show by an example that causal effects among observed variables is not uniquely identifiable even if the faithfulness assumption holds true and the exogenous noises are non-Gaussian. We propose a method to identify the set of all possible causal effects efficiently in time that are compatible with the observational data. Furthermore, we present some structural conditions on the causal graph under which causal effects among the observed variables can be identified uniquely. We also provide necessary and sufficient graphical conditions under which the number of latent variables is uniquely identifiable. One of the applications of determining the number of latent variables from the observational data is in psychometrics, where the analysis of testing data often requires to estimate how many latent variables, the items are measuring [109, 108].

The rest of this chapter is organized as follows. In Section 6.1, we define the problem of identifying causal orders and causal effects in linear causal systems with latent variables. In Section 6.2, we propose our approach to learn the causal order among the observed variables and provide necessary and sufficient graphical conditions under which the number of latent variables is uniquely identifiable. In Section 6.3, we present a method to find the set of all possible causal effects which are consistent with the observational data and give conditions under which causal effects are uniquely identifiable. We conduct experiments to evaluate the performance of proposed solutions in Section 6.4 and conclude in Section 6.5.

The material in this chapter is taken from [110].

6.1 Problem Definition

6.1.1 Notations

The notation used in this chapter is different from the rest of the dissertation. We denote the variables of the system with $\mathcal{V} = \{V_1, ..., V_p\}$ (as opposed to $V = \{X_1, ..., X_p\}$). In a directed graph $G = (\mathcal{V}, E)$ with the vertex set $\mathcal{V}$ and the edge set E, we denote a directed edge from V_i to V_j by (V_i, V_j). A directed path $P = (V_{i_0}, V_{i_1}, \cdots, V_{i_k})$ in G is a sequence of vertices of G where there is a directed edge from V_{i_j} to $V_{i_{j+1}}$ for any $0 \leq j \leq k-1$. We define the set of variables $\{V_{i_1}, \cdots, V_{i_{k-1}}\}$ as the intermediate variables on the path P. We say that a path is a latent path if all the intermediate variables on the path are latent. We use notation $V_i \rightsquigarrow V_j$ to show that there exists a directed path from V_i to V_j. If there is a directed path from V_i to V_j, V_i is ancestor of V_j and that V_j is a descendant of V_i. More formally, $Anc(V_i) = \{V_j | V_j \rightsquigarrow V_i\}$ and $Des(V_i) = \{V_j | V_i \rightsquigarrow V_j\}$. Recall that each variable V_i is an ancestor and a descendant of itself.

We denote vectors and matrices by boldface letters. The vectors $\mathbf{A}_{i,:}$ and $\mathbf{A}_{:,i}$ represent i-th row and column of matrix $\mathbf{A}$, respectively. The (i, j) entry of matrix $\mathbf{A}$ is denoted by $[\mathbf{A}]_{i,j}$. For $n \times m$ matrix $\mathbf{A}$ and $n \times p$ matrix $\mathbf{B}$, the notation $[\mathbf{A}, \mathbf{B}]$ denotes the horizontal concatenation. For $n \times m$ matrix $\mathbf{A}$ and $p \times m$ matrix $\mathbf{B}$, the notation $[\mathbf{A}; \mathbf{B}]$ shows the vertical concatenation.

6.1.2 System Model

Consider a linear SCM among a set of variables $\mathcal{V} = \{V_1, \cdots, V_p\}$:

$$\mathbf{V} = \mathbf{A}\mathbf{V} + \mathbf{N}, \tag{6.1}$$

where the vectors $\mathbf{V}$ and $\mathbf{N}$ denote the random variables in $\mathcal{V}$ and their corresponding exogenous noises, respectively. Note that we use $\mathbf{A}$ to denote the weighted adjacency matrix as opposed to the notation B introduced in Section 2.2.1. The entry (i, j) of matrix $\mathbf{A}$ shows the strength of direct causal effect of variable V_j on variable V_i. We assume that the causal relations among random variables can be represented by a DAG. Thus, the variables in $\mathcal{V}$ can be arranged in a causal order, such that no latter variable causes any earlier variable. We denote such a causal order on the variables by k in which $k(i), i \in \{1, \cdots, p\}$ shows the position of variable V_i in the causal order. $\mathbf{A}$ can be converted to a strictly lower triangular

matrix by permuting its rows and columns simultaneously based on the causal order.

Example 8. *Consider the following linear SCM with four random variables* $\{V_1, \cdots, V_4\}$*:*

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} 0 & e & 0 & d \\ 0 & 0 & 0 & 0 \\ 0 & a & 0 & 0 \\ 0 & b & c & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} + \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{bmatrix},$$

where a, b, c, d *and* e *are some constants (see Figure 6.2). A causal order in this SCM model would be:* $k(1) = 4, k(2) = 1, k(3) = 2, k(4) = 3$*. Hence, matrix* $\mathbf{PAP}^T$ *is strictly lower triangular where* $\mathbf{P}$ *is a permutation matrix associated with* k *defined by the following non-zero entries:* $\{(k(i), i) | 1 \leq i \leq 4\}$*.*

We split random variables in $\mathbf{V}$ into an observed vector $\mathbf{V_o} \in \mathbb{R}^{p_o}$ and a latent vector $\mathbf{V_l} \in \mathbb{R}^{p_l}$ where p_o and p_l are the number of observed and latent variables, respectively. Without loss of generality, we assume that first p_o entries of $\mathbf{V}$ are observable, i.e. $\mathbf{V_o} = [V_1, \cdots, V_{p_o}]^T$ and $\mathbf{V_l} = [V_{p_o+1}, \cdots, V_p]^T$. Therefore,

$$\begin{bmatrix} \mathbf{V_o} \\ \mathbf{V_l} \end{bmatrix} = \begin{bmatrix} \mathbf{A_{oo}} & \mathbf{A_{ol}} \\ \mathbf{A_{lo}} & \mathbf{A_{ll}} \end{bmatrix} \begin{bmatrix} \mathbf{V_o} \\ \mathbf{V_l} \end{bmatrix} + \begin{bmatrix} \mathbf{N_o} \\ \mathbf{N_l} \end{bmatrix}, \tag{6.2}$$

where $\mathbf{N_o}$ and $\mathbf{N_l}$ are the vectors of exogenous noises of $\mathbf{V_o}$ and $\mathbf{V_l}$, respectively. Furthermore, we have: $\mathbf{A} = [\mathbf{A_{oo}}, \mathbf{A_{ol}}; \mathbf{A_{lo}}, \mathbf{A_{ll}}]$.

The causal order among all variables k induces a causal order k_o among the observed variables as follows: For any two observed variables V_i, V_j, $1 \leq i, j \leq p_o$, $k_o(i) < k_o(j)$ if $k(i) < k(j)$. Similarly, k induces a causal order among latent variables. We denote this causal order by k_l. It can be easily shown that $\mathbf{A_{oo}}$ and $\mathbf{A_{ll}}$ can be converted to strictly lower triangular matrices by permuting rows and columns simultaneously based on causal orders k_o and k_l, respectively.

Example 9. *In Example 8, suppose that only variables* V_1 *and* V_2 *are observable. Then, the causal order among observed variables would be:* $k_o(1) = 2$ *and* $k_o(2) = 1$*. Thus,* $\mathbf{PA_{oo}P}^T$ *is a strictly lower triangular matrix where* $\mathbf{P} = [0, 1; 1, 0]$*. For the latent variables,* $k_l(3) = 1$ *and* $k_l(4) = 2$*.*

In the remainder of this section, we briefly describe LiNGAM algorithm, which is capable of recovering the matrix $\mathbf{A}$ uniquely if all variables in the model are observable and exogenous

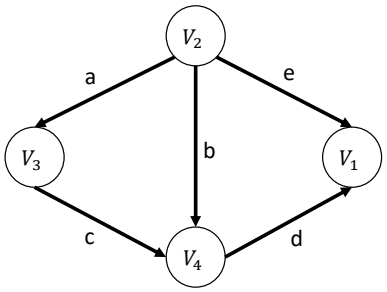

Figure 6.2: Causal graph of Example 8.

noises are non-Gaussian [6]. The vector $\mathbf{V}$ in Equation (6.1) can be written as a linear combination of exogenous noises as follows:

$$\mathbf{V} = \mathbf{B}\mathbf{N}, \tag{6.3}$$

where $\mathbf{B} = (\mathbf{I} - \mathbf{A})^{-1}$. The above equation fits into the standard linear Independent Component Analysis (ICA) framework, where independent non-Gaussian components are all variables in $\mathbf{N}$. By utilizing statistical techniques in ICA [99], matrix $\mathbf{B}$ can be identified up to scaling and permutations of its columns. More specifically, the independent components of ICA as well as the estimated $\mathbf{B}$ matrix are not uniquely determined because permuting and rescaling them does not change their mutual independence. So without knowledge of the ordering and scaling of the noise terms, the following general ICA model for $\mathbf{V}$ holds:

$$\mathbf{V} = \tilde{\mathbf{B}}\tilde{\mathbf{N}}, \tag{6.4}$$

where $\tilde{\mathbf{N}}$ contains independent components and these components (resp. the columns of $\tilde{\mathbf{B}}$) are a permuted and rescaled version of those in $\mathbf{N}$ (resp. the columns of $\mathbf{B}$). In what follows, we use $\mathbf{B}$ for matrix $\mathbf{B} = (\mathbf{I} - \mathbf{A})^{-1}$ while $\tilde{\mathbf{B}}$ is the mixing matrix for the ICA model, as given in (6.4). Hence $\tilde{\mathbf{B}}$ can be written as:

$$\tilde{\mathbf{B}} = \mathbf{B}\mathbf{P}\mathbf{\Lambda},$$

where $\mathbf{P}$ is a permutation matrix and $\mathbf{\Lambda}$ is a diagonal scaling matrix. Yet the corresponding causal model, represented by $\mathbf{A}$, can be uniquely identified because of its acyclicity constraint. In particular, the inverse of $\mathbf{B}$ can be converted uniquely to a lower triangular matrix having all-ones on its diagonal by some scaling and permutation of the rows.

6.2 Identifying Causal Orders among Observed Variables

Since the graph with adjacency matrix $\mathbf{A}$ is acyclic, there exists an integer d such that $\mathbf{A}^d = 0$. Thus, we can rewrite $\mathbf{B}$ in the following form:

$$\mathbf{B} = (\mathbf{I} - \mathbf{A})^{-1} = \sum_{k=0}^{d-1} \mathbf{A}^k. \tag{6.5}$$

It can be seen that there exists a causal path of length k from the exogenous noise of variable V_i to variable V_j if entry (j, i) of matrix $\mathbf{A}^k$ is nonzero. We define $[\mathbf{B}]_{j,i}$ as the total causal effect of variable V_i on variable V_j.

Assumption 9. *(Faithfulness assumption) The total causal effect from variable V_i to V_j is nonzero if there is a causal path from V_i to V_j. Thus, we have: $[\mathbf{B}]_{j,i} \neq 0$ if $V_i \rightsquigarrow V_j$.*

In the following lemma, we list two consequences of the faithfulness assumption that are immediate from the definition.

Lemma 9. *Under the faithfulness assumptions, for any two observed variables V_i and V_j, $1 \leq i, j \leq p_o$, the following holds:*
(i) Suppose that $V_i \rightsquigarrow V_j$. If $[\mathbf{B}]_{i,k} \neq 0$ for some $k \neq j$, then $[\mathbf{B}]_{j,k} \neq 0$.
(ii) If there is no causal path between V_i and V_j, then $[\mathbf{B}]_{i,j} = 0$ and $[\mathbf{B}]_{j,i} = 0$.

Based on Equation (6.2), we can write $\mathbf{V_o}$ in terms of $\mathbf{N_o}$ and $\mathbf{N_l}$ as follows.

$$\mathbf{V_o} = (\mathbf{I} - \mathbf{D})^{-1}\mathbf{N_o} + (\mathbf{I} - \mathbf{D})^{-1}\mathbf{A_{ol}}(\mathbf{I} - \mathbf{A_{ll}})^{-1}\mathbf{N_l}, \tag{6.6}$$

where $\mathbf{D} = \mathbf{A_{oo}} + \mathbf{A_{ol}}(\mathbf{I} - \mathbf{A_{ll}})^{-1}\mathbf{A_{lo}}$. Let $\mathbf{B_o} := (\mathbf{I} - \mathbf{D})^{-1}$, $\mathbf{B_l} := (\mathbf{I} - \mathbf{D})^{-1}\mathbf{A_{ol}}(\mathbf{I} - \mathbf{A_{ll}})^{-1}$, and $\mathbf{N} := [\mathbf{N_o}; \mathbf{N_l}]$. Thus, $\mathbf{V_o} = \mathbf{B'N}$ where $\mathbf{B'} := [\mathbf{B_o}, \mathbf{B_l}]$. This equation fits into a linear over-complete ICA where the exogenous noises are non-Gaussian and the number of observed variables is less than the number of variables in the system. The following proposition asserts when the columns of matrix $\mathbf{B'}$ are still identifiable up to permutations and scaling.

Definition 24. *(Reducibility of a matrix) A matrix is reducible if two of its columns are linearly dependent.*

Proposition 15. *([111], Theorem 3) In the linear over-completer ICA problem, the columns of mixing matrix can be identified up to some scaling and permutation if it is not reducible.*

Lemma 10. *The columns of* $\mathbf{B}'$ *corresponding to any two observed variables are linearly independent.*

Although columns of $\mathbf{B}'$ corresponding to the observed variables are pairwise linearly independent, a column corresponding to a latent variable V_i might be linearly dependent on a column corresponding to an observed or latent variable V_j (see Example 10). In that case, we can remove the column $[\mathbf{B}']_{:,i}$ and N_i from matrix $\mathbf{B}'$ and vector $\mathbf{N}$, respectively and replace N_j by $N_j + \alpha N_i$ where α is a constant such that $[\mathbf{B}']_{:,i} = \alpha[\mathbf{B}']_{:,j}$. We can continue this process until all the remaining columns are pairwise linearly independent. Let $\mathbf{B}''$ and $\mathbf{N}''$ be the resulting mixing matrix and exogenous noise vector, respectively. According to Lemma 10, all the columns of $\mathbf{B}'$ corresponding to observed variables are in $\mathbf{B}''$. We utilize matrix $\mathbf{B}''$ to recover a causal order among the observed variables.

Since matrix $\mathbf{B}''$ is not reducible, its column can be identified up to some scaling and permutation according to Proposition 15. Let $\tilde{\mathbf{B}}''$ be the recovered matrix containing columns of $\mathbf{B}''$. Consider any two observed variables V_i and V_j, i.e., $1 \leq i, j \leq p_o$. We extract two rows of $\tilde{\mathbf{B}}''$ corresponding to variables V_i and V_j. Let n_{0*} be the number of columns in $[\tilde{\mathbf{B}}''_{i,:}; \tilde{\mathbf{B}}''_{j,:}]$ whose first entries are zero but second entries are nonzero. Similarly, let n_{*0} be the number of columns that their first entries are nonzero but their second entries are zero. The following lemma asserts that the existence of a causal path between V_i and V_j can be checked from n_{0*} and n_{*0} (or equivalently, $\tilde{\mathbf{B}}''$).

Lemma 11. *Under the faithfulness assumption, the existence of a causal path between any two observed variable can be inferred from matrix* $\tilde{\mathbf{B}}''$.

We can construct an auxiliary directed graph whose vertices are the observed variables and a directed edge exists from V_i to V_j if $V_i \rightsquigarrow V_j$ (which we can infer from n_{*0} and n_{0*}). Any causal order over the auxiliary graph is a correct causal order among the observed variables $\mathbf{V_o}$.

Example 10. *Consider the causal graph in Figure 6.3. Suppose that variables* V_3 *and* V_4 *are latent.* $\mathbf{B}'$ *would be:*

$$\begin{bmatrix} 1 & 0 & 0 & a \\ d & 1 & e & c + ad + be \end{bmatrix}.$$

We can remove the third column from $\mathbf{B}'$ *and update the vector* $\mathbf{N}$ *to* $[N_1; N_2 + eN_3; N_4]$. *Thus, matrix* $\mathbf{B}''$ *is equal to:*

$$\begin{bmatrix} 1 & 0 & a \\ d & 1 & c + ad + be \end{bmatrix},$$

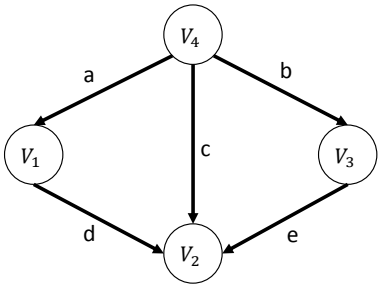

Figure 6.3: Causal graph of Example 10.

which is not reducible. Without loss of generality, assume that the recovered matrix $\tilde{\mathbf{B}}''$ *is equal to* $\mathbf{B}''$*. Therefore,* $n_{0*} = 1$ *and* $n_{*0} = 0$*. Hence, we can infer that there is a causal path from* V_1 *to* V_2*.*

Recovering the Number of Variables in the System

According to Proposition 15, the number of variables in the system can be recovered if and only if matrix $\mathbf{B}'$ is not reducible. Furthermore, Equation (6.6) implies that matrix $\mathbf{B}'$ is not reducible if and only if the columns of matrix $[\mathbf{I}_{p_o \times p_o}, \mathbf{A_{ol}}(\mathbf{I} - \mathbf{A_{ll}})^{-1}]$ are not linearly independent. In the rest of this section, we will present equivalent necessary and sufficient graphical conditions under which the number of variables in the systems can be uniquely identified. But before that, we present a simple example where $[\mathbf{I}_{p_o \times p_o}, \mathbf{A_{ol}}(\mathbf{I} - \mathbf{A_{ll}})^{-1}]$ is reducible and give a graphical interpretation of it.

Example 11. *Consider a linear SCM with three variables* V_1, V_2*, and* V_3 *where* $V_3 = N_3$*,* $V_1 = \alpha V_3 + N_1$*, and* $V_2 = \beta V_1 + N_2$*. Thus, the corresponding causal graph would be:* $V_3 \to V_1 \to V_2$*. Suppose that* V_3 *is the only latent variable. Hence,* $\mathbf{A_{ll}} = 0$*,* $\mathbf{A_{ol}} = [\alpha; 0]$*, and* $\mathbf{A_{ol}}(\mathbf{I} - \mathbf{A_{ll}})^{-1} = [\alpha; 0]$ *which is linearly dependent on the first column of* $\mathbf{I}$*. In fact, latent variable* V_3 *can be absorbed in variable* V_1 *by changing the exogenous noise of* V_1 *from* N_1 *to* $N_1 + \alpha N_3$*. Thus, the number of variables in this model cannot be identified uniquely in this model.*

Definition 25. *(Absorbing) Variable* V_i *is said to be absorbed in variable* V_j *if the exogenous noise of* V_i *is set to zero* $N_i \leftarrow 0$*, and the exogenous noise of* V_j *is replaced by* $N_j \leftarrow N_j + [\mathbf{B}]_{j,i} N_i$*. We define absorbing a variable in* $\emptyset$ *by setting its exogenous noise to zero.*

Definition 26. *(Absorbablity) Let* P'_{V_o} *be the joint distribution of the observed variables after absorbing* V_i *in* V_j*. We say* V_i *is absorbable in* V_j *if* $P'_{V_o} = P_{V_o}$*.*

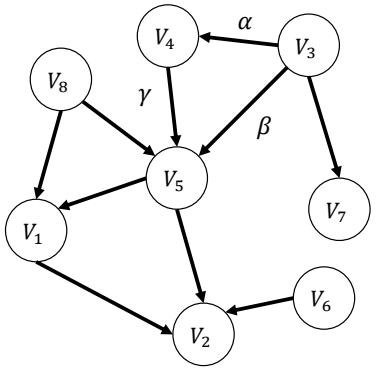

Figure 6.4: Causal graph of Example 12. V_1 and V_2 are the only observed variables.

The following theorem characterizes the graphical conditions where a latent variable is absorbable.

Theorem 15.
(a) A latent variable is absorbable in $\emptyset$ if and only if it has no observable descendant.
(b) A latent variable V_j is absorbable in variable V_i (observed or latent), if and only if all paths from V_j to its observable descendants go through V_i.

Example 12. *Consider a linear SCM with corresponding causal graph in Figure 6.4 where V_1 and V_2 are the only observed variables. V_7 satisfies condition (a) and its exogenous noise can be set to zero. Furthermore, V_3 and V_4 satisfy condition (b) with respect to V_5 and they can be absorbed in V_5 by setting the exogenous noise of V_5 to $N_5+(\alpha\gamma+\beta)N_3+\gamma N_4$. Finally, V_6 satisfies condition (b) and it can be absorbed in V_2. Note that V_8 and V_5 cannot be absorbed in V_1 or V_2.*

Definition 27. *We say a causal graph is minimal if none of its variables are absorbable.*

Based on above definition, a causal graph is minimal if none of the latent variables satisfy the conditions in Theorem 15. We borrowed the terminology of minimal causal graphs from [94] for polytree causal structures. In [94], a causal graph is called minimal if it has no redundant latent variables in the sense that the joint distribution without latent variables remains a connected tree. Later, [112] showed that in minimal latent directed information polytrees, each node has at least two children. The following lemma asserts that the same argument holds true for the non-absorbable latent variables in our setting.

Lemma 12. *A latent variable is non-absorbable if it has at least two non-absorbable children.*

The next theorem gives necessary and sufficient graphical conditions for non-reducibility of matrix $\mathbf{B}'$.

Theorem 16. *$\mathbf{B}'$ is not reducible almost surely if and only if the corresponding causal graph G is minimal.*

Corollary 4. *Under faithfulness assumption and non-Gaussianity of exogenous noises, the number of variables in the system is identifiable almost surely if the corresponding graph is minimal.*

6.3 Identifying Total Causal Effects among Observed Variables

In this section, first, we will show by an example that total causal effects among observed variables cannot be identified uniquely under the faithfulness assumption and non-Gaussianity of exogenous noises.[3] However, we can obtain all the possible solutions. Furthermore, under some additional assumptions on linear SCM, we show that one can uniquely identify total causal effects among observed variables.

6.3.1 Example of non-Uniqueness of Total Causal Effects

Consider the causal graph in Figure 6.5 where V_i and V_j are observed variables and V_k is a latent variable. The direct causal effects from V_k to V_i, from V_k to V_j, and from V_i to V_j are α, γ, and β, respectively. We can write V_i and V_j based on the exogenous noises of their ancestors as follows:

$$\begin{aligned} V_i &= \alpha N_k + N_i, \\ V_j &= \beta N_i + (\alpha\beta + \gamma) N_k + N_j. \end{aligned} \tag{6.7}$$

Now, we construct a second causal graph depicted in Figure 6.5 where the exogenous noises of variables V_i and V_k are changed to αN_k and N_i, respectively. Furthermore, we set the direct causal effects from V_k to V_i, from V_k to V_j, and from V_i to V_j to 1, $-\gamma/\alpha$, and $\beta + (\gamma/\alpha)$, respectively. It can be seen that equations in (6.7) do not change while the direct causal effect from V_i to V_j becomes $\beta + (\gamma/\alpha)$ in the second causal graph. Thus, we cannot identify causal effect from V_i to V_j merely by observational data from V_i and V_j. In Appendix

[3] This example has also been studied in [98].

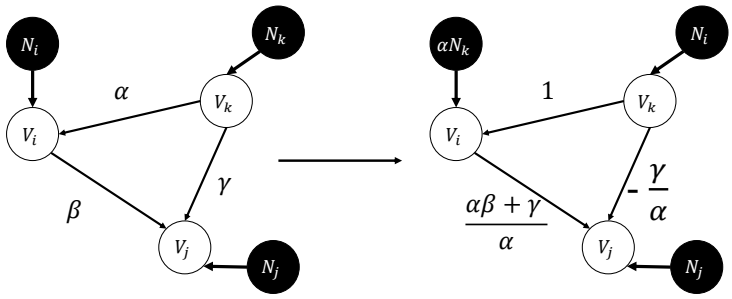

Figure 6.5: An example of non-identifiability of causal effects from observed variable V_i to observed variable V_j.

D, we extend this example to the case where there might be multiple latent variables on the path from V_k to V_i and V_j, and from V_i to V_j.

The above example shows that causal effects may not be identified even by assuming non-Gaussianity of exogenous noises if we have some latent variables in the system. In the following, we first show that the set of all possible total causal effects can be identified. Afterwards, we will present a set of structural conditions under which we can uniquely identify total causal effects among observed variables.

6.3.2 Identifying the Set of All Possible Total Causal Effects

Since the subgraph corresponding to $\mathbf{A_{ll}}$ is a DAG, there exists an integer d_l such that ${\mathbf{A_{ll}}}^{d_l} = 0$. Hence, we can rewrite matrix $\mathbf{D}$ given in (6.6) as follows.

$$\mathbf{D} = \mathbf{A_{oo}} + \sum_{k=0}^{d_l-1} \mathbf{A_{ol}}{\mathbf{A_{ll}}}^k \mathbf{A_{lo}}. \tag{6.8}$$

Lemma 13. *Matrix* $\mathbf{D}$ *in* (6.6) *can be converted to a strictly lower triangular matrix by permuting columns and rows simultaneously based on the causal order* k_o.

Previously, we showed that existence of a causal path between any two observed variables V_i and V_j can be determined by performing over-complete ICA. Let $des_o(V_i)$ be the set of all observed descendants of V_i, i.e., $des_o(V_i) = \{V_j | V_i \rightsquigarrow V_j, 1 \leq j \leq p_o\}$. We will utilize $des_o(V_i)$'s to enumerate all possible total causal effects among the observed variables.

Remark 9. *From Lemma 10, we have:* $des_o(V_i) \neq des_o(V_j)$ *for any* $1 \leq i, j \leq p_o$.

As we discussed in Section 6.2, under non-Gaussianity of exogenous noises, the columns of $\mathbf{B}''$ can be determined up to some scalings and permutations by solving an overcomplete ICA problem. Let p_r be the number of columns of $\mathbf{B}''$. Furthermore, without loss of generality, assume that variables $V_{p_o+1}, V_{p_o+2}, \cdots, V_{p_r}$ are the latent variables in the system whose corresponding columns remain in $\mathbf{B}''$.

Theorem 17. *Let $r_i := |\{j : des_o(V_i) = des_o(V_j), 1 \leq j \leq p_r\}|$, for any $1 \leq i \leq p_o$. Under the assumptions of faithfulness and non-Gaussianity of exogenous noises, the number of all possible $\mathbf{D}$'s that can generate the same distribution for $\mathbf{V_o}$ according to* (6.2)*, is equal to $\Pi_{i=1}^{p_o} r_i$.*

Comparing our results with [98], we can obtain all sets $des_o(V_i)$'s and determine which columns can be selected as corresponding columns of observed variables in $O(p_o^2 p_r)$ and then enumerate all the possible total causal effects while the proposed algorithm in [98] requires to search a space of $\binom{p_r}{p_o}$ different possible choices. Moreover, we can identify a causal order uniquely with the same time complexity by utilizing the method proposed in Section 6.2.

6.3.3 Unique Identification of Causal Effects under Structural Conditions

Based on Theorem 17, in this part, we propose a method to identify total causal effects uniquely under some structural conditions.

Assumption 10. *Assume that for any observed variables V_i and any latent variable V_k, we have: $des_o(V_k) \neq des_o(V_i)$.*

Assumption 10 is a very natural condition that one expects to hold for unique identifiability of causal effects. This is because if Assumption 10 fails, then based on Theorem 17, there are multiple sets of total causal effects that are compatible with the observed data.

Theorem 18. *Under Assumptions 9-10, and non-Gaussianity of exogenous noises, the total causal effect between any two observed variables can be identified uniquely.*

The description of the proposed solution in Theorem 18 is given in Algorithm 11. It is noteworthy that the example in Section 6.3.1 (given in Figure 6.5) violates the conditions in Theorem 18 since $des_o(V_k) = des_o(V_i)$. We have shown for this example that the causal effect from V_i to V_j cannot be identified uniquely.

Algorithm 11

Input: Collection of the sets $des_o(V_i), 1 \leq i \leq p_o$.
Run an over-complete ICA algorithm over observed variables $\mathbf{V_o}$ and obtain matrix $\tilde{\mathbf{B}}''$.
for $i = 1 : p_r$ **do**
    $I_i = \{k | [\tilde{\mathbf{B}}''_{:,i}]_k \neq 0\}$
    for $j = 1 : p_o$ **do**
        if $I_i = des_o(V_j)$ **then**
            $[\hat{\mathbf{B}}_\mathbf{o}]_{:,j} = \tilde{\mathbf{B}}''_{:,i} / [\tilde{\mathbf{B}}''_{:,i}]_j$
        end if
    end for
end for
Output: $\hat{\mathbf{B}}_\mathbf{o}$

6.4 Experiments

In this section, we first evaluate the performance of the proposed method in recovering causal orders from synthetic data, generated according to the causal graph in Figure 6.1. Our experiments show that the proposed method returns a correct causal order while, as we mentioned in Introduction section, previous methods proposed for linear non-Gaussian SCM with latent variables, might require additional assumptions in order to recover causal relations. More specifically, they do not have theoretical guarantee to recover the causal order or checking the existence of causal paths in our setting. Nevertheless, we evaluated the performances of lvLiNGAM [98], Pairwise lvLiNGAM [100], ParceLiNGAM [102], ICA-LiNGAM [6], Direct-LiNGAM [64] and FCI algorithm [1]. We also consider another causal graph which satisfies Assumption 10 and demonstrate that the proposed method can return the correct causal effects. Next, we evaluate the performance of the proposed method for different number of variables in the system. Afterwards, for real data, we consider the daily closing prices of four world stock indices and check the existence of causal paths between any two indices. The results are compatible with common beliefs in economy.

6.4.1 Synthetic data

First, for the causal graph in Figure 6.1, we generated 1000 samples of observed variables V_1 and V_2 where nonzero entries of matrix $\mathbf{A}$ is equal to 0.9. We utilized the Reconstruction ICA (RICA) algorithm [113] to solve the over-complete ICA problem as follows: Let $\mathbf{v_o}$ be a $p_o \times n$ matrix containing observational data where $[v_o]_{i,j}$ is j-th sample of variable V_i and

n is the number of samples. First, the sample covariance matrix of $\mathbf{v_o}$ is eigen-decomposed, i.e., $1/(n-1)(\mathbf{v_o}-\bar{\mathbf{v}}_\mathbf{o})(\mathbf{v_o}-\bar{\mathbf{v}}_\mathbf{o})^T = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^T$ where $\mathbf{U}$ is the orthogonal matrix, $\mathbf{\Sigma}$ is a diagonal matrix, and $\bar{\mathbf{v}}_\mathbf{o}$ is the sample mean vector. Then, the observed data is pre-whitened as follows: $\mathbf{w} = \mathbf{\Sigma}^{-1/2}\mathbf{U}(\mathbf{v_o}-\bar{\mathbf{v}}_\mathbf{o})$. The RICA algorithm tries to find matrix $\mathbf{Z}$ that is the minimizer of the following objective function:

$$\underset{\mathbf{Z}}{\text{minimize}} \sum_{i=1}^{n}\sum_{j=1}^{p_r} g(\mathbf{Z}_{:,j}^T\mathbf{w}_{:,i}) + \frac{\lambda}{n}\sum_{i=1}^{n}\|\mathbf{Z}\mathbf{Z}^T\mathbf{w}_{:,i}-\mathbf{w}_{:,i}\|_2^2,$$

where parameter λ controls the cost of penalty term. We estimated matrix $\tilde{\mathbf{B}}''$ by $\mathbf{U}\mathbf{\Sigma}^{1/2}\mathbf{Z}^*$ where $\mathbf{Z}^*$ is the optimal solution of the above optimization problem.

In order to estimate the number of columns of $\tilde{\mathbf{B}}''$, we held out 250 of samples for model selection. More specifically, we solved the over-complete ICA problem for different number of columns, evaluated the fitness of each model by computing the objective function of RICA over the hold-out set, and selected the model with minimum cost. In order to check whether an entry is equal to zero, we used the bootstrapping method [114], which generates 10 bootstrap samples by sampling with replacement from training data. For each bootstrap sample, we executed RICA algorithm to obtain an estimation of $\tilde{\mathbf{B}}''$. Since in each estimation, columns are in arbitrary permutation, we need to match similar columns in estimations of $\tilde{\mathbf{B}}''$. To do so, in each estimation, we divided all entries of a column by the entry with the maximum absolute value in that column. Then, we picked each column from the estimated mixing matrix, computed its l_2 distance from each column of another estimated mixing matrix, and matched to the one with a minimum distance. Afterwards, we used a t-test with confidence level of 95% to check whether an entry is equal to zero from the bootstrap samples. An estimation of $\tilde{\mathbf{B}}''$ from a bootstrap sample is given as follows:

$$\begin{bmatrix} -0.0272 & 0.5238 & 1 \\ 1 & 1 & 0.8579 \end{bmatrix}.$$

Moreover, experimental results showed the correct support of $\tilde{\mathbf{B}}''$, i.e., $[0,1,1;1,1,1]$ can be recovered with merely 10 bootstrap samples. Thus, there is a causal path from V_1 to V_2. Furthermore, for the causal graph $V_1 \leftarrow V_3 \rightarrow V_2$ in which V_3 is only the latent variable, we repeated the same procedure explained above. An estimation of $\tilde{\mathbf{B}}''$ from one of the bootstrap samples is given as follows:

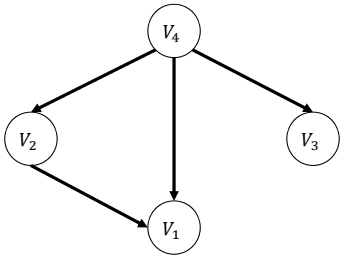

Figure 6.6: An example of causal graphs satisfying structural conditions.

$$\begin{bmatrix} 1 & -0.046 & 0.9838 \\ -0.031 & 1 & 1 \end{bmatrix}.$$

From experiments, the estimated support of $\tilde{\mathbf{B}}''$ from bootstrap samples was: $[0, 1, 1; 1, 0, 1]$. Thus, we can conclude that there is no causal path between V_1 and V_2. Next, we considered the causal graph in Figure 6.6 where V_4 is the only latent variable. The direct causal effects of all directed edges are equal to 0.9. An estimation of $\tilde{\mathbf{B}}''$ from one of the bootstrap samples is given as follows:

$$\begin{bmatrix} -0.049 & 0.892 & 1 & 1 \\ -0.024 & 1 & 0.523 & -0.042 \\ 1 & -0.02 & 0.527 & -0.032 \end{bmatrix}.$$

Thus, we can deduce that there is only a causal path from V_2 to V_1. We can also estimate total causal effects between observed variables since this causal graph satisfies Assumption 10. The output of Algorithm 11 is:

$$\begin{bmatrix} 1 & 0.892 & -0.049 \\ -0.042 & 1 & -0.024 \\ -0.032 & -0.02 & 1 \end{bmatrix},$$

which is close to the true causal effects. We evaluated previous methods for learning the causal graphs in Figure 6.1, Figure 6.6, and the causal graph $V_1 \leftarrow V_3 \rightarrow V_2$. Table 6.1 shows whether each of them can find all causal paths correctly. It can be seen that only the proposed algorithm is successful in recovering the causal paths in all considered causal graphs.

We generated 1000 DAGs of size p by first selecting a causal order among variables randomly and then connecting each pair of variables with probability $c/(p-1)$, where c is the

Table 6.1: Comparison of methods in recovering causal paths for the causal graphs in Figure 6.1, Figure 6.6, and the causal graph $V_1 \leftarrow V_3 \rightarrow V_2$.

	Figure 6.1	Figure 6.6	$V_1 \leftarrow V_3 \rightarrow V_2$
lvLiNGAM [98]	✓	×	✓
Pairwise lvLiNGAM [100]	×	×	✓
ParceLiNGAM [102]	×	×	×
ICA-LiNGAM [6]	✓	×	×
Direct-LiNGAM [64]	✓	×	×
FCI [1]	×	×	×
Proposed algorithm	✓	✓	✓

Table 6.2: Running time (in seconds) of Algorithm 11 for differnet number of variables in the system and different graph densities $c = 2, 3$.

p	10	15	20	25	30
$c = 2$	0.7	1.41	1.66	3.09	3.48
$c = 3$	0.76	1.48	1.75	3.33	3.84

average degree of each node. We generated data from a linear SCM where nonzero entries of matrix $\mathbf{A}$ were drawn uniformly from the range $[-0.9, -0.5] \cup [0.5, 0.9]$ and the exogenous noises followed a uniform distribution. In the remainder of this part, we assume that the number of latent variable is known. We first evaluated the running time of Algorithm 11 and compared it with the proposed algorithm in [98], which can provide all possible total causal effects. In the experiments, we selected $p_l = p/2$ variables randomly as latent variables. The running time of Algorithm 11 is given in Table 6.2 for $c = 2, 3$. In our experiments, the algorithm in [98] did not return any output in 10 minutes and it is only feasible on small graphs with fewer than six variables.

We evaluated the performance of the proposed algorithm and compared it with the previous ones, including Pairwise lvLiNGAM [100], ParceLiNGAM [102], LiNGAM [6], and Direct-LiNGAM [64], in the presence of latent variables. More specifically, we define precision of an algorithm as the fraction of correctly recovered causal paths among recovered causal paths between any two observed variables. We also define its recall as the fraction of recovered causal paths among actual causal paths between any two observed variables. Figure 6.7 shows precisions and recalls of the mentioned algorithms for different number of variables $p = 10, 15, 20$, different number of observed variables, and different average degrees $c = 4, 7$. One can see that none of the algorithms has the best performance in all

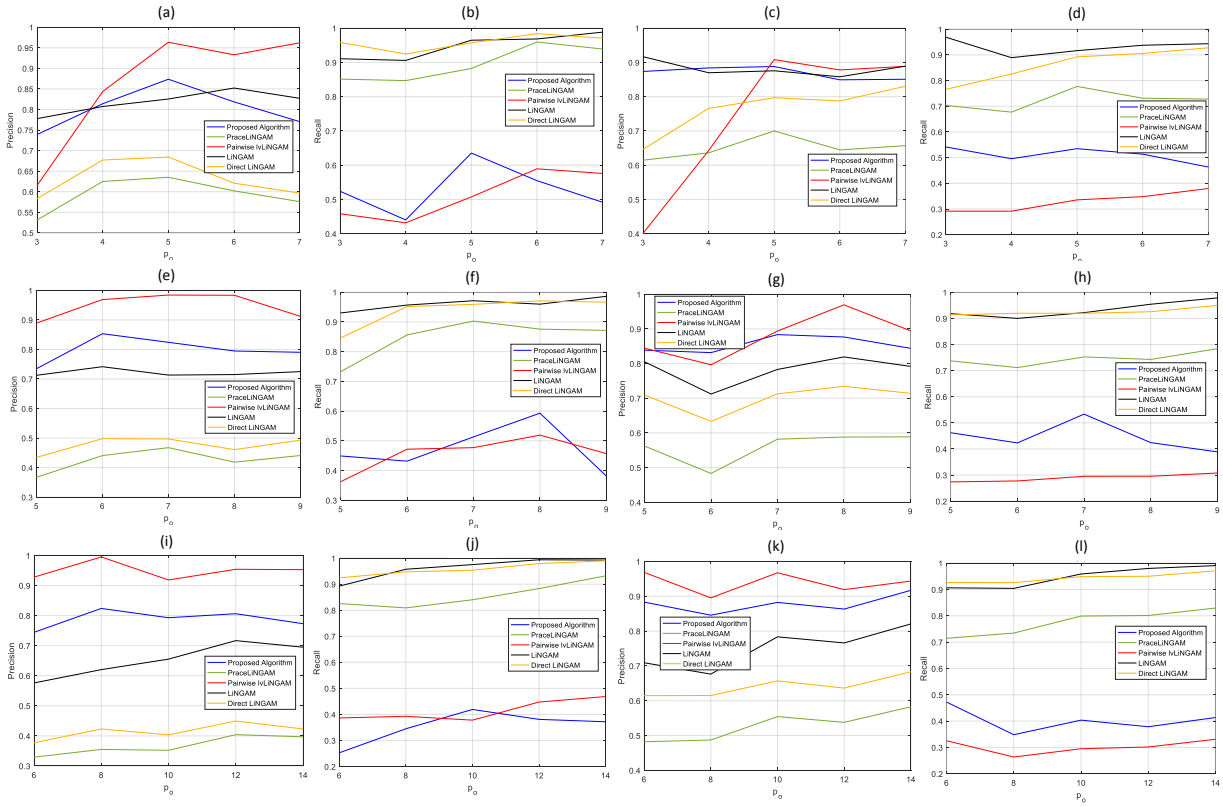

Figure 6.7: Precisions/Recalls of Pairwise lvLiNGAM [100], ParceLiNGAM [102], ICA-LiNGAM [6], Direct-LiNGAM [64] and the proposed algorithm in the presence of latent variables: (a) Precisions for $p = 10$, $c = 4$, (b) Recalls for $p = 10$, $c = 4$, (c) Precisions for $p = 10$, $c = 7$, (d) Recalls for $p = 10$, $c = 7$, (e) Precisions for $p = 15$, $c = 4$, (f) Recalls for $p = 15$, $c = 4$, (g) Precisions for $p = 15$, $c = 7$, (h) Recalls for $p = 15$, $c = 7$, (i) Precisions for $p = 20$, $c = 4$, (j) Recalls for $p = 20$, $c = 4$, (k) Precisions for $p = 20$, $c = 7$, (l) Recalls for $p = 20$, $c = 7$.

settings. However, the proposed algorithm and Pairwise lvLiNGAM [100] are the top two algorithms in terms of precision. Moreover, LiNGAM [6] and Direct-LiNGAM [64] have the best performance in terms of recall.

6.4.2 Real data

We considered the daily closing prices of the following world stock indices from 10/12/2012 to 10/12/2018, obtained from Yahoo financial database: Dow Jones Industrial Average (DJI) in USA, Nikkei 225 (N225) in Japan, Euronext 100 (N100) in Europe, Hang Seng Index (HSI) in Hong Kong, and the Shanghai Stock Exchange Composite Index (SSEC) in China.

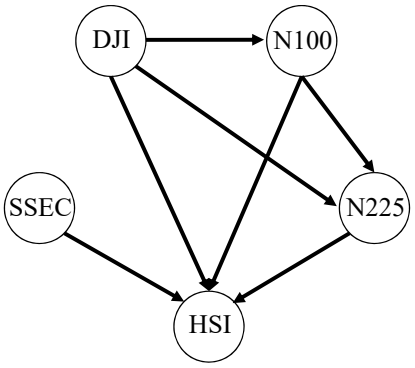

Figure 6.8: The causal relationships among five world stock indices obtained from the proposed method in Section 6.2.

Let $c_i(t)$ be the closing price of i-th index on day t. We define the corresponding return by $R_i(t) := (c_i(t) - c_{i-1}(t))/c_{i-1}(t)$. We considered the returns of indices as an observational data and applied the proposed method in Section 6.2 in order to check the existence of a causal path between any two indices. Figure 6.8 depicts the causal relationships among the indices. In this figure, there is a directed edge from index i to index j if we find a causal path from i to j. As can be seen, there are causal paths from DJI to HSI, N225, and N100 which is commonly known to be true in the stock market [115]. Furthermore, HSI is influenced by all other indices and SSEC only affects HSI which these findings are compatible with the previous results in [115].

6.5 Conclusion

We considered the problem of learning causal models from observational data in linear non-Gaussian acyclic models with latent variables. Under the faithfulness assumption, we proposed a method to check whether there exists a causal path between any two observed variables. Moreover, we gave necessary and sufficient graphical conditions to uniquely identify the number of variables in the system. From the information about the existence of a directed path, we could obtain a causal order among the observed variables. Additionally, we considered the problem of estimating total causal effects. We showed by an example that causal effects among observed variables cannot be identified uniquely even under the assumptions of faithfulness and non-Gaussianity of exogenous noises. However, we can identify all possible set of total causal effects that are compatible with the observational data

efficiently in time. Furthermore, we presented structural conditions under which we can learn total causal effects among observed variables uniquely. Experiments on synthetic data and real-world data showed the effectiveness of our proposed algorithms on learning causal models. One of our future research directions is to extend the results to the case of cyclic linear SCMs. We believe that methods similar to the one proposed can recover some of the causal paths in the system. Another direction of future work entails developing causal structure learning algorithms for nonlinear SCM with latent variables by exploiting recent progress in non-linear ICA. In addition, it is desirable to develop a principled, efficient approach to selecting the optimal number of latent variables.

APPENDIX A

APPENDIX OF CHAPTER 3

A.1 Example of Comparison with the Influence Maximization Problem

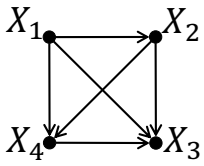

Figure A.1: Example of comparison with the influence maximization problem.

Suppose $k = 1$. Figure A.1 depicts a graph for which the optimal solution to the influence maximization problem is different from the optimal solution to the budgeted experiment design problems. Clearly, influencing vertex X_1 leads to influencing all the vertices in the graph, and hence, this vertex is the solution to the influence maximization problem. But, intervening on X_1 leads to discovering the orientation of only 3 edges, while intervening on, say X_2, leads to discovering the orientation of 5 edges.

A.2 Proof of Lemma 3

From the passive observational stage, the set of all edges incident with X_i is known. Suppose X_j is adjacent with X_i with unknown edge direction. If this edge in the ground truth structure has direction $X_i \to X_j$, then in the interventional distribution, there exists a subset of vertices X_S containing X_i, for which $W_i \perp X_j | X_S$, where W_i is the intervention variable corresponding to the singleton intervention on X_i. On the other hand, if this edge

in the ground truth structure has direction $X_i \leftarrow X_j$, then in the interventional distribution, for all subsets of vertices X_S containing X_i, we have $W_i \not\perp X_j | X_S$.

The proof above works for both cases of hard and soft interventions. [27] provided an alternative proof for the case of hard interventions, and [18] provided alternative proofs for both cases of soft and hard interventions.

A.3 Proof of Lemma 5

Suppose the root vertex is X. Since $\tilde{T}$ is a tree, there is a unique path from X to every other vertex. For every vertex with path length 1 from the root, i.e., every vertex adjacent to the root, by definition, the edge is from X to that vertex. For every vertex X_j with path length 2 from the root, we have the induced subgraph $X \rightarrow X_i - X_j$, and hence, since there cannot be any v-structures in the graph, the edge $X_i - X_j$ should be oriented as $X_i \rightarrow X_j$. As the induction hypothesis, assume that for every vertex X_i with path length m from the root, we have the induced subgraph $X \rightarrow \cdots \rightarrow X_i$. Now for every vertex X_j with path length $m+1$ from the root, we have the induced subgraph $X \rightarrow \cdots \rightarrow X_i - X_j$. Again, since there cannot be any v-structures in the graph, the edge $X_i - X_j$ should be oriented as $X_i \rightarrow X_j$. Therefore, the location of the root variable identifies the direction of all the edges.

A.4 Proof of Lemma 6

We use the following lemma for the proof.

Lemma 14. *For a tree UCEG $\tilde{T}$ on variable set V, an intervention on a variable $X_k \in V$ only determines the direction of all the edges incident to $Desc(X_k)$, where descendants of a variable are defined with respect to the ground truth directed tree.*

Proof. By Lemma 3, an intervention on X_k identifies the direction of all edges incident to X_k. Since $\tilde{T}$ is a tree, there is a unique path from X to every other vertex. For every vertex for which the path from X_k to that vertex goes through a child of X_k, similar to Lemma 5, the direction of incident edges to that vertex will be identified. Therefore, we learn the direction of all the edges incident to $Desc(X_k)$. Now, suppose X_i is a parent of X_k. Therefore, for every vertex X_j adjacent to X_i, we have the induced subgraph $X_j - X_i \rightarrow X_k$. Hence the edge $X_j - X_i$ can have either of the directions without creating a v-structure, and hence,

the direction of such edge cannot be identified. Therefore, the direction of any of the edges incident to X_j cannot be identified either. Consequently, we do not learn the direction of all any of the edges incident to *Non-Desc*(X_k). □

Suppose the ground truth directed tree is T_r^X. By Lemma 14, after an experiment with target set $\mathcal{I}_r$, the edges whose directions are remained unresolved are those which are incident only to $\cap_{X_k \in \mathcal{I}_r} \textit{Non-Desc}(X_k)$, which are the edges of the component $C_j(\mathcal{I}_r)$, where $X \in C_j(\mathcal{I}_r)$. Noting that the size of a tree of order p is $p-1$ concludes that the number of unresolved edges are $|C_j(\mathcal{I}_r)| - 1$. If $X \in \mathcal{I}_r$, then $\cap_{X_k \in \mathcal{I}_r} \textit{Non-Desc}(X_k) = \emptyset$, i.e., the direction of all the edges are identified and the gain will be $D(\mathcal{I}_r, T_r^X) = |\tilde{T}_r| - 1$. Otherwise the gain will be $D(\mathcal{I}_r, T_r^X) = |\tilde{T}_r| - 1 - |C_j(\mathcal{I}_r)| + 1 = |\tilde{T}_r| - |C_j(\mathcal{I}_r)|$.

A.5 Proof of Proposition 1

We can write the average gain $\mathcal{D}(\mathcal{I})$ as follows:

$$
\begin{aligned}
\mathcal{D}(\mathcal{I}) &= \frac{1}{p_u} \sum_{r=1}^{R} \sum_{X \in V(\tilde{T}_r)} D(\mathcal{I}_r, T_r^X) \\
&\overset{(a)}{=} \frac{1}{p_u} \sum_{r=1}^{R} \sum_{X \in \mathcal{I}_r \cap V(\tilde{T}_r)} (|\tilde{T}_r| - 1) + \frac{1}{p_u} \sum_{r=1}^{R} \sum_{j=1}^{J(\mathcal{I}_r)} \sum_{X \in C_j(\mathcal{I}_r)} |\tilde{T}_r| - |C_j(\mathcal{I}_r)| \\
&= \frac{1}{p_u} \sum_{r=1}^{R} |\mathcal{I}_r|(|\tilde{T}_r| - 1) + \frac{1}{p_u} \sum_{r=1}^{R} \sum_{j=1}^{J(\mathcal{I}_r)} |\tilde{T}_r||C_j(\mathcal{I}_r)| - |C_j(\mathcal{I}_r)|^2 \\
&\overset{(b)}{=} \frac{1}{p_u} \sum_{r=1}^{R} |\mathcal{I}_r|(|\tilde{T}_r| - 1) + \frac{1}{p_u} \sum_{r=1}^{R} |\tilde{T}_r|(|\tilde{T}_r| - |\mathcal{I}_r|) - \frac{1}{p_u} \sum_{r=1}^{R} \sum_{j=1}^{J(\mathcal{I}_r)} |C_j(\mathcal{I}_r)|^2 \\
&= \frac{1}{p_u} \sum_{r=1}^{R} |\tilde{T}_r|^2 - \frac{k}{p_u} - \frac{1}{p_u} \sum_{r=1}^{R} \sum_{j=1}^{J(\mathcal{I}_r)} |C_j(I_r)|^2,
\end{aligned}
$$

where (a) is due to Lemma 6 and (b) follows from the fact that vertices which belong to component, only exclude vertices in $\mathcal{I}$.

A.6 Proof of Theorem 1

We use the following lemma for the proof.

Lemma 15. *Among all algorithms achieving a threshold mid, Algorithm 1 uses the least number of vertex removals.*

Proof. Proof by induction. We show for each subtree, the smallest number of vertex removal is used. Since the proposed algorithm removes a vertex only if not doing so results in having a subtree with the order larger than the threshold, it delays a removal as much as possible. Now suppose for vertex X_j, we have used the smallest number of removals, say l, in subtrees rooted at the children of X_j. Because in each of those subtrees, the removals have been delayed the most, the order of remaining part for the subtree rooted at X_j with l removals is minimum. Therefore the subtree rooted at X_j also contributes the least value (zero if it is chosen to intervene on) to the order of the subtree rooted at its parent.

□

Now, suppose for the optimum experiment target set $\mathcal{I}_r^*$, that is,

$$\mathcal{I}_r^* = \arg \min_{\mathcal{I}_r : \mathcal{I}_r \subseteq V(\tilde{T}_r)} \max_{1 \le j \le J(\mathcal{I}_r)} |C_j(\mathcal{I}_r)|,$$

with $|\mathcal{I}_r^*| = k_r$ we have $M^* := \max_{1 \le j \le J(\mathcal{I}_r^*)} |C_j(\mathcal{I}_r^*)| < \min_{X_i} mid(X_i)$. In this case, in the binary search in Algorithm 1, when the threshold is set to mid such that $M^* - 1 < mid \le M^*$, then by Lemma 15, Algorithm 1 should have used less than or equal to k_r vertex removals. If it has used less than k_r vertex removals, it means that it can achieve M^* with $|\hat{\mathcal{I}}_r| < k_r$, and hence, can achieve a value less than M^* with k_r vertex removals, which implies that $\mathcal{I}_r^*$ is not optimum. Therefore, we should have

$$\min_{X_i} mid(X_i) = \min_{\mathcal{I}_r : \mathcal{I}_r \subseteq V(\tilde{T}_r)} \max_{1 \le j \le J(\mathcal{I}_r)} |C_j(\mathcal{I}_r)|.$$

A.7 Proof of Proposition 2

Monotonicity. Consider $\mathcal{I}^1 \subseteq \mathcal{I}^2$. Target set $\mathcal{I}^2$ divides some of the components of target set $\mathcal{I}^1$ into smaller components, or removes vertices from some of them, and keeps the rest unchanged. Suppose C_j is a changed component. Therefore, corresponding to this component, for $\mathcal{I}^1$ we have the term $|C_j|^2$, and for $\mathcal{I}^2$ we have $\sum_{l=1}^{L} |C_{jl}|^2$ such that $\sum_{l=1}^{L} |C_{jl}| < |C_j|$.

Basic algebra and induction on L indicates that under this condition $\sum_{l=1}^{L} |C_{jl}|^2$ is always less that $|C_j|^2$. Hence, $\mathcal{D}(\mathcal{I}^1) \leq \mathcal{D}(\mathcal{I}^2)$.

Submodularity. We first show that the for every root vertex X_i, the set function $D(\mathcal{I}, T^{X_i})$ is submodular. i.e., for $\mathcal{I}^1 \subseteq \mathcal{I}^2$, vertex X,

$$D(\mathcal{I}^1 \cup \{X\}, T^{X_i}) - D(\mathcal{I}^1, T^{X_i}) \geq D(\mathcal{I}^2 \cup \{X\}, T^{X_i}) - D(\mathcal{I}^2, T^{X_i}).$$

By Lemma 6, the value of the function $D(\mathcal{I}, T^{X_i})$ only depends on the component containing the root. Suppose under experiment $\mathcal{I}^1$ the root vertex falls in component $C_{\mathcal{I}^1}$, and under experiment $\mathcal{I}^2$ the root vertex falls in component $C_{\mathcal{I}^2}$. If $C_{\mathcal{I}^1} = C_{\mathcal{I}^2}$, the result is immediate, as without intervening on X, $\mathcal{I}^1$ and $\mathcal{I}^2$ result in the same value for function D, and intervening on X will also have the same result in both experiments. Otherwise, since $\mathcal{I}^1 \subseteq \mathcal{I}^2$, we have $C_{\mathcal{I}^2} \subseteq C_{\mathcal{I}^1}$. Hence, the cardinality of the set of the edges which are incident to $Desc(X)$ in $C_{\mathcal{I}^1}$ is larger than the cardinality of the set of the edges which are incident to $Desc(X)$ in $C_{\mathcal{I}^2}$. This implies that we have a larger gain by intervening on X starting from $\mathcal{I}_1$ compared to $\mathcal{I}^2$, i.e., $D(\mathcal{I}^1 \cup \{X\}, T^{X_i}) - D(\mathcal{I}^1, T^{X_i}) \geq D(\mathcal{I}^2 \cup \{X\}, T^{X_i}) - D(\mathcal{I}^2, T^{X_i})$.

Finally, using equality $\mathcal{D}(\mathcal{I}) = \frac{1}{p_u} \sum_{r=1}^{R} \sum_{X \in V(\tilde{T}_r)} D(\mathcal{I}_r, T_r^X)$, since a non-negative linear combination of submodular functions is also submodular, the desired result is concluded.

A.8 Proof of Proposition 3

First we show that for a given directed graph $G_i \in MEC(G^*)$ the function $D(\mathcal{I}, G_i)$ is a monotonically increasing function of $\mathcal{I}$. In the proposed method, intervening on elements of $\mathcal{I}$, we first discover the orientation of the edges in $A(\mathcal{I}, G_i)$, and then applying the Meek rules, we possibly learn the orientation of some extra edges. Having $\mathcal{I}_1 \subseteq \mathcal{I}_2$ implies that $A(\mathcal{I}_1, G_i) \subseteq A(\mathcal{I}_2, G_i)$. Therefore using $\mathcal{I}_2$, we have more information about the direction of edges. Hence, in the step of applying Meek rules, by soundness and order-independence of Meek algorithm, we recover the direction of more extra edges, i.e., $R(\mathcal{I}_1, G_i) \subseteq R(\mathcal{I}_2, G_i)$, which in turn implies that $D(\mathcal{I}_1, G_i) \leq D(\mathcal{I}_2, G_i)$. Finally, from the equation $\mathcal{D}(\mathcal{I}) = \frac{1}{|MEC(G^*)|} \sum_{G_i \in MEC(G^*)} D(\mathcal{I}, G_i)$, the desired result is immediate.

A.9 Proof of Lemma 7

The direction $R(\mathcal{I}_1, G^*) \cup R(\mathcal{I}_2, G^*) \subseteq R(\mathcal{I}_1 \cup \mathcal{I}_2, G^*)$ is proved in the proof of Proposition 3. Define $A(\tilde{G}^*)$ as the set of directed edges in $\tilde{G}^*$, and let $R(M, G^*)$ be the set of undirected edges of $\tilde{G}^*$ whose directions can be identified by applying Meek rules starting from $A(\tilde{G}^*) \cup R(\mathcal{I}_1, G^*) \cup R(\mathcal{I}_2, G^*)$. Again by the reasoning in the proof of Proposition 3, we have $R(\mathcal{I}_1 \cup \mathcal{I}_2, G^*) \subseteq R(M, G^*)$. Therefore, in order to prove that $R(\mathcal{I}_1 \cup \mathcal{I}_2, G^*) \subseteq R(\mathcal{I}_1, G^*) \cup R(\mathcal{I}_2, G^*)$, it suffices to show that $R(M, G^*) \subseteq R(\mathcal{I}_1, G^*) \cup R(\mathcal{I}_2, G^*)$, for which it suffices to show that for every directed edge e, if $e \notin R(\mathcal{I}_1, G^*)$ and $e \notin R(\mathcal{I}_2, G^*)$, then $e \notin R(M, G^*)$.

Proof by contradiction. Let $e \notin R(\mathcal{I}_1, G^*)$ and $e \notin R(\mathcal{I}_2, G^*)$, but its orientation is learned in the first iteration of applying Meek rules to $A(\tilde{G}^*) \cup R(\mathcal{I}_1, G^*) \cup R(\mathcal{I}_2, G^*)$. Then, we have learned the orientation of e due to one of Meek rules [116]:

- **Rule 1.** $e = A - B$ is oriented as $A \to B$ if there exists C such that $e_1 = C \to A \in A(\tilde{G}^*) \cup R(\mathcal{I}_1, G^*) \cup R(\mathcal{I}_2, G^*)$, and $C - B \notin$ skeleton of G^*.
- **Rule 2.** $e = A - B$ is oriented as $A \to B$ if there exists C such that $e_1 = A \to C \in A(\tilde{G}^*) \cup R(\mathcal{I}_1, G^*) \cup R(\mathcal{I}_2, G^*)$, and $e_2 = C \to B \in A(\tilde{G}^*) \cup R(\mathcal{I}_1, G^*) \cup R(\mathcal{I}_2, G^*)$.
- **Rule 3.** $e = A - B$ is oriented as $A \to B$ if there exist C and D such that $e_1 = C \to B \in A(\tilde{G}^*) \cup R(\mathcal{I}_1, G^*) \cup R(\mathcal{I}_2, G^*)$, $e_2 = D \to B \in A(\tilde{G}^*) \cup R(\mathcal{I}_1, G^*) \cup R(\mathcal{I}_2, G^*)$, $A - C \in$ skeleton of G^*, $A - D \in$ skeleton of G^*, and $C - D \notin$ skeleton of G^*.
- **Rule 4.** $e = A - B$ is oriented as $A \to B$ and $e = B - C$ is oriented as $C \to B$ if there exists D such that $e_1 = D \to C \in A(\tilde{G}^*) \cup R(\mathcal{I}_1, G^*) \cup R(\mathcal{I}_2, G^*)$, $A - C \in$ skeleton of G^*, $A - D \in$ skeleton of G^*, and $B - D \notin$ skeleton of G^*.

In what follows, we show that the orientation of e cannot be learned due to any of the Meek rules unless directed edge e belongs to $R(\mathcal{I}_1, G^*)$ or $R(\mathcal{I}_2, G^*)$.

Rule 1.

Without loss of generality, assume $e_1 \in A(\tilde{G}^*) \cup R(\mathcal{I}_1, G^*)$. Therefore, we should have the condition of rule 1 satisfied when only intervening on $\mathcal{I}_1$ as well, which implies that $e \in R(\mathcal{I}_1, G^*)$, which is a contradiction.

Rule 2.

If both e_1 and e_2 belong to $A(\tilde{G}^*) \cup R(\mathcal{I}_1, G^*)$ (or $A(\tilde{G}^*) \cup R(\mathcal{I}_2, G^*)$), then we should have the condition of rule 2 satisfied when only intervening on $\mathcal{I}_1$ (or $\mathcal{I}_2$) as well, which implies

that $e \in R(\mathcal{I}_1, G^*)$ (or $e \in R(\mathcal{I}_1, G^*)$), which is a contradiction. Therefore, it suffices to show that the case that e_1 belongs to exactly one of $A(\tilde{G}^*) \cup R(\mathcal{I}_1, G^*)$ or $A(\tilde{G}^*) \cup R(\mathcal{I}_2, G^*)$ and e_2 belongs only to the other one, does not happen. To this end, it suffices to show that there exists no experiment target set $\mathcal{I}$ such that $e_1 \in A(\tilde{G}^*) \cup R(\mathcal{I}, G^*)$, and $e, e_2 \notin A(\tilde{G}^*) \cup R(\mathcal{I}, G^*)$, i.e., there exists no experiment target set $\mathcal{I}$ that has structure S_0, depicted in Figure A.2, as a subgraph of $\tilde{G}^*$ after applying the orientations learned from $R(\mathcal{I}, G^*)$.

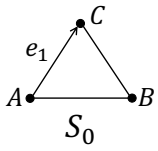

Figure A.2: Structure S_0.

If $e_1 \in A(\mathcal{I}, G^*)$, then $A \in \mathcal{I}$ or $C \in \mathcal{I}$, which implies $e \in A(\mathcal{I}, G^*)$ or $e_2 \in A(\mathcal{I}, G^*)$, respectively, and hence, $e \in R(\mathcal{I}, G^*)$ or $e_2 \in R(\mathcal{I}, G^*)$, respectively. Therefore, in either case, $e \in R(\mathcal{I}, G^*)$, and S_0 will not be a subgraph. Therefore, $e_1 \notin A(\mathcal{I}, G^*)$, and hence, e_1 was learned by applying one of the Meek rules. We consider each or the rules in the following:

- If we have learned the orientation of e_1 from rule 1, then we should have had one of the structures in Figure A.3 as a subgraph of $\tilde{G}^*$ after applying the orientations learned from $R(\mathcal{I}, G^*)$. In case of structure S_1, using rule 1 on subgraph induced on vertices $\{X_1, A, B\}$, we will also learn $A \to B$. In case of structure S_2, using rule 4, we will also learn $B \to C$. Therefore, we cannot learn only the direction of e_1 and hence, S_0 will not be a subgraph.

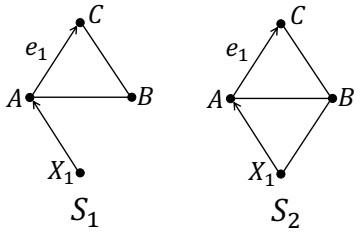

Figure A.3: Rule 1.

- If we have learned the orientation of e_1 from rule 3, then we have had one of the structures in Figure A.4 as a subgraph of $\tilde{G}^*$ after applying the orientations learned from $R(\mathcal{I}, G^*)$. In case of structures S_3 and S_4, using rule 1 on subgraph induced on vertices $\{X_2, C, B\}$, we will also learn $C \to B$. In case of structure S_5, using rule 3 on subgraph induced on vertices $\{B, X_2, C, X_1\}$, we will also learn $B \to C$. Therefore, we cannot learn only the direction of e_1 and hence, S_0 will not be a subgraph.

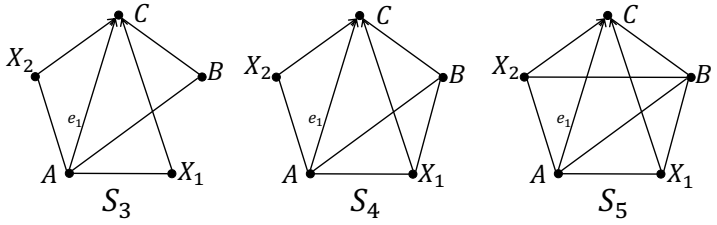

Figure A.4: Rule 3.

- If we have learned the orientation of e_1 from rule 4, then we have had one of the structures in Figure A.5 as a subgraph of $\tilde{G}^*$ after applying the orientations learned from $R(\mathcal{I}, G^*)$. In case of structures S_6, using rule 1 on subgraph induced on vertices $\{X_1, C, B\}$, we will also learn $C \to B$. In case of structure S_7, using rule 1 on subgraph induced on vertices $\{X_2, X_1, B\}$, we will also learn $X_1 \to B$, and then using rule 4 on subgraph induced on vertices $\{B, A, X_2, X_1\}$, we will also learn $A \to B$. In case of structures S_8, using rule 4 on subgraph induced on vertices $\{B, X_2, X_1, C\}$, we will also learn $B \to C$. Therefore, we cannot learn only the direction of e_1 and hence, S_0 will not be a subgraph.

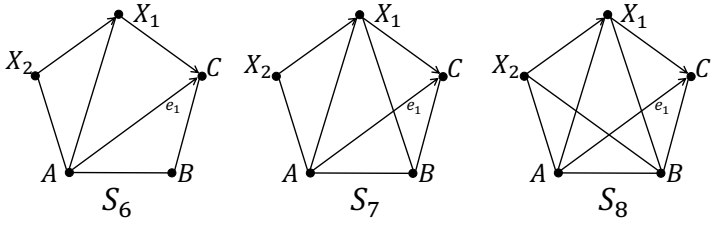

Figure A.5: Rule 4.

- If we have learned the orientation of e_1 from rule 2, then we should have had one of the structures in Figure A.6 as a subgraph of $\tilde{G}^*$ after applying the orientations learned from $R(\mathcal{I}, G^*)$. In case of structure S_9, using rule 1 on subgraph induced on vertices $\{X_1, C, B\}$, we will also learn $C \to B$ and hence, S_0 will not be a subgraph. In case of structure S_{10}, if $X_1 \in \mathcal{I}$, then the direction of the edge $X_1 - B$ will be also known. If the direction of this edge is $X_1 \to B$, then using rule 2 on subgraph induced on vertices $\{A, X_1, B\}$, we will also learn $A \to B$; otherwise, using rule 2 on subgraph induced on vertices $\{B, X_1, C\}$, we will also learn $C \to B$. Therefore, $X_1 \notin \mathcal{I}$. Also, as mentioned earlier, $A \notin \mathcal{I}$. Therefore, we have learned the orientation of $A \to X_1$ from applying Meek rules.

  In the triangle induced on vertices $\{X_1, B, A\}$, we have learned only the orientation of one edge, which is $A \to X_1$. But as seen in structures S_1 to S_9, all of them lead to learning the orientation of at least 2 edges of a triangle. In the following, we will show that a structure of form S_{10}, does not lead to learning the orientation of only $A \to X_1$ and making S_{10} a subgraph either.

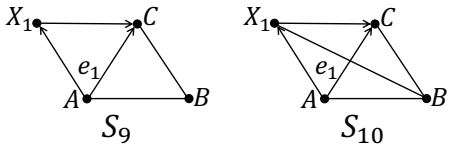

Figure A.6: Rule 2.

Suppose we had learned $A \to X_1$ via a structure of form S_{10}, as depicted in Figure A.7(a). Using rule 4 on subgraph induced on vertices $\{X_2, X_1, C, B\}$, we will also learn $B \to C$. Therefore, we should have the edge $X_2 - C$ too. Also, using rule 2 on triangle induced on vertices $\{X_2, X_1, C\}$, the orientation of this edges should be $X_2 \to C$. Therefore, in order to have S_{10} as a subgraph, we need to have the structure depicted in Figure A.7(b) as a subgraph. As seen in Figure A.7(b), we again have a structure similar to S_{10}: a complete skeleton K_5, which contains $X_j \to C$, $A \to X_j$, $X_j - B$, for $j \in \{1, 2\}$ and $X_2 \to X_1$, with a triangle on vertices $\{X_2, B, A\}$, in which we have learned only the orientation of $A \to X_2$.

We claim that this procedure always repeats, i.e., at step i, we end up with skeleton K_i, which contains $X_j \to C$, $A \to X_j$, $X_j - B$, for $j \in \{1, ..., i\}$ and $X_k \to X_j$, for $1 \leq j < k \leq i$, with a triangle induced on vertices $\{X_i, B, A\}$, in which we have

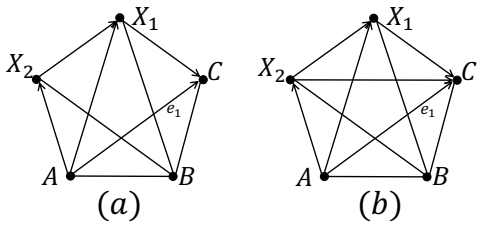

Figure A.7: Step of the induction.

learned only the orientation of $A \to X_i$. We prove this claim by induction. We have already proved the base of the induction above. For the step of the induction, suppose the hypothesis is true for $i-1$. Add vertex X_i to form a structure of form S_{10} for $A \to X_{i-1}$. X_i should be adjacent to X_j, for $j \in \{1, ..., i-2\}$; otherwise, using rule 4 on subgraph induced on vertices $\{X_i, X_{i-1}, X_j, B\}$, we will also learn $B \to X_j$. Moreover, using rule 2 on triangle induced on vertices $\{X_i, X_{i-1}, X_j\}$, the direction of $X_i - X_j$ should be $X_i \to X_j$. Also, using rule 4 on subgraph induced on vertices $\{X_i, X_{i-1}, C, B\}$, we will also learn $B \to C$. Therefore, we should have the edge $X_i - C$ too.

We showed that S_0 is a subgraph only if S_{10} is a subgraph, and S_{10} is a subgraph only if the structure in Figure A.7(b) is a subgraph, and this chain of required subgraphs continues. Therefore, since the order of the graph is finite, there exist a step where since we cannot add a new vertex, it is not possible to have one of the required subgraphs, and hence we conclude that S_0 is not a subgraph.

Rule 3.

Since edges e_1 and e_2 form a v-structure, they should appear in $A(\tilde{G}^*)$ as well. Therefore, we should have the condition of rule 3 satisfied when only intervening on $\mathcal{I}_1$ as well, which implies that $e \in R(\mathcal{I}_1, G^*)$, which is a contradiction.

Rule 4.

Without loss of generality, assume $e_1 \in R(\mathcal{I}_1, G^*) \cup A(\tilde{G}^*)$. Therefore, we should have the condition of rule 4 satisfied when only intervening on $\mathcal{I}_1$ as well, which implies that $e \in R(\mathcal{I}_1, G^*)$, which is a contradiction.

The argument above proves that there is no edge e such that $e \notin R(\mathcal{I}_1, G^*)$ and $e \notin R(\mathcal{I}_2, G^*)$, but $e \in R(M, G^*)$.

A.10 Proof of Theorem 2

Due to Proposition 3, it suffices to show that for $\mathcal{I}_1 \subseteq \mathcal{I}_2 \subseteq V$, and $X_i \in V$, we have $\mathcal{D}(\mathcal{I}_1 \cup \{X_i\}) - \mathcal{D}(\mathcal{I}_1) \geq \mathcal{D}(\mathcal{I}_2 \cup \{X_i\}) - \mathcal{D}(\mathcal{I}_2)$. First we show that for a given directed graph $G_i \in \mathit{MEC}(G^*)$ the function $D(\mathcal{I}, G_i)$ is a submodular function of $\mathcal{I}$. From Lemma 7, we have $R(\mathcal{I}_1 \cup \{X_i\}, G_i) = R(\mathcal{I}_1, G_i) \cup R(\{X_i\}, G_i)$. Therefore,

$$\begin{aligned} D(\mathcal{I}_1 \cup \{X_i\}, G_i) - D(\mathcal{I}_1, G_i) &= |R(\mathcal{I}_1 \cup \{X_i\}, G_i)| - |R(\mathcal{I}_1, G_i)| \\ &= |R(\mathcal{I}_1, G_i) \cup R(\{X_i\}, G_i)| - |R(\mathcal{I}_1, G_i)| \\ &= |R(\{X_i\}, G_i)| - |R(\mathcal{I}_1, G_i) \cap R(\{X_i\}, G_i)|. \end{aligned}$$

Similarly,

$$D(\mathcal{I}_2 \cup \{X_i\}, G_i) - D(\mathcal{I}_2, G_i) = |R(\{X_i\}, G_i)| - |R(\mathcal{I}_2, G_i) \cap R(\{X_i\}, G_i)|.$$

Since $\mathcal{I}_1 \subseteq \mathcal{I}_2$, as seen in the proof of Proposition 3, $R(\mathcal{I}_1, G_i) \subseteq R(\mathcal{I}_2, G_i)$. Therefore, $-|R(\mathcal{I}_1, G_i) \cap R(\{X_i\}, G_i)| \geq -|R(\mathcal{I}_2, G_i) \cap R(\{X_i\}, G_i)|$, which implies that

$$D(\mathcal{I}_1 \cup \{X_i\}, G_i) - D(\mathcal{I}_1, G_i) \geq D(\mathcal{I}_2 \cup \{X_i\}, G_i) - D(\mathcal{I}_2, G_i).$$

This together with the fact that the function $D(\mathcal{I}, G_i)$ is a monotonically increasing function of $\mathcal{I}$ (observed in the proof of Proposition 3) shows that $D(\mathcal{I}, G_i)$ is a submodular function of $\mathcal{I}$.

Finally, we have $\mathcal{D}(\mathcal{I}) = \frac{1}{|\mathit{MEC}(G^*)|} \sum_{G_i \in \mathit{MEC}(G^*)} D(\mathcal{I}, G_i)$. Since a non-negative linear combination of submodular functions is also submodular, the proof is concluded.

A.11 Proof of Proposition 4

The worst case in terms of computational complexity happens when $H = \tilde{G}$, as it requires maximum number of recursions. In function Counter, we set each vertex X_i as the root and call the function Counter for the rooted essential graph $\tilde{G}_r^{X_i}$ to compute the number

of DAGs in the MEC corresponding to $\tilde{G}_r^{X_i}$. Using Meek rules, the directed edges in $\tilde{G}_r^{X_i}$ can be recovered in time $\mathcal{O}(p^3)$.

Now, we show that the degree of each vertex X_j in $\tilde{G}_r^{X_i}$ decreases at least by one after removing directed edges. To do so, we prove that there exists a directed edge in $\tilde{G}_r^{X_i}$ that goes to vertex X_j. If X_j is a neighbor of X_i the proof is done, as edges are always directed from the root vertex towards its neighbors. Otherwise, consider the shortest path from X_i to X_j in $\tilde{G}_r^{X_i}$. This path must pass through one of the neighbors of X_j, say, X_k. Since the distance from X_i to X_k is less than X_i to X_j, $X_k - X_j$ should be oriented as $X_k \to X_j$ [19]. Therefore, the degree of each vertex X_j in $\tilde{G}_r^{X_i}$ decreases at least by one after removing directed edges in $\tilde{G}_r^{X_i}$.

Let $t(\Delta)$ be the computational complexity of Algorithm 3 on a graph with maximum degree Δ. Based on what we proved above, we have

$$t(\Delta) \leq pt(\Delta - 1) + Cp^3,$$

where C is a constant. The above inequality holds true since we have at most p chain component in $\tilde{G}_r^{X_i}$, where the maximum degree in each of them is at most $\Delta - 1$. From this inequality, it can be shown that $t(\Delta)$ is in the order of $\mathcal{O}(p^{\Delta+1})$. Since we may have at most p chain components in essential graph $\tilde{G}$, the computational complexity of Algorithm 3 is in the order of $\mathcal{O}(p^{\Delta+2})$.

A.12 Proof of Theorem 3

The objective is to show that for the input essential graph $\tilde{G}$, any DAG G in $MEC(\tilde{G})$ is generated with probability $1/Size(\tilde{G})$.

Proof by induction: The function COUNTER finds the size of a chain component recursively, i.e., after setting a vertex X as the root and finding the orientations in $\tilde{G}_r^X$, it calls itself to obtain the size of the chain components of $\tilde{G}_r^X$. We induct on the maximum number of recursive calls required for complete orienting.

Induction base: For the base of the induction, we consider an essential graph with no required recursive call: Consider essential graph $\tilde{G}$ with chain component set $\mathcal{G}$, for which, for all $\tilde{G}_r \in \mathcal{G}$, for all $X \in V(\tilde{G}_r)$, $Size(\tilde{G}_r^X) = 1$ (as an example, consider the case that $\tilde{G}_r$ is a tree). Consider G in the MEC represented by $\tilde{G}$, and assume vertex $X_{\tilde{G}_r}$ is required to

be set as the root in chain component $\tilde{G}_r \in \mathcal{G}$ for G to be obtained. We have

$$\begin{aligned} P(G) &= \prod_{\tilde{G}_r \in \mathcal{G}} P(X_{\tilde{G}_r} \text{ picked}) = \prod_{\tilde{G}_r \in \mathcal{G}} \frac{Size(\tilde{G}_r^{X_{\tilde{G}_r}})}{Size(\tilde{G}_r)} \\ &= \prod_{\tilde{G}_r \in \mathcal{G}} \frac{1}{Size(\tilde{G}_r)} = \frac{1}{\prod_{\tilde{G}_r \in \mathcal{G}} Size(\tilde{G}_r)} \\ &= \frac{1}{Size(\tilde{G})}, \end{aligned}$$

where, the last equality follows from equation (3.13).

Induction hypothesis: For an essential graph $\tilde{G}$ with maximum required recursions of $l-1$, any DAG G in the MEC represented by $\tilde{G}$ is generated with probability $1/Size(\tilde{G})$.

Induction step: We need to show that for an essential graph $\tilde{G}$ with maximum required recursions of l, any DAG G in the MEC represented by $\tilde{G}$ is generated with probability $1/Size(\tilde{G})$. Assume vertex $X_{\tilde{G}_r}$ is required to be set as the root in chain component $\tilde{G}_r \in \mathcal{G}$, and $V_{\tilde{G}_r^{X_{\tilde{G}_r}}}$ is the set of vertices required to be set as root in the next recursions in obtained chain components in $\tilde{G}_r^{X_{\tilde{G}_r}}$ for G to be obtained. We have

$$\begin{aligned} P(G) &= \prod_{\tilde{G}_r \in \mathcal{G}} P(X_{\tilde{G}_r} \text{ picked}) P(V_{\tilde{G}_r^{X_{\tilde{G}_r}}} \text{ picked}) \\ &= \prod_{\tilde{G}_r \in \mathcal{G}} \frac{Size(\tilde{G}_r^{X_{\tilde{G}_r}})}{Size(\tilde{G}_r)} P(V_{\tilde{G}_r^{X_{\tilde{G}_r}}} \text{ picked}). \end{aligned}$$

By the induction hypothesis,

$$P(V_{\tilde{G}_r^{X_{\tilde{G}_r}}} \text{ picked}) = 1/Size(\tilde{G}_r^{X_{\tilde{G}_r}}).$$

Therefore,

$$\begin{aligned} P(G) &= \prod_{\tilde{G}_r \in \mathcal{G}} \frac{Size(\tilde{G}_r^{X_{\tilde{G}_r}})}{Size(\tilde{G}_r)} \frac{1}{Size(\tilde{G}_r^{X_{\tilde{G}_r}})} \\ &= \frac{1}{\prod_{\tilde{G}_r \in \mathcal{G}} Size(\tilde{G}_r)} \\ &= \frac{1}{Size(\tilde{G})}, \end{aligned}$$

where, the last equality follows from equation (3.13).

A.13 Proof of Corollary 1

For any chain component $\tilde{G}$, for calculating $\textsc{Counter}(\tilde{G}, \tilde{G})$ we are required to calculate the size of all possible subsequent rooted classes. Therefore, we do not need to calculate the size of any rooted subclasses anymore. Hence, by Proposition 4, we obtain all probabilities of the from $\frac{\textsc{Counter}(\tilde{G}^X, \tilde{G}^X)}{\textsc{Counter}(\tilde{G}, \tilde{G})}$ in $\mathcal{O}(p^{\Delta+2})$. After selecting one of the vertices in $\tilde{G}$ as the root, say X, we recover all directed edges in $\tilde{G}^X$ in $\mathcal{O}(p^3)$ and obtain chain components of $\tilde{G}^X$. Similar to the proof of Proposition 4, let $t(\Delta)$ be the running time of the algorithm on a chain component in $\mathcal{G}$ with maximum degree of Δ. We have

$$t(\Delta) \leq pt(\Delta - 1) + Cp^3,$$

where C is a constant. It can be shown that $t(\Delta)$ is in the order of $\mathcal{O}(\Delta p^{\Delta+1})$. Since we may have at most p chain components in $\mathcal{G}$, the computational complexity of uniform sampler would be in the order of $\mathcal{O}(p^{\Delta+2})$. Therefore, the computational complexity of the approach is $\mathcal{O}(p^{\Delta+2} + p^{\Delta+2}) = \mathcal{O}(p^{\Delta+2})$.

A.14 Proof of Theorem 4

Proposition 16 (**Chernoff Bound**). *Let $X_1, ..., X_N$ be independent random variables such that for all i, $0 \leq X_i \leq 1$. Let $\mu = \mathbb{E}[\sum_{i=1}^{N} X_i]$. Then*

$$P(|\sum_{i=1}^{N} X_i - \mu| \geq \epsilon\mu) \leq 2\exp(-\frac{\epsilon^2}{2+\epsilon}\mu).$$

Proof of Proposition 5. For $i \in \{1, ..., N\}$, define $X_i = \frac{D(\mathcal{I}, G_i)}{|\bar{A}(\tilde{G})|}$. We note that for the estimator in Algorithm 4, we have $\mathbb{E}[D(\mathcal{I}, G_i)] = \mathcal{D}(\mathcal{I})$, where G_i is a random generated DAG in

the sampler in Algorithm 4. This can be proven as follows:

$$\begin{aligned}\mathbb{E}[D(\mathcal{I}, G_i)] &= \sum_{G_i' \in MEC(G^*)} P(G_i = G_i') D(\mathcal{I}, G_i') \\ &= \sum_{G_i' \in MEC(G^*)} \frac{1}{|MEC(G^*)|} D(\mathcal{I}, G_i') \\ &= \mathcal{D}(\mathcal{I}).\end{aligned}$$

Therefore, $\mathbb{E}[X_i] = \frac{1}{|\bar{A}(\tilde{G})|}\mathcal{D}(\mathcal{I})$.

Using Chernoff bound we have

$$\begin{aligned}P(|\sum_{i=1}^{N} X_i - \frac{N}{|\bar{A}(\tilde{G})|}\mathcal{D}(\mathcal{I})| \geq \epsilon \frac{N}{|\bar{A}(\tilde{G})|}\mathcal{D}(\mathcal{I})) &\leq 2\exp(-\frac{N\epsilon^2}{|\bar{A}(\tilde{G})|(2+\epsilon)}\mathcal{D}(\mathcal{I})) \\ &\leq 2\exp(-\frac{N\epsilon^2}{|\bar{A}(\tilde{G})|(2+\epsilon)}).\end{aligned}$$

Therefore,

$$P(|\frac{1}{N}\sum_{i=1}^{N} D(\mathcal{I}, G_i) - \mathcal{D}(\mathcal{I})| \geq \epsilon\mathcal{D}(\mathcal{I})) \leq 2\exp(-\frac{N\epsilon^2}{|\bar{A}(\tilde{G})|(2+\epsilon)}).$$

Hence,

$$P(|\hat{\mathcal{D}}(\mathcal{I}) - \mathcal{D}(\mathcal{I})| < \epsilon\mathcal{D}(\mathcal{I})) > 1 - 2\exp(-\frac{N\epsilon^2}{|\bar{A}(\tilde{G})|(2+\epsilon)}).$$

Setting $N > \frac{|\bar{A}(\tilde{G})|(2+\epsilon)}{\epsilon^2}\ln(\frac{2}{\delta})$, upper bounds the right hand side with $1-\delta$ and concludes the desired result.

□

A.15 Proof of Theorem 5

Let $\mathcal{I}^* = \{X_1^*, ..., X_k^*\} \in \arg\max_{\mathcal{I}:\mathcal{I}\subseteq V,|\mathcal{I}|=k} \mathcal{D}(\mathcal{I})$. We have

$$\begin{aligned}\mathcal{D}(\mathcal{I}^*) &\overset{(a)}{\leq} \mathcal{D}(\mathcal{I}^* \cup \mathcal{I}_i) = \mathcal{D}(\mathcal{I}_i) + \sum_{j=1}^{k} [\mathcal{D}(\mathcal{I}_i \cup \{X_1^*, ..., X_j^*\}) - \mathcal{D}(\mathcal{I}_i \cup \{X_1^*, ..., X_{j-1}^*\})] \\ &\overset{(b)}{\leq} \mathcal{D}(\mathcal{I}_i) + \sum_{j=1}^{k} [\mathcal{D}(\mathcal{I}_i \cup \{X_j^*\}) - \mathcal{D}(\mathcal{I}_i)],\end{aligned} \tag{A.1}$$

where (a) follows from Proposition 3, and (b) follows from Theorem 2. Define $\hat{\mathcal{D}}_{i,X,1}$ and $\hat{\mathcal{D}}_{i,X,2}$ as the first and second calls of the estimator in i-th step for variable X, respectively. By the assumption of the theorem we have

$$\mathcal{D}(\mathcal{I}_i \cup \{X_j^*\}) - \epsilon \mathcal{D}(\mathcal{I}_i \cup \{X_j^*\}) < \hat{\mathcal{D}}_{i,X_j^*,1}(\mathcal{I}_i \cup \{X_j^*\}),$$

with probability larger than $1 - \delta$. Therefore,

$$\mathcal{D}(\mathcal{I}_i \cup \{X_j^*\}) < \hat{\mathcal{D}}_{i,X_j^*,1}(\mathcal{I}_i \cup \{X_j^*\}) + \epsilon \mathcal{D}(\mathcal{I}^*),$$

with probability larger than $1 - \delta$. Similarly

$$\begin{aligned}\hat{\mathcal{D}}_{i,X_j^*,2}(\mathcal{I}_i) < \mathcal{D}(\mathcal{I}_i) + \epsilon \mathcal{D}(\mathcal{I}_i) \qquad & w.p. > 1 - \delta, \\ \Rightarrow -\mathcal{D}(\mathcal{I}_i) < -\hat{\mathcal{D}}_{i,X_j^*,2}(\mathcal{I}_i) + \epsilon \mathcal{D}(\mathcal{I}^*) \qquad & w.p. > 1 - \delta,\end{aligned}$$

Therefore,

$$\begin{aligned}\mathcal{D}(\mathcal{I}_i \cup \{X_j^*\}) - \mathcal{D}(\mathcal{I}_i) < \hat{\mathcal{D}}_{i,X_j^*,1}(\mathcal{I}_i \cup \{X_j^*\}) \\ - \hat{\mathcal{D}}_{i,X_j^*,2}(\mathcal{I}_i) + 2\epsilon \mathcal{D}(\mathcal{I}^*) \qquad w.p. > 1 - 2\delta.\end{aligned} \tag{A.2}$$

Also, by the definition of the greedy algorithm,

$$\begin{aligned}&\hat{\mathcal{D}}_{i,X_j^*,1}(\mathcal{I}_i \cup \{X_j^*\}) - \hat{\mathcal{D}}_{i,X_j^*,2}(\mathcal{I}_i) \\ &\qquad \leq \hat{\mathcal{D}}_{i,X_{i+1},1}(\mathcal{I}_i \cup \{X_{i+1}\}) - \hat{\mathcal{D}}_{i,X_{i+1},2}(\mathcal{I}_i) \\ &\qquad = \hat{\mathcal{D}}_{i,X_{i+1},1}(\mathcal{I}_{i+1}) - \hat{\mathcal{D}}_{i,X_{i+1},2}(\mathcal{I}_i),\end{aligned} \tag{A.3}$$

and similar to (A.2), we have

$$\begin{aligned}\hat{\mathcal{D}}_{i,X_{i+1},1}(\mathcal{I}_{i+1}) - \hat{\mathcal{D}}_{i,X_{i+1},2}(\mathcal{I}_i) < \mathcal{D}(\mathcal{I}_{i+1}) \\ - \mathcal{D}(\mathcal{I}_i) + 2\epsilon\mathcal{D}(\mathcal{I}^*) \qquad w.p. > 1 - 2\delta.\end{aligned} \tag{A.4}$$

Therefore, from equations (A.2), (A.3), and (A.4) we have

$$\mathcal{D}(\mathcal{I}_i \cup \{X_j^*\}) - \mathcal{D}(\mathcal{I}_i) < \mathcal{D}(\mathcal{I}_{i+1}) - \mathcal{D}(\mathcal{I}_i) + 4\epsilon\mathcal{D}(\mathcal{I}^*), \tag{A.5}$$

with probability larger than $1 - 4\delta$. Plugging (A.5) back in (A.1), we get

$$\begin{aligned}\mathcal{D}(\mathcal{I}^*) &< \mathcal{D}(\mathcal{I}_i) + \sum_{j=1}^{k}[\mathcal{D}(\mathcal{I}_{i+1}) - \mathcal{D}(\mathcal{I}_i) + 4\epsilon\mathcal{D}(\mathcal{I}^*)] \\ &= \mathcal{D}(\mathcal{I}_i) + k[\mathcal{D}(\mathcal{I}_{i+1}) - \mathcal{D}(\mathcal{I}_i)] + 4k\epsilon\mathcal{D}(\mathcal{I}^*),\end{aligned}$$

with probability larger than $1 - 4k\delta$. Therefore,

$$\begin{aligned}&\mathcal{D}(\mathcal{I}^*) - \mathcal{D}(\mathcal{I}_i) \\ &< k[\mathcal{D}(\mathcal{I}^*) - \mathcal{D}(\mathcal{I}_i)] - k[\mathcal{D}(\mathcal{I}^*) - \mathcal{D}(\mathcal{I}_{i+1})] + 4k\epsilon\mathcal{D}(\mathcal{I}^*),\end{aligned}$$

with probability larger than $1 - 4k\delta$. Defining $a_i \coloneqq \mathcal{D}(\mathcal{I}^*) - \mathcal{D}(\mathcal{I}_i)$, and noting that $a_0 = \mathcal{D}(\mathcal{I}^*)$, by induction we have

$$\begin{aligned}a_k &= \mathcal{D}(\mathcal{I}^*) - \mathcal{D}(\mathcal{I}_k) \\ &< (1 - \frac{1}{k})^k\mathcal{D}(\mathcal{I}^*) + 4\epsilon\mathcal{D}(\mathcal{I}^*)\sum_{j=0}^{k-1}(1 - \frac{1}{k})^j \\ &< [\frac{1}{e} + 4\epsilon k]\mathcal{D}(\mathcal{I}^*) \qquad w.p. > 1 - 4k^2\delta.\end{aligned}$$

It concludes that

$$\mathcal{D}(\mathcal{I}_k) > (1 - \frac{1}{e} - 4\epsilon k)\mathcal{D}(\mathcal{I}^*) \qquad w.p. > 1 - 4k^2\delta.$$

Therefore, for $\epsilon = \frac{\epsilon'}{4k}$ and $\delta = \frac{\delta'}{4k^2}$, Algorithms 2 is a $(1 - \frac{1}{e} - \epsilon')$-approximation algorithm with probability larger than $1 - \delta'$.

A.16 Proof of Proposition 5

We require the following lemma for the proof.

Lemma 16. *If a directed chordal graph has a directed cycle then it has a directed cycle of size 3.*

Proof. If the directed cycle is of size 3 itself, the claim is trivial. Suppose the directed cycle C_n is of size $n > 3$. Relabel the vertices of C_n to have $C_n = (X_1, ..., X_n, X_1)$. Since the graph is chordal, C_n has a chord and hence we have a triangle induced on vertices $\{X_i, X_{i+1}, X_{i+2}\}$ for some i. If the direction of $X_i - X_{i+2}$ is $X_{i+2} \to X_i$, we have the directed cycle $(X_i, X_{i+1}, X_{i+2}, X_i)$ which is of size 3. Otherwise, we have the directed cycle $C_{n-1} = (X_1, ..., X_i, X_{i+2}, .., X_n, X_1)$ on $n-1$ vertices. Relabeling the vertices from 1 to $n-1$ and repeating the above reasoning concludes the lemma.

□

Proof of Proposition 5. All the components in the undirected subgraph of $\tilde{G}$ are chordal [28]. Therefore, by Lemma 16, to insure that a generated directed graph is a DAG, it suffices to make sure that it does not have any directed cycles of length 3, which is one of the checks that we do in the proposed procedure. For checking if the generated DAG is in the same Markov equivalence class as G^*, since they have the same skeleton, it suffices to check if they have the same set of v-structures [16], which is the other check that we do in the sampler in Algorithm 5.

□

APPENDIX B

APPENDIX OF CHAPTER 4

B.1 Proof of Theorem 6

For the choice of $X_S = Pa(X_k)$, we have $X_k^{(i)} - (\beta_{k|S}^{(i)})^\top X_S^{(i)} = N_k^{(i)}$. Therefore, if the variance of N_k is not changed, then for this choice of X_S, we have

$$\mathbb{E}[(X_k^{(i)} - (\beta_{k|S}^{(i)})^\top X_S^{(i)})^2] = \mathbb{E}[(X_k^{(j)} - (\beta_{k|S}^{(j)})^\top X_S^{(j)})^2].$$

To prove the only if side, define $Anc(X_i)$ as the set of ancestors of vertex X_i. For any set $X_S \subseteq N(Y)$ such that $\beta_{k|S}^{(i)} = \beta_{k|S}^{(j)}$, using representation (2.3), we have:

$$\begin{aligned} X_k^{(i)} &= \sum_{X_a \in Anc(X_k)\setminus\{X_k\}} c_a N_a^{(i)} + N_k^{(i)}, \\ (\beta_{k|S}^{(i)})^\top X_S^{(i)} &= \sum_{X_a \in Anc(X_k)\setminus\{X_k\}} b_a N_a^{(i)} + \sum_{X_a \in Anc(S_{CH})\setminus Anc(X_k)} b'_a N_a^{(i)} + b_k N_k^{(i)}, \end{aligned}$$

where $S_{CH} := X_S \cap Ch(X_k)$. Moreover, coefficients b_a's and c_a's are functions of B and $\beta_{k|S}$, which are fixed across the two domains. Therefore,

$$X_k^{(i)} - (\beta_{k|S}^{(i)})^\top X_S^{(i)} = \sum_{X_a \in Anc(X_k)\setminus\{X_k\}} (c_a - b_a) N_a^{(i)} - \sum_{X_a \in Anc(S_{CH})\setminus Anc(X_k)} b'_a N_a^{(i)} + (1 - b_k) N_k^{(i)}. \tag{B.1}$$

If the variance of N_k varies, then by PIC, $\mathbb{E}[(X_k^{(i)} - (\beta_{k|S}^{(i)})^\top X_S^{(i)})^2] \neq \mathbb{E}[(X_k^{(j)} - (\beta_{k|S}^{(j)})^\top X_S^{(j)})^2]$ for all $X_S \subseteq N(X_k)$ almost surely.

B.2 Proof of Theorem 7

We first prove that for a pair of domains $(D^{(i)}, D^{(j)})$ with target set Δ_{ij} and for every target variable $X_k \in \Delta_{ij}$, we have $\beta^{(i)}_{k|S} \neq \beta^{(j)}_{k|S}$ almost surely if $Pa(X_k) \subsetneq X_S$ where $X_S \subseteq N(X_k)$. We know that the regression coefficients can be obtained as follows:

$$\beta_{k|S} = \mathbb{E}[X_S X_S^T]^{-1}\mathbb{E}[X_S X_k].$$

Moreover, for any invertible matrix A, we have: $A^{-1} = adj(A)/det(A)$ where (i,j)-th entry of $adj(A)$ is equal to $(-1)^{i+j}M_{ji}$ where M_{ij} is the determinant of a matrix resulted by deleting i-th row and j-th column of A. Furthermore, for any matrix A, $det(A)$ is a multivariate polynomial function of its entries. For any $X_i, X_j \in Ch(X_k)$, it can be easily seen that the corresponding entry to (X_i, X_j) in $\mathbb{E}[X_S X_S^T]$ has a term $var(N_k)$. Thus, any entry of $\beta_{k|S}$ is a polynomial fraction of the form $f(var(N_k))/g(var(N_k))$ where f and g are two polynomial functions. About the function $g(.)$, the constant term of $g(.)$ is the determinant of $\mathbb{E}[X_S X_S^T]$ by setting the term $var(N_k)$ to zero. We will prove that the constant term of $g(.)$ is equal to zero if it is corresponded to a child entry of $\beta_{k|S}$. We need to show that the regression coefficients of such entries are zero if $var(N_k) = 0$. This is true since by setting the regression coefficients of $Pa(X_k)$ to their true values in the model and the rest to zero, the mean square error would be zero. Since for a polynomial fraction corresponding to a child variable, the constant of function in numerator is zero while the one in denominator is nonzero, the value of fraction will change almost surely by changing the value of $var(N_k)$.

For the cases that $Pa(X_k) \not\subseteq X_S$, we consider the following assumption:

Assumption 11. *Let c^f_r and c^g_r be the constant coefficients of the term $var^r(N_k)$ in polynomial functions f and g. We assume that there exist coefficients c^f_u, c^g_u, c^f_w, and c^g_w such that $\frac{c^f_u}{c^g_u} \neq \frac{c^f_w}{c^g_w}$.*

Based on the above assumption, the polynomial fraction $f(var(N_k))/g(var(N_k))$ cannot be a constant by varying $var(N_k)$. To see this, suppose that this fraction is equal to some constant γ. However, the equation $f - \gamma g = 0$ has finite roots due to fundamental theorem of algebra (note that all the coefficients of $f - \gamma g$ are not zero due to Assumption 11). Thus, the polynomial fraction cannot remain unchanged by varying $var(N_k)$ and the proof is complete.

B.3 Proof of Theorem 9

We first note that if for domains $D^{(i)}$ and $D^{(j)}$, $V^{(i,j)}_{(C,E)} \neq 0$, then at least one of the variables a, σ^2_C, or σ^2_E has varied across the two domains and hence, by faithfulness assumption, $V^{(i,j)}_{(E,C)} = 3$. Noting that $0 \leq V^{(i,j)}_{\pi} \leq 3$, this implies that

$$V^{(i,j)}_{(C,E)} \leq V^{(i,j)}_{(E,C)}, \qquad \forall i, j.$$

Summing up over $\{i, j\}$, it implies that $\mathcal{T}^{MC}_{(C,E)}(\mathcal{D}) = 0$, and hence $\mathcal{T}^{MC}_{(C,E)}(\mathcal{D}) \leq \mathcal{T}^{MC}_{(E,C)}(\mathcal{D})$.

If there exists a pair of domains $\{D^{(i)}, D^{(j)}\}$ for which $1 \leq V^{(i,j)}_{(C,E)} \leq 2$, then since $V^{(i,j)}_{(E,C)} = 3$, we have $V^{(i,j)}_{(C,E)} < V^{(i,j)}_{(E,C)}$. Therefore, $\mathcal{T}^{MC}_{(E,C)}(\mathcal{D}) \geq 1$. Also, as mentioned earlier, $\mathcal{T}^{MC}_{(C,E)}(\mathcal{D}) = 0$. Therefore, in this case, we have $\mathcal{T}^{MC}_{(C,E)}(\mathcal{D}) < \mathcal{T}^{MC}_{(E,C)}(\mathcal{D})$.

B.4 Proof of Theorem 10

We relabel variables according to π_c to have $\pi_c(i) = X_i$, that is, in the causal order, any variable with smaller label proceeds variables with larger labels. Since π_c is causal, $\hat{B}_{\pi_c} = B$, and $\hat{\Omega}_{\pi_c} = \Omega$. Therefore, Γ'_{π_c} is exactly the set of parameters of the system. Therefore, for a pair of domains $\{D^{(i)}, D^{(j)}\}$, $V^{(i,j)}_{\pi_c}$ denotes exactly how many of the parameters of the system have changed across domains $D^{(i)}$ and $D^{(j)}$.

On the other hand, since π' is not causal, there exist parent variables who are regressed on their children, and hence, the corresponding elements of $\hat{B}_{\pi'}$ and $\hat{\Omega}_{\pi'}$ will be functions of more than one parameter of the system. Therefore, by faithfulness assumption, they will vary by a change in any of the involved parameters across any two domains $D^{(i)}$ and $D^{(j)}$. Therefore, an argument similar to the one in the proof of Theorem 9 implies that

$$\mathcal{T}_{\pi^*}(\mathcal{D}) \leq \mathcal{T}_{\pi'}(\mathcal{D}).$$

Also, since π' is not causal, there exist indices i and j, such that $X_i \to X_j \in G$, but $(\pi')^{-1}(X_i) > (\pi')^{-1}(X_j)$. Having π' as the order, we regress X_i on a set X_S including X_j. We denote the coefficient corresponding to X_j by β, and the variance of the residual of the regression by σ^2.

First, we note that β will be non-zero, as X_j is in the Markov blanket of X_i. Applying

the result of [117], β and σ^2 can be written as follows:

$$\beta = \frac{\tilde{B}_{i,j}\tilde{\sigma}_j^{-2} - \sum_{k:X_k \in S} \tilde{B}_{i,k}\tilde{B}_{j,k}\tilde{\sigma}_k^{-2}}{\tilde{\sigma}_i^{-2} + \sum_{k:X_k \in S} \tilde{B}_{i,k}^2\tilde{\sigma}_k^{-2}},$$

and

$$\sigma^2 = (\tilde{\sigma}_i^{-2} + \sum_{k:X_k \in S} \tilde{B}_{i,k}^2\tilde{\sigma}_k^{-2})^{-1},$$

where $\tilde{\sigma}_i^2$ and $\tilde{B}_{i,j}$ are the variance of the residual and the coefficient in the subgraph induced on $\{X_i\} \cup X_S$. Due to the faithfulness assumption, the correlations will not be cancelled out, and hence, β and σ^2 depend on $\tilde{\sigma}_i^2$ and $\tilde{B}_{i,j}$, which in turn depend on σ_i^2 and $B_{i,j}$. Therefore, if , say, $B_{i,j}$ remains fixed and σ_i^2 varies across two domains $D^{(i)}$ and $D^{(j)}$, then similar to the proof of Theorem 9, we will have

$$\mathcal{T}_{\pi^*}(\mathcal{D}) < \mathcal{T}_{\pi'}(\mathcal{D}).$$

B.5 An Example For Requirement of considering both orders $\pi_{X,-1}$ and $\pi_{X,-2}$ in Algorithm 10

Suppose the ground truth structure is $X_1 \to X_2 \to X_3 \to X_4 \to X_5$, and suppose we start with initial ordering $\pi_t = \{1, 5, 4, 2, 3\}$. If Algorithm 10 does not consider $\pi_{X,-2}$, the following can happen:

Round 1: Algorithm 10 forms $\Pi_{X_1} = \{\pi_t, \pi_{X_1,-1}\}$. We have $\pi_t = \arg\min_{\pi \in \Pi_{X_1}} \mathcal{T}_\pi^{MC}(\mathcal{D})$. Therefore, the ordering will not change.

Round 2: Algorithm 10 forms $\Pi_{X_5} = \{\pi_t, \pi_{X_5,-1}\}$. We have $\pi_{X_5,-1} = \arg\min_{\pi \in \Pi_{X_5}} \mathcal{T}_\pi^{MC}(\mathcal{D})$. Therefore, the ordering will change to $\pi_t = \{1, 4, 2, 3, 5\}$.

Round 3: Algorithm 10 forms $\Pi_{X_4} = \{\pi_t, \pi_{X_4,-1}\}$. We have $\pi_{X_4,-1} = \arg\min_{\pi \in \Pi_{X_4}} \mathcal{T}_\pi^{MC}(\mathcal{D})$. Therefore, the ordering will change to $\pi_t = \{1, 2, 3, 5, 4\}$.

Round 4: Algorithm 10 forms $\Pi_{X_2} = \{\pi_t, \pi_{X_2,-1}\}$. We have $\pi_t = \arg\min_{\pi \in \Pi_{X_2}} \mathcal{T}_\pi^{MC}(\mathcal{D})$. Therefore, the ordering will not change.

Round 5: Algorithm 10 forms $\Pi_{X_3} = \{\pi_t, \pi_{X_3,-1}\}$. We have $\pi_t = \arg\min_{\pi \in \Pi_{X_3}} \mathcal{T}_\pi^{MC}(\mathcal{D})$. Therefore, the ordering will not change.

Therefore Algorithm 10 outputs $\pi_t(-1) = 4$ as a sink variable while it is not a sink.

B.6 Proof of Theorem 11

Since X_s is a sink variable, by moving it to the last position in the order, none of its ancestors will be regressed on it, and hence, this move minimizes the dependencies among estimated regression parameters, which in turn minimizes the number of varying parameters. Therefore, for all $\pi \in \Pi_{X_s}$, $\mathcal{T}^{MC}_{\pi_{X_s,-1}}(\mathcal{D}) \leq \mathcal{T}^{MC}_{\pi}(\mathcal{D})$.

Suppose in the initial order π_t, there is a sink variable X_s as $\pi_t(-1)$. Then for any other variable X_v, moving it to $\pi_t(-1)$ either increases the dependencies or if , say, X_v is also a sink variable, will not change it. Therefore, based on our prioritization, X_s will remain in position $\pi_t(-1)$ until the end of the round. If in the initial order there is a non-sink variable as $\pi_t(-1)$, when we are checking its source ancestor X_s, since there exists a pair of domains across which at least 1 and at most 2 of variables $Var(X_v)$, $B_{v,s}$, σ^2_s varies, moving X_s below X_v will increase the value of the causal order indicator; that is, for all $\pi \in \Pi_{X_s}$, $\mathcal{T}^{MC}_{\pi_{X_s,-1}}(\mathcal{D}) > \mathcal{T}^{MC}_{\pi}(\mathcal{D})$. Therefore, X_s will move to the bottom of the order, and similar to the previous case, it will remain at that position until the end of that round. Therefore, in either case, at the end of round, $\pi_t(-1)$ will be a sink variable.

APPENDIX C

APPENDIX OF CHAPTER 5

C.1 Proof of Proposition 6

Two DAGs are I-equivalent if and only if they have the same skeleton and v-structures [16]. Therefore, it suffices to show that two DAGs G_1 and G_2 are distribution equivalent if and only if they have the same skeleton and v-structures.

By Corollary 3, DAGs G_1 and G_2 are equivalent if and only if there exist sequences of parent exchanges that map them to one another. Suppose G_1 and G_2 are distribution equivalent. Therefore there exists a sequence of parent exchanges mapping one to another. Since DAGs do not have 2-cycles, parent exchange for them will only result in flipping an edge, and since the other parents of the vertices at the two ends of that edge should be the same, it does not generate or remove a v-structure. Therefore, the sequence of parent exchanges does not change the skeleton or change the set of v-structures. Therefore, G_1 and G_2 are I-equivalent.

If two DAGs G_1 and G_2 have the same skeleton and v-structures, then their difference can be demonstrated as a sequence of edge flips such that in each flip, all the parent of the two ends have been the same, which means this flip is a parent exchange. Therefore, by Corollary 3, DAGs G_1 and G_2 are distribution equivalent.

C.2 Proof of Proposition 7

If side:
If $\text{supp}(Q_1U^{(1)}) \subseteq \text{supp}(Q_{G_2})$, then we can simply choose the entries of $Q_1U^{(1)}$ as the entries of Q_2 (as they are all free variables). Therefore,

$$Q_2Q_2^\top = Q_1U^{(1)}(U^{(1)})^\top Q_1^\top = Q_1Q_1^\top.$$

That is, Q_2 can generate the distribution which was generated by Q_1. Since this is true for all choices of Q_1, and since the reverse (i.e., starting with Q_2) is also true, by definition, G_1 is distribution equivalent to G_2.

Only if side:
If G_1 is distribution equivalent to G_2, then for all choices of Q_1, generating $Q_1Q_1^\top = \Theta$, there exists Q_2 generated by G_2, such that $Q_2Q_2^\top = \Theta$. Since Q_2 is generated by G_2, by definition, $\text{supp}(Q_2) \subseteq \text{supp}(Q_{G_2})$. Also, since $Q_1Q_1^\top = \Theta$ and $Q_2Q_2^\top = \Theta$, we have $Q_2 = Q_1U$, for some orthogonal transformation U, due to the fact that the generating vectors of a Gramian matrix can be determined up to isometry. Therefore, since $Q_2 = Q_1U$ and $\text{supp}(Q_2) \subseteq \text{supp}(Q_{G_2})$, we conclude that $\text{supp}(Q_1U) \subseteq \text{supp}(Q_{G_2})$. It remains to show that there exists a rotation $U^{(1)}$, for which $\text{supp}(Q_1U^{(1)}) \subseteq \text{supp}(Q_{G_2})$. Note that U is an orthogonal transformation and hence, $UU^\top = I$ and $\det(U) = 1$ or -1.

- If $\det(U) = 1$, it means that U is a rotation and we are done by choosing $U^{(1)} = U$.
- If $\det(U) = -1$ (i.e., U is an improper rotation), all we need is to find an orthogonal transformation V, such that (a) $\text{supp}(Q_1U) = \text{supp}(Q_1UV)$, i.e., it does not change the support, (b) $\det(V) = -1$, which implies that $\det(UV) = 1$. That is, adding the transformation V to U does not change the support but makes the combination UV into a rotation. Finding such a V is easy: simply choosing a diagonal matrix with an odd number of diagonal entries equal to -1 and the rest equal to 1. This will not change the support and only changes the sign of a subset of the entries. Therefore, we are done by choosing $U^{(1)} = UV$. Note that we are not forced to add a specific reflection at the end; we just add a particular one to do a sign flipping to show that the improper rotation can be changed into a rotation.

C.3 Proof of Proposition 8

- If $\xi_{i,j} = 0$, then by definition, the Givens rotation corresponding to $A(i, j, k)$ is a zero degree rotation. Therefore, applying $A(i, j, k)$ has no effect.
- If $\xi_{i,j} = \xi_{i,k} = \times$, then there exists a matrix Q for which zeroing $\xi_{i,j}$ is an acute rotation and the other rows of Q either have no element in the (j, k) plane, or if they do, they will not become aligned with either j or k axis in the (j, k) plane after the rotation. Therefore, support $(0, 0)$ will stay at $(0, 0)$, and any other support will become $(\times, \times)$.

- If $\xi_{i,j} = \times$ and $\xi_{i,k} = 0$, then the i-th row has been aligned with the j axis in the (j,k) plane before the rotation and since the rotation is planar, will become aligned with the k axis after the rotation, and hence we have a $\pi/2$ rotation. Therefore, all other rows aligned with one axis will become aligned with the other axis, and any vector not aligned with either axes will remain the same. Therefore, we have support transformations $(\times, 0) \to (0, \times)$, $(0, \times) \to (\times, 0)$, $(\times, \times) \to (\times, \times)$, and $(0, 0) \to (0, 0)$, which is equivalent to switching columns j and k.

C.4 Proof of Theorem 12

We first prove the following weaker result:

Theorem 19. *Let ξ_1 and ξ_2 be the support matrices of directed graphs G_1 and G_2, respectively. G_1 is distribution equivalent to G_2 if and only if both following conditions hold:*

- *There exists a sequence of support rotations that maps ξ_1 to a subset of ξ_2.*
- *There exists a sequence of support rotations that maps ξ_2 to a subset of ξ_1.*

We need the following lemma for the proof.

Lemma 17. *Consider a matrix Q and a support matrix ξ. If the support matrix of Q is a subset of ξ, then for all i, j, k, the support matrix of $QG(j,k,\theta)$ is subset of $\xi A(i,j,k)$, where,*

$$\theta = \begin{cases} 0, & \text{if } Q_{i,j} = Q_{i,k} = 0 \text{ and } \xi_{i,j} = \xi_{i,k} \neq 0, \\ 0, & \text{if } Q_{i,j} = Q_{i,k} = 0 \text{ and } \xi_{i,k} \neq \xi_{i,j} = 0, \\ \pi/2, & \text{if } Q_{i,j} = Q_{i,k} = 0 \text{ and } \xi_{i,j} \neq \xi_{i,k} = 0, \\ \tan^{-1}(-Q_{i,j}/Q_{i,k}), & \text{otherwise.} \end{cases}$$

Proof. The rotation and the support rotation do not alter any columns except the j-th and k-th columns. Hence we only need to see if the desired property is satisfied by those two columns. If the support of Q and ξ are the same on those two columns, the desired result follows from the definition of support rotation. Otherwise,

- If the support of $(Q_{i,j}, Q_{i,k})$ is the same as $(\xi_{i,j}, \xi_{i,k})$, then the effect of the rotation on Q is the same as the effect of the support rotation on ξ, except that if we are in the second case of Proposition 8, the support rotation cannot introduce any extra zeros

in rows $[p] \setminus \{i\}$, while this is possible for the rotation on Q. Therefore, the support matrix of $QG(j,k,\theta)$ is subset of $\xi A(i,j,k)$.

- If $Q_{i,j} \neq 0$ and $Q_{i,k} = 0$, and $(\xi_{i,j}, \xi_{i,k}) = (\times, \times)$, then the rotation is a $\pm\pi/2$ while we have an acute rotation for ξ (second case of Proposition 8). Hence, if a zero entry of Q in a row in $[p] \setminus \{i\}$ has become non-zero after the rotation, ξ has non-zero entries in both entries of that row. Therefore, the support matrix of $QG(j,k,\theta)$ is subset of $\xi A(i,j,k)$.

- If $[Q_{i,j} = 0$ and $Q_{i,k} \neq 0$, and $(\xi_{i,j}, \xi_{i,k}) = (\times, \times)]$, or $[Q_{i,j} = 0$ and $Q_{i,k} = 0$, and $(\xi_{i,j}, \xi_{i,k}) = (0, \times)]$, or $[Q_{i,j} = 0$ and $Q_{i,k} = 0$, and $(\xi_{i,j}, \xi_{i,k}) = (\times, \times)]$, then the rotation has no effect on Q, while the support rotation can only turn some of the zero entries in rows $[p] \setminus \{i\}$ to non-zero. Therefore, the support matrix of $QG(j,k,\theta)$ is subset of $\xi A(i,j,k)$.

- Finally, if $[Q_{i,j} = 0$ and $Q_{i,k} = 0$, and $(\xi_{i,j}, \xi_{i,k}) = (\times, 0)]$, then by the statement of the lemma, the rotation on Q will be $\pi/2$. Due to this fact and part three of Proposition 8, for both Q and ξ, columns j and k will be flipped. Therefore, the support matrix of $QG(j,k,\theta)$ is subset of $\xi A(i,j,k)$.

□

Proof of Theorem 19. By Propositions 7, it suffices to show that there exists a sequence of support rotations $A_1, \cdots A_m$, such that $\xi_1 A_1, \cdots A_m \subseteq \xi_2$ if and only if for all choices of Q_1, there exists a sequence of Givens rotations $G_1, \cdots G_{m'}$ such that $\text{supp}(Q_1 G_1, \cdots G_{m'}) \subseteq \text{supp}(Q_{G_2})$.

Only if side:

For any matrix Q_1, by definition, the support matrix of Q_1 is a subset of ξ_1. In the sequence of support rotations, use the first support rotation $A_1(i,j,k)$ to generate Givens rotation $G_1(j,k,\theta)$, where θ is defined in the statement of Lemma 17. Therefore, by Lemma 17, the support matrix of $Q_1 G_1(j,k,\theta)$ is a subset of $\xi_1 A_1(i,j,k)$. Repeating this procedure, we see that the support matrix of $Q_1 G_1, \cdots G_m$ is a subset of $\xi_1 A_1, \cdots A_m$. Now, by the assumption, $\xi_1 A_1, \cdots A_m \subseteq \xi_2$, and by definition, $\text{supp}(\xi_2) = \text{supp}(Q_{G_2})$. Therefore, $\text{supp}(Q_1 G_1, \cdots G_m) \subseteq \text{supp}(Q_{G_2})$.

If side:

Consider Givens rotation $G(j,k,\theta)$ applied to matrix Q. The effect of this rotation is one of the following:

1. For an acute rotation, zeroing a subset of entries in columns j and k.
2. For a $\pm\pi/2$ rotation, swapping the support of columns j and k.
3. For an acute rotation, making no entries zero, while making a subset of the entries in columns j and k non-zero.
4. For an acute rotation, no change to $\text{supp}(Q)$.

Since the assumption is true for all Q, we focus on matrices with support matrix ξ_1 (i.e., none of the free parameters are set at zero). If in case 1 above the subset has more than one element, more than one rows of Q have been aligned on the (j,k) plane, not on the j and k axes. Therefore, there exists another Q (i.e., another choice of free parameters), in which those rows are not aligned. Consider Q^* for which no such alignment happens, and hence, each of the Givens rotations in its sequence of rotations that causes case 1 above, only makes one entry zero. Therefore, its corresponding sequence of rotations acts exactly the same as support rotations for effects 1 and 2 above, in terms of their effect on the support.

Hence, the proof is complete by showing that cases 3 and 4 can be ignored, because we assumed that the support matrix of Q^* is ξ_1, and each not ignored Givens rotation corresponds to a support rotation, and by definition, $\text{supp}(Q_{G_2}) = \text{supp}(\xi_2)$. Clearly, case 4 can be ignored as it has no effect on the support. For case 3, we note that this effect only adds elements to the support, and hence we want the support after rotations to be a subset of $\text{supp}(Q_{G_2})$, the rotations of this type do not serve for that purpose. Therefore, if we ignore such rotations, the resulting support would be smaller compared to the case of considering these rotations. Note that if due to such rotation entry $Q_{i,j}$ has become non-zero and later in the sequence there exists a type 1 rotation making $Q_{i,j}$ zero again, we already have zero in position (i,j) and that type 1 rotation should be ignored as well.

□

Similar to the notion of distribution set, for a support matrix ξ we define

$$\Theta(\xi) \coloneqq \{\Theta : \Theta = \tilde{Q}\tilde{Q}^\top, \textit{ for any } \tilde{Q} \textit{ s.t. } \text{supp}(\tilde{Q}) \subseteq \text{supp}(\xi)\}.$$

Note that unlike Q, the matrix $\tilde{Q}$ is allowed to have zeros on its diagonal.

Definition 28. *A support rotation mapping ξ to ξ' is lossless if $\Theta(\xi) = \Theta(\xi')$.*

Similar to the test for distribution equivalence, losslessness can be evaluated by checking if there exists a sequence of support rotations that maps ξ' back to a subset of ξ. Clearly,

reduction, reversible acute rotation, and column swap are lossless, as they are reversible. In most of the cases, irreversible acute rotations are lossy and lead to expansion of $\Theta(\xi)$, as it introduces capacity for having extra free variables. However, this is not necessarily the case.

We have the following observations regarding checking for distribution equivalence.

Lemma 18. *All the support rotations for checking the distribution equivalence of two directed graphs should be lossless.*

We need the following lemma for the proof.

Lemma 19. *If support matrix ξ is mapped to ξ' via a support rotation, then $\Theta(\xi) \subseteq \Theta(\xi')$.*

Proof. For reduction, reversible acute rotation, and column swap, we have $\Theta(\xi) = \Theta(\xi')$, and irreversible acute rotation only introduces extra free variables, and hence, leads to $\Theta(\xi) \subseteq \Theta(\xi')$. To make the argument regarding irreversible acute rotation rigorous, consider irreversible acute rotation $A(i, j, k)$, which zeros $\xi_{i,j}$. For all $l \in [p] \setminus \{i\}$, if $\xi_{l,j} \neq \xi_{l,k}$, this rotation results in $(\xi_{l,j}, \xi_{l,k}) = (\times, \times)$. Suppose $(\xi_{i',j}, \xi_{i',k}) = (0, \times)$. $A(i', j, k)$ will be a reversible acute rotation for ξ' and leads to ξ'' such that $\xi \subsetneq \xi''$. Therefore, $\Theta(\xi) \subseteq \Theta(\xi'') = \Theta(\xi')$.

□

Proof of Lemma 18. If support matrix ξ is mapped to ξ' via a lossy support rotation, i.e., $\Theta(\xi) \neq \Theta(\xi')$ then by Lemma 19, we have $\Theta(\xi) \subsetneq \Theta(\xi')$. Suppose we want to check the equivalence of directed graphs G_1 and G_2 with support matrices ξ_1 and ξ_2, respectively. We note that $\Theta(G_1) = \Theta(\xi_1)$. Suppose ξ_1 is mapped to ξ through a sequence of support rotations, including a lossy rotation, which in turn is mapped to $\xi' \subseteq \xi_2$. Therefore,

$$\Theta(G_1) = \Theta(\xi_1) \subsetneq \Theta(\xi) \subseteq \Theta(\xi') \subseteq \Theta(\xi_2) = \Theta(G_2).$$

Therefore,

$$\Theta(G_1) \neq \Theta(G_2).$$

□

Using Lemma 18, we can prove Theorem 12:

Proof. The if side is clear by Theorem 19. For the only if side, by Theorem 19 and Lemma 18 we show that if ξ_1 can be mapped to ξ_2 via a sequence of lossless support rotations (i.e., $\Theta(\xi_1) = \Theta(\xi_2)$) including an irreversible acute rotation, then there exists a sequence of

support rotations which does not include any irreversible acute rotations that maps ξ_1 to a subset of ξ_2.

We show that every irreversible acute rotation can be replaced by other types of support rotation. Consider the first irreversible acute rotation $A(i,j,k)$ in the sequence, which maps ξ to ξ'. Applying this rotation, we have $(\xi'_{i,j}, \xi'_{i,k}) = (0, \times)$, and columns $\xi'_{\cdot,j}$ and $\xi'_{\cdot,k}$ agree on the rest of the entries. Suppose, prior to applying this rotation, columns $\xi_{\cdot,j}$ and $\xi_{\cdot,k}$ disagree on m entries in rows with indices $\mathit{diff} = \{s_1, \cdots, s_m\}$. Let

$$\mathit{diff}_j = \{l : l \in \mathit{diff}, \xi_{l,j} = 0\},$$

$$\mathit{diff}_k = \{l : l \in \mathit{diff}, \xi_{l,k} = 0\},$$

and

$$M = \begin{cases} \max\{m_j, m_k\}, & m_j \neq m_k, \\ m_j + 1, & \mathit{otherwise}. \end{cases}$$

where $m_j = |\mathit{diff}_j|$ and $m_k = |\mathit{diff}_k|$. We can always swap two columns, hence, without loss of generality, assume $M = m_j + \mathbb{1}_{\{m_j = m_k\}}$.

Claim 1. *ξ can be transformed via reduction and reversible acute rotation to a support matrix, in which there exist columns with indices $\{t_1, \cdots, t_{M-1}\}$ such that the sub-matrix of ξ on columns $\{t_1, \cdots, t_{M-1}, j, k\}$ and rows $\mathit{diff} \cup \{i\}$ has a column with i zeros, for all $i \in \{0, 1, ..., M\}$, and the sub-matrix of ξ on columns $\{t_1, \cdots, t_{M-1}, j, k\}$ and the rest of the rows has equal columns.*

Proof of Claim 1. Since $A(i,j,k)$ is lossless, we can map ξ' to a subset of ξ. Therefore, we should be able to introduce zeros in ξ' in indices diff_j of column j and indices diff_k of column k, without removing the existing zeros, except potentially ξ'_{ij}. We first use a reversible acute rotation on columns j and k to move the newly introduce zero in ξ'_{ij} to the first index in diff_j, and we denote the resulting support matrix by $\xi^{(1)}$. We note that reduction is the only support rotation, which increases the number of zeros in the support matrix. Therefore, we need one reduction for reviving each of the $m-1$ other removed zeros in the transformation of ξ to ξ'.

The claim can be proven by induction. The base of the induction, i.e., for $M = 2$ can be proven as follows:

- **Case 1:** $m_j = m_k = 1$. In order to have the zero in column k, we need to perform a reduction, for which, we need another column $\xi^{(1)}_{\cdot,t_1}$ equal to $\xi^{(1)}_{\cdot,k}$, i.e., $d_H(\xi^{(1)}_{\cdot,t_1}, \xi^{(1)}_{\cdot,k}) = 0$,

where $d_H(\cdot,\cdot)$ denotes the Hamming distance between its two arguments. Since the original irreversible acute rotation was on the (j,k) plane and did not affect other columns, the column t_1 with the aforementioned property exists in the original support matrix ξ as well, i.e., $\xi_{\cdot,t_1} = \xi^{(1)}_{\cdot,t_1}$. Now, a reversible acute rotation can be performed on columns t_1 and k to set $d_H(\xi_{\cdot,j}, \xi_{\cdot,j}) = 0$, and then a reduction can be performed to introduce another zero in column j of ξ. The resulting support matrix has the desired property stated in the claim.

- **Case 2:** $m_j = 2, m_k = 0$. In order to have the zero in the second index of diff_j, we need to perform a reduction, for which, we need another column equal to $\xi^{(1)}_{\cdot,j}$. This can be obtained by one of the following cases:
  - There already exists a column t_1, such that $d_H(\xi^{(1)}_{\cdot,t_1}, \xi^{(1)}_{\cdot,j}) = 0$. Similar to Case 1, This implies that column t_1 also exists in ξ. Therefore, ξ has the desired property.
  - There exists a column t_1, such that $d_H(\xi^{(1)}_{\cdot,t_1}, \xi^{(1)}_{\cdot,j}) \neq 0$, but $d_H(\xi^{(1)}_{\cdot,t_1}, \xi^{(1)}_{\cdot,k}) = 1$. Similar to Case 1, This implies that column t_1 also exists in ξ. Therefore, a reversible acute rotation can transform ξ to a support matrix with the desired property.
  - There exists a column t_1, such that $d_H(\xi^{(1)}_{\cdot,t_1}, \xi^{(1)}_{\cdot,k}) = 0$. Similar to Case 1, This implies that column t_1 also exists in ξ. Therefore, two reductions, one on columns (t_1,k), and then one on columns (t_1,j) can transform ξ to a support matrix with the desired property.
- **Case 3:** $m_j = 2, m_k = 1$. In order to have the zero in column k, we need to perform a reduction, for which, we need another column t_1 equal to column k, i.e., $d_H(\xi^{(1)}_{\cdot,t_1}, \xi^{(1)}_{\cdot,k}) = 0$. Similar to Case 1, This implies that column t_1 also exists in ξ. Therefore, ξ has the property desired in the claim.

Now, suppose the property holds for $M = n$. To show that it also holds for $M = n+1$, a reasoning same as the one provided for the base case of the induction can be used, and it can be shown that for the required extra reduction, an extra column t_n should exist in ξ.

□

By Claim 1, ξ can be transformed via reduction and reversible acute rotation to a support matrix with the stated property. Therefore, we assume ξ has the property. Therefore, we have columns $\{t_1, \cdots, t_{M-1}, j, k\}$ with any number of zeros $0 \leq i \leq M$ on rows $\mathit{diff} \cup \{i\}$,

and it is easy to see the i zeros in these columns can be relocated to any other indices via only reversible acute rotations amongst these columns. Therefore, any effect sought to be achieved via columns j and k of ξ', can be obtained via columns $\{t_1, \cdots, t_{M-1}, j, k\}$ of ξ, and hence, the irreversible acute rotation could have been replaced by other types of rotations.

□

C.5 Proof of Proposition 9

To show that the property holds for cycle $C = (X_1, \cdots, X_m, X_1)$, we note that our desired support matrix is ξ_1, when columns 2 to m are all shifted to left by one, and column 1 is moved to location m. Therefore, it suffices to first flip columns 1 and 2, then 2 and 3, all the way to $m-1$ and m. For each flip, we use the third part of Proposition 8. For instance, for flipping columns j and $j+1$, we find row i such that $\xi_{i,j} \neq \xi_{i,j+1}$ (if there is no such row, then no flip for those columns is needed as they are already the same). If, say $\xi_{i,j} = \times$, we use support rotation $A(i, j, j+1)$ for flipping columns j and $j+1$. Following the same reasoning, we see that support rotation of ξ_2 leads to a subset of ξ_1.

C.6 Proof of Proposition 10

If side:
If columns of ξ_2 are permutation of columns of ξ_1, then ξ_1 can be mapped to ξ_2 and vice versa via a sequence of column swap rotations. Therefore, by Theorem 12, $G_1 \equiv G_2$.

Only if side:
If $G_1 \equiv G_2$, the by Theorem 12, ξ_1 can be mapped to a subset of ξ_2 and ξ_2 can be mapped to a subset of ξ_1, both via only reductions, reversible acute rotations and column swaps. If each pair of column of ξ_1 are different in more than one entry, then we are not able to perform any reversible acute rotations and reductions. Therefore, we have been able to perform the mapping merely via column swaps. Therefore, columns of ξ_2 are permutation of columns of ξ_1.

C.7 Proof of Proposition 11

Only if side:
By definition, digraph G is reducible if there exists digraph G' such that $G \equiv G'$ and $\xi' \subset \xi$. By Theorem 12, ξ can be mapped to a subset of ξ' via a sequence of support rotations comprised of reductions, reversible acute rotations and column swaps. We note that reduction is the only support rotation, which increases the number of zeros in the support matrix. Therefore, there should be a reduction in the sequence. We can always swap any two columns and the location of two columns does not influence the feasibility of reduction or reversible acute rotations. Therefore, column swaps can be ignored in reducibility.

If side:
Suppose the performed reduction turns a non-zero entry in column j to zero, using a reduction on columns j and k. Note that prior to the reduction, these columns have the same number of zeros and in order to be able to perform the reduction a sequence of reversible acute rotations have been performed to prepare column k such that the hamming distance of columns j and k be equal to zero. That is, its zeros have been moved to match the zero pattern of column j. We can always assume that we only moved the zeros of column k, as if there are columns to move the zeros of column j, they can be used to move the zeros of column k as well. The only concern is that the zeroed entry may be on the diagonal. In this case, a reversible acute rotation can be performed on columns j and k to move the new zero to another index of column j. Also, entry (j, j) cannot be the only non-zero entry of column j; otherwise, column k should also have only one non-zero entry, which should initially be located at (k, k). Therefore, to perform a reversible acute rotation on any other column l and k, column l should have only two non-zero entries, on (k, l) and (j, l), while one of them should initially be located at (l, l). This reasoning can be repeated p times and leads to the contradiction that the final column is not allowed to have a non-zero entry on the diagonal, which contradicts the fact that ξ is the support matrix corresponding to a digraph. Finally, all the performed reversible acute rotations can be done in the reverse direction to obtain the initial zero pattern for columns $[p] \setminus \{j\}$.

C.8 Proof of Proposition 12

Using Proposition 11, we show that for directed graph G with support matrix ξ, if there exists a sequence of reversible support rotations that enables us to apply a reduction to ξ, then G has a 2-cycle. Suppose the reduction is performed on columns j and k, to turn a non-zero entry of column j to zero. If no reversible support rotations prior to the reduction is needed, it implies that already columns j and k are identical. Therefore, $\xi_{j,k} = \xi_{j,j} = \times$, and $\xi_{k,j} = \xi_{k,k} = \times$. Therefore, there exists a 2-cycle between j and k and the proof is complete. Therefore, we assume some reversible support rotations are needed.

Consider the first rotation in the sequence of reversible support rotations applied to column k. Assume it is performed on columns t_1 and k. Therefore, the support of column t_1 has one element more than the support of column k, and the Hamming distance between these two columns is one. The only way that this does not cause a 2-cycle between t_1 and k is that $\xi_{t_1,k} = 0$, and $\xi_{k,t_1} = \times$, and all the entries show be the same. This rotation is supposed to move the extra zero in column k to an index, which is zero in column j (to reduce the Hamming distance between columns j and k). Therefore, since after this rotation, $\xi_{t_1,k}$ will become non-zero, we should have $\xi_{t_1,j} = \times$. This will lead to a 2-cycle unless if $\xi_{j,t_1} = 0$. Now, if $\xi_{j,t_1} = 0$, because all the entries of columns t_1 and k where the same, we also have $\xi_{j,k} = 0$. This gives us two options for $\xi_{k,j}$:

- If $\xi_{k,j} = 0$, then we need another column t_2 so that we perform a reversible acute rotation on columns t_2 and k to move $\xi_{j,k} = 0$ to entry $\xi_{k,k}$, which is currently non-zero. This means that columns t_2 and k should be the same on all the entries, except that $\xi_{j,t_2} = \times$, but $\xi_{j,k} = 0$. Therefore, $\xi_{k,t_2} = \xi_{k,k} = \times$ and $\xi_{t_2,k} = \xi_{t_2,t_2} = \times$, which implies that there is a 2-cycle between t_2 and k.

- If $\xi_{k,j} = \times$, then in order for columns k and j to have the same number of non-zero entries, there should exist index l such that $\xi_{l,k} = \times$, and $\xi_{l,j} = 0$. Now, we need another column t_2 so that we perform a reversible acute rotation on columns t_2 and k to move $\xi_{j,k} = 0$ to entry $\xi_{l,k}$. This means that columns t_2 and k should be the same on all the entries, except that $\xi_{j,t_2} = \times$, but $\xi_{j,k} = 0$. Therefore, $\xi_{k,t_2} = \xi_{k,k} = \times$ and $\xi_{t_2,k} = \xi_{t_2,t_2} = \times$, which implies that there is a 2-cycle between t_2 and k.

C.9 Proof of Corollary 2

We first prove the following corollary:

Corollary 5. *Irreducible directed graphs G_1 and G_2 with support matrices ξ_1 and ξ_2 are equivalent if and only if there exist sequences of reversible acute rotations and column swaps that map their support matrices to one another.*

Proof. By Proposition 11, there exists no sequence of reversible acute rotations that enables us to apply a reduction to the support matrix. Therefore, we only need to consider reversible acute rotations and column swaps, and we need to map one support matrix to the other, rather than mapping it to a subset of the other.

□

Proof of Corollary 2. DAGs do not have 2-cycles. Therefore, by Proposition 12, DAGs are irreducible. Therefore, the result follows from Corollary 5.

□

C.10 Proof of Theorem 13

If side:
If there exist sequences of parent reduction, parent exchange, and cycle reversion, mapping one graph to a subgraph of the other, then there exist sequences of reduction, reversible acute rotation, and column swap mapping the support matrix of one graph to a subset of the support matrix of the other. Therefore, by Theorem 12, G_1 is distribution equivalent to G_2.

Only if side:
The proof of the only if side consists of two steps:

- **Step 1.** We note that

    1. All support rotations of reduction type, that do not make a diagonal entry zero are representable by a parent reduction. This is clear from the definitions of reduction and parent reduction.
    2. All reversible acute rotations, that do not make a diagonal entry zero are representable by a parent exchange. This is clear from the definitions of reversible acute rotation and parent exchange.

3. If we have a reversible acute rotation and a column swap on columns j and k such that the reversible acute rotation makes the diagonal entry $\xi_{j,j}$ zero and then the column swap swaps columns j and k (we call such a pair a flip pair), then this pair can be replaced by a reversible acute rotation that makes the non-diagonal entry $\xi_{j,k}$ zero, and hence, is representable by a parent exchange.

4. If we start with a support matrix with no diagonal entries equal to zero and by performing a sequence of column swaps reach another support matrix with no diagonal entries equal to zero, then this sequence is representable by a cycle reversion. To see this, we note that if after the sequence of column swaps, column j has moved to location k, it implies that its j-th and k-th elements are non-zero. Therefore, the original support matrix corresponds to a graph containing the edge $j \to k$, and the final support matrix corresponds to a graph containing the edge $k \to j$. This reasoning identifies the cycle before, and the reversed cycle after the transformation.

Step 1 implies that if we have a sequence of support rotations which includes 1. reduction rotations, that do not make a diagonal entry zero, 2. reversible acute rotations, that do not make a diagonal entry zero, 3. flip pairs, and 4. sequence of column swaps starting and ending on a support matrix with non-zero diagonal entries, (we call such a sequence, a representable sequence) then we can represent this sequence with a sequence of parent reductions, parent exchanges, and cycle reversions.

- **Step 2.** If G_1 is distribution equivalent to G_2, then by Theorem 12, there exists a sequence of reduction, reversible acute rotations, and column swap mapping the support matrix of one to the other. We show that in this case, there exists a representable sequence as well that maps the support matrix of one to the other. Therefore, by Step 1 the only if side will be concluded.

  We note that since ξ_1 is a support matrix of a directed graphs, it does not have any zeros on the main diagonal. Given the sequence of support rotations, the column swaps do not enable us or prevent us from performing reversible acute rotations and reductions, and merely change the indices of the columns. Therefore, we can have an equivalent sequence of support rotations, in which we have moved all the column swaps, except those involved in flip pairs, to the end of the sequence. Consider the first rotation in the sequence of the rotations which zeros out a diagonal entry. If this rotation is of reduction type and has zeroed out $\xi_{i,i}$ using columns i and j, then $\xi_{i,j}$

should have been non-zero. Therefore, we can instead replace it by zeroing $\xi_{i,j}$, and use column j instead of column i in the next steps. If this rotation is of reversible acute rotation type and has zeroed out $\xi_{i,i}$ using columns i and j, then $\xi_{i,j}$ should have been non-zero. Therefore, again we can instead replace it by zeroing $\xi_{i,j}$, and use column j instead of column i in the next steps. Therefore, we can perform all the reductions and reversible acute rotations and from ξ_1 obtain ξ_1', which does not have any zeros on the main diagonal, and via a sequence of column swaps can be mapped to a subset of ξ_2.

Now, we perform the reverse of that sequence of column swaps on ξ_2, which gives us a superset of ξ_1' (call it ξ_2''), and hence, does not have any zeros on the main diagonal. Therefore, since ξ_2 is a support matrix of a directed graph and hence, it also does not have any zeros on the main diagonal, by part 4 of Step 1, this is equivalent to a cycle reversion. ξ_2'' is a superset of ξ_1', and both ξ_2'' and ξ_1' are graphically representable. By Lemma 18, the corresponding directed graph of ξ_2'' is the same (if the directed graph corresponding to ξ_2'' is irreducible) or reducible to the directed graph corresponding to ξ_1'. Therefore, by Proposition 11 we can perform the reduction via a sequence of reversible acute rotations. Similar to the reasoning in the previous paragraph, since we start with a support matrix with no zeros on the main diagonal, this can be done without zeroing any element of the main diagonal, and hence, we can map ξ_2'' to ξ_1'. Finally, reversing the reversible acute rotations of the sequence from ξ_1 to ξ_1', we obtain a subset of ξ_1, and the whole sequence from ξ_2 to a subset of ξ_1 is a representable sequence. Similarly, we can construct a representable sequence mapping ξ_1 to a subset of ξ_2, which completes the proof.

C.11 Proof of Corollary 3

DAGs do not have 2-cycles. Therefore, by Proposition 12, DAGs are irreducible. Hence, a parent reduction cannot be performed. Also, DAGs do not have cycles. Hence, there will not be any cycle reversions. Therefore, the result follows from Theorem 13.

C.12 Proof of Proposition 13

To violate faithfulness, there are finite number of sets of hard constraints that should be satisfied (since hard constraints are distributional constraints and hence limited). Let θ_i be the set of values satisfying the i-th set of constraints. By the definitions of hard constraints, θ_i is Lebesgue measure zero. Therefore, the set of distributions not Gen-faithful to G, which is the finite union is also Lebesgue measure zero.

C.13 Proof of Proposition 14

Suppose G^* is the ground truth DG and it generates distribution Θ, and G_1 is a candidate DG which we want to decide whether it is the ground truth or not.

Suppose $G_1 \cong G^*$. Then there exists a set of distribution with non-zero Lebesgue measure that both G_1 and G^* can generate. Suppose Θ is a distribution coming from this intersection which also satisfies Assumption 8. Then clearly, since both DGs can generate Θ, there is no way to realize which one has been the ground truth, and hence, G_1 is non-identifiable from G^*.

For the opposite direction, suppose $G_1 \ncong G^*$ then either there is no distribution that they can both generate, or the measure of such distributions is zero. In the first case, Θ is not generatable by G_1 and hence we can identify that G_1 is not the ground truth. In the second case, by Assumption 8, Θ cannot be from the intersection and hence again is not generatable by G_1 and hence we can identify that G_1 is not the ground truth.

C.14 Proof of Theorem 14

Let G^* and Θ be the ground truth structure and the generated distribution, and for an ML estimator, assume we are capable of finding a correct pair $(\hat{B}_{ML}, \hat{\Omega}_{ML})$, such that $(I - \hat{B}_{ML})\hat{\Omega}_{ML}^{-1}(I - \hat{B}_{ML})^\top = \Theta$ and denote the directed graph corresponding to $\hat{B}_{ML}$ by $\hat{G}_{ML}$. We have $\Theta \in \Theta(\hat{G}_{ML})$, which implies that Θ contains all the distributional constraints of $\hat{G}_{ML}$. Therefore, under Assumption 8, we have $H(\hat{G}_{ML}) \subseteq H(G^*)$.

Let $(\hat{B}_{\ell_0}, \hat{\Omega}_{\ell_0})$ be the output of ℓ_0-regularized ML estimator, and denote the directed graph corresponding to $\hat{B}_{\ell_0}$ by $\hat{G}_{\ell_0}$. Since the likelihood term increases much faster with the sample size compared to the penalty term, asymptotically, we still have the desired properties that

Θ contains all the distributional constraints of $\hat{G}_{\ell_0}$, and hence, under Assumption 8, we again have $H(\hat{G}_{\ell_0}) \subseteq H(G^*)$.

Now, consider an irreducible equivalent of G^*, denoted by $G^\dagger$. Since $H(G^*) = H(G^\dagger)$, we have $H(\hat{G}_{\ell_0}) \subseteq H(G^\dagger)$. Also, because of the penalty term we have $|E(\hat{G}_{\ell_0})| \leq |E(G^\dagger)|$, otherwise the algorithm would have outputted $G^\dagger$. Therefore, by Assumption 8, we have $H(\hat{G}_{\ell_0}) = H(G^\dagger)$, and hence $H(\hat{G}_{\ell_0}) = H(G^*)$. Therefore, by definition, $\hat{G}_{\ell_0} \cong G^*$.

C.15 Algorithm for Enumerating Members of a Distribution Equivalence Class and Determining the Equivalence of Two Structures

We first propose an algorithm for enumerating members of the distribution equivalence class of a directed graph with support matrix ξ, based on a depth-first traversal. The algorithm is based on a search tree that is rooted at ξ and branches out via Reduction and AcuteRotation operations. These two operations are defined in Algorithm 12. Since those two rotation operations are independent of column swaps, we perform a similar depth-first traversal of column swaps at the end, leveraging the graphical, cycle reversion representation for efficiency.

Algorithm 12 Reduction and Acute Rotation Operations

function Reduction(ξ, i, j)
    Initialize $\xi' \leftarrow \xi$
    $\xi'_{i,j} \leftarrow 0$
    return ξ'
end function

function AcuteRotation(ξ, i, j, k, ℓ)
    Initialize $\xi' \leftarrow \xi$
    $\xi'_{i,j} \leftarrow 0$
    $\xi'_{\ell,j} \leftarrow 1$
    $\xi'_{\ell,k} \leftarrow 1$
    return ξ'
end function

Each vertex in the search tree corresponds to a support matrix and each of its children corresponds to the outputs of an admissible Reduction and AcuteRotation operation.

Algorithm 13 represents the pseudo-code of the function which compiles a set of those operations for a given support matrix.

Algorithm 13 Finding Legal Rotations

```
1: function FindRotations(ξ)
2:     Initialize Rotations = ∅
3:     // Find Legal Reductions
4:     for j, k such that ‖ξ_{·,j} − ξ_{·,k}‖_1 = 0 do
5:         for i such that ξ_{i,j} = 1 do
6:             if i ≠ j then
7:                 Rotations ← Rotations ∪ {Reduction(ξ, i, j)}
8:             end if
9:             if i ≠ k then
10:                Rotations ← Rotations ∪ {Reduction(ξ, i, k)}
11:            end if
12:        end for
13:    end for
14:    // Find Legal Acute Rotations
15:    for j, k such that ‖ξ_{·,j} − ξ_{·,k}‖_1 = 1 do
16:        ℓ ← index such that ξ_{ℓ,j} ≠ ξ_{ℓ,k}
17:        for i ≠ ℓ such that ξ_{i,j} = 1 do
18:            if i ≠ j then
19:                Rotations ← Rotations ∪ {AcuteRotation(ξ, i, j, k, ℓ)}
20:            end if
21:            if i ≠ k then
22:                Rotations ← Rotations ∪ {AcuteRotation(ξ, i, k, j, ℓ)}
23:            end if
24:        end for
25:    end for
26:    return Rotations
27: end function
```

Algorithm 14 enumerates the equivalence class. The algorithm keeps track of the search tree state using a stack S which contain sets of rotated support matrices. The first step of the algorithm enumerates a subset of the equivalence class of ξ^* by finding sequences of Reduction and AcuteRotation operations. The second step enumerates column swaps in a similar depth-first fashion. It is made efficient by using the fact that sequences of legal column swaps correspond to sequences of cycle reversions.

Algorithm 14 Enumerating equivalent structures

```
1: function ReverseCycles(ξ)
2:     Reversed ← ∅
3:     C ← list of cycles in ξ
4:     for C in C do
5:         ξ′ ← Column-permuted ξ with cycle C reversed
6:         Reversed ← Reversed ∪ {ξ′}
7:     end for
8:     return Reversed
9: end function
10:
11: procedure EnumerateEquiv(p × p support matrix ξ*)
12:     Initialize Equiv ← {ξ*}.
13:     Initialize empty stack S
14:     S.push(FindRotations(ξ*))
15:     while S is not empty do
16:         Rotations ← S.pop()
17:         if |Rotations| = 0 then
18:             continue
19:         else
20:             ξ ← a support matrix in the set Rotations
21:             Rotations ← Rotations \ {ξ}
22:             S.push(Rotations)
23:             if ξ not in Equiv then
24:                 Equiv ← Equiv ∪ {ξ}
25:                 S.push(FindRotations(ξ))
26:             end if
27:         end if
28:     end while
29:     // Enumerate legal column swaps via cycle reversion
30:     for ξ̃ in Equiv do
31:         Initialize empty stack S
32:         S.push(ReverseCycles(ξ̃))
33:         while S is not empty do
34:             Reversals ← S.pop()
35:             if |Reversals| = 0 then
36:                 continue
37:             else
38:                 ξ ← a support matrix in the set Reversals
39:                 Reversals ← Reversals \ {ξ}
40:                 S.push(Reversals)
41:                 if ξ not in Equiv then
42:                     Equiv ← Equiv ∪ {ξ}
43:                     S.push(ReverseCycles(ξ))
44:                 end if
45:             end if
46:         end while
47:     end for
48: end procedure
```

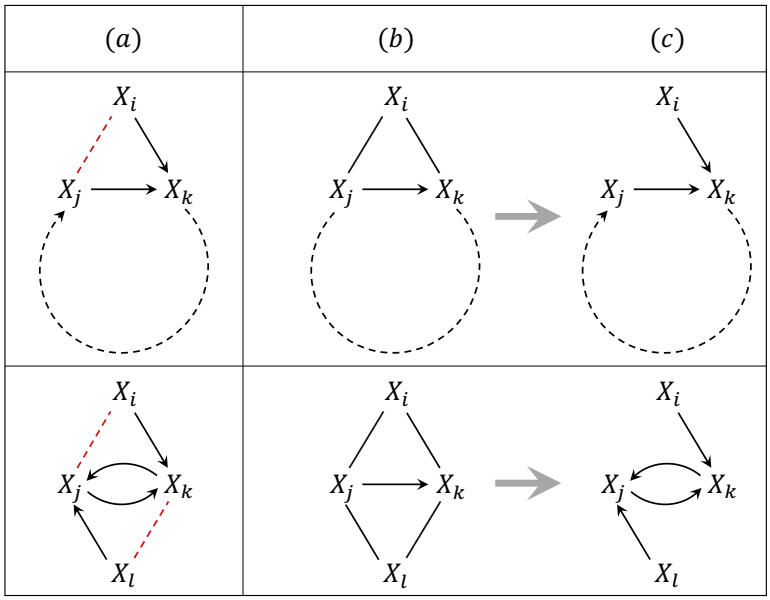

Figure C.1: Virtual edge search operator.

Finally, the procedure EnumerateEquiv in Algorithm 14 may be used to determine whether or not two DGs with respective support matrices ξ_1 and ξ_2 are equivalent by enumerating the equivalence class of ξ_1 and checking whether or not ξ_2 is in that equivalence class.

C.16 Virtual Edge Search Operator

For acyclic DGs, under the Markov and faithfulness assumptions, a variable X_i is adjacent to a variable X_j if and only if X_i and X_j are dependent conditioned on any subset of the rest of the variables. This is not the case for cyclic DGs [79]. Two non-adjacent variables X_i and X_j are dependent conditioned on any subset of the rest of the variables if they have a common child X_k which is an ancestor of X_i or X_j. In this case, we say there exists a virtual edge between X_i and X_j. Figure C.1(a) demonstrates two examples. In this figure, virtual edges are shown with dashed red edges.

There are two cases that detecting a virtual edge as a real edge can trap the greedy search into a local optima which can be improved.

Case 1. This case is shown in the first row of Figure C.1. If a greedy search algorithm finds the edges between X_k and X_j but does not find X_k and X_j to be on a cycle, that is, if it does not find the directions correctly, it can significantly increase the likelihood by adding an edge at the location of the virtual edge between X_i and X_j. The algorithm would therefore be trapped in a local optimum shown in Figure C.1(b) with one more edge than the ground truth shown in Figure C.1(c). To resolve this issue, we propose adding the following search operator: Suppose we have a triangle over three variables X_i, X_j and X_k, and there exists an additional sequence of edges connecting X_j and X_k. In one atomic move, we perform a series of edge reversals to form a cycle containing $X_j \to X_k$ along the sequence, delete the edge connecting X_i to X_j, and orient the edge $X_i \to X_k$. If the likelihood is unchanged, the edge deletion improves the score.

Case 2. This case is shown in the second row of Figure C.1. This case involves the case that the cycle over X_j and X_k in the ground truth is a 2-cycle. If a greedy search algorithm finds one edges between X_k and X_j, it can significantly increase the likelihood by adding edges at the location of the virtual edges between X_i and X_j and between X_l and X_k. The algorithm would therefore be trapped in a local optimum shown in Figure C.1(b) with one more edge than the ground truth shown in Figure C.1(c). To resolve this issue, we propose adding the following search operator: Suppose we have triangles over three variables X_i, X_j and X_k and X_l, X_j and X_k, as shown in the figure. In one atomic move, we delete the edge connecting X_i to X_j and the edge connecting X_l to X_k, and add the edge $X_k \to X_j$. If the likelihood is unchanged, the edge deletion improves the score.

In order to evaluate the proposed search operator, we performed two experiments. The first involves the ground truth structure shown in Figure C.2b, Graph 1. This graph has one equivalent structure, which is Graph 2 in the same figure. We run the tabu search algorithm with and without the proposed search operator for 100 instantiations of the edge weights and variances. The 5 most commonly found structures found by tabu search without and with the proposed operator are shown in Figures C.2a and C.2b, respectively. While the proposed algorithm finds an equivalent structure 89% of the time, the nominal tabu search never finds an equivalent structure.

Next, we consider the ground truth structure shown in Figure C.3b, Graph 1. This structure has one equivalent, which is Graph 2 in the same figure. While the nominal tabu search algorithm finds an equivalent structure 45% of the time, the proposed algorithm is

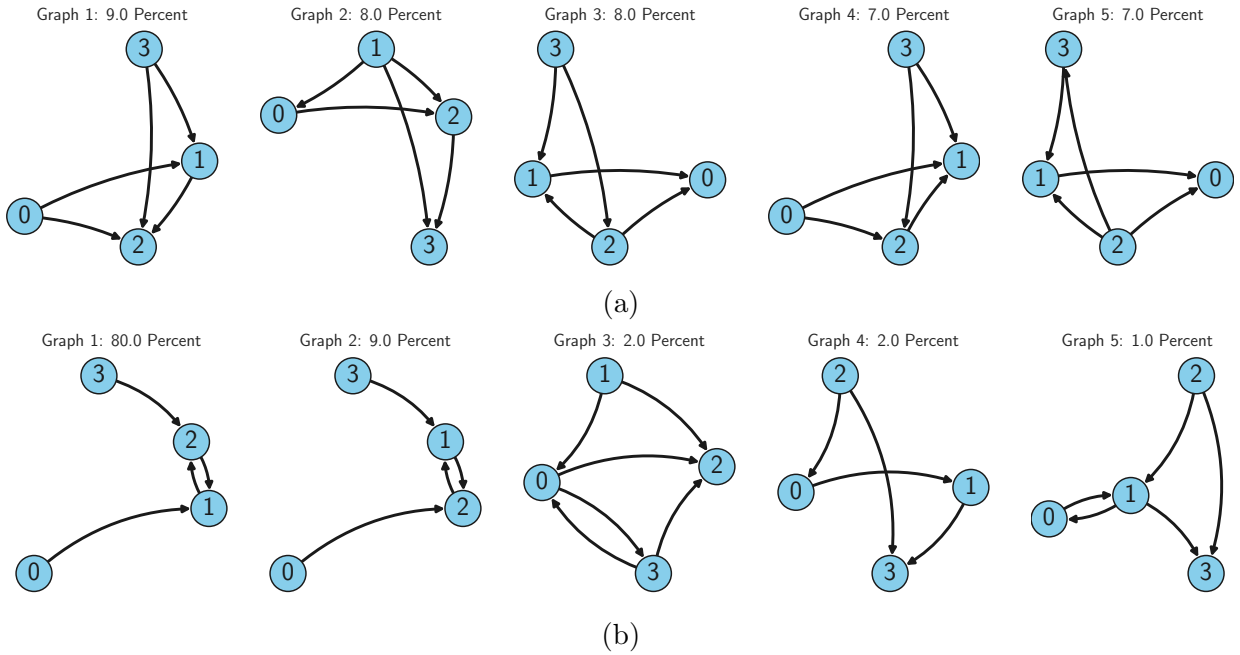

Figure C.2: Example 1. Comparison of 5 most commonly learned structures.

much more reliable, finding an equivalent structure 83% of the time.

C.17 Score Decomposability

When the DG is acyclic, the distribution generated by a linear Gaussian structural equation model satisfies the local Markov property. This implies that the joint distribution can be factorized into the product of the distributions of the variables conditioned on their parents as follows.

$$P(V) = \prod_{X_i \in V} P(X_i | Pa(X_i)).$$

The benefit of this factorization is that the computational complexity of evaluating the effect of operators can be dramatically reduced since a local change in the structure does not change the score of other parts of the DAG.

In contrast, for the case of cyclic DGs the distribution does not necessarily satisfy the local Markov property. However, the distribution still satisfies the global Markov property [73]. Therefore, our search procedure factorizes the joint distribution into the product of conditional distributions. Each of these distributions is over the variables in a maximal

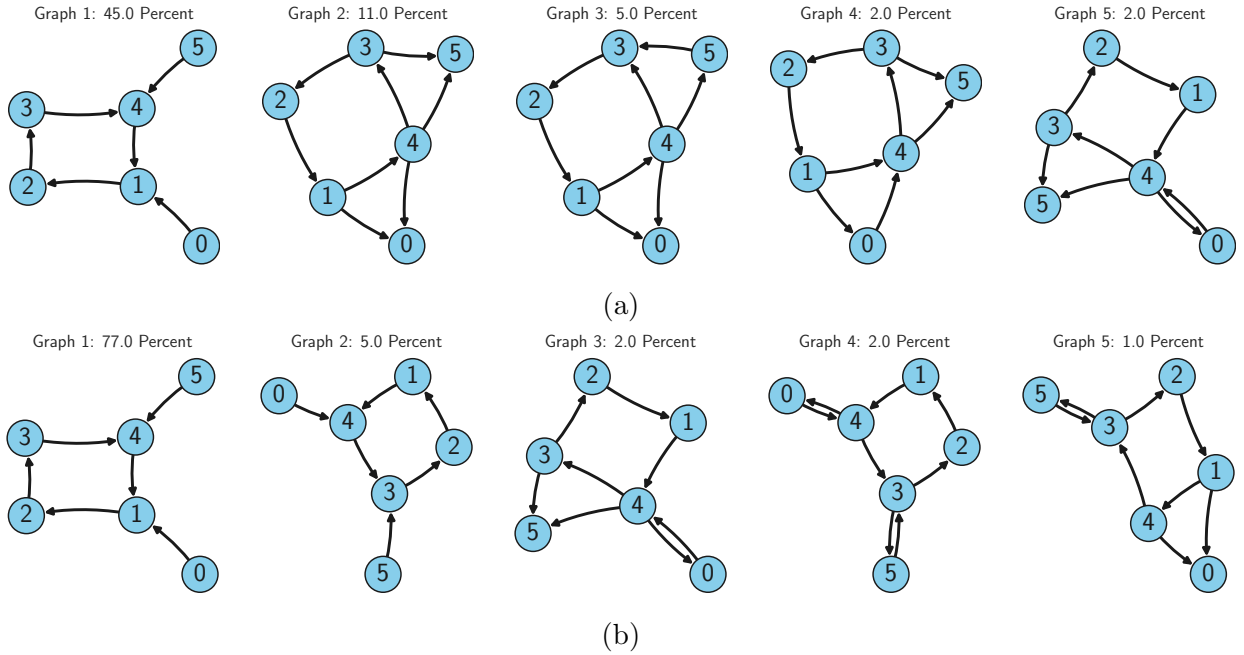

Figure C.3: Example 2. Comparison of 5 most commonly learned structures.

strongly connected subgraph (MSCS), conditioned on their parents outside of the MSCS. This can be shown as follows, where an MSCS is denoted by S.

$$P(V) = \prod_{S_i \subseteq V} P(S_i | Pa(S_i)).$$

After applying an operation, the likelihoods of all involved MSCSs are updated. Note that an operation can merge several MSCSs or break one into several smaller MSCSs. We perform the updates as follows:

- If the change adds an edge from MSCS S_1 to S_2, These two MSCSs and any MSCS on any path from S_2 to S_1 will fused into a new large MSCS.
- If the change is performed inside an MSCS, the score of the rest of MSCSs do not change.
- If the change removes or reverses an edge inside an MSCS, we find the MSCSs in that subset again, as it may be divided into smaller MSCSs.

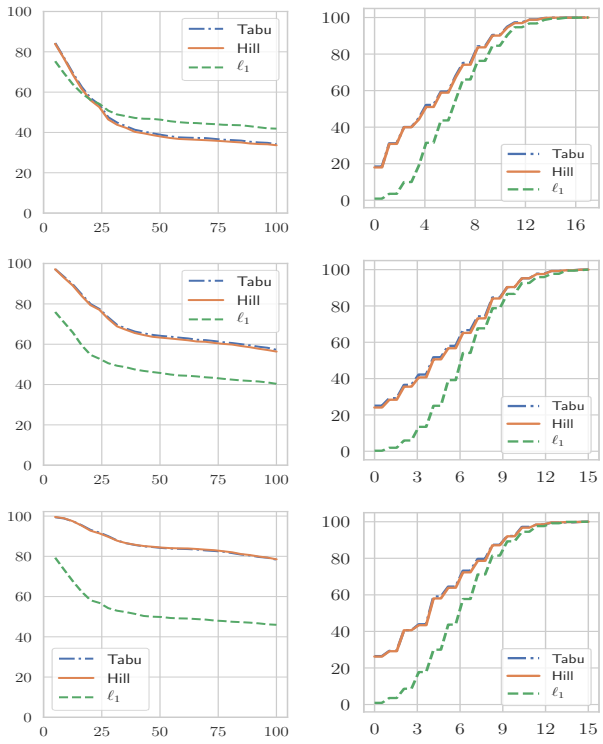

Figure C.4: Results for $n = 10^3, 10^4, 10^5$, top to bottom. **Left column:** multi-domain evaluation. The percentage of outputs with success rate larger than a certain value is plotted vs. success percentages. **Right column:** SHD evaluation. The percentage of outputs with SHD less than or equal to a certain value is plotted vs. SHD.

C.18 Effect of Sample Size on the Performance

In this section, we compare the performance of the discussed structure learning algorithms in the case of $p = 5$ variables and three different sample sizes: $n = 10^3, 10^4$, and 10^5. The results of the comparison are shown in Figure C.4. As can be seen in the figure, the performance of the ℓ_0-regularized local search methods show marked improvement as sample size is increased.

For all experiments, including those in the main text, we use the following hyperparameters for the search algorithms. For the ℓ_1-regularized MLE, we use a regularization coefficient of 0.1, and threshold the learned B matrix at 0.05. See [14] for details on greedy hill search and tabu search and its parameters. For tabu search, we use a tabu length of 5 for the $p = 5$ case and 10 for the $p = 20$ and $p = 50$ cases. In all cases, we used a tabu search patience of 5.

APPENDIX D

APPENDIX OF CHAPTER 6

D.1 Proof of Lemma 10

Consider any two observed variables V_i and V_j. We know that $[\mathbf{B}']_{i,i}$ and $[\mathbf{B}']_{j,j}$ are non-zero. Furthermore, $\mathbf{B}'$ is a sub-matrix of $\mathbf{B}$. Hence, based on Lemma 9 (ii), if there is no causal path between V_i and V_j, we have: $[\mathbf{B}']_{i,j} = 0$ and $[\mathbf{B}']_{j,i} = 0$. Thus, $[\mathbf{B}']_{:,i}$ and $[\mathbf{B}']_{:,j}$ are not linearly dependent. Furthermore, if one of the variable is the ancestor of the another one, let say $V_i \in anc(V_j)$, according to Lemma 9 (i), $[\mathbf{B}']_{j,i} \neq 0$ while $[\mathbf{B}']_{i,j} = 0$. Thus, $[\mathbf{B}']_{:,i}$ and $[\mathbf{B}']_{:,j}$ are also not linearly dependent in this case and the proof is complete.

D.2 Proof of Lemma 11

First, we show that if $V_i \rightsquigarrow V_j$, then $n_{0*} > 0$ and $n_{*0} = 0$. We know that matrix $[\tilde{\mathbf{B}}''_{i,:}; \tilde{\mathbf{B}}''_{j,:}]$ can be converted to $[\mathbf{B}''_{i,:}; \mathbf{B}''_{j,:}]$ by some permutation and scaling of its columns. Moreover, $\mathbf{B}''$ contains some of the columns of $\mathbf{B}'$ including all the columns corresponding to the observed variables. Thus, from Lemma 9, we know that if $[\mathbf{B}'']_{i,k} \neq 0$ for any $k \neq j$, then $[\mathbf{B}'']_{j,k} \neq 0$. Moreover, we have: $[\mathbf{B}'']_{j,j} \neq 0$ and $[\mathbf{B}'']_{i,j} = 0$. Hence, we can conclude that: $n_{0*} > 0$ and $n_{*0} = 0$.

If $n_{0*} > 0$ and $n_{*0} = 0$, then $V_i \rightsquigarrow V_j$. By contradiction, suppose that there is no causal path between V_i and V_i or $V_j \rightsquigarrow V_i$. The second case ($V_j \rightsquigarrow V_i$) does not happen due to what we just proved. Furthermore, from Lemma 9, we know that $[\mathbf{B}'']_{i,i} \neq 0$, $[\mathbf{B}'']_{i,j} = 0$. Therefore, $n_{*0} > 0$ which is in contradiction with our assumption. Hence, we can conclude that $n_{0*} > 0$ and $n_{*0} = 0$ if and only if $V_i \rightsquigarrow V_j$.

D.3 Proof of Theorem 15

"if" part:

We say a directed path is latent if all the variables on the path except the endpoint are latent. The "if" parts of conditions in Theorem 15 can be rewritten as follows:

(a) Latent variable V_{p_o+j}, $1 \leq j \leq p_l$, is absorbable in $\emptyset$ if it has no observable descendant.

(b1) Latent variable V_{p_o+j}, $1 \leq j \leq p_l$, is absorbable in observed variable V_i, $1 \leq i \leq p_o$, if V_i is the only observed variable influenced by V_{p_o+j} through some latent paths.

(b2) Latent variable V_{p_o+j}, $1 \leq j \leq p_l$, is absorbable in latent variable V_{p_o+k}, $1 \leq k \leq p_l$, if all latent paths from V_{p_o+j} to observed variables go through V_{p_o+k}.

It is easy to show that conditions (b1) and (b2) are equivalent to "if" part of condition (b) in Theorem 15. From (6.6), we know that $\mathbf{V_o} = (\mathbf{I}-\mathbf{D})^{-1}[\mathbf{I}, \mathbf{A_{ol}}(\mathbf{I}-\mathbf{A_{ll}})^{-1}]\mathbf{N}$ where entry (i,j) of matrix $(\mathbf{I}-\mathbf{D})^{-1}\mathbf{A_{ol}}(\mathbf{I}-\mathbf{A_{ll}})^{-1}$ is the total causal effect of latent variable V_{p_o+j} to the observed variable V_i. This entry would be zero if no directed path exists from latent variable V_{p_o+j} to observed variable V_i. Now, we prove the correctness of above conditions:

(a) If a latent variable V_{p_o+j} has no observable descendant, then the j-th column of $\mathbf{A_{ol}}(\mathbf{I}-\mathbf{A_{ll}})^{-1}$ is all zeros. Hence, there would be no changes in $[\mathbf{I}, \mathbf{A_{ol}}(\mathbf{I}-\mathbf{A_{ll}})^{-1}]\mathbf{N}$ by setting N_{p_o+j} to zero. Therefore, there would be no change in P_{V_o}.

(b1) Since latent variable V_{p_o+j} only influences one observed variable through latent paths, $[\mathbf{A_{ol}}(\mathbf{I}-\mathbf{A_{ll}})^{-1}]_{:,j}$ has only one non-zero entry and therefore linearly dependent on one of columns of identity matrix, let say i-th column. Moreover, the total causal effect from V_{p_o+j} to V_i, i.e., $[\mathbf{B}]_{i,p_o+j}$ is equal to $[\mathbf{A_{ol}}(\mathbf{I}-\mathbf{A_{ll}})^{-1}]_{i,j}$ since there is no causal path from V_{p_o+j} to V_i that goes through an observed variable other than V_i. Thus, we replace N_i by $N_i + [\mathbf{A_{ol}}(\mathbf{I}-\mathbf{A_{ll}})^{-1}]_{i,j}N_{p_o+j}$ and set N_{p_o+j} to zero and there would be no change in $[\mathbf{I}, \mathbf{A_{ol}}(\mathbf{I}-\mathbf{A_{ll}})^{-1}]\mathbf{N}$.

(b2) Consider any observed variable V_i, $1 \leq i \leq p_o$. If all latent paths of V_{p_o+j} go though V_{p_o+k}, then $[\mathbf{A_{ol}}(\mathbf{I}-\mathbf{A_{ll}})^{-1}]_{i,j} = [\mathbf{A_{ol}}(\mathbf{I}-\mathbf{A_{ll}})^{-1}]_{i,k}[\mathbf{B}]_{p_o+k,p_o+j}$ since all the paths from V_{p_o+j} to V_{p_o+k} are latent. Thus, we can change N_{p_o+k} to $N_{p_o+k} + [\mathbf{B}]_{p_o+k,p_o+j}N_{p_o+j}$ and set N_{p_o+j} to zero and there would be no change in $[\mathbf{I}, \mathbf{A_{ol}}(\mathbf{I}-\mathbf{A_{ll}})^{-1}]\mathbf{N}$.

"only if" part:

Now, we prove that the conditions (a), (b1), and (b2) are the only absorbable case. It can be easily shown that an observed variable cannot be absorbed into any other observed or latent variables. Thus, it is just needed to consider the following cases:

- Absorbing a latent variable in an observed variable: Suppose that a latent variable

V_j can be absorbed in an observed variable V_i. Furthermore, assume that V_j also influences other observed variable V_k through latent path(s). That is, there exist some paths that start from V_j and end in V_k without traversing, V_i. Let $\gamma \neq 0$ be the causal strength of such paths. Then, $[\mathbf{B}]_{k,j} = [\mathbf{B}]_{k,i} \times [\mathbf{B}]_{i,j} + \gamma$. To absorb V_j in V_i, γ should be zero which would contradict the faithfulness assumption.

- Absorbing a latent variable in another latent variable: Suppose that a latent variable V_j can be absorbed in another latent variable V_i but for some observed variable V_k, all latent paths from V_j do not go through V_i. Let γ be the causal strength of such paths. Then, $[\mathbf{B}]_{k,j} = [\mathbf{B}]_{k,i} \times [\mathbf{B}]_{i,j} + \gamma$. To absorb V_j in V_i, γ should be zero which contradicts the faithfulness assumption.

D.4 Proof of Lemma 12

Suppose that a latent variable V_i has at least two non-absorbable children such as V_j and V_k. We need to consider three cases:

- If both of V_j and V_k are observed variables, then V_i is not absorbable according to Theorem 15.

- Suppose that V_j and V_k are latent variables. Each of them must reach at least two observed variables through latent paths (due to condition (b) in Theorem 15). Thus, V_i also reaches those observed variables through latent paths. Furthermore, all latent paths starting from V_i do not go through only one latent variable. Hence, none of the conditions in Theorem 15 are satisfied and V_i is not absorbable.

- One of V_j or V_k, let say variable V_j, is observed. V_k must reach an observed variable other than V_j through some latent paths. Otherwise, it is absorbable. Therefore, V_i is not absorbable since it does not satisfy any conditions in Theorem 15.

D.5 Proof of Theorem 16

If G is not minimal, then it can be easily seen that $\mathbf{B}'$ is also reducible. Now, suppose that G is minimal. We want to show that $\mathbf{B}'$ is also not reducible almost surely. By contradiction,

suppose that $\mathbf{B}'$ is reducible. Then two columns of $[\mathbf{I}, \mathbf{A_{ol}}(\mathbf{I} - \mathbf{A_{ll}})^{-1}]$ must be linearly dependent. Now, two cases should be considered:

- One column of $\mathbf{A_{ol}}(\mathbf{I} - \mathbf{A_{ll}})^{-1}$, let say i-th column, and one column of $\mathbf{I}$ are linearly dependent. Hence, all the latent paths starting from latent variable V_{p_o+i} influences only one observed variable (Condition (b) in Theorem 15). Thus, G is not minimal which is a contradiction.

- Two columns of $\mathbf{A_{ol}}(\mathbf{I} - \mathbf{A_{ll}})^{-1}$, let say i, j are linearly dependent. If the corresponding columns have only one non-zero entry, then both of them can be absorbed in an observed variable (Condition (b) in Theorem 15). Thus, G is not minimal. Now, suppose that these columns have more than one nonzero entry each, let say entries k and l. Without loss of generality, suppose that V_{p_o+i} is the ancestor of V_{p_o+j} (the same argument still holds true if neither is an ancestor of the other). Let h_i be the maximum length of latent paths starting from latent variable V_{p_o+i}. By induction on h_i, we will show that i, j-th columns of $\mathbf{A_{ol}}(\mathbf{I} - \mathbf{A_{ll}})^{-1}$ are linearly dependent with measure zero. The case of $h_i = 1$ is trivial. Suppose that for $h_i = r$, the statement holds true. We will prove it for $h_i = r + 1$. Let latent variable V_{p_o+u} be a child of V_{p_o+i} and assume some paths from V_{p_o+u} do not go through V_{p_o+j}. Let $[\mathbf{C}]_{i,j}$ be the total causal strength of only latent paths from V_j to V_i. We know that:

$$[\mathbf{C}]_{k,p_o+j}/[\mathbf{C}]_{l,p_o+j} = [\mathbf{C}]_{k,p_o+i}/[\mathbf{C}]_{l,p_o+i}. \tag{D.1}$$

Furthermore,

$$[\mathbf{C}]_{k,p_o+i} = [\mathbf{C}]_{k,p_o+u}[\mathbf{C}]_{p_o+u,p_o+i} + c', [\mathbf{C}]_{l,p_o+i} = [\mathbf{C}]_{l,p_o+u}[\mathbf{C}]_{p_o+u,p_o+i} + c'', \tag{D.2}$$

for some values c', c''. Moreover, $[\mathbf{C}]_{p_o+u,p_o+i} = [\mathbf{A}]_{p_o+u,p_o+i} + c'''$ for some c'''. Plugging (D.2) into (D.1), we have:

$$\begin{aligned}([\mathbf{C}]_{k,p_o+u}[\mathbf{C}]_{l,p_o+j} - [\mathbf{C}]_{k,p_o+j}[\mathbf{C}]_{l,p_o+u})[\mathbf{A}]_{p_o+u,p_o+i} = \\ [\mathbf{C}]_{l,p_o+j}c' - [\mathbf{C}]_{k,p_o+j}c'' - ([\mathbf{C}]_{k,p_o+u}[\mathbf{C}]_{l,p_o+j} - [\mathbf{C}]_{k,p_o+j}[\mathbf{C}]_{l,p_o+u})c'''.\end{aligned}$$

The above equation holds with measure zero if $[\mathbf{C}]_{k,p_o+u}[\mathbf{C}]_{l,p_o+j} - [\mathbf{C}]_{k,p_o+j}[\mathbf{C}]_{l,p_o+u} \neq 0$ which is true with measure one from the induction hypothesis.

D.6 Proof of Corollary 4

Based on Theorem 16, we know that matrix $\mathbf{B}'$ is not reducible almost surely if the corresponding causal graph G is minimal. Furthermore, according to Proposition 15, the number of variables in the systems is identifiable if matrix $\mathbf{B}'$ is not reducible. This completes the proof.

D.7 An Example of Non-Identifiability of Total Causal Effects

Let $P = (V_{i_0}, V_{i_1}, \cdots, V_{i_{r-1}}, V_{i_r})$ be a causal path of length r from variable V_{i_0} to variable V_{i_r}. We define the weight of path P, denoted by ω_p , as the product of direct causal strengths of edges on the path:

$$\omega_P = \prod_{s=0}^{r-1} [\mathbf{A}]_{i_{s+1}, i_s}. \tag{D.3}$$

Suppose that Π_{V_i,V_j} be the set of all causal paths from variable V_i to variable V_j. It can be shown that the total causal effect from V_i to V_j can be computed by the following equation:

$$[\mathbf{B}]_{j,i} = \sum_{P \in \Pi_{V_i,V_j}} \omega_P. \tag{D.4}$$

Now, consider a causal graph in Figure 6.5 where V_i and V_j are observed variables and V_k is latent variable. There exist causal paths from V_k to V_i and V_j, and from V_i to V_j with the following properties:

- Let Π'_{V_k,V_j} be the causal paths from variable V_k to variable V_j where V_i is not on any of these paths. We assume that $\Pi'_{V_k,V_j} \neq \emptyset$.
- All intermediate variables in Π_{V_k,V_i}, Π'_{V_k,V_j} and Π_{V_i,V_j} are latent.

We can write V_i and V_j based on the exogenous noises of their ancestors as follows:

$$\begin{aligned} V_i &= \alpha N_k + \sum_{V_r \in anc(V_i) \setminus V_k} [\mathbf{B}]_{i,r} N_r, \\ V_j &= \beta N_i + \gamma N_k + \sum_{V_r \in anc(V_j) \setminus \{V_k, V_i\}} [\mathbf{B}]_{i,r} N_r, \end{aligned} \tag{D.5}$$

where $\alpha = \sum_{P \in \Pi_{V_k,V_i}} \omega_P$, $\beta = \sum_{P \in \Pi_{V_i,V_j}} \omega_P$, and $\gamma = \sum_{P \in \Pi'_{V_k,V_j}} \omega_P$.

Now, we construct a causal graph depicted in Figure 6.5 where the exogenous noises of variables V_i and V_k are changed to αN_k and N_i, respectively. Furthermore, we pick three paths $P_1 \in \Pi_{V_k,V_i}$, $P_2 \in \Pi'_{V_k,V_j}$, $P_3 \in \Pi_{V_i,V_j}$ where:

$$\begin{aligned} P_1 &= (V_k, V_{u_1}, \cdots, V_i), \\ P_2 &= (V_k, V_{u_2}, \cdots, V_j), \\ P_3 &= (V_i, V_{u_3}, \cdots, V_j). \end{aligned}$$

By our first property on the paths, we can find two paths P_1 and P_2 such that $V_{u_1} \neq V_{u_2}$. We also change matrix $\mathbf{A}$ to matrix $\mathbf{A}'$ where all the entries of $\mathbf{A}'$ are the same as $\mathbf{A}$ except three entries $[\mathbf{A}']_{u_1,k}$, $[\mathbf{A}']_{u_2,k}$, and $[\mathbf{A}']_{u_3,i}$. We will adjust these three entries such that the total causal effects from V_k to V_i, from V_k to V_j, and from V_i to V_j become 1, $-\gamma/\alpha$, and $\beta + \gamma/\alpha$, respectively. Moreover, these adjustments should not change the dependencies of observed variables V_i and V_j to the exogenous noises of their ancestors given in Equation (D.5). It can be shown that we can change the three mentioned causal effects to our desired values by the following adjustments:

$$\begin{aligned} [\mathbf{A}']_{u_1,k} &= \frac{1 - \sum_{P \in \Pi_{V_k,V_i} \setminus \{P_1\}} \omega_P}{\omega_{P_1} / [\mathbf{A}]_{u_2,k}}, \\ [\mathbf{A}']_{u_2,k} &= \frac{-\gamma/\alpha - \sum_{P \in \Pi'_{V_k,V_j} \setminus \{P_2\}} \omega_P}{\omega_{P_2} / [\mathbf{A}]_{u_2,k}}, \\ [\mathbf{A}']_{u_3,i} &= \frac{\beta + \gamma/\alpha - \sum_{P \in \Pi_{V_i,V_j} \setminus \{P_3\}} \omega_P}{\omega_{P_3} / [\mathbf{A}]_{u_3,i}}. \end{aligned}$$

Now, consider any latent variable V_u which is on one of the paths in Π_{V_k,V_i}, Π'_{V_k,V_j}, or Π_{V_i,V_j}. Changes in those mentioned three edges cannot affect the total causal effect from V_u to V_i or V_j since the edges (V_k, V_{u_1}), (V_k, V_{u_2}), and (V_i, V_{u_3}) are not a part of any paths from V_u to V_i or V_j. Thus, equations in (D.5) do not change while the total causal effect from V_i to V_j becomes $\beta + \gamma/\alpha$ in the second causal graph. It is noteworthy that changes in the equations of latent variables are not important since we are not observing these variables.

D.8 Proof of Lemma 13

Let $\mathbf{P}$ be the permutation matrix corresponding to the causal order k_o. We want to show that $\mathbf{PDP}^T$ is strictly lower triangular. It suffices to prove $\mathbf{PA_{ol}A_{ll}}^k\mathbf{A_{lo}P}^T$ is strictly lower triangular for any $0 \leq k \leq d_l - 1$. Suppose that there exists a nonzero entry, (i, j), in $\mathbf{PA_{ol}A_{ll}}^k\mathbf{A_{lo}P}^T$ where $j \geq i$. Then, there should be a directed path from observed variable $V_{k_o^{-1}(j)}$ to $V_{k_o^{-1}(i)}$ of length $k+2$ through latent variables in the causal graph where $k_o^{-1}(i)$ is the index of an observed variable whose order is i in the causal order k_o. This means variable $V_{k_o^{-1}(j)}$ should come before variable $V_{k_o^{-1}(i)}$ in any causal order. But this violates the causal order k_o.

D.9 Proof of Theorem 17

According to Proposition 15, under non-Gaussianity of exogenous noises, the columns of $\mathbf{B}''$ can be determined up to some scalings and permutations by solving an overcomplete ICA problem. Furthermore, for the column corresponding to the noise N_i, $1 \leq i \leq p_o$, we have r_i possible candidates with the same set of indices of non-zero entries where all of them are pairwise linearly independent. Let $\mathbf{B'_o}$ be a $p_o \times p_o$ matrix by selecting one of the candidates for each column corresponding to noise N_i, $1 \leq i \leq p_o$. Thus, we have $\Pi_{i=1}^{p_o} r_i$ possible matrices.[1] Now, for each $\mathbf{B'_o}$, we just need to show that there exists an assignment for $\mathbf{A_{oo}}$, $\mathbf{A_{lo}}$, $\mathbf{A_{ol}}$, and $\mathbf{A_{ll}}$ such that they satisfy (6.6) and $\mathbf{A_{oo}}$ and $\mathbf{A_{ll}}$ can be converted to strictly lower triangular matrices with some simultaneous permutations of columns and rows.

Let $\mathbf{A_{lo}} = \mathbf{0}_{p_l \times p_o}$ and $\mathbf{A_{ll}} = \mathbf{0}_{p_l \times p_l}$. Assume that $\mathbf{B'_l}$ consists of the remaining columns which are not in $\mathbf{B'_o}$. We also add columns corresponding to latent absorbed variables to $\mathbf{B'_l}$. Now, we set $\mathbf{A_{oo}}$ and $\mathbf{A_{ol}}$ to $\mathbf{I} - \mathbf{B'_o}^{-1}$ and $\mathbf{B'_o}^{-1}\mathbf{B'_l}$, respectively. By these assignments, the proposed matrix $\mathbf{A} = [\mathbf{A_{oo}}, \mathbf{A_{ol}}; \mathbf{A_{lo}}, \mathbf{A_{ll}}]$ satisfies in (6.6). Thus, we just need to show that $\mathbf{I} - \mathbf{B'_o}^{-1}$ can be converted to a strictly lower triangular matrix by some permutations. To do so, first note that from Lemma 13, we know that matrix $\mathbf{D}$ can be converted to a strictly lower triangular matrix by a permutation matrix $\mathbf{P}$. Furthermore, based on this property of

[1] Note that diagonal entries of $\mathbf{B'_o}$ should be equal to one. Otherwise we can normalize each column to its on-diagonal entry.

matrix $\mathbf{D}$, we have: $\mathbf{D}^{p_o} = \mathbf{0}$. Thus, we can write:

$$\mathbf{P}(\mathbf{I}-\mathbf{D})^{-1}\mathbf{P}^T = \sum_{k=0}^{p_o-1} \mathbf{P}\mathbf{D}^k\mathbf{P}^T = \sum_{k=0}^{p_o-1} (\mathbf{P}\mathbf{D}\mathbf{P}^T)^k.$$

Since matrix $(\mathbf{P}\mathbf{D}\mathbf{P}^T)^k$ is a lower triangular matrix for any $k \geq 0$, $(\mathbf{I}-\mathbf{D})^{-1}$ can be converted to a lower triangular matrix by permutation matrix $\mathbf{P}$. Furthermore, the set of nonzero entries of $\mathbf{B'_o}$ is the same as the one of $(\mathbf{I}-\mathbf{D})^{-1}$. Thus, $\mathbf{P}\mathbf{B'_o}\mathbf{P}^T$ is also a lower triangular matrix where all diagonal elements of it are equal to one. Hence, we can write $\mathbf{B'_o}$ in the form of $\mathbf{B'_o} = \mathbf{I} + \mathbf{B''_o}$ where $\mathbf{P}\mathbf{B''_o}\mathbf{P}^T$ is a strictly lower triangular matrix. Therefore, we have:

$$\mathbf{P}(\mathbf{I}-{\mathbf{B'_o}}^{-1})\mathbf{P}^T = \mathbf{P}(\mathbf{I}-\sum_{k=0}^{p_o-1}(-1)^k{\mathbf{B''_o}}^k)\mathbf{P}^T = \mathbf{P}(\sum_{k=1}^{p_o-1}(-1)^{k+1}{\mathbf{B''_o}}^k)\mathbf{P}^T, \tag{D.6}$$

where the last term shows that $\mathbf{I}-{\mathbf{B'_o}}^{-1}$ can be converted to a strictly lower triangular matrix and the proof is complete.

D.10 Proof of Theorem 18

Let matrix $[\tilde{\mathbf{B}}'']_{p_o \times p_r}$ be the output of over-complete ICA problem whose columns are the columns in matrix $\mathbf{B}''$. We define I_i as the set of indices of nonzero entries of column $\tilde{\mathbf{B}}''_{:,i}$, i.e. $I_i = \{k | [\tilde{\mathbf{B}}''_{:,i}]_k \neq 0\}$. We know that $I_i = des_o(V_j)$ if $\tilde{\mathbf{B}}''_{:,i}$ corresponds to the observed variable V_j. Moreover, under Assumption 10, any observed variable V_i and any variable V_j (observed or latent) have different sets $des_o(V_i)$ and $des_o(V_j)$. Thus, each set I_i is just equal to one of $des_o(V_i)$'s, let say $des_o(V_j)$. The column $\tilde{\mathbf{B}}''_{:,i}$ normalized by $[\tilde{\mathbf{B}}''_{:,i}]_j$ shows the total causal effects from variable j to other observed variables.

REFERENCES

[1] P. Spirtes, C. N. Glymour, and R. Scheines, *Causation, Prediction, and Search*. MIT Press, 2000.

[2] T. Richardson, "A discovery algorithm for directed cyclic graphs," in *Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 1996, pp. 454–461.

[3] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," *Machine Learning*, vol. 20, no. 3, pp. 197–243, 1995.

[4] D. M. Chickering, "Optimal structure identification with greedy search," *Journal of Machine Learning Research*, vol. 3, no. Nov, pp. 507–554, 2002.

[5] I. Tsamardinos, L. E. Brown, and C. F. Aliferis, "The max-min hill-climbing Bayesian network structure learning algorithm," *Machine Learning*, vol. 65, no. 1, pp. 31–78, 2006.

[6] S. Shimizu, P. O. Hoyer, A. Hyvärinen, and A. Kerminen, "A linear non-Gaussian acyclic model for causal discovery," *Journal of Machine Learning Research*, vol. 7, no. Oct, pp. 2003–2030, 2006.

[7] G. Lacerda, P. L. Spirtes, J. Ramsey, and P. O. Hoyer, "Discovering cyclic causal models by independent components analysis," in *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, 2008, pp. 366–374.

[8] P. O. Hoyer, D. Janzing, J. M. Mooij, J. Peters, and B. Schölkopf, "Nonlinear causal discovery with additive noise models," in *Advances in Neural Information Processing Systems*, 2009, pp. 689–696.

[9] K. Zhang and A. Hyvärinen, "On the identifiability of the post-nonlinear causal model," in *Proceedings of the Twenty-fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2009, pp. 647–655.

[10] J. M. Mooij, D. Janzing, T. Heskes, and B. Schölkopf, "On causal discovery with cyclic additive noise models," in *Advances in Neural Information Processing Systems*, 2011, pp. 639–647.

[11] J. Pearl, *Causality*. Cambridge University Press, 2009.

[12] S. A. Andersson, D. Madigan, and M. D. Perlman, "A characterization of Markov equivalence classes for acyclic digraphs," *The Annals of Statistics*, vol. 25, no. 2, pp. 505–541, 1997.

[13] S. L. Lauritzen, *Graphical Models*. Clarendon Press, 1996, vol. 17.

[14] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.

[15] J. Pearl and T. S. Verma, "A theory of inferred causation," in *Studies in Logic and the Foundations of Mathematics*. Elsevier, 1995, vol. 134, pp. 789–811.

[16] T. Verma and J. Pearl, "Equivalence and synthesis of causal models," in *Proceedings of Sixth Conference on Uncertainty in Artificial Intelligence*, 1991, pp. 220–227.

[17] C. Meek, "Causal inference and causal explanation with background knowledge," in *UAI 1995*, 1995, pp. 403–410.

[18] Y.-B. He and Z. Geng, "Active learning of causal networks with intervention experiments and optimal designs," *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2523–2547, 2008.

[19] M. Bernstein and P. Tetali, "On sampling graphical Markov models," *arXiv preprint arXiv:1705.09717*, 2017.

[20] F. Eberhardt, "Causation and intervention," Doctoral Dissertation, Carnegie Mellon University, 2007.

[21] F. Eberhardt, "Almost optimal intervention sets for causal discovery," *arXiv preprint arXiv:1206.3250*, 2012.

[22] K. Shanmugam, M. Kocaoglu, A. G. Dimakis, and S. Vishwanath, "Learning causal graphs with small interventions," in *Advances in Neural Information Processing Systems*, 2015, pp. 3195–3203.

[23] A. Ghassami, S. Salehkaleybar, N. Kiyavash, and E. Bareinboim, "Budgeted experiment design for causal structure learning," in *International Conference on Machine Learning*, 2018, pp. 1719–1728.

[24] A. Ghassami, S. Salehkaleybar, N. Kiyavash, and K. Zhang, "Counting and sampling from Markov equivalent DAGs using clique trees," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3664–3671.

[25] A. Ghassami, S. Salehkaleybar, and N. Kiyavash, "Interventional experiment design for causal structure learning," *arXiv preprint arXiv:1910.05651*, 2019.

[26] K. B. Korb, L. R. Hope, A. E. Nicholson, and K. Axnick, "Varieties of causal intervention," in *Pacific Rim International Conference on Artificial Intelligence*. Springer, 2004, pp. 322–331.

[27] F. Eberhardt, C. Glymour, and R. Scheines, "On the number of experiments sufficient and in the worst case necessary to identify all causal relations among n variables," in *Proceedings of the 21st Conference on Uncertainty and Artificial Intelligence (UAI-05)*, 2005, pp. 178–184.

[28] A. Hauser and P. Bühlmann, "Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs," *Journal of Machine Learning Research*, vol. 13, no. Aug, pp. 2409–2464, 2012.

[29] K. D. Yang, A. Katcoff, and C. Uhler, "Characterizing and learning equivalence classes of causal DAGs under interventions," *arXiv preprint arXiv:1802.06310*, 2018.

[30] A. Hyttinen, F. Eberhardt, and P. O. Hoyer, "Experiment selection for causal discovery." *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 3041–3071, 2013.

[31] A. Hauser and P. Bühlmann, "Two optimal strategies for active learning of causal models from interventional data," *International Journal of Approximate Reasoning*, vol. 55, no. 4, pp. 926–939, 2014.

[32] M. Kocaoglu, A. Dimakis, and S. Vishwanath, "Cost-optimal learning of causal graphs," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1875–1884.

[33] E. Lindgren, M. Kocaoglu, A. G. Dimakis, and S. Vishwanath, "Experimental design for cost-aware learning of causal graphs," in *Advances in Neural Information Processing Systems*, 2018, pp. 5279–5289.

[34] F. Eberhardt, C. Glymour, and R. Scheines, "N-1 experiments suffice to determine the causal relations among n variables," in *Innovations in Machine Learning*. Springer, 2006, pp. 97–112.

[35] M. Kocaoglu, K. Shanmugam, and E. Bareinboim, "Experimental design for learning causal graphs with latent variables," in *Advances in Neural Information Processing Systems*, 2017, pp. 7021–7031.

[36] S. Tong and D. Koller, "Active learning for structure in Bayesian networks," in *International Joint Conference on Artificial Intelligence*, vol. 17, no. 1. Citeseer, 2001, pp. 863–869.

[37] A. R. Masegosa and S. Moral, "An interactive approach for Bayesian network learning using domain/expert knowledge," *International Journal of Approximate Reasoning*, vol. 54, no. 8, pp. 1168–1181, 2013.

[38] R. Agrawal, C. Squires, K. Yang, K. Shanmugam, and C. Uhler, "Abcd-strategy: Budgeted experimental design for targeted causal structure discovery," *arXiv preprint arXiv:1902.10347*, 2019.

[39] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2003, pp. 137–146.

[40] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2007, pp. 420–429.

[41] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2009, pp. 199–208.

[42] F. Eberhardt and R. Scheines, "Interventions and causal inference," *Philosophy of Science*, vol. 74, no. 5, pp. 981–995, 2007.

[43] Y. He, J. Jia, and B. Yu, "Counting and exploring sizes of Markov equivalence classes of directed acyclic graphs," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 2589–2609, 2015.

[44] K. Dudziński and S. Walukiewicz, "Exact methods for the knapsack problem and its generalizations," *European Journal of Operational Research*, vol. 28, no. 1, pp. 3–21, 1987.

[45] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions-I," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[46] S. B. Gillispie and M. D. Perlman, "The size distribution for Markov equivalence classes of acyclic digraph models," *Artificial Intelligence*, vol. 141, no. 1-2, pp. 137–155, 2002.

[47] M. Minoux, "Accelerated greedy algorithms for maximizing submodular set functions," *Optimization Techniques*, pp. 234–243, 1978.

[48] B. Mirzasoleiman, A. Badanidiyuru, A. Karbasi, J. Vondrák, and A. Krause, "Lazier than lazy greedy." in *AAAI*, 2015, pp. 1812–1818.

[49] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[50] A.-L. Barabási, *Network Science*. Cambridge University Press, 2016.

[51] D. Marbach, T. Schaffter, C. Mattiussi, and D. Floreano, "Generating realistic in silico gene networks for performance assessment of reverse engineering methods," *Journal of Computational Biology*, vol. 16, no. 2, pp. 229–239, 2009.

[52] K. Hoover, "The logic of causal inference," *Economics and Philosophy*, vol. 6, pp. 207–234, 1990.

[53] J. Tian and J. Pearl, "Causal discovery from changes," in *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2001, pp. 512–521.

[54] J. Peters, P. Bühlmann, and N. Meinshausen, "Causal inference by using invariant prediction: Identification and confidence intervals," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 78, no. 5, pp. 947–1012, 2016.

[55] K. Zhang, B. Huang, J. Zhang, C. Glymour, and B. Schölkopf, "Causal discovery in the presence of distribution shift: Skeleton estimation and orientation determination," in *Proc. International Joint Conference on Artificial Intelligence (IJCAI 2017)*, 2017.

[56] B. Huang, K. Zhang, J. Zhang, R. S. Romero, C. Glymour, and B. Schölkopf, "Behind distribution shift: Mining driving forces of changes and causal arrows," in *Proceedings of IEEE 17th International Conference on Data Mining (ICDM 2017)*, 2017.

[57] A. Ghassami, S. Salehkaleybar, N. Kiyavash, and K. Zhang, "Learning causal structures using regression invariance," in *Advances in Neural Information Processing Systems*, 2017, pp. 3015–3025.

[58] A. Ghassami, N. Kiyavash, B. Huang, and K. Zhang, "Multi-domain causal structure learning in linear systems," in *Advances in Neural Information Processing Systems*, 2018, pp. 6266–6276.

[59] H. Reichenbach, *The Direction of Time*. Dover Publications, 1999.

[60] P. Daniusis, D. Janzing, J. Mooij, J. Zscheischler, B. Steudel, K. Zhang, and B. Schölkopf, "Distinguishing causes from effects using nonlinear acyclic causal models," in *Proc. 26th Conference on Uncertainty in Artificial Intelligence (UAI2010)*, 2010.

[61] J. Peters, D. Janzing, and B. Schölkopf, *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press, 2017.

[62] K. Zhang, J. Zhang, and B. Schölkopf, "Distinguishing cause from effect based on exogeneity," in *Proc. 15th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 2015)*, 2015.

[63] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *Journal of Statistical Planning and Inference*, vol. 90, no. 2, pp. 227–244, 2000.

[64] S. Shimizu, T. Inazumi, Y. Sogawa, A. Hyvärinen, Y. Kawahara, T. Washio, P. O. Hoyer, and K. Bollen, "Directlingam: A direct method for learning a linear non-Gaussian structural equation model," *Journal of Machine Learning Research*, vol. 12, no. Apr, pp. 1225–1248, 2011.

[65] J. Peters and P. Bühlmann, "Identifiability of Gaussian structural equation models with equal error variances," *Biometrika*, vol. 101, no. 1, pp. 219–228, 2013.

[66] Y. Wang, C. Squires, A. Belyaeva, and C. Uhler, "Direct estimation of differences in causal graphs," *arXiv preprint arXiv:1802.05631*, 2018.

[67] C. Jutten and J. Herault, "Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture," *Signal Processing*, vol. 24, no. 1, pp. 1–10, 1991.

[68] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[69] S. Shimizu, "LiNGAM: Non-Gaussian methods for estimating causal structures," *Behaviormetrika*, vol. 41, no. 1, pp. 65–98, 2014.

[70] A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. J. Smola, "A kernel statistical test of independence," in *NIPS 20*. Cambridge, MA: MIT Press, 2008, pp. 585–592.

[71] R. Poldrack, T. Laumann, et al., "Myconnectome dataset," 2015, https://openfmri.org/dataset/ds000031/.

[72] C. M. Bird and N. Burgess, "The hippocampus and memory: Insights from spatial processing," *Nature Reviews Neuroscience*, vol. 9, no. 3, p. nrn2335, 2008.

[73] P. Spirtes, "Directed cyclic graphical representations of feedback models," in *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 1995, pp. 491–498.

[74] A. Hyttinen, F. Eberhardt, and P. O. Hoyer, "Learning linear cyclic causal models with latent variables," *Journal of Machine Learning Research*, vol. 13, no. Nov, pp. 3387–3439, 2012.

[75] J. Tian and J. Pearl, "On the testable implications of causal models with hidden variables," in *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 2002, pp. 519–527.

[76] I. Shpitser, R. J. Evans, T. S. Richardson, and J. M. Robins, "Introduction to nested Markov models," *Behaviormetrika*, vol. 41, no. 1, pp. 3–39, 2014.

[77] J. Pearl and R. Dechter, "Identifying independencies in causal graphs with feedback," in *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 1996, pp. 420–426.

[78] R. M. Neal, "On deducing conditional independence from d-separation in causal graphs with feedback (research note)," *Journal of Artificial Intelligence Research*, vol. 12, pp. 87–91, 2000.

[79] T. Richardson, "A polynomial-time algorithm for deciding Markov equivalence of directed cyclic graphical models," in *Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 1996, pp. 462–469.

[80] A. Ghassami, A. Yang, N. Kiyavash, and K. Zhang, "Characterizing distribution equivalence and structure learning for cyclic and acyclic directed graphs," *arXiv preprint arXiv:1910.12993*, 2020.

[81] E. V. Strobl, "A constraint-based algorithm for causal discovery with cycles, latent variables and selection bias," *International Journal of Data Science and Analytics*, vol. 8, no. 1, pp. 33–56, 2019.

[82] A. Hyttinen, P. O. Hoyer, F. Eberhardt, and M. Jarvisalo, "Discovering cyclic causal models with latent variables: A general sat-based procedure," *arXiv preprint arXiv:1309.6836*, 2013.

[83] A. Hyttinen, F. Eberhardt, and M. Järvisalo, "Constraint-based causal discovery: Conflict resolution with answer set programming," in *UAI*, 2014, pp. 340–349.

[84] P. Forré and J. M. Mooij, "Constraint-based causal discovery for non-linear structural causal models with cycles and latent confounders," *arXiv preprint arXiv:1807.03024*, 2018.

[85] G. H. Golub and C. F. Van Loan, *Matrix Computations*. JHU Press, 2012, vol. 3.

[86] C. Meek, "Strong completeness and faithfulness in Bayesian networks," *arXiv preprint arXiv:1302.4973*, 2013.

[87] M. Teyssier and D. Koller, "Ordering-based search: A simple and effective algorithm for learning Bayesian networks," *arXiv preprint arXiv:1207.1429*, 2012.

[88] L. Solus, Y. Wang, L. Matejovicova, and C. Uhler, "Consistency guarantees for permutation-based causal inference algorithms," *arXiv preprint arXiv:1702.03530*, 2017.

[89] S. Van de Geer and P. Bühlmann, "ℓ_0-penalized maximum likelihood for sparse directed acyclic graphs," *The Annals of Statistics*, vol. 41, no. 2, pp. 536–567, 2013.

[90] F. Fu and Q. Zhou, "Learning sparse causal Gaussian networks with experimental intervention: regularization and coordinate descent," *Journal of the American Statistical Association*, vol. 108, no. 501, pp. 288–300, 2013.

[91] B. Aragam and Q. Zhou, "Concave penalized estimation of sparse Gaussian Bayesian networks." *Journal of Machine Learning Research*, vol. 16, pp. 2273–2328, 2015.

[92] G. Raskutti and C. Uhler, "Learning directed acyclic graph models based on sparsest permutations," *Stat*, vol. 7, no. 1, p. e183, 2018.

[93] X. Zheng, B. Aragam, P. K. Ravikumar, and E. P. Xing, "DAGs with NO TEARS: Continuous optimization for structure learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 9472–9483.

[94] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Elsevier, 1988.

[95] S. Salehkaleybar, J. Etesami, N. Kiyavash, and K. Zhang, "Learning vector autoregressive models with latent processes," in *International Conference on Machine Learning*, 2018, pp. 4000–4007.

[96] S. Salehkaleybar, J. Etesami, and N. Kiyavash, "Identifying nonlinear 1-step causal influences in presence of latent variables," in *2017 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2017, pp. 1341–1345.

[97] D. Janzing, J. Mooij, K. Zhang, J. Lemeire, J. Zscheischler, P. Daniušis, B. Steudel, and B. Schölkopf, "Information-geometric approach to inferring causal directions," *Artificial Intelligence*, vol. 182, pp. 1–31, 2012.

[98] P. O. Hoyer, S. Shimizu, A. J. Kerminen, and M. Palviainen, "Estimation of causal effects using linear non-Gaussian causal models with hidden variables," *International Journal of Approximate Reasoning*, vol. 49, no. 2, pp. 362–378, 2008.

[99] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. John Wiley & Sons, 2004, vol. 46.

[100] D. Entner and P. O. Hoyer, "Discovering unconfounded causal relationships using linear non-Gaussian models," in *JSAI International Symposium on Artificial Intelligence*. Springer, 2010, pp. 181–195.

[101] Z. Chen and L. Chan, "Causality in linear non-Gaussian acyclic models in the presence of latent Gaussian confounders," *Neural Computation*, vol. 25, no. 6, pp. 1605–1641, 2013.

[102] T. Tashiro, S. Shimizu, A. Hyvärinen, and T. Washio, "ParceLiNGAM: A causal ordering method robust against latent confounders," *Neural Computation*, vol. 26, no. 1, pp. 57–83, 2014.

[103] S. Shimizu and K. Bollen, "Bayesian estimation of causal direction in acyclic structural equation models with individual-specific confounder variables and non-Gaussian distributions," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2629–2652, 2014.

[104] G. Elidan and N. Friedman, "Learning hidden variable networks: The information bottleneck approach," *Journal of Machine Learning Research*, vol. 6, no. Jan, pp. 81–127, 2005.

[105] R. I. Jennrich and P. M. Bentler, "Exploratory bi-factor analysis," *Psychometrika*, vol. 76, no. 4, pp. 537–549, 2011.

[106] V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky, "Latent variable graphical model selection via convex optimization," in *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2010, pp. 1610–1613.

[107] P. Spirtes, C. Meek, and T. Richardson, "Causal inference in the presence of latent variables and selection bias," in *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 1995, pp. 499–506.

[108] E. Kummerfeld and J. Ramsey, "Causal clustering for 1-factor measurement models," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 1655–1664.

[109] R. Silva and R. Scheines, "Generalized measurement models," Carnegie Mellon University, School of Computer Science, Tech. Rep. no. CMU-CALD-05-108, 2005.

[110] S. Salehkaleybar, A. Ghassami, N. Kiyavash, and K. Zhang, "Learning linear non-Gaussian causal models in the presence of latent variables," *Journal of Machine Learning Research*, vol. 21, no. 39, pp. 1–24, 2020.

[111] J. Eriksson and V. Koivunen, "Identifiability, separability, and uniqueness of linear ICA models," *IEEE Signal Processing Letters*, vol. 11, no. 7, pp. 601–604, 2004.

[112] J. Etesami, N. Kiyavash, and T. Coleman, "Learning minimal latent directed information polytrees," *Neural Computation*, vol. 18, no. 9, pp. 1723–1768, 2016.

[113] Q. V. Le, A. Karpenko, J. Ngiam, and A. Y. Ng, "ICA with reconstruction cost for efficient overcomplete feature learning," in *Advances in Neural Information Processing Systems*, 2011, pp. 1017–1025.

[114] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. CRC Press, 1994.

[115] A. Hyvärinen, K. Zhang, S. Shimizu, and P. O. Hoyer, "Estimation of a structural vector autoregression model using non-Gaussianity," *Journal of Machine Learning Research*, vol. 11, no. May, pp. 1709–1731, 2010.

[116] T. Verma and J. Pearl, "An algorithm for deciding if a set of observed independencies has a causal explanation," in *Proceedings of the Eighth International Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 1992, pp. 323–330.

[117] M. Pourahmadi, "Covariance estimation: The GLM and regularization perspectives," *Statistical Science*, pp. 369–387, 2011.