Instance-Specific Causal Bayesian Network

Structure Learning

by

Fattaneh Jabbari

BS in Computer Engineering, Shiraz University, 2008

MS in Computer Engineering, Sharif University of Technology, 2011

MS in Intelligent Systems, University of Pittsburgh, 2016

Submitted to the Graduate Faculty of

the Kenneth P. Dietrich School of Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2020

UNIVERSITY OF PITTSBURGH

KENNETH P. DIETRICH SCHOOL OF ARTS AND SCIENCES

This dissertation was presented

by

Fattaneh Jabbari

It was defended on

September 24th 2020

and approved by

Gregory F. Cooper, Intelligent Systems Program,University of Pittsburgh

Dr. Shyam Visweswaran, Intelligent Systems Program, University of Pittsburgh

Dr. Xinghua Lu, Intelligent Systems Program, University of Pittsburgh

Dr. Peter Spirtes, Department of Philosophy, Carnegie Mellon University

Instance-Specific Causal Bayesian Network Structure Learning

Fattaneh Jabbari, PhD

University of Pittsburgh, 2020

Much of science consists of discovering and modeling causal relationships in nature. Causal knowledge provides insight into the mechanisms acting currently (e.g., the side-effects caused by a new medication) and the prediction of outcomes that will follow when actions are taken (e.g., the chance that a disease will be cured if a particular medication is taken). In the past 30 years, there has been tremendous progress in developing computational methods for discovering causal knowledge from observational data. Some of the most significant progress in causal discovery research has occurred using causal Bayesian networks (CBNs). A CBN is a probabilistic graphical model that includes nodes and edges. Each node corresponds to a domain variable and each edge (or arc) is interpreted as a causal relationship between a parent node (a cause) and a child node (an effect), relative to the other nodes in the network.

In this dissertation, I focus on two problems: (1) developing efficient CBN structure learning methods that learn CBNs in the presence of latent variables (i.e., unmeasured or hidden variables). Handling latent variables is important in causal discovery since it can induce dependencies that need to be distinguished from direct causation. (2) developing instance-specific CBN structure learning algorithms to learn a CBN that is specific to an instance (e.g., patient), both with and without latent variables. Learning instance-specific CBNs is important in many areas of science, especially the biomedical domain; however, it is an under-studied research problem. In this dissertation, I develop various novel instance-specific CBN structure learning methods and evaluate them using simulated and real-world data.

Table of Contents

List of Tables

List of Figures

Dedication

To my baby brother, Maziar – who taught me to never give up, even when battling two deadly diseases. You will always be in my heart and my mind.

Preface

This dissertation would not have been possible without the incredible support and guidance of my advisor Greg Cooper. I am sincerely grateful to him for his never-ending patience, intellectual inputs, contagious enthusiasm, and invaluable editing advice. I feel privileged to have had the opportunity to work with Greg. I would also like to thank the members of my dissertation committee: Xinghua Lu, Peter Spirtes, and Shyam Visweswaran, for their insightful comments and critiques of this dissertation.

It has been a great pleasure to work with members of the Center for Causal Discovery (CCD) who helped me to gain a deeper understanding of causal discovery. I would like to especially thank Kevin Bui, Jeremey Espino, Clark Glymour, and Joe Ramsey. Many thanks to my postdoctoral supervisor, Sofia Triantafillou, for her patience and flexibility while I completed my dissertation.

I wish to thank the ISP directors: Vanathi Gopalakrishnan, Diane Litman, Jan Wiebe, and other ISP faculty members for their efforts in providing a unique environment for the ISP students to pursue multidisciplinary research in artificial intelligence. I would like to acknowledge the administrative staff at ISP and DBMI: Wendy Bergstein, Toni Porterfield, Cleat Szczepaniak, and Michele Thomas, for their help and support.

Being surrounded with a wonderful group of friends is perhaps the greatest aspect of my time in academia. I am grateful to my friends of many years: Hadi, Marzyeh, Saeedeh, Sara, Sharareh, and Vahid for always being there for me. Many special thanks to my friends in Pittsburgh who made my life here so enjoyable: Afsoon, Anahita, Arash, Azarin, Bryan, Faezeh, Homa, Hossein, Jana, Jaromir, Javad, Jeya, Jonathan, Mahbaneh, Mahdi, Mahdis, Mahnoush, Majid, Mayam, Mostafa, Omid, Sadaf, Salim, Sareh, Shadi, Yashar, and Zahra.

And finally, my most special and profound thanks to my family for inspiring me to follow my dreams. I am forever grateful to my dear parents Farahnaz and Jabbar, for all they have given me in life. Being thousands of miles away from each other and knowing that we will not be able to see each other for several years, was the most difficult part of my studies, and their life too. Nonetheless, they sent me their unconditional love and constant support from

afar by calling me twice every day at the beginning and end of the day. I must also thank my siblings, Elmira, Maziar, and Salar for their love, and for always believing in me and having my back. I would like to thank Amin, my love and my best friend, for supporting me through the ups and downs of this journey; I could not have come this far without his endless love, encouragement, and companionship.

This research was financially supported by grants from the Pennsylvania Department of Health, the National Human Genome Research Institute of the National Institutes of Health (NIH), the National Library of Medicine of the NIH, the National Science Foundation, and two fellowships from the University of Pittsburgh.

1.0 Introduction

Almost all disciplines of science devote much of their attention to the discovery and modeling of causal relationships [Spirtes et al., 2000, Pearl, 2009, Illari et al., 2011]. Causal knowledge provides insight into the mechanisms acting currently (e.g., the side-effects caused by a new medication) and the prediction of outcomes that will follow when actions are taken (e.g., the chance that a disease will be cured if a particular medication is taken). Traditionally, causal relationships were identified through interventions or experiments, which can be very expensive, unethical, and even impossible, in many cases. Therefore, numerous computational methods have been developed to discover causal relationships from a combination of existing background knowledge, experimental data, and observational data. In this dissertation, I focus on using observational data and optional background knowledge for learning causal relationships.

Given the increasing amounts of data that are being collected in all fields of science, this line of research has significant potential to accelerate scientific causal discovery. During the past few decades, some of the most significant progress in causal discovery research has occurred using causal Bayesian networks (CBNs) [Spirtes et al., 2000, Pearl, 2009]. A Bayesian network (BN) is a well-studied graphical model that represents probabilistic relationships among a set of variables that are being investigated in a domain. Under assumptions, BNs can be interpreted as causal models and learned from observational data, which has wide applicability [Spirtes et al., 2000, Pearl, 2009, Illari et al., 2011]. In this dissertation, for domain emphasis, we focus on learning CBNs, although the methods apply to BN structure learning in general.

The remainder of this chapter discusses the two main topics to which this dissertation research makes contributions. The first is a novel method that uses a Bayesian approach to score constraints in learning a CBN (or an equivalence class of CBNs). The second is a new method for learning instance-specific CBNs (or an equivalence class of them). I also investigate a combination of these two methods.

1.1 A Bayesian Method for Scoring Constraints

There are two main approaches to learning CBN structures from data: (1) constraint-based and (2) score-based (e.g., Bayesian) approaches, although other methods are also being actively developed and investigated [Peters et al., 2012, Daly et al., 2011]. A constraint-based approach iteratively performs many statistical independence tests on data to constrain the structures that are consistent with the test results; it then outputs the CBN structure that is most consistent with the test results. A constraint is an arbitrary conditional independence of the form $X \perp\!\!\!\perp Y|\boldsymbol{Z}$ which is hypothesized to hold in the data-generating model that produced dataset D, where X and Y are two variables of dataset D and $\boldsymbol{Z}$ is a subset of variables of D that excludes X and Y. If such a constraint holds, then by the axioms of probability: $P(X, Y|\boldsymbol{Z}) = P(X|\boldsymbol{Z}) \cdot P(Y|\boldsymbol{Z})$. A score-based approach, on the other hand, typically involves a scoring function and a heuristic search strategy to investigate the space of the possible CBN structures and output the most probable CBN it can find.

The constraint-based and score-based approaches each have significant, but different, strengths and weaknesses. Constraint-based methods can model and discover causal models with latent (hidden) variables relatively efficiently. These methods do not, however, provide a meaningful summary score of the chance that a causal model is correct. In contrast, a score-based method can generate and probabilistically score multiple models, and output the most probable one at the end of the search. However, the Bayesian scoring of causal models that contain latent confounders is computationally very expensive and rarely performed, particularly for large causal models. In addition, while constraint-based methods can incorporate domain beliefs known with certainty, score-based methods can use prior probabilities to represent beliefs about what is likely to be true in a domain but is not certain, which is a common situation.

The first hypothesis of this dissertation is related to developing a hybrid approach that combines the strengths of constraint-based and score-based Bayesian methods. This hybrid method derives a Bayesian probability of relevant independence constraints being true. Consider a causal model (or an equivalence class of models) that can represent latent confounding and entails a set of conditional independence constraints on the measured variables.

In this hybrid approach, the probability of a model being correct is equal to the probability that the constraints that uniquely characterize the model (or an equivalence class of models) are correct. This hybrid method exhibits the computational efficiency of a constraint-based method combined with the ability of a Bayesian approach to quantitatively compare alternative causal models according to their posterior probabilities.

I introduce three methods to compute the joint probability of constraints. The first and simplest method assumes the constraints are independent of each other. In this case, the joint probability of constraints is factored into the product of probabilities of single constraints. However, with finite data, constraints are often dependent. Indeed, the statistical relationships among the constraints can be quite complicated, and to our knowledge, they have not been modeled previously. In this dissertation, I introduce two empirical methods to model the relationships among constraints. In summary, I propose a Bayesian method that derives the joint probability that a set of dependent constraints corresponding to a given CBN (or an equivalence class of CBNs) is true. This approach is called the Bayesian Scoring of Constraints (BSC). I hypothesize the following:

The Bayesian scoring of constraints (BSC) method will perform CBN structure learning better than a method that uses frequentist statistical tests in terms of discrimination.

In order to measure *discrimination*, we use measures that evaluate the accuracy of arc adjacency, arc orientation, and overall error rates in structure learning.

1.2 Instance-Specific CBN Structure Discovery

Almost all of the existing CBN structure learning algorithms are designed to recover a CBN structure that models the causal relationships that are shared by the instances in a population; we call this a *population-wide CBN model*. While learning such population-wide CBNs accurately is useful, it is important to learn CBNs that are specific to each instance in domains in which different instances may have varying causal structures, such as in human biology. For example, a breast-cancer tumor (instance) in a patient can have a set of causal mechanisms that are different from that of another breast-cancer tumor in

a different patient. To determine the most effective treatment for a tumor in the current patient, it is important to discover the particular set of causal mechanisms that are driving that tumor to be cancerous.

In reality, a given tumor usually is a composite of cellular mechanisms that rarely all occur together, yet each individual mechanism may appear relatively commonly in other tumors. A population-wide CBN would at best capture the more common mechanisms operating in breast cancer and not all of the particular mechanisms that are active in the current patient's breast-cancer tumor. The task, then, is to construct the joint set of mechanisms of a given tumor from the individual mechanisms seen in previous tumors. To do so, we use the known features (i.e., the variable-value pairs) of the current tumor to help identify and construct the individual mechanisms that compose the set of mechanisms that are jointly driving the current tumor. In the extreme scenario, if the individual mechanisms in every tumor are not seen in other tumors, we have little hope of learning its mechanisms from a training set of prior tumors. The reality is that each of the individual mechanisms that is active in a tumor typically occurs in *some* other tumors, but not in *all* other tumors.

More generally, a given person can be viewed as a joint set of causal mechanisms, where each mechanism is typically shared with many other people, but the joint set is almost certainly unique to that person. In a given person, the causal learning task is to construct the correct set of causal mechanisms for that person from the features we know about the person and from a training set of data on many other people; we refer to such a model as *instance-specific CBN model*. Moreover, this instance-specific causal learning approach is applicable to other causal systems, beyond human biology.

The second hypothesis in this dissertation is about developing an instance-specific CBN structure learning approach. I introduce a novel, Bayesian, instance-specific structure learning method that searches the space of instance-specific CBNs to build a model that is specific to an instance T by guiding the search based on T's attributes. I hypothesize the following:

The instance-specific CBN structure learning approach will perform structure learning better than a population-wide method, in terms of discrimination.

I will also investigate the combination of instance-specific modeling and Bayesian scoring of constraints. I hypothesize the following:

The combination of instance-specific modeling and Bayesian scoring of constraints will perform CBN structure learning better than either method alone, in terms of discrimination.

1.3 Dissertation Overview

In this dissertation, I focus on developing instance-specific CBN structure learning algorithms assuming that latent variables might be absent or present. First, I review the necessary background material on CBNs and CBN structure learning in Chapter 2. Then, in Chapter 3, I introduce a novel hybrid CBN structure learning method, called BSC, that combines the strengths of the score-based and constraint-based methods, which not only allows us to model latent variables but also provides a method to approximate the score associated with learned CBN models. In Chapter 4, I present a score-based instance-specific CBN structure learning algorithm, called IGES, which assumes no latent variables (aka *causal sufficiency*). I combine the BSC and IGES algorithms to develop an algorithm that learns instance-specific causal models in the presence of latent variables; this method is introduced in Chapter 5. Finally, I conclude this dissertation by summarizing the contributions and describing possible extensions to future work in Chapter 6.

2.0 Background

In this chapter, I provide the required background information for this dissertation. First, I describe the notation used in the dissertation. I denote a random variable with an upper-case letter (e.g., X) and denote its assigned value or state with a lower-case letter (e.g., $X = x$ when variable X takes value x). I use a bold upper-case letter to represent a set of random variables (e.g., $\boldsymbol{Z}$) and a bold lower-case letter to denote the assignment of a set of values to the variables in that set (e.g., $\boldsymbol{Z} = \boldsymbol{z}$). However, to denote that an entire set of variables takes a single assignment, I use an unbold lower-case letter (e.g., $\boldsymbol{Z} = z$).

First, I present an overview of graphical terminology and definitions in Section 2.1. In Section 2.2, I provide a high-level review of previous approaches to learning population-wide CBN structures from observational data. Finally, In Section 2.3, I discuss how to evaluate the performance of CBN structure learning algorithms.

2.1 Graphical Concepts and Definitions

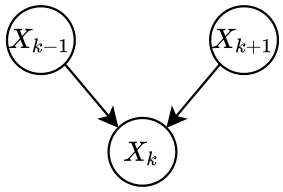
A Bayesian network (BN) is a graphical model that represents probabilistic relationships among a set of variables. Under assumptions, BNs can be interpreted as causal models and learned from observational data, which has wide applicability [Spirtes et al., 2000, Pearl, 2009, Illari et al., 2011, Peters et al., 2017]. In this dissertation, for domain emphasis, we focus on learning causal Bayesian networks (CBNs), although the methods apply to BN structure learning in general.

A BN model $M = (\mathcal{G}, \boldsymbol{\Theta})$ is composed of a graphical model structure $\mathcal{G}$ and a set of parameters $\boldsymbol{\Theta}$ for $\mathcal{G}$ [Neapolitan et al., 2004]. The graphical structure is a *directed acyclic graph (DAG)* that is given as a pair of $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V}$ is a set of nodes that correspond to the variables $\boldsymbol{V} = \{X_1, X_2, ..., X_n\}$ of the domain[1]. A DAG $\mathcal{G}$ also contains a set of directed

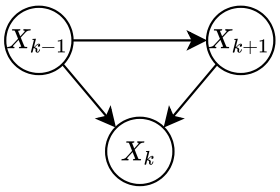
[1]We use the terms nodes and variables interchangeably because random variables are being represented by nodes in a CBN.

edges (arcs[2]) $\boldsymbol{E}$ between pairs of nodes, where these edges should not form any cycles. The presence of an edge $X_i \rightarrow X_j$ between a pair of nodes $(X_i, X_j) \in \boldsymbol{V}$ denotes probabilistic dependence between the corresponding variables; also, it denotes that X_i is a *direct cause* of X_j and that X_j is a *direct effect* of X_i. The absence of an edge between (X_i, X_j) denotes probabilistic conditional independence between these variables; more specifically, there is a set of variables $\boldsymbol{Z}$, such that conditioning on $\boldsymbol{Z}$ renders X and Y independent. If X_i and X_j are connected by an edge in either direction, we say that X_i and X_j are *adjacent*; we denote the set of nodes that are adjacent to X_i as $\boldsymbol{Adj}(X_i)$.

An *undirected path* (often called a path) π from X_i to X_j is a sequence of edges (without considering edge directions) that connects X_i to X_j such that no node is visited more than once. A *directed path* π from X_i to X_j is a sequence of directed edges that connects X_i to X_j such that no node is visited more than once. A node X_k on a path π is called a *collider* if its immediately preceding and succeeding nodes have directed edges into it: $X_{k-1} \rightarrow X_k \leftarrow X_{k+1}$. X_k is called an *unshielded collider* if its immediately preceding and succeeding nodes have directed edges into X_k but they are not adjacent to each other (Figure 1a); we also refer to this sub-structure as a *v-structure*. Similarly, X_k is a *shielded collider* if its immediately preceding and succeeding nodes are adjacent (Figure 1b). Finally, a node X_k on a path π is called a *non-collider* if it is not a collider.



(a) X_k is an unshielded collider.



(b) X_k is a shielded collider.

Figure 1: An example that shows two types of colliders.

The *parents* (or *causes*) of a node X_i are the nodes that immediately precede it (i.e., have an incoming arc into X_i); we denote parents of X_i as $\boldsymbol{Pa}(X_i)$. The nodes that immediately succeed X_i (i.e., have an outgoing edge from X_i) are called its *children* (or *effects*); we denote

[2]We use the terms directed edge and arc interchangeably because they are synonyms in CBNs.

children of X_i as $\boldsymbol{Ch}(X_i)$. Nodes are said to be *spouses* of each other if they have a common child. The *ancestors* of a node X_i, denoted as $\boldsymbol{An}(X_i)$, are the set of nodes that precede X_i and contain a directed path to X_i. Similarly, the *descendants* of X_i, denoted as $\boldsymbol{De}(X_i)$, are the set of nodes that succeed X_i and can be reached from X_i via a directed path. We denote the Markov blanket of a node X_i is a set that includes the parents of X_i, the children of X_i, and the spouses of X_i (the parents of X_i's children).

The second component of a BN is the parameter set $\boldsymbol{\Theta}$ that encodes the joint probability distribution over the set of variables $\boldsymbol{V} = \{X_1, X_2, ..., X_n\}$, which can be efficiently factored based on the parent-child relationships in the corresponding DAG using the local Markov condition [Neapolitan et al., 2004]. The *local Markov condition* states that each node is independent of its non-descendants given just the values of its parents. This property results in a compact representation of the joint probability distribution of the domain variables $\boldsymbol{V}$. According to the chain rule of probability, the joint probability distribution of variables $\boldsymbol{V}$ is as follows:

$$P(\boldsymbol{V}) = \prod_{i=1}^{n} P(X_i | X_1, ..., X_{i-1}). \tag{2.1}$$

Applying the local Markov condition to Equation (2.1) results in the following factorization of the joint probability distribution over variables $\boldsymbol{V}$:

$$P(\boldsymbol{V}) = \prod_{i=1}^{n} P(X_i | \boldsymbol{Pa}(X_i)). \tag{2.2}$$

where $\boldsymbol{Pa}(X_i)$ denotes the parents of X_i, which is the empty set when X_i has no parents.

Figure 2 shows Pearl's classic Holmes's burglar example [Kim and Pearl, 1983]. In the Bayesian network that corresponds to this example, the DAG consists of 5 nodes that correspond to 5 binary variables: burglary (B), earthquake (E), alarm (A), John calls (JC), and Mary calls (MC). The DAG also contains 4 edges that encode probabilistic dependencies among the variables. The edges $B \rightarrow A$ and $E \rightarrow A$ show that either a burglar or an earthquake can set the alarm on or off. Similarly, the edges $A \rightarrow JC$ and $A \rightarrow MC$ indicate that an alarm can cause Mary or John to make a call. In this example, B and E are parents of A (i.e., $\boldsymbol{Pa}(A) = \{B, E\}$); also, JC and MC are A's children (i.e., $\boldsymbol{Ch}(A) = \{JC, MC\}$). The parameters, which correspond to the probability distributions of each variable given its

parents, are shown in the tables. As an example, the probability of John calling given the alarm is on is 0.9 (i.e., $P(JC = t|A = t) = 0.90$) but if the alarm is off, there is 0.05 chance of John calling (i.e., $P(JC = t|A = f) = 0.05$). As this example shows, the parameters of the full joint probability distribution over these 5 binary variables reduces from $2^5 - 1 = 31$ to 10 when using BNs (the second columns in each table is redundant).

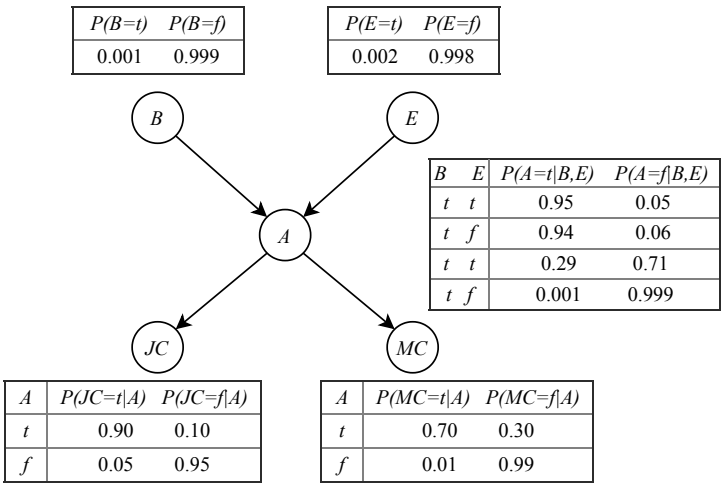


Figure 2: Pearl's Holmes's burglar example. This BN is composed of 5 nodes that correspond to 5 binary variables $\boldsymbol{V} = \{B, E, A, JC, MC\}$. It also contains 4 edges that encode the probabilistic dependencies among those variables. The local probability distributions of each variable given its parents are shown in the tables.

2.1.1 Directed acyclic graphs (DAGs) and their properties

A causal Bayesian network structure can be represented using a directed acyclic graph (DAG) when the *causal sufficiency assumption* holds. The causal sufficiency assumption means that the data-generating CBN does not contain a latent variable that is a common cause of two or more measured variables[3] [Spirtes et al., 2000]. This assumption, while being unrealistic in most practical applications, is nevertheless sometimes made because it significantly reduces the size of the search space of causal models, and it can provide

[3]There might be also variables that determine a specific sub-population from which the data is sampled; such variables are called selection variables.

some initial insights into the causal relationships among the measured variables. In this dissertation, I develop and evaluate some algorithms that assume causal sufficiency and others that do not.

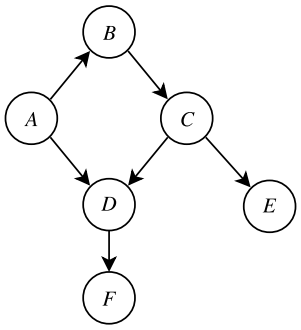
A DAG structure implies a set of marginal and conditional independence relations that are called the local and global Markov conditions. The *local Markov condition*, as described earlier, states that each node is independent of its non-descendants given its parents. This property provides a compact representation of the joint probability distribution that is associated with a DAG (see Equation (2.2)). The *global Markov condition* explicitly characterizes the complete set of independencies among disjoint sets of nodes in a DAG. It states that for all non-overlapping subsets of nodes $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$, if $\boldsymbol{A}$ and $\boldsymbol{B}$ are d-separated given $\boldsymbol{C}$ (i.e., $\boldsymbol{A} \perp\!\!\!\perp_d \boldsymbol{B}|\boldsymbol{C}$) then $\boldsymbol{A}$ and $\boldsymbol{B}$ are independent conditional on $\boldsymbol{C}$ (i.e., $\boldsymbol{A} \perp\!\!\!\perp \boldsymbol{B}|\boldsymbol{C}$). Global and local Markov conditions can be read from the DAG by applying d-separation criterion [Pearl, 2003], which is as follows:

Definition 2.1.1. (d-separation) Let $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$ be a DAG, $X_i, X_j, \in \boldsymbol{V}$ be two variables, and $\boldsymbol{Z} \subset \boldsymbol{V} \backslash \{X_i, X_j\}$ be a subset of variables that excludes X_i, X_j. Then X_i and X_j are *d-separated* given a disjoint set of nodes $\boldsymbol{Z}$ ($X_i \perp\!\!\!\perp_d X_j|\boldsymbol{Z}$) if and only if all (undirected) paths from X_i to X_j are blocked by $\boldsymbol{Z}$ (i.e., there is no active path between X_i and X_j). A path π between X_i and X_j is blocked by $\boldsymbol{Z}$ if it includes either:

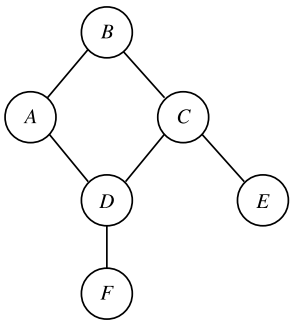
- A collider node Z_i, and neither Z_i nor its descendants are in $\boldsymbol{Z}$. A collider node is a node with converging arrows (e.g., D is a collider node in sub-path $A \rightarrow D \leftarrow C$ in the example shown in Figure 3a); or,
- A non-collider node Z_i, and Z_i is in $\boldsymbol{Z}$. In Figure 3a, C is a non-collider node in sub-paths $B \rightarrow C \rightarrow E$ or $D \leftarrow C \rightarrow E$.

In the DAG shown in Figure 3a, there are two undirected paths between A and E: (1) $A \rightarrow B \rightarrow C \rightarrow E$, and (2) $A \rightarrow D \leftarrow C \rightarrow E$. As mentioned earlier, A and E are d-separated given a set $\boldsymbol{Z}$ when both of these paths are blocked by that set. The first path is blocked if the conditioning set includes either of the non-collider nodes B or C on this path. To block the second path, the conditioning set should not include the collider node D or its descendant F. Therefore, $\boldsymbol{Z} = \{B\}$ or $\boldsymbol{Z} = \{C\}$ are two sets that d-separate A and

E; we denote these d-separations by $A \perp\!\!\!\perp_d E|B$ and $A \perp\!\!\!\perp_d E|C$, respectively. Similarly, if two nodes are not d-separated by a set, they are *d-connected* by it. This means that there is at least one path that remains active. For instance, A and E are marginally d-connected (i.e., $A \not\perp\!\!\!\perp_d E$) since path $A \to B \to C \to E$ is active.



(a) An example DAG $\mathcal{G}$.

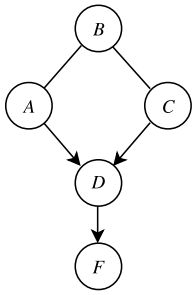


(b) The skeleton of $\mathcal{G}$.

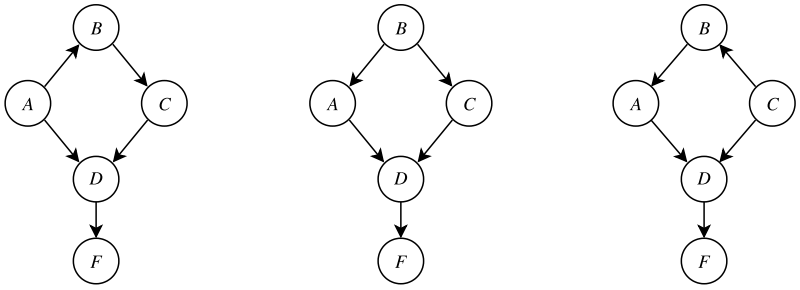
Figure 3: An example DAG $\mathcal{G}$ and its skeleton.

Multiple DAGs sometimes encode the same d-separation relationships over the same set of nodes. A set of DAGs that have the same d-separation properties form a *Markov equivalence class* of DAGs [Verma and Pearl, 1990]. Two DAGs are Markov equivalent if and only if (1) they have the same skeleton and (2) they have the same unshielded colliders (i.e., v-structures). A *skeleton* is composed of all edges that are included in the graph without considering edge orientations (i.e., adjacencies). Figure 3b shows the skeleton of the DAG $\mathcal{G}$ given in Figure 3a. Also, an unshielded collider refers to a collider node in which there are at least two parents that are not adjacent to each other.

Markov equivalence class of DAGS can be represented by a graph called a *completed partially directed acyclic graph* (*CPDAG*), also known as a *pattern*. A pattern is a graph that contains both directed ($\to$) and undirected ($—$) edges. If none of the DAGs in $\boldsymbol{\mathcal{G}}$ contain an edge between X_i and X_j, then there is no edge between X_i and X_j in the pattern. If $X_i \to X_j$ exists in every DAG in $\boldsymbol{\mathcal{G}}$, then $X_i \to X_j$ appears in the pattern; otherwise, if some DAGs in $\boldsymbol{\mathcal{G}}$ have $X_i \to X_j$ and other DAGs have $X_j \to X_i$, then $X_i — X_j$ appears in the pattern. The graph in Figure 4a shows a pattern that represents the Markov equivalence class of DAGs, which are shown in Figure 4b.



(a) A pattern $\mathcal{G}_p$ of the DAGs in $\boldsymbol{\mathcal{G}}$.



(b) The DAGs that belong to set $\boldsymbol{\mathcal{G}}$.

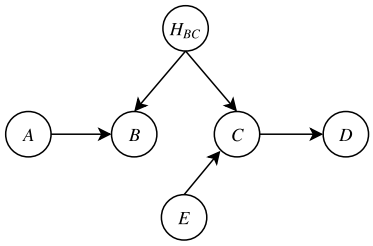
Figure 4: An example pattern shown in (a) that represents a Markov equivalence class of the DAGs in $\boldsymbol{\mathcal{G}}$ shown in (b).

2.1.2 Maximal ancestral graphs (MAGs) and their properties

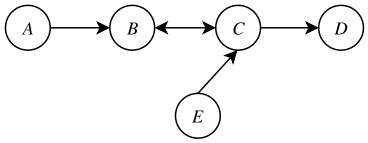
In practice, some variables may not be measured or recorded; such variables are called *latent* or *hidden variables*. Also, there might be variables that determine a specific subpopulation from which the data is sampled; such variables are called *selection variables*. Although some methods have been developed to perform causal inference under selection bias [Cooper, 1995, Spirtes et al., 1995, Richardson et al., 2002], in this dissertation, we assume the samples are drawn randomly from the population and the selection bias does not hold. Therefore, the definitions and discussions in this section are restricted to causal models that may include latent variables but not selection variables. The research to model selection is an area for future research.

DAGs are not closed under marginalization in the presence of latent variables. To illus-

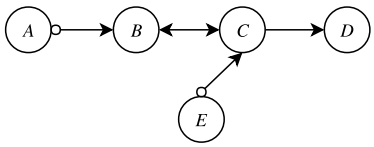
trate this, let the DAG $\mathcal{G}$ shown in Figure 5a be the ground truth Bayesian network that is generating the data. This DAG includes 5 observed variables $\boldsymbol{O} = \{A, B, C, D, E\}$ and one latent variable $\boldsymbol{L} = \{H_{BC}\}$ that is an unmeasured common cause (i.e., *latent confounder*) of B and C. In this DAG, the independence relations that hold among the observed variables in $\boldsymbol{O}$ are as follows: $A \perp\!\!\!\perp \{C, D, E\}$, $B \perp\!\!\!\perp D|C$, $B \perp\!\!\!\perp E$, and $D \perp\!\!\!\perp E|C$. However, there is no DAG that contains only these 5 observed variables and entails all and only these independence relationships without entailing either additional or fewer independence constraints.



(a) The ground truth DAG $\mathcal{G}$.



(b) An estimated MAG $\mathcal{M}$ in the large sample limit.



(c) An estimated PAG $\mathcal{P}$ in the large sample limit.

Figure 5: An example that shows DAGs are not closed under marginalization if there are latent variables. In this example, variables $\{A, B, C, D, E\}$ are observed while H_{BC} is a latent confounder of B and C.

Maximal ancestral graphs (MAGs) [Richardson et al., 2002] are graphical objects that encode independence relationships that hold among the observed variables in a DAG that may include both observed and latent variables[4]. MAGs are mixed graphs that include directed ($\rightarrow$) and bi-directed ($\leftrightarrow$) edges[5]. Similar to DAGs, MAGs do not contain any *directed* or *almost directed cycles*. A directed cycle occurs when there is X_i and X_j such that there is a directed path from X_i to X_j and a directed path from X_j to X_i. An almost directed

[4]MAGs do not provide any information on the structure among latent variables; rather, they implicitly model latent variables.

[5]MAGs use undirected edge (—) to model selection variables, which we do not consider in this dissertation.

cycle occurs when there is a directed path from X_i to X_j and $X_j \leftrightarrow X_i$. Let $\boldsymbol{V} = \boldsymbol{O} \cup \boldsymbol{L}$ be a set of variables that includes two non-overlapping sets of observed variables $\boldsymbol{O}$ and latent variables $\boldsymbol{L}$. In order to obtain a MAG $\mathcal{M} = (\boldsymbol{O}, \boldsymbol{E}')$ from DAG $\mathcal{G} = (\boldsymbol{O} \cup \boldsymbol{L}, \boldsymbol{E})$, we apply the following two steps:

1. For every pair of nodes $(X_i, X_j) \in \boldsymbol{O}$, add an undirected edge X_i — X_j if and only if there exists an inducing path (defined below) relative to $\boldsymbol{L}$ between X_i and X_j in $\mathcal{G}$.
2. Orient X_i — X_j as:
   - $X_i \rightarrow X_j$ if $X_i \in \boldsymbol{An}(X_j)$, or
   - $X_i \leftarrow X_j$ if $X_j \in \boldsymbol{An}(X_i)$, or
   - $X_i \leftrightarrow X_j$ if $X_j \notin \boldsymbol{An}(X_i)$ and $X_i \notin \boldsymbol{An}(X_j)$.

An inducing path is defined as follows [Verma and Pearl, 1990, Richardson et al., 2002]:

Definition 2.1.2. (Inducing path) A path π between X_i and X_j is called *inducing* relative to $\boldsymbol{L}$ if and only if every non-collider on π (except the endpoints) is in $\boldsymbol{L}$ and every collider on π is an ancestor of either X_i or X_j.

DAG to MAG conversion generates a marginal graph that represents the ancestral relationships that exist among the observed variables $\boldsymbol{O}$ in DAG $\mathcal{G}$ that contains latent variables. The presence of an edge between two variables X_i and X_j in MAG $\mathcal{M}$ corresponds to a conditional dependence in $\mathcal{G}$ since there is an inducing path between them, while the absence of an edge corresponds to conditional independence since there is at least one subset of variables $\boldsymbol{Z} \backslash \{X_i, X_j\} \in \boldsymbol{O}$ (which can be possibly an empty set) such that $X_i \perp\!\!\!\perp X_j | \boldsymbol{Z}$. Figure 5b is a MAG that represents DAG $\mathcal{G}$ in Figure 5a, which can be obtained by applying the two steps mentioned above.

Conditional independence relationships among observed variables can be read off ancestral graphs (i.e., MAGs or their Markov equivalence class) via m-separation criterion [Richardson et al., 2002], which is a generalization of d-separation and is defined as follows:

Definition 2.1.3. (m-separation) Let $\mathcal{M} = (\boldsymbol{V}, \boldsymbol{E})$ be an ancestral graph, $X_i, X_j \in \boldsymbol{V}$ be two nodes, and $\boldsymbol{Z} \subset \boldsymbol{V} \backslash \{X_i, X_j\}$ be a subset of nodes that excludes X_i, X_j. X_i and X_j

are m-separated by $\boldsymbol{Z}$ if there is no m-connecting path between X_i and X_j given $\boldsymbol{Z}$ (i.e., all paths are blocked by $\boldsymbol{Z}$). A path π is blocked by $\boldsymbol{Z}$ if it includes either:

- A collider node Z_i, and neither Z_i nor its descendants are not in $\boldsymbol{Z}$. A collider node is a node with converging arrows (e.g., in Figure 5c, B in sub-path $A \circ\!\!\rightarrow B \leftrightarrow C$ is a collider node); or,
- A non-collider node Z_i, and Z_i is in $\boldsymbol{Z}$. In Figure 5c, C in sub-path $E \circ\!\!\rightarrow C \rightarrow D$ is a non-collider node.

Likewise, if two nodes are not m-separated by a set, they are *m-connected* by it, which means that there is at least one path that remains active. In the MAG shown in Figure 5b, A and D are marginally m-separated (i.e., $A \perp\!\!\!\perp_m D$) since the only path $A \rightarrow B \leftrightarrow C \rightarrow D$ is blocked by the collider node B. However, A and D are m-connected conditioned on B (i.e., $A \not\perp\!\!\!\perp_m D|B$) since path $A \rightarrow B \leftrightarrow C \rightarrow D$ becomes active.

Similar to DAGs, some MAGs may entail the same m-separation properties over the same set of nodes. Such MAGs belong to the same *Markov equivalence class* which is represented by an entity called a *partial ancestral graph* (*PAG*). DAGs are to CPDAGs as MAGs are to PAGs. Two MAGs are Markov equivalent if and only if (1) they have the same skeleton, (2) they have the same unshielded colliders, and (3) if both MAGs include a discriminating path π for node Z, then Z is a collider on π in one MAG if and only if it is a collider on π in the other MAG [Ali et al., 2009], where the discriminating path is defined as follows:

Definition 2.1.4. (Discriminating path) A path π between X and Y is called discriminating for Z if X is not adjacent to Y and every node on π from X to Z is a collider and a parent of Y.

Conditional independence relationships in PAGs are represented using an expanded set of edge marks: directed ($\rightarrow$), bi-directed ($\leftrightarrow$), partially directed ($\circ\!\!\rightarrow$), and non-directed ($\circ\!\!-\!\!\circ$), where a circle indicates uncertainty about whether the associated endpoint of an edge is an arrow or not[6]. Figure 5c shows the PAG $\mathcal{P}$ that represents the Markov equivalence class of all MAGs that encode the causal relationships among the observed variables in DAG $\mathcal{G}$ (Figure 5a). In Figure 5c, the edge $B \leftrightarrow C$ represents that B and C are both caused

[6]Similar to MAGs, PAGs use undirected edge (—) to model selection variables.

by one or more latent variables (i.e., they are confounded by a latent variable). The edge $C \to D$ represents that C is a cause of D and that there are no latent confounders of C and D. The edge $A \circ\!\!\to B$ represents that either A causes B, A and B are confounded by a latent variable, or both. Another edge possibility, which does not appear in the example, is $X \circ\!\!-\!\!\circ Y$, which is compatible with the true causal model having X as a cause of Y, Y as a cause of X, a latent confounder of X and Y, or some acyclic combination of these three possibilities.

2.1.3 Faithfulness and Markov conditions

There are two conditions that bind the graphs and probability distributions: the faithfulness and Markov conditions. These are commonly used assumptions in causal discovery algorithms, which are defined as follows. Let $M = (\mathcal{G}, \boldsymbol{\Theta})$ be a CBN model in which $\boldsymbol{\Theta} = P(\boldsymbol{V})$ is a probability distribution over a set of variables $\boldsymbol{V}$ that is encoded by DAG $\mathcal{G}$.

The distribution $P(\boldsymbol{V})$ is *faithful* to $\mathcal{G}$ if every independence constraint that holds in the distribution $P(\boldsymbol{V})$ entails the corresponding d-separation condition in $\mathcal{G}$. That is, if X and Y are conditionally independent given $\boldsymbol{Z}$ (i.e., $X \perp\!\!\!\perp Y|\boldsymbol{Z}$) according to distribution $P(\boldsymbol{V})$, then X is d-separated from Y given $\boldsymbol{Z}$ (i.e., $X \perp\!\!\!\perp_d Y|\boldsymbol{Z}$) in $\mathcal{G}$.

The converse of the faithfulness condition is known as the *Markov condition*, which states that every d-separation condition (e.g., $X \perp\!\!\!\perp_d Y|\boldsymbol{Z}$) in $\mathcal{G}$ entails an independence (e.g., $X \perp\!\!\!\perp Y|\boldsymbol{Z}$) in $\mathcal{G}$ in $P(\boldsymbol{V})$. If both the faithfulness and Markov conditions hold, it implies that the d-separation relationships in $\mathcal{G}$ have a one-to-one correspondence to independence constraints in $P(\boldsymbol{V})$. This correspondence enables us to learn $\mathcal{G}$ from data generated by a CBN model M, when there is sufficient data for doing so.

2.1.4 Context-specific independence (CSI)

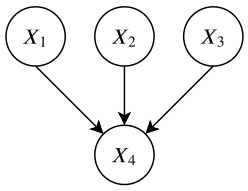
A standard DAG structure $\mathcal{G}$ encodes the conditional independence relationships that hold among a set of variables $\boldsymbol{V}$. Any such conditional independence relationship is represented in $\mathcal{G}$ if it holds for all combinations of values of the variables involved. Despite their desirable properties, DAGs are unable to capture more refined conditional independence re-

lationships that are true in specific contexts. The notion of context-specific independence (CSI) in Bayesian networks was introduced in [Boutilier et al., 1996] to represent the independence relationships that hold between a variable and some but not all combinations of values of its parents:

Definition 2.1.5. (Context-specific independence (CSI)) [Boutilier et al., 1996] Let $\boldsymbol{X}$, $\boldsymbol{Y}$, $\boldsymbol{Z}$, and $\boldsymbol{C}$ be pairwise disjoint sets of variables, and $\boldsymbol{c}$ be a particular assignment to $\boldsymbol{C}$ (i.e., a context). $\boldsymbol{X}$ and $\boldsymbol{Y}$ are contextually independent given $\boldsymbol{Z}$ and the context $\boldsymbol{c}$ ($\boldsymbol{X} \perp\!\!\!\perp_c \boldsymbol{Y} | \{\boldsymbol{Z}, \boldsymbol{C} = \boldsymbol{c}\}$) if $P(\boldsymbol{X}|\boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{c}) = P(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{c})$ whenever $P(\boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{c}) > 0$.

Figure 6 shows an example CBN that includes two CSI structures:

- $X_4 \perp\!\!\!\perp_c \{X_2, X_3\} | X_1 = 0$: This means that X_4 is independent of $\{X_2, X_3\}$ when conditioned on $X_1 = 0$, which implies that $X_2 \rightarrow X_4$ and $X_3 \rightarrow X_4$ can be removed when $X_1 = 0$, since changing X_2 and X_3 do not affect the distribution of X_4 (i.e., $P(X_4|X_2, X_3, X_1 = 0) = P(X_4|X_1 = 0)$).
- $X_4 \perp\!\!\!\perp_c X_3 | \{X_1 = 1, X_2 = 1\}$: This indicates that X_4 is conditionally independent of X_3 when $\{X_1 = 1, X_2 = 1\}$, which implies that $X_3 \rightarrow X_4$ can be removed when $\{X_1 = 1, X_2 = 1\}$, since changing X_3 does not affect the distribution of X_4 (i.e., $P(X_4|X_3, X_2 = 1, X_1 = 1) = P(X_4|X_2 = 1, X_1 = 1)$).



X_1	X_2	X_3	$P(X_4\|X_1, X_2, X_3)$
0	0	0	0.90
0	0	1	0.90
0	1	0	0.90
0	1	1	0.90
1	0	0	0.73
1	0	1	0.46
1	1	0	0.15
1	1	1	0.15

Figure 6: This CBN example contains two context-specific independence (CSI) structures: $X_4 \perp\!\!\!\perp_c \{X_2, X_3\} | X_1 = 0$ and $X_4 \perp\!\!\!\perp_c X_3 | \{X_1 = 1, X_2 = 1\}$. The first CSI structure, for example, indicates that X_4 is conditionally independent of $\{X_2, X_3\}$ when conditioned on $X_1 = 0$, which means that $X_2 \rightarrow X_4$ and $X_3 \rightarrow X_4$ do not affect the distribution of X_4 when $X_1 = 0$ (i.e., $P(X_4|X_2, X_3, X_1 = 0) = P(X_4|X_1 = 0)$). Such CSI structures are hidden beneath the DAG structure.

As shown in Figure 6, local CSI structures in a CBN model $M = (\mathcal{G}, \boldsymbol{\Theta})$ can be determined by utilizing local distributions in $\boldsymbol{\Theta}$ for each variable X_i given its parents $\boldsymbol{Pa}(X_i)$ and a particular context $\boldsymbol{c}$. Given such a CSI relationship $X_i \perp\!\!\!\perp_c Y | \{\boldsymbol{Pa}(X_i), \boldsymbol{C} = \boldsymbol{c}\}$, where $\boldsymbol{C} \subseteq \boldsymbol{Pa}(X_i)$, we can derive a new parent structure for X_i that encodes the CSI by removing the edge $Y \rightarrow X$. By repeating this procedure for all variables X_i, the instance-specific CBN structure $\mathcal{G}_{IS}$ can be derived from $M = (\mathcal{G}, \boldsymbol{\Theta})$ that encodes the CSI parent structures that hold for given a test instance T (i.e., the contexts are determined according to the values of the variables in T). Then, we can define *CSI-separation* as follows (adapted from [Boutilier et al., 1996]):

Definition 2.1.6. (CSI-separation) Let $M = (\mathcal{G}, \boldsymbol{\Theta})$ be a CBN model that includes some CSI parent structures encoded in its distribution component $\boldsymbol{\Theta}$ and T be an instance sampled from M, which also includes CSI parent structures. Also, let $\mathcal{G}_{IS}$ be an instance-specific CBN structure for T in which the spurious edges due to CSI parent structures are removed. We say that $\boldsymbol{X}$ is CSI-separated from $\boldsymbol{Y}$ given $\boldsymbol{Z}$ in context $\boldsymbol{c}$ in $\mathcal{G}$ if and only if $\boldsymbol{X}$ is d-separated from $\boldsymbol{Y}$ given $\{\boldsymbol{Z}, \boldsymbol{C}\}$ in $\mathcal{G}_{IS}$.

Therefore, by transforming $\mathcal{G}$ to $\mathcal{G}_{IS}$ for a given test instance T, we can use d-separation on $\mathcal{G}_{IS}$, as we do in standard CBNs, to define faithfulness and Markov conditions described in Section 2.1.3.

These types of local CSI structures cannot be captured completely in the structure of standard CBNs, wherein the CBN structure is invariant to CSI relationships. In this dissertation, I introduce instance-specific CBN structure learning algorithms to model such local structures in a test instance T.

2.2 CBN Structure Discovery Algorithms

Considerable CBN research has focused on score-based and constraint-based approaches, although other approaches, such as hybrid methods, have been developed and investigated [Peters et al., 2012]. A score-based method typically uses a scoring function to derive

the score of each candidate CBN structure. The score is then incorporated into a search algorithm, which is often a greedy heuristic, to find the highest scoring CBN structure in the hypothesis space of the possible structures. I provide an overview of score-based methods in Section 2.2.1. A constraint-based approach uses tests of conditional independence; causal discovery occurs by finding patterns of conditional independence and dependence that are likely to be present only when particular causal relationships exist. An overview of these methods is discussed in Section 2.2.2.

2.2.1 Score-based approaches

A score-based method involves two main components: (1) a scoring metric and (2) a search algorithm. Given a dataset D, which is a flat-file in which columns denote domain variables $\boldsymbol{V} = \{X_1, X_2, ..., X_n\}$ and rows denote samples (cases), and possibly prior knowledge or belief, a score is derived for a CBN that quantifies how well the model describes the data. The score is then incorporated into a search algorithm that seeks to find the highest scoring CBN structure in the hypothesis space of the possible structures. However, the number of the possible structures grows super-exponentially with respect to the number of domain variables; indeed, finding the highest scoring structure is an NP-hard problem [Chickering, 1996]. Nevertheless, some exact search methods have been developed that are applicable to small-sized graphs. Some examples of exact CBN structure learning methods include [De Campos et al., 2009] that uses a branch and bound technique, [Koivisto and Sood, 2004, Singh and Moore, 2005, Koivisto, 2012, Silander and Myllymaki, 2012] that utilize dynamic programming methods, and [Jaakkola et al., 2010, Bartlett and Cussens, 2013, Studenỳ and Haws, 2014] that apply integer linear programming approaches.

For larger graphs, which is the main focus of this dissertation research, the application of exact methods is computationally intractable. Therefore, several heuristic algorithms, such as greedy hill-climbing have been proposed. In the following sections, I review some scoring functions and heuristic search algorithms.

2.2.1.1 Scoring functions Different types of non-Bayesian and Bayesian scores have been developed and investigated to measure how well a CBN structure is supported by the data and background beliefs (priors). The simplest form of such a score is the likelihood score. However, maximizing the likelihood score will in general result in a highly connected CBN that overfits the data. Therefore, more sophisticated scores are designed to favor a model that not only matches the data better, but also has a simpler structure with fewer parameters. For example, the Bayesian information criterion (BIC) [Schwarz, 1978] is a scoring function that includes a likelihood criterion for rewarding goodness of data fit and a penalty term for penalizing the model's parameter complexity. The BIC score for a CBN $\mathcal{G}$ given a dataset D is defined as follows:

$$\mathrm{BIC}(\mathcal{G}, D) = \log P(D|\hat{\boldsymbol{\Theta}}, \mathcal{G}) - \frac{df}{2} \log N, \tag{2.3}$$

where $\log P(D|\hat{\boldsymbol{\Theta}}, \mathcal{G})$ is the log-likelihood of the data given $\mathcal{G}$, $\hat{\boldsymbol{\Theta}}$ denotes the maximum-likelihood parameters of $\mathcal{G}$, *df* corresponds to the number of free parameters in $\mathcal{G}$, and N is the sample size of the dataset D.

Another example of such scoring functions is the minimum description length (MDL) score [Rissanen, 1978], which is based on the MDL principle. The MDL principle selects the model that minimizes the sum of the encoding length of the model, which here is a CBN (considering both DAG and parameters), and the encoding length of the data using that model. In the case of CBN structure learning, the MDL is defined as follows [Daly et al., 2011]:

$$\mathrm{MDL}(\mathcal{G}, D) = -\log P(D|\hat{\boldsymbol{\Theta}}, \mathcal{G}) + \frac{df}{2} \log N + C_n, \tag{2.4}$$

where $\log P(D|\hat{\boldsymbol{\Theta}}, \mathcal{G})$ is the log-likelihood of the data given $\mathcal{G}$, $\hat{\boldsymbol{\Theta}}$ denotes the maximum-likelihood parameters of $\mathcal{G}$, *df* corresponds to the number of free parameters in $\mathcal{G}$, and C_n is defined as follows:

$$C_n = \sum_{i=1}^{n} (1 + |\boldsymbol{Pa}(X_i)|) \cdot \log n, \tag{2.5}$$

where n denotes the number of variables and $|\boldsymbol{Pa}(X_i)|$ is the number of parents of variable X_i. The MDL score given in Equation (2.4) includes an additional term, C_n, compared to

the BIC score given in Equation (2.3). This term becomes irrelevant as the sample size grows and MDL and BIC will become equivalent, consequently[7].

The first widely used Bayesian scoring function to score a CBN was derived by [Cooper and Herskovits, 1992], which is called the K2 score and defined as follows:

$$\mathrm{K2}(\mathcal{G}, D) = P(\mathcal{G}) \cdot \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}!, \tag{2.6}$$

where the first product term is over all n variables, the second product term is over the q_i parent instantiations of variable X_i, and the third product term is over all r_i values of variable X_i. The term N_{ijk} is the number of cases in dataset D in which variable $X_i = k$ and its parent $\boldsymbol{Pa}(X_i) = j$; also, $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. In Equation (2.6), $P(\mathcal{G})$ is the prior structure probability of CBN $\mathcal{G}$.

A generalization of K2 score is called Bayesian Dirichlet (BD) score and defined as follows [Cooper and Herskovits, 1992, Heckerman et al., 1995]:

$$\mathrm{BD}(\mathcal{G}, D) = P(\mathcal{G}) \cdot \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \cdot \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}, \tag{2.7}$$

where all terms are similar to Equation (2.6), except for the $\alpha_{(\cdot)}$ terms. α_{ijk} is a Dirichlet prior parameter that may be interpreted as representing "pseudo-counts" and $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$. Note that BD and K2 are equivalent if all hyperparameters α_{ijk} are set to 1. More details about derivation of Bayesian scores are given in Section 4.3.

Scoring criteria often have some desirable properties that make them efficient to be used in score-based searches; these criteria include decomposability, score equivalence, and consistency. A decomposable scoring function can be factorized into local terms that are a function of a node and its parents according to the DAG structure as follows:

$$S(\mathcal{G}, D) = \prod_{i=1}^{n} s(X_i, Pa(X_i)), \tag{2.8}$$

or equivalently

$$\log S(\mathcal{G}, D) = \sum_{i=1}^{n} \log s(X_i, Pa(X_i)). \tag{2.9}$$

[7]When incorporated in a BN learning algorithm, the BIC score is used to maximize the score while MDL is used to be minimized. Therefore, BIC and MDL are negative inverses of each other.

All scoring criteria described above are decomposable at the node level. This property leads to an efficient implementation of a score-based search algorithm, either exact or heuristic, since only the local changes need to be re-scored when we want to compare the scores of two DAGs while performing a search. For example, if we only add an edge into (or delete an edge into) a node X_i in a DAG, only X_i needs to be re-scored to compute the effect of this operation on the score of the DAG.

Another useful property of a scoring function is score equivalence. If two DAGs $\mathcal{G}_1$ and $\mathcal{G}_2$ are Markov equivalent, then S is a score equivalent function if and only if $S(\mathcal{G}_1, D) = S(\mathcal{G}_2, D)$. This property results in the same score for all the graphs that are in the same Markov equivalence class. For example, all the DAGs that are represented by a pattern will score the same when using a score equivalent criterion. This property allows us to search directly over the space of Markov equivalence classes.

Lastly, consistency of a scoring function is useful when we study its asymptotic properties. A score S is a consistent in the large sample limit if:

- S ranks DAG $\mathcal{G}_1$ that represents the data-generating distribution P higher than DAG $\mathcal{G}_2$ that does not represent P: $S(\mathcal{G}_1, D) > S(\mathcal{G}_2, D)$.
- If two DAGs $\mathcal{G}_1$ and $\mathcal{G}_2$ both represent the data-generating distribution P and $\mathcal{G}_1$ contains fewer parameters, then S ranks $\mathcal{G}_1$ higher than $\mathcal{G}_2$: $S(\mathcal{G}_1, D) > S(\mathcal{G}_2, D)$.

If a score is both decomposable and consistent, it is called a *locally consistent* score. BIC is score equivalent and locally consistent [Chickering, 2002].

2.2.1.2 Heuristic score-based algorithms As mentioned earlier, learning CBN structures is, in general, an NP-hard problem [Chickering, 1996]; hence, numerous heuristic search algorithms have been developed to explore the space of the CBN structures in computationally feasible and efficient ways [Daly et al., 2011, Koski and Noble, 2012]. Such algorithms usually involve applying local changes to the current model and replacing it with the one that leads to the greatest score improvement in a greedy fashion. Therefore, a heuristic search requires multiple components:

- A search space that consists of valid states of the problem, e.g., DAGs or patterns.

- A search operator to generate the legal neighboring states from the current state. For instance, if the states are DAGs, single edge addition, deletion, and reversal operations could be applied to the current state to generate neighboring DAGs; however, these operations should not induce cycles.
- A search method that identifies which neighboring state to select, e.g. greedy hill climbing.

Although efficient, greedy searches are prone to getting stuck in local maxima, several empirical solutions have been proposed to address this problem, including for instance, random restarts, simulated annealing, and TABU lists [Blum and Roli, 2003].

K2 algorithm is one of the earliest heuristic search methods for learning a CBN structure from data [Cooper and Herskovits, 1992]. K2 is a polynomial-time greedy search algorithm that assumes a prior ordering of the nodes is given. For computational efficiency, we can also assume the number of parents of each node is limited to a user-specified upper bound. Based on a predetermined ordering, the K2 algorithm iterates over the nodes to learn a set of parents for each of them. For each node X_i, the algorithm starts with no parent assigned to X_i. Then, it greedily adds as a parent of X_i the node that most improves the K2 score for X_i (from among the nodes that are located before X_i in the given ordering). The search stops when no further improvements can be achieved or the maximum number of parents is met. Despite being computationally efficient, providing a good ordering of nodes for K2 requires domain expertise or temporal ordering of the nodes and may not be available in many applications. The authors suggest the possibility of using multiple random orderings and choosing the best network found in doing so, but they do not evaluate this approach.

Another well-known heuristic algorithm is greedy equivalence search (GES) that operates on the space of equivalence class of CBNs (i.e., CPDAGs or patterns) [Chickering, 2002]. GES is a two-stage search. During the forward phase, it adds the single edge to the current graph that most improves the score; it stops when no further improvement can be achieved. Similarly, during the backward phase, it removes the single edge from the current graph that most improves the score; it stops when no further improvement can be achieved and returns the resultant graph. Under assumptions, GES learns the data generating CBN in the large sample limit. More details about GES are given in Section 4.2.

2.2.2 Constraint-based approaches

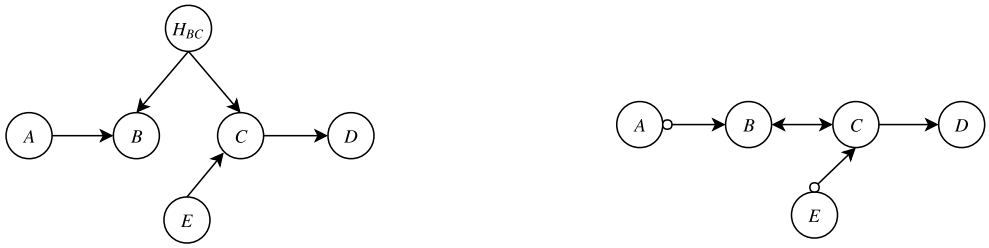
A constraint-based CBN structure learning algorithm searches for a set of Bayesian networks, all of which entail a particular set of conditional independence constraints (or simply *constraints*), that are judged to hold in a dataset of samples based on the results of tests applied to that data. It is usually not computationally or statistically feasible to actually test each possible constraint on the measured variables for more than a few dozen variables. Therefore, constraint-based algorithms typically use an efficient test schedule to prune the space of possible tests, and therefore, select a sufficient subset of constraints to test. Generally, the subset of constraint tests that are performed within a sequence of such tests depends upon the results of previous tests.

The PC algorithm [Spirtes et al., 2000] is one of the most well-known examples of constraint-based CBN structure learning methods, which assumes causal sufficiency (i.e., there are no unmeasured variables that cause two or more measured variables). PC takes as input dataset D, which is a flat-file in which columns denote domain variables $\boldsymbol{V}$ and rows denote observed samples (cases), and optional deterministic background knowledge, and it outputs a pattern, which represents Markov equivalence class of DAGs (see Section 2.1.1). PC learns the pattern in two main stages: the adjacency stage and the orientation stage. During the adjacency stage, the PC algorithm starts with a fully connected graph (i.e., all pairs of nodes are connected by an undirected edge). Then, for each adjacency X_i — X_j, it removes the edge if X_i and X_j become conditionally independent given some subset of the nodes that are adjacent to X_i (i.e., $\boldsymbol{Adj}(X_i)\backslash X_j$) or to X_j (i.e., $\boldsymbol{Adj}(X_j)\backslash X_i$). Once the skeleton is recovered, PC applies multiple edge orientation rules to orient as many arrowheads as possible in the output pattern [Spirtes et al., 2000].

Assuming the tests of conditional independence are correct, the pattern returned by PC represents as much about the true causal graph as can be determined from the conditional independence relations among the variables [Spirtes et al., 2000]. In particular, the PC algorithm is guaranteed to converge to the true pattern in the large sample limit, assuming the data-generating model is a CBN without latent confounders, the tests of conditional independence are correct, and the Markov and faithfulness conditions hold (see Section 2.1.3

for more detail about these conditions) [Spirtes et al., 2000].

Fast Causal Inference (FCI) [Spirtes et al., 2000] is another prominent constraint-based causal discovery algorithm, which can model latent variables. FCI takes as input observed sample data D and optional deterministic background knowledge, and it outputs a PAG, which represent the Markov equivalence class of DAGs with latent variables (see Section 2.1.2). Similar to PC, FCI learns the PAG by performing an adjacency search and applying orientation rules. Under assumptions, the FCI algorithm is guaranteed to recover the correct PAG with probability 1.0 in the large sample limit, even if there are latent confounders [Zhang, 2008]. As an example, Figure 7 shows in panel (b) the PAG that would be output by the FCI search if given a large enough sample of data from the data-generating CBN shown in panel (a), when the assumptions hold [Spirtes et al., 2000]. We discuss the FCI algorithm in more detail in Section 3.2.



(a) The data-generating CBN.

(b) The PAG that is output.

Figure 7: The PAG in (b) is learnable in the large sample limit from observational data generated by the causal model in (a), where H_{BC} is a latent variable and the other variables are measured.

2.3 CBN Structure Discovery Performance

In this section, I describe the evaluation measures that are used to calculate the structural similarity of the discovered CBN $\mathcal{G}_{output}$ versus the gold-standard CBN $\mathcal{G}_{truth}$. One such measure is structural Hamming distance (SHD) that counts the edge modifications, which can include added, deleted, and reoriented edges, by comparing each possible edge in $\mathcal{G}_{output}$ and $\mathcal{G}_{truth}$. We define two versions of SHD for patterns in Section 4.5.1.1 and three versions

of SHD for PAGs in Section 3.6.1.1.

Other performance criteria we use to evaluate discrimination are precision (P) and recall (R) for adjacencies and arrowheads:

- **Adjacency precision (AP)**: we compute the ratio of correctly predicted edges in $\mathcal{G}_{output}$ to all predicted edges in $\mathcal{G}_{output}$ (without considering orientations of edges) as follows:

$$\text{AP} = \frac{\#\text{correctly predicted adjacencies}}{\#\text{predicted adjacencies}} \tag{2.10}$$

- **Adjacency recall (AR)**: we compute the ratio of correctly predicted edges in $\mathcal{G}_{output}$ to all true edges in $\mathcal{G}_{truth}$ (without considering the edges' orientations) as follows:

$$\text{AR} = \frac{\#\text{correctly predicted adjacencies}}{\#\text{true adjacencies}} \tag{2.11}$$

- **Arrowhead precision (AHP)**: considering the pairs of variables that have an edge between them in the predicted graph $\mathcal{G}_{output}$, we compute the ratio of correctly predicted arrowheads in $\mathcal{G}_{output}$ to all predicted arrowheads in $\mathcal{G}_{output}$ as follows:

$$\text{AHP} = \frac{\#\text{correctly predicted arrowheads}}{\#\text{predicted arrowheads}} \tag{2.12}$$

- **Arrowhead recall (AHR)**: considering the pairs of variables that have an edge between them in the ground-truth graph $\mathcal{G}_{truth}$, we compute the ratio of correctly predicted arrowheads in $\mathcal{G}_{output}$ to all true arrowheads in $\mathcal{G}_{truth}$ as follows:

$$\text{AHR} = \frac{\#\text{correctly predicted arrowheads}}{\#\text{true arrowheads}} \tag{2.13}$$

We also develop specialized subtypes of these measures when we are evaluating methods using data that have been generated by instance-specific models; these measures are described in Sections 4.5.1.1 and 5.3.1.1.

3.0 CBN Structure Learning Using Bayesian Scoring of Constraints

As mentioned in Chapter 2, two main categories of algorithms to learn CBN structures from observational data are constraint-based and score-based approaches. These two approaches each have significant, but different, strengths and weaknesses. The constraint-based approach can model and discover causal models with latent (hidden) variables relatively efficiently (depending upon what the true causal structure is, which variables are measured, and how many and what kind of latent confounders exist). This capability is important because oftentimes there are latent variables that cause measured variables to be statistically associated (confounded). If such confounded relationships are not considered, erroneous causal discoveries may occur. The constraint-based approaches do not, however, provide a meaningful summary score of the chance that a causal model is correct. Rather, a single model is derived and output, without quantification regarding how likely it is to be correct, relative to alternative models. In addition, while constraint-based methods can incorporate domain beliefs known with certainty (e.g., that a gene X is regulated by gene Y), they cannot incorporate domain beliefs about what is likely but not certain (e.g., that there is a 0.8 chance that gene X is regulated by gene Z).

In contrast, score-based methods can generate and probabilistically score multiple models, outputting the most probable one. By doing so, they may increase the chance of finding a model that is causally correct. They also can quantify the probability of the top-scoring model relative to other models that are considered in the search. The top-scoring model might be close, or alternatively far away, from other models, which could be helpful to know. In addition, score-based methods can incorporate as prior probabilities domain beliefs about what is likely but not certain, which is a common situation. However, the Bayesian scoring of causal models that contain latent confounders is computationally very expensive. In particular, there are two major problems when learning a CBN with latent variables using score-based approaches:

- Problem 1 (model search): There is an infinite space of latent-variable models, both in terms of parameters and latent structure. Even when restrictions are assumed, the

search space generally remains enormous in size, making it challenging to find the highest scoring CBNs.

- Problem 2 (model scoring): Scoring a given CBN with latent variables is challenging. In particular, marginalizing over the latent variables greatly complicates Bayesian scoring in terms of accuracy and computational tractability.

Consequently, the practical application of score-based methods is largely relegated to CBNs that do not contain latent variables, which significantly decreases the general applicability of these methods for causal discovery.

This chapter describes a novel hybrid approach, called Bayesian scoring of constraints (BSC), that combines strengths of the constraint-based and score-based approaches to learn CBN structures from observational data in the presence of latent variables [Jabbari et al., 2017b]. BSC uses a Bayesian method to score an independence constraint, it then derives the probability that the set of independence constraints associated with a given causal model are jointly correct, rather than scoring the CBNs directly. The posterior probability of a CBN is taken to be proportional to the posterior probability that the constraints that characterize that CBN (or class of CBNs) are jointly true, which enables us to score multiple causal models and output the most probable one(s). The BSC approach, therefore, attenuates both of the following problems of score-based approaches:

- Problem 1 (model search): In the BSC approach, the search space is finite, not infinite as in the general score-based approach, because the number of possible constraints on a given set of measured variables is finite.
- Problem 2 (model scoring): In a constraint-based approach, the constraints are assessed on measured variables only, as discussed in Section 2.2.2. Thus, when BSC uses a Bayesian approach to derive the probability of a set of constraints and thereby score a CBN, it needs only to consider measured variables. In contrast, a traditional score-based approach must marginalize over latent variables, which is a difficult and computationally expensive operation.

In the remainder of this chapter, I first review the related work in Section 3.1. Then, I discuss in more detail a widely-used constraint-based CBN learning algorithm, namely fast

causal inference (FCI) [Spirtes et al., 2000] in Section 3.2. I introduce a Bayesian method to compute the probability of an independence constraint, called (BSC), in Section 3.3. In Section 3.4, I describe how to incorporate the BSC method into a constraint-based method (e.g., FCI) to learn PAG models. Then, in Section 3.5, I introduce three approaches to score and rank the learned PAG models by approximating the posterior probability of each PAG as the joint probability of the constraints that characterize that PAG. Finally, I present the results on the performance of the methods introduced in this chapter using simulated datasets from both randomly generated CBN models and manually constructed CBN models in Section 3.6.

3.1 Related Work

Several heuristic algorithms have been developed and investigated for scoring CBNs containing latent variables. An early algorithm for this task was developed by [Friedman, 1998]; it interleaved structure search with the application of expectation-maximization (EM) to optimize the Bayesian score within EM iterations when learning the structure. Other approaches include those based on variational EM [Beal and Zoubin, 2003] and a greedy search that incorporates EM [Borchani et al., 2006]. These and related approaches were primarily developed to deal with missing data, rather than latent variables for which all data are missing for those variables.

Other Bayesian algorithms have been developed to score CBNs with latent variables, including methods that use a Laplace approximation [Heckerman et al., 1999], an approach that uses EM and a form of clustering [Elidan and Friedman, 2005], and a structural expectation propagation method [Lazic et al., 2013]. However, these methods do not search over the space of all CBNs that include a given set of measured variables. Rather, they require that the user manually provides the proposed CBN models to be scored; they search a very restricted space of models, such as bipartite graphs [Lazic et al., 2013] or trees of hidden structure [Choi et al., 2011, Elidan and Friedman, 2005], or they score ancestral relations between pairs of variables [Parviainen and Koivisto, 2011]. Thus, within a Bayesian frame-

work, the automated discovery of CBNs that contain latent variables remains an important open problem.

Researchers have also developed algorithms that combine constraint-based and score-based approaches for learning CBNs [Claassen and Heskes, 2012, Dash and Druzdzel, 1999, De Campos et al., 2003, Magliacane et al., 2016, Nandy et al., 2018, Ogarrio et al., 2016, Singh and Valtorta, 1995, Triantafillou et al., 2014, Tsamardinos et al., 2006]. However, most of these hybrid methods, do not include the possibility that the CBNs being modeled contain latent variables. Exceptions include [Claassen and Heskes, 2012, Magliacane et al., 2016, Ogarrio et al., 2016, Triantafillou et al., 2014]], which do model latent variables.

In [Claassen and Heskes, 2012], a Bayesian method is proposed to score and rank order constraints; then, it uses those rank-ordered constraints as inputs to a constraint-based causal discovery method. However, it does not derive the posterior probability of a causal model from the probability of the constraints that characterize the model. The method in [Ogarrio et al., 2016] models the possibility of latent confounders but it does not provide any quantification of the output graph. In [Triantafillou et al., 2014], a method is proposed to convert p-values to posterior probabilities of adjacencies and non-adjacencies in a graph; then, those probabilities are used to identify neighborhoods of the graph in which all relations have probabilities above a certain threshold. This method is, in fact, a post-processing step on the skeleton of the output network and is not applicable to convert p-values to probabilities during the search phase of constraint-based learning. It also does not provide a way of computing the posterior probability of the whole output PAG. [Magliacane et al., 2016] introduces a logic-based method to reconstruct ancestral relations and score their marginal probabilities; it does not provide the probability of the output graph, however. In [Magliacane et al., 2016], authors mentioned that modeling the relationships among the constraints may be an improvement; in this dissertation, I introduce an empirical way of modeling such relationships.

The research reported in [Hyttinen et al., 2014] is the closest previous work of which we are aware to that introduced in this dissertation (see Section 3.4 below). It describes how to score constraints on graphs by treating the constraints as independent of each other. The

method is very expensive computationally, however, and is reported as working for up to 7 measured variables only. The method we introduce was feasibly applied to a dataset containing 70 variables and plausibly is practical for considerably larger datasets (see Section 3.6 below). Also, the method in [Hyttinen et al., 2014], as described, is limited to deriving just the most probable graph, rather than deriving a set of graphs, as we do, which can be rank ordered, compared, and used to perform selective model averaging that derives (for example) distributions over edge types.

In this chapter, I introduce a hybrid approach, called BSC, that combines strengths of constraint-based and score-based methods. The BSC method derives the probability that relevant constraints are true. Consider a CBN model (or an equivalence class of CBN models) that entails a set of conditional independence constraints over the distribution of the measured variables. In the BSC approach, the probability of the model being correct is equal to the probability that the constraints that uniquely characterize the CBN model (or class of CBN models) are true. The BSC method exhibits the computational efficiency of a constraint-based method combined with the ability of a score-based approach to quantitatively compare alternative causal models according to their posterior probabilities.

3.2 Overview of the FCI Algorithm

Constraint-based algorithms are often used to discover the causal structure in a causally insufficient system. That is, there is an unknown DAG $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$ over a set of random variables $\boldsymbol{V}$ that includes both observed $\boldsymbol{O}$ and latent $\boldsymbol{L}$ variables (i.e., $\boldsymbol{V} = \boldsymbol{O} \cup \boldsymbol{L}$). These algorithms rely on two main assumptions: Markov and faithfulness, as described in Section 2.1.3. If these two assumptions hold, given an oracle of conditional independence, a constraint-based algorithm applies a selective search for the constraints among observed variables $\boldsymbol{O}$ to recover the ancestral relationships up to its Markov equivalence class using a partial ancestral graph (PAG). In this section, I provide an overview of the FCI algorithm [Spirtes et al., 2000], which is a well-known constraint-based algorithm for discovering the causal structure in the presence of latent variables. Given a dataset D on observed

variables $\boldsymbol{O}$, the FCI algorithm reconstructs a PAG $\mathcal{P}$ by applying tests of conditional independence on pairs of observed variables in two stages: adjacency and orientation.

In the adjacency stage, FCI first initializes $\mathcal{P}$ to a fully connected graph with nondirected edges ($\circ\!\!-\!\!\circ$). Then, for every adjacent pair of nodes $X_i \circ\!\!-\!\!\circ X_j$, it removes the edge if X_i and X_j are independent given some subset of nodes $\boldsymbol{Z}$ that are adjacent to them (i.e., $\boldsymbol{Z} \subseteq \boldsymbol{Adj}(X_i) \backslash X_j$ or $\boldsymbol{Z} \subseteq \boldsymbol{Adj}(X_j) \backslash X_i$) and stores $\boldsymbol{Z}$ as the set that d-separates X_i and X_j (i.e., $\boldsymbol{D\text{-}Sep}(X_i, X_j) = \boldsymbol{D\text{-}Sep}(X_j, X_i) = \boldsymbol{Z}$). Algorithm 2 shows pseudo-code of this procedure. This step will remove some but not all of the edges that should be in $\mathcal{P}$ (see Section 6.7 in [Spirtes et al., 2000] for more details). To refine $\mathcal{P}$, FCI orients each unshielded triple of variables $X_j *\!\!-\!\!\circ X_k \circ\!\!-\!\!* X_j$ as $X_j *\!\!\rightarrow X_k \leftarrow\!\!* X_j$ if and only if $X_k \notin \boldsymbol{D\text{-}Sep}(X_i, X_j)$, where "$*$" is used as a metasymbol to denote that an endpoint can be "$>$", "$-$", or "$\circ$"[1]. This is called v-structure orientation and is summarized in Algorithm 3. After orienting the colliders, graph $\mathcal{P}$ contains required information to identify subsets of variables that can "possibly" d-separate two adjacent nodes X_i and X_j, which are called $\boldsymbol{Possible\text{-}D\text{-}Sep}(X_i)$ and $\boldsymbol{Possible\text{-}D\text{-}Sep}(X_j)$. Given graph $\mathcal{P}$, $\boldsymbol{Possible\text{-}D\text{-}Sep}(X_i)$ is defined as follows:

Definition 3.2.1. ($\boldsymbol{Possible\text{-}D\text{-}Sep}$) Y is in $\boldsymbol{Possible\text{-}D\text{-}Sep}(X_i)$ if and only if there exists a path π between X_i and Y in $\mathcal{P}$ such that for every subpath X_h, X_l, X_m, either X_l is a collider on the subpath in $\mathcal{P}$ or X_h, X_l, X_m form a triangle in $\mathcal{P}$ (i.e., they are all adjacent).

For each adjacent pair of nodes X_i and X_j, FCI tests whether they are conditionally independent given any subset $\boldsymbol{Z} \subseteq \boldsymbol{Possible\text{-}D\text{-}Sep}(X_i) \backslash X_j$ or $\boldsymbol{Possible\text{-}D\text{-}Sep}(X_j) \backslash X_i$. If such a subset $\boldsymbol{Z}$ exists, FCI removes the edge between X_i and X_j and stores $\boldsymbol{D\text{-}Sep}(X_i, X_j) = \boldsymbol{D\text{-}Sep}(X_j, X_i) = \boldsymbol{Z}$; this procedure is shown in Algorithm 4.

In the orientation stage, FCI uses 10 orientation rules [Zhang, 2008] to orient the skeleton found by the adjacency stage. The overall pseudo-code for FCI is provided in Algorithm 1. Given a conditional independence oracle and the Markov and faithfulness assumptions, in the large sample limit the FCI algorithm is guaranteed to recover a PAG that contains the data-generating DAG, which may contain latent variables [Zhang, 2008].

[1] "$*$" is used for notations purposes and is not an actual endpoint in a PAG.

Algorithm 1 FCI:(D, d)

Input: a dataset D, the maximum conditioning set size d

Output: a PAG $\mathcal{P}$

$\mathcal{P}$, $\boldsymbol{D\text{-}Sep} \leftarrow$ Initial Skeleton (D, d) ▷ Algorithm 2

$\mathcal{P} \leftarrow$ V-structure Orientation ($\mathcal{P}$, $\boldsymbol{D\text{-}Sep}$) ▷ Algorithm 3

$\mathcal{P}$, $\boldsymbol{D\text{-}Sep} \leftarrow$ Final Skeleton (D, d, $\mathcal{P}$, $\boldsymbol{D\text{-}Sep}$) ▷ Algorithm 4

$\mathcal{P} \leftarrow$ V-structure Orientation ($\mathcal{P}$, $\boldsymbol{D\text{-}Sep}$) ▷ Algorithm 3

Apply orientation rules $\mathcal{R}1$-$\mathcal{R}10$ in [Zhang, 2008] to further orient the edges in $\mathcal{P}$

return $\mathcal{P}$

Algorithm 2 Initial Skeleton(D, d)

Input: a dataset D, the maximum conditioning set size d

Output: a graph $\mathcal{P}$, d-separation sets $\boldsymbol{D\text{-}Sep}$

Let $\mathcal{P}$ be a fully connected graph with nondirected edges ($\circ\!\!-\!\!\circ$)

$n = 0$

while $n \leq d$ **do**

  for all $(X_i, X_j) \in \mathcal{P}$ **do**

    if $X_j \in \boldsymbol{Adj}(X_i)$ and $|\boldsymbol{Adj}(X_i) \backslash X_j| \geq n$ **then**

      repeat

        Choose a subset $\boldsymbol{Z} \subseteq \boldsymbol{Adj}(X_i) \backslash X_j$ where $|\boldsymbol{Z}| = n$

        if $X_i \perp\!\!\!\perp X_j | \boldsymbol{Z}$ **then**

          Remove $X_i \circ\!\!-\!\!\circ X_j$ from $\mathcal{P}$

          Record $\boldsymbol{D\text{-}Sep}(X_i, X_j) = \boldsymbol{D\text{-}Sep}(X_j, X_i) = \boldsymbol{Z}$

        end if

      until $X_j \notin \boldsymbol{Adj}(X_i)$ or all $\boldsymbol{Z} \subseteq \boldsymbol{Adj}(X_i) \backslash X_j$ with $|\boldsymbol{Z}| = n$ have been tested

    end if

  end for

  $n = n + 1$

end while

return $\mathcal{P}$ and $\boldsymbol{D\text{-}Sep}$

Algorithm 3 V-structure Orientation($\mathcal{P}$, ***D-Sep***)

Input: a graph $\mathcal{P}$, d-separation set ***D-Sep***

Output: a graph $\mathcal{P}$

Form a list $\mathcal{T}$ of all unshielded triple of variables $X_i \ast\!\!-\!\!\circ X_k \circ\!\!-\!\!\ast X_j$
for all $X_i \ast\!\!-\!\!\circ X_k \circ\!\!-\!\!\ast X_j \in \mathcal{T}$ **do**
    if $X_k \notin$ ***D-Sep***(X_j, X_j) **then**
        Orient $X_i \ast\!\!-\!\!\circ X_k \circ\!\!-\!\!\ast X_j$ as $Xi \ast\!\!\rightarrow X_k \leftarrow\!\!\ast X_j$
    end if
end for
return $\mathcal{P}$

Algorithm 4 Final Skeleton(D, d, $\mathcal{P}$, ***D-Sep***)

Input: a dataset D, the maximum conditioning set size d, a graph $\mathcal{P}$, d-separation sets ***D-Sep***

Output: a graph $\mathcal{P}$, d-separation sets ***D-Sep***

for all $X_i \in \mathcal{P}$ **do**
    for all $X_j \in$ ***Adj***(X_i) **do**
        $n = 0$
        repeat
            repeat
                Choose a subset $\boldsymbol{Z} \subseteq$ ***Possible-D-Sep***$(X_i)\backslash X_j$ with $|\boldsymbol{Z}| = n$
                if $X_i \perp\!\!\!\perp X_j | \boldsymbol{Z}$ **then**
                    Remove $X_i \ast\!\!-\!\!\ast X_j$ from $\mathcal{P}$
                    Record ***D-Sep***$(X_i, X_j) =$ ***D-Sep***$(X_j, X_i) = \boldsymbol{Z}$
                end if
            until $X_j \notin$ ***Adj***(X_i) or no $\boldsymbol{Z} \subseteq$ ***Possible-D-Sep***$(X_i)\backslash X_j$ with $|\boldsymbol{Z}| = n$
            $n = n + 1$
        until $n \leq d$ and $\left[X_j \notin \textbf{\textit{Adj}}(X_i) \text{ or } |\textbf{\textit{Possible-D-Sep}}(X_i)\backslash X_j| < n \right]$
    end for
end for
Reorient all edges in $\mathcal{P}$ as $\circ\!\!-\!\!\circ$
return $\mathcal{P}$ and ***D-Sep***

3.3 Bayesian Scoring of Constraints (BSC)

This section describes how to derive the posterior probability of an independence constraint R_i from data. Let D be an i.i.d dataset that is generated from a distribution that is faithful to a ground-truth CBN structure $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V}$ is a set of domain variables with $\boldsymbol{O} \subseteq \boldsymbol{V}$ observed variables and $\boldsymbol{E}$ is a set of edges that encodes independence relationships among $\boldsymbol{V}$. Let $R_i = X \perp\!\!\!\perp Y | \boldsymbol{Z}$ be an arbitrary conditional independence constraint, which is hypothesized to hold in the data-generating model that produced dataset D, where $X, Y \in \boldsymbol{O}$ and $\boldsymbol{Z} \backslash \{X, Y\} \subseteq \boldsymbol{O}$. Each R_i is called a *conditional independence constraint*, or *constraint* for short, and it has a value of either *true* or *false*.

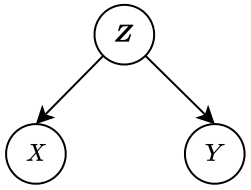
In order to score the posterior probability of a constraint R_i given dataset D, we assume that the only parts of data D that influence belief about R_i are the data D_i (i.e., data about X and Y). We call this the *data relevance assumption*, which results in:

$$P(R_i|D) = P(R_i|D_i). \tag{3.1}$$

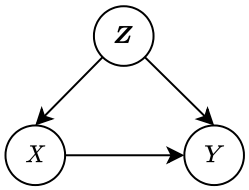
Applying Bayes' rule, the posterior probability of a constraint R_i given D_i is defined as:

$$\begin{aligned} P(R_i|D_i) &= \frac{P(R_i) \cdot P(D_i|R_i)}{P(D_i)} \\ &= \frac{P(R_i) \cdot P(D_i|R_i)}{\sum\limits_{R_i = \{true, false\}} P(R_i) \cdot P(D_i|R_i)}, \end{aligned} \tag{3.2}$$

where $P(D_i|R_i)$ is the marginal likelihood of data, $P(R_i)$ is the prior of constraint R_i being true, and R_i can be *true* or *false*.



(a) BN_{ind} corresponds to independence (i.e., $R_i = (X \perp\!\!\!\perp Y | \boldsymbol{Z}) = true$).



(b) BN_{dep} corresponds to dependence (i.e., $R_i = (X \perp\!\!\!\perp Y | \boldsymbol{Z}) = false$).

Figure 8: The independence and dependence BN structures that we use to score a constraint.

We use the BN structure BN_{ind} (Figure 8a) to compute the marginal likelihood of data $P(D_i|R_i = true)$ when we model the independence relationship. In BN_{ind}, $\boldsymbol{Z}$ is a set of parents for X and Y that renders X independent of Y conditional on $\boldsymbol{Z}$. Similarly, we use the BN structure BN_{dep} (Figure 8b) to compute the marginal likelihood of data $P(D_i|R_i = false)$ when we model the dependence relationship, which means that conditioning on $\boldsymbol{Z}$ does not render X independent of Y and there is an arc between X and Y. For CBNs that contain discrete variables, we assume there might be specific instantiations of $\boldsymbol{Z}$ that make X and Y dependent. Since the dependence relationship between X and Y holds even if it holds only for one instantiation of $\boldsymbol{Z}$, we score BN_{dep} in a special way to allow for this possibility; this method is defined in Section 3.3.1. In contrast, for CBNs that contain continuous and a mixture of discrete and continuous variables[2], we assume there are no specific values of $\boldsymbol{Z}$ that makes X and Y dependent. Therefore, the dependence relationship between X and Y must hold for all values of $\boldsymbol{Z}$; we score BN_{dep} using all values of $\boldsymbol{Z}$. These methods are introduced in Sections 3.3.2 and 3.3.3.

3.3.1 BSC for discrete variables

In this section, I describe how to compute the posterior probability of an arbitrary constraint $R_i = (X \perp\!\!\!\perp Y|\boldsymbol{Z})$ given D that contains discrete random variables using Equation (3.2). In this case, we can use the BDeu score [Heckerman et al., 1995], which provides a closed-form solution for deriving the marginal likelihood for $P(D_i|R_i)$. More specifically, to compute $P(D_i|R_i = true)$, we derive the BDeu score using BN_{ind} (Figure 8a). For $R_i = (X \perp\!\!\!\perp Y|\boldsymbol{Z}) = true$, the independence relation should hold for all possible instantiations of $\boldsymbol{Z}$, denoted as $\boldsymbol{Z} = k$. Assuming parameter independence and parameter modularity [Heckerman et al., 1995], $P(D_i|R_i = true)$ can be computed as follows:

$$P(D_i|R_i = true) = \prod_{k=1}^{q} P(D_i|R_i = true, \boldsymbol{Z} = k), \tag{3.3}$$

where q denotes all possible instantiations of variables in $\boldsymbol{Z}$. Similarly, we can compute the overall likelihood of D_i per each instantiation $\boldsymbol{Z} = k$, assuming D_i is modeled either by

[2]We transform mixed variables to all continuous variables using the degenerate Gaussian method introduced in [Andrews et al., 2019].

BN_{ind} or BN_{dep} (Figure 8), as follows:

$$\begin{aligned} P(D_i) &= \prod_{k=1}^{q} \sum_{R_i=\{true,false\}} P(R_i|\boldsymbol{Z}=k) \cdot P(D_i|R_i,\boldsymbol{Z}=k) \\ &= \prod_{k=1}^{q} \Big[P(R_i=true|\boldsymbol{Z}=k) \cdot P(D_i|R_i=true,\boldsymbol{Z}=k) \\ &\qquad + P(R_i=false|\boldsymbol{Z}=k) \cdot P(D_i|R_i=false,\boldsymbol{Z}=k) \Big], \end{aligned} \tag{3.4}$$

where q denotes all possible instantiations of $\boldsymbol{Z}$, $P(D_i|R_i = true, \boldsymbol{Z} = k)$ denotes the marginal likelihood of D_i when $\boldsymbol{Z} = k$ and using BN_{ind} (Figure 8a), and $P(D_i|R_i = false, \boldsymbol{Z} = k)$ denotes the the marginal likelihood of D_i when $\boldsymbol{Z} = k$ and using BN_{dep} (Figure 8b). Consider the sum of products that results from expanding the product of sums in Equation (3.4). In that expansion, there is only one product term that corresponds to the independence relation (i.e., $P(D_i|R_i = true)$ given in Equation (3.3)); it is the term in which independence holds for all instantiations of $\boldsymbol{Z}$; the rest of the product terms correspond to dependence. We formulate the dependence relationship this way since X and Y will become dependent even if the dependence holds for only one instantiation of $\boldsymbol{Z}$. The terms $P(R_i = true|\boldsymbol{Z} = k)$ and $P(R_i = false|\boldsymbol{Z} = k)$ are structure priors per each $\boldsymbol{Z} = k$, which are defined as follows:

$$\begin{aligned} P(R_i = true|\boldsymbol{Z}=k) &= \sqrt[q]{P(R_i = true)} \text{ and} \\ P(R_i = false|\boldsymbol{Z}=k) &= 1.0 - P(R_i = true|\boldsymbol{Z}=k), \end{aligned} \tag{3.5}$$

where q is the number of possible instantiations of $\boldsymbol{Z}$. If we assume independence and dependence are a priori equally likely, then $P(R_i = true) = P(R_i = false) = 0.5$.

By applying Equations (3.3)-(3.5) to Equation (3.2), the posterior probability of a constraint $R_i = (X \perp\!\!\!\perp Y|\boldsymbol{Z}) = true$ can be re-written as follows:

$$P(R_i = true|D_i) = \frac{\prod_{k=1}^{q} P(R_i = true|\boldsymbol{Z}=k) \cdot P(D_i|R_i = true,\boldsymbol{Z}=k)}{\prod_{k=1}^{q} \sum_{R_i=\{true,false\}} P(R_i|\boldsymbol{Z}=k) \cdot P(D_i|R_i,\boldsymbol{Z}=k)}. \tag{3.6}$$

Finally, the posterior probability of a constraint $R_i = (X \perp\!\!\!\perp Y|\boldsymbol{Z}) = false$ is as follows:

$$P(R_i = false|D_i) = 1.0 - P(R_i = true|D_i). \tag{3.7}$$

3.3.1.1 Proof of correctness for BSC-discrete

In this section, I first provide a lemma that will then be used to prove Theorem 3.3.1, which shows the correctness of BSC when using the BD score [Heckerman et al., 1995] for discrete variable types. The proof of Theorem 3.3.1 is influenced by the proof of Theorem 6.3 in Section 6.3 of [Herskovits, 1991]. Theorem 3.3.1 generalizes that theorem from using K2 priors to using BD priors in developing the BSC independence test.

Lemma 3.3.1. Let P be the full joint probability distribution over a set of random variables $\boldsymbol{V}$, and $X, Y \in \boldsymbol{V}$ be two variables and $\boldsymbol{Z} \setminus \{X, Y\} \subset \boldsymbol{V}$ be a set of random variables that excludes X and Y. Also, let $H_j(Y|\boldsymbol{Z} = j)$ denote the conditional entropy of Y given $\boldsymbol{Z} = j$, where j denotes a particular instantiation of the variables in $\boldsymbol{Z}$. Similarly, $H_j(Y|X, \boldsymbol{Z} = j)$ denote the conditional entropy of Y given X and $\boldsymbol{Z} = j$, which are defined as follows [Cover, 1999] (page 17):

$$\begin{aligned} H_j(Y|\boldsymbol{Z} = j) &= -\sum_y P(y|\boldsymbol{Z} = j) \cdot \log P(y|\boldsymbol{Z} = j) \\ H_j(Y|X, \boldsymbol{Z} = j) &= -\sum_y \sum_x P(y, x|\boldsymbol{Z} = j) \cdot \log P(y|x, \boldsymbol{Z} = j)\,, \end{aligned} \tag{3.8}$$

where x and y iterate over all possible instantiations of X and Y, respectively. Then, $H_j(Y|\boldsymbol{Z} = j) \geq H_j(Y|X, \boldsymbol{Z} = j)$ and the equality holds if and only if $X \perp\!\!\!\perp Y|\boldsymbol{Z} = j$ holds.

Proof. Applying the chain rule of entropy, the conditional mutual information can be computed as follows [Cover, 1999]:

$$I(X; Y|\boldsymbol{Z} = j) = H(Y|\boldsymbol{Z} = j) - H(Y|X, \boldsymbol{Z} = j)\,. \tag{3.9}$$

Given that the mutual information is nonnegative (i.e., $I(X; Y|\boldsymbol{Z} = j) \geq 0$) and $I(X; Y|\boldsymbol{Z} = j) = 0$ if and only if $X \perp\!\!\!\perp Y|\boldsymbol{Z} = j$ [Cover, 1999] (page 29), it follows that:

$$\begin{aligned} H(Y|\boldsymbol{Z} = j) - H(Y|X, \boldsymbol{Z} = j) &\geq 0 \\ H(Y|\boldsymbol{Z} = j) &\geq H(Y|X, \boldsymbol{Z} = j)\,, \end{aligned} \tag{3.10}$$

where the equality holds if and only if $X \perp\!\!\!\perp Y|\boldsymbol{Z} = j$. □

Theorem 3.3.1. Let D be a dataset that contains N cases with no missing values on a set of discrete variables $\boldsymbol{V}$ that is sampled from distribution P, which is strictly positive as $N \to \infty$. Let $X, Y \in \boldsymbol{V}$ be two variables, $\boldsymbol{Z} \setminus \{X, Y\} \subset \boldsymbol{V}$ be a set of random variables that excludes X and Y. Also, let $(X \perp\!\!\!\perp Y | \boldsymbol{Z} = j)$ be the independence constraint that we want to score for a particular instantiation of $\boldsymbol{Z} = j$. Using BN_{ind} shown in Figure 8a to score independence, BN_{dep} shown in Figure 8b to score dependence, and using BD score [Heckerman et al., 1995], BSC assigns the correct constraint hypothesis a probability that approaches 1.0 in the large sample limit:

$$\lim_{N\to\infty} \frac{P(D_Y|\boldsymbol{Z}=j)}{P(D_Y|X,\boldsymbol{Z}=j)} = \begin{cases} \infty & \text{if and only if } (X \perp\!\!\!\perp Y|\boldsymbol{Z}=j) = true \\ 0 & \text{otherwise} \end{cases}, \tag{3.11}$$

which indicates that BSC is correct for a particular instantiation of $\boldsymbol{Z}$ using the BD score.

Proof. The BD score for $P(D_Y|\boldsymbol{Z}=j)$ is calculated as follows [Heckerman et al., 1995]:

$$P(D_Y|\boldsymbol{Z}=j) = \frac{\Gamma(\alpha_j)}{\Gamma(\alpha_j + N_j)} \cdot \prod_{k=1}^{r} \frac{\Gamma(\alpha_{jk} + N_{jk})}{\Gamma(\alpha_{jk})}, \tag{3.12}$$

where j denotes instantiations of variables in $\boldsymbol{Z}$ and the product is over all r values of variable Y. The term N_{jk} is the number of cases in data in which variable $Y = k$ and its parent $\boldsymbol{Z} = j$; also, $N_j = \sum_{k=1}^{r} N_{jk}$. The term α_{jk} is a finite positive real number that is called Dirichlet prior parameter and may be interpreted as representing "pseudo-counts", where $\alpha_j = \sum_{k=1}^{r} \alpha_{jk}$. BD can be re-written in *log* form as follows:

$$\log P(D_Y|\boldsymbol{Z}=j) = \log\Gamma(\alpha_j) - \log\Gamma(\alpha_j + N_j) + \sum_{k=1}^{r} (\log\Gamma(\alpha_{jk} + N_{jk}) - \log\Gamma(\alpha_{jk})). \tag{3.13}$$

We can re-arrange the terms in Equation (3.13) to omit the constant terms as follows:

$$\begin{aligned} \log P(D_Y|\boldsymbol{Z}=j) &= -\log\Gamma(\alpha_j + N_j) + \sum_{k=1}^{r} \log\Gamma(\alpha_{jk} + N_{jk}) + \log\Gamma(\alpha_j) - \sum_{k=1}^{r} \log\Gamma(\alpha_{jk}) \\ &= -\log\Gamma(\alpha_j + N_j) + \sum_{k=1}^{r} \log\Gamma(\alpha_{jk} + N_{jk}) + const. \end{aligned} \tag{3.14}$$

Using the Stirling's approximation of $\log\Gamma(n)$, which is defined as follows:

$$\lim_{n\to\infty} \log\Gamma(n) = (n - \frac{1}{2})\log(n) - n + const., \tag{3.15}$$

we can re-write Equation (3.14) as follows:

$$\begin{aligned}
&\lim_{N\to\infty} \log P(D_Y|\boldsymbol{Z} = j) = \lim_{N\to\infty} -(\alpha_j + N_j - \frac{1}{2})\log(\alpha_j + N_j) + (\alpha_j + N_j) \\
&+ \sum_{k=1}^{r} \Big((\alpha_{jk} + N_{jk} - \frac{1}{2})\log(\alpha_{jk} + N_{jk}) - (\alpha_{jk} + N_{jk}) \Big) + const. \\
&= \lim_{N\to\infty} -N_j \log(\alpha_j + N_j) - \alpha_j \log(\alpha_j + N_j) + \frac{1}{2}\log(\alpha_j + N_j) + \alpha_j + N_j \\
&+ \sum_{k=1}^{r} \Big(N_{jk}\log(\alpha_{jk} + N_{jk}) + \alpha_{jk}\log(\alpha_{jk} + N_{jk}) - \frac{1}{2}\log(\alpha_{jk} + N_{jk}) - \alpha_{jk} - N_{jk} \Big) + const. \\
&= \lim_{N\to\infty} -N_j \log(\alpha_j + N_j) + \sum_{k=1}^{r} N_{jk}\log(\alpha_{jk} + N_{jk}) \\
&- \alpha_j \log(\alpha_j + N_j) + \sum_{k=1}^{r} \alpha_{jk}\log(\alpha_{jk} + N_{jk}) \\
&+ \frac{1}{2}\log(\alpha_j + N_j) - \frac{1}{2}\sum_{k=1}^{r} \log(\alpha_{jk} + N_{jk}) \\
&+ \alpha_j + N_j - \sum_{k=1}^{r} (\alpha_{jk} + N_{jk}) + const.
\end{aligned} \tag{3.16}$$

Since $\sum_{k=1}^{r} N_{jk} = N_j$ and $\sum_{k=1}^{r} \alpha_{jk} = \alpha_j$, Equation (3.16) can be re-written as follows:

$$\begin{aligned}
\lim_{N\to\infty} \log P(D_Y|\boldsymbol{Z} = j) = \lim_{N\to\infty} &\sum_{k=1}^{r} \left[N_{jk}\log(\frac{\alpha_{jk} + N_{jk}}{\alpha_j + N_j}) + \alpha_{jk}\log(\frac{\alpha_{jk} + N_{jk}}{\alpha_j + N_j}) \right] \\
&+ \frac{1}{2}\left[\log(\alpha_j + N_j) - \sum_{k=1}^{r} \log(\alpha_{jk} + N_{jk}) \right] + const.,
\end{aligned} \tag{3.17}$$

Given that

$$\lim_{N\to\infty} \frac{\alpha_{jk} + N_{jk}}{\alpha_j + N_j} = \frac{N_{jk}}{N_j}$$

and

$$\lim_{N\to\infty} \sum_{k=1}^{r} \alpha_{jk}\log(\frac{\alpha_{jk} + N_{jk}}{\alpha_j + N_j}) = const.,$$

in the limit, Equation (3.17) becomes:

$$\begin{aligned}
&\lim_{N\to\infty} \log P(D_Y|\boldsymbol{Z}=j) = \\
&\lim_{N\to\infty} \sum_{k=1}^{r} N_{jk} \log \frac{N_{jk}}{N_j} + \frac{1}{2}\left[\log(\alpha_j + N_j) - \sum_{k=1}^{r} \log(\alpha_{jk} + N_{jk})\right] + const.\,,
\end{aligned} \tag{3.18}$$

or equivalently:

$$\begin{aligned}
&\lim_{N\to\infty} \log P(D_Y|\boldsymbol{Z}=j) = \\
&\lim_{N\to\infty} N \cdot \sum_{k=1}^{r} \frac{N_{jk}}{N} \log \frac{N_{jk}}{N_j} + \frac{1}{2}\left[\log(\alpha_j + N_j) - \sum_{k=1}^{r} \log(\alpha_{jk} + N_{jk})\right] \\
&= \lim_{N\to\infty} N \cdot \sum_{k=1}^{r} P(Y=k, \boldsymbol{Z}=j) \log P(Y=k|\boldsymbol{Z}=j) \\
&+ \frac{1}{2}\left[\log(\alpha_j + N_j) - \sum_{k=1}^{r} \log(\alpha_{jk} + N_{jk})\right] \\
&= \lim_{N\to\infty} -N \cdot H_j(Y|\boldsymbol{Z}=j) + \frac{1}{2}\left[\log(\alpha_j + N_j) - \sum_{k=1}^{r} \log(\alpha_{jk} + N_{jk})\right].
\end{aligned} \tag{3.19}$$

To simplify the second term in this equation, we divide the log terms by N and equivalently add $\log N$ terms as follows:

$$\begin{aligned}
\lim_{N\to\infty} \log P(D_Y|\boldsymbol{Z}=j) &= \lim_{N\to\infty} -N \cdot H_j(Y|\boldsymbol{Z}=j) \\
&+ \frac{1}{2}\left[\log(\frac{\alpha_j + N_j}{N}) + \log N - \sum_{k=1}^{r}\left[\log(\frac{\alpha_j + N_j}{N}) + \log N\right]\right] \\
&= \lim_{N\to\infty} -N \cdot H_j(Y|\boldsymbol{Z}=j) \\
&+ \frac{1}{2}\left[\log N - \sum_{k=1}^{r} \log N\right] + \frac{1}{2}\left[\log(\frac{\alpha_j + N_j}{N}) - \sum_{k=1}^{r} \log(\frac{\alpha_j + N_j}{N})\right] \\
&= \lim_{N\to\infty} -N \cdot H_j(Y|\boldsymbol{Z}=j) - \frac{(r-1)}{2} \log N + const.\,,
\end{aligned} \tag{3.20}$$

Similarly, we can derive $\lim_{N\to\infty} \log P(D_Y|X, \boldsymbol{Z}=j)$ as follows:

$$\lim_{N\to\infty} \log P(D_Y|X, \boldsymbol{Z}=j) = \lim_{N\to\infty} -N \cdot H_j(Y|X, \boldsymbol{Z}=j) - \frac{q' \cdot (r-1)}{2} \log N + const.\,, \tag{3.21}$$

where q' denotes all possible instantiations of X. Finally, using Equations (3.20) and (3.21), we have:

$$\begin{aligned}\lim_{N\to\infty} \log \frac{P(D_Y|\boldsymbol{Z}=j)}{P(D_Y|X,\boldsymbol{Z}=j)} &= \lim_{N\to\infty} N\cdot[H_j(Y|X,\boldsymbol{Z}=j) - H_j(Y|\boldsymbol{Z}=j)] \\ &\quad + \frac{q'\cdot(r-1)}{2}\log N - \frac{(r-1)}{2}\log N.\end{aligned} \tag{3.22}$$

According to Lemma 3.3.1, $H_j(Y|\boldsymbol{Z}=j) \geq H_j(Y|X,\boldsymbol{Z}=j)$. There are two possible cases:

Case 1: $X \perp\!\!\!\perp Y|\boldsymbol{Z}=j$ **is true.** In this case, $H_j(Y|X,\boldsymbol{Z}=j) = H_j(Y|\boldsymbol{Z}=j)$ according to Lemma 3.3.1, which results in:

$$\begin{aligned}\lim_{N\to\infty} \log \frac{P(D_Y|\boldsymbol{Z}=j)}{P(D_Y|X,\boldsymbol{Z}=j)} &= \lim_{N\to\infty} \frac{q'\cdot(r-1)}{2}\log N - \frac{(r-1)}{2}\log N \\ &= \lim_{N\to\infty} \frac{(q'-1)\cdot(r-1)}{2}\log N,\end{aligned} \tag{3.23}$$

or equivalently:

$$\lim_{N\to\infty} \frac{P(D_Y|\boldsymbol{Z}=j)}{P(D_Y|X,\boldsymbol{Z}=j)} = \lim_{N\to\infty} N^{\frac{(q'-1)\cdot(r-1)}{2}}. \tag{3.24}$$

Given that $q' > 1$ and $r > 1$, the term $\frac{(q'-1)\cdot(r-1)}{2}$ in Equation (3.24) becomes positive; therefore, Equation (3.24) becomes ∞ as $N \to \infty$.

Case 2: $X \perp\!\!\!\perp Y|\boldsymbol{Z}=j$ **is false.** In this case, we have:

$$\begin{aligned}\lim_{N\to\infty} \log \frac{P(D_Y|\boldsymbol{Z}=j)}{P(D_Y|X,\boldsymbol{Z}=j)} &= \lim_{N\to\infty} N\cdot[H_j(Y|X,\boldsymbol{Z}=j) - H_j(Y|\boldsymbol{Z}=j)] \\ &\quad + \frac{q'\cdot(r-1)}{2}\log N - \frac{(r-1)}{2}\log N,\end{aligned} \tag{3.25}$$

where the first term is of $O(N)$ and dominates the second and third terms, which are of $O(\log N)$. Since $H_j(Y|\boldsymbol{Z}=j) > H_j(Y|X,\boldsymbol{Z}=j)$ according to Lemma 3.3.1, the first term (i.e., $H_j(Y|X,\boldsymbol{Z}=j) - H_j(Y|\boldsymbol{Z}=j)$) in Equation (3.25) becomes a negative number and dominates the second and the third terms. As a result, Equation (3.25) becomes $-\infty$, which is implies:

$$\lim_{N\to\infty} \frac{P(D_Y|\boldsymbol{Z}=j)}{P(D_Y|X,\boldsymbol{Z}=j)} = 0. \tag{3.26}$$

□

Proposition 3.3.1. Let D be a dataset that contains N cases with no missing values on a set of discrete variables $\boldsymbol{V}$ that is sampled from distribution P, which is strictly positive as $N \to \infty$. Let $X, Y \in \boldsymbol{V}$ be two variables, $\boldsymbol{Z} \setminus \{X, Y\} \subset \boldsymbol{V}$ be a set of random variables that excludes X and Y. Also, let $(X \perp\!\!\!\perp Y | \boldsymbol{Z})$ be the independence constraint that we want to score. Using the BD score [Heckerman et al., 1995], the BSC method assigns the correct constraint hypothesis a probability that approaches 1.0 in the large sample limit:

$$\lim_{N\to\infty} \frac{P(D_Y|\boldsymbol{Z})}{P(D_Y|X,\boldsymbol{Z})} = \lim_{N\to\infty} \frac{\prod_{j=1}^{q} P(D_Y|\boldsymbol{Z}=j)}{\left[\prod_{j=1}^{q} P(D_Y|\boldsymbol{Z}=j) + P(D_Y|X,\boldsymbol{Z}=j)\right] - \prod_{j=1}^{q} P(D_Y|\boldsymbol{Z}=j)}$$
$$= \begin{cases} \infty & \text{if and only if } (X \perp\!\!\!\perp Y|\boldsymbol{Z}) = true \\ 0 & \text{otherwise} \end{cases}, \tag{3.27}$$

which indicates that the BSC method is correct using the BD score.

Proof. If $(X \perp\!\!\!\perp Y|\boldsymbol{Z}) = true$, then the independence relation holds for all instantiations of Z (i.e., the product term in the numerator). Accordingly, by Theorem 3.3.1, the numerator will dominate the terms in the denominator and the fraction approaches ∞. Additionally, if $(X \perp\!\!\!\perp Y|\boldsymbol{Z}) = false$, then the dependence relationship holds at least for one of the terms in the denominator, which equivalently implies that at least one of the $P(D_Y|\boldsymbol{Z}=j)$ terms does not hold. Therefore, by Theorem 3.3.1, at least one of the terms in the denominator will dominate the numerator, and the fraction becomes 0. □

Corollary 3.3.1. Let D be a dataset that contains N cases with no missing values on a set of discrete variables $\boldsymbol{V}$ that is sampled from distribution P, which is strictly positive as $N \to \infty$. Let $X, Y \in \boldsymbol{V}$ be two variables, $\boldsymbol{Z} \setminus \{X, Y\} \subset \boldsymbol{V}$ be a set of random variables that excludes X and Y. Also, let $(X \perp\!\!\!\perp Y | \boldsymbol{Z})$ be the independence constraint that we want to score. Using the K2 score [Cooper and Herskovits, 1992], the BSC method assigns the correct constraint hypothesis a probability that approaches 1.0 in the large sample limit, which indicates that the BSC method is correct using the K2 score.

3.3.2 BSC for continuous variables

In this section, I explain how to formulate the posterior probability of a constraint $R_i = (X \perp\!\!\!\perp Y|\boldsymbol{Z})$ given a dataset D that contains random variables with Gaussian distributions. In this case, we can use the BIC score [Schwarz, 1978] for approximating the marginal likelihood for $P(D_i|R_i)$ as follows:

$$\begin{aligned} P(D_i|R_i = true) &= \ell(\hat{\Theta}_{ind}) - \frac{df_{ind}}{2}\log N \\ P(D_i|R_i = false) &= \ell(\hat{\Theta}_{dep}) - \frac{df_{dep}}{2}\log N\,, \end{aligned} \tag{3.28}$$

where N denotes the number of cases in D_i, $\ell(\hat{\Theta}_{ind})$ and $\ell(\hat{\Theta}_{dep})$ are maximum log-likelihood of the data using the independence and dependence BN structures shown in Figures 8a and 8b, respectively. Note that in this case, we score BN_{dep} using all values of $\boldsymbol{Z}$ since we assume the dependence relationship holds over all values of the continuous variables $\boldsymbol{Z}$. The terms df_{ind} and df_{dep} are degrees of freedom in those BN models, respectively. Since the BIC score is decomposable at the node level for each node X given its parents $\boldsymbol{Pa}(X)$, the log-likelihood term can be computed for each parent-child relationship using maximum likelihood estimates of the parameters, and $df_{X|\boldsymbol{Pa}(X)} = |\boldsymbol{Pa}(X)|+1$ is the degrees of freedom. Finally, we can apply Equations (3.28) to Equation (3.2) to obtain $P(R_i|D_i)$.

3.3.2.1 Proof of correctness for BSC-continuous In this section, I provide two lemmas that will be used to prove a theorem of asymptotic correctness (consistency) of BSC when using the BIC score for continuous variable type.

Lemma 3.3.2. Given a set of continuous random variables $\boldsymbol{V} = \{X_1, X_2, ..., X_n\}$ that follow a multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, the entropy is defined as follows [Cover, 1999] (page 249):

$$H(\boldsymbol{X}) = \frac{1}{2}\log(2\pi e)^n|\boldsymbol{\Sigma}| = \frac{1}{2}\log|2\pi e\boldsymbol{\Sigma}| \tag{3.29}$$

where $|\boldsymbol{\Sigma}|$ denotes the determinant of $\boldsymbol{\Sigma}$. Both terms are equivalent due to the fact that $|c\boldsymbol{A}| = c^n|\boldsymbol{A}|$ for a $n \times n$ matrix.

Proof. See the proof of Theorem 8.4.1 in [Cover, 1999] for the derivation of this equation. □

Lemma 3.3.3. Let f be a positive continuous joint probability density function over a set of continuous random variables $\boldsymbol{V}$, and $X, Y \in \boldsymbol{V}$ be two random variables and $\boldsymbol{Z} \setminus \{X, Y\} \subset \boldsymbol{V}$ be a set of random variables. Also, let $H(Y|\boldsymbol{Z})$ denote the conditional entropy of Y given $\boldsymbol{Z}$, and similarly, $H(Y|X, \boldsymbol{Z})$ denote the conditional entropy of Y given X and $\boldsymbol{Z}$, which are defined as follows:

$$\begin{aligned} H(Y|\boldsymbol{Z}) &= -\int_y \int_{\boldsymbol{z}} f(y, \boldsymbol{z}) \cdot \log f(y|\boldsymbol{z}) \; dyd\boldsymbol{z} \\ H(Y|X, \boldsymbol{Z}) &= -\int_y \int_x \int_{\boldsymbol{z}} f(y, x, z) \cdot \log f(y|x, \boldsymbol{z}) \; dydxd\boldsymbol{z} \,, \end{aligned} \tag{3.30}$$

Then, $H(Y|\boldsymbol{Z}) \geq H(Y|X, \boldsymbol{Z})$ and the equality holds if and only if $X \perp\!\!\!\perp Y|\boldsymbol{Z}$ holds.

Proof. Applying the chain rule of entropy, the conditional mutual information can be computed as follows [Cover, 1999]:

$$I(X; Y|\boldsymbol{Z}) = H(Y|\boldsymbol{Z}) - H(Y|X, \boldsymbol{Z}) \,. \tag{3.31}$$

Given that the mutual information is nonnegative (i.e., $I(X; Y|\boldsymbol{Z}) \geq 0$) and the equality holds if and only if $X \perp\!\!\!\perp Y|\boldsymbol{Z}$ [Cover, 1999] (page 253), it follows that:

$$\begin{aligned} H(Y|\boldsymbol{Z}) - H(Y|X, \boldsymbol{Z}) &\geq 0 \\ H(Y|\boldsymbol{Z}) &\geq H(Y|X, \boldsymbol{Z}) \,, \end{aligned} \tag{3.32}$$

where the equality holds if and only if $X \perp\!\!\!\perp Y|\boldsymbol{Z}$. □

Theorem 3.3.2. Let D be a dataset that contains N cases with no missing values on a set of continuous variables $\boldsymbol{V}$ that is sampled from a multivariate Gaussian distribution P. Let $X, Y \in \boldsymbol{V}$ be two variables, $\boldsymbol{Z} \setminus \{X, Y\} \subset \boldsymbol{V}$ be a set of random variables, and also $(X \perp\!\!\!\perp Y|\boldsymbol{Z})$ be the independence constraint that we want to score. Using BN_{ind} shown in Figure 8a to score independence, BN_{dep} shown in Figure 8b to score dependence, and using the BIC score [Schwarz, 1978], then the correct constraint hypothesis is assigned a probability that approaches 1.0 in the large sample limit:

$$\lim_{N \to \infty} \frac{P(D_Y|\boldsymbol{Z})}{P(D_Y|X, \boldsymbol{Z})} = \begin{cases} \infty & \text{if and only if } (X \perp\!\!\!\perp Y|\boldsymbol{Z}) = true \\ 0 & \text{otherwise} \end{cases} \,, \tag{3.33}$$

which indicates that the Bayesian scoring of independence constraint (BSC) using the BIC score is correct.

Proof. The BIC score for a Bayesian network $\mathcal{G}$ given dataset D is decomposable at the node level for each node $X_i \in \boldsymbol{V}$ given its parents $\boldsymbol{Pa}(X_i)$:

$$S(\mathcal{G}, D) = \sum_{i=1}^{n} s(X_i, \boldsymbol{Pa}(X_i)) \,, \tag{3.34}$$

where $s(X_i, \boldsymbol{Pa}(X_i))$ is defined as follows, according to the BIC score:

$$s(X_i, \boldsymbol{Pa}(X_i)) = \ell(X_i|\boldsymbol{Pa}(X_i); \hat{\Theta}) - \frac{1}{2} df_{X_i|\boldsymbol{Pa}(X_i)} \cdot \log N, \tag{3.35}$$

where $\ell(X_i|\boldsymbol{Pa}(X_i); \hat{\Theta})$ denotes the conditional log-likelihood of data for the given parent-child relationship using maximum likelihood estimate of the parameters and $df_{X_i|\boldsymbol{Pa}(X_i)} = |\boldsymbol{Pa}(X_i)| + 1$ is the degrees of freedom. Given that the conditional likelihood of variable X_i given its parents $\boldsymbol{Pa}(X_i)$ is defined as follows:

$$P(X_i|\boldsymbol{Pa}(X_i)) = \frac{P(X_i, \boldsymbol{Pa}(X_i))}{P(\boldsymbol{Pa}(X_i))}, \tag{3.36}$$

the conditional log-likelihood $\ell(X_i|\boldsymbol{Pa}(X_i); \hat{\Theta})$ becomes:

$$\ell(X_i|\boldsymbol{Pa}(X_i); \hat{\Theta}) = \ell(X_i, \boldsymbol{Pa}(X_i); \hat{\Theta}) - \ell(\boldsymbol{Pa}(X_i); \hat{\Theta}). \tag{3.37}$$

Assuming a multivariate Gaussian distribution, the likelihood for variables $\boldsymbol{V}$ is defined as follows [Bishop, 2006] (page 78):

$$\mathcal{L}(\boldsymbol{V}; \Theta) = (2\pi)^{-\frac{N \cdot n}{2}} |\boldsymbol{\Sigma}|^{-\frac{N}{2}} \exp\left(-\frac{1}{2} \sum_{i=1}^{N} (\boldsymbol{v}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{v}_i - \boldsymbol{\mu}) \right) , \tag{3.38}$$

where N denotes the number of observations in dataset D, n is the number of variables in $\boldsymbol{V}$, $\boldsymbol{v}_i$ denotes the values of ith observation in D, $\boldsymbol{\mu}$ is the vector valued mean, and $\boldsymbol{\Sigma}$ is the covariance matrix with $|\boldsymbol{\Sigma}|$ and $\boldsymbol{\Sigma}^{-1}$ denoting its determinant and inverse, respectively. We then take the log of Equation (3.38) to obtain the Gaussian log-likelihood as follows:

$$\ell(\boldsymbol{V}; \Theta) = -\frac{N \cdot n}{2} \log 2\pi - \frac{N}{2} \log |\boldsymbol{\Sigma}| - \frac{1}{2} \sum_{i=1}^{N} (\boldsymbol{v}_i - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{v}_i - \boldsymbol{\mu}) \cdot \log e. \tag{3.39}$$

Given the maximum likelihood estimates of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ [Bishop, 2006] (pages 93 and 94):

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{v}_i$$
$$\hat{\boldsymbol{\Sigma}} = \frac{1}{N}\sum_{i=1}^{N} (\boldsymbol{v}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{v}_i - \hat{\boldsymbol{\mu}})^T,$$

the log-likelihood in Equation (3.39) simplifies to:

$$\ell(\boldsymbol{V}; \hat{\Theta}) = -\frac{N \cdot n}{2}\log 2\pi - \frac{N}{2}\log|\hat{\boldsymbol{\Sigma}}| - \frac{1}{2}\sum_{i=1}^{N}(\boldsymbol{v}_i - \hat{\boldsymbol{\mu}})^T \hat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{v}_i - \hat{\boldsymbol{\mu}}) \cdot \log e. \tag{3.40}$$

Since $A^T BA = tr(A^T BA)$, where tr denotes the trace of a square matrix and is defined as the sum of the diagonal elements of the matrix, Equation (3.40) can be re-written as follows:

$$\ell(\boldsymbol{V}; \hat{\Theta}) = -\frac{N \cdot n}{2}\log 2\pi - \frac{N}{2}\log|\hat{\boldsymbol{\Sigma}}| - \frac{1}{2}\sum_{i=1}^{N} tr((\boldsymbol{v}_i - \hat{\boldsymbol{\mu}})^T \hat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{v}_i - \hat{\boldsymbol{\mu}})) \cdot \log e. \tag{3.41}$$

Also, since trace is a linear operation that is invariant under cyclic permutations of matrix products (i.e., $tr(A^T BA) = tr(AA^T B) = tr(BAA^T)$), Equation (3.41) becomes:

$$\begin{aligned}
\ell(\boldsymbol{V}; \hat{\Theta}) &= -\frac{N \cdot n}{2}\log 2\pi - \frac{N}{2}\log|\hat{\boldsymbol{\Sigma}}| - \frac{1}{2} tr(\sum_{i=1}^{N}(\boldsymbol{v}_i - \hat{\boldsymbol{\mu}})(\boldsymbol{v}_i - \hat{\boldsymbol{\mu}})^T \hat{\boldsymbol{\Sigma}}^{-1}) \cdot \log e \\
&= -\frac{N \cdot n}{2}\log 2\pi - \frac{N}{2}\log|\hat{\boldsymbol{\Sigma}}| - \frac{N}{2} tr(\hat{\boldsymbol{\Sigma}}^{-1}\hat{\boldsymbol{\Sigma}}) \cdot \log e \\
&= -\frac{N \cdot n}{2}\log 2\pi - \frac{N}{2}\log|\hat{\boldsymbol{\Sigma}}| - \frac{N}{2} tr(\mathbb{I}) \cdot \log e \\
&= -\frac{N \cdot n}{2}\log 2\pi - \frac{N}{2}\log|\hat{\boldsymbol{\Sigma}}| - \frac{N \cdot n}{2} \cdot \log e \\
&= -\frac{N \cdot n}{2}(\log 2\pi + \log e) - \frac{N}{2}\log|\hat{\boldsymbol{\Sigma}}| \\
&= -\frac{N \cdot n}{2}(\log 2\pi e) - \frac{N}{2}\log|\hat{\boldsymbol{\Sigma}}| \\
&= -\frac{N}{2}\left(\log(2\pi e)^n + \log|\hat{\boldsymbol{\Sigma}}|\right) \\
&= -\frac{N}{2}\left(\log(2\pi e)^n |\hat{\boldsymbol{\Sigma}}|\right) \\
&= -\frac{N}{2}\log|2\pi e\hat{\boldsymbol{\Sigma}}|,
\end{aligned} \tag{3.42}$$

where $\mathbb{I}$ in the third line is the identity matrix with n dimensions and $tr(\mathbb{I}) = n$. The last two lines are equivalent due to the fact that $|c\boldsymbol{A}| = c^n|\boldsymbol{A}|$ for a $n \times n$ matrix. We can use Equation (3.42) to compute the marginal log-likelihood for any subset of variables $\boldsymbol{U} \subseteq \boldsymbol{V}$ by using the sub-matrix $\hat{\boldsymbol{\Sigma}}_{\boldsymbol{U}}$ of the covariance matrix that is restricted to the variables in $\boldsymbol{U}$ [Bishop, 2006] (page 89). For example, if $\boldsymbol{U} = \{X_1, X_2\}$, then we can use the following covariance matrix to compute the marginal log-likelihood:

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{U}} = \begin{bmatrix} \hat{\boldsymbol{\Sigma}}_{X_1X_1} & \hat{\boldsymbol{\Sigma}}_{X_1X_2} \\ \hat{\boldsymbol{\Sigma}}_{X_2X_1} & \hat{\boldsymbol{\Sigma}}_{X_2X_2} \end{bmatrix}$$

Therefore, we can apply Equation (3.42) to obtain:

$$\begin{aligned} &\lim_{N\to\infty} \log \frac{P(D_Y|\boldsymbol{Z})}{P(D_Y|X,\boldsymbol{Z})} = \\ &\lim_{N\to\infty} -\frac{N}{2}\log|2\pi e\hat{\boldsymbol{\Sigma}}_{Y,\boldsymbol{Z}}| + \frac{N}{2}\log|2\pi e\hat{\boldsymbol{\Sigma}}_{\boldsymbol{Z}}| - \frac{1}{2}(|\boldsymbol{Z}|+1)\cdot\log N \\ &\quad + \frac{N}{2}\log|2\pi e\hat{\boldsymbol{\Sigma}}_{Y,X,\boldsymbol{Z}}| - \frac{N}{2}\log|2\pi e\hat{\boldsymbol{\Sigma}}_{X,\boldsymbol{Z}}| + \frac{1}{2}(|X|+|\boldsymbol{Z}|+1)\cdot\log N. \end{aligned} \tag{3.43}$$

Given that the conditional entropy of a variable A given another variable B is calculated as $H(A|B) = H(A,B) - H(B)$, and according to Lemma (3.3.2) we have:

$$\begin{aligned} H(Y|\boldsymbol{Z}) &= \frac{1}{2}\log|2\pi e\hat{\boldsymbol{\Sigma}}_{Y,\boldsymbol{Z}}| - \frac{1}{2}\log|2\pi e\hat{\boldsymbol{\Sigma}}_{\boldsymbol{Z}}| \\ H(Y|X,\boldsymbol{Z}) &= \frac{1}{2}\log|2\pi e\hat{\boldsymbol{\Sigma}}_{Y,X,\boldsymbol{Z}}| - \frac{1}{2}\log|2\pi e\hat{\boldsymbol{\Sigma}}_{X,\boldsymbol{Z}}|. \end{aligned} \tag{3.44}$$

Therefore, Equation (3.43) becomes:

$$\begin{aligned} \lim_{N\to\infty} \log \frac{P(D_Y|\boldsymbol{Z})}{P(D_Y|X,\boldsymbol{Z})} = \lim_{N\to\infty} &N\cdot[H(Y|X,\boldsymbol{Z}) - H(Y|\boldsymbol{Z})] \\ &- \frac{1}{2}(|\boldsymbol{Z}|+1)\cdot\log N + \frac{1}{2}(|X|+|\boldsymbol{Z}|+1)\cdot\log N. \end{aligned} \tag{3.45}$$

According to Lemma 3.3.3, $H(Y|X,\boldsymbol{Z}) \leq H(Y|\boldsymbol{Z})$. There are two possible cases here:

Case 1: $X \perp\!\!\!\perp Y|\boldsymbol{Z}$ is true. In this case, according to Lemma 3.3.3 $H(Y|X,\boldsymbol{Z}) = H(Y|\boldsymbol{Z})$; therefore, Equation (3.45) becomes:

$$\lim_{N\to\infty} \log \frac{P(D_Y|\boldsymbol{Z})}{P(D_Y|X,\boldsymbol{Z})} = \lim_{N\to\infty} -\frac{1}{2}(|\boldsymbol{Z}|+1)\cdot\log N + \frac{1}{2}(|X|+|\boldsymbol{Z}|+1)\cdot\log N, \tag{3.46}$$

or equivalently

$$\lim_{N\to\infty} \frac{P(D_Y|\boldsymbol{Z})}{P(D_Y|X,\boldsymbol{Z})} = \lim_{N\to\infty} const.\frac{N^{(|X|+|\boldsymbol{Z}|+1)}}{N^{(|\boldsymbol{Z}|+1)}} = \lim_{N\to\infty} const.N^{|X|} = \infty. \tag{3.47}$$

Case 2: $X \perp\!\!\!\perp Y|\boldsymbol{Z}$ **is false**. In this case, according to Equation (3.45) we have:

$$\begin{aligned}\lim_{N\to\infty} \log \frac{P(D_Y|\boldsymbol{Z})}{P(D_Y|X,\boldsymbol{Z})} = \lim_{N\to\infty} N \cdot [H(Y|X,\boldsymbol{Z}) - H(Y|\boldsymbol{Z})] \\ - \frac{1}{2}(|\boldsymbol{Z}|+1)\cdot \log N + \frac{1}{2}(|X|+|\boldsymbol{Z}|+1)\cdot \log N.\end{aligned}$$

where the first term is of $O(N)$ and the second and third terms are of $O(\log N)$. Therefore, the first term dominates this equation as follows:

$$\lim_{N\to\infty} \log \frac{P(D_Y|\boldsymbol{Z})}{P(D_Y|X,\boldsymbol{Z})} = \lim_{N\to\infty} N \cdot [H(Y|X,\boldsymbol{Z}) - H(Y|\boldsymbol{Z})]\,. \tag{3.48}$$

Since $H(Y|\boldsymbol{Z}) > H(Y|X,\boldsymbol{Z})$ according to Lemma 3.3.3, the term $[H(Y|X,\boldsymbol{Z}) - H(Y|\boldsymbol{Z})]$ becomes a negative number. Consequently, Equation (3.48) becomes $-\infty$, which is equivalent to:

$$\lim_{N\to\infty} \frac{P(D_Y|\boldsymbol{Z})}{P(D_Y|X,\boldsymbol{Z})} = 0. \tag{3.49}$$

□

3.3.3 BSC for mixed variables

This section describes a method to compute the posterior probability of a constraint $R_i = (X \perp\!\!\!\perp Y|\boldsymbol{Z})$ given a dataset D on a set of random variables that includes a mixture of continuous and discrete variable types. In this case, we use the degenerate Gaussian (DG) score introduced in [Andrews et al., 2019]. The key idea in the DG method is to transform discrete variables into a continuous space by using their one-hot vector representation, which results in all continuous variables. The encoding of each variable $X_i \in \boldsymbol{V}$ is as follows [Andrews et al., 2019]:

$$X_i' = \begin{cases} X_i & \text{if } X_i \text{ is continuous} \\ [\mathbb{1}_1(X_i), ...\mathbb{1}_{k-1}(X_i)] & \text{if } X_i \text{ is discrete with } k \text{ values} \end{cases}, \tag{3.50}$$

where the indicator function $\mathbb{1}_k = 1$ if $X_i = k$. After applying this transformation to all variables in $\boldsymbol{V}$, DG uses the BIC score [Schwarz, 1978] to derive marginal likelihoods of the data as follows [Andrews et al., 2019]:

$$\begin{aligned} P(D_i|R_i = true) &= \ell(\hat{\Theta}_{ind}) - \frac{df_{ind}}{2} \log N \\ P(D_i|R_i = false) &= \ell(\hat{\Theta}_{dep}) - \frac{df_{dep}}{2} \log N \,, \end{aligned} \tag{3.51}$$

where N denotes the number of cases in D_i, $\ell(\hat{\Theta}_{ind})$ and $\ell(\hat{\Theta}_{dep})$ are maximum log-likelihood of the data using the independence and dependence BN structures shown in Figures 8a and 8b, respectively. Note that in this case, since the variables are transformed to continuous, we score BN_{dep} using all values of $\boldsymbol{Z}$ because we assume the dependence relationship holds for all values of the continuous variables $\boldsymbol{Z}$. The terms df_{ind} and df_{dep} are degrees of freedom in those BN models, respectively. These equations can then be applied to Equation (3.2) to obtain $P(R_i|D_i)$.

3.4 Combine the FCI Algorithm with BSC

In this section, I describe a hybrid CBN structure learning algorithm that incorporates the BSC test to derive a Bayesian probability that an independence constraint holds (described in Section 3.3) into a constraint-based search (e.g., FCI [Spirtes et al., 2000] or RFCI [Colombo et al., 2012]) to discover the causal structure of the data-generating process in the presence of latent variables. Using this hybrid method, we can then derive a Bayesian probability that the set of independence tests associated with a given causal model are jointly correct, which then can be used to score multiple causal models and output the most probable one(s). In this section, I describe how to combine BSC with a constraint-based method to learn a PAG.

This method adapts a constraint-based CBN structure learning algorithm (e.g., FCI[3]) that applies the BSC method to compute the probability that an independence constraint

[3]Although we use the FCI algorithm throughout this chapter, any other constraint-based method, such as RFCI [Colombo et al., 2012], can be applied in general.

holds instead of using a statistical independence test; we call this algorithm *FCI-BSC*. During the first stage of the search, when FCI requests that an independence constraint to be tested, FCI-BSC uses BSC to determine the probability p that an independence constraint holds. It then samples with probability p whether independence constraint holds and returns that result to the search algorithm. To do so, FCI-BSC generates a random number U from Uniform$[0, 1]$; if $U \leq p$ then it returns *true*, and otherwise, it returns *false*. Ultimately, FCI-BSC will complete the adjacency search in this manner; it then applies the orientation stage (using the BSC test when required), and finally, it returns the learned PAG.

FCI-BSC then repeats the procedure in the previous paragraph s times to generate up to s unique PAG models. Let each repetition be called a *round*. Since the set of constraints generated in each round is determined stochastically (i.e., sampling with probability p), these rounds will produce many different sets of constraints, and consequently, different PAGs. It then outputs a set of at most s PAGs and for each PAG, an associated set of constraints that were queried during the FCI search. Algorithm 5 shows pseudo-code of the FCI-BSC method that inputs dataset D and the number of rounds s. Note that $FCI^{\star}$ in this procedure denotes the FCI search that uses BSC to evaluate each constraint, rather than using frequentist significant testing. The computational complexity of FCI-BSC is $O(n)$ times that of FCI, since it calls the FCI algorithm s times. In the following sections, I introduce three methods to score each generated PAG model $\mathcal{P}_j$ using BSC.

Algorithm 5 FCI-BSC(D, s)

Input: a dataset D, the number of rounds s

Output: a set $\boldsymbol{\mathcal{P}}$ containing PAG members $\mathcal{P}_j$, a set $\boldsymbol{R}$ of constraints

1: Let $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{R}$ be empty sets
2: **for** $j = 1$ **to** n **do**
3: $\quad \mathcal{P}_j, \boldsymbol{R}_j \leftarrow \text{FCI}^{\star}(D)$ ▷ FCI* uses BSC to evaluate independence constraints
4: $\quad \boldsymbol{\mathcal{P}} \leftarrow \boldsymbol{\mathcal{P}} \cup \mathcal{P}_j$
5: $\quad \boldsymbol{R} \leftarrow \boldsymbol{R} \cup \boldsymbol{R}_j$
6: **end for**
7: return $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{R}$

3.5 Scoring a PAG Using BSC

Let $\boldsymbol{R}$ be the union of all the independence conditions tested by FCI-BSC over all rounds, which we will use to score each generated PAG model $\mathcal{P}_j$. Based on the axioms of probability, we have the following equation:

$$P(\mathcal{P}_j|D) = \sum_{\boldsymbol{R}} P(\mathcal{P}_j|\boldsymbol{R}, D) \cdot P(\boldsymbol{R}|D), \tag{3.52}$$

where the sum is over all possible value assignments to the constraints in set $\boldsymbol{R}$. Although Equation (3.52) is valid, it does not provide a useful method for calculating $P(\mathcal{P}_j|D)$. In this section, I introduce a method to derive a way of computing $P(\mathcal{P}_j|D)$ effectively.

Assume that data D only influence belief about a causal model via belief about the conditional independence constraints given by $\boldsymbol{R}$ (i.e., $P(\mathcal{P}_j|\boldsymbol{R}, D) = P(\mathcal{P}_j|\boldsymbol{R})$), which is a standard assumption of constraint-based methods. Therefore, we can rewrite Equation (3.52) as follows:

$$P(\mathcal{P}_j|D) = \sum_{\boldsymbol{R}} P(\mathcal{P}_j|\boldsymbol{R}) \cdot P(\boldsymbol{R}|D). \tag{3.53}$$

Although Equation (3.53) is less general than the full Bayesian approach in Equation (3.52), it is nonetheless more expressive than existing constraint-based methods that in essence assume that $P(\boldsymbol{R}|D) = 1$ for a set of constraints $\boldsymbol{R}$ that are derived using frequentist statistical tests.

Let $\boldsymbol{r}'$ denote the values of all the constraints in $\boldsymbol{R}$ (i.e., $\boldsymbol{R} = \boldsymbol{r}'$), according to the independencies implied by graph $\mathcal{P}_j$. Assuming a constraint-based method finds a set of sufficient independence conditions that distinguishes $\mathcal{P}_j$ from all other PAGs, so that $P(\mathcal{P}_j|\boldsymbol{R} = \boldsymbol{r}') = 1$ and $P(\mathcal{P}_j|\boldsymbol{R} \neq \boldsymbol{r}') = 0$, Equation (3.53) becomes:

$$P(\mathcal{P}_j|D) = \sum_{\boldsymbol{R}} P(\mathcal{P}_j|\boldsymbol{R}) \cdot P(\boldsymbol{R}|D) = P(\boldsymbol{r}'|D). \tag{3.54}$$

Section 3.3 describes a method to compute the probability of one constraint given data, namely, $P(R_i|D_i)$. Now, we need to extend it for a set of constraints $P(\boldsymbol{r}'|D)$ in Equation (3.54). Applying the chain rule of probability, we obtain:

$$
\begin{aligned}
P(\boldsymbol{r}'|D) = P(r'_1, r'_2, ..., r'_m|D) &= \prod_{i=1}^{m} P(r'_i|r'_1, r'_2, ..., r'_{i-1}, D) \\
&= \prod_{i=1}^{m} P(r'_i|r'_1, r'_2, ..., r'_{i-1}, D_i) \text{ (assuming data relevance)},
\end{aligned}
\tag{3.55}
$$

where r'_i denotes the value of constraint R_i according its value given in $\boldsymbol{r}'$. Using Equation (3.55), FCI-BSC determines the most probable generated PAG and its posterior probability.

For each pair of measured nodes, we can also use model averaging to estimate the probability distribution over each PAG edge type as follows: Since PAGs are being sampled (generated) according to their posterior distribution (under the assumption that the constraints are independent of each other), the probability of edge E existing between nodes X_i and X_j is estimated as the fraction of the sampled PAGs that contain edge E between X_i and X_j. In the following subsections, I describe three methods to approximate the joint posterior probability of constraints.

3.5.1 BSC with independence assumption (BSC-I)

In the first method, we assume that constraints in set $\boldsymbol{R} = \{R_1, R_2, ..., R_m\}$, which is a set of all independence constraints obtained by running the FCI-BSC algorithm, are independent of each other. We call this approach BSC-I. Given this assumption and Equation (3.55), BSC-I scores an output graph as follows:

$$
P(\mathcal{P}_j|D) = P(\boldsymbol{r}'|D) = \prod_{i=1}^{m} P(r'_i|D_i), \tag{3.56}
$$

where $\boldsymbol{r}'$ denotes the values of the constraints in $\boldsymbol{R}$ and $P(r'_i|D_i)$ can be computed as described in Section 3.3.

3.5.2 BSC with dependence assumption (BSC-D)

In this scoring approach, we model the possibility that the constraints are dependent, which often happens. The relationships among the constraints can be complicated, and to our knowledge, they have not been modeled previously. In the remainder of this section, we introduce an empirical method to model the relationships among conditional constraints.

Similar to BSC-I, consider $\boldsymbol{R}$ as a set of all the independence constraints queried by the FCI-BSC method. As we mentioned earlier, each constraint $R_i \in \boldsymbol{R}$ has the form $X \perp\!\!\!\perp Y | \boldsymbol{Z}$, where X and Y are variables of dataset D, and $\boldsymbol{Z}$ is a subset of variables not containing X or Y. Each R_i can take two values, *true* (1) or *false* (0); therefore, it can be considered as a binary random variable.

We build a dataset, D_R, of these binary random variables using bootstrap sampling [Efron and Tibshirani, 1994] and the BSC method. To do so, we first bootstrap (resample with replacement) the data D; let $sample_b$ denote a resulting dataset. Then, for each constraint $R_i \in \boldsymbol{R}$, we compute the BSC score using $sample_b$ and set its value to 1 if its BSC score is greater than or equal to 0.5, and 0 otherwise. We repeat this entire procedure bs times to fill in bs rows of empirical data for the constraints. Algorithm 6 provides pseudo-code of this procedure. It takes as input the original dataset D, the number of bootstraps bs, and a set of constraints $\boldsymbol{R}$. It outputs an empirical dataset D_R with bs rows and $m = |\boldsymbol{R}|$ columns. The Bootstrap(D) function in this procedure creates a bootstrap sample from D, and BSC($R_i, sample_b$) computes the BSC score of constraint R_i using $sample_b$.

The empirical data D_R can then be used to learn the relations among the constraints $\boldsymbol{R}$. In particular, we learn a Bayesian network because doing so can be done efficiently with thousands of variables, such networks are expressive in representing the joint relationships among the variables, and inference of the joint state of the variables (the constraints in this application) can be derived efficiently. We use an optimized implementation of the Greedy Equivalence Search (GES) [Chickering, 2002], which is called Fast GES (FGES) [Ramsey, 2015] to learn a Bayesian network structure, $\mathcal{G}_R$, that encodes the dependency relationships among the constraints $\boldsymbol{R}$. We then apply a maximum *a posteriori* estimation method to learn the parameters of $\mathcal{G}_R$ given D_R, which we denote as $\hat{\boldsymbol{\Theta}}_R$. Finally, we use $\mathcal{G}_R$ and $\hat{\boldsymbol{\Theta}}_R$ to factorize $P(\boldsymbol{r}'|D)$ and score the output PAG as follows:

$$P(\mathcal{P}_j|D) = P(\boldsymbol{r}'|D) = \prod_{i=1}^{m} P(r'_i|\boldsymbol{r}'_{Pa(R_i)}, D), \tag{3.57}$$

where r'_i and $\boldsymbol{r}'_{Pa(R_i)}$ denote the values of R_i and its parents $\boldsymbol{Pa}(R_i)$ in $\boldsymbol{r}'$, respectively. Algorithm 7 provides pseudo-code of BSC-D method.

Algorithm 6 ConstraintDataGeneration(D, bs, $\boldsymbol{R}$)

Input: a dataset D, the number of bootstraps bs, and a set of constraints $\boldsymbol{R} = \{R_1, R_2, \ldots, R_m\}$

Output: an empirical dataset D_R with bs rows and $m = |\boldsymbol{R}|$ columns

Let $D_R[bs, m]$ be an empty 2-d array with bs rows and m columns
for $b = 1$ **to** bs **do**
    $sample_b \leftarrow \text{Bootstrap}(D)$
    for $R_i \in \boldsymbol{R}$ **do**
        $p \leftarrow \text{BSC}(R_i, sample_b)$
        if $p \geq 0.5$ **then**
            $D_R[b, i] \leftarrow 1$
        else
            $D_R[b, i] \leftarrow 0$
        end if
    end for
end for
return $D_R[bs, m]$

Algorithm 7 BSC-D(D_R, $\boldsymbol{R} = \boldsymbol{r}'$)

Input: an empirical dataset D_R (generated using Algorithm 6), a set of constraints $\boldsymbol{R} = \boldsymbol{r}'$

Output: compute $P(\boldsymbol{r}'|D)$ in Equation (3.57) using D_R

$\mathcal{G}_R \leftarrow \text{GES}\ (D_R)$
$\hat{\boldsymbol{\Theta}}_R \leftarrow \arg\max_{\boldsymbol{\Theta}_R} P(\boldsymbol{\Theta}_R|\mathcal{G}_R, D_R)$ ▷ $\hat{\boldsymbol{\Theta}}_R$ is the maximum likelihood estimates of the probabilities in $\mathcal{G}_R$.
$p = 1$
for $R_i = r'_i \in \boldsymbol{R} = \boldsymbol{r}'$ **do**
    $p = p * \hat{\boldsymbol{\Theta}}_R(r'_i|\boldsymbol{r}'_{Pa(R_i)})$ ▷ $\boldsymbol{r}'_{Pa(R_i)}$ denotes the values of the parents of R_i in $\mathcal{G}_R$
end for
return p

3.5.3 BSC with a local dependence assumption (BSC-LD)

In this section, similar to Section 3.5.2, we introduce a method to approximate the joint posterior probabilities of constraints $\boldsymbol{R}$ assuming that the constraints are dependent. This section describes a more efficient way of locally modeling the relationships among the constraints by grouping them based on distinct pairs of variables (X_i, X_j) that exist in $\boldsymbol{R}$. For each pair of variables (X_i, X_j), we collect all the independence constraints that are about X_i and X_j, regardless of the conditioning set of variables; let $\boldsymbol{R}_{X_iX_j}$ denote this set. Then, we learn a BN structure $\mathcal{G}_{X_iX_j}$ and its parameters $\hat{\boldsymbol{\Theta}}_{X_iX_j}$ to model the relationships among constraints in $\boldsymbol{R}_{X_iX_j}$ using the parts of D_R that correspond to the constraints in $\boldsymbol{R}_{X_iX_j}$ (i.e., $D_{R_{X_iX_j}}$). Doing this for all distinct pairs of variables in $\boldsymbol{R}$ will result in multiple BNs. Finally, we aggregate these BNs and their parameters as one BN model denoted as $(\mathcal{G}_R, \hat{\boldsymbol{\Theta}}_R)$. Note that aggregating these BNs will not produce any cycles because the variables in each BN, which is a set of constraints associated with pairs of variables, are mutually exclusive. Finally, we use $(\mathcal{G}_R, \hat{\boldsymbol{\Theta}}_R)$ to factorize the joint probability of the constraints $P(\boldsymbol{r}|D)$ to score $\mathcal{P}_j$. Algorithm 8 provides pseudo-code of the BSC-LD method.

The BSC-D and BSC-LD methods do not consider how those relationships might be influenced (via structure priors) by the restrictions imposed by the underlying data-generating CBN models, which is an area for future research.

3.6 Experimental Results

This section describes the experimental methods and results that we used to investigate the performance of the FCI-BSC method, which uses BSC test, versus the FCI method, which uses a frequentist statistical test. For FCI-BSC, we also report the results using each of the BSC-I, BSC-D, and BSC-LD scoring techniques. To do so, we simulated data from both randomly generated BN models and manually constructed BN models, which are described in Sections 3.6.1 and 3.6.2, respectively.

Algorithm 8 BSC-LD(D_R, $\boldsymbol{R} = \boldsymbol{r}'$)

Input: an empirical dataset D_R (generated using Algorithm 6), a set of constraints $\boldsymbol{R} = \boldsymbol{r}'$

Output: compute $P(\boldsymbol{r}'|D)$ in Equation (3.57) using D_R

1: **for** $(X_i, X_j) \in \boldsymbol{R}$ **do**
2: Let $\boldsymbol{R}_{X_iX_j}$ be the set of constraints that are about X_i and X_j
3: Let $D_{R_{X_iX_j}}$ be the part of the dataset D_R that corresponds to $\boldsymbol{R}_{X_iX_j}$
4: $\{\mathcal{G}_{X_iX_j}\} \leftarrow$ GES $(D_{R_{X_iX_j}})$
5: $\{\hat{\boldsymbol{\Theta}}_{X_iX_j}\} \leftarrow \arg\max_{\boldsymbol{\Theta}_{X_iX_j}} P(\boldsymbol{\Theta}_{X_iX_j}|\mathcal{G}_{X_iX_j}, D_{R_{X_iX_j}})$ ▷ $\{\hat{\boldsymbol{\Theta}}_{X_iX_j}\}$ is the maximum likelihood estimates of the probabilities in $\mathcal{G}_{X_iX_j}$.
6: **end for**
7: Let $\mathcal{G}_R$ be the aggregate of all BN structures in $\{\mathcal{G}_{X_iX_j}\}$
8: Let $\hat{\boldsymbol{\Theta}}_R$ be the aggregate of all BN parameters in $\{\hat{\boldsymbol{\Theta}}_{X_iX_j}\}$
9: $p = 1$
10: **for** $R_i = r'_i \in \boldsymbol{R} = \boldsymbol{r}'$ **do**
11: $p = p * \hat{\boldsymbol{\Theta}}_R(r'_i|\boldsymbol{r}'_{Pa(R_i)})$ ▷ $\boldsymbol{r}'_{Pa(R_i)}$ denotes the values of the parents of R_i in $\mathcal{G}_R$
12: **end for**
13: return p

3.6.1 Simulated data from randomly generated BN models

In order to evaluate the performance of FCI-BSC versus FCI, we conducted simulation studies to randomly generate BNs that are used to simulate data as follows.

1. For each Bayesian network $\mathcal{M} = (\mathcal{G}, \boldsymbol{\Theta})$, we first created a DAG $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$ with $|\boldsymbol{V}| = \{10, 20, 50\}$ random variables and $|\boldsymbol{E}| = \{2|\boldsymbol{V}|, 4|\boldsymbol{V}|, 6|\boldsymbol{V}|\}$ edges. To generate a DAG $\mathcal{G}$, we first create an arbitrary ordering of variables[4]. Then, we uniformly randomly added edges to $\mathcal{G}$ in a forward direction until obtaining the specified number of edges. The DAGs generated in this way have a power-law-type distribution over the number of parents, with some variables having many more than the average number of parents.
2. We then parametrized the distribution of each random variable $X \in \boldsymbol{V}$ given its parents

[4]This ordering is only used to generate the BNs; we do not use it when applying FCI-BSC or FCI.

$\boldsymbol{Pa}(X)$ according to DAG $\mathcal{G}$. Given different types of variables in $\boldsymbol{V}$, we used different settings as follows:

- **All variables are discrete**: In this case, each variable X may have 2, 3, or 4 categories, which is chosen randomly. Given the number of categories of X and its parents $\boldsymbol{Pa}(X)$, we randomly initialized the conditional probability table for $P(X|\boldsymbol{Pa}(X))$ under the constraints that follow from the axioms of probability theory.
- **All variables are continuous**: In this case, we parametrized $P(X|\boldsymbol{Pa}(X))$ as a structural equation model (SEM):

$$X = \sum_{Y \in \boldsymbol{Pa}(X)} \beta_Y \cdot Y + \epsilon_X \ ,$$

  where ϵ_X is a zero-mean Gaussian noise term and β_Y is a linear coefficient. In our experiments, similar to [Ramsey, 2015, Silva et al., 2006], the variance of noise term ϵ_X is uniformly randomly chosen from the interval $[1.0, 3.0]$ and β_Y is uniformly randomly drawn from the interval $[-0.7, -0.2] \cup [0.2, 0.7]$. This choice of parameter values for the simulations implies that the variance of the variables is largely due to the error term, which makes structure learning more difficult.
- **The variables are a mixture of discrete and continuous**: In this case, we randomly assigned each variable X to be either continuous or discrete with probability 0.5. Then, we parametrized $P(X|\boldsymbol{Pa}(X))$ using the conditional Gaussian model introduced in [Andrews et al., 2018], using similar parameters as described for discrete and continuous variable types.

3. We randomly set $L = 20\%$ of variables to be latent (i.e., hidden). These variables were chosen at random from a list of all variables that are common causes of two or more of the measured variables. If there are fewer common causes than $L = 20\%$ of variables, we selected from a list of all variables that are common effects of two or more of the measured variables.
4. We used each BN model $\mathcal{M} = (\mathcal{G}, \boldsymbol{\Theta})$ to generate a training dataset D with $N = \{200, 1000, 5000\}$ training samples.

5. We used the training dataset D generated in step 4 to learn a PAG structure $\mathcal{P}_F$ using the FCI algorithm (Section 3.2). For the independence testing used in FCI, we applied a chi-squared test of independence when the data includes discrete variables, Fisher's Z test when the data includes continuous variables, and a likelihood ratio test using the degenerate Gaussian (DG) score introduced by [Andrews et al., 2019] when the data includes a mixture of discrete and continuous variables. The DG method first transforms mixed variables to continuous variables using Equation (3.50); then, it uses the BIC score to perform a log-likelihood ratio test. We used $\alpha = 0.05$ for these statistical tests, which is a common alpha value used with FCI.
6. We also used the training dataset D generated in step 4 to learn a set of PAG structures using the FCI-BSC algorithm. In applying FCI-BSC, we used appropriate versions of BSC test described in Sections 3.3.1, 3.3.2, 3.3.3, which are developed for discrete, continuous, and mixed data types, respectively. We sampled 100 PAG models $\boldsymbol{\mathcal{P}}$ according to the FCI-BSC method (i.e., $s = 100$ in Algorithm 5). We computed the posterior probability of each PAG $\mathcal{P}_i \in \boldsymbol{\mathcal{P}}$ using the BSC-I, BSC-D, and BSC-LD methods to obtain the most probable PAG by each scoring method; we denote the highest scoring PAGs obtained by these methods as $\mathcal{P}_I$, $\mathcal{P}_D$, $\mathcal{P}_{LD}$, respectively. For the BSC-D and BSC-LD scoring methods, we bootstrapped the data 500 times (i.e., $bs = 500$ in Algorithm 6) to create the empirical data.
7. Finally, we computed evaluation measures (described below) to compare the structure recovery performance of FCI-BSC versus FCI. To do so, we compared $\mathcal{P}_I$, $\mathcal{P}_D$, $\mathcal{P}_{LD}$ (which are learned by FCI-BSC), and $\mathcal{P}_F$ (which is learned by FCI) to the ground-truth PAG that is consistent with the data-generating DAG (with latent variables). We obtained the ground-truth PAG structure $\mathcal{P}_{truth}$ by using an independence oracle that has access to the data-generating model described in lines 1-3 above.

For each simulation setting mentioned above, steps 1 through 7 were repeated for 10 randomly generated BNs and the performance results were averaged. The evaluation measures we used include structural Hamming distance (SHD), and precision (P) and recall (R) for edge adjacency and arrowhead orientation, which are described in the following section.

3.6.1.1 PAG structure discovery performance measures In this section, I describe the evaluation measures that are used to calculate the structural similarity of the discovered PAG $\mathcal{P}_{output}$, which can be $\mathcal{P}_I$, $\mathcal{P}_D$, $\mathcal{P}_{LD}$ when using FCI-BSC and $\mathcal{P}_F$ when using FCI, versus the ground-truth PAG $\mathcal{P}_{truth}$.

We used structural Hamming distance (SHD) that counts the edge modifications that include added, deleted, and reoriented edges, by comparing each possible edge in $\mathcal{P}_{output}$ and $\mathcal{P}_{truth}$. We define three versions of SHD for PAGs as follows:

- **Strict SHD (S-SHD)**: This version counts any edge modifications, which are added, deleted, and reoriented edges. The S-SHD would be 0 if for a given pair of measured variables the edge in $\mathcal{P}_{output}$ is exactly the same as the edge in PAG $\mathcal{P}_{truth}$; otherwise, it is 1. Any extra or missing edge would also count as 1 in terms of S-SHD. Table 1a shows how to compute S-SHD for PAGs.
- **Lenient SHD (L-SHD)**: This version allows general edges that include circle endpoints to be compatible with their specializations. For example, the L-SHD between $A \circ\!\!\rightarrow B$ and $A \rightarrow B$ is 0 because these edges are compatible. However, the L-SHD between $A \rightarrow B$ and $B \rightarrow A$ is 1 because they are not compatible. L-SHD is symmetric regarding the output and the truth edges, as shown in Table 1b.
- **Adjacency SHD (A-SHD)**: In this version, we compute SHD on the skeleton-level by comparing the adjacencies of two graphs, which disregards the edge orientations and only counts the edge modifications of the adjacency graph that includes added and deleted edges. For example, if one graph includes $A \circ\!\!-\!\!\circ B$ but there is no edge between A and B in the other one, then A-SHD would be 1.

Other performance criteria we used to evaluate discrimination are precision (P) and recall (R) for adjacencies and arrowheads:

- **Adjacency precision (AP)**: we compute the ratio of correctly predicted edges in $\mathcal{P}_{output}$ to all predicted edges in $\mathcal{P}_{output}$ (without considering orientations of edges) as follows:

$$\text{AP} = \frac{\#\text{correctly predicted adjacencies}}{\#\text{predicted adjacencies}} \tag{3.58}$$

Table 1: Two types of SHD for PAGs. The rows and columns correspond to the edge types output by the algorithm and the data-generating edge types, respectively.

(a) Strict SHD (S-SHD) for PAGs.

Output Edge/ Truth Edge	$A \to B$	$A \leftrightarrow B$	$A \circ\!\!\to B$	$A \circ\!\!-\!\!\circ B$	$A \quad B$
$A \to B (B \to A)$	0 (1)	1	1	1	1
$A \leftrightarrow B$	1	0	1	1	1
$A \circ\!\!\to B (B \circ\!\!\to A)$	1	1	0 (1)	1	1
$A \circ\!\!-\!\!\circ B$	1	1	1	0	1
$A \quad B$	1	1	1	1	0

(b) Lenient SHD (L-SHD) for PAGs.

Output Edge/ Truth Edge	$A \to B$	$A \leftrightarrow B$	$A \circ\!\!\to B$	$A \circ\!\!-\!\!\circ B$	$A \quad B$
$A \to B (B \to A)$	0 (1)	1	0	0	1
$A \leftrightarrow B$	1	0	0	0	1
$A \circ\!\!\to B (B \circ\!\!\to A)$	0 (1)	0	0	0	1
$A \circ\!\!-\!\!\circ B$	0	0	0	0	1
$A \quad B$	1	1	1	1	0

- **Adjacency recall (AR)**: we compute the ratio of correctly predicted edges in $\mathcal{P}_{output}$ to all true edges in $\mathcal{P}_{truth}$ (without considering the edges' orientations) as follows:

$$\text{AR} = \frac{\#\text{correctly predicted adjacencies}}{\#\text{true adjacencies}} \tag{3.59}$$

- **Arrowhead precision (AHP)**: considering the pairs of measured variables that have an edge between them in the predicted graph $\mathcal{P}_{output}$, we compute the ratio of correctly predicted arrowheads in $\mathcal{P}_{output}$ to all predicted arrowheads in $\mathcal{P}_{output}$ as follows:

$$\text{AHP} = \frac{\#\text{correctly predicted arrowheads}}{\#\text{predicted arrowheads}} \tag{3.60}$$

- **Arrowhead recall (AHR)**: considering the pairs of measured variables that have an edge between them in the ground-truth PAG $\mathcal{P}_{truth}$, we compute the ratio of correctly

predicted arrowheads in $\mathcal{P}_{output}$ to all true arrowheads in $\mathcal{P}_{truth}$ as follows:

$$\text{AHR} = \frac{\#\text{correctly predicted arrowheads}}{\#\text{true arrowheads}} \tag{3.61}$$

Note that an arrowhead in a PAG indicates causation due to either a measured or a latent variable (see Section 2.1.2 and the example given in Figure 5).

3.6.1.2 Simulation results for discrete variable type Tables 2, 3, and 4 show the average adjacency P (AP) and R (AR), and arrowhead P (AHP) and R (AHR) results of the FCI-BSC (with BSC-I scoring method)[5] and the FCI (with chi-squared test) algorithms over 10 randomly generated CBNs described in Section 3.6.1, using $N = \{200, 1000, 5000\}$ training instances, respectively[6]. For $N = 200$, both FCI-BSC and FCI almost always perform similarly (Table 2). When the number of training instances increases to $N = \{1000, 5000\}$, the FCI method performs significantly better than FCI-BSC in terms of AR and AHR measures based on Wilcoxon signed rank test at 5% significance level, while FCI-BSC has a slightly better AP and AHP performance as shown in the summary statistics (Tables 3 and Tables 4).

Tables 5, 6, and 7 show the SHD results of the FCI-BSC (with BSC-I scoring method) and the FCI (with chi-squared test) algorithms over 10 randomly generated CBNs described in Section 3.6.1, using $N = \{200, 1000, 5000\}$ training instances, respectively. For all these cases, both FCI-BSC and FCI almost always perform similarly in terms of added edges S-SHD, while FCI-BSC has fewer orientation errors. Also, FCI has fewer number of deleted edges and performs better in terms of L-SHD and A-SHD measures.

[5]Results using BSC-D and BSC-LD scoring methods are similar to using BSC-I; therefore, we only report the results for BCS-I, which is a simpler and more efficient method.

[6]Omitted rows in the tables represent the settings that failed to return a result in under 72 hours.

Table 2: Discrete variable type: Adjacency precision (AP) and recall (AR), and arrowhead precision (AHP) and recall (AHR) results for FCI-BSC (with BSC-I scoring method) and FCI (with chi-squared test) when using $N = 200$ training cases. The numbers after '±' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	AP	AR	AHP	AHR
10	20	BSC-I	1.00 ± 0.00	0.12 ± 0.09	0.10 ± 0.30	0.01 ± 0.04
		FCI	0.97 ± 0.07	**0.25 ± 0.09**	0.20 ± 0.21	0.06 ± 0.07
	40	BSC-I	0.40 ± 0.49	0.03 ± 0.04	0.40 ± 0.49	0.40 ± 0.49
		FCI	**1.00 ± 0.00**	**0.12 ± 0.04**	0.27 ± 0.40	0.22 ± 0.39
	60	BSC-I	0.90 ± 0.30	0.05 ± 0.03	0.60 ± 0.49	0.60 ± 0.49
		FCI	1.00 ± 0.00	**0.14 ± 0.06**	0.12 ± 0.30	0.11 ± 0.30
20	40	BSC-I	0.98 ± 0.06	0.07 ± 0.03	0.20 ± 0.33	0.01 ± 0.02
		FCI	0.88 ± 0.15	**0.15 ± 0.05**	0.33 ± 0.18	**0.05 ± 0.04**
	80	BSC-I	1.00 ± 0.00	0.03 ± 0.01	0.15 ± 0.32	0.01 ± 0.01
		FCI	0.95 ± 0.08	**0.08 ± 0.02**	0.37 ± 0.17	**0.03 ± 0.02**
	120	BSC-I	0.80 ± 0.40	0.02 ± 0.01	0.00 ± 0.00	0.00 ± 0.00
		FCI	0.95 ± 0.10	**0.05 ± 0.02**	0.06 ± 0.13	0.01 ± 0.02
50	100	BSC-I	**0.97 ± 0.06**	0.04 ± 0.01	0.30 ± 0.46	0.00 ± 0.00
		FCI	0.75 ± 0.09	**0.11 ± 0.03**	0.33 ± 0.07	**0.04 ± 0.01**
Summary statistics		BSC-I	0.86 ± 0.20	0.05 ± 0.03	0.25 ± 0.19	0.15 ± 0.23
		FCI	0.93 ± 0.08	0.13 ± 0.06	0.24 ± 0.11	0.07 ± 0.06

Table 3: Discrete variable type: Adjacency precision (AP) and recall (AR), and arrowhead precision (AHP) and recall (AHR) results for FCI-BSC (with BSC-I scoring method) and FCI (with chi-squared test) when using $N = 1000$ training cases. The numbers after '±' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	AP	AR	AHP	AHR
10	20	BSC-I	1.00 ± 0.00	0.26 ± 0.07	0.08 ± 0.16	0.02 ± 0.04
		FCI	0.99 ± 0.03	**0.43 ± 0.12**	0.17 ± 0.12	**0.15 ± 0.11**
	40	BSC-I	1.00 ± 0.00	0.11 ± 0.04	0.20 ± 0.33	0.12 ± 0.30
		FCI	1.00 ± 0.00	**0.28 ± 0.09**	0.16 ± 0.22	0.13 ± 0.19
	60	BSC-I	1.00 ± 0.00	0.16 ± 0.06	0.24 ± 0.39	0.23 ± 0.39
		FCI	1.00 ± 0.00	**0.34 ± 0.09**	0.06 ± 0.13	0.08 ± 0.17
20	40	BSC-I	1.00 ± 0.00	0.14 ± 0.07	0.42 ± 0.31	0.04 ± 0.03
		FCI	0.97 ± 0.04	**0.27 ± 0.09**	0.43 ± 0.12	**0.17 ± 0.08**
	80	BSC-I	1.00 ± 0.00	0.07 ± 0.02	0.44 ± 0.32	0.03 ± 0.02
		FCI	0.99 ± 0.03	**0.15 ± 0.04**	0.38 ± 0.14	**0.09 ± 0.05**
	120	BSC-I	1.00 ± 0.00	0.04 ± 0.01	0.08 ± 0.19	0.01 ± 0.01
		FCI	1.00 ± 0.00	**0.09 ± 0.02**	**0.31 ± 0.19**	**0.05 ± 0.03**
50	100	BSC-I	**1.00 ± 0.00**	0.11 ± 0.02	0.43 ± 0.14	0.03 ± 0.01
		FCI	0.97 ± 0.02	**0.22 ± 0.04**	0.47 ± 0.06	**0.13 ± 0.04**
Summary statistics		BSC-I	1.00 ± 0.00	0.13 ± 0.06	0.27 ± 0.15	0.07 ± 0.07
		FCI	0.99 ± 0.01	0.26 ± 0.10	0.28 ± 0.14	0.11 ± 0.04

Table 4: Discrete variable type: Adjacency precision (AP) and recall (AR), and arrowhead precision (AHP) and recall (AHR) results for FCI-BSC (with BSC-I scoring method) and FCI (with chi-squared test) when using $N = 5000$ training cases. The numbers after '$\pm$' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	AP	AR	AHP	AHR
10	20	BSC-I	1.00 ± 0.00	0.39 ± 0.11	0.45 ± 0.28	0.17 ± 0.08
		FCI	1.00 ± 0.00	**0.66 ± 0.09**	0.36 ± 0.13	**0.44 ± 0.16**
	40	BSC-I	1.00 ± 0.00	0.22 ± 0.08	0.14 ± 0.18	0.07 ± 0.09
		FCI	1.00 ± 0.00	**0.48 ± 0.09**	0.17 ± 0.17	**0.21 ± 0.22**
	60	BSC-I	1.00 ± 0.00	0.27 ± 0.09	0.13 ± 0.30	0.12 ± 0.30
		FCI	1.00 ± 0.00	**0.60 ± 0.15**	0.07 ± 0.13	0.09 ± 0.18
20	40	BSC-I	1.00 ± 0.00	0.27 ± 0.08	0.47 ± 0.09	0.16 ± 0.08
		FCI	1.00 ± 0.00	**0.45 ± 0.12**	0.45 ± 0.06	**0.33 ± 0.12**
	80	BSC-I	1.00 ± 0.00	0.13 ± 0.04	0.39 ± 0.13	0.08 ± 0.03
		FCI	0.99 ± 0.02	**0.26 ± 0.06**	0.42 ± 0.09	**0.21 ± 0.06**
	120	BSC-I	1.00 ± 0.00	0.06 ± 0.01	0.42 ± 0.26	0.03 ± 0.02
		FCI	0.99 ± 0.02	**0.15 ± 0.02**	0.34 ± 0.14	**0.11 ± 0.05**
50	100	BSC-I	**1.00 ± 0.01**	0.19 ± 0.04	0.50 ± 0.06	0.10 ± 0.03
		FCI	0.98 ± 0.02	**0.33 ± 0.07**	0.50 ± 0.06	**0.24 ± 0.06**
Summary statistics		BSC-I	1.00 ± 0.00	0.22 ± 0.10	0.36 ± 0.15	0.11 ± 0.05
		FCI	1.00 ± 0.01	0.42 ± 0.17	0.33 ± 0.14	0.23 ± 0.11

Table 5: Discrete variable type: Strict SHD (S-SHD), lenient SHD (L-SHD), and adjacency SHD (A-SHD) results for FCI-BSC (with BSC-I scoring method) and FCI (with chi-squared test) when using $N = 200$ training cases. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	Added	Deleted	Reoriented	S-SHD	L-SHD	A-SHD
10	20	BSC-I	0	16.7	**1.30**	18	16.9	16.7
		FCI	0.1	**14.40**	2.7	17.2	**14.90**	**14.50**
	40	BSC-I	0	26.3	**0.20**	26.5	26.3	26.3
		FCI	0	**23.90**	1.9	25.8	**24.00**	**23.90**
	60	BSC-I	0	26.5	**0.70**	27.2	26.5	26.5
		FCI	0	**23.90**	3.3	27.2	**24.00**	**23.90**
20	40	BSC-I	0.1	48.6	**1.90**	50.6	49.1	48.7
		FCI	1	**44.40**	5.8	51.2	**47.40**	**45.40**
	80	BSC-I	0	88.3	**1.70**	90	88.7	88.3
		FCI	0.4	**83.40**	6	89.8	**86.00**	**83.80**
	120	BSC-I	0	112.9	**0.90**	113.8	112.9	112.9
		FCI	0.2	**108.80**	3.8	**112.80**	**109.80**	**109.00**
50	100	BSC-I	**0.20**	143.8	**3.70**	**147.70**	144.2	144
		FCI	5.1	**134.10**	12.8	152	144.4	**139.20**
Summary statistics		BSC-I	0.04	66.16	1.49	67.69	66.37	66.2
		FCI	0.97	61.84	5.19	68	64.36	62.81

Table 6: Discrete variable type: Strict SHD (S-SHD), lenient SHD (L-SHD), and adjacency SHD (A-SHD) results for FCI-BSC (with BSC-I scoring method) and FCI (with chi-squared test) when using $N = 1000$ training cases. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	Added	Deleted	Reoriented	S-SHD	L-SHD	A-SHD
10	20	BSC-I	0	14.2	**3.90**	18.1	14.9	14.2
		FCI	0.1	**11.00**	7.3	18.4	**12.80**	**11.10**
	40	BSC-I	0	24.1	**2.00**	26.1	24.3	24.1
		FCI	0	**19.40**	6.2	25.6	**19.90**	**19.40**
	60	BSC-I	0	23.3	**3.60**	26.9	23.5	23.3
		FCI	0	**18.40**	8.1	26.5	**18.40**	**18.40**
20	40	BSC-I	0	45	**4.40**	49.4	46.2	45
		FCI	0.4	**38.30**	10.8	49.5	**42.90**	**38.70**
	80	BSC-I	0	85	**3.80**	88.8	86.4	85
		FCI	0.2	**77.10**	11.8	89.1	**82.40**	**77.30**
	120	BSC-I	0	108.8	**3.80**	112.6	109.4	108.8
		FCI	0	**102.70**	9	111.7	**104.40**	**102.70**
50	100	BSC-I	**0.00**	134.4	**12.20**	146.6	137.9	134.4
		FCI	1.1	**117.70**	26.4	145.2	**130.40**	**118.80**
Summary statistics		BSC-I	0	62.11	4.81	66.93	63.23	62.11
		FCI	0.26	54.94	11.37	66.57	58.74	55.2

Table 7: Discrete variable type: Strict SHD (S-SHD), lenient SHD (L-SHD), and adjacency SHD (A-SHD) results for FCI-BSC (with BSC-I scoring method) and FCI (with chi-squared test) when using $N = 5000$ training cases. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	Added	Deleted	Reoriented	S-SHD	L-SHD	A-SHD
10	20	BSC-I	0	11.7	**4.90**	16.6	12.6	11.7
		FCI	0	**6.50**	10.2	16.7	**8.40**	**6.50**
	40	BSC-I	0	21.1	**5.00**	26.1	21.4	21.1
		FCI	0	**14.00**	12	26	**14.70**	**14.00**
	60	BSC-I	0	20.4	**6.50**	26.9	20.8	20.4
		FCI	0	**11.20**	15	26.2	**11.70**	**11.20**
20	40	BSC-I	0	38.8	**10.10**	48.9	42	38.8
		FCI	0	**29.90**	19.3	49.2	**36.10**	**29.90**
	80	BSC-I	0	79.4	**9.50**	88.9	82.8	79.4
		FCI	0.3	**68.20**	20.3	88.8	**74.90**	**68.50**
	120	BSC-I	0	105.9	**5.60**	111.5	106.9	105.9
		FCI	0.1	**96.00**	15.1	111.2	**100.00**	**96.10**
50	100	BSC-I	**0.10**	122.4	**20.50**	143	131.1	122.5
		FCI	0.9	**101.40**	40.4	142.7	**118.10**	**102.30**
Summary statistics		BSC-I	0.01	57.1	8.87	65.99	59.66	57.11
		FCI	0.19	46.74	18.9	65.83	51.99	46.93

3.6.1.3 Simulation results for continuous variable type Tables 8, 9, and 10 show the average adjacency P (AP) and R (AR), and arrowhead P (AHP) and R (AHR) results of the FCI-BSC (with BSC-I scoring method)[7] and the FCI (with Fisher's Z test) algorithms over 10 randomly generated CBNs described in Section 3.6.1, using $N = \{200, 1000, 5000\}$ training instances, respectively. For $N = 200$, both methods have similar performance for smaller BNs (e.g., 10 variables), but for the larger BNs (e.g., 50 variables), FCI-BSC shows better AP and AHP performance and FCI shows better AR and AHR performance (Table 8). As the sample size increases to $N = 5000$, the FCI-BSC method performs better in terms of AP and AHP measures, while it has a slightly lower AR and AHR performance as shown in the summary statistics of Table 10.

Tables 11, 12, and 13 show the SHD results of the FCI-BSC (with BSC-I scoring method) and the FCI (with Fisher's Z test) algorithms over 10 randomly generated CBNs described in Section 3.6.1, using $N = \{200, 1000, 5000\}$ training instances, respectively. For $N = \{200, 1000\}$, FCI-BSC almost always performs similar to FCI for the smaller CBNs (e.g., 10 variables) in terms of all SHD measures, but as the CBN gets larger (e.g., 50 variables) FCI-BSC performs significantly better in terms of the number of added and reoriented edges, and S-SHD measures, while FCI often has fewer deleted edges (Tables 11 and 12). When the number of training instances increases to $N = 5000$, all SHD measures notably improved for both methods, while FCI-BSC almost always has lower S-SHD, L-SHD, and A-SHD (Table 13), which is mainly due to a fewer number of added and reoriented edges.

[7]Results using BSC-D and BSC-LD scoring methods are similar to using BSC-I; therefore, we only report the results for BCS-I, which is a simpler and more efficient method.

Table 8: Continuous variable type: Adjacency precision (AP) and recall (AR), and arrowhead precision (AHP) and recall (AHR) results for FCI-BSC (with BSC-I scoring method) and FCI (with Fisher's Z test) when using $N = 200$ training cases. The numbers after '±' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	AP	AR	AHP	AHR
10	20	BSC-I	0.86 ± 0.14	0.64 ± 0.16	0.55 ± 0.33	0.56 ± 0.37
		FCI	0.85 ± 0.15	0.64 ± 0.16	0.55 ± 0.33	0.55 ± 0.37
	40	BSC-I	0.97 ± 0.04	0.57 ± 0.09	0.46 ± 0.13	0.47 ± 0.14
		FCI	0.96 ± 0.05	0.58 ± 0.10	0.43 ± 0.12	0.48 ± 0.15
	60	BSC-I	0.99 ± 0.02	0.40 ± 0.04	0.45 ± 0.12	0.31 ± 0.05
		FCI	0.99 ± 0.02	0.41 ± 0.05	0.47 ± 0.14	0.34 ± 0.05
20	40	BSC-I	**0.86 ± 0.06**	0.61 ± 0.16	**0.44 ± 0.19**	0.40 ± 0.20
		FCI	0.79 ± 0.07	0.61 ± 0.13	0.38 ± 0.16	0.43 ± 0.20
	80	BSC-I	0.97 ± 0.03	0.57 ± 0.06	**0.51 ± 0.06**	0.47 ± 0.08
		FCI	0.94 ± 0.04	**0.59 ± 0.06**	0.49 ± 0.06	**0.51 ± 0.07**
	120	BSC-I	1.00 ± 0.00	0.35 ± 0.07	0.51 ± 0.04	0.27 ± 0.07
		FCI	0.99 ± 0.01	**0.37 ± 0.07**	0.51 ± 0.04	**0.30 ± 0.06**
50	100	BSC-I	**0.72 ± 0.04**	0.56 ± 0.08	**0.33 ± 0.06**	0.35 ± 0.08
		FCI	0.57 ± 0.05	**0.59 ± 0.08**	0.23 ± 0.03	**0.41 ± 0.08**
	200	BSC-I	**0.94 ± 0.03**	0.48 ± 0.07	**0.51 ± 0.05**	0.36 ± 0.07
		FCI	0.88 ± 0.03	**0.52 ± 0.07**	0.44 ± 0.03	**0.42 ± 0.08**
	300	BSC-I	**0.98 ± 0.02**	0.32 ± 0.04	**0.53 ± 0.03**	0.26 ± 0.04
		FCI	0.95 ± 0.03	**0.35 ± 0.05**	0.50 ± 0.02	**0.29 ± 0.05**
Summary statistics		BSC-I	0.92 ± 0.09	0.50 ± 0.11	0.48 ± 0.06	0.38 ± 0.09
		FCI	0.88 ± 0.13	0.52 ± 0.11	0.44 ± 0.09	0.41 ± 0.08

Table 9: Continuous variable type: Adjacency precision (AP) and recall (AR), and arrowhead precision (AHP) and recall (AHR) results for FCI-BSC (with BSC-I scoring method) and FCI (with Fisher's Z test) when using $N = 1000$ training cases. The numbers after '±' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	AP	AR	AHP	AHR
10	20	BSC-I	0.97 ± 0.05	0.80 ± 0.09	0.74 ± 0.23	0.63 ± 0.20
		FCI	0.93 ± 0.06	0.84 ± 0.09	0.67 ± 0.19	0.70 ± 0.18
	40	BSC-I	0.96 ± 0.05	0.59 ± 0.14	0.42 ± 0.11	0.52 ± 0.22
		FCI	0.94 ± 0.05	0.62 ± 0.15	0.42 ± 0.07	0.58 ± 0.21
	60	BSC-I	0.99 ± 0.02	0.54 ± 0.07	0.42 ± 0.10	0.44 ± 0.07
		FCI	0.99 ± 0.02	0.55 ± 0.06	0.43 ± 0.10	**0.49 ± 0.06**
20	40	BSC-I	**0.95 ± 0.03**	0.74 ± 0.13	**0.61 ± 0.16**	0.63 ± 0.23
		FCI	0.81 ± 0.08	0.78 ± 0.12	0.43 ± 0.15	**0.68 ± 0.22**
	80	BSC-I	0.99 ± 0.02	0.67 ± 0.12	0.57 ± 0.04	0.58 ± 0.15
		FCI	0.97 ± 0.03	**0.70 ± 0.11**	0.56 ± 0.05	**0.61 ± 0.14**
	120	BSC-I	0.98 ± 0.02	0.42 ± 0.11	0.54 ± 0.05	0.37 ± 0.11
		FCI	0.97 ± 0.03	**0.46 ± 0.12**	0.51 ± 0.04	**0.40 ± 0.11**
50	100	BSC-I	**0.92 ± 0.04**	0.72 ± 0.11	**0.55 ± 0.06**	0.58 ± 0.14
		FCI	0.63 ± 0.04	**0.77 ± 0.11**	0.30 ± 0.04	**0.67 ± 0.15**
	200	BSC-I	**0.98 ± 0.01**	0.66 ± 0.09	**0.58 ± 0.05**	0.56 ± 0.11
		FCI	0.92 ± 0.03	**0.70 ± 0.09**	0.52 ± 0.05	**0.61 ± 0.11**
	300	BSC-I	**1.00 ± 0.01**	0.45 ± 0.06	**0.57 ± 0.03**	0.38 ± 0.06
		FCI	0.98 ± 0.01	**0.49 ± 0.07**	0.55 ± 0.02	**0.42 ± 0.07**
Summary statistics		BSC-I	0.97 ± 0.02	0.62 ± 0.12	0.56 ± 0.09	0.52 ± 0.10
		FCI	0.90 ± 0.11	0.65 ± 0.13	0.49 ± 0.10	0.57 ± 0.11

Table 10: Continuous variable type: Adjacency precision (AP) and recall (AR), and arrowhead precision (AHP) and recall (AHR) results for FCI-BSC (with BSC-I scoring method) and FCI (with Fisher's Z test) when using $N = 5000$ training cases. The numbers after '±' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	AP	AR	AHP	AHR
10	20	BSC-I	**1.00 ± 0.00**	0.95 ± 0.06	0.76 ± 0.24	0.92 ± 0.11
		FCI	0.94 ± 0.07	0.96 ± 0.05	0.74 ± 0.21	0.93 ± 0.09
	40	BSC-I	0.99 ± 0.03	0.72 ± 0.14	0.59 ± 0.19	0.64 ± 0.15
		FCI	0.98 ± 0.04	0.73 ± 0.14	0.58 ± 0.19	0.65 ± 0.15
	60	BSC-I	0.97 ± 0.05	0.57 ± 0.08	**0.45 ± 0.08**	0.52 ± 0.11
		FCI	0.96 ± 0.04	**0.62 ± 0.09**	0.41 ± 0.09	0.57 ± 0.15
20	40	BSC-I	**0.98 ± 0.03**	0.87 ± 0.09	**0.73 ± 0.22**	0.83 ± 0.11
		FCI	0.85 ± 0.07	**0.92 ± 0.06**	0.50 ± 0.13	0.88 ± 0.07
	80	BSC-I	**1.00 ± 0.00**	0.79 ± 0.08	0.64 ± 0.05	0.73 ± 0.12
		FCI	0.97 ± 0.03	**0.81 ± 0.08**	0.59 ± 0.07	**0.76 ± 0.12**
	120	BSC-I	0.99 ± 0.02	0.55 ± 0.08	0.52 ± 0.04	0.49 ± 0.08
		FCI	0.98 ± 0.02	**0.56 ± 0.08**	0.51 ± 0.04	0.50 ± 0.08
50	100	BSC-I	**0.96 ± 0.03**	0.87 ± 0.10	**0.72 ± 0.11**	0.83 ± 0.14
		FCI	0.66 ± 0.05	**0.91 ± 0.10**	0.35 ± 0.07	**0.89 ± 0.11**
	200	BSC-I	**1.00 ± 0.01**	0.77 ± 0.08	**0.67 ± 0.06**	0.70 ± 0.10
		FCI	0.93 ± 0.03	**0.81 ± 0.07**	0.60 ± 0.05	**0.75 ± 0.09**
	300	BSC-I	**1.00 ± 0.01**	0.52 ± 0.08	**0.59 ± 0.04**	0.45 ± 0.09
		FCI	0.98 ± 0.01	**0.54 ± 0.08**	0.57 ± 0.04	**0.47 ± 0.09**
Summary statistics		BSC-I	0.99 ± 0.01	0.73 ± 0.15	0.63 ± 0.10	0.68 ± 0.16
		FCI	0.92 ± 0.10	0.76 ± 0.15	0.54 ± 0.11	0.71 ± 0.16

Table 11: Continuous variable type: Strict SHD (S-SHD), lenient SHD (L-SHD), and adjacency SHD (A-SHD) results for FCI-BSC (with BSC-I scoring method) and FCI (with Fisher's Z test) when using $N = 200$ training cases. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	Added	Deleted	Reoriented	S-SHD	L-SHD	A-SHD
10	20	BSC-I	0.9	3	2.4	6.3	4.3	3.9
		FCI	1	3	2.4	6.4	4.4	4
	40	BSC-I	0.3	8.4	9.6	18.3	11	8.7
		FCI	0.5	8.3	9.8	18.6	11	8.8
	60	BSC-I	0.1	19.7	11.1	30.9	23	19.8
		FCI	0.1	19.5	11	30.6	22.6	19.6
20	40	BSC-I	**1.70**	7.9	6.6	**16.20**	**10.30**	9.6
		FCI	3	7.7	7.5	18.2	11.7	10.7
	80	BSC-I	0.9	18.5	**19.50**	38.9	26.3	19.4
		FCI	1.5	**17.50**	21.3	40.3	27.1	19
	120	BSC-I	0	58	**24.00**	82	66	58
		FCI	0.2	**56.30**	26	82.5	66.6	**56.50**
50	100	BSC-I	**10.40**	21.5	**17.50**	**49.40**	**33.90**	**31.90**
		FCI	21.4	**20.10**	21.4	62.9	44.1	41.5
	200	BSC-I	**3.20**	61.4	**41.10**	**105.70**	**78.20**	64.6
		FCI	8.5	**57.10**	48.1	113.7	81.7	65.6
	300	BSC-I	**1.80**	163.5	**62.70**	**228.00**	192.7	165.3
		FCI	4.1	**157.30**	70.1	231.5	192.6	**161.40**
Summary statistics		BSC-I	2.14	40.21	21.61	63.97	49.52	42.36
		FCI	4.48	38.53	24.18	67.19	51.31	43.01

Table 12: Continuous variable type: Strict SHD (S-SHD), lenient SHD (L-SHD), and adjacency SHD (A-SHD) results for FCI-BSC (with BSC-I scoring method) and FCI (with Fisher's Z test) when using $N = 1000$ training cases. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	Added	Deleted	Reoriented	S-SHD	L-SHD	A-SHD
10	20	BSC-I	0.2	1.9	2.2	4.3	2.5	2.1
		FCI	0.6	1.6	2.3	4.5	2.6	2.2
	40	BSC-I	0.5	10.1	10.7	21.3	13.9	10.6
		FCI	0.9	9.5	11.3	21.7	14.2	10.4
	60	BSC-I	0.1	15.2	14.6	29.9	18.3	15.3
		FCI	0.2	14.8	14.9	29.9	18	15
20	40	BSC-I	**0.70**	5.4	**6.00**	**12.10**	**6.60**	6.1
		FCI	3.5	4.6	8.4	16.5	8.8	8.1
	80	BSC-I	0.3	15.7	21.5	37.5	21.3	16
		FCI	0.8	**14.30**	22.4	37.5	20.8	15.1
	120	BSC-I	0.7	52.7	**28.00**	81.4	60.6	53.4
		FCI	1	**49.80**	30.8	81.6	**59.10**	**50.80**
50	100	BSC-I	**3.50**	15.1	**17.50**	**36.10**	**20.40**	**18.60**
		FCI	23.6	**12.50**	27.8	63.9	38.1	36.1
	200	BSC-I	**1.40**	41.7	**52.90**	**96.00**	56.8	43.1
		FCI	7.1	**36.50**	60.8	104.4	59.1	43.6
	300	BSC-I	**0.40**	133.9	**83.20**	**217.50**	167.3	134.3
		FCI	2.6	**125.20**	93.6	221.4	165.5	**127.80**
Summary statistics		BSC-I	0.87	32.41	26.29	59.57	40.86	33.28
		FCI	4.48	29.87	30.26	64.6	42.91	34.34

Table 13: Continuous variable type: Strict SHD (S-SHD), lenient SHD (L-SHD), and adjacency SHD (A-SHD) results for FCI-BSC (with BSC-I scoring method) and FCI (with Fisher's Z test) when using $N = 5000$ training cases. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	Added	Deleted	Reoriented	S-SHD	L-SHD	A-SHD
10	20	BSC-I	**0.00**	0.5	2.9	3.4	0.9	0.5
		FCI	0.6	0.4	2.8	3.8	1.3	1
	40	BSC-I	0.2	6.5	10	16.7	8	6.7
		FCI	0.4	6.4	10.1	16.9	8.1	6.8
	60	BSC-I	0.5	13.6	**16.00**	30.1	16.9	14.1
		FCI	0.8	**12.30**	17.8	30.9	16.5	13.1
20	40	BSC-I	**0.30**	2.8	**5.00**	**8.10**	**3.20**	**3.10**
		FCI	3.3	**1.70**	10.1	15.1	5.4	5
	80	BSC-I	**0.00**	9.1	**20.80**	29.9	12.1	9.1
		FCI	1.2	**8.00**	22.8	32	12.4	9.2
	120	BSC-I	0.6	37.4	35.9	73.9	52	38
		FCI	1	**36.30**	37.2	74.5	52.1	37.3
50	100	BSC-I	**1.90**	7.5	**12.20**	**21.60**	**10.00**	**9.40**
		FCI	24.4	**5.20**	26.7	56.3	30.9	29.6
	200	BSC-I	**0.50**	28.3	**53.00**	**81.80**	40.5	28.8
		FCI	6.8	**23.60**	58	88.4	44.2	30.4
	300	BSC-I	**0.40**	118.3	**94.60**	213.3	151.4	118.7
		FCI	2.4	**112.80**	102.8	218	153.1	**115.20**
Summary statistics		BSC-I	0.49	24.89	27.82	53.2	32.78	25.38
		FCI	4.54	22.97	32.03	59.54	36	27.51

3.6.1.4 Simulation results for mixed variable type Tables 14, 15, and 16 show the average adjacency P (AP) and R (AR), and arrowhead P (AHP) and R (AHR) results of the FCI-BSC (with BSC-I scoring method)[8] and the FCI (with log-likelihood ratio test that uses the degenerate Gaussian score [Andrews et al., 2019]) algorithms over 10 randomly generated CBNs described in Section 3.6.1, using $N = \{200, 1000, 5000\}$ training instances, respectively. For $N = 200$, FCI-BSC performs better in terms of AP in most of the cases, but AR and AHR are better when using FCI (Table 14). Both methods often have similar performance in terms of AHP. As the sample size increases to $N = \{1000, 5000\}$ training instances, both methods perform better in terms of all these measures, where FCI-BSC almost always performs significantly better in terms of AP and AHP, while FCI always performs significantly better in terms of AR and AHR, while FCI performs better in terms of AR and AHR based on Wilcoxon signed rank test at 5% significance level (Tables 15 and 16).

Tables 17, 18, and 19 show the SHD results of the FCI-BSC (with BSC-I scoring method) and the FCI (with log-likelihood ratio test that uses the degenerate Gaussian score [Andrews et al., 2019]) algorithms over 10 randomly generated CBNs described in Section 3.6.1, using $N = \{200, 1000, 5000\}$ training instances, respectively. As these tables demonstrate, FCI-BSC almost always performs significantly better compared to FCI in terms of the number of added and reoriented edges, while FCI often has significantly fewer deleted edges based on Wilcoxon signed rank test at 5% significance level. Overall, FCI-BSC almost always performs significantly better in terms of S-SHD, L-SHD, and A-SHD measures based on Wilcoxon signed rank test at 5% significance level.

[8]Results using BSC-D and BSC-LD scoring methods are similar to using BSC-I; therefore, we only report the results for BCS-I, which is a simpler and more efficient method.

Table 14: Mixed variable type: Adjacency precision (AP) and recall (AR), and arrowhead precision (AHP) and recall (AHR) results for FCI-BSC (with BSC-I scoring method) and FCI (with log-likelihood ratio test using degenerate Gaussian score) when using $N = 200$ training cases. The numbers after '±' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	AP	AR	AHP	AHR
10	20	BSC-I	0.65 ± 0.40	0.24 ± 0.17	0.12 ± 0.30	0.05 ± 0.12
		FCI	0.50 ± 0.20	**0.44 ± 0.11**	0.13 ± 0.14	0.23 ± 0.24
	40	BSC-I	0.85 ± 0.30	0.25 ± 0.13	0.27 ± 0.29	0.11 ± 0.12
		FCI	0.71 ± 0.14	**0.45 ± 0.13**	0.29 ± 0.11	**0.33 ± 0.17**
	60	BSC-I	0.84 ± 0.29	0.17 ± 0.08	0.33 ± 0.30	0.17 ± 0.28
		FCI	0.78 ± 0.07	**0.36 ± 0.08**	0.25 ± 0.11	0.21 ± 0.09
20	40	BSC-I	**0.67 ± 0.13**	0.28 ± 0.10	0.16 ± 0.18	0.10 ± 0.09
		FCI	0.26 ± 0.07	**0.45 ± 0.16**	0.08 ± 0.04	**0.34 ± 0.18**
	80	BSC-I	**0.85 ± 0.15**	0.17 ± 0.07	0.40 ± 0.27	0.07 ± 0.07
		FCI	0.48 ± 0.05	**0.37 ± 0.07**	0.21 ± 0.04	**0.31 ± 0.07**
	120	BSC-I	**0.88 ± 0.08**	0.12 ± 0.02	**0.40 ± 0.13**	0.07 ± 0.03
		FCI	0.63 ± 0.11	**0.28 ± 0.06**	0.30 ± 0.06	**0.25 ± 0.06**
50	100	BSC-I	**0.58 ± 0.08**	0.24 ± 0.06	**0.17 ± 0.12**	0.07 ± 0.07
		FCI	0.16 ± 0.02	**0.40 ± 0.05**	0.05 ± 0.01	**0.28 ± 0.06**
	200	BSC-I	**0.72 ± 0.08**	0.15 ± 0.04	**0.27 ± 0.13**	0.06 ± 0.03
		FCI	0.26 ± 0.03	**0.29 ± 0.05**	0.11 ± 0.02	**0.23 ± 0.05**
	300	BSC-I	**0.76 ± 0.05**	0.09 ± 0.02	**0.34 ± 0.10**	0.03 ± 0.01
		FCI	0.38 ± 0.04	**0.22 ± 0.03**	0.18 ± 0.02	**0.19 ± 0.03**
Summary statistics		BSC-I	0.76 ± 0.10	0.19 ± 0.06	0.27 ± 0.10	0.08 ± 0.04
		FCI	0.46 ± 0.20	0.36 ± 0.08	0.18 ± 0.09	0.26 ± 0.05

Table 15: Mixed variable type: Adjacency precision (AP) and recall (AR), and arrowhead precision (AHP) and recall (AHR) results for FCI-BSC (with BSC-I scoring method) and FCI (with log-likelihood ratio test using degenerate Gaussian score) when using $N = 1000$ training cases. The numbers after '±' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	AP	AR	AHP	AHR
10	20	BSC-I	**0.97 ± 0.07**	0.52 ± 0.16	0.07 ± 0.14	0.12 ± 0.25
		FCI	0.58 ± 0.14	**0.72 ± 0.15**	0.17 ± 0.14	**0.44 ± 0.33**
	40	BSC-I	**0.99 ± 0.04**	0.34 ± 0.09	0.31 ± 0.17	0.16 ± 0.09
		FCI	0.80 ± 0.10	**0.56 ± 0.09**	0.35 ± 0.08	**0.47 ± 0.11**
	60	BSC-I	**0.99 ± 0.03**	0.28 ± 0.06	0.31 ± 0.16	0.18 ± 0.09
		FCI	0.87 ± 0.06	**0.50 ± 0.06**	0.32 ± 0.08	**0.42 ± 0.10**
20	40	BSC-I	**0.89 ± 0.08**	0.43 ± 0.12	**0.41 ± 0.24**	0.20 ± 0.12
		FCI	0.31 ± 0.04	**0.66 ± 0.11**	0.13 ± 0.03	**0.56 ± 0.11**
	80	BSC-I	**0.98 ± 0.04**	0.32 ± 0.11	**0.44 ± 0.10**	0.19 ± 0.10
		FCI	0.56 ± 0.06	**0.54 ± 0.12**	0.26 ± 0.05	**0.47 ± 0.13**
	120	BSC-I	**0.97 ± 0.04**	0.19 ± 0.05	**0.41 ± 0.07**	0.12 ± 0.05
		FCI	0.71 ± 0.09	**0.42 ± 0.06**	0.34 ± 0.04	**0.37 ± 0.07**
50	100	BSC-I	**0.87 ± 0.05**	0.41 ± 0.06	**0.37 ± 0.12**	0.22 ± 0.09
		FCI	0.17 ± 0.01	**0.63 ± 0.08**	0.06 ± 0.01	**0.55 ± 0.11**
	200	BSC-I	**0.93 ± 0.03**	0.23 ± 0.07	**0.40 ± 0.05**	0.12 ± 0.05
		FCI	0.31 ± 0.04	**0.40 ± 0.08**	0.14 ± 0.02	**0.34 ± 0.07**
	300	BSC-I	**0.93 ± 0.04**	0.15 ± 0.03	**0.44 ± 0.06**	0.09 ± 0.03
		FCI	0.43 ± 0.03	**0.34 ± 0.04**	0.21 ± 0.02	**0.29 ± 0.04**
Summary statistics		BSC-I	0.95 ± 0.04	0.32 ± 0.11	0.35 ± 0.11	0.16 ± 0.04
		FCI	0.53 ± 0.23	0.53 ± 0.12	0.22 ± 0.10	0.43 ± 0.08

Table 16: Mixed variable type: Adjacency precision (AP) and recall (AR), and arrowhead precision (AHP) and recall (AHR) results for FCI-BSC (with BSC-I scoring method) and FCI (with log-likelihood ratio test using degenerate Gaussian score) when using $N = 5000$ training cases. The numbers after '±' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	AP	AR	AHP	AHR
10	20	BSC-I	**1.00 ± 0.00**	0.55 ± 0.11	**0.69 ± 0.40**	0.32 ± 0.18
		FCI	0.49 ± 0.11	**0.67 ± 0.06**	0.21 ± 0.14	**0.50 ± 0.16**
	40	BSC-I	**0.99 ± 0.02**	0.52 ± 0.15	**0.48 ± 0.15**	0.43 ± 0.20
		FCI	0.75 ± 0.05	**0.74 ± 0.13**	0.34 ± 0.06	**0.68 ± 0.16**
	60	BSC-I	**0.98 ± 0.04**	0.39 ± 0.09	0.39 ± 0.08	0.28 ± 0.06
		FCI	0.88 ± 0.04	**0.65 ± 0.06**	0.38 ± 0.05	**0.56 ± 0.11**
20	40	BSC-I	**0.96 ± 0.05**	0.45 ± 0.10	**0.62 ± 0.18**	0.23 ± 0.13
		FCI	0.33 ± 0.06	**0.67 ± 0.11**	0.13 ± 0.05	**0.56 ± 0.18**
	80	BSC-I	**0.98 ± 0.02**	0.41 ± 0.10	**0.48 ± 0.08**	0.30 ± 0.09
		FCI	0.63 ± 0.09	**0.60 ± 0.11**	0.31 ± 0.04	**0.55 ± 0.10**
	120	BSC-I	**0.99 ± 0.02**	0.28 ± 0.07	**0.45 ± 0.05**	0.21 ± 0.06
		FCI	0.73 ± 0.09	**0.54 ± 0.07**	0.33 ± 0.06	**0.47 ± 0.07**
50	100	BSC-I	**0.92 ± 0.03**	0.43 ± 0.08	**0.44 ± 0.08**	0.25 ± 0.10
		FCI	0.16 ± 0.02	**0.67 ± 0.09**	0.06 ± 0.01	**0.57 ± 0.11**
	200	BSC-I	**0.98 ± 0.02**	0.40 ± 0.08	**0.49 ± 0.05**	0.28 ± 0.08
		FCI	0.34 ± 0.02	**0.58 ± 0.11**	0.16 ± 0.02	**0.51 ± 0.11**
	300	BSC-I	**0.96 ± 0.02**	0.26 ± 0.05	**0.48 ± 0.03**	0.18 ± 0.05
		FCI	0.50 ± 0.05	**0.44 ± 0.06**	0.25 ± 0.03	**0.40 ± 0.06**
Summary statistics		BSC-I	0.97 ± 0.02	0.41 ± 0.09	0.50 ± 0.09	0.28 ± 0.07
		FCI	0.53 ± 0.22	0.62 ± 0.08	0.24 ± 0.10	0.53 ± 0.07

Table 17: Mixed variable type: Strict SHD (S-SHD), lenient SHD (L-SHD), and adjacency SHD (A-SHD) results for FCI-BSC (with BSC-I scoring method) and FCI (with log-likelihood ratio test using degenerate Gaussian score) when using $N = 200$ training cases. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	Added	Deleted	Reoriented	S-SHD	L-SHD	A-SHD
10	20	BSC-I	**0.60**	7.3	**1.50**	**9.40**	**8.10**	7.9
		FCI	4.7	**5.30**	3.6	13.6	10.9	10
	40	BSC-I	**0.20**	13.6	**3.10**	**16.90**	15.3	13.8
		FCI	3.4	**10.10**	7.1	20.6	16	13.5
	60	BSC-I	**0.40**	23.1	**3.70**	**27.20**	25.1	23.5
		FCI	2.8	**17.90**	8.7	29.4	23.4	**20.70**
20	40	BSC-I	**2.50**	13.7	**3.00**	**19.20**	**16.40**	**16.20**
		FCI	23	**10.60**	7.7	41.3	33.8	33.6
	80	BSC-I	**1.40**	39.2	**5.50**	**46.10**	**42.40**	**40.60**
		FCI	17.9	**29.80**	15.4	63.1	52.4	47.7
	120	BSC-I	**1.30**	70.5	**7.90**	**79.70**	74.7	71.8
		FCI	13.1	**57.90**	20.7	91.7	79.3	71
50	100	BSC-I	**8.10**	38.4	**7.70**	**54.20**	**47.90**	**46.50**
		FCI	105.3	**30.00**	18.7	154	136.5	135.3
	200	BSC-I	**7.00**	101.7	**14.30**	**123.00**	**114.40**	**108.70**
		FCI	96.4	**85.80**	32.4	214.6	192.3	182.2
	300	BSC-I	**6.00**	208.7	**15.50**	**230.20**	**220.30**	**214.70**
		FCI	80.4	**178.80**	44.1	303.3	277.6	259.2
Summary statistics		BSC-I	3.06	57.36	6.91	67.32	62.73	60.41
		FCI	38.56	47.36	17.6	103.51	91.36	85.91

Table 18: Mixed variable type: Strict SHD (S-SHD), lenient SHD (L-SHD), and adjacency SHD (A-SHD) results for FCI-BSC (with BSC-I scoring method) and FCI (with log-likelihood ratio test using degenerate Gaussian score) when using $N = 1000$ training cases. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	Added	Deleted	Reoriented	S-SHD	L-SHD	A-SHD
10	20	BSC-I	**0.20**	4.3	**3.50**	**8.00**	**4.60**	**4.50**
		FCI	5.2	**2.50**	5.5	13.2	8.2	7.7
	40	BSC-I	**0.10**	13.9	**6.20**	**20.20**	16.4	14
		FCI	3	**9.10**	10.1	22.2	14.7	**12.10**
	60	BSC-I	**0.10**	22	**7.70**	**29.80**	24.1	22.1
		FCI	2.2	**15.10**	14.4	31.7	**21.60**	**17.30**
20	40	BSC-I	**1.00**	10.6	**4.30**	**15.90**	**12.60**	**11.60**
		FCI	26.7	**6.40**	10.4	43.5	34.3	33.1
	80	BSC-I	**0.30**	30.9	**10.20**	**41.40**	**35.90**	**31.20**
		FCI	17.9	**21.40**	20.6	59.9	46.4	39.3
	120	BSC-I	**0.50**	65.6	**12.80**	**78.90**	72.2	66.1
		FCI	13.3	**47.80**	30.6	91.7	73	61.1
50	100	BSC-I	**3.00**	28.1	**11.90**	**43.00**	**33.20**	**31.10**
		FCI	147.1	**17.60**	27.6	192.3	166.6	164.7
	200	BSC-I	**2.30**	107.4	**26.00**	**135.70**	**117.60**	**109.70**
		FCI	121.3	**84.40**	48.8	254.5	220	205.7
	300	BSC-I	**2.70**	202.6	**31.10**	**236.40**	**218.10**	**205.30**
		FCI	105.5	**159.20**	74.1	338.8	295.2	264.7
Summary statistics		BSC-I	1.13	53.93	12.63	67.7	59.41	55.07
		FCI	49.13	40.39	26.9	116.42	97.78	89.52

Table 19: Mixed variable type: Strict SHD (S-SHD), lenient SHD (L-SHD), and adjacency SHD (A-SHD) results for FCI-BSC (with BSC-I scoring method) and FCI (with log-likelihood ratio test using degenerate Gaussian score) when using $N = 5000$ training cases. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	Added	Deleted	Reoriented	S-SHD	L-SHD	A-SHD
10	20	BSC-I	**0.00**	4.2	**2.30**	**6.50**	**4.50**	**4.20**
		FCI	6.7	**3.10**	4.8	14.6	10.3	9.8
	40	BSC-I	**0.10**	9.3	**7.60**	**17.00**	11.8	9.4
		FCI	4.6	**5.20**	12.5	22.3	13.5	9.8
	60	BSC-I	**0.30**	19.6	**11.60**	31.5	23.4	19.9
		FCI	2.9	**11.20**	18.8	32.9	**18.60**	**14.10**
20	40	BSC-I	**0.40**	11.6	**3.30**	**15.30**	**12.60**	**12.00**
		FCI	29.1	**7.00**	11.5	47.6	36.9	36.1
	80	BSC-I	**0.50**	32	**16.90**	**49.40**	**38.50**	**32.50**
		FCI	18.3	**21.70**	28	68	48.4	40
	120	BSC-I	**0.30**	58.9	**18.90**	**78.10**	65.5	59.2
		FCI	15.6	**38.30**	39.7	93.6	67	53.9
50	100	BSC-I	**1.80**	28.7	**11.90**	**42.40**	**32.40**	**30.50**
		FCI	174.8	**16.70**	31.1	222.6	193	191.5
	200	BSC-I	**0.80**	77.2	**38.70**	**116.70**	**89.90**	**78.00**
		FCI	139.4	**55.10**	64.9	259.4	213.5	194.5
	300	BSC-I	**2.30**	178.9	**51.40**	**232.60**	**204.60**	**181.20**
		FCI	105.6	**134.00**	96.5	336.1	280.6	239.6
Summary statistics		BSC-I	0.72	46.71	18.07	65.5	53.69	47.43
		FCI	55.22	32.48	34.2	121.9	97.98	87.7

3.6.2 Simulated data from manually constructed BN models

To further compare FCI-BSC versus FCI, we also simulated data from manually constructed, previously published CBNs, with some variables designated as being latent in order to perform an evaluation on the FCI-BSC method versus the FCI method. In particular, we simulated data from the Alarm [Beinlich et al., 1989], Hailfinder [Abramson et al., 1996], and Hepar II [Onisko, 2003] CBNs, which we obtained from [Scutari, 2010]. Table 20 shows some key characteristics of each CBN. Using these benchmarks is beneficial in several ways. First, they are more likely to represent real-world distributions. Also, we can evaluate the results using the true underlying causal model, which we know by construction; otherwise, it is rare to find known causal models on more than a few variables and associated real, observational data.

Table 20: Information about the Alarm, Hailfinder, and Hepar II CBNs.

CBN Name	Alarm	Hailfinder	Hepar II
Domain	Medicine	Weather	Medicine
Number of nodes	37	56	70
Number of edges	46	66	123
Average indegree	1.24	1.18	1.76
Average degree	2.49	2.36	3.51
Number of parameters	509	2656	1453

To simulate data from each of these CBNs, we randomly designated $L = 20\%$ of the confounder variables to be latent, which means data about those variables were not provided to the discovery algorithms. Then, we performed the following steps:

1. We used each CBN model to simulate a training dataset D with $N = \{200, 1000, 5000\}$ training samples.
2. We used the training dataset D generated in step 1 to learn a PAG structure $\mathcal{P}_F$ using the FCI algorithm with a chi-squared test of independence, which is a standard test and approach. We used $\alpha = 0.05$, which is a common alpha value used with FCI.
3. We also used the training dataset D generated in step 1 to learn a set of PAG structures $\boldsymbol{\mathcal{P}}$

using the FCI-BSC algorithm with the BSC test of independence for discrete variables. In applying the FCI-BSC algorithm, we sampled 100 PAG models, according to the method described in Section 3.4 (i.e., $s = 100$ in Algorithm 5). We scored the PAGs using the three PAG scoring methods BSC-I, BSC-D, and BSC-LD, to obtain the PAG with the highest posterior probability by each scoring method, which we denote as $\mathcal{P}_I$, $\mathcal{P}_D$, and $\mathcal{P}_{LD}$, respectively. For the BSC-D and BSC-LD methods, we bootstrapped the data 500 times (i.e., $bs = 500$ in Algorithm 6) to create the empirical data.

4. Finally, we computed the evaluation measures described in Section 3.6.1.1 to compare the structure recovery performance of FCI-BSC versus FCI. To do so, we compared $\mathcal{P}_I$, $\mathcal{P}_D$, and $\mathcal{P}_{LD}$ (which are the most probable PAGs found by the BSC-I, BSC-D, and BSC-LD methods) and $\mathcal{P}_F$ (which is found by FCI) to the PAG $\mathcal{P}_{truth}$ that is consistent with the data-generating CBN. To obtain $\mathcal{P}_{truth}$, we used an independence oracle that has access to the data-generating model and latent variables. $\mathcal{P}_{truth}$ represents all the causal relationships that can be learned about a CBN in the large sample limit when assuming Markov, faithfulness, and using correct independence tests that are applied to (infinite) observational data on the measured variables in a CBN.

For each CBN, the analyses in steps 1 to 4 were repeated 10 times, each time randomly sampling a different dataset, and the performance measures were averaged.

3.6.2.1 Simulation results Table 21 shows the experimental results on Alarm, Hailfinder, Hepar II CBNs with $L = 20\%$ latent variables and $N = \{200, 1000, 5000\}$ training cases. This table shows that all scoring methods (BSC-I, BSC-D, and BSC-LD) resulted in a similar performance of the FCI-BSC algorithm. Table 21a shows that FCI-BSC always improves adjacency precision (AP) measure, and also has fewer number of added edges for Alarm network than does FCI, while both methods almost always perform similarly in terms of other measures. For the Hailfinder network, we observed that FCI-BSC has significant improvements in AP and AHP, as well as all SHD measures when the sample size is $N = \{1000, 5000\}$ based on Wilcoxon signed rank test at 5% significance level (Table 21b). However, FCI performs better in terms of AR and AHR for Hailfinder with $N = 200$ samples; both methods perform closely in terms of other measures. Similar results were obtained on

Table 21: Experimental results for FCI-BSC (with BSC-I, BSC-D, and BSC-LD scoring methods) and FCI (with a chi-squared test). AP, AR, AHP, AHR, S-SHD, L-SHD, and A-SHD denote adjacency P and R, arrowhead P and R, strict, lenient, and adjacency SHD. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

(a) Experimental results on Alarm network.

# Cases	Method	AP	AR	AHP	AHR	Added	Deleted	Reoriented	S-SHD	L-SHD	A-SHD
200	BSC-I	**0.96 ± 0.03**	0.47 ± 0.04	0.66 ± 0.13	0.26 ± 0.11	**0.9**	22.2	11.3	34.4	25.2	**23.1**
	BSC-D	**0.95 ± 0.03**	0.47 ± 0.04	0.66 ± 0.14	0.26 ± 0.11	**1.0**	22.4	11.1	34.5	25.5	**23.4**
	BSC-LD	**0.96 ± 0.03**	0.47 ± 0.04	0.66 ± 0.13	0.26 ± 0.11	**0.9**	22.2	11.3	34.4	25.2	**23.1**
	FCI	0.91 ± 0.04	0.46 ± 0.03	0.76 ± 0.18	0.29 ± 0.08	2	22.5	**9.7**	34.2	25.7	24.5
1000	BSC-I	**0.99 ± 0.02**	0.66 ± 0.03	0.64 ± 0.06	0.42 ± 0.08	**0.4**	14.4	15.5	30.3	17.4	14.8
	BSC-D	**0.99 ± 0.02**	0.66 ± 0.03	0.64 ± 0.06	0.42 ± 0.08	**0.4**	14.4	15.5	30.3	17.4	14.8
	BSC-LD	**0.99 ± 0.02**	0.66 ± 0.03	0.64 ± 0.06	0.42 ± 0.08	**0.4**	14.4	15.5	30.3	17.4	14.8
	FCI	0.95 ± 0.04	0.65 ± 0.02	0.72 ± 0.15	0.43 ± 0.05	1.6	14.9	13.7	30.2	18.3	16.5
5000	BSC-I	**1.00 ± 0.01**	0.78 ± 0.02	0.76 ± 0.08	0.55 ± 0.07	**0.1**	9.4	12.2	21.7	12.3	9.5
	BSC-D	**1.00 ± 0.01**	0.78 ± 0.02	0.76 ± 0.08	0.55 ± 0.07	**0.1**	9.4	12.2	21.7	12.3	9.5
	BSC-LD	**1.00 ± 0.01**	0.78 ± 0.02	0.76 ± 0.08	0.55 ± 0.07	**0.1**	9.4	12.2	21.7	12.3	9.5
	FCI	0.96 ± 0.03	0.79 ± 0.02	0.77 ± 0.08	0.61 ± 0.08	1.3	**8.7**	11.3	21.3	11.8	10

(b) Experimental results on Hailfinder network.

# Cases	Method	AP	AR	AHP	AHR	Added	Deleted	Reoriented	S-SHD	L-SHD	A-SHD
200	BSC-I	**0.82 ± 0.11**	0.18 ± 0.02	0.66 ± 0.16	0.12 ± 0.04	**2.7**	53.2	7.0	62.9	58.4	55.9
	BSC-D	**0.82 ± 0.11**	0.18 ± 0.02	0.66 ± 0.16	0.12 ± 0.04	**2.7**	53.2	7.0	62.9	58.4	55.9
	BSC-LD	**0.82 ± 0.11**	0.18 ± 0.02	0.66 ± 0.16	0.12 ± 0.04	**2.7**	53.2	7.0	62.9	58.4	55.9
	FCI	0.69 ± 0.04	**0.27 ± 0.02**	0.56 ± 0.05	**0.29 ± 0.04**	7.9	**47.4**	7.3	62.6	57.7	55.3
1000	BSC-I	**0.82 ± 0.05**	0.31 ± 0.01	**0.88 ± 0.10**	0.33 ± 0.01	**4.4**	45.2	**5.9**	**55.5**	**51.2**	**49.6**
	BSC-D	**0.82 ± 0.05**	0.31 ± 0.01	**0.88 ± 0.10**	0.33 ± 0.01	**4.4**	45.2	**5.9**	**55.5**	**51.2**	**49.6**
	BSC-LD	**0.82 ± 0.05**	0.31 ± 0.01	**0.88 ± 0.10**	0.33 ± 0.01	**4.4**	45.2	**5.9**	**55.5**	**51.2**	**49.6**
	FCI	0.48 ± 0.04	0.31 ± 0.01	0.34 ± 0.05	0.34 ± 0.02	21.3	45	7.9	74.2	67.5	66.3
5000	BSC-I	**0.69 ± 0.07**	0.30 ± 0.01	**0.69 ± 0.14**	0.33 ± 0.01	**8.8**	45.5	6.3	**60.6**	**55.1**	**54.3**
	BSC-D	**0.69 ± 0.07**	0.30 ± 0.01	**0.66 ± 0.15**	0.33 ± 0.01	**9.2**	45.5	6.2	**60.9**	**55.5**	**54.7**
	BSC-LD	**0.69 ± 0.07**	0.30 ± 0.01	**0.66 ± 0.15**	0.33 ± 0.01	**9.2**	45.5	6.2	**60.9**	**55.5**	**54.7**
	FCI	0.31 ± 0.01	**0.34 ± 0.01**	0.20 ± 0.01	**0.40 ± 0.02**	48.2	**43.2**	7.1	98.5	92.3	91.4

(c) Experimental results on Hepar II network.

# Cases	Method	AP	AR	AHP	AHR	Added	Deleted	Reoriented	S-SHD	L-SHD	A-SHD
200	BSC-I	**0.69 ± 0.11**	0.03 ± 0.01	0.32 ± 0.29	0.01 ± 0.00	**3.9**	260.4	**6.7**	**271.0**	**266.0**	**264.3**
	BSC-D	**0.70 ± 0.12**	0.03 ± 0.01	0.45 ± 0.33	0.01 ± 0.00	**3.7**	260.4	**6.5**	**270.6**	**265.4**	**264.1**
	BSC-LD	**0.69 ± 0.11**	0.03 ± 0.01	0.36 ± 0.28	0.01 ± 0.00	**3.9**	260.5	**6.6**	**271.0**	**266.0**	**264.4**
	FCI	0.49 ± 0.05	**0.06 ± 0.01**	0.21 ± 0.04	**0.03 ± 0.01**	16.4	**252.9**	12.8	282.1	275.0	269.3
1000	BSC-I	**0.90 ± 0.06**	0.06 ± 0.00	**0.37 ± 0.15**	0.01 ± 0.00	**1.8**	253.7	**11.2**	**266.7**	**260.8**	255.5
	BSC-D	**0.90 ± 0.06**	0.06 ± 0.00	**0.37 ± 0.15**	0.01 ± 0.00	**1.8**	253.7	**11.2**	**266.7**	**260.8**	255.5
	BSC-LD	**0.90 ± 0.06**	0.06 ± 0.00	**0.37 ± 0.15**	0.01 ± 0.00	**1.8**	253.7	**11.2**	**266.7**	**260.8**	255.5
	FCI	0.70 ± 0.04	**0.10 ± 0.01**	0.27 ± 0.04	**0.05 ± 0.01**	12.2	**241.1**	23.4	276.7	263.9	**253.3**
5000	BSC-I	**0.98 ± 0.02**	0.10 ± 0.00	**0.37 ± 0.05**	0.04 ± 0.01	**0.6**	241.6	**20.9**	**263.1**	251.0	242.2
	BSC-D	**0.98 ± 0.02**	0.10 ± 0.00	**0.37 ± 0.05**	0.04 ± 0.01	**0.6**	241.6	**20.9**	**263.1**	251.0	242.2
	BSC-LD	**0.98 ± 0.02**	0.10 ± 0.00	**0.37 ± 0.05**	0.04 ± 0.01	**0.6**	241.6	**20.9**	**263.1**	251.0	242.2
	FCI	0.83 ± 0.04	**0.15 ± 0.01**	0.33 ± 0.06	**0.08 ± 0.02**	8.4	**227.7**	33.8	269.9	249.5	**236.1**

Hepar II network: FCI-BSC performs significantly better in terms of AP, AHP, and S-SHD, while FCI's performance in terms of AR and AHR is significantly better based on Wilcoxon signed rank test at 5% significance level (Table 21c). We also observed that both methods have low recall for the networks with more parameters and denser structures (i.e., Hailfinder and Hepar II).

We observed that using BSC-I, BSC-D, and BSC-LD scoring methods often result in different scores for the sampled PAGs; however, the ordering of the PAGs according to their scores is almost always the same. For example, for a BN with 20 discrete variables and 80 edges, when using $N = 200$ training samples, the scores for the top-ranked PAGs using BSC-I, BSC-D, and BSC-LD are -2278.45, -2243.56, and -2252.70 (in log scale), respectively. We conjecture that the performance of BSC-I is analogous to a naive Bayes classifier, which often performs classification well, even though it can be highly miscalibrated due to its universal assumption of conditional independence.

3.7 Summary and Discussion

In this chapter, we introduced a general approach for Bayesian scoring of constraints (BSC) that was then applied to a constraint-based method (e.g., FCI) to learn PAG structures; we call this method FCI-BSC. This method can generate multiple PAGs and quantifies the PAGs by their posterior probabilities. In contrast, a constraint-based method that uses a frequentist statistical test (e.g., FCI with a chi-squared test) outputs a single PAG structure and does not provide a score of the output PAG structure. We implemented and experimentally evaluated three methods for scoring PAGs called BSC-I, BSC-D, and BSC-LD. Using simulated data from randomly generated BNs and from manually constructed BNs, we compared these methods to a method that applies the FCI algorithm using frequentist tests of independence.

The empirical results we obtained on simulated data from randomly generated CBNs indicate that for CBNs that contain discrete variables, FCI-BSC performs similar to FCI, especially for smaller sample sizes (i.e., $N = 200$). For the CBNs that contain continuous

and mixed variables, FCI-BSC almost always performs better in terms of adjacency and arrowhead P (AP and AHP), while FCI performs better in terms of adjacency and arrowhead R (AR, AHR). In terms of SHD, we found that FCI-BSC performs better in terms of added and reoriented edges, and overall SHD measures, while FCI has fewer deleted edges. We also observed similar performance results on simulated data from manually constructed CBNs. Overall, the results indicate that the FCI-BSC method tends to be more accurate than FCI in predicting and orienting edges; these results partially support our first hypothesis that is given in Section 1.1, which states that the BSC method will perform CBN structure learning better than a method that uses frequentist statistical tests in terms of discrimination.

4.0 Instance-specific CBN Structure Learning Assuming Causal Sufficiency

Almost all CBN structure learning algorithms that have been developed to date learn a DAG (or equivalence class of DAGs) that encodes the causal relationships that are shared by a population of instances; we call such a model a *population-wide CBN model*. A contrasting paradigm is to learn a CBN that is specific to a particular instance (e.g., a patient); we call such a model an *instance-specific CBN model*. Instance-specific CBN learning is appropriate in domains where the instances may have varying causal structures. For example, a cancerous tumor (or any other complex biological process) in a current patient can be considered as a composite of causal mechanisms. Each of these individual causal mechanisms may appear relatively commonly in other patient tumors, but the particular combination of mechanisms in the tumor of the current patient is unique. Therefore, it is problematic to try to learn the unique set of causal mechanisms for each possible patient by learning a single, population-wide CBN; such a CBN would at best recover the more common mechanisms operating in a population of tumors, but not the unique (or at least rare) combination of causal mechanisms in each tumor. As an alternative approach, we explore learning the joint set of mechanisms for the current patient from the features of that patient and from a training set of data on many other patients. We use the features of the current patient to help find the composite set of mechanisms that are scattered among the patients in the training set. In this chapter, I introduce an instance-specific CBN structure learning method by using CBNs that represent context-specific independence (CSI) (see Section 2.1.4) in order to include instance-specific information in the CBN models.

In the remainder of this chapter, I first review the existing instance-specific modeling methods in Sections 4.1. Then in Sections 4.2 and 4.3, I explain a state-of-the-art score-based CBN structure learning method, called greedy equivalence search (GES), and how to derive a score for a CBN using a score-based method. I introduce an instance-specific score-based CBN structure learning algorithm, called IGES [Jabbari et al., 2018], in Section 4.4. In Section 4.5, I give a quantitative assessment of the IGES method using simulated and real-world biomedical datasets. Finally, Section 4.6 concludes this chapter.

4.1 Related Work

In this section, I review relevant literature to instance-specific modeling in two parts. The first part is related to instance-specific CBN structure learning, which is discussed in Section 4.1.1. The second part is related to instance-specific predictive modeling in machine learning, which is discussed in Section 4.1.2.

4.1.1 Instance-specific causal Bayesian network structure learning

As mentioned earlier in Section 2.1.4, the notion of context-specific independence (CSI) was introduced by [Boutilier et al., 1996] to capture independence relationships that hold between the parents and a child node in a CBN in certain contexts (i.e., when the parent variables take on particular values). In general, these types of independencies cannot be captured completely in the structure of standard CBNs, wherein the CBN structure is invariant to CSI relationships. In this dissertation, I use CBNs that include CSI structures in order to generate instance-specific information in CBN models.

Several greedy search algorithms have been developed to learn CSI structures in Bayesian networks. A number of these methods use structured representations of conditional probability tables (CPTs) to capture CSI relationships, rather than representing them explicitly in the graph structure. [Friedman and Goldszmidt, 1998] introduced a method that uses tree-structured CPTs to partition the outcome space of the parents of a variable to learn the regularities in the CPTs, which correspond to local CSI structures. Then, they incorporated the tree-structured CPTs into a CBN structure search algorithm using a minimum description length (MDL) score. Similarly, [Chickering et al., 1997] proposed using decision-graph CPTs that can represent a richer set of independence relationships, compared to tree-structured CPTs. [Chickering et al., 1997] also developed a Bayesian score to evaluate the posterior probability of Bayesian networks that contain decision-graph CPTs. This score is applied along with a greedy search algorithm to learn a global CBN structure over all variables in which the relationship between each node and its parents is represented using a decision graph. Recently, [Zou et al., 2017] proposed an ordering-based algorithm to learn

local structures using Lasso regression [Tibshirani, 1996] on linear combinations of Boolean functions, where linear combinations of Boolean functions define the interactions among parents of each variable.

There are other methods that explicitly represent CSI relationships in the graph structure using, for example, Bayesian multinets [Geiger and Heckerman, 1996]. Recently, [Pensar et al., 2015] introduced a method to label the edges of a DAG to encode CSI structures; such graphs are called *labeled directed acyclic graphs* (*LDAGs*). In LDAGs, the edges of a Bayesian network are labeled to encode local CSI structures, where an edge can be removed from the DAG if a CSI relationship exists. [Pensar et al., 2015] also proposed an LDAG-based Bayesian score and MCMC search to learn an LDAG structure. [Hyttinen et al., 2018] introduced a constraint-based algorithm and an exact score-based method for learning LDAGs. Also, [Corander et al., 2019] developed a variant of conditional independence logic to formalize CSI statements in LDAGs using first-order logic. [Oates et al., 2016] proposed a method that uses integer linear programming to learn multiple DAGs from multiple units of data, where each unit contains a set of data cases. Recently, [Huang et al., 2019] developed an algorithm, called the specific and shared causal model (SSCM), that utilizes the differences and similarities in heterogeneous (and non-stationary) data to learn a causal model that is shared across the population and also a specific causal model for each individual assuming that multiple samples are observed for each individual.

The methods mentioned above try to capture all possible local structures in a single model, which has several downsides. First, doing so adds to the computational complexity of the CBN structure learning task, which is already an NP-hard problem [Chickering, 1996]. For example in the case of LDAGs, searching over the space of possible labels for edges results in a substantially larger search space than the already superexponential space of possible DAGs. Second, none of the methods learns a model that is specialized to a given test instance (e.g., a given patient), which is one of the main goals and novel contributions of the current dissertation. Doing so has two advantages. First and foremost, the learned causal model is specific to the current instance. Such a tailored model is likely to be more comprehensible to the user, because it includes only the parents of each node that are found to be relevant to

the current instance. We dynamically search to define the clusters of cases associated with the test instance T. Importantly, this search occurs at the node level, not at the DAG level. Second, given that we seek an instance-specific model, searching for it directly is generally much more efficient than is searching for all (or at least many) possible instance-specific models and then choosing the one that matches the current test instance.

[Liu et al., 2016] introduced a method to learn instance-specific networks. It first uses a dataset to build a reference network using Pearson correlation coefficients. Then, it learns a perturbed network by adding a single test sample to the original data. Finally, it obtains the differential network between the reference and perturbed networks to characterize the specific features of the test sample. This method does not learn a causal model, rather, it constructs a correlation model. Additionally, this method is only effective for very small sample sizes since the removal of a single sample may not result in changes in the reference network versus the perturbed networks. In other related work, [Cai et al., 2019] developed a method to learn tumor-specific causal models from data; this is the closest work to the IGES method. However, that method is limited to searching over bipartite causal graphs on binary variables in which one partition contains causes and the other contains effects. Also, the method assumes there is one and only one cause for each effect. Both assumptions are reasonable for that application, but restrict generality. The IGES method is able to learn unrestricted, instance-specific CBNs.

In this chapter, I describe a general, fully Bayesian approach for learning unrestricted instance-specific CBNs on discrete variables. This method searches the space of CBNs to build a model that is specific to an instance T by guiding the search based on T's attributes. We hypothesize that such an instance-specific learning approach will model the causal relationships for T better than does a population-wide one, in terms of discrimination measures.

4.1.2 Instance-specific methods in machine learning

Most machine learning methods for predicting outcomes construct a single model M from training data. M is then applied to predict outcomes in future instances. We refer to such a model as a *population-wide model* because it predicts outcomes for a future population

of instances. It may be difficult for population-wide models to perform well in domains in which instances are highly heterogeneous. In such domains, a reasonable approach is to learn a model that is tailored to a particular instance (e.g., a patient), which we refer to as an *instance-specific model*. An instance-specific approach builds a model M_T for a given instance T from the features that we know about T (e.g., clinical and molecular features) and from a training set of data on many other instances. It then uses M_T to predict the outcome for T. This procedure repeats for each instance that is to be predicted in the population. In this section, I review a representative set of prior work on instance-specific (also known as instance-based) machine learning methods. In these methods, a specific model or parameterization is learned for a given instance (e.g., a data sample) based on the features of the given instance (i.e., variable-value pairs).

The k-nearest neighbor (kNN) method is a canonical instance-specific method. This approach uses a similarity metric (e.g., Euclidean distance) to identify the k most similar training cases to a given test case; it then predicts the target variable of the test case by computing some function (e.g., the average or a majority vote) of the k selected nearest neighbors. One variation of kNN is the weighted kNN algorithm, in which the k most similar cases are weighted according to their similarity to the test case (i.e., assigning greater weights to closer cases) [Dasarathy, 1991]. Another extension of kNN is locally weighted regression [Cleveland, 1979, Cleveland and Devlin, 1988]. This method selects the nearby training cases to the test case; it then fits a surface to those cases using a distance-weighted regression model.

[Zheng and Webb, 2000] introduced a lazy Bayesian rule learning (LBR) method to learn a model that is specific to a test case. In particular, given a test instance, an LBR rule consists of two components: (1) an antecedent that is a conjunction of the variable-value pairs that are present in the test instance; (2) a consequent that is a local naïve Bayes classifier in which the target variable is the parent of the variables that do not appear in the antecedent. The model parameters are estimated in a greedy step-forward search. At each step, the variable that reduces the error rate the most is removed from the local naive Bayes classifier and added it to the antecedent; the search stops when the error rate is not improved anymore. The model is then applied to the test case to predict the target value.

[Visweswaran and Cooper, 2010] developed a two-stage instance-specific Markov Blanket (ISMB) algorithm that searches over the space of Markov blankets (MB) of the target variable by utilizing the features of a given test instance. ISMB finds MBs that are optimized to improve the prediction for each specific test instance. In particular, for a given test instance, the ISMB algorithm first uses a greedy hill-climbing search to find a set of MBs that best fit the training data. Then, it greedily adds single edges to the MB structures from the previous step, if doing so improves the prediction of a given test instance. This algorithm uses a selective Bayesian model averaging method to predict the target variable over a set of MB structures.

[Ferreira et al., 2013] developed two patient-specific decision path (PSDP) algorithms using two variable selection criteria: balanced accuracy and information gain. A PSDP algorithm learns a decision path tailored to the features available for a specific test instance. A decision path is a conjunction of features that are present in a given test instance and a leaf node that contains the probability distribution of the target variable. The PSDP algorithms include (1) PSDP-BA that uses balanced accuracy (BA) to decide which variable is selected for the decision path, and (2) PSDP-IG that uses information gain (IG) to select path variables. Compared to a population-wide decision tree, a PSDP is a simpler model as it consists of only a single path; also, since a PSDP model is tailored to the patient at hand, it has the potential to be more accurate for that specific patient. The results showed that these patient-specific methods outperform the population-wide model on AUROC but have similar performance on balanced accuracy.

Recently, [Lengerich et al., 2018] introduced an instance-specific regression model that learns a specific set of parameters for each test instance, with no a priori knowledge of relationships between data samples. Instead, they used an exogenous set of covariates (e.g., clinical variables can be used as covariates when modeling genomic data), in addition to the variables they use in the regression model; the idea is that the similarity between instance-specific parameters is related to the similarity between the covariates. Accordingly, they developed a distance-matching regularizer to regularize instance-specific parameters by assuming that similarity in parameters corresponds to the similarity in covariates [Lengerich et al., 2018]). Later, [Lengerich et al., 2019] developed an extension of the

instance-specific regression model by using a low-rank latent representation of the regression parameters.

Several studies have developed patient-specific models from patient time-series data. [Qu and Gotman, 1997, Shoeb et al., 2004] proposed methods that use each patient's time series data separately to develop patient-specific models, while ignoring the data from the rest of the population. Other studies have proposed methods that take into account the population data in addition to the time-series data about the patient at hand. For example, [Sheiner et al., 1979] take advantage of population data to estimate patient-specific parameters at the initial time when no measurements are available for the patient yet; then, they are updated with patient data, as they become available, to make the model more patient-specific. Another example is the use of hierarchical models with multiple levels of parameters including population and individual level parameters [Schulam and Saria, 2015, Schulam et al., 2015, Schulam and Saria, 2016]. [Schulam and Saria, 2015] proposed a hierarchical probabilistic graphical model, called Latent Trajectory Model (LTM), to predict patient-specific disease trajectories for patients with complex and chronic diseases. This model contains three levels: population, sub-population, and individual. At the individual level, patient-specific models are learned while sharing statistical power among individuals through higher-level parameters. The individual-level parameters are updated dynamically at the prediction time using Bayesian inference. We investigate atemporal causal models in this dissertation.

The instance-specific models reviewed above are different from the instance-specific CBN learning methods that are introduced in this dissertation in several ways. First, our methods model causal relationships among variables while the above-mentioned methods only perform predictive modeling. Second, we cluster the training data at the variable-level to learn the causes (i.e., parents) of each variable in a given test case; therefore, we dynamically perform clustering during the CBN search. However, some of the previous methods consider the complete set of variables to find cases that are similar to a given test case.

4.2 Overview of Greedy Equivalence Search (GES)

Greedy Equivalence Search (GES) [Chickering, 2002] is a state-of-the-art method for learning a CBN structure from observational data. GES identifies a CBN structure by searching over Markov equivalence classes of DAGs. As described in Section 2.1.1, the Markov equivalence class of DAGs represents a set of DAGs that have the same d-separation properties and are statistically indistinguishable. *A completed partially directed acyclic graph* (CPDAG), also known as *pattern*, represents the Markov equivalence class of DAGs. A pattern is a mixed graph that contains both directed and undirected edges. This section provides an overview of GES and the Bayesian Dirichlet equivalent uniform (BDeu) score [Heckerman, 1998], which we can use together with GES to learn a CBN structure from data.

The GES algorithm is a two-phase score-based method that includes a forward equivalence search (FES) and backward equivalence search (BES) as follows. Let $\mathcal{G}$ be the current pattern during the search. Also, let $\boldsymbol{\mathcal{P}}^{+}(\mathcal{G})$ represent the set of patterns that are generated by adding a single edge to $\mathcal{G}$ for each legal edge addition during the FES [Chickering, 2002, Chickering, 1995], and $\boldsymbol{\mathcal{P}}^{-}(\mathcal{G})$ be the set of patterns that are obtained by deleting each single edge from $\mathcal{G}$ during the BES. The forward phase of GES starts with an empty graph (i.e., $\mathcal{G} = \emptyset$) and replaces the current state with the pattern in $\boldsymbol{\mathcal{P}}^{+}(\mathcal{G})$ that has the highest score. It continues this phase until no further score increase can be achieved. The backward phase starts from the local maximum achieved by the forward phase and performs a backward search by replacing $\mathcal{G}$ with the highest scoring pattern in $\boldsymbol{\mathcal{P}}^{-}(\mathcal{G})$. It stops when it reaches a local maximum. Algorithm 9 provides high-level pseudo-code for GES. Assuming i.i.d sampling, causal sufficiency, the Markov condition, the faithfulness condition, and a locally consistent score, it has been proven that in the large sample limit the GES algorithm learns a pattern that represents the data-generating CBN [Chickering, 2002, Chickering and Meek, 2015, Chickering, 2020]. In this dissertation, we use an efficient implementation of GES called Fast GES (FGES) [Ramsey et al., 2017].

Algorithm 9 GES(D)

Input: a dataset D

Output: a population-wide model $\mathcal{G}_{PW}$

1: $\mathcal{G}_{PW} = \text{FES}(D)$

2: $\mathcal{G}_{PW} = \text{BES}(D, \mathcal{G}_{PW})$

3: return $\mathcal{G}_{PW}$

Since each step in GES (either during the FES or BES) involves a single edge modification (i.e., addition or deletion), GES requires a node-wise decomposable scoring function to locally re-score the effect of the edge modification applied to a single node given its parents. The Bayesian information criterion (BIC) score [Schwarz, 1978] is often used to learn a CBN structure when variables follow a Gaussian distribution and the BDeu score [Heckerman, 1998] is often used for discrete variables, although other scores are possible. In the following section, I review the BDeu score since we concentrate on using discrete variables in this chapter.

4.3 Scoring Bayesian Networks

A Bayesian approach for learning a CBN structure involves searching for a structure with a high posterior probability on a given dataset. Let D be a dataset containing n discrete variables $\boldsymbol{V} = \{X_1, X_2, ..., X_n\}$, where each variable X_i can take r_i values and its parents $\boldsymbol{Pa}(X_i)$ can take q_i distinct instantiations. Also, let $\mathcal{G}$ be the structure we wish to score. According to Bayes' theorem, the posterior probability of graph $\mathcal{G}$ given data D is as follows:

$$P(\mathcal{G}|D) = \frac{P(\mathcal{G}) \cdot P(D|\mathcal{G})}{P(D)}\,, \tag{4.1}$$

where $P(\mathcal{G})$ is the structure prior, $P(D|\mathcal{G})$ is the marginal likelihood of the data, and $P(D)$ is the probability of the data. Since $P(D)$ is a normalization constant and independent of the model, we define the score of model G as follows:

$$\text{Score}(\mathcal{G}, D) = P(\mathcal{G}) \cdot P(D|\mathcal{G})\,, \tag{4.2}$$

where we can compute $P(D|\mathcal{G})$ by integrating over all unknown parameters θ as follows:

$$P(D|\mathcal{G}) = \int_{\theta} P(D|\mathcal{G},\theta) \cdot P(\theta|\mathcal{G})\ d\theta\,. \tag{4.3}$$

The marginal likelihood of the data has a closed-form solution called the Bayesian Dirichlet (BD) score under the following assumptions: (1) the data are discrete, (2) the data are complete (i.e., there are no missing values in D), (3) the parameters are mutually independent, (5) the parameters are modular (i.e., the distributions for parameters of a variable X_i depend only on the local structure of X_i in the Bayesian network, namely, X_i and its parents $\boldsymbol{Pa}(X_i)$), and (5) the parameter priors follow Dirichlet distributions. The BD score is as follows [Cooper and Herskovits, 1992, Heckerman et al., 1995]:

$$P(D|G) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \cdot \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}\,, \tag{4.4}$$

where the first product is over all n variables, the second product is over the q_i parent instantiations of variable i, and the third product is over all r_i values of variable X_i. The term N_{ijk} is the number of cases in D in which variable $X_i = k$ and its parent $\boldsymbol{Pa}(X_i) = j$; also, $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. The term α_{ijk} is a Dirichlet prior parameter that may be interpreted as representing "pseudo-counts" and $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$. The pseudo-counts associated with an event e (or a conditional event) being modeled express prior belief in terms of the number of pseudo counts that the event would have needed to have occurred in the past to yield the strength of current prior belief about e. We may define the pseudo-counts to be uniformly distributed, in which every state of the joint space is equally likely. By incorporating the uniform parameter priors, Equation (4.4) represents the so-called *BDeu score* [Heckerman, 1998]. The uniform parameter priors are formulated as follows:

$$\alpha_{ijk} = \frac{\alpha}{r_i \cdot q_i}\,, \tag{4.5}$$

where α is a positive constant called the prior equivalent sample size (PESS). The BDeu score described here is a modular score that is decomposable at node level and is also score equivalent, as required by the GES algorithm.

4.4 Instance-Specific GES (IGES)

In this section, I describe a novel algorithm called instance-specific GES (IGES) that takes as input a set D of training instances and an instance $T = \{X_1 = x_1, X_2 = x_2, ..., X_n = x_n\}$ that may not be in D, and it returns as output a CBN structure $\mathcal{G}_{IS}$ for instance T and a (often different) CBN structure $\mathcal{G}_{PW}$ for the remaining instances in D. The goal of IGES is to find causal structures $\mathcal{G}_{IS}$ and $\mathcal{G}_{PW}$ that maximize $P(\mathcal{G}_{IS}, \mathcal{G}_{PW}|D, T)$ by deriving $P(D|T, \mathcal{G}_{IS}, \mathcal{G}_{PW})$ and $P(\mathcal{G}_{IS}, \mathcal{G}_{PW})$. Since finding a global optimum for $P(\mathcal{G}_{IS}, \mathcal{G}_{PW}|D, T)$ is generally not computationally tractable, IGES performs GES-style greedy search.

IGES operates in two phases. The first phase uses GES (as described in Section 4.2) with the BDeu score to find $\mathcal{G}_{PW}$ given D. GES uses heuristic search that seeks to find the $\mathcal{G}_{PW}$ that optimizes $P(\mathcal{G}_{PW}|D)$. The second phase uses GES with a novel, instance-specific Bayesian score called the IS-Score (see below) to find the instance-specific structure $\mathcal{G}_{IS}$ given D, T, and $\mathcal{G}_{PW}$; we use the name GES2 to denote this application of GES. GES2 uses heuristic search that seeks to find $\mathcal{G}_{IS}$ that optimizes $P(\mathcal{G}_{IS}|D, T, \mathcal{G}_{PW})$. Algorithm 10 shows the high-level procedure of the IGES method. The order of the computational complexity of IGES is the same as that of GES, since it runs the GES algorithm 2 times.

Algorithm 10 IGES(D, T)

Input: a dataset D, a test instance T

Output: an instance-specific model $\mathcal{G}_{IS}$ and a population-wide model $\mathcal{G}_{PW}$

1: $\mathcal{G}_{PW}$ = GES(D)
2: $\mathcal{G}_{IS}$ = GES2(D, T, $\mathcal{G}_{PW}$)
3: return $\mathcal{G}_{IS}$ and $\mathcal{G}_{PW}$

GES2 is a modification of GES that uses a node-wise decomposable score, called IS-Score (defined below), to score a node X_i given its instance-specific parents $\boldsymbol{Pa}_{IS}(X_i)$ in $\mathcal{G}_{IS}$ and its population-wide parents $\boldsymbol{Pa}_{PW}(X_i)$ in $\mathcal{G}_{PW}$. Let $\boldsymbol{Pa}_{IS}(X_i) = j$ denote that the variables in vector $\boldsymbol{Pa}_{IS}(X_i)$ have the values denoted by vector j in instance T. The basic idea behind the IS-Score is to find those instances (samples) in D in which $\boldsymbol{Pa}_{IS}(X_i) = j$ and use them to score $\boldsymbol{Pa}_{IS}(X_i) \rightarrow X_i$ in $\mathcal{G}_{IS}$. In essence, those instances in D form a cluster that are

similar to instance T in the context of scoring $\boldsymbol{Pa}_{IS}(X_i) \rightarrow X_i$. Since those instances are being used to score $\mathcal{G}_{IS}$, in order to avoid duplicate scoring they can no longer be used to also score $\mathcal{G}_{PW}$; thus, the score for $\mathcal{G}_{PW}$ must be adjusted accordingly. More specifically, let $D_{\boldsymbol{Pa}_{IS}(X_i)=j}$ denote the instances in D in which $\boldsymbol{Pa}_{IS}(X_i) = j$; let $D_{\boldsymbol{Pa}_{IS}(X_i)\neq j}$ denote the remaining instances in D.

Using data $D_{\boldsymbol{Pa}_{IS}(X_i)=j}$, the marginal likelihood of data given $\boldsymbol{Pa}_{IS}(X_i) \rightarrow X_i$ in instance-specific model $\mathcal{G}_{IS}$ is as follows:

$$P(D_{\boldsymbol{Pa}_{IS}(X_i)=j} | \boldsymbol{Pa}_{IS}(X_i) \rightarrow X_i) = \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \cdot \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}, \tag{4.6}$$

where r_i denotes all the possible instantiations of X_i, N_{ijk} is the number of instances in $D_{\boldsymbol{Pa}_{IS}(X_i)=j}$ in which X_i has the value k, and $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$; the terms α_{ijk} and $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$ are the corresponding Dirichlet priors.

Let $\boldsymbol{Pa}_{PW}(X_i)$ denote the parents of X_i in the population-wide model $\mathcal{G}_{PW}$, which in general may be different than the parents of X_i in $\mathcal{G}_{IS}$, as given by $\boldsymbol{Pa}_{IS}(X_i)$. The marginal likelihood of data $D_{\boldsymbol{Pa}_{IS}(X_i)\neq j}$ given $\boldsymbol{Pa}_{PW}(X_i) \rightarrow X_i$ in population-wide model $\mathcal{G}_{PW}$ is as follows:

$$P(D_{\boldsymbol{Pa}_{IS}(X_i)\neq j} | \boldsymbol{Pa}_{PW}(X_i) \rightarrow X_i) = \prod_{l=1}^{q_i} \frac{\Gamma(\alpha_{il})}{\Gamma(\alpha_{il} + N_{il})} \cdot \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ilk} + N_{ilk})}{\Gamma(\alpha_{ilk})}, \tag{4.7}$$

where r_i and q_i are the number of possible instantiations of X_i and $\boldsymbol{Pa}_{PW}(X_i)$, respectively. N_{ilk} is the number of instances in $D_{\boldsymbol{Pa}_{IS}(X_i)\neq j}$ for which X_i takes the value k and its parents $\boldsymbol{Pa}_{PW}(X_i)$ take value l, and $N_{il} = \sum_{k=1}^{r_i} N_{ilk}$. The terms α_{ilk} and $\alpha_{il} = \sum_{k=1}^{r_i} \alpha_{ilk}$ are the corresponding Dirichlet priors.

We calculate the parameter priors in Equations (4.6) and (4.7) as follows. First, we combine the instance-specific and population-wide parents of variable X_i (i.e., $\boldsymbol{Pa}(X_i) = \boldsymbol{Pa}_{PW}(X_i) \cup \boldsymbol{Pa}_{IS}(X_i)$); let $|\boldsymbol{Pa}(X_i)| = q_i'$ be all possible instantiations of $\boldsymbol{Pa}(X_i)$. Then, the uniform priors for the combined parent set $\boldsymbol{Pa}(X_i)$ is formulated as follows:

$$\alpha'_{ijk} = \frac{\alpha}{r_i \cdot q_i'}, \tag{4.8}$$

where r_i denotes all possible instantiations of X_i and α is the prior equivalent sample size. Then, to compute the instance-specific parameter priors for Equation (4.6), we aggregate the

prior terms of the combined parent set in Equation (4.8) that correspond to the instance-specific parents $\boldsymbol{Pa}_{IS}(X_i)$ as follows:

$$\alpha_{ijk} = \alpha \cdot \begin{cases} \sum_{j=1}^{q_i''} \frac{1}{r_i \cdot q_i'} & \text{if } |\boldsymbol{Pa}_{PW}(X_i) \backslash \boldsymbol{Pa}_{IS}(X_i)| > 0 \\ \frac{1}{r_i \cdot q_i'} & \text{otherwise} \end{cases}, \tag{4.9}$$

where $\backslash$ denotes set difference and q_i'' is all possible instantiations of the variables that are in $\boldsymbol{Pa}_{PW}(X_i)$ but not in $\boldsymbol{Pa}_{IS}(X_i)$, which is formulated as follows:

$$q_i'' = \prod_{X_j \in \{\boldsymbol{Pa}_{PW}(X_i) \backslash \boldsymbol{Pa}_{IS}(X_i)\}} |X_j|. \tag{4.10}$$

Similarly, to compute the population-wide parameter priors for Equation (4.7), we aggregate the prior terms of the combined parent set in Equation (4.8) that correspond to the population-wide parents $\boldsymbol{Pa}_{PW}(X_i)$, which are calculated as follows:

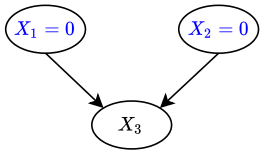
$$\alpha_{ilk} = \alpha \cdot \begin{cases} \sum_{j=1}^{q_i''} \frac{1}{r_i \cdot q_i'} & \text{if } |\boldsymbol{Pa}_{IS}(X_i) \backslash \boldsymbol{Pa}_{PW}(X_i)| > 0 \\ \frac{1}{r_i \cdot q_i'} & \text{otherwise} \end{cases}, \tag{4.11}$$

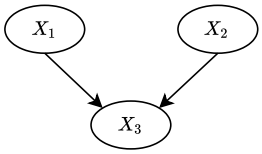
where q_i'' denotes all possible instantiations of the variables that are in $\boldsymbol{Pa}_{IS}(X_i)$ but not in $\boldsymbol{Pa}_{PW}(X_i)$, which is defined as follows when the variables in $\boldsymbol{Pa}_{IS}(X_i) \cap \boldsymbol{Pa}_{PW}(X_i)$ are not instantiated to the same values as of those variables in $\boldsymbol{Pa}_{IS}(X_i)$

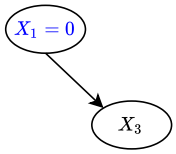
$$q_i'' = \prod_{X_j \in \{\boldsymbol{Pa}_{IS}(X_i) \backslash \boldsymbol{Pa}_{PW}(X_i)\}} |X_j|. \tag{4.12}$$

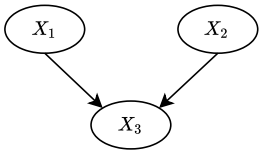
However, when the values of the variables in $\boldsymbol{Pa}_{IS}(X_i) \cap \boldsymbol{Pa}_{PW}(X_i)$ are the same as the values of those variables in $\boldsymbol{Pa}_{IS}(X_i)$, we need to subtract 1 form Equation (4.12) to account for the setting that corresponds to the instance-specific model:

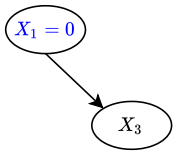
$$q_i'' = \left(\prod_{X_j \in \{\boldsymbol{Pa}_{IS}(X_i) \backslash \boldsymbol{Pa}_{PW}(X_i)\}} |X_j| \right) - 1 \tag{4.13}$$

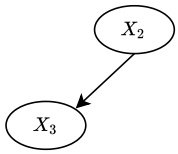
<table>
<tr><th>$\boldsymbol{Pa}(X3) = \boldsymbol{Pa}_{IS}(X3) \cup \boldsymbol{Pa}_{PW}(X3)$</th><th>$\boldsymbol{Pa}_{IS}(X3)$</th><th>$\boldsymbol{Pa}_{PW}(X3)$</th></tr>
<tr>
<td>
<table>
<tr><th>X_1</th><th>X_2</th><th colspan="2">combined priors for X_3
0 1</th></tr>
<tr><td>0</td><td>0</td><td>1/8</td><td>1/8</td></tr>
<tr><td>0</td><td>1</td><td>1/8</td><td>1/8</td></tr>
<tr><td>1</td><td>0</td><td>1/8</td><td>1/8</td></tr>
<tr><td>1</td><td>1</td><td>1/8</td><td>1/8</td></tr>
</table>
$\alpha'_{3jk} = \alpha \cdot \frac{1}{8}$
</td>
<td> 

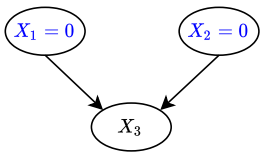
$\alpha_{30k} = \alpha \cdot \frac{1}{8}$</td>
<td> 

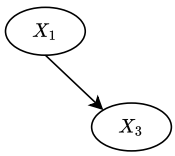
$\alpha_{3lk} = \alpha \cdot \frac{1}{8}$</td>
</tr>
<tr>
<td>
<table>
<tr><th>X_1</th><th>X_2</th><th colspan="2">combined priors for X_3
0 1</th></tr>
<tr><td>0</td><td>0</td><td>1/8</td><td>1/8</td></tr>
<tr><td>0</td><td>1</td><td>1/8</td><td>1/8</td></tr>
<tr><td>1</td><td>0</td><td>1/8</td><td>1/8</td></tr>
<tr><td>1</td><td>1</td><td>1/8</td><td>1/8</td></tr>
</table>
$\alpha'_{3jk} = \alpha \cdot \frac{1}{8}$
</td>
<td> 

$\alpha_{30k} = \alpha \cdot \left(\frac{1}{8} + \frac{1}{8}\right)$</td>
<td> 

$\alpha_{3lk} = \alpha \cdot \frac{1}{8}$</td>
</tr>
<tr>
<td>
<table>
<tr><th>X_1</th><th>X_2</th><th colspan="2">combined priors for X_3
0 1</th></tr>
<tr><td>0</td><td>0</td><td>1/8</td><td>1/8</td></tr>
<tr><td>0</td><td>1</td><td>1/8</td><td>1/8</td></tr>
<tr><td>1</td><td>0</td><td>1/8</td><td>1/8</td></tr>
<tr><td>1</td><td>1</td><td>1/8</td><td>1/8</td></tr>
</table>
$\alpha'_{3jk} = \alpha \cdot \frac{1}{8}$
</td>
<td> 

$\alpha_{30k} = \alpha \cdot \left(\frac{1}{8} + \frac{1}{8}\right)$</td>
<td> 

$\alpha_{3lk} = \alpha \cdot \frac{1}{8}$</td>
</tr>
<tr>
<td>
<table>
<tr><th>X_1</th><th>X_2</th><th colspan="2">combined priors for X_3
0 1</th></tr>
<tr><td>0</td><td>0</td><td>1/8</td><td>1/8</td></tr>
<tr><td>0</td><td>1</td><td>1/8</td><td>1/8</td></tr>
<tr><td>1</td><td>0</td><td>1/8</td><td>1/8</td></tr>
<tr><td>1</td><td>1</td><td>1/8</td><td>1/8</td></tr>
</table>
$\alpha'_{3jk} = \alpha \cdot \frac{1}{8}$
</td>
<td> 

$\alpha_{30k} = \alpha \cdot \frac{1}{8}$</td>
<td> 

$\alpha_{31k} = \begin{cases} \alpha \cdot \frac{1}{8} & \text{if } X_1 = 0 \\ \alpha \cdot \left(\frac{1}{8} + \frac{1}{8}\right) & \text{else} \end{cases}$</td>
</tr>
</table>

Figure 9: Examples of how to compute parameter priors for variable X_3 given various sets of instance-specific parents $\boldsymbol{Pa}_{IS}(X_3)$ and population-wide parents $\boldsymbol{Pa}_{PW}(X_3)$.

Figure 9 shows some examples of parameter prior calculations for variable X_3 with different sets of instance-specific and population-wide parents.

Assuming parameter independence and parameter modularity [Heckerman et al., 1995], as is commonly done, the overall marginal likelihood of data given the the instance-specific and the population-wide parents of node X_i is calculated as follows:

$$\begin{aligned} &P(D|\boldsymbol{Pa}_{IS}(X_i) \to X_i, \boldsymbol{Pa}_{PW}(X_i) \to X_i) = \\ &P(D_{\boldsymbol{Pa}_{IS}(X_i)=j}|\boldsymbol{Pa}_{IS}(X_i) \to X_i) \cdot P(D_{\boldsymbol{Pa}_{IS}(X_i)\neq j}|\boldsymbol{Pa}_{PW}(X_i) \to X_i)) \,. \end{aligned} \tag{4.14}$$

This score represents the marginal likelihood of X_i given the instance-specific and population-wide parents of X_i. Algorithm 11 shows pseudo-code for the IS-Score procedure that derives this marginal likelihood as the overall score for X_i. It is this procedure that GES2 calls when scoring a node given its parents during the forward and backward greedy search (line 2 in Algorithm 10).

Algorithm 11 IS-Score(D, T, X_i, $\boldsymbol{Pa}_{IS}(X_i)$, $\boldsymbol{Pa}_{PW}(X_i)$)

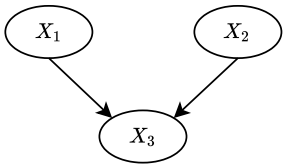
Input: a dataset D, a test instance T, variable X_i that is being scored, X_i's instance-specific parent set $\boldsymbol{Pa}_{IS}(X_i)$, and X_i's population-wide parent set $\boldsymbol{Pa}_{PW}(X_i)$

Output: the overall score for X_i

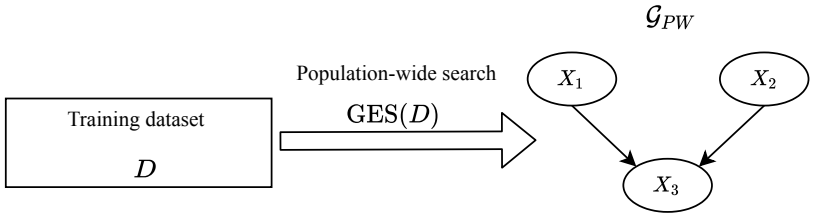
1: Derive $D_{\boldsymbol{Pa}_{IS}(X_i)=j}$ and $D_{\boldsymbol{Pa}_{IS}(X_i)\neq j}$ from D and the values j of $\boldsymbol{Pa}_{IS}(X_i)$ in T
2: $s_{IS} \leftarrow P(D_{\boldsymbol{Pa}_{IS}(X_i)=j}|\boldsymbol{Pa}_{IS}(X_i) \to X_i)$ ▷ Equation (4.6)
3: $s_{PW} \leftarrow P(D_{\boldsymbol{Pa}_{IS}(X_i)\neq j}|\boldsymbol{Pa}_{PW}(X_i) \to X_i)$ ▷ Equation (4.7)
4: $s_{overall} \leftarrow s_{IS} \cdot s_{PW}$ ▷ Equation (4.14)
5: return $s_{overall}$

Figure 10 shows an example of the IGES procedure. Let Figure 10a represent the data-generating CBN structure and parameters for variable X_3. In the large sample limit, by applying GES with the BDeu score we expect to learn $\mathcal{G}_{PW}$ (Figure 10b), which is the same as the data-generating structure. However, $\mathcal{G}_{PW}$ does not capture the independence of X_2 and X_3 when $X_1 = 0$ (i.e., $X_2 \perp\!\!\!\perp_c X_3|X_1 = 0$) in the given instance $T = \{X_1 = 0, X_2 = 1, X_3 = 0\}$. Figure 10c shows the instance-specific CBN structure $\mathcal{G}_{IS}$ and the population-wide structure $\mathcal{G}_{PW}$ that would be learned by the IGES algorithm, in the large sample limit.

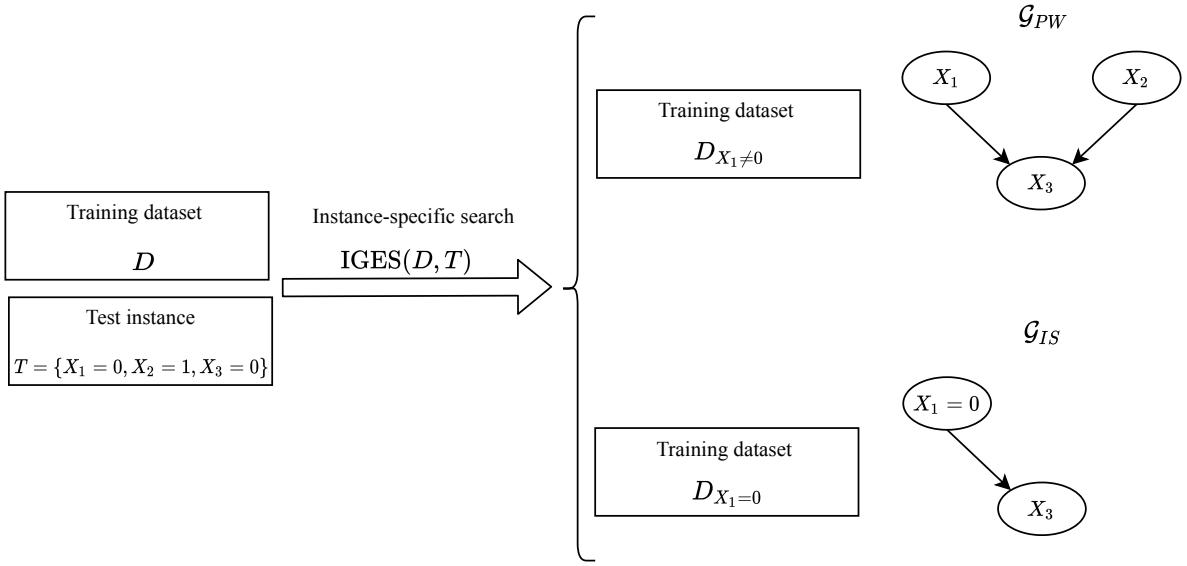
X_1	X_2	$P(X_3 \mid X_1, X_2)$ 0	1
0	0	0.90	0.10
0	1	0.90	0.10
1	0	0.23	0.77
1	1	0.52	0.48



(a) The conditional probability table on the left represents $P(X_3|X_1, X_2)$ and the graph on the right shows the data-generating CBN structure.



(b) The result of applying GES to the example in the large sample limit.



(c) The results of applying IGES to the example in the large sample limit.

Figure 10: This example illustrates a situation in which the population-wide CBN structure learning is not capable of capturing context-specific independence in the CBN structure while the instance-specific approach is.

As mentioned, the IS-Score derives the marginal likelihood of the data on X_i, relative to the instance-specific and population-wide parents of X_i. Assuming parameter independence and parameter modularity, the marginal likelihood of all the data given T, $\mathcal{G}_{IS}$, and $\mathcal{G}_{PW}$ is as follows:

$$P(D|T, \mathcal{G}_{IS}, \mathcal{G}_{PW}) = \prod_{i=1}^{n} \text{IS-Score}(D, T, X_i, \boldsymbol{Pa}_{IS}(X_i), \boldsymbol{Pa}_{PW}(X_i)) \,, \tag{4.15}$$

where i iterates over the set of all nodes being modeled. This equation will be used later in Equation (4.19) to derive an overall CBN structure score.

We can also define modular structure priors that are decomposable at the node level to be applied when scoring the parent-child relationship for each node. We use the following structure priors when applying GES to learn the population-wide model [Ramsey et al., 2017]:

$$P(\mathcal{G}_{PW}) = \prod_{i=1}^{n} \left(\frac{e}{n-1} \right)^{|\boldsymbol{Pa}_{PW}(X_i)|} \cdot \left(1 - \frac{e}{n-1} \right)^{n-1-|\boldsymbol{Pa}_{PW}(X_i)|} \,, \tag{4.16}$$

where i iterates over the set of all n nodes in $\mathcal{G}_{PW}$, $|\boldsymbol{Pa}_{PW}(X_i)|$ is the number of parents of node X_i in $\mathcal{G}_{PW}$, and e is a prior weight, which we set to be $e = 1$ in this dissertation. In this structure prior, each node being a parent of another node is modeled as a Bernoulli trial.

To compute the prior probabilities of the instance-specific CBN structure $\mathcal{G}_{IS}$, we modify the modular structure prior introduced in [Heckerman et al., 1995] by considering $\mathcal{G}_{PW}$ as the prior network:

$$P(\mathcal{G}_{IS}) = c \prod_{i=1}^{n} \kappa^{\delta_i} \,, \tag{4.17}$$

where c is a normalization constant, i iterates over the set of all nodes, δ_i is the absolute edge difference between instance-specific parents of X_i in $\mathcal{G}_{IS}$ (i.e., $\boldsymbol{Pa}_{IS}(X_i)$) and its population-wide parents in $\mathcal{G}_{PW}$ (i.e., $\boldsymbol{Pa}_{PW}(X_i)$), which is calculated as follows:

$$\delta_i = |\{\boldsymbol{Pa}_{IS}(X_i) \cup \boldsymbol{Pa}_{PW}(X_i)\} - \{\boldsymbol{Pa}_{IS}(X_i) \cap \boldsymbol{Pa}_{PW}(X_i)\}|. \tag{4.18}$$

Finally, κ ($0 < \kappa \leq 1$) is a penalty factor for the instance-specific parents differing from the population-wide parents. We combine Equations (4.15), (4.16), and (4.17) to derive a probability that is proportional to the posterior probability of $\mathcal{G}_{IS}$ and $\mathcal{G}_{PW}$:

$$P(\mathcal{G}_{IS}, \mathcal{G}_{PW}|D, T) \propto P(D|T, \mathcal{G}_{IS}, \mathcal{G}_{PW}) \cdot P(\mathcal{G}_{IS}) \cdot P(\mathcal{G}_{PW}) \,. \tag{4.19}$$

Theorem 4.4.1. Given the Markov, faithfulness, and causal sufficiency assumptions, i.i.d. sampling, a node ordering[1], a locally consistent score, and a test instance T, in the large sample limit the IGES algorithm learns a CBN that represents the instance-specific data-generating CBN of T.

Proof. In the first stage, IGES applies the GES algorithm with the BDeu score, which is locally consistent [Chickering, 2002], and recovers the data-generating CBNs in the large sample limit, given the stated assumptions [Chickering, 2002]. In the second stage, IGES applies the same GES algorithm using the IS-Score. In Theorem 4.4.2 (Section 4.4.1) we prove that doing so leads to finding the data-generating parents of each node, given a node ordering. Therefore, assuming the stated assumptions, IGES outputs the instance-specific data-generating CBN for instance T. □

4.4.1 IS-Score consistency

In this section, I provide a proof that IS-Score is consistent when we assume an ordering of variables. Before that, I describe the possible situations that may occur while running the IGES method to learn an instance-specific CBN for a given test instance T (Section 4.4). To do so, I use the example in Figure 11 that shows the data-generating model of a single variable $X_i \in \boldsymbol{V}$, which can be extended to all domain variables in $\boldsymbol{V}$. In this example, $\boldsymbol{W} = \{X_j, ..., X_m\}$ denotes the data-generating parents of X_i, where all the variables in $\boldsymbol{W}$ precede X_i. Also, $\boldsymbol{U}_T = \{X_k = a, X_{k+1} = b, X_{k+2} = c\}$ denotes the CSI parent structure that represents the distribution of X_i for T, where $\boldsymbol{U}_T \subseteq \boldsymbol{W}$ based on the ordering.

As described in Algorithm 10, in the first stage of IGES, we apply the GES method to learn the population-wide model. Under assumptions, GES will discover the correct parents of X_i in the large sample limit (i.e., $\boldsymbol{Pa}_{PW}(X_i) = \boldsymbol{W}$ that is shown in the third column of Figure 12), as proven in [Chickering, 2002]. However, $\boldsymbol{Pa}_{PW}(X_i)$ does not explicitly represent the particular CSI parent structure of X_i for T. In the second stage of IGES, we apply GES2

[1] A node ordering can be used in GES and IGES algorithms in the form of tiered background knowledge $\boldsymbol{T} = \{T_1, T_2, ..., T_n\}$, where T_1 includes the first node in the ordering, T_2 includes the second node in the ordering, and so forth. By using $\boldsymbol{T}$ as the background knowledge, BN structures can only have edges from the variable in T_i to the variable in T_j if $1 \leq i < j \leq n$.

using IS-Score to learn the instance-specific parents that encode the CSI structure of X_i for T. During the GES2 search, three possible situations may occur when we score the marginal likelihood given an arbitrary instance-specific hypothesis parent set, which we denote as $\boldsymbol{Pa}'_{IS}(X_i)$, and the population-wide data-generating parents $\boldsymbol{Pa}_{PW}(X_i) = \boldsymbol{W}$:

Case 1: $\boldsymbol{Pa}'_{IS}(X_i) = \boldsymbol{U}_T$ (Figure 12 row 1). In this case, the instance-specific hypothesis parent set that is being scored is the proper subset of the population-wide data-generating parents that encodes the CSI parent structure of X_i for T.

Case 2: $\boldsymbol{Pa}'_{IS}(X_i) \neq \boldsymbol{U}_T$ and $\boldsymbol{Pa}'_{IS}(X_i) \cap \boldsymbol{U}_T = \boldsymbol{U}_T$. In this case, the instance-specific hypothesis parent set that is being scored includes all CSI parent structure $\boldsymbol{U}_T$ for T in addition to some variables outside of $\boldsymbol{U}_T$. Two examples of this case are shown in the second row of Figure 12.

Case 3: $\boldsymbol{Pa}'_{IS}(X_i) \neq \boldsymbol{U}_T$ and $\boldsymbol{Pa}'_{IS}(X_i) \cap \boldsymbol{U}_T \neq \boldsymbol{U}_T$. In this case, the instance-specific hypothesis parent set that is being scored may include a subset of the instance-specific data-generating parents $\boldsymbol{U}_T$ and/or some additional variables outside of $\boldsymbol{U}_T$. Two examples of this case are shown in Figure 12 row 3.

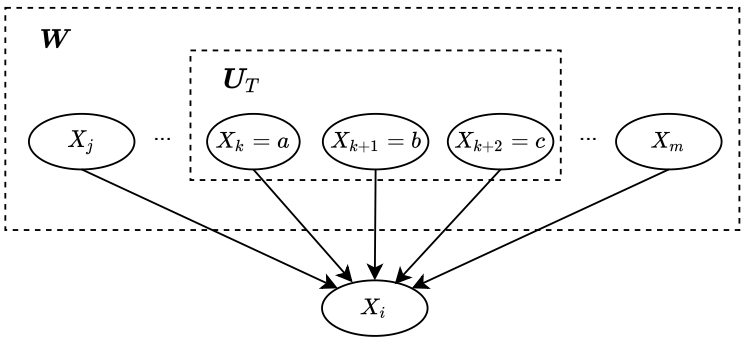


Figure 11: This example shows the data-generating model of variable X_i in which $\boldsymbol{W} = \{X_j, ..., X_m\}$ are parents X_i, where all variables in $\boldsymbol{W}$ precede X_i since we assume an ordering on variables. In this data-generating model, a subset $\boldsymbol{U}_T = \{X_k = a, X_{k+1} = b, X_{k+2} = c\} \subseteq \boldsymbol{W}$ denotes the context-specific independence (CSI) parent structure that represents the distribution of X_i for T. In the large sample limit, the population-wide GES method learns $\boldsymbol{W}$, which does not explicitly represent the particular CSI structure $\boldsymbol{U}_T$.

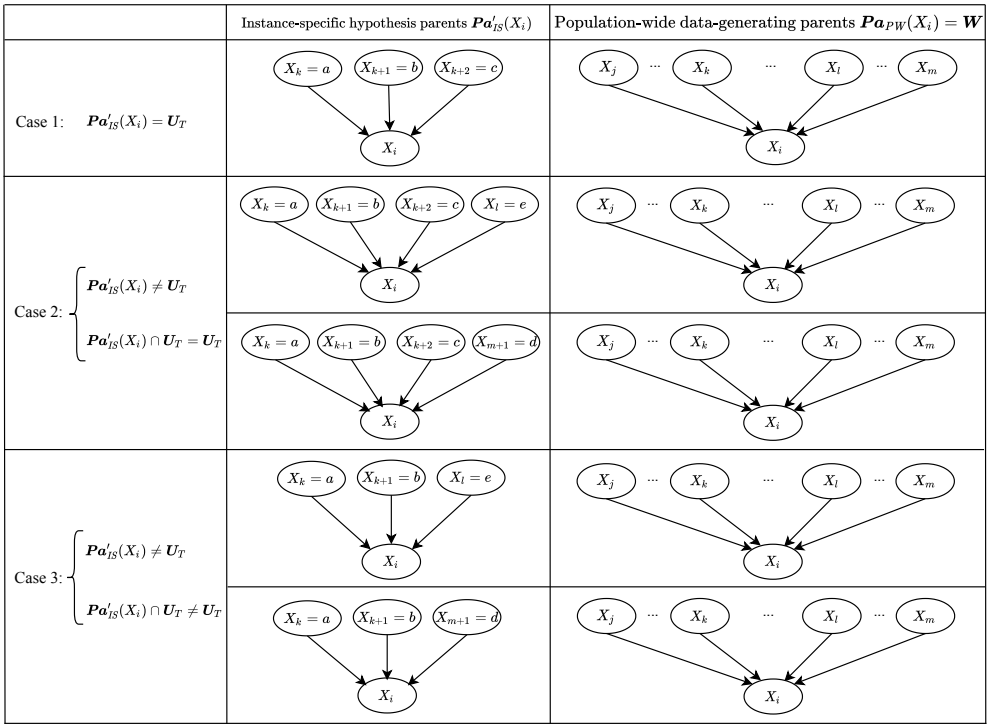


Figure 12: This example illustrates the key situations that may occur while applying IGES with IS-Score to learn an instance-specific model for given a test instance T based on the data-generating model in Figure 11, where $\boldsymbol{U}_T = \{X_k = a, X_{k+1} = b, X_{k+2} = c\}$ denotes the CSI parent structure that represents the distribution of X_i for T. As described in Algorithm 10, we first apply GES to learn the population-wide parents, which we assume has discovered the data-generating parents $\boldsymbol{Pa}_{PW}(X_i) = \boldsymbol{W}$ in the large sample limit (column 3). Then, when we apply GES2 with IS-Score, three possible cases may occur while scoring an arbitrary instance-specific hypothesis parent set $\boldsymbol{Pa}'_{IS}(X_i)$; examples are given in rows 1-3.

Theorem 4.4.2. Let D be a complete dataset on a set of discrete variables $\boldsymbol{V} = \{X_1, X_2, ..., X_n\}$ that contains N samples from distribution P, which is strictly positive, and T be a single additional sample from P. Also, let $\mathcal{G}_{PW}$ be the data-generating CBN on $\boldsymbol{V}$ that is Markov and faithful to P, and $\mathcal{G}_{IS}$ be the instance-specific data-generating CBN on $\boldsymbol{V}$ that is that is Markov and faithful to P according to CSI parent structures in T (see Section 2.1.4 for more details), where both $\mathcal{G}_{PW}$ and $\mathcal{G}_{IS}$ have the same ordering on the variables. IS-Score is consistent given a node ordering that is consistent with the node ordering of the data-generating CBN $\mathcal{G}_{PW}$.

Proof. To facilitate the proof of this theorem, we first derive how the IS-Score and the BD score are calculated. Since both scores are decomposable at the node level, we prove this theorem for a single variable X_i, which is straightforward to extend to all variables of the BN. Finally, we prove the consistency of the IS-Score in considering different instance-specific hypothesis parents structures for X_i.

As described in Section 4.4, the IS-Score is composed of two components: (1) the instance-specific structure that includes X_i's parents in the hypothesis CBN $\mathcal{G}'$ that take value j according to T (i.e., $\boldsymbol{Pa}'_{IS}(X_i) = j$) and (2) the population-wide structure that includes X_i's parents in the data-generating BN $\mathcal{G}_{PW}$ (i.e., $\boldsymbol{Pa}_{PW}(X_i)$):

$$\begin{aligned} P(D_{\boldsymbol{Pa}'_{IS}(X_i)=j} | \boldsymbol{Pa}'_{IS}(X_i) \rightarrow X_i) &= \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})} \cdot \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})} \\ P(D_{\boldsymbol{Pa}'_{IS}(X_i)\neq j} | \boldsymbol{Pa}_{PW}(X_i) \rightarrow X_i) &= \prod_{l=1}^{q_i} \frac{\Gamma(\alpha_{il})}{\Gamma(\alpha_{il}+N_{il})} \cdot \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ilk}+N_{ilk})}{\Gamma(\alpha_{ilk})}, \end{aligned} \tag{4.20}$$

where r_i denotes all values of X_i. N_{ijk} is the number of instances in $D_{\boldsymbol{Pa}'_{IS}(X_i)=j}$ in which $X_i = k$ ($N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$) and the terms α_{ijk} ($\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$) are the corresponding Dirichlet priors. In the second equation, q_i is the number of possible values of $\boldsymbol{Pa}_{PW}(X_i)$, which excludes the instantiations that overlap with $\boldsymbol{Pa}'_{IS}(X_i) = j$. N_{ilk} is the number of instances in $D_{\boldsymbol{Pa}'_{IS}(X_i)\neq j}$ in which $X_i = k$ and $\boldsymbol{Pa}_{PW}(X_i) = l$ ($N_{il} = \sum_{k=1}^{r_i} N_{ilk}$), and the terms α_{ilk} ($\alpha_{il} = \sum_{k=1}^{r_i} \alpha_{ilk}$) are the corresponding Dirichlet priors. We can combine Equations (4.20) as follows:

$$P(D_{X_i} | \boldsymbol{Pa}^*(X_i)) = \prod_{d=1}^{q_i^*} \frac{\Gamma(\alpha_{id})}{\Gamma(\alpha_{id}+N_{id})} \cdot \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{idk}+N_{idk})}{\Gamma(\alpha_{idk})}, \tag{4.21}$$

where $\boldsymbol{Pa}^*(X_i)$ denotes the combined parent set of X_i (i.e., the union of $\boldsymbol{Pa}'_{IS}(X_i)$ and $\boldsymbol{Pa}_{PW}(X_i)$) that has q_i^* distinguishable instantiations by grouping the parents that have equivalent effect on X_i based on the CSI parent structure that exists in T. Also, N_{idk} denotes the number of cases in which $X_i = k$ and its distinguishable parent instantiation takes value d ($N_{id} = \sum_{k=1}^{r_i} N_{idk}$), and α_{idk} denotes the corresponding pseudo-counts ($\alpha_{id} = \sum_{k=1}^{r_i} \alpha_{idk}$). We use D_{X_i} in Equation (4.21) since we only score the data about X_i here. Equation (4.21) can be re-written in log form as follows:

$$\log P(D_{X_i}|\boldsymbol{Pa}^*(X_i)) =$$
$$\sum_{d=1}^{q_i^*}\left[\log\Gamma(\alpha_{id}) - \log\Gamma(\alpha_{id}+N_{id}) + \sum_{k=1}^{r_i}\left[\log\Gamma(\alpha_{idk}+N_{idk}) - \log\Gamma(\alpha_{idk})\right]\right]. \tag{4.22}$$

We can re-arrange the terms in Equation (4.22) to gather the constant terms as follows:

$$\log P(D_{X_i}|\boldsymbol{Pa}^*(X_i)) =$$
$$\sum_{d=1}^{q_i^*}\left[-\log\Gamma(\alpha_{id}+N_{id}) + \sum_{k=1}^{r_i}\log\Gamma(\alpha_{idk}+N_{idk})\right] + \sum_{d=1}^{q_i^*}\left[\log\Gamma(\alpha_{id}) - \sum_{k=1}^{r_i}\log\Gamma(\alpha_{idk})\right]$$
$$= \sum_{d=1}^{q_i^*}\left[-\log\Gamma(\alpha_{id}+N_{id}) + \sum_{k=1}^{r_i}\log\Gamma(\alpha_{idk}+N_{idk})\right] + const. \tag{4.23}$$

Using the Stirling's approximation of $\lim_{n\to\infty}\log\Gamma(n) = (n-\frac{1}{2})\log(n) - n + const.$, we can re-write Equation (4.23) as follows:

$$\lim_{N\to\infty}\log P(D_{X_i}|\boldsymbol{Pa}^*(X_i)) = \lim_{N\to\infty}\sum_{d=1}^{q_i^*}\Bigg[-(\alpha_{id}+N_{id}-\frac{1}{2})\log(\alpha_{id}+N_{id}) + (\alpha_{id}+N_{id})$$
$$+\sum_{k=1}^{r_i}\left((\alpha_{idk}+N_{idk}-\frac{1}{2})\log(\alpha_{idk}+N_{idk}) - (\alpha_{idk}+N_{idk})\right)\Bigg] + const.$$
$$= \lim_{N\to\infty}\sum_{d=1}^{q_i^*}\Bigg[-\alpha_{id}\log(\alpha_{id}+N_{id}) - N_{id}\log(\alpha_{id}+N_{id}) + \frac{1}{2}\log(\alpha_{id}+N_{id}) + \alpha_{id} + N_{id} + \sum_{k=1}^{r_i}$$
$$\left(\alpha_{idk}\log(\alpha_{idk}+N_{idk}) + N_{idk}\log(\alpha_{idk}+N_{idk}) - \frac{1}{2}\log(\alpha_{idk}+N_{idk}) - \alpha_{idk} - N_{idk}\right)\Bigg] + const.$$
$$= \lim_{N\to\infty}\sum_{d=1}^{q_i^*}\left[-N_{id}\log(\alpha_{id}+N_{id}) + \sum_{k=1}^{r_i}N_{idk}\log(\alpha_{idk}+N_{idk})\right]$$
$$+\sum_{d=1}^{q_i^*}\left[-\alpha_{id}\log(\alpha_{id}+N_{id}) + \sum_{k=1}^{r_i}\alpha_{idk}\log(\alpha_{idk}+N_{idk})\right]$$
$$+\frac{1}{2}\sum_{d=1}^{q_i^*}\left[\log(\alpha_{id}+N_{id}) - \sum_{k=1}^{r_i}\log(\alpha_{idk}+N_{idk}) + \alpha_{id} + N_{id} - \sum_{k=1}^{r_i}(\alpha_{idk}+N_{idk})\right] + const.$$
$$= \lim_{N\to\infty}\sum_{d=1}^{q_i^*}\left[-N_{id}\log(\alpha_{id}+N_{id}) + \sum_{k=1}^{r_i}N_{idk}\log(\alpha_{idk}+N_{idk})\right]$$
$$+\sum_{d=1}^{q_i^*}\left[-\alpha_{id}\log(\alpha_{id}+N_{id}) + \sum_{k=1}^{r_i}\alpha_{idk}\log(\alpha_{idk}+N_{idk})\right]$$
$$+\frac{1}{2}\sum_{d=1}^{q_i^*}\left[\log(\alpha_{id}+N_{id}) - \sum_{k=1}^{r_i}\log(\alpha_{idk}+N_{idk})\right] + const. \tag{4.24}$$

In the last step of Equation (4.24), we used the facts that $\sum_{k=1}^{r_i} N_{idk} = N_{id}$ and $\sum_{k=1}^{r_i} \alpha_{idk} = \alpha_{id}$, and we can apply these identities again to that equation to obtain the following:

$$\begin{aligned}
&\lim_{N\to\infty} \log P(D_{X_i}|\boldsymbol{Pa}^*(X_i)) = \\
&\qquad \lim_{N\to\infty} \sum_{d=1}^{q_i^*} \sum_{k=1}^{r_i} \left[N_{idk} \log(\frac{\alpha_{idk} + N_{idk}}{\alpha_{id} + N_{id}}) + \alpha_{idk} \log(\frac{\alpha_{idk} + N_{idk}}{\alpha_{id} + N_{id}}) \right] \\
&\qquad + \frac{1}{2} \sum_{d=1}^{q_i^*} \left[\log(\alpha_{id} + N_{id}) - \sum_{k=1}^{r_i} \log(\alpha_{idk} + N_{idk}) \right] + const.
\end{aligned} \tag{4.25}$$

Given that

$$\lim_{N\to\infty} \frac{\alpha_{idk} + N_{idk}}{\alpha_{id} + N_{id}} = \frac{N_{idk}}{N_{id}}$$

and

$$\lim_{N\to\infty} \sum_{d=1}^{q_i^*} \sum_{k=1}^{r_i} \alpha_{idk} \log(\frac{\alpha_{idk} + N_{idk}}{\alpha_{id} + N_{id}}) = const.,$$

in the limit, Equation (4.25) becomes:

$$\begin{aligned}
&\lim_{N\to\infty} \log P(D_{X_i}|\boldsymbol{Pa}^*(X_i)) = \\
&\lim_{N\to\infty} \sum_{d=1}^{q_i^*} \sum_{k=1}^{r_i} N_{idk} \log \frac{N_{idk}}{N_{id}} + \frac{1}{2} \sum_{d=1}^{q_i^*} \left[\log(\alpha_{id} + N_{id}) - \sum_{k=1}^{r_i} \log(\alpha_{idk} + N_{idk}) \right] + const.,
\end{aligned} \tag{4.26}$$

or equivalently:

$$\begin{aligned}
&\lim_{N\to\infty} \log P(D_{X_i}|\boldsymbol{Pa}^*(X_i)) = \\
&\lim_{N\to\infty} N \cdot \sum_{d=1}^{q_i^*} \sum_{k=1}^{r_i} \frac{N_{idk}}{N} \log \frac{N_{idk}}{N_{id}} + \frac{1}{2} \sum_{d=1}^{q_i^*} \left[\log(\alpha_{id} + N_{id}) - \sum_{k=1}^{r_i} \log(\alpha_{idk} + N_{idk}) \right] + const. \\
&= \lim_{N\to\infty} -N \cdot H_{X_i|\boldsymbol{Pa}^*(X_i)} + \frac{1}{2} \sum_{d=1}^{q_i^*} \left[\log(\alpha_{id} + N_{id}) - \sum_{k=1}^{r_i} \log(\alpha_{idk} + N_{idk}) \right] + const.
\end{aligned} \tag{4.27}$$

To simplify the second term in Equation (4.27), we divide the log terms by N and equivalently add $\log N$ terms as follows:

$$\begin{aligned}
&\frac{1}{2}\sum_{d=1}^{q_i^*}\left[\log(\alpha_{id}+N_{id})-\sum_{k=1}^{r_i}\log(\alpha_{idk}+N_{idk})\right]= \\
&\frac{1}{2}\sum_{d=1}^{q_i^*}\left[\log(\frac{\alpha_{id}+N_{id}}{N})+\log N-\sum_{k=1}^{r_i}\log(\frac{\alpha_{idk}+N_{idk}}{N})+\log N\right] \\
&=\frac{1}{2}\sum_{d=1}^{q_i^*}\left(\log N-\sum_{k=1}^{r_i}\log N\right)+\frac{1}{2}\sum_{d=1}^{q_i^*}\left[\log(\frac{\alpha_{id}+N_{id}}{N})-\sum_{k=1}^{r_i}\log(\frac{\alpha_{idk}+N_{idk}}{N})\right] \\
&=-\frac{q_i^*(r_i-1)}{2}\log N+const.
\end{aligned} \tag{4.28}$$

Combining Equations (4.27) and (4.28), we obtain:

$$\lim_{N\to\infty}\log P(D_{X_i}|\boldsymbol{Pa}^*(X_i))=\lim_{N\to\infty}-N\cdot H_{X_i|\boldsymbol{Pa}^*(X_i)}-\frac{q_i^*\cdot(r_i-1)}{2}\log N+const. \tag{4.29}$$

Similarly, we can derive $\lim_{N\to\infty}\log P(D_{X_i}|\boldsymbol{Pa}_{PW}(X_i))$ using the BD score (Equation (4.4)) as follows:

$$\lim_{N\to\infty}\log P(D_{X_i}|\boldsymbol{Pa}_{PW}(X_i))=\lim_{N\to\infty}-N\cdot H_{X_i|\boldsymbol{Pa}_{PW}(X_i)}-\frac{q_i\cdot(r_i-1)}{2}\log N+const., \tag{4.30}$$

where q_i is the number of possible parent instantiations of X_i in the data-generating model $\mathcal{G}_{PW}$, without considering the CSI structure.

Suppose $\boldsymbol{U}_T\subseteq\boldsymbol{Pa}_{PW}(X_i)$ denotes the data-degenerating instance-specific parents of X_i in $\mathcal{G}_{IS}$ for instance T; as described in the example shown in Figure 12, there are three possible cases.

Case 1: $\boldsymbol{Pa}'_{\boldsymbol{IS}}(X_i)=\boldsymbol{U}_T$, which indicates that the instance-specific hypothesis parent set $\boldsymbol{Pa}'_{IS}(X_i)$ is the same as the instance-specific data-generating parents of X_i for T. There are two possible situations:

Case 1a: $\boldsymbol{U}_T=\boldsymbol{Pa}_{PW}(X_i)$, which indicates that X_i does not include any CSI parent structure for T. To compare the scores of the instance-specific hypothesis structure to the data-generating structure, we combine Equations (4.29) and (4.30) as follows:

$$\begin{aligned}
&\lim_{N\to\infty}\log\frac{P(D_{X_i}|\boldsymbol{Pa}^{*,1a}(X_i))}{P(D_{X_i}|\boldsymbol{Pa}_{PW}(X_i))}= \\
&\lim_{N\to\infty}N\cdot[-H_{X_i|\boldsymbol{Pa}^{*,1a}(X_i)}+H_{X_i|\boldsymbol{Pa}_{PW}(X_i)}]+\frac{(r_i-1)(q_i-q_i^{*,1a})}{2}\log N,
\end{aligned} \tag{4.31}$$

where $\boldsymbol{Pa}^{*,1a}(X_i)$ denotes the combined parent set given in case 1a and $q_i^{*,1a}$ denotes the number of its instantiations. In Equation (4.31), $\boldsymbol{Pa}^{*,1a}(X_i)$ contains exactly the same information as $\boldsymbol{Pa}_{PW}(X_i)$, and as a result, the entropy of X_i remains the same given either $\boldsymbol{Pa}^{*,1a}(X_i)$ or $\boldsymbol{Pa}_{PW}(X_i)$. Also, the number of parameters will remain exactly the same. Therefore, Equation (4.31) goes to 1.0 in the limit as $N \to \infty$.

Case 1b: $\boldsymbol{U}_T \subset \boldsymbol{Pa}_{PW}(X_i)$, which indicates that X_i include a CSI parent structure that holds in T. Similar to case 1a, we compare the scores of the instance-specific hypothesis structure to the data-generating structure as follows:

$$\begin{aligned} &\lim_{N\to\infty} \log \frac{P(D_{X_i}|\boldsymbol{Pa}^{*,1b}(X_i))}{P(D_{X_i}|\boldsymbol{Pa}_{PW}(X_i))} = \\ &\lim_{N\to\infty} N \cdot [-H_{X_i|\boldsymbol{Pa}^{*,1b}(X_i)} + H_{X_i|\boldsymbol{Pa}_{PW}(X_i)}] + \frac{(r_i-1)(q_i - q_i^{*,1b})}{2} \log N. \end{aligned} \tag{4.32}$$

The combined parent set $\boldsymbol{Pa}^{*,1b}(X_i)$ does not change the distribution; rather, it compacts the parameters that are the same due to the CSI structure in $\boldsymbol{U}_T$. Therefore, $\boldsymbol{Pa}^{*,1b}(X_i)$ contains exactly the same information as the data-degenerating population-wide parents $\boldsymbol{Pa}_{PW}(X_i)$, and as a result, the entropy of X_i remains the same given either $\boldsymbol{Pa}^{*,1b}(X_i)$ or $\boldsymbol{Pa}_{PW}(X_i)$. Consequently, the first term in Equation (4.32) cancels and we obtain:

$$\lim_{N\to\infty} \log \frac{P(D_{X_i}|\boldsymbol{Pa}^{*,1}(X_i))}{P(D_{X_i}|\boldsymbol{Pa}_{PW}(X_i))} = \lim_{N\to\infty} \frac{(r_i-1)(q_i - q_i^{*,1})}{2} \log N. \tag{4.33}$$

Given that $q_i > q_i^{*,1b}$, the term $(q_i - q_i^{*,1b})$ becomes a positive constant; also, the term $\frac{(r_i-1)}{2}$ is a positive constant. Therefore, Equation (4.33) goes to infinity in the limit as $N \to \infty$.

Case 2: $\boldsymbol{Pa}'_{IS}(X_i) \neq \boldsymbol{U}_T$ and $\boldsymbol{Pa}'_{IS}(X_i) \cap \boldsymbol{U}_T = \boldsymbol{U}_T$, which indicates that the instance-specific hypothesis parent set $\boldsymbol{Pa}'_{IS}(X_i)$ includes all variables in $\boldsymbol{U}_T$ and may include other variables outside of $\boldsymbol{U}_T$. We need to compare case 1 [2] versus case 2 as follows:

$$\begin{aligned} &\lim_{N\to\infty} \log \frac{P(D_{X_i}|\boldsymbol{Pa}^{*,1}(X_i))}{P(D_{X_i}|\boldsymbol{Pa}^{*,2}(X_i))} = \\ &\lim_{N\to\infty} N \cdot [-H_{X_i|\boldsymbol{Pa}^{*,1}(X_i)} + H_{X_i|\boldsymbol{Pa}^{*,2}(X_i)}] + \frac{(r_i-1)(q_i^{*,2} - q_i^{*,1})}{2} \log N, \end{aligned} \tag{4.34}$$

[2]The result holds for both case 1a and case 1b, and thus, we do not make a distinction in our discussion here.

where $\boldsymbol{Pa}^{*,1}(X_i)$ and $\boldsymbol{Pa}^{*,2}(X_i)$ denote the combined parent set in case 1 and case 2, and $q_i^{*,1}$ and $q_i^{*,2}$ denote the number of instantiations of these parent sets, respectively. The additional variables in the combined parent set $\boldsymbol{Pa}^{*,2}(X_i)$ do not affect the distribution of X_i, and consequently, the entropy of X_i remains the same given $\boldsymbol{Pa}^{*,2}(X_i)$. Therefore, the first term in Equation (4.34) cancels and we obtain:

$$\lim_{N\to\infty} \log \frac{P(D_{X_i}|\boldsymbol{Pa}^{*,1}(X_i))}{P(D_{X_i}|\boldsymbol{Pa}^{*,2}(X_i))} = \lim_{N\to\infty} \frac{(r_i-1)(q_i^{*,2}-q_i^{*,1})}{2}\log N. \tag{4.35}$$

Since $q_i^{*,2} > q_i^{*,1}$, the term $(q_i^{*,2} - q_i^{*,1})$ becomes a positive constant; also, the term $\frac{(r_i-1)}{2}$ is a positive constant. Thus, Equation (4.35) approaches to ∞ as $N \to \infty$. This result holds regardless of whether $\boldsymbol{U}_T$ does include CSI structure (case 1a) or not (case 1b).

Case 3: $\boldsymbol{Pa}'_{IS}(X_i) \neq \boldsymbol{U}_T$ and $\boldsymbol{Pa}'_{IS}(X_i) \cap \boldsymbol{U}_T \neq \boldsymbol{U}_T$, which indicates that the instance-specific hypothesis parent set $\boldsymbol{Pa}'_{IS}(X_i)$ does not include all of the variables in U_T may include variables outside of U_T. Comparing case 1 [3] versus case 3 we have:

$$\begin{aligned}&\lim_{N\to\infty} \log \frac{P(D_{X_i}|\boldsymbol{Pa}^{*,1}(X_i))}{P(D_{X_i}|\boldsymbol{Pa}^{*,3}(X_i))} = \\ &\lim_{N\to\infty} N\cdot[-H_{X_i|\boldsymbol{Pa}^{*,1}(X_i)} + H_{X_i|\boldsymbol{Pa}^{*,3}(X_i)}] + \frac{(r_i-1)(q_i^{*,3}-q_i^{*,1})}{2}\log N,\end{aligned} \tag{4.36}$$

where the first term is of $O(N)$ and dominates the second and third terms, which are of $O(\log N)$. Therefore, we get:

$$\lim_{N\to\infty} \log \frac{P(D_{X_i}|\boldsymbol{Pa}^{*,1}(X_i))}{P(D_{X_i}|\boldsymbol{Pa}^{*,3}(X_i))} = \lim_{N\to\infty} N\cdot[-H_{X_i|\boldsymbol{Pa}^{*,1}(X_i)} + H_{X_i|\boldsymbol{Pa}^{*,3}(X_i)}]. \tag{4.37}$$

Using the combined parent set $\boldsymbol{Pa}^{*,3}(X_i)$ implies that the probability distribution for all instantiations of the variables in $\boldsymbol{Y}_i = \boldsymbol{Pa}^{*,3}(X_i) \backslash \boldsymbol{Pa}'_{IS}(X_i)$ are the same according to the CSI structure encoded in $\boldsymbol{Pa}'_{IS}(X_i)$; however, they are not all the same according to the data-generating model. Therefore, the entropy of X_i given the combined parent set in case 3 (i.e., $H_{X_i|\boldsymbol{Pa}^{*,3}(X_i)}$) will increase compared to X_i's entropy given the data-generating combined parents in case 1 (i.e., $H_{X_i|\boldsymbol{Pa}^{*,1}(X_i)}$) due to information loss. As a result, the term $-H_{X_i|\boldsymbol{Pa}^{*,1}(X_i)} + H_{X_i|\boldsymbol{Pa}^{*,3}(X_i)}$ in Equation (4.37) becomes a positive number; thus, Equation

[3] The result holds for both case 1a and case 1b, and thus, we do not make a distinction in our discussion here.

(4.37) becomes ∞ as $N \to \infty$. This result holds regardless of whether $\boldsymbol{U}_T$ does include CSI structure (case 1a) or not (case 1b).

□

4.5 Experimental Results

In this section, we evaluate the performance of the instance-specific structure discovery algorithm, IGES, versus a state-of-the-art population-wide method, GES. We applied these two algorithms on both simulated and real-world datasets.

4.5.1 Simulated data

To generate simulated data, we randomly generated Bayesian networks that are used to simulate data by applying the following steps:

1. For each Bayesian network $\mathcal{M} = (\mathcal{G}, \boldsymbol{\Theta})$, we first created a DAG $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$ with $|\boldsymbol{V}| = \{10, 20, 50\}$ discrete random variables and $|\boldsymbol{E}| = \{2|\boldsymbol{V}|, 4|\boldsymbol{V}|, 6|\boldsymbol{V}|\}$ expected edge densities. In order to generate $\mathcal{G}$, we created an arbitrary ordering of variables [4]. Then, we uniformly randomly added edges to $\mathcal{G}$ in a forward direction until obtaining the specified number of edges. The DAGs generated in this way have a power-law-type distribution over the number of parents, with some variables having many more than the average number of parents.
2. We then parametrized the distribution of each random variable $X \in \boldsymbol{V}$ given its parents $\boldsymbol{Pa}(X)$ according to DAG $\mathcal{G}$. Each discrete variable X may have 2, 3, or 4 categories, which is chosen randomly. Given the number of categories of X and its parents $\boldsymbol{Pa}(X)$, we randomly initialized the conditional probability table for $P(X|\boldsymbol{Pa}(X))$ under the constraints that follow from the axioms of probability theory. We also included context-specific independencies (CSIs) in the CPTs so that each node that has more than one parent includes at least one CSI relationship. CSI parents generated this way are a

[4] This ordering is only used to generate the BNs; we do not use it when applying GES or IGES.

proper subset of the population-wide parents in the data-generating model. In the CBNs with the edge density of $2|\boldsymbol{V}|$, $4|\boldsymbol{V}|$, and $6|\boldsymbol{V}|$ about 28%, 38%, and 48% of the variables (on average) exhibit CSI in each simulated test case T, respectively.

3. Given the randomly generated CBN $\mathcal{M} = (\mathcal{G}, \boldsymbol{\Theta})$, we simulated a training dataset D with $N = \{200, 1000, 5000\}$ training samples.
4. We also generated $M = 500$ test instances from the randomly generated CBN $\mathcal{M} = (\mathcal{G}, \boldsymbol{\Theta})$; we refer to each test instance as T.
5. We used the training set D generated in step 3 along with each of the 500 test instances generated in step 4 to learn 500 instance-specific CBN structures for each test instance T using IGES (Algorithm 10); we denote this CBN structure by $\mathcal{G}_{IS}$[5]. We also used D to learn a single population-wide CBN structure for all test instances using the GES method (Algorithm 9); we refer to this CBN structure as $\mathcal{G}_{PW}$. We used a prior equivalence sample size of PESS = 1.0 for both IGES and GES methods.
6. Finally, we computed evaluation measures (described below) to compare the structure recovery performance of IGES versus GES. To do so, we obtained the ground-truth pattern for each test instance T considering the existing CSIs associated with T (steps 1 and 2); we refer to this graph as $\mathcal{G}_{truth}$. We compared $\mathcal{G}_{PW}$ and $\mathcal{G}_{IS}$ versus $\mathcal{G}_{truth}$ for each test case and reported the average of measures over the $M = 500$ test cases.

We repeated the above steps 10 times and computed the average of the evaluation measures (defined below) over those runs.

4.5.1.1 Pattern structure discovery performance measures In this section, I describe the evaluation measures that are used to calculate the structural similarity of a discovered pattern $\mathcal{G}_{output}$, which is $\mathcal{G}_{PW}$ when using GES and $\mathcal{G}_{IS}$ when using IGES, versus the ground-truth pattern $\mathcal{G}_{truth}$, which is derived for a given test instance T. One such measure is structural Hamming distance (SHD) that counts the edge modifications, which can include added, deleted, and reoriented edges, by comparing each possible edge in $\mathcal{G}_{output}$ and $\mathcal{G}_{truth}$. We define two versions of SHD as follows:

[5]IGES outputs both $\mathcal{G}_{IS}$ and $\mathcal{G}_{PW}$ for completeness, but $\mathcal{G}_{IS}$ is what it actually learns as the instance-specific CBN structure for a given instance T.

- **Strict SHD (S-SHD)**: This version counts any edge modifications, which are added, deleted, and reoriented edges. The S-SHD would be 0 if two edges are exactly the same; otherwise, it is 1. Any extra or missing edge would also count as 1 in terms of S-SHD. Table 22 shows how to compute S-SHD for patterns.
- **Adjacency SHD (A-SHD)**: In this version, we compute SHD on the skeleton-level by comparing the adjacencies of two graphs, which disregards the edge orientations and only counts the edge modifications of the adjacency graph that includes added and deleted edges. For example, if one graph includes $A \to B$ but there is no edge between A and B in the other one, then A-SHD would be 1.

Table 22: Strict SHD (S-SHD) for patterns. The rows and columns correspond to the edge types output by the algorithm and the data-generating edge types, respectively.

Output Edge/ Truth Edge	$A \to B$	A — B	A B
$A \to B (B \to A)$	0 (1)	1	1
A — B	1	0	1
A B	1	1	0

Other performance criteria we used to evaluate discrimination are precision (P) and recall (R) for adjacencies and arrowheads as follows:

- **Adjacency precision (AP)**: we compute the ratio of correctly predicted edges in $\mathcal{G}_{output}$ to all predicted edges in $\mathcal{G}_{output}$ (without considering orientations of edges) as follows:

$$\text{AP} = \frac{\#\text{correctly predicted adjacencies}}{\#\text{predicted adjacencies}} \tag{4.38}$$

- **Adjacency recall (AR)**: we compute the ratio of correctly predicted edges in $\mathcal{G}_{output}$ to all true edges in $\mathcal{G}_{truth}$ (without considering the edges' orientations) as follows:

$$\text{AR} = \frac{\#\text{correctly predicted adjacencies}}{\#\text{true adjacencies}} \tag{4.39}$$

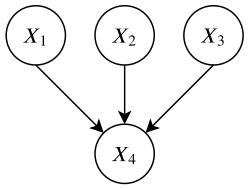
- **Arrowhead precision (AHP)**: considering the pairs of variables that have an edge between them in the predicted graph $\mathcal{G}_{output}$, we compute the ratio of correctly predicted arrowheads in $\mathcal{G}_{output}$ to all predicted arrowheads in $\mathcal{G}_{output}$ as follows:

$$\text{AHP} = \frac{\#\text{correctly predicted arrowheads}}{\#\text{predicted arrowheads}} \tag{4.40}$$

- **Arrowhead recall (AHR)**: considering the pairs of variables that have an edge between them in the ground-truth graph $\mathcal{G}_{truth}$, we compute the ratio of correctly predicted arrowheads in $\mathcal{G}_{output}$ to all true arrowheads in $\mathcal{G}_{truth}$ as follows:

$$\text{AHR} = \frac{\#\text{correctly predicted arrowheads}}{\#\text{true arrowheads}} \tag{4.41}$$

Since we are evaluating methods using data that have been generated by instance-specific models, the ground-truth CBN $\mathcal{G}_{truth}$ is derived based on the given test instance T. There are two possibilities: nodes that include context-specific independence (CSI), for which we derive precision (P_{IS}) and recall statistics(R_{IS}), and nodes that do not include CSI, for which we derive separate precision (P_{other}) and recall (R_{other}) statistics. We also combine these two types of nodes to derive overall precision (P) and recall (R) statistics. Consider the CBN example in Figure 13. Given a test instance $T = \{X_1 = 0, X_2 = 0, X_3 = 0, X_4 = 0\}$, in which the CSI relationship $X_4 \perp\!\!\!\perp_c \{X_2, X_3\}|X_1 = 0$ holds, X_4 is considered in the P_{IS} and R_{IS} calculations. However, in another test instance $T = \{X_1 = 1, X_2 = 0, X_3 = 0, X_4 = 0\}$, which does not encode any CSI relationship, X_4 will be considered in the P_{other} and R_{other} calculations. Both test instances are used in deriving P and R. As this example demonstrates, the ground-truth for each node is therefore either an instance-specific structure if it includes CSI (which can vary with the instance) or a population-wide structure if it does not include CSI (which does not vary). The predicted parent set for X_i is the population-wide parents of X_i in $\mathcal{G}_{PW}$ when using the GES algorithm, and it is the instance-specific parents of X_i in $\mathcal{G}_{IS}$ when using IGES.



X_1	X_2	X_3	$P(X_4 \mid X_1, X_2, X_3)$
0	0	0	0.90
0	0	1	0.90
0	1	0	0.90
0	1	1	0.90
1	0	0	0.73
1	0	1	0.46
1	1	0	0.15
1	1	1	0.15

Figure 13: This CBN example contains two context-specific independence (CSI) structures: $X_4 \perp\!\!\!\perp_c \{X_2, X_3\} | X_1 = 0$ and $X_4 \perp\!\!\!\perp_c X_3 | \{X_1 = 1, X_2 = 1\}$.

4.5.1.2 Simulation results Tables 23, 24, and 25 show the average adjacency precision and recall of the instance-specific CBNs found by IGES ($\kappa = 0.1$)[6] and population-wide CBN found by GES over the randomly generated CBNs described above, using $N = \{200, 1000, 5000\}$ training instances, respectively. As shown in these tables, when $N = 200$, both IGES and GES methods perform similarly in terms of adjacency precision, while IGES has a slightly lower adjacency recall. As the sample size increases to $N = 1000$, both methods perform better in terms of adjacency recall. However, the adjacency precision performance of GES often decreases, while the adjacency precision performance of IGES ($\kappa = 0.1$) slightly increased or remained the same. Increasing the training set size to $N = 5000$ results in even better adjacency recall for both methods; however, GES often loses more adjacency precision compared to IGES, especially for the CBNs with more variables and edges. For example, in CBNs with 50 variables and 300 edges, GES adjacency precision decreases from 0.93 ($N = 200$ cases) to 0.77 ($N = 5000$ cases). However, IGES ($\kappa = 0.1$) adjacency precision decreases from 0.93 ($N = 200$ cases) to 0.90 ($N = 5000$ cases), while IGES and GES perform similarly in terms of adjacency recall (~ 0.50). In most cases, $\kappa = 0.1$ gives the best results for the IGES method.

Tables 26, 27, and 28 show the average arrowhead precision and recall of the instance-specific CBNs and population-wide CBN over 10 randomly generated CBNs described above, using $N = \{200, 1000, 5000\}$ training instances, respectively. As these tables demonstrate,

[6]Results using different values of $\kappa = \{0.001, 0.1, 0.5, 0.9\}$ values are reported in Appendix A.

when using $N = 200$ training instances, both IGES ($\kappa = 0.1$) and GES methods perform similarly in terms of arrowhead precision and recall for CBNs with 10 variables but IGES performs better, especially in terms of arrowhead precision, for CBNs with 20 and 50 variables. As the sample size increases to $N = 1000$ and $N = 5000$ cases, both methods perform better in terms of arrowhead precision and recall, but IGES outperforms GES in terms of arrowhead precision, without hurting the arrowhead recall. For example, when using $N = 5000$ cases for CBNs with 50 variables and 200 edges, arrowhead precision of IGES ($\kappa = 0.1$) increased to 0.74 (compared to 0.61 with $N = 200$ training instances) and arrowhead precision of GES became 0.61 (compared to 0.58 with $N = 200$ training instances), while both methods perform similar in terms of arrowhead recall ~ 0.50 (compared to ~ 0.10 with $N = 200$).

We also computed the structural Hamming distance (SHD) to compare the performance of the search procedures on each given instance T. As described in Section 4.5.1.1, the SHD between two graphs (patterns in the case of IGES and GES algorithms) is composed of three edge modifications: added, deleted, and reversed edges, which we refer to as strict SHD (S-SHD). We also computed the adjacency SHD (A-SHD) that only counts the number of added and deleted edges on the skeletons of the graphs. Tables 29, 30, and 31 show the average results on the IGES and GES methods when using $N = \{200, 1000, 5000\}$ training samples, respectively. In these experiments, when using $N = 200$ training instances, the average S-SHD is similar using both IGES and GES methods. By increasing the training samples to $N = 1000$ and $N = 5000$, both methods perform better in terms of A-SHD and S-SHD; however, IGES performs notably better, especially in CBNs with more variables and edges. This improvement is mainly due to fewer number of added edges, especially in the nodes with CSI structure (denoted by *IS* in tables). Based on these simulations, the IGES algorithm often results in less erroneously added and reversed edges but more deleted edges when compared to the GES method.

Table 23: Adjacency precision (P) and recall (R) results for $N = 200$ training instances. For the IGES method, a penalty factor $\kappa = 0.1$ is used to penalize the structural difference between the population-wide and instance-specific CBNs. The numbers after '±' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGES	0.73 ± 0.14	0.94 ± 0.10	0.88 ± 0.09	0.43 ± 0.14	0.42 ± 0.12	0.42 ± 0.10
		GES	0.75 ± 0.16	0.97 ± 0.05	0.89 ± 0.07	**0.60 ± 0.18**	**0.50 ± 0.11**	**0.54 ± 0.08**
	40	IGES	0.79 ± 0.10	0.89 ± 0.10	0.87 ± 0.06	0.29 ± 0.07	0.26 ± 0.08	0.27 ± 0.07
		GES	0.76 ± 0.13	0.90 ± 0.12	0.84 ± 0.11	**0.35 ± 0.06**	**0.32 ± 0.07**	**0.33 ± 0.05**
	60	IGES	0.85 ± 0.08	0.92 ± 0.13	0.94 ± 0.05	0.38 ± 0.11	0.23 ± 0.08	0.30 ± 0.09
		GES	0.85 ± 0.09	**0.99 ± 0.02**	0.92 ± 0.07	**0.43 ± 0.09**	**0.29 ± 0.07**	**0.34 ± 0.06**
20	40	IGES	0.83 ± 0.10	0.95 ± 0.07	0.89 ± 0.07	0.44 ± 0.14	0.37 ± 0.06	0.39 ± 0.08
		GES	0.81 ± 0.10	**0.97 ± 0.05**	0.89 ± 0.08	**0.54 ± 0.10**	**0.48 ± 0.10**	**0.49 ± 0.08**
	80	IGES	0.87 ± 0.07	0.92 ± 0.12	0.90 ± 0.05	0.35 ± 0.07	0.24 ± 0.08	0.29 ± 0.07
		GES	0.85 ± 0.07	**0.96 ± 0.09**	0.91 ± 0.05	**0.40 ± 0.06**	0.26 ± 0.07	**0.34 ± 0.04**
	120	IGES	0.89 ± 0.09	0.96 ± 0.04	0.93 ± 0.06	0.28 ± 0.07	0.19 ± 0.07	0.23 ± 0.07
		GES	0.88 ± 0.10	**0.99 ± 0.02**	0.92 ± 0.07	**0.33 ± 0.05**	**0.23 ± 0.05**	**0.28 ± 0.05**
50	100	IGES	0.85 ± 0.04	0.92 ± 0.05	0.88 ± 0.04	0.43 ± 0.07	0.35 ± 0.05	0.38 ± 0.05
		GES	0.86 ± 0.04	**0.98 ± 0.03**	**0.92 ± 0.03**	**0.47 ± 0.06**	**0.41 ± 0.06**	**0.44 ± 0.04**
	200	IGES	0.86 ± 0.05	0.93 ± 0.03	0.89 ± 0.04	0.31 ± 0.04	0.20 ± 0.03	0.24 ± 0.03
		GES	0.88 ± 0.03	**0.98 ± 0.03**	**0.92 ± 0.03**	**0.35 ± 0.02**	0.21 ± 0.01	**0.27 ± 0.01**
	300	IGES	0.89 ± 0.04	0.97 ± 0.03	0.92 ± 0.03	0.25 ± 0.07	0.15 ± 0.03	0.19 ± 0.05
		GES	**0.92 ± 0.02**	**0.99 ± 0.02**	**0.95 ± 0.02**	**0.30 ± 0.05**	**0.19 ± 0.04**	**0.24 ± 0.04**
Summary statistics		IGES	0.84 ± 0.05	0.93 ± 0.02	0.90 ± 0.02	0.35 ± 0.07	0.27 ± 0.09	0.30 ± 0.07
		GES	0.84 ± 0.05	0.97 ± 0.03	0.91 ± 0.03	0.42 ± 0.10	0.32 ± 0.11	0.36 ± 0.10

Table 24: Adjacency precision (P) and recall (R) results for $N = 1000$ training instances. For the IGES method, a penalty factor $\kappa = 0.1$ is used to penalize the structural difference between the population-wide and instance-specific CBNs. The numbers after '±' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGES	**0.82 ± 0.10**	0.93 ± 0.06	**0.89 ± 0.06**	0.72 ± 0.13	0.63 ± 0.12	0.65 ± 0.11
		GES	0.72 ± 0.11	0.93 ± 0.06	0.83 ± 0.07	**0.83 ± 0.11**	**0.70 ± 0.14**	**0.73 ± 0.12**
	40	IGES	**0.83 ± 0.07**	0.97 ± 0.04	0.88 ± 0.05	0.53 ± 0.12	0.40 ± 0.06	0.46 ± 0.09
		GES	0.77 ± 0.08	**0.98 ± 0.03**	0.85 ± 0.05	**0.60 ± 0.07**	0.47 ± 0.10	0.53 ± 0.07
	60	IGES	0.78 ± 0.10	0.96 ± 0.08	0.87 ± 0.07	0.43 ± 0.10	0.36 ± 0.09	0.38 ± 0.09
		GES	0.72 ± 0.12	0.96 ± 0.08	0.84 ± 0.09	**0.48 ± 0.10**	**0.43 ± 0.11**	**0.45 ± 0.09**
20	40	IGES	**0.88 ± 0.09**	0.93 ± 0.06	**0.91 ± 0.07**	0.69 ± 0.08	0.58 ± 0.07	0.61 ± 0.06
		GES	0.77 ± 0.09	0.96 ± 0.05	0.87 ± 0.05	**0.76 ± 0.11**	0.61 ± 0.04	**0.66 ± 0.05**
	80	IGES	**0.88 ± 0.06**	0.96 ± 0.05	**0.91 ± 0.05**	0.50 ± 0.05	0.40 ± 0.07	0.45 ± 0.05
		GES	0.73 ± 0.09	**0.98 ± 0.05**	0.82 ± 0.07	0.53 ± 0.06	0.42 ± 0.07	0.48 ± 0.06
	120	IGES	**0.85 ± 0.06**	0.92 ± 0.06	0.88 ± 0.06	0.46 ± 0.05	0.32 ± 0.06	0.39 ± 0.06
		GES	0.79 ± 0.07	0.94 ± 0.06	0.85 ± 0.06	**0.49 ± 0.05**	0.33 ± 0.04	0.42 ± 0.04
50	100	IGES	**0.88 ± 0.05**	0.96 ± 0.01	**0.93 ± 0.02**	0.68 ± 0.05	0.61 ± 0.07	0.64 ± 0.05
		GES	0.74 ± 0.05	**0.99 ± 0.01**	0.87 ± 0.03	**0.73 ± 0.05**	0.63 ± 0.07	**0.67 ± 0.04**
	200	IGES	**0.88 ± 0.03**	0.93 ± 0.04	**0.91 ± 0.03**	0.52 ± 0.06	**0.38 ± 0.06**	0.44 ± 0.06
		GES	0.79 ± 0.03	**0.95 ± 0.04**	0.86 ± 0.03	0.52 ± 0.05	0.36 ± 0.06	0.43 ± 0.05
	300	IGES	**0.88 ± 0.03**	0.95 ± 0.02	**0.91 ± 0.02**	0.46 ± 0.04	0.31 ± 0.05	0.37 ± 0.04
		GES	0.78 ± 0.03	**0.98 ± 0.03**	0.86 ± 0.03	0.47 ± 0.03	**0.32 ± 0.05**	**0.38 ± 0.04**
Summary statistics		IGES	0.85 ± 0.03	0.95 ± 0.02	0.90 ± 0.02	0.55 ± 0.10	0.44 ± 0.12	0.49 ± 0.11
		GES	0.76 ± 0.03	0.96 ± 0.02	0.85 ± 0.02	0.60 ± 0.13	0.47 ± 0.13	0.53 ± 0.12

Table 25: Adjacency precision (P) and recall (R) results for $N = 5000$ training instances. For the IGES method, a penalty factor $\kappa = 0.1$ is used to penalize the structural difference between the population-wide and instance-specific CBNs. The numbers after '±' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGES	**0.86 ± 0.10**	0.93 ± 0.07	**0.90 ± 0.07**	0.83 ± 0.06	0.83 ± 0.07	0.83 ± 0.05
		GES	0.56 ± 0.08	0.93 ± 0.07	0.75 ± 0.06	**0.92 ± 0.08**	0.90 ± 0.05	**0.90 ± 0.05**
	40	IGES	**0.81 ± 0.11**	0.96 ± 0.05	**0.87 ± 0.07**	0.69 ± 0.08	0.59 ± 0.07	0.63 ± 0.07
		GES	0.60 ± 0.10	0.96 ± 0.07	0.75 ± 0.08	**0.80 ± 0.11**	**0.74 ± 0.10**	**0.76 ± 0.09**
	60	IGES	**0.76 ± 0.11**	0.97 ± 0.04	**0.85 ± 0.07**	0.65 ± 0.08	0.55 ± 0.07	0.59 ± 0.06
		GES	0.57 ± 0.05	0.98 ± 0.05	0.75 ± 0.05	**0.70 ± 0.11**	**0.65 ± 0.10**	**0.67 ± 0.10**
20	40	IGES	**0.84 ± 0.07**	0.93 ± 0.07	**0.89 ± 0.06**	0.82 ± 0.08	0.79 ± 0.05	0.80 ± 0.05
		GES	0.58 ± 0.07	0.92 ± 0.08	0.74 ± 0.06	**0.92 ± 0.04**	**0.86 ± 0.04**	**0.88 ± 0.01**
	80	IGES	**0.84 ± 0.06**	0.92 ± 0.05	**0.88 ± 0.05**	0.64 ± 0.05	0.58 ± 0.07	0.62 ± 0.04
		GES	0.61 ± 0.08	**0.95 ± 0.05**	0.73 ± 0.04	**0.69 ± 0.06**	**0.61 ± 0.07**	**0.66 ± 0.04**
	120	IGES	**0.81 ± 0.04**	0.89 ± 0.05	**0.85 ± 0.04**	0.58 ± 0.07	0.47 ± 0.06	0.52 ± 0.06
		GES	0.62 ± 0.03	0.88 ± 0.07	0.73 ± 0.03	**0.62 ± 0.07**	0.51 ± 0.07	**0.56 ± 0.06**
50	100	IGES	**0.87 ± 0.06**	0.94 ± 0.03	**0.90 ± 0.04**	0.81 ± 0.03	0.79 ± 0.05	0.80 ± 0.03
		GES	0.61 ± 0.05	**0.95 ± 0.04**	0.78 ± 0.05	**0.87 ± 0.04**	0.82 ± 0.06	**0.84 ± 0.04**
	200	IGES	**0.85 ± 0.02**	0.93 ± 0.02	**0.89 ± 0.02**	0.65 ± 0.04	0.55 ± 0.05	0.59 ± 0.05
		GES	0.63 ± 0.02	**0.95 ± 0.02**	0.76 ± 0.02	**0.67 ± 0.05**	0.54 ± 0.06	0.59 ± 0.06
	300	IGES	**0.88 ± 0.04**	0.92 ± 0.03	**0.90 ± 0.03**	0.59 ± 0.04	**0.49 ± 0.04**	0.53 ± 0.04
		GES	0.66 ± 0.04	**0.95 ± 0.03**	0.78 ± 0.04	0.61 ± 0.05	0.47 ± 0.05	0.53 ± 0.04
Summary statistics		IGES	0.83 ± 0.04	0.93 ± 0.02	0.88 ± 0.02	0.70 ± 0.09	0.63 ± 0.13	0.66 ± 0.11
		GES	0.61 ± 0.03	0.94 ± 0.03	0.75 ± 0.02	0.76 ± 0.12	0.68 ± 0.15	0.71 ± 0.13

Table 26: Arrowhead precision (P) and recall (R) results for $N = 200$ training instances. For the IGES method, a penalty factor $\kappa = 0.1$ is used to penalize the structural difference between the population-wide and instance-specific CBNs. The numbers after '±' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGES	0.40 ± 0.16	0.30 ± 0.35	0.34 ± 0.36	0.41 ± 0.15	0.14 ± 0.14	0.14 ± 0.13
		GES	0.37 ± 0.18	0.37 ± 0.37	0.33 ± 0.36	0.41 ± 0.17	0.17 ± 0.18	0.14 ± 0.14
	40	IGES	0.30 ± 0.18	**0.14 ± 0.14**	**0.23 ± 0.19**	**0.25 ± 0.16**	0.04 ± 0.04	**0.07 ± 0.06**
		GES	0.22 ± 0.18	0.12 ± 0.14	0.14 ± 0.17	0.18 ± 0.15	0.04 ± 0.03	0.05 ± 0.05
	60	IGES	0.45 ± 0.23	0.34 ± 0.22	0.40 ± 0.33	**0.38 ± 0.15**	**0.09 ± 0.04**	0.10 ± 0.08
		GES	0.40 ± 0.24	0.27 ± 0.26	0.37 ± 0.33	0.33 ± 0.18	0.07 ± 0.04	0.09 ± 0.07
20	40	IGES	**0.19 ± 0.13**	0.52 ± 0.35	0.49 ± 0.32	0.17 ± 0.14	0.12 ± 0.09	0.13 ± 0.08
		GES	0.13 ± 0.12	0.50 ± 0.39	0.42 ± 0.33	0.13 ± 0.14	0.12 ± 0.10	0.12 ± 0.10
	80	IGES	0.32 ± 0.16	0.44 ± 0.17	0.46 ± 0.18	0.22 ± 0.14	**0.09 ± 0.06**	**0.11 ± 0.07**
		GES	0.26 ± 0.18	0.40 ± 0.25	0.42 ± 0.26	0.18 ± 0.17	0.07 ± 0.06	0.09 ± 0.08
	120	IGES	**0.38 ± 0.22**	**0.45 ± 0.29**	**0.44 ± 0.25**	0.22 ± 0.15	0.08 ± 0.05	0.10 ± 0.06
		GES	0.30 ± 0.23	0.39 ± 0.31	0.35 ± 0.27	0.18 ± 0.13	0.07 ± 0.05	0.09 ± 0.06
50	100	IGES	0.34 ± 0.21	0.62 ± 0.25	0.58 ± 0.23	**0.14 ± 0.09**	**0.13 ± 0.05**	**0.14 ± 0.05**
		GES	0.27 ± 0.16	0.69 ± 0.32	0.58 ± 0.27	0.11 ± 0.08	0.11 ± 0.05	0.11 ± 0.05
	200	IGES	0.53 ± 0.12	0.63 ± 0.17	0.61 ± 0.13	**0.25 ± 0.09**	**0.08 ± 0.03**	**0.11 ± 0.04**
		GES	0.50 ± 0.12	0.67 ± 0.22	0.61 ± 0.16	0.22 ± 0.08	0.07 ± 0.03	0.09 ± 0.03
	300	IGES	0.57 ± 0.14	0.59 ± 0.18	0.66 ± 0.12	**0.16 ± 0.12**	0.04 ± 0.03	**0.06 ± 0.05**
		GES	0.56 ± 0.11	0.67 ± 0.22	0.72 ± 0.15	0.12 ± 0.07	0.03 ± 0.02	0.04 ± 0.03
Summary statistics		IGES	0.39 ± 0.11	0.45 ± 0.15	0.47 ± 0.13	0.24 ± 0.09	0.09 ± 0.03	0.11 ± 0.03
		GES	0.33 ± 0.13	0.45 ± 0.19	0.44 ± 0.16	0.20 ± 0.10	0.08 ± 0.04	0.09 ± 0.03

Table 27: Arrowhead precision (P) and recall (R) results for $N = 1000$ training instances. For the IGES method, a penalty factor $\kappa = 0.1$ is used to penalize the structural difference between the population-wide and instance-specific CBNs. The numbers after '±' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGES	0.52 ± 0.21	0.57 ± 0.21	**0.55 ± 0.17**	0.53 ± 0.20	0.38 ± 0.09	0.40 ± 0.10
		GES	0.36 ± 0.22	0.49 ± 0.24	0.43 ± 0.13	0.42 ± 0.22	0.37 ± 0.15	0.40 ± 0.14
	40	IGES	0.42 ± 0.20	**0.52 ± 0.15**	0.47 ± 0.16	0.47 ± 0.20	**0.27 ± 0.09**	**0.29 ± 0.10**
		GES	0.38 ± 0.20	0.44 ± 0.21	0.44 ± 0.20	0.46 ± 0.18	0.21 ± 0.10	0.25 ± 0.11
	60	IGES	0.31 ± 0.17	0.41 ± 0.17	0.39 ± 0.17	0.30 ± 0.18	**0.17 ± 0.10**	0.19 ± 0.11
		GES	0.23 ± 0.12	0.36 ± 0.18	0.32 ± 0.19	0.29 ± 0.20	0.15 ± 0.10	0.16 ± 0.12
20	40	IGES	**0.66 ± 0.15**	0.68 ± 0.15	**0.68 ± 0.14**	0.59 ± 0.17	0.41 ± 0.12	0.44 ± 0.12
		GES	0.42 ± 0.15	0.66 ± 0.18	0.58 ± 0.13	0.54 ± 0.22	0.40 ± 0.11	0.43 ± 0.12
	80	IGES	**0.53 ± 0.13**	0.79 ± 0.12	**0.73 ± 0.11**	0.53 ± 0.17	0.27 ± 0.07	0.32 ± 0.07
		GES	0.35 ± 0.08	0.82 ± 0.16	0.60 ± 0.09	0.58 ± 0.12	0.25 ± 0.08	0.30 ± 0.08
	120	IGES	**0.43 ± 0.10**	0.66 ± 0.13	**0.58 ± 0.10**	0.45 ± 0.12	0.20 ± 0.05	0.24 ± 0.05
		GES	0.36 ± 0.11	0.67 ± 0.18	0.54 ± 0.13	0.47 ± 0.18	0.19 ± 0.04	0.23 ± 0.05
50	100	IGES	**0.60 ± 0.12**	0.84 ± 0.10	**0.79 ± 0.07**	0.49 ± 0.14	**0.43 ± 0.11**	0.44 ± 0.10
		GES	0.31 ± 0.08	0.86 ± 0.11	0.63 ± 0.09	0.53 ± 0.17	0.41 ± 0.11	0.43 ± 0.11
	200	IGES	**0.64 ± 0.16**	0.77 ± 0.10	**0.73 ± 0.11**	0.57 ± 0.09	**0.28 ± 0.04**	**0.33 ± 0.05**
		GES	0.46 ± 0.08	0.75 ± 0.13	0.62 ± 0.09	0.57 ± 0.06	0.24 ± 0.03	0.29 ± 0.04
	300	IGES	**0.66 ± 0.06**	0.77 ± 0.08	**0.73 ± 0.07**	0.54 ± 0.12	**0.21 ± 0.05**	0.26 ± 0.06
		GES	0.48 ± 0.05	0.78 ± 0.10	0.65 ± 0.07	0.51 ± 0.09	0.19 ± 0.05	0.24 ± 0.05
Summary statistics		IGES	0.53 ± 0.12	0.67 ± 0.14	0.63 ± 0.13	0.50 ± 0.08	0.29 ± 0.09	0.32 ± 0.08
		GES	0.37 ± 0.07	0.65 ± 0.17	0.53 ± 0.11	0.49 ± 0.09	0.27 ± 0.09	0.30 ± 0.09

Table 28: Arrowhead precision (P) and recall (R) results for $N = 5000$ training instances. For the IGES method, a penalty factor $\kappa = 0.1$ is used to penalize the structural difference between the population-wide and instance-specific CBNs. The numbers after '±' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGES	**0.56 ± 0.22**	0.70 ± 0.25	**0.66 ± 0.21**	0.59 ± 0.21	0.70 ± 0.11	0.68 ± 0.13
		GES	0.23 ± 0.18	0.69 ± 0.26	0.49 ± 0.14	0.52 ± 0.30	0.74 ± 0.14	0.72 ± 0.16
	40	IGES	**0.53 ± 0.26**	0.56 ± 0.22	**0.54 ± 0.21**	0.60 ± 0.28	0.41 ± 0.16	0.45 ± 0.16
		GES	0.27 ± 0.16	0.54 ± 0.28	0.42 ± 0.17	0.58 ± 0.23	0.43 ± 0.22	0.49 ± 0.20
	60	IGES	**0.32 ± 0.16**	0.44 ± 0.19	**0.41 ± 0.18**	0.45 ± 0.15	0.30 ± 0.07	0.32 ± 0.08
		GES	0.16 ± 0.09	0.42 ± 0.23	0.34 ± 0.16	0.43 ± 0.23	0.29 ± 0.09	0.32 ± 0.09
20	40	IGES	**0.59 ± 0.25**	0.75 ± 0.17	**0.70 ± 0.16**	0.72 ± 0.17	0.66 ± 0.10	0.68 ± 0.09
		GES	0.28 ± 0.09	0.72 ± 0.20	0.52 ± 0.10	0.78 ± 0.12	0.70 ± 0.08	**0.73 ± 0.08**
	80	IGES	**0.53 ± 0.14**	0.75 ± 0.13	**0.68 ± 0.11**	0.68 ± 0.12	**0.47 ± 0.06**	0.50 ± 0.06
		GES	0.28 ± 0.07	0.73 ± 0.16	0.51 ± 0.07	**0.74 ± 0.12**	0.44 ± 0.07	0.49 ± 0.07
	120	IGES	**0.52 ± 0.15**	**0.66 ± 0.13**	**0.61 ± 0.10**	0.58 ± 0.09	0.35 ± 0.04	0.39 ± 0.05
		GES	0.34 ± 0.10	0.60 ± 0.13	0.49 ± 0.06	**0.65 ± 0.12**	0.34 ± 0.07	0.39 ± 0.07
50	100	IGES	**0.63 ± 0.16**	0.81 ± 0.08	**0.76 ± 0.08**	0.77 ± 0.08	0.69 ± 0.07	0.71 ± 0.07
		GES	0.29 ± 0.06	0.78 ± 0.09	0.58 ± 0.07	**0.85 ± 0.09**	0.69 ± 0.08	0.72 ± 0.08
	200	IGES	**0.59 ± 0.10**	0.80 ± 0.06	**0.74 ± 0.06**	0.79 ± 0.06	**0.47 ± 0.05**	**0.51 ± 0.05**
		GES	0.32 ± 0.04	0.79 ± 0.05	0.58 ± 0.05	0.80 ± 0.06	0.44 ± 0.06	0.49 ± 0.06
	300	IGES	**0.67 ± 0.10**	0.76 ± 0.09	**0.73 ± 0.09**	0.76 ± 0.08	**0.39 ± 0.05**	**0.44 ± 0.05**
		GES	0.39 ± 0.03	0.76 ± 0.11	0.59 ± 0.06	**0.79 ± 0.07**	0.36 ± 0.05	0.43 ± 0.05
Summary statistics		IGES	0.55 ± 0.09	0.69 ± 0.11	0.65 ± 0.11	0.66 ± 0.10	0.49 ± 0.15	0.52 ± 0.13
		GES	0.28 ± 0.06	0.67 ± 0.12	0.50 ± 0.08	0.68 ± 0.14	0.49 ± 0.16	0.53 ± 0.15

Table 29: Adjacency SHD (A-SHD) and strict SHD (S-SHD) for $N = 200$ training instances. For the IGES method, a penalty factor $\kappa = 0.1$ is used to penalize the structural difference between the population-wide and instance-specific CBNs. The best result for each setting (e.g., 10 variables and 20 nodes) is shown in bold (the lower the better).

			Added			Deleted			Reoriented				
# Variables	# Edges	Method	IS	Other	Overall	IS	Other	Overall	IS	Other	Overall	A-SHD	S-SHD
10	20	IGES	**0.76**	0.15	**0.91**	2.87	5.56	8.43	**0.95**	**1.44**	**2.40**	9.34	11.73
		GES	0.91	**0.13**	1.04	**2.03**	**4.79**	**6.83**	1.53	1.86	3.39	**7.86**	**11.25**
	40	IGES	**0.69**	**0.31**	**0.99**	6.78	10.60	17.38	**1.62**	**2.31**	**3.93**	18.37	**22.30**
		GES	1.16	0.43	1.59	**6.12**	**9.85**	**15.97**	2.32	2.95	5.27	**17.55**	22.82
	60	IGES	**0.52**	0.03	**0.55**	6.53	12.14	18.66	**1.75**	**1.95**	**3.70**	19.22	**22.92**
		GES	0.80	0.03	0.83	**6.02**	**11.45**	**17.47**	2.22	2.58	4.79	**18.30**	23.10
20	40	IGES	**1.04**	0.31	**1.36**	5.82	11.47	17.29	**2.13**	**2.64**	**4.77**	18.65	23.42
		GES	1.52	**0.27**	1.79	**4.83**	**9.50**	**14.33**	2.94	4.10	7.04	**16.12**	**23.16**
	80	IGES	**1.32**	0.30	**1.62**	15.13	18.44	33.57	**3.78**	**2.94**	**6.71**	35.20	**41.91**
		GES	1.69	**0.06**	1.75	**13.88**	**17.71**	**31.59**	4.64	4.05	8.69	**33.34**	42.03
	120	IGES	**1.12**	0.11	**1.23**	17.88	25.95	43.83	**4.04**	**2.49**	**6.53**	45.07	51.60
		GES	1.30	**0.06**	1.35	**16.71**	**24.52**	**41.22**	5.15	3.74	8.88	**42.58**	**51.46**
50	100	IGES	**2.17**	1.38	3.56	14.97	27.35	42.32	**4.76**	**6.85**	**11.61**	45.88	57.49
		GES	2.21	**0.41**	**2.62**	**13.93**	**24.70**	**38.62**	6.35	9.03	15.39	**41.24**	**56.63**
	200	IGES	2.82	1.05	3.86	35.52	60.02	95.53	**7.34**	**6.08**	**13.43**	99.40	**112.82**
		GES	**2.46**	**0.36**	**2.82**	**33.74**	**58.49**	**92.23**	9.31	8.49	17.80	**95.05**	112.85
	300	IGES	2.04	0.47	2.51	46.35	68.70	115.05	**6.91**	**5.98**	**12.89**	117.56	130.45
		GES	**1.58**	**0.17**	**1.76**	**43.20**	**65.58**	**108.78**	10.21	9.16	19.37	**110.53**	**129.90**
Summary statistics		IGES	1.39	0.46	1.84	16.87	26.69	43.56	3.70	3.63	7.33	45.41	52.74
		GES	1.51	0.21	1.73	15.61	25.18	40.78	4.96	5.11	10.07	42.51	52.58

Table 30: Adjacency SHD (A-SHD) and strict SHD (S-SHD) for $N = 1000$ training instances. For the IGES method, a penalty factor $\kappa = 0.1$ is used to penalize the structural difference between the population-wide and instance-specific CBNs. The best result for each setting (e.g., 10 variables and 20 nodes) is shown in bold (the lower the better).

			Added			Deleted			Reoriented				
# Variables	# Edges	Method	IS	Other	Overall	IS	Other	Overall	IS	Other	Overall	A-SHD	S-SHD
10	20	IGES	**0.72**	0.54	**1.27**	1.41	3.84	5.25	**1.08**	**2.52**	**3.59**	6.52	**10.11**
		GES	1.73	**0.53**	2.26	**0.87**	**3.20**	**4.06**	1.80	3.18	4.98	**6.32**	11.30
	40	IGES	**1.38**	0.12	**1.50**	4.85	7.89	12.73	**1.99**	**2.08**	**4.07**	14.23	**18.30**
		GES	2.08	**0.09**	2.17	**4.19**	**7.24**	**11.42**	2.64	2.84	5.49	**13.60**	19.09
	60	IGES	**1.31**	**0.08**	**1.38**	5.88	10.57	16.45	**1.79**	**3.10**	**4.89**	17.84	**22.73**
		GES	2.11	0.09	2.21	**5.38**	**9.53**	**14.92**	2.33	3.81	6.14	**17.12**	23.26
20	40	IGES	**1.06**	0.94	**2.00**	3.34	8.06	11.40	**1.90**	**2.73**	**4.62**	13.40	**18.03**
		GES	2.42	**0.55**	2.98	**2.61**	**7.48**	**10.09**	2.78	3.35	6.13	**13.06**	19.20
	80	IGES	**1.90**	0.43	**2.33**	12.32	14.62	26.95	**3.55**	**1.84**	**5.39**	**29.27**	**34.67**
		GES	5.04	**0.22**	5.26	**11.70**	**14.19**	**25.89**	5.08	2.11	7.19	31.16	38.35
	120	IGES	**2.19**	0.76	**2.95**	14.64	20.45	35.09	**5.26**	**3.19**	**8.45**	38.03	**46.48**
		GES	3.59	**0.52**	4.11	**13.68**	**20.07**	**33.76**	6.01	3.27	9.28	**37.87**	47.15
50	100	IGES	**2.54**	0.97	**3.51**	8.31	15.65	23.96	**3.60**	**5.12**	**8.72**	**27.47**	**36.19**
		GES	6.51	**0.28**	6.79	**7.01**	**14.90**	**21.91**	6.07	6.02	12.10	28.70	40.80
	200	IGES	**3.74**	1.88	**5.62**	25.55	**43.66**	**69.21**	**7.46**	**6.72**	**14.18**	**74.83**	**89.01**
		GES	7.74	**1.23**	8.97	**25.20**	44.99	70.19	9.76	7.70	17.46	79.15	96.61
	300	IGES	**3.76**	1.57	**5.33**	32.50	59.91	92.42	**7.22**	**6.44**	**13.65**	**97.74**	**111.40**
		GES	7.83	**0.74**	8.57	**31.87**	**58.69**	**90.56**	9.27	8.00	17.27	99.13	116.40
Summary statistics		IGES	2.06	0.81	2.87	12.09	20.52	32.61	3.76	3.75	7.51	35.48	42.99
		GES	4.34	0.47	4.81	11.39	20.03	31.42	5.08	4.48	9.56	36.24	45.80

Table 31: Adjacency SHD (A-SHD) and strict SHD (S-SHD) for $N = 5000$ training instances. For the IGES method, a penalty factor $\kappa = 0.1$ is used to penalize the structural difference between the population-wide and instance-specific CBNs. The best result for each setting (e.g., 10 variables and 20 nodes) is shown in bold (the lower the better).

			Added			Deleted			Reoriented				
# Variables	# Edges	Method	IS	Other	Overall	IS	Other	Overall	IS	Other	Overall	A-SHD	S-SHD
10	20	IGES	**0.88**	0.67	**1.55**	0.81	1.63	2.44	**1.18**	**1.92**	**3.09**	**3.99**	**7.08**
		GES	3.82	**0.64**	4.46	**0.44**	**0.99**	**1.43**	1.93	2.12	4.04	5.89	9.93
	40	IGES	**1.96**	**0.32**	**2.27**	3.02	5.83	8.85	**2.50**	**3.12**	**5.63**	**11.12**	**16.75**
		GES	5.41	0.40	5.80	**1.90**	**3.86**	**5.76**	3.68	4.09	7.77	11.56	19.33
	60	IGES	**2.66**	0.26	**2.92**	3.83	7.59	11.42	**3.05**	**5.60**	**8.65**	**14.34**	**22.99**
		GES	5.76	**0.18**	5.94	**3.20**	**6.09**	**9.29**	3.65	6.81	10.47	15.23	25.70
20	40	IGES	**1.82**	**1.10**	**2.92**	1.87	3.74	5.61	**1.92**	**2.69**	**4.61**	**8.53**	**13.14**
		GES	6.96	1.35	8.32	**0.84**	**2.41**	**3.25**	2.99	3.28	6.26	11.57	17.84
	80	IGES	**3.10**	1.30	**4.40**	8.41	10.08	18.49	**4.15**	**2.78**	**6.93**	**22.89**	**29.82**
		GES	10.56	**0.86**	11.42	**7.33**	**9.31**	**16.64**	6.05	3.25	9.30	28.06	37.36
	120	IGES	**3.51**	**1.61**	**5.13**	10.68	15.76	26.44	**4.42**	**3.68**	**8.10**	**31.56**	**39.66**
		GES	9.33	1.78	11.11	**9.68**	**14.75**	**24.43**	5.35	4.87	10.22	35.54	45.75
50	100	IGES	**3.66**	2.27	**5.93**	5.24	8.86	14.10	**3.69**	**4.53**	**8.23**	**20.03**	**28.26**
		GES	14.95	**1.63**	16.58	**3.49**	**7.85**	**11.33**	7.15	5.69	12.84	27.91	40.75
	200	IGES	**6.11**	3.03	**9.14**	18.56	**32.86**	51.43	**6.50**	**5.79**	**12.29**	**60.57**	**72.86**
		GES	21.11	**1.87**	22.97	**17.38**	33.86	**51.24**	10.05	7.11	17.16	74.22	91.38
	300	IGES	**5.27**	3.28	**8.55**	25.02	**40.92**	**65.95**	**7.06**	**6.96**	**14.02**	**74.50**	**88.52**
		GES	19.53	**1.83**	21.36	**24.17**	42.53	66.69	9.95	7.59	17.53	88.06	105.59
Summary statistics		IGES	3.22	1.54	4.76	8.61	14.14	22.75	3.83	4.12	7.95	27.50	35.45
		GES	10.83	1.17	12.00	7.60	13.52	21.12	5.64	4.98	10.62	33.12	43.74

4.5.2 Real-world data

The main goal of this section is to perform an empirical investigation of instance-specific learning of CBN structures using the IGES algorithm in several real-world biomedical datasets and compare its performance to population-wide CBN structure learning using GES. The data-generating CBN structures are not known for the real-world datasets that we used in this dissertation, as is often the case with such datasets. However, all of them contain a target variable of interest. Therefore, we use predictive performance of target variables when using these models as proxy indicators of causal fidelity [Jabbari et al., 2019, Jabbari et al., 2020]. Although imperfect, such an investigation is nonetheless informative about model fit.

In order to predict the target variable, we first ran the IGES (described in Section 4.4) and GES (described in Section 4.2) methods to construct a BN structure over all variables (i.e., the predictors and target). Then, we obtained the Markov blanket (MB) of the target variable and used it to predict the target variable. The MB of a variable includes the variable's parents, children, and its children's parents. Finally, we calculated the probability distribution of the target variable given its MB, and output the most probable outcome as the prediction. We report several evaluation criteria to measure the effectiveness of instance-specific BN models learned by the IGES algorithm versus the population-wide BN model learned by GES. In particular, as a measure of discrimination, we report the area under the ROC curve (AUROC) when predicting the target variable. We also report the differences between the MB of the target variable in the MB of the target variable found by the instance-specific models compared to the population-wide model.

4.5.2.1 Pneumonia dataset Pneumonia is a type of lung infection that can be caused by bacteria, viruses, or fungi. It is often categorized according to the site of acquisition. Community-acquired pneumonia (CAP) refers to pneumonia acquired outside of the health-care system, which is one of the most frequent and fatal conditions encountered in clinical practice. Pneumonia is among the leading causes of infectious-disease-related death worldwide. It is also among the most common causes of hospitalization of adults and children in

the U.S. [American Thoracic Society, 2019]. Therefore, developing machine learning methods that can accurately predict the outcome in pneumonia patients is an important area of research that can facilitate patient care and clinical decision making.

The dataset we used was collected by the Pneumonia Patient Outcomes Research Team (PORT) from October 1991 to March 1994 at five hospitals in Pittsburgh, Boston, and Halifax, Nova Scotia to identify low-risk patients with community-acquired pneumonia [Fine et al., 1997]. This dataset includes 2287 pneumonia patients, where each patient has 156 clinical variables, out of which we selected the top 40 variables based on univariate feature selection using the mutual information criterion. These variables include demographic information, history information, physical examination, and laboratory and chest X-ray findings. The target variable is called *dire outcome*; it is a binary variable that is set to 1 if any of the following occurred for a patient: (1) death within 30 days of presentation, (2) an initial intensive care unit admission for respiratory failure, respiratory or cardiac arrest, or shock, or (3) the presence of one or more severe complications. The outcome-prediction research reported on the Pneumonia dataset performed under the auspices of study protocol number PRO15030462 from the University of Pittsburgh Institutional Review Board (IRB).

The pneumonia dataset was split into a training set D with $N = 1601$ samples and a test set with $M = 686$ samples while preserving the distribution of dire outcome in the original dataset. Given each instance T, and D, we applied the IGES search using IS-Score to learn an instance-specific BN model for T and used it to predict the outcome for T. We also applied the GES search using the BDeu score to learn a population-wide BN model for T given D and used it to predict the outcome for T. We repeated this procedure for every instance in the dataset. We used prior equivalence sample sizes of PESS $= \{0.1, 1.0, 10.0\}$ for both the IGES and GES methods. IGES also has a parameter that penalizes the structural difference between the population-wide BN and instance-specific BN model, called κ ($0.0 < \kappa \leq 1.0$), where a lower value indicates more penalty for differences; see Section 4.4 for more details. We report the results of using multiple values of κ.

Table 33 shows the AUROC results of GES and IGES on the pneumonia dataset; boldface indicates that the results are statistically significantly better, based on DeLong's nonparametric test [DeLong et al., 1988] at a 0.05 significance level. The results indicate that

Table 32: Type, name, and description of the variables of the pneumonia dataset.

Variable Type	**Variable Name**	**Variable Description**
Demographics	AGE	Age
	PTMRSTA	Marital status
	PTLIVLO	Living arrangements
	PTEMPLO	Emplyment status
Symptoms	COUGHY	Cough
	FEVERY	Fever
	SWEATSY	Sweats
	HEADACHEY	Headache
	CONFSDYY	Confusion
Comorbidities	CADA	Coronary artery disease
	HTNA	Hypertension
	LIVERDIA	Liver disease
	CVDA	Cerebrovascular disease
	LUNGOTA	Pneumonectomy
History	MYEL90A	Myelosuppressive drugs used in the past 90 days (which impair the immune system)
	CSTERDUR	Steriod duration
	DNR	Do-not-resuscitate order status
	CWTLOSS	Weight-loss
Physical Exam	CONFUSA	Confusion noted in chart
Categorized Vitals	CPULSE	Heart rate
	CRESPRAT	Respiratory rate
	CTEMPC	Temperature
	RTEMP	Route temperature taken
	PULRALES	Rales
	PULRHONC	Rhonchi
Categorized Laboratory Results	CWBC,	White blood cell count
	CHCT	Hematocrit
	CHGB	Hemoglobin count
	CGLU	Glucose
	CAN	Sodium
	CHCO3	Bicarbonate
	CBUN	Blood urea nitrogen
	CCR	Creatinine
	CSGOT	Serum glutamic oxaloacetic transaminase
	CALB	Serum albumin
Categorized ABG (Arterial blood gas)	O2SATC	O2 saturation
	CPH	Arterial ph
	CPCO2	pCO2
	CPO2	pO2

with PESS = 0.1 and $\kappa = \{0.2\}$ the IGES search resulted in the highest AUROC but for almost all values of κ, both methods perform similarly.

Table 33: AUROC of the GES and IGES methods on the pneumonia dataset for *dire outcome*. Boldface indicates statistically significantly better results.

Method	**GES**	**IGES**									
PESS	-	$\kappa = 0.1$	$\kappa = 0.2$	$\kappa = 0.3$	$\kappa = 0.4$	$\kappa = 0.5$	$\kappa = 0.6$	$\kappa = 0.7$	$\kappa = 0.8$	$\kappa = 0.9$	$\kappa = 1.0$
0.1	0.73	**0.77**	**0.78**	0.76	0.76	0.76	0.75	0.76	0.75	0.74	0.74
1.0	0.73	**0.77**	**0.77**	0.75	0.75	0.76	0.76	0.75	0.76	0.75	0.75
10.0	0.73	0.76	0.76	0.76	0.75	0.75	0.75	0.75	0.75	0.74	0.74

Table 34a shows the results of comparing the target variable's MB in the instance-specific models versus the population-wide models with PESS = 0.1 and $\kappa = 0.2$. It indicates that in 2.8% of the patient cases, the MB of the target variable in instance-specific CBNs is exactly the same as the the MB of the target variable in the population-wide BN. It also shows that in 7.7% of the patient cases, the MB of the target variable had 20 additional edges in instance-specific CBNs compared to the population-wide BN. Table 34b also shows the percentage of 9 variables that occurred the most in the instance-specific MBs. Table 34 supports that instance-specific structures exist for the cases in the dataset we used.

4.5.2.2 Sepsis dataset Sepsis is the body's severe and toxic response to an infection, which triggers a chain of inflammations that may lead to organ dysfunction and death. Older adults, adults with chronic medical conditions or weaker immune systems, and young children are more susceptible to develop sepsis. Each year, more than 1.7 million patients develop sepsis in the U.S., which costs hospitals more than 20 billion dollars [Singer et al., 2016]. Early and accurate diagnosis of sepsis is essential for reducing its morbidity and mortality; however, it is a challenging task since sepsis presents in multiple ways due to differences in patient genetic background, comorbidities, the microbiology of the infection, and other factors. Therefore, utilizing instance-specific modeling could potentially provide valuable diagnostic and prognostic information.

The sepsis dataset that we used was collected in the Genetic and Inflammatory Markers of Sepsis (GenIMS) project from patients with community-acquired pneumonia in 30 hospitals in southwestern Pennsylvania, Connecticut, Michigan, and Tennessee [Kellum et al., 2007].

Table 34: Comparison of the target variable's Markov blanket (MB) in the instance-specific BNs vs. the population-wide BN in the pneumonia dataset with PESS = 0.1 and $\kappa = 0.2$.

(a) Structural differences of the MBs of the target variable in instance-specific BNs vs. the population-wide BN.

# Added	# Deleted	# Reoriented	% Patients
20	1	0	7.7
21	0	0	4.5
18	1	0	4.4
20	0	0	4.4
23	1	0	3.6
12	1	0	3.2
16	1	0	3.1
0	0	0	2.8
7	1	0	2.8
other	other	other	63.6

(b) Percentage of top-9 variables in the MBs of instance-specific BNs. The MB of the population-wide BN is denoted by *.

Variable Name	% Occurrence in Patients
SWEATSY (Sweats)	80.9
CONFUSA (Confusion noted in chart)	75.7
CALB (Categorized serum albumin)	67.2
FEVERY (Fever)	63.7
CPH (Categorized arterial ph)	58.2
CPO2* (Categorized pO2)	57.1
CPCO2 (Categorized pCO2)	56.1
CGLU (Categorized glucose)	50
O2SATC (Categorized O2 saturation)	44.0

It consists of 1673 patients and 20 predictor variables, including demographic, clinical, inflammatory markers, and genetic variables. The binary target variable is *death* within 90 days of inclusion in the study. The Sepsis data were collected under the auspices of study protocol number PRO15030462 from the University of Pittsburgh Institutional Review Board (IRB).

We performed leave-one-out cross-validation on the sepsis dataset as follows. For each instance T, we used T as a test instance and all the remaining instances as the training set D. Given T and D, we applied IGES search using IS-Score to learn an instance-specific BN model for T to predict its outcome. We also applied the GES search using the BDeu score to learn a population-wide BN model for T to predict its outcome. We repeated this procedure for every instance in the sepsis dataset. We used prior equivalence sample sizes of PESS = $\{0.1, 1.0, 10.0\}$ for both IGES and GES methods. We also report the results of using multiple values of $\kappa (0.0 < \kappa \leq 1.0)$ in IGES; see Section 4.4 for more details.

Table 36 shows the AUROC results on the sepsis dataset, using both GES and IGES searches; boldface indicates that the results are statistically significantly better, based on

Table 35: Type, name, and description of the variables of the sepsis dataset.

Variable Type	**Variable Name**	**Variable Description**
Demographics	Age	Age
	Sex	Sex
	Race	Race
Clinical	time0 psi, day1 psi	Pneumonia severity index (PSI) at time of admission and end of first day of stay. PSI uses 20 clinical variables to classify pneumonia patients into five strata of increased risk for short-term mortality.
	Charlson	Charlson score evaluates comorbidity of patients based on the presence or absence of several medical conditions at admission time.
	Apache day1, Apache day2, Apache day3	APACHE III score on the first, second, and third day of stay. APACHE III is a scoring system that evaluates severity of disease from a number of physiological and clinical parameters.
Inflammatory Markers	IL6-M174	Interleukin-6
	IL10-M1082, IL10-M819	Interleukin-10
Genetic Markers	MIF-M173, MIF-Repeat, TNFA-M376, TNF-M308, rs361525, rs1799724, rs909253	Genetic polymorphisms for the macrophage migration inhibitory factor, the tumor necrosis factor A, the interleukin-6, the interleukin-10, and the heme oxygenase genes

Table 36: AUROC of the GES and IGES methods on the sepsis dataset for *death* outcome. Boldface indicates statistically significantly better results.

Method	**GES**	**IGES**									
PESS	-	$\kappa = 0.1$	$\kappa = 0.2$	$\kappa = 0.3$	$\kappa = 0.4$	$\kappa = 0.5$	$\kappa = 0.6$	$\kappa = 0.7$	$\kappa = 0.8$	$\kappa = 0.9$	$\kappa = 1.0$
0.1	0.56	**0.69**	**0.69**	**0.70**	**0.70**	**0.70**	**0.70**	**0.70**	**0.70**	0.54	0.54
1.0	0.56	**0.69**	**0.69**	**0.77**	**0.78**	**0.77**	**0.78**	**0.78**	**0.78**	**0.78**	**0.74**
10.0	0.71	0.69	0.70	0.75	0.73	0.71	0.72	0.74	0.73	0.74	0.73

DeLong's non-parametric test [DeLong et al., 1988] at a 0.001 significance level. The results indicate that with PESS = 1.0 and $\kappa = \{0.4, 0.6, 0.7, 0.9\}$, the instance-specific search resulted in the highest AUROC; also, for almost all values of κ, IGES performs better.

Table 37a shows the results of comparing the target variable's MB in the instance-specific models versus the population-wide models with PESS = 1.0 and $\kappa = 0.7$. It indicates that in 19.7% of the patient cases, the MB of the target variable had 4 additional and 1 reoriented edges in instance-specific CBNs compared to the population-wide BN. Table 37b shows the percentage of 8 variables that occurred the most in the instance-specific MBs. Table 37 supports that instance-specific structures exist when predicting the target (i.e., *death* within 90 days of inclusion in the study) for the cases in the sepsis dataset.

Table 37: Comparison of the target variable's Markov blanket (MB) in the instance-specific BNs vs. the population-wide BN in the sepsis dataset with PESS = 1.0 and $\kappa = 0.7$.

(a) Structural differences of the variables in the MBs in instance-specific BNs vs. the population-wide BN.

# Added	# Deleted	# Reoriented	% Patients
4	0	1	19.7
2	1	0	18.5
1	1	0	15.8
3	0	0	12.4
3	0	1	8.3
4	0	0	4.5
1	0	1	3.9
other	other	other	16.8

(b) Percentage of the variables in the MBs of instance-specific BNs. The MB of the population-wide BN is denoted by *.

Variable Name	% Occurrence in Patients
MIF-M173	63.4
Charlson	63.3
day1 psi*	60.4
Apache day2	38.0
Age	15.5
Apache day3	15.0
Apache day1	6.3
time0 psi	5.5

4.5.2.3 Lung cancer dataset Lung cancer is the most frequent cause of cancer-related death in men worldwide and the second most common cause in women [Bray et al., 2018], despite significant improvements in diagnosis and treatment during the past decade. The overall 5-year survival rate for lung cancer is 19% but it can be increased to 57% if diagnosis occurs at a localized stage of the disease, which is not often the case [American Cancer Society, 2020]. Studies have revealed that heterogeneity exists both within individual lung cancer tumors

and between patients [Kris et al., 2014, Network et al., 2014, Travis et al., 2011]. Therefore, it is plausible that instance-specific approaches for outcome prediction may perform relatively well [Jabbari et al., 2020].

The lung cancer dataset we used was a retrospective analysis of banked tumor specimens that were collected from patients with lung cancer at the University of Pittsburgh Medical Center (UPMC) in 2016. Baseline demographics, smoking history, staging, treatment, and survival data were collected through the UPMC Network Cancer Registry. We replaced the missing values of the predictor variables with a new category called "missing" and removed the cases for which the value of the outcome variable was not known. Demographic and clinical characteristics of the 261 patients are summarized in Table 38. DNA sequencing was performed using the Ion AmpliSeq^{TM} Cancer Panel (Ion Torrent, Life Technologies, Fisher Scientific). Gene rearrangements of ALK, ROS1, and RET, and MET amplification were detected using FISH. PD-L1 SP263 and PD-L1 22C2 assays were performed on lung cancer samples to determine the PD-L1 tumor proportion score (TPS). Table 39 provides information about the type, name, and description of the variables that are included in the lung cancer dataset. The outcome-prediction research using the lung cancer dataset was performed under the auspices of study protocol number PRO15070164 from the University of Pittsburgh Institutional Review Board (IRB).

Table 40 shows the AUROC results on the lung cancer dataset, using both GES and IGES searches; boldface indicates that the results are statistically significantly better, based on DeLong's non-parametric test [DeLong et al., 1988] at a 0.001 significance level. The results indicate that with PESS = 1.0 and $\kappa = 1.0$, the instance-specific search resulted in the highest AUROC; also, for almost all values of κ, IGES performs better. Table 40 also suggests that it is important to define PESS properly when applying a Bayesian method on a dataset with a small to moderate sample size, which is the case in here.

Table 41a shows the results of comparing the target variable's MB in the instance-specific models versus the population-wide models with PESS = 1.0 and $\kappa = 1.0$. It indicates that in 16.9% of the patient cases, the MB of the target variable was exactly the same in instance-specific and population-wide BNs. Also, in 10.7% of the cases, the MB of the target variable had 5 additional variables in instance-specific models compared to the population-

Table 38: One-year survival given demographic and clinical characteristics. A 95% confidence interval is included for each sub-group of patients.

		Greater than 1 year	
Variable Name	**Variable Value**	**# Patients (Total)**	**% Patients (Confidence Interval)**
Age	22-62	54 (84)	64.29 (54.04, 74.54)
	63-72	41 (88)	46.59 (36.17, 57.01)
	73-88	45 (89)	50.56 (40.17, 60.95)
Sex	Female	80 (135)	59.26 (50.97, 67.55)
	Male	60 (126)	47.62 (38.9, 56.34)
Race	White	119 (224)	53.13 (46.58, 59.66)
	Black	16 (31)	51.61 ((34.02, 69.2)
	Other	5 (6)	83.33 (53.33, 113.33)
Tobacco History	Cigar/pipe smoker	0 (1)	0
	Cigarette smoker	42 (85)	49.41 (38.78, 60.04)
	Never used	22 (32)	68.75 (52.69, 84.81)
	Previous tobacco use	76 (142)	53.52 (45.32, 61.72)
	Snuff/chew/smokeless	0 (1)	0
Diagnosis	Adenocarcinoma	53 (89)	59.55 (49.35, 69.75)
	Squamous	3 (7)	42.86 (45.3, 62.06)
	Other	11 (29)	37.93 (6.20, 79.52)
	NA	73 (136)	53.68 (20.27, 55.59)

Table 39: Type, name, and description of the variables in the lung cancer dataset.

Variable Type	**Variable Name**	**Variable Description**
Demographics	Age	Age
	Sex	Sex
	Race	Race
	Tobacco history	Tobacco history
Clinical	Site	Location of tumor
	Surgical Procedure	Type of surgical resection or biopsy
	Diagnosis	Lung cancer type (Adenocarcinoma, Squamous, Other, NA)
	Mets at Dx-Brain, Mets at Dx-Bone, Mets at Dx-Distant Lymph Nodes, Mets at Dx-Lung, Mets at Dx-Liver, Mets at Dx-Other	Location of metastasis at diagnosis (Dx), if any
	Histo Behavior ICD-O-3	Histological classification
	cT, cN, cM, cStage Group	Clinical staging
	pT, pN, pM, pStage Group, Pathologic Stage Descriptor	Pathologic staging
Molecular	PD-L1 IHC, PD-L1 Comment	PD-L1 immunohistochemistry measures the amount of PD-L1 staining on tumor cells
	MET, KRAS, EGFR-summary, EGFR-Exon-18, EGFR-Exon-19, EGFR-Exon-20, EGFR-Exon-21, BRAF, PIK3CA, ALK Mutation	Status of gene mutations
	ALK IHC	ALK gene immunohistochemistry
	ALK Trans	ALK gene translocation
	ROS Trans	ROS gene translocation
	RET Trans	RET gene translocation
	cMET Ratio	Measurement of cMET gene amplification

wide model. Table 41b also shows the percentage of the 7 variables that occurred the most in the instance-specific MBs. According to literature, EGFR-Exon-19 is one of the most commonly found EGFR mutations in lung cancer patients while other subtypes of EGFR mutations (e.g., Exon-18 and Exon-20) have been found to be predictive of non-response to therapy in some patients as well [Ettinger et al., 2017]; these findings are also supported by the IGES method as shown in Table 41b. Table 41 supports that instance-specific structures exist for the lung cancer cases for the dataset we used.

Table 40: AUROC of the GES and IGES methods on the lung cancer dataset for *one-year survival*. Boldface indicates statistically significantly better results.

Method	**GES**	**IGES**									
PESS	-	$\kappa = 0.1$	$\kappa = 0.2$	$\kappa = 0.3$	$\kappa = 0.4$	$\kappa = 0.5$	$\kappa = 0.6$	$\kappa = 0.7$	$\kappa = 0.8$	$\kappa = 0.9$	$\kappa = 1.0$
0.1	0.68	0.67	0.70	0.70	0.70	0.70	0.71	0.71	0.70	0.71	0.72
1.0	0.68	0.68	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**	**0.74**	**0.75**	**0.75**	**0.81**
10.0	0.65	**0.75**	**0.75**	**0.77**	**0.76**	**0.75**	**0.72**	**0.73**	**0.73**	0.69	0.70

Table 41: Comparison of the target variable's Markov blanket (MB) in the instance-specific BNs vs. the population-wide BN in the lung cancer dataset with PESS = 1.0 and $\kappa = 1.0$.

(a) Structural differences of the MBs of the target variable in instance-specific BNs vs. the population-wide BN.

# Added	**# Deleted**	**# Reoriented**	**% Patients**
0	0	0	16.9
5	0	0	10.7
4	0	0	7.7
1	0	0	6.9
3	0	2	4.2
0	0	2	4.2
6	0	0	3.8
other	other	other	45.6

(b) Percentage of top-7 variables in the MBs of instance-specific BNs. The MB of the population-wide BN includes the first two variables denoted by *.

Variable Name	**% Occurrence in Patients**
EGFR-Exon-19*	98.1
Mets at Dx-Other*	92.8
Race	37.9
EGFR-Exon-18	35.6
EGFR-Exon-20	31.8
cM	26.8
cStage Group	24.5

4.6 Summary and Discussion

This chapter introduces a Bayesian instance-specific structure learning algorithm called IGES that outputs a Bayesian network structure that is specific to a given instance T (e.g, a patient) by guiding the search based on T's attributes. Although we applied a GES-style algorithm in the research reported here, the proposed method is quite general and can be adapted for use with other score-based search methods. We evaluated the performance of the IGES method on simulated and real-world biomedical datasets.

The results on simulated data indicate that IGES performs better in terms of adjacency and arrowhead precision (especially when a node exhibits CSI) for discovering the instance-specific CBN structure of each test instance T. However, the recall decreases for the small sample sizes due to more edges being deleted when applying IGES. As the sample size increases, both methods perform comparably similar in terms of recall. The structural Hamming distance is lower on average when using IGES (the lower the better) with moderate to large datasets. These results suggest that the CBN structures learned by IGES are more probable and better fit the relationships among variables for each instance T.

We also evaluated the performance of the IGES structure learning method on three real-world biomedical datasets. This is a challenging task because the true underlying causal relationships are not all known in many biomedical domains, including the datasets we used here; therefore, we used other criteria to evaluate performance. In particular, we compared the predictive performance of target variables using AUROC and the structural differences between the instance-specific and population-wide CBN models. The results provide support that the instance-specific CBN models are often different and have better predictive performance than the population-wide ones.

Overall, the proposed IGES method is a promising approach to discover a CBN structure that better models the relationships among variables of a given instance T, rather than a population-wide model, which supports the second hypothesis in Section 1.2, which states that the instance-specific CBN structure learning approach will perform structure learning better than a population-wide method, in terms of discrimination.

5.0 Instance-Specific CBN Structure Learning Assuming Latent Variables

In Chapter 3, I introduced a hybrid PAG learning approach, called Bayesian scoring of constraints (BSC). The BSC algorithm uses a Bayesian method to perform an independence test (Section 3.3) that can be incorporated into any search that requires independence testing (e.g., FCI), rather than using a frequentist significance testing (Section 3.4). Using BSC, we can compute the posterior probability of a PAG as the joint posterior probability of all the independence constraints that characterize that PAG [Jabbari et al., 2017b], which is the major advantage of the BSC method over constraint-based methods. However, the BSC method learns a population-wide PAG. As mentioned earlier, a population-wide model would at best recover the more common causal structure relationships in a population of instances, and consequently, would fail to capture the particular causal structure relationships in a given instance (e.g., a patient).

In Chapter 4, I introduced a fully Bayesian instance-specific structure learning method, called IGES, that searches the space of CBNs to build a model that is specific to an instance T by guiding the search from the features we know about T and from a training set of data on many other instances [Jabbari et al., 2018] (Section 4.4). The IGES method assumes that there are no latent confounders (i.e., causal sufficiency). However, relying on the causal sufficiency assumption could be a major drawback since this assumption is unrealistic in many practical applications.

In the current chapter, I introduce a novel hybrid approach that combines both BSC and IGES methods to construct an instance-specific PAG structure, which models latent confounders, to learn a specialized PAG for a given instance T by leveraging the features (i.e., the variable-value pairs) of T and a training set of data on many other instances. We hypothesize that such an instance-specific PAG learning approach will model the causal relationships for T better than does a population-wide one when accounting for the possibility of latent confounders. I evaluate this method using both simulated and real-world data.

In the remainder of this chapter, I provide an overview of a well-known hybrid population-wide PAG learning algorithm, called Greedy Fast Causal Inference [Ogarrio et al., 2016], in

Section 5.1. I introduce an instance-specific version of GFCI, called IGFCI [1], in Section 5.2. Finally, I report experimental results on both simulated and real-world biomedical datasets in Section 5.3.

5.1 Overview of the GFCI Algorithm

GFCI [Ogarrio et al., 2016] is a hybrid search algorithm that combines a score-based approach (i.e., GES [Chickering, 2002]) and a constraint-based approach (i.e., FCI [Spirtes et al., 1995]). It does so because GES is fast and effective at finding the variables that are directly dependent (i.e., have some type of edge between them) and FCI is effective at determining the specific edges types in forming a PAG. The GES and FCI methods are discussed in Sections 4.2 and 3.2, respectively. The GFCI algorithm learns a PAG structure in two steps:

1. The first step of GFCI involves an adjacency search. To find the adjacency graph, GFCI first applies the GES algorithm using a dataset D, which discovers a pattern $\mathcal{G}$ (line 1 in Algorithm 12). It then removes the edge orientations of $\mathcal{G}$ to obtain the skeleton graph called $\mathcal{P}$ (line 2 in Algorithm 12). This adjacency graph may contain extraneous edges if the model includes latent confounders [Ogarrio et al., 2016]. To eliminate such extraneous edges, GFCI resumes the adjacency search in a way similar to the first stage of the FCI algorithm. As denoted in line 3 of Algorithm 12, GFCI uses $\mathcal{P}$ as the initial graph in the initial skeleton search of the FCI algorithm, which is described in Algorithm 13. It also applies the v-structure orientation (Algorithm 14) and the final skeleton search of FCI (Algorithm 4) as shown in lines 4 and 5 of Algorithm 12.
2. In the second step, GFCI applies v-structure orientation and additional orientation rules from [Zhang, 2008] to obtain the PAG structure (lines 6 and 7 of Algorithm 12).

The overall pseudo-code for GFCI is provided in Algorithm 12. The GFCI algorithm outputs the correct PAG with probability 1.0 in the large sample limit, given i.i.d. sampling and the Markov and faithfulness assumptions [Ogarrio et al., 2016].

[1] We introduced a variation of this algorithm in [Jabbari and Cooper, 2020].

Algorithm 12 GFCI(D)

Input: a dataset D with n observed variables

Output: a population-wide PAG $\mathcal{P}$

1: $\mathcal{G} \leftarrow$ GES (D) ▷ Algorithm 9
2: $\mathcal{P} \leftarrow$ Remove edge orientation of $\mathcal{G}$
3: $\mathcal{P}, \boldsymbol{D\text{-}Sep} \leftarrow$ Initial Skeleton $(D, n, \mathcal{P})$ ▷ Algorithm 13
4: $\mathcal{P} \leftarrow$ V-structure Orientation $(\mathcal{G}, \mathcal{P}, \boldsymbol{D\text{-}Sep})$ ▷ Algorithm 14
5: $\mathcal{P}, \boldsymbol{D\text{-}Sep} \leftarrow$ Final Skeleton$(D, n, \mathcal{P}, \boldsymbol{D\text{-}Sep})$ ▷ Algorithm 4
6: $\mathcal{P} \leftarrow$ V-structure Orientation $(\mathcal{P}, \mathcal{G}, \boldsymbol{D\text{-}Sep})$ ▷ Algorithm 14
7: Apply orientation rules $\mathcal{R}1$-$\mathcal{R}10$ in [Zhang, 2008] to further orient the edges in $\mathcal{P}$
8: return PAG $\mathcal{P}$

Algorithm 13 Initial Skeleton(D, d, $\mathcal{P}$)

Input: a dataset D, the maximum conditioning set size d, an initial adjacency graph $\mathcal{P}$

Output: a graph $\mathcal{P}$, d-separation sets $\boldsymbol{D\text{-}Sep}$

$m = 0$
while $m \leq d$ **do**
  for all $(X_i, X_j) \in \mathcal{P}$ **do**
    if $X_j \in \boldsymbol{Adj}(X_i)$ and $|\boldsymbol{Adj}(X_i) \backslash X_j| \geq m$ **then**
      repeat
        Choose a subset $\boldsymbol{Z} \subseteq \boldsymbol{Adj}(X_i) \backslash X_j$ where $|\boldsymbol{Z}| = m$
        if $X_i \perp\!\!\!\perp X_j | \boldsymbol{Z}$ **then**
          Remove $X_i \circ\!\!-\!\!\circ X_j$ from $\mathcal{P}$
          Record $\boldsymbol{D\text{-}Sep}(X_i, X_j) = \boldsymbol{D\text{-}Sep}(X_j, X_i) = \boldsymbol{Z}$
        end if
      until $X_j \notin \boldsymbol{Adj}(X_i)$ or all $\boldsymbol{Z} \subseteq \boldsymbol{Adj}(X_i) \backslash X_j$ with $|\boldsymbol{Z}| = m$ have been tested
    end if
  end for
  $m = m + 1$
end while
return $\mathcal{P}$ and $\boldsymbol{D\text{-}Sep}$

Algorithm 14 V-structure Orientation($\mathcal{G}$, $\mathcal{P}$, ***D-Sep***)

Input: a graph $\mathcal{G}$, a graph $\mathcal{P}$ and d-separation sets ***D-Sep***

Output: a graph $\mathcal{P}$

1: Form a list $\mathcal{T}$ of all unshielded triple of variables X_i—X_k—X_j in $\mathcal{P}$
2: **for all** X_i—X_k—$X_j \in \mathcal{T}$ **do**
3: **if** (X_k is an unshielded collider in $\mathcal{G}$) or (X_k is shielded in $\mathcal{G}$ and $X_k \notin$ ***D-Sep***(X_i, X_j)) **then**
4: Orient it as $X_i \circ\!\!\rightarrow X_k \leftarrow\!\!\circ X_j$
5: **end if**
6: **end for**
7: return $\mathcal{P}$

5.2 Instance-Specific GFCI (IGFCI)

In this section, I describe a novel instance-specific PAG learning algorithm that applies the idea of instance-specific modeling to GFCI. Instance-specific GFCI (IGFCI) takes as input a set D of observational training instances and a test instance T, and it returns as output an instance-specific PAG $\mathcal{P}_{IS}$. IGFCI algorithm operates in two steps:

- In the first step (line 1 in Algorithm 15), it applies the population-wide GFCI algorithm (described in Section 5.1) using dataset D. GFCI initially learns a population-wide CBN by performing GES search using the BDeu score (line 1 in Algorithm 12), which we denote as $\mathcal{G}_{PW}$. Then, GFCI uses $\mathcal{G}_{PW}$ as the initial adjacency graph and performs additional conditional independence tests to further prune the adjacency structure (lines 2-5 in Algorithm 12) using the BSC test (Section 3.3). Finally, GFCI applies the orientation rules (lines 6 and 7 in Algorithm 12) to obtain a population-wide PAG, which we denote as $\mathcal{P}_{PW}$.
- In the second step (line 2 in Algorithm 15), it applies an instance-specific version of GFCI. For the score-based part, it applies the IGFCI algorithm with IS-Score (described in Section 4.4) to learn an instance-specific CBN given D, T, and the population-wide

CBN from the first step that produced $\mathcal{G}_{PW}$. For the constraint-based part, it applies a novel instance-specific BSC test, called IS-BSC, to find an instance-specific PAG $\mathcal{P}_{IS}$ given D, T, and $\mathcal{G}_{PW}$; we use the name GFCI2 to denote this application of GFCI. Algorithm 15 shows the high-level procedure of IGFCI algorithm. In the following section, I explain the IS-BSC test.

Algorithm 15 IGFCI (D, T)

Input: a dataset D and a test case T

Output: a population-wide PAG $\mathcal{P}_{PW}$ and an instance-specific PAG $\mathcal{P}_{IS}$.

1: $\mathcal{G}_{PW}$, $\mathcal{P}_{PW} \leftarrow$ GFCI(D)

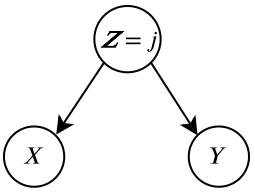
2: $\mathcal{G}_{IS}$, $\mathcal{P}_{IS} \leftarrow$ GFCI2(D, T, $\mathcal{G}_{PW}$)

3: return $\mathcal{P}_{PW}$ and $\mathcal{P}_{IS}$

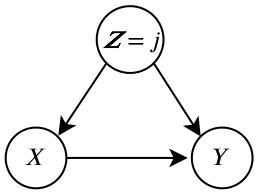
5.2.1 Instance-specific Bayesian scoring of constraints (IS-BSC)

This section describes how to derive the posterior probability of an instance-specific independence constraint from data for a given test instance. Let D be an i.i.d dataset and T be a single test instance that are generated from a distribution that is faithful to a ground-truth CBN structure $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V}$ is a set of domain variables with $\boldsymbol{O} \subseteq \boldsymbol{V}$ observed variables and $\boldsymbol{E}$ is a set of edges that encodes independence relationships in $\boldsymbol{V}$. Let $R = (X \perp\!\!\!\perp Y | \boldsymbol{Z} = j)$ be an arbitrary instance-specific conditional independence constraint that is hypothesized to hold in the test instance T, where $X, Y \in \boldsymbol{O}$ and $\boldsymbol{Z} \backslash \{X, Y\} \subseteq \boldsymbol{O}$. In such a constraint, the conditioning set $\boldsymbol{Z}$ takes specific values j that correspond to the values of $\boldsymbol{Z}$ in T. The goal is to determine whether R holds in the context of $\boldsymbol{Z} = j$ using a Bayesian scoring method. We consider the BN structures shown in Figure 14a and 14b to model the independence and dependence relationships between X and Y given $\boldsymbol{Z} = j$, respectively.

The basic idea behind IS-BSC is to find those cases in D in which $\boldsymbol{Z} = j$ and use them to score the instance-specific constraint R. In essence, those instances in D form a cluster that are similar to instance T in the context of $\boldsymbol{Z} = j$; we use that cluster to determine



(a) The BN structure that corresponds to independence (i.e., $R = (X \perp\!\!\!\perp Y | \boldsymbol{Z} = j) = true$).



(b) The BN structure that corresponds to dependence (i.e., $r = (X \perp\!\!\!\perp Y | \boldsymbol{Z} = j) = false$).

Figure 14: Independence and dependence structures that are used to score an instance-specific constraint.

whether the independence constraint holds between (X, Y). More specifically, let $D_{Z=j}$ denote the instances in D in which $\boldsymbol{Z} = j$ and $D_{Z \neq j}$ denote the remaining instances in D (line 1 in Algorithm 16). We use $D_{Z=j}$ to determine whether the independence constraint $R = (X \perp\!\!\!\perp Y | \boldsymbol{Z} = j)$ as follows (line 3 in Algorithm 16):

$$\begin{aligned} P(R = true | D_{Z=j}) &= \frac{P(R = true) \cdot P(D_{Z=j} | R = true)}{P(D_{Z=j})} \\ &= \frac{P(R = true) \cdot P(D_{Z=j} | R = true)}{\sum\limits_{R=\{true, false\}} P(R) \cdot P(D_{Z=j} | R)}, \end{aligned} \tag{5.1}$$

where $P(D_{Z=j} | R = true)$ and $P(D_{Z=j} | R = false)$ are calculated using BNs in Figures 14a and 14b, respectively. We use the remaining cases of data in which $\boldsymbol{Z} \neq j$ (i.e., $D_{\boldsymbol{Z} \neq j}$) to estimate the prior probability $P(R = true)$ as follows:

$$P(R = true) = \sqrt[q-1]{P(X \perp\!\!\!\perp Y | \boldsymbol{Z} \neq j) = true | D_{Z \neq j})} \tag{5.2}$$

where $X \perp\!\!\!\perp Y | \boldsymbol{Z} \neq j$ denotes the same conditional independence constraint but for the $q-1$ remaining values of $\boldsymbol{Z}$. We use the BSC test described in Section 3.3.1 (Equation (3.6)) to compute this quantity.

Algorithm 16 IS-BSC(D, T, $R = (X \perp\!\!\!\perp Y | \boldsymbol{Z} = j)$)

Input: a dataset D, a test case T, an instance-specific constraint $R = (X \perp\!\!\!\perp Y | \boldsymbol{Z} = j)$

Output: the posterior probability of independence constraint R

1: Derive $D_{Z=j}$ and $D_{Z \neq j}$ from D based on the values j of $\boldsymbol{Z}$ in T

2: Compute $P(R = true)$ using Equation (5.2)

3: Compute $P(R = true | D_{Z=j})$ using $P(R = true)$ in Equation(5.1)

4: return $P(R = true | D_{Z=j})$

5.3 Experimental Results

This section describes the experimental methods and results that we used to investigate the performance of the instance-specific GFCI (IGFCI) versus GFCI, which is a state-of-the-art, non-instance-specific PAG-learning algorithm. To do so, we used both simulated and real data, which are described below in Sections 5.3.1 and 5.3.2, respectively.

5.3.1 Simulated data

To investigate the performance of IGFCI versus GFCI, we conducted simulation studies to generate data as follows.

1. We created random BNs with $|V| = \{10, 20, 50\}$ discrete random variables where each variable has 2, 3, or 4 categories, which is chosen randomly. The expected number of edges are $|\boldsymbol{E}| = \{2|\boldsymbol{V}|, 4|\boldsymbol{V}|, 6|\boldsymbol{V}|\}$. To generate a BN structure $\mathcal{G} = (\boldsymbol{V}, \boldsymbol{E})$, we first created an arbitrary ordering of variables[2]. Then, we uniformly randomly added edges to $\mathcal{G}$ in a forward direction until obtaining the specified number of edges. The DAGs generated in this way have a power-law-type distribution over the number of parents, with some variables having many more than the average number of parents.

[2]This ordering is only used to generate the BNs; we do not use it when applying GFCI or IGFCI.

2. Given a BN structure $\mathcal{G}$, we then parametrized the distribution of each random variable $X \in \boldsymbol{V}$ given its parents $\boldsymbol{Pa}(X)$ according to $\mathcal{G}$ under the constraints that follow from the axioms of probability theory. We also included context-specific independencies (CSIs) in the CPTs so that each variable that has more than one parent includes at least one CSI. CSI parents generated this way are a proper subset of the population-wide parents in the data-generating model. In the BNs with the edge density of $2|\boldsymbol{V}|$, $4|\boldsymbol{V}|$, and $6|\boldsymbol{V}|$, about 28%, 38%, and 48% of the variables (on average) exhibit CSI in each simulated test case T, respectively.
3. We randomly set $L = 20\%$ of variables to be latent (i.e., hidden). These variables were chosen at random from a list of all variables that are common causes of two or more of the measured variables. If there are fewer common causes than $L = 20\%$ of variables, we randomly selected from a list of the variables that are common effects of two or more of the measured variables.
4. We used each randomly generated BN $\mathcal{G}$ and its parameters to generate a training dataset D with $N = \{200, 1000, 5000\}$ training samples.
5. We also generated $M = 500$ test instances from each randomly generated BN $\mathcal{G}$ and its parameters; we refer to each instance as T.
6. We used the training dataset D generated in step 4 to learn a population-wide PAG structure using the GFCI algorithm (Section 5.1), which uses GES and FCI methods in its two steps. For GES, we used the BDeu score [Heckerman, 1998] with a prior equivalence sample size (PESS) of 1.0 to learn a population-wide pattern $\mathcal{G}_{PW}$. For the independence testing used in FCI, we applied BSC (Section 3.3) with a 0.5 decision threshold. This means that if $P(R = (X \perp\!\!\!\perp Y|\boldsymbol{Z})|D) \geq 0.5$, then BSC returns *true* for R, otherwise, it returns *false*. The final output of GFCI is a PAG model, which we refer to as $\mathcal{P}_{PW}$.
7. For each test instance T generated in step 5, we used T and the training dataset D to learn an instance-specific PAG structure using the IGFCI algorithm described in Section 5.2. Similar to GFCI, IGFCI uses a score-based method (i.e., IGES with IS-Score) and a constraint-based method (i.e., FCI with IS-BSC independence test) in its two steps. For IGES, we used $\mathcal{G}_{PW}$ as the population-wide model. Also, we set PESS = 1.0 and the

structure prior penalty $\kappa = \{0.001, 0.1, 0.5, 0.9\}$, where κ $(0 < \kappa \leq 1)$ is a penalty factor that is used when computing the prior probabilities of the instance-specific BN structures; it penalizes the structural difference between the population-wide and instance-specific BNs (see Section 4.4 for more details). For the FCI part, we used IS-BSC independence tests (Section 5.2.1) with the decision threshold of 0.5. The final output of IGFCI is a PAG model for the given test instance T; we refer to this PAG as $\mathcal{P}_{IS}$.

8. Finally, we computed evaluation measures (described below) to compare the structure recovery performance of GFCI and IGFCI versus the ground-truth PAG structure for each test instance T (steps 1-3); which we denote as $\mathcal{P}_{truth}$. To obtain $\mathcal{P}_{truth}$, we first derived the ground-truth CBN $\mathcal{G}_{truth}$ for each test instance T considering the existing CSIs associated with T. Then, we ran FCI using an independence oracle on the observed variables in $\mathcal{G}_{truth}$ to derive the instance-specific PAG that is consistent with t$\mathcal{G}_{truth}$. We compared $\mathcal{P}_{PW}$ and $\mathcal{P}_{IS}$ versus $\mathcal{P}_{truth}$ for each test case and reported the average of measures over the $M = 500$ test cases.

For each simulation setting mentioned above, steps 1 through 8 were repeated for 10 randomly generated BNs and the performance results were averaged using the evaluation measures described in the following section.

5.3.1.1 PAG structure discovery performance measures

In this section, I describe the evaluation measures that are used to calculate the structural similarity of the discovered PAG $\mathcal{P}_{output}$, which is $\mathcal{P}_{PW}$ when using GFCI and $\mathcal{P}_{IS}$ when using IGFCI, versus the ground-truth PAG $\mathcal{P}_{truth}$.

We used structural Hamming distance (SHD) that counts the edge modifications that include added, deleted, and reoriented edges, by comparing each possible edge in $\mathcal{P}_{output}$ and $\mathcal{P}_{truth}$. We define three versions of SHD for PAGs as follows:

- **Strict SHD (S-SHD)**: This version counts any edge modifications, which are added, deleted, and reoriented edges. The S-SHD would be 0 if for a given pair of measured variables the edge in $\mathcal{P}_{output}$ is exactly the same as the edge in PAG $\mathcal{P}_{truth}$; otherwise, it is 1. Any extra or missing edge would also count as 1 in terms of S-SHD. Table 42a

shows how to compute S-SHD for PAGs.

- **Lenient SHD (L-SHD)**: This version allows general edges that include circle endpoints to be compatible with their specializations. For example, the L-SHD between $A \circ\!\!\rightarrow B$ and $A \rightarrow B$ is 0 because these edges are compatible. However, the L-SHD between $A \rightarrow B$ and $B \rightarrow A$ is 1 because they are not compatible. L-SHD is symmetric regarding the output and the truth edges, as shown in Table 42b.
- **Adjacency SHD (A-SHD)**: In this version, we compute SHD on the skeleton-level by comparing the adjacencies of two graphs, which disregards the edge orientations and only counts the edge modifications of the adjacency graph that includes added and deleted edges. For example, if one graph includes $A \circ\!\!-\!\!\circ B$ but there is no edge between A and B in the other one, then A-SHD would be 1.

Table 42: Two types of SHD for PAGs. The rows and columns correspond to the edge types output by the algorithm and the data-generating edge types, respectively.

(a) Strict SHD (S-SHD) for PAGs.

Output Edge/ Truth Edge	$A \rightarrow B$	$A \leftrightarrow B$	$A \circ\!\!\rightarrow B$	$A \circ\!\!-\!\!\circ B$	$A \quad B$
$A \rightarrow B (B \rightarrow A)$	0 (1)	1	1	1	1
$A \leftrightarrow B$	1	0	1	1	1
$A \circ\!\!\rightarrow B (B \circ\!\!\rightarrow A)$	1	1	0 (1)	1	1
$A \circ\!\!-\!\!\circ B$	1	1	1	0	1
$A \quad B$	1	1	1	1	0

(b) Lenient SHD (L-SHD) for PAGs.

Output Edge/ Truth Edge	$A \rightarrow B$	$A \leftrightarrow B$	$A \circ\!\!\rightarrow B$	$A \circ\!\!-\!\!\circ B$	$A \quad B$
$A \rightarrow B (B \rightarrow A)$	0 (1)	1	0	0	1
$A \leftrightarrow B$	1	0	0	0	1
$A \circ\!\!\rightarrow B (B \circ\!\!\rightarrow A)$	0 (1)	0	0	0	1
$A \circ\!\!-\!\!\circ B$	0	0	0	0	1
$A \quad B$	1	1	1	1	0

Other performance criteria we used to evaluate discrimination are precision (P) and recall (R) for adjacencies and arrowheads as follows:

- **Adjacency precision (AP)**: we compute the ratio of correctly predicted edges in $\mathcal{P}_{output}$ to all predicted edges in $\mathcal{P}_{output}$ (without considering orientations of edges) as follows:

$$\text{AP} = \frac{\#\text{correctly predicted adjacencies}}{\#\text{predicted adjacencies}} \tag{5.3}$$

- **Adjacency recall (AR)**: we compute the ratio of correctly predicted edges in $\mathcal{P}_{output}$ to all true edges in $\mathcal{P}_{truth}$ (without considering the edges' orientations) as follows:

$$\text{AR} = \frac{\#\text{correctly predicted adjacencies}}{\#\text{true adjacencies}} \tag{5.4}$$

- **Arrowhead precision (AHP)**: considering the pairs of measured variables that have an edge between them in the predicted graph $\mathcal{P}_{output}$, we compute the ratio of correctly predicted arrowheads in $\mathcal{P}_{output}$ to all predicted arrowheads in $\mathcal{P}_{output}$ as follows:

$$\text{AHP} = \frac{\#\text{correctly predicted arrowheads}}{\#\text{predicted arrowheads}} \tag{5.5}$$

- **Arrowhead recall (AHR)**: considering the pairs of measured variables that have an edge between them in the ground-truth graph $\mathcal{P}_{truth}$, we compute the ratio of correctly predicted arrowheads in $\mathcal{P}_{output}$ to all true arrowheads in $\mathcal{P}_{truth}$ as follows:

$$\text{AHR} = \frac{\#\text{correctly predicted arrowheads}}{\#\text{true arrowheads}} \tag{5.6}$$

Note that an arrowhead in a PAG indicates causation due to either a measured or a latent variable (see Section 2.1.2 and the example given in Figure 5 for more details).

In this chapter, since we are evaluating methods using data that have been generated by instance-specific models, the ground-truth PAGs are derived based on the given instance. Therefore, similar to Chapter 4, we derived three subtypes for precision and recall evaluation measurements: (1) the nodes that include context-specific independence (CSI), for which we derive precision (P_{IS}) and recall statistics(R_{IS}), (2) the nodes that do not include CSI, for which we derive separate precision (P_{other}) and recall (R_{other}) statistics, and (3) we also combine these two types of nodes to derive overall precision (P) and recall (R) statistics (see the example given in Section 4.5.1.1).

5.3.1.2 Simulation results Tables 43, 44, and 45 show the average adjacency P and R results of the IGFCI ($\kappa = 0.1$)[3] and GFCI algorithms over 10 randomly generated CBNs described above, using $N = \{200, 1000, 5000\}$ training instances, respectively. For $N = 200$, IGFCI ($\kappa = 0.1$) and GFCI perform similar in terms of adjacency P, but adjacency R is better when using GFCI (Table 43). As the sample size increases to $N = 1000$ (Table 44), both methods perform better in terms of adjacency R, but GFCI performs better in terms of this measure. Additionally, IGFCI outperforms GFCI in terms of adjacency P, for *IS* subtype and overall. When using $N = 5000$ training instances, IGFCI almost always performs significantly better in terms of adjacency P for *IS* subtype and overall, while GFCI always performs significantly better in terms of adjacency R based on Wilcoxon signed rank test at 5% significance level (Table 45).

Tables 46, 47, and 48 show the average arrowhead P and R results of the IGFCI ($\kappa = 0.1$) and GFCI algorithms over 10 randomly generated CBNs described above, using $N = \{200, 1000, 5000\}$ training instances, respectively. As shown in these tables, when using $N = 200$ training instances, both IGFCI and GFCI methods perform similarly in terms of arrowhead P and R (Table 46). As the sample size increases to $N = 1000$, both methods perform better in terms of arrowhead P and R, while arrowhead P is almost always better for IGFCI (Table 47). Increasing the number of training instances to $N = 5000$ results in better arrowhead P and R for both methods. In this case, the arrowhead P performance of IGFCI is almost always significantly better than GFCI, while GFCI has significantly better arrowhead R based on Wilcoxon signed rank test at 5% significance level (Table 48).

[3]Results using additional values of $\kappa = \{0.001, 0.1, 0.5, 0.9\}$ are reported in Appendix B. Also, omitted rows in the tables represent the settings that failed to return a result in under 72 hours.

Table 43: Adjacency precision (P) and recall (R) results for $N = 200$ training cases. A penalty factor $\kappa = 0.1$ is used to penalize the structural difference between the population-wide and instance-specific CBNs in the first stage of IGFCI. The numbers after '±' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGFCI	0.90 ± 0.20	1.00 ± 0.01	0.95 ± 0.08	0.30 ± 0.16	0.30 ± 0.12	0.30 ± 0.12
		GFCI	0.88 ± 0.20	1.00 ± 0.01	0.95 ± 0.07	**0.40 ± 0.14**	**0.43 ± 0.14**	**0.42 ± 0.12**
	40	IGFCI	0.84 ± 0.25	1.00 ± 0.00	0.90 ± 0.16	0.22 ± 0.12	0.19 ± 0.09	0.19 ± 0.09
		GFCI	0.90 ± 0.13	1.00 ± 0.00	0.93 ± 0.11	**0.30 ± 0.12**	**0.25 ± 0.11**	**0.27 ± 0.10**
	60	IGFCI	0.95 ± 0.07	1.00 ± 0.00	0.97 ± 0.05	0.25 ± 0.12	0.16 ± 0.08	0.21 ± 0.06
		GFCI	0.90 ± 0.10	1.00 ± 0.00	0.94 ± 0.06	**0.34 ± 0.14**	**0.23 ± 0.12**	**0.30 ± 0.06**
20	40	IGFCI	0.86 ± 0.11	1.00 ± 0.01	0.94 ± 0.04	0.27 ± 0.10	0.24 ± 0.08	0.24 ± 0.07
		GFCI	0.80 ± 0.12	1.00 ± 0.00	0.92 ± 0.04	**0.33 ± 0.13**	**0.30 ± 0.11**	**0.30 ± 0.09**
	80	IGFCI	0.92 ± 0.09	0.99 ± 0.02	0.94 ± 0.07	0.18 ± 0.07	0.12 ± 0.03	0.15 ± 0.05
		GFCI	0.90 ± 0.07	0.99 ± 0.02	0.93 ± 0.04	**0.23 ± 0.06**	**0.15 ± 0.04**	**0.19 ± 0.04**
	120	IGFCI	0.90 ± 0.06	0.97 ± 0.08	0.93 ± 0.06	0.16 ± 0.06	0.09 ± 0.04	0.12 ± 0.05
		GFCI	0.89 ± 0.07	0.98 ± 0.07	0.92 ± 0.06	**0.21 ± 0.07**	**0.13 ± 0.04**	**0.17 ± 0.05**
50	100	IGFCI	0.87 ± 0.05	0.97 ± 0.05	0.93 ± 0.03	0.24 ± 0.08	0.22 ± 0.05	0.22 ± 0.05
		GFCI	0.86 ± 0.06	0.99 ± 0.03	0.94 ± 0.03	**0.29 ± 0.09**	**0.26 ± 0.06**	**0.27 ± 0.07**
Summary statistics		GFCI	0.89 ± 0.03	0.99 ± 0.01	0.93 ± 0.02	0.23 ± 0.04	0.19 ± 0.07	0.21 ± 0.05
		GFCI	0.88 ± 0.03	0.99 ± 0.01	0.93 ± 0.01	0.30 ± 0.06	0.25 ± 0.09	0.27 ± 0.08

Table 44: Adjacency precision (P) and recall (R) results for $N = 1000$ training cases. A penalty factor $\kappa = 0.1$ is used to penalize the structural difference between the population-wide and instance-specific CBNs in the first stage of IGFCI. The numbers after '±' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGFCI	**0.91 ± 0.11**	0.99 ± 0.02	**0.96 ± 0.05**	0.47 ± 0.15	0.43 ± 0.14	0.44 ± 0.12
		GFCI	0.82 ± 0.14	0.97 ± 0.07	0.91 ± 0.09	**0.59 ± 0.21**	**0.54 ± 0.18**	**0.55 ± 0.15**
	40	IGFCI	**0.95 ± 0.05**	1.00 ± 0.00	0.97 ± 0.03	0.38 ± 0.07	0.26 ± 0.07	0.31 ± 0.06
		GFCI	0.89 ± 0.07	1.00 ± 0.00	0.93 ± 0.05	**0.46 ± 0.09**	**0.34 ± 0.10**	**0.39 ± 0.08**
	60	IGFCI	**0.90 ± 0.04**	1.00 ± 0.00	**0.93 ± 0.04**	0.35 ± 0.14	0.26 ± 0.10	0.32 ± 0.07
		GFCI	0.84 ± 0.10	1.00 ± 0.00	0.90 ± 0.07	**0.44 ± 0.13**	**0.35 ± 0.12**	**0.42 ± 0.06**
20	40	IGFCI	**0.85 ± 0.07**	0.96 ± 0.03	**0.91 ± 0.04**	0.39 ± 0.11	0.38 ± 0.09	0.37 ± 0.07
		GFCI	0.71 ± 0.06	**0.98 ± 0.03**	0.85 ± 0.02	**0.46 ± 0.10**	**0.42 ± 0.10**	**0.42 ± 0.08**
	80	IGFCI	**0.89 ± 0.05**	0.97 ± 0.04	0.92 ± 0.04	0.29 ± 0.10	0.20 ± 0.05	0.24 ± 0.05
		GFCI	0.82 ± 0.07	0.99 ± 0.01	0.89 ± 0.06	**0.35 ± 0.08**	**0.24 ± 0.06**	**0.29 ± 0.04**
	120	IGFCI	**0.92 ± 0.04**	0.97 ± 0.07	**0.93 ± 0.05**	0.25 ± 0.07	0.16 ± 0.04	0.20 ± 0.05
		GFCI	0.85 ± 0.07	0.97 ± 0.07	0.89 ± 0.07	**0.31 ± 0.07**	**0.20 ± 0.04**	**0.25 ± 0.05**
50	100	IGFCI	**0.84 ± 0.06**	0.98 ± 0.02	**0.92 ± 0.03**	0.40 ± 0.09	0.38 ± 0.07	0.38 ± 0.08
		GFCI	0.73 ± 0.03	0.99 ± 0.02	0.87 ± 0.03	**0.45 ± 0.11**	**0.42 ± 0.07**	**0.42 ± 0.08**
Summary statistics		IGFCI	0.89 ± 0.04	0.98 ± 0.01	0.94 ± 0.02	0.36 ± 0.07	0.29 ± 0.09	0.32 ± 0.08
		GFCI	0.81 ± 0.06	0.99 ± 0.01	0.89 ± 0.02	0.44 ± 0.08	0.36 ± 0.11	0.39 ± 0.09

Table 45: Adjacency precision (P) and recall (R) results for $N = 5000$ training cases. A penalty factor $\kappa = 0.1$ is used to penalize the structural difference between the population-wide and instance-specific CBNs in the first stage of IGFCI. The numbers after '±' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGFCI	**0.89 ± 0.12**	0.98 ± 0.04	**0.94 ± 0.06**	0.59 ± 0.14	0.64 ± 0.15	0.60 ± 0.11
		GFCI	0.75 ± 0.14	1.00 ± 0.00	0.86 ± 0.09	**0.73 ± 0.14**	**0.75 ± 0.13**	**0.73 ± 0.11**
	40	IGFCI	**0.91 ± 0.07**	1.00 ± 0.00	**0.95 ± 0.04**	0.42 ± 0.09	0.31 ± 0.06	0.36 ± 0.05
		GFCI	0.77 ± 0.08	1.00 ± 0.00	0.86 ± 0.05	**0.62 ± 0.09**	**0.53 ± 0.09**	**0.57 ± 0.05**
	60	IGFCI	**0.92 ± 0.07**	1.00 ± 0.00	**0.94 ± 0.05**	0.42 ± 0.15	0.36 ± 0.10	0.40 ± 0.10
		GFCI	0.78 ± 0.09	1.00 ± 0.00	0.85 ± 0.05	**0.65 ± 0.13**	**0.51 ± 0.13**	**0.60 ± 0.09**
20	40	IGFCI	**0.86 ± 0.10**	0.98 ± 0.02	**0.93 ± 0.05**	0.48 ± 0.11	0.51 ± 0.14	0.48 ± 0.10
		GFCI	0.66 ± 0.07	0.99 ± 0.02	0.83 ± 0.04	**0.53 ± 0.11**	**0.55 ± 0.12**	**0.53 ± 0.09**
	80	IGFCI	**0.89 ± 0.04**	0.97 ± 0.04	**0.92 ± 0.04**	0.33 ± 0.09	0.25 ± 0.06	0.28 ± 0.06
		GFCI	0.76 ± 0.05	0.98 ± 0.02	0.85 ± 0.04	**0.42 ± 0.08**	**0.30 ± 0.08**	**0.36 ± 0.06**
	120	IGFCI	**0.91 ± 0.05**	0.98 ± 0.03	**0.93 ± 0.03**	0.30 ± 0.08	0.18 ± 0.05	0.24 ± 0.05
		GFCI	0.77 ± 0.04	1.00 ± 0.00	0.85 ± 0.03	**0.37 ± 0.07**	**0.26 ± 0.05**	**0.31 ± 0.05**
50	100	IGFCI	**0.82 ± 0.06**	0.97 ± 0.02	**0.90 ± 0.04**	0.50 ± 0.11	0.49 ± 0.08	0.49 ± 0.09
		GFCI	0.66 ± 0.05	**0.99 ± 0.01**	0.84 ± 0.04	**0.54 ± 0.10**	**0.51 ± 0.08**	**0.52 ± 0.09**
Summary statistics		IGFCI	0.89 ± 0.03	0.98 ± 0.01	0.93 ± 0.01	0.44 ± 0.09	0.39 ± 0.15	0.41 ± 0.12
		GFCI	0.74 ± 0.05	0.99 ± 0.01	0.85 ± 0.01	0.55 ± 0.12	0.49 ± 0.15	0.52 ± 0.13

Table 46: Arrowhead precision (P) and recall (R) results for $N = 200$ training cases. A penalty factor $\kappa = 0.1$ is used to penalize the structural difference between the population-wide and instance-specific CBNs in the first stage of IGFCI. The numbers after '±' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGFCI	**0.38 ± 0.40**	0.22 ± 0.29	**0.36 ± 0.39**	0.02 ± 0.05	0.02 ± 0.04	0.02 ± 0.03
		GFCI	0.23 ± 0.19	**0.31 ± 0.39**	0.28 ± 0.32	0.02 ± 0.04	0.04 ± 0.10	0.03 ± 0.06
	40	IGFCI	**0.41 ± 0.36**	0.37 ± 0.28	0.36 ± 0.26	0.07 ± 0.09	0.02 ± 0.02	0.03 ± 0.03
		GFCI	0.34 ± 0.12	**0.50 ± 0.35**	**0.46 ± 0.30**	0.05 ± 0.08	0.02 ± 0.05	0.03 ± 0.05
	60	IGFCI	0.03 ± 0.05	0.15 ± 0.21	0.06 ± 0.09	0.04 ± 0.07	0.01 ± 0.03	0.02 ± 0.03
		GFCI	**0.04 ± 0.06**	**0.36 ± 0.37**	**0.13 ± 0.14**	0.03 ± 0.07	0.02 ± 0.04	0.02 ± 0.04
20	40	IGFCI	**0.55 ± 0.32**	0.73 ± 0.39	**0.65 ± 0.36**	0.11 ± 0.13	0.06 ± 0.07	0.06 ± 0.06
		GFCI	0.49 ± 0.27	**0.86 ± 0.35**	0.62 ± 0.29	0.12 ± 0.13	0.06 ± 0.08	0.07 ± 0.07
	80	IGFCI	**0.48 ± 0.36**	0.70 ± 0.29	0.61 ± 0.28	0.09 ± 0.10	0.03 ± 0.03	0.04 ± 0.03
		GFCI	0.46 ± 0.27	**0.74 ± 0.27**	**0.63 ± 0.21**	0.07 ± 0.12	0.03 ± 0.03	0.03 ± 0.03
	120	IGFCI	**0.62 ± 0.31**	0.56 ± 0.32	**0.57 ± 0.30**	0.07 ± 0.06	0.01 ± 0.01	0.02 ± 0.02
		GFCI	0.44 ± 0.31	**0.58 ± 0.37**	0.50 ± 0.29	0.07 ± 0.09	0.01 ± 0.01	0.02 ± 0.03
50	100	IGFCI	0.33 ± 0.29	0.81 ± 0.16	0.76 ± 0.19	0.02 ± 0.03	0.06 ± 0.02	0.05 ± 0.02
		GFCI	0.14 ± 0.18	**0.91 ± 0.20**	0.66 ± 0.18	0.03 ± 0.05	0.05 ± 0.03	0.05 ± 0.02
Summary statistics		IGFCI	0.40 ± 0.18	0.50 ± 0.24	0.48 ± 0.22	0.06 ± 0.03	0.03 ± 0.02	0.03 ± 0.02
		GFCI	0.30 ± 0.16	0.61 ± 0.22	0.47 ± 0.18	0.06 ± 0.03	0.03 ± 0.02	0.03 ± 0.02

Table 47: Arrowhead precision (P) and recall (R) results for $N = 1000$ training cases. A penalty factor $\kappa = 0.1$ is used to penalize the structural difference between the population-wide and instance-specific CBNs in the first stage of IGFCI. The numbers after '±' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGFCI	**0.29 ± 0.24**	0.50 ± 0.38	0.35 ± 0.30	0.07 ± 0.09	0.14 ± 0.12	0.11 ± 0.10
		GFCI	0.18 ± 0.19	**0.65 ± 0.37**	0.37 ± 0.25	0.08 ± 0.10	0.20 ± 0.18	0.15 ± 0.12
	40	IGFCI	0.14 ± 0.15	**0.37 ± 0.32**	**0.33 ± 0.27**	0.11 ± 0.19	0.07 ± 0.07	0.08 ± 0.07
		GFCI	0.06 ± 0.09	0.24 ± 0.31	0.15 ± 0.18	0.17 ± 0.26	0.09 ± 0.11	0.10 ± 0.13
	60	IGFCI	0.22 ± 0.26	0.27 ± 0.18	0.28 ± 0.21	0.21 ± 0.26	0.09 ± 0.07	0.12 ± 0.14
		GFCI	0.19 ± 0.24	0.26 ± 0.18	0.25 ± 0.20	0.23 ± 0.26	0.13 ± 0.06	0.16 ± 0.14
20	40	IGFCI	**0.68 ± 0.24**	0.74 ± 0.20	**0.74 ± 0.19**	0.24 ± 0.17	0.19 ± 0.08	0.19 ± 0.08
		GFCI	0.38 ± 0.18	0.73 ± 0.23	0.60 ± 0.11	0.34 ± 0.19	0.20 ± 0.09	0.21 ± 0.09
	80	IGFCI	**0.62 ± 0.21**	**0.75 ± 0.19**	**0.68 ± 0.20**	0.25 ± 0.15	0.10 ± 0.05	0.12 ± 0.05
		GFCI	0.46 ± 0.16	0.63 ± 0.22	0.56 ± 0.17	**0.34 ± 0.17**	**0.13 ± 0.06**	**0.15 ± 0.06**
	120	IGFCI	0.59 ± 0.27	0.64 ± 0.22	**0.62 ± 0.20**	0.20 ± 0.07	0.07 ± 0.03	0.09 ± 0.03
		GFCI	0.50 ± 0.28	0.50 ± 0.35	0.48 ± 0.27	0.26 ± 0.14	0.08 ± 0.05	0.11 ± 0.06
50	100	IGFCI	**0.66 ± 0.20**	0.82 ± 0.12	**0.79 ± 0.12**	0.25 ± 0.11	0.20 ± 0.07	0.20 ± 0.07
		GFCI	0.36 ± 0.15	0.79 ± 0.14	0.64 ± 0.08	0.30 ± 0.16	**0.21 ± 0.08**	**0.22 ± 0.09**
Summary statistics		IGFCI	0.45 ± 0.21	0.58 ± 0.19	0.54 ± 0.20	0.19 ± 0.07	0.12 ± 0.05	0.13 ± 0.04
		GFCI	0.30 ± 0.15	0.54 ± 0.20	0.43 ± 0.17	0.24 ± 0.09	0.15 ± 0.05	0.15 ± 0.04

Table 48: Arrowhead precision (P) and recall (R) results for $N = 5000$ training cases. A penalty factor $\kappa = 0.1$ is used to penalize the structural difference between the population-wide and instance-specific CBNs in the first stage of IGFCI. The numbers after '±' are standard deviations. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGFCI	**0.41 ± 0.21**	0.65 ± 0.30	0.60 ± 0.24	0.32 ± 0.31	0.32 ± 0.20	0.30 ± 0.14
		GFCI	0.29 ± 0.28	0.64 ± 0.31	0.50 ± 0.22	0.40 ± 0.39	0.35 ± 0.21	0.35 ± 0.18
	40	IGFCI	0.32 ± 0.31	0.44 ± 0.26	**0.40 ± 0.22**	0.27 ± 0.17	0.17 ± 0.12	0.19 ± 0.11
		GFCI	0.19 ± 0.18	0.35 ± 0.16	0.26 ± 0.09	**0.53 ± 0.21**	**0.32 ± 0.15**	**0.35 ± 0.13**
	60	IGFCI	**0.16 ± 0.16**	0.30 ± 0.24	0.25 ± 0.16	0.30 ± 0.23	0.18 ± 0.12	0.20 ± 0.18
		GFCI	0.07 ± 0.07	0.33 ± 0.23	0.20 ± 0.12	**0.46 ± 0.26**	**0.36 ± 0.10**	**0.36 ± 0.16**
20	40	IGFCI	**0.72 ± 0.21**	0.81 ± 0.24	**0.79 ± 0.22**	0.39 ± 0.22	0.32 ± 0.14	0.32 ± 0.14
		GFCI	0.35 ± 0.07	0.78 ± 0.21	0.62 ± 0.13	**0.55 ± 0.25**	0.36 ± 0.13	**0.37 ± 0.12**
	80	IGFCI	**0.65 ± 0.17**	**0.73 ± 0.13**	**0.69 ± 0.13**	0.32 ± 0.16	0.15 ± 0.05	0.17 ± 0.05
		GFCI	0.38 ± 0.13	0.57 ± 0.13	0.48 ± 0.12	**0.50 ± 0.20**	**0.20 ± 0.07**	**0.23 ± 0.07**
	120	IGFCI	**0.57 ± 0.20**	**0.67 ± 0.11**	**0.63 ± 0.10**	0.26 ± 0.07	0.11 ± 0.05	0.13 ± 0.05
		GFCI	0.34 ± 0.10	0.54 ± 0.15	0.45 ± 0.10	**0.43 ± 0.10**	**0.15 ± 0.06**	**0.18 ± 0.05**
50	100	IGFCI	**0.59 ± 0.16**	0.79 ± 0.17	**0.74 ± 0.14**	0.38 ± 0.14	0.29 ± 0.09	0.30 ± 0.09
		GFCI	0.30 ± 0.05	0.79 ± 0.15	0.61 ± 0.08	**0.44 ± 0.16**	**0.32 ± 0.09**	**0.33 ± 0.09**
Summary statistics		IGFCI	0.49 ± 0.19	0.63 ± 0.18	0.59 ± 0.18	0.32 ± 0.05	0.22 ± 0.08	0.23 ± 0.07
		GFCI	0.27 ± 0.10	0.57 ± 0.17	0.45 ± 0.15	0.47 ± 0.05	0.29 ± 0.08	0.31 ± 0.07

We also computed three types of structural Hamming distance (SHD) to compare the performance of the PAGs learned by IGFCI and GFCI search procedures on each given instance T: S-SHD, L-SHD, A-SHD (see Section 5.3.1.1). Tables 49, 50, and 51 show the average results on the IGFCI and GFCI methods when using $N = \{200, 1000, 5000\}$ training samples, respectively. In these experiments, when using $N = 200$ training instances, IGFCI often performs better in terms of added edges in the *IS* group and reoriented edges, while GFCI has a lower number of deleted edges.

Overall, the average S-SHD is similar using both IGFCI and GFCI methods, where GFCI performs better in terms of L-SHD and A-SHD (Table 49). By increasing the training samples to $N = 1000$, both methods perform better in terms of S-SHD, L-SHD, and A-SHD; however, IGFCI performs slightly better than GFCI in terms of S-SHD but similar in terms of L-SHD and A-SHD. When using $N = 5000$ training instances, GFCI has significantly fewer number of deleted edges, while IGFCI performs significantly better in terms added and reoriented edges, and S-SHD, especially in data-generating CBNs with more variables and edges (e.g., 50 variables and 100 edges). IGFCI's improvement in S-SHD is mainly due to fewer number of added and reoriented edges, especially in the nodes with CSI structure (denoted by IS in tables). Based on these simulations, the IGFCI algorithm often results in less erroneously added and reoriented edges but more deleted edges when compared to the GFCI method.

Table 49: Strict SHD (S-SHD), lenient SHD (L-SHD), and adjacency SHD (A-SHD) for $N = 200$ training cases. A penalty factor $\kappa = 0.1$ is used to penalize the structural difference between the population-wide and instance-specific CBNs in the first stage of IGFCI. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level (the lower the better).

			Added			Deleted			Reoriented					
# Variables	# Edges	Method	IS	Other	Overall	IS	Other	Overall	IS	Other	Overall	S-SHD	L-SHD	A-SHD
10	20	IGFCI	0.2	0.01	0.21	3.27	4.29	7.57	**0.46**	**1.00**	**1.46**	9.24	7.88	7.78
		GFCI	0.34	0.01	0.36	**2.86**	**3.45**	**6.31**	0.73	1.37	2.1	8.76	**6.85**	**6.66**
	40	IGFCI	0.18	0	0.18	7.6	8.51	16.12	0.61	**0.52**	**1.13**	17.43	16.44	16.3
		GFCI	0.31	0	0.31	**6.78**	**7.87**	**14.65**	0.75	0.87	1.62	**16.58**	**15.09**	**14.96**
	60	IGFCI	**0.17**	0	**0.17**	8.1	7.4	15.5	**1.23**	0.69	**1.93**	17.6	15.74	15.67
		GFCI	0.41	0	0.41	**7.22**	**6.72**	**13.94**	1.58	1.05	2.64	**16.99**	**14.42**	**14.35**
20	40	IGFCI	**0.44**	0.01	**0.45**	8.1	14.77	22.87	**1.12**	**1.96**	**3.09**	26.41	23.43	23.32
		GFCI	0.77	0	0.77	**7.48**	**13.76**	**21.24**	1.58	2.65	4.23	26.24	**22.08**	**22.01**
	80	IGFCI	0.45	0.05	0.5	22.83	25.25	48.09	**2.72**	1.71	**4.43**	53.03	48.97	48.59
		GFCI	0.66	**0.02**	0.68	**21.68**	**24.24**	**45.92**	3.68	2.29	5.97	52.56	**46.93**	**46.60**
	120	IGFCI	**0.50**	0.14	0.64	25.49	28.34	53.84	**2.96**	**1.69**	**4.65**	59.13	54.82	54.48
		GFCI	0.69	**0.11**	0.8	**24.03**	**27.34**	**51.38**	4.22	2.63	6.85	59.03	**52.56**	**52.17**
50	100	IGFCI	1.01	0.35	1.36	21.12	41.02	62.13	**3.26**	**4.95**	**8.21**	71.71	63.88	63.49
		GFCI	1.15	**0.11**	1.26	**19.94**	**38.92**	**58.87**	4.02	6.61	10.63	70.76	**60.26**	**60.13**
Summary statistics		IGFCI	0.42	0.08	0.5	13.79	18.51	32.3	1.77	1.79	3.56	36.36	33.03	32.8
		GFCI	0.62	0.04	0.65	12.86	17.47	30.33	2.37	2.49	4.86	35.84	31.17	30.98

Table 50: Strict SHD (S-SHD), lenient SHD (L-SHD), and adjacency SHD (A-SHD) for $N = 1000$ training cases. A penalty factor $\kappa = 0.1$ is used to penalize the structural difference between the population-wide and instance-specific CBNs in the first stage of IGFCI. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level (the lower the better).

			Added			Deleted			Reoriented					
# Variables	# Edges	Method	IS	Other	Overall	IS	Other	Overall	IS	Other	Overall	S-SHD	L-SHD	A-SHD
10	20	IGFCI	**0.24**	0.04	**0.27**	2.59	3.48	6.07	1.07	1.53	2.59	8.94	6.52	6.35
		GFCI	0.65	0.1	0.75	**2.17**	**2.94**	**5.12**	1.42	1.72	3.14	9.01	6.04	5.87
	40	IGFCI	**0.17**	0	**0.17**	5.97	7.68	13.65	**1.82**	**1.32**	**3.14**	16.96	14.13	13.82
		GFCI	0.65	0	0.65	**5.18**	**6.88**	**12.06**	3.01	2.02	5.03	17.74	**13.26**	**12.71**
	60	IGFCI	**0.43**	0	**0.43**	7.18	6.46	13.64	**1.70**	1.58	**3.28**	17.35	14.5	14.07
		GFCI	0.97	0	0.97	**6.24**	**5.45**	**11.68**	2.46	2.48	4.94	17.59	**13.24**	**12.65**
20	40	IGFCI	**0.70**	0.27	**0.97**	6.79	12.11	18.9	**1.71**	2.86	**4.57**	**24.44**	20.34	19.87
		GFCI	1.83	0.19	2.02	**6.08**	**11.46**	**17.54**	2.71	3.3	6	25.56	20.07	19.56
	80	IGFCI	**0.95**	0.14	**1.09**	20.11	22.97	43.08	**4.14**	**2.61**	**6.75**	50.92	44.92	44.17
		GFCI	1.92	0.05	1.97	**18.41**	**21.92**	**40.33**	5.23	3.62	8.85	51.15	**43.52**	**42.30**
	120	IGFCI	**0.69**	0.17	**0.86**	22.84	26.25	49.08	**4.31**	**3.14**	**7.46**	57.4	50.52	49.94
		GFCI	1.59	0.22	1.81	**21.25**	**24.95**	**46.20**	6.01	4.61	10.62	58.63	**49.05**	**48.01**
50	100	IGFCI	**2.11**	0.41	**2.52**	16.92	33.34	50.26	**4.24**	7.11	**11.35**	**64.14**	53.61	52.78
		GFCI	4.37	0.18	4.55	**15.76**	**31.03**	**46.79**	6.56	8.03	14.59	65.92	52.34	**51.34**
Summary statistics		IGFCI	0.75	0.15	0.9	11.77	16.04	27.81	2.71	2.88	5.59	34.31	29.22	28.71
		GFCI	1.71	0.11	1.82	10.73	14.95	25.67	3.91	3.68	7.6	35.09	28.22	27.49

Table 51: Strict SHD (S-SHD), lenient SHD (L-SHD), and adjacency SHD (A-SHD) for $N = 5000$ training cases. A penalty factor $\kappa = 0.1$ is used to penalize the structural difference between the population-wide and instance-specific CBNs in the first stage of IGFCI. Boldface indicates that the results are statistically significantly better, based on Wilcoxon signed rank test at 5% significance level (the lower the better).

			Added			Deleted			Reoriented					
# Variables	# Edges	Method	IS	Other	Overall	IS	Other	Overall	IS	Other	Overall	S-SHD	L-SHD	A-SHD
10	20	IGFCI	**0.42**	0.05	**0.48**	2.07	2.51	4.58	0.95	**1.57**	2.52	7.58	5.35	5.06
		GFCI	1.26	0	1.26	**1.44**	**1.74**	**3.18**	1.46	2.15	3.61	8.04	4.7	4.43
	40	IGFCI	**0.38**	0	**0.38**	5.62	7.09	12.71	**2.13**	**1.88**	**4.01**	**17.10**	13.5	13.1
		GFCI	1.78	0	1.78	**3.70**	**4.82**	**8.52**	4.77	4.78	9.55	19.85	**11.33**	**10.30**
	60	IGFCI	**0.48**	0	**0.48**	6.55	5.55	12.1	**2.51**	**2.42**	**4.92**	**17.51**	13.33	12.59
		GFCI	2.08	0	2.08	**4.17**	**3.95**	**8.11**	5.58	4.09	9.67	19.86	**11.00**	**10.19**
20	40	IGFCI	**0.83**	0.14	**0.97**	5.83	10.1	15.93	**1.86**	2.94	**4.80**	**21.70**	17.55	16.9
		GFCI	2.78	0.17	2.95	**5.26**	**9.19**	**14.45**	3.19	3.29	6.47	23.88	18.07	17.4
	80	IGFCI	**1.08**	0.22	**1.30**	19.16	22.07	41.22	**4.17**	**3.43**	**7.60**	**50.13**	43.58	42.53
		GFCI	3.33	0.09	3.42	**16.69**	**20.29**	**36.98**	7.07	5.58	12.65	53.05	42.63	**40.40**
	120	IGFCI	**0.85**	0.13	**0.99**	21.57	25.86	47.43	**4.53**	**3.23**	**7.76**	**56.18**	49.53	48.42
		GFCI	3.13	**0.01**	3.13	**19.25**	**23.41**	**42.66**	7.21	5.65	12.85	58.64	**48.10**	**45.79**
50	100	IGFCI	**3.09**	0.7	**3.79**	14.4	27.91	42.31	**5.38**	9.23	**14.61**	**60.71**	**48.16**	46.09
		GFCI	7.39	**0.17**	7.56	**13.37**	**26.52**	**39.90**	7.71	9.56	17.27	64.74	50.18	47.46
Summary statistics		IGFCI	1.02	0.18	1.2	10.74	14.44	25.18	3.08	3.53	6.6	32.99	27.28	26.38
		GFCI	3.11	0.06	3.17	9.13	12.85	21.97	5.28	5.01	10.3	35.44	26.57	25.14

5.3.2 Real-world data

We evaluated the performance of IGFCI on multiple real-world datasets that were introduced in Section 4.5.2, which include pneumonia, sepsis, and lung cancer datasets. See Section 4.5.2 for more information about these datasets.

The pneumonia dataset includes 2287 patients, which was split into a training set D with $N = 1601$ samples and a test set with $M = 686$ samples while preserving the distribution of dire outcome in the original dataset. For this dataset, given each instance T in the test set and all the training instances in D, we applied IGFCI search using IS-Score and IS-BSC to learn an instance-specific PAG $\mathcal{P}_{IS}$ for T. We also applied the GFCI search using the BDeu score and BSC test to learn a population-wide PAG $\mathcal{P}_{PW}$ given the training set D. We repeated this procedure for every instance in the test set of the pneumonia dataset.

For the sepsis and lung cancer datasets, we performed leave-one-out cross-validation on each of the datasets. For a given dataset D, we selected a single instance T and used it as the test instance; we used all the remaining instances as the training set. Given each T and D, we learned an instance-specific PAG $\mathcal{P}_{IS}$ for T using IS-GFCI. We repeated this procedure for every instance in D. We also learned a population-wide PAG $\mathcal{P}_{PW}$ for all the instances in D using GFCI.

Since the true causal relationships are not known for these real datasets, as is often the case with real-world datasets in general, we compared the average of structural differences of $\mathcal{P}_{IS}$ versus $\mathcal{P}_{PW}$, assuming $\mathcal{P}_{PW}$ as the reference. We report results using multiple values of κ ($0.0 < \kappa \leq 1.0$). The results are shown in Table 52. The results indicate that for lower values of κ (e.g., 0.001), the structural differences are lower because a lower value of κ penalizes more the structural difference between the population-wide BN and instance-specific BN model. The results also indicate that as the data includes fewer variables and more instances (e.g., sepsis dataset), the structural differences decrease between $\mathcal{P}_{PW}$ and $\mathcal{P}_{IS}$. Since we do not know the true causal structures for the real datasets, we cannot determine whether IGFCI or GFCI is performing better in learning the causal structures. The results do show, however, that instance-specific causal structure frequently exists when we learn PAGs from real-world data.

Table 52: Average strict SHD (S-SHD), lenient SHD (L-SHD), and adjacency SHD (A-SHD) between the instance-specific PAGs $\mathcal{P}_{IS}$ found by IGFCI and the population-wide PAG $\mathcal{P}_{PW}$ found by GFCI, where $\mathcal{P}_{PW}$ is considered as the reference, on the real-world datasets. κ is a parameter that penalizes the structural difference between the population-wide BN and instance-specific BN model in the first stage of IGFCI.

Dataset	# Patients	# Variables	κ	Added	Deleted	Reoriented	S-SHD	L-SHD	A-SHD
Pneumonia	train:1601	41	0.001	6.33	13.06	6.83	26.22	19.40	19.40
			0.1	13.07	15.48	11.23	39.77	28.58	28.55
	test: 686		0.5	23.10	18.73	10.93	52.76	41.85	41.83
			0.9	26.07	20.70	9.04	55.80	46.77	46.76
Sepsis	1673	21	0.001	0.27	5.28	0.00	5.56	5.56	5.56
			0.1	1.53	4.80	0.91	7.24	6.33	6.33
			0.5	2.37	5.32	1.79	9.48	7.69	7.69
			0.9	3.08	5.56	1.67	10.30	8.65	8.64
Lung cancer	261	42	0.001	2.46	15.45	0.51	18.41	17.91	17.91
			0.1	4.94	12.26	2.75	19.95	17.20	17.20
			0.5	6.72	12.54	5.49	24.75	19.26	19.26
			0.9	10.51	13.79	7.25	31.55	24.31	24.31

5.4 Summary and Discussion

The instance-specific IGES method introduced in Chapter 4 builds a causal model for a given instance assuming causal sufficiency, but this assumption rarely holds in practice. The current chapter introduced an instance-specific PAG-learning algorithm called IGFCI that outputs a PAG that is specific to a given instance T (e.g, a patient) by guiding causal model search based on the attributes of T. The approach used by IGFCI is quite general and can be readily applied to develop an instance-specific version of other graphical causal discovery methods.

The empirical results we obtained on simulated data for discovering the instance-specific PAG structure of each test instance T indicate that when fewer samples are available (i.e., $N = 200$), IGFCI performs similar to GFCI in terms of adjacency P and arrowhead P and R, but GFCI performs slightly better in terms of adjacency R and SHD, where the differences are due to missing edges by IGFCI. However, when the sample size is sufficiently large (i.e., $N = 5000$), IGFCI performs better in terms of adjacency and arrowhead P, erroneously added and reoriented edges, and S-SHD. On the other hand, GFCI performs better in terms of adjacency and arrowhead R, erroneously deleted edges, L-SHD, and A-SHD. We conjecture that the missing edges are weak enough to make instance-specific detection difficult without more samples.

Overall, the proposed IGFCI method is a promising approach to discover a PAG structure that better models the relationships among variables of a given instance T in terms of adjacency and arrowhead P, and fewer edge addition and reorientation errors, rather than a population-wide model, which partially supports the third hypothesis in Section 1.2, which states that the combination of instance-specific modeling and Bayesian scoring of constraints will perform CBN structure learning better than either method alone, in terms of discrimination.

6.0 Conclusion and Future Work

This dissertation introduces and investigates three novel CBN structure learning algorithms:

1. A hybrid CBN structure learning method that uses Bayesian scoring of constraints (BSC): this contribution addresses limitations of constraint-based algorithms to recover CBN structures that may contain latent confounders.
2. A score-based instance-specific CBN structure learning method (IGES): this contribution addresses the necessity and importance of instance-specific causal modeling and discovery in heterogeneous domains, such as human biology. This method relies on the causal sufficiency assumption (i.e., there are no latent confounders).
3. A hybrid instance-specific CBN structure learning method (IGFCI): similar to the second contribution, this algorithm performs instance-specific causal discovery in heterogeneous domains. In contrast to IGES, this algorithm is able to model latent confounding by combining the first and second contributions.

The experimental evaluations of these algorithms can be expanded in numerous ways. As an example, it would be interesting to evaluate the structure discovery performance of these algorithms on real biomedical datasets for which the causal knowledge is available. Developing more informative structure and parameter prior probabilities would also be helpful. Moreover, the evaluations can be extended by varying the hyperparameters of the introduced methods (e.g., IGFCI) and of the previously existing methods (e.g., GFCI) and then plotting and comparing precision-recall curves of each method. Additionally, the Bayesian methods that we have introduced are amenable to Bayesian modeling averaging. The remainder of this chapter provides more detailed conclusions and suggestions for future work for each of the above three contributions.

6.1 Bayesian Scoring of Constraints

Chapter 3 introduces a Bayesian method called BSC to compute the posterior probability of a constraint $R_i = (X \perp\!\!\!\perp Y | \boldsymbol{Z})$. The BSC method can then be incorporated into a constraint-based algorithm (e.g., FCI) to learn multiple PAG structures; we call this algorithm FCI-BSC. Also, we introduced three scoring methods to compute the posterior probability of the PAGs (using BSC) and output the highest scoring PAG. We proved theorems that under assumptions show for CBNs with discrete/continuous variable types, BSC assigns the correct constraint hypothesis in the large sample limit; therefore, FCI-BSC will recover the data-generating PAG structure in the large sample limit. We performed experiments on a wide range of simulated data from randomly generated BNs that contain discrete, continuous, and a mixture of discrete and continuous variable types. The experimental results show that the FCI-BSC method performed similarly compared to FCI (with a chi-squared test) for discrete data. However, for continuous and mixed data, FCI-BSC performed better in terms of adjacency and arrowhead precision, and SHD measures, but performed worse in terms of adjacency and arrowhead recall, compared to FCI (with commonly used frequentist statistical tests). We also evaluated this method using simulated data generated from manually constructed benchmark BNs that contain discrete variables. For BNs with denser structures and more parameters, FCI-BSC performed better in terms of adjacency and arrowhead precision, and SHD measures, but performed worse in terms of adjacency and arrowhead recall, compared to FCI (using a chi-squared test). For almost all simulations, all scoring methods performed similarly in terms of ranking the highest scoring PAG. The theoretical and experimental results partially support the first hypothesis: *the BSC method will perform CBN structure learning better than a method that uses frequentist statistical tests in terms of discrimination.*

A primary use of CBN structure learning methods is to analyze observational data to generate novel causal hypotheses that are likely to be correct when subjected to experimental validation. Such an approach could significantly increase the efficiency of causal discovery in science. To make informed decisions about which novel causal hypotheses to investigate experimentally, scientists need to know how likely the hypotheses are to be confirmed. A

causal discovery method that has a better precision performance, as BSC does, will have a higher success rate in confirming a causal hypothesis; thus, such a method can help scientists prioritize experiments.

The BSC method can be extended in several ways, including the following:

- Understand better the reason for the relatively lower recall of BSC and try to increase it while retaining precision. BSC performed poorly on unconditional independence queries, especially for discrete variables, which results in too many edge removals at the early stage of the FCI search. One possible solution to this problem is to develop informative prior probabilities on constraints. We currently assume that dependence and independence are a priori equally likely; however, this assumption is not true in general.
- Develop other hybrid PAG learning algorithms by combining other constraint-based methods (e.g., RFCI [Colombo et al., 2012])) with the BSC method.
- As described in Section 3.5, we can use model averaging to estimate the probability distribution over the edge types of output PAGs as follows: Since PAGs are being sampled (generated) according to their posterior distribution, the probability of edge E existing between nodes X_i and X_j is estimated as the fraction of the sampled PAGs that contain edge E between X_i and X_j. These probabilities can then be used to study the calibration performance on the edge-type probabilities produced by FCI-BSC by measuring the expected calibration error (ECE) and maximum calibration error (MCE) [Naeini et al., 2015, Jabbari et al., 2017a]. We say that a method has good calibration if models that are predicted to be true with probability p, are true about p fraction of the time. Producing well-calibrated probabilities is important when making decisions using decision theory. As an example, a well-calibrated causal discovery algorithm can help scientists prioritize which causal hypotheses to investigate experimentally depending on how high are the calibrated probability of those hypotheses.
- In terms of theoretical work, I conjecture that the BSC method for mixed data using the degenerate Gaussian score [Andrews et al., 2019] is correct in the large sample limit, under assumptions made in [Andrews et al., 2019]. The outline of the proof will be similar to the proof of correctness for BSC using the BIC score (Theorem 3.3.2).

- I conjecture that in the large sample limit, the independence constraints asked by FCI-BSC are independent of each other, and as a result, the BSC-I scoring method is correct. However, in general, independence does not hold when the sample size is finite, as can be readily shown through examples. It would be interesting to study the convergence results for the BSC-D and BSC-LD scoring methods with finite sample size, but with (effectively) an infinitely large number of bootstrap samples.

6.2 Instant-Specific Causal Discovery without Modeling Latent Confounding

Chapter 4 introduces a score-based instance-specific CBN learning algorithm, called IGES, that learns a CBN for a given test instance T by utilizing the information we have about T as well as the information on many other training instances. The order of the computational time complexity of IGES is the same as that of its population-wide counterpart (i.e., GES), while it has a substantially smaller search space compared to the algorithms that try to model all CSI structures (e.g., using decision graphs [Chickering et al., 1997]), rather than representing CSI structures for T only. We proved theorems that under reasonable assumptions IGES will recover the data-generating CBN for T, which encodes the CSI structures associated with T, in the large sample limit.

We also studied the performance of IGES on simulated and real-world biomedical datasets. On simulated data, IGES outperformed its population-wide counterpart, GES, in terms of adjacency and arrowhead precision (especially for the nodes with CSI structures). However, IGES performed worse in terms of recall for small sample sizes, while both methods had comparable recalls as the sample size increased. For moderate to large datasets, IGES had better SHD performance compared to GES. Using real-world biomedical datasets, we compared the predictive performance of target variables using AUROC and observed that instance-specific CBNs better predicted the target variables. The theoretical and experimental results support the second hypothesis: *the instance-specific CBN structure learning approach will perform structure learning better than a population-wide method, in terms of discrimination. Since IGES has a better precision performance compared to GES,*

it could help scientists prioritize experiments since such algorithms will have a higher success rate in suggesting a causal hypothesis that will be confirmed.

There are several directions for extending the IGES method, including the following:

- Understand better the reason for the relatively lower recall of the instance-specific BN models and try to increase it while retaining precision.
- Extend the IGES algorithm to iteratively learn an instance-specific model for each instance in the training set and use an aggregate of those instance-specific models to define the population-wide model.
- Generalize the type of instant-specific models beyond CSIs. One general framework is using decision-graphs to represent conditional probability tables (CPTs), which was introduced by [Chickering et al., 1997]. A decision-graph is a generalization of a decision tree. It represents the distribution of a node X_i given its parents $\boldsymbol{Pa}(X_i)$ as a set of disjunctions of X_i's parents instantiations. Doing so enables it to capture CSI structures, as well as other predictive patterns. To use decision graphs, we could first run a population-wide search (e.g., GES) to learn a population-wide CBN. Then using the decision-graph representation of each node given its parents, apply local search in a way that is influenced by the decision path of the given test instance T to find an instance-specific decision-graph for each node X_i given its parents $\boldsymbol{Pa}(X_i)$.
- Develop an instance-specific score to learn BN structures that contain other types of variables (e.g., continuous or a mixture of continuous and discrete variables). For continuous variables, this extension involves developing approximate instance-matching methods to cluster training instances while learning an instance-specific CBN, since unlike the discrete variable type, exact instance-matching would not work in this case. For example, [Lengerich et al., 2019] used Euclidean distance to cluster the training instances based on how similar are their covariates (e.g., variable-value pairs) to a given test instance. As described in Section 4.1.2, they developed a method to learn an instance-specific regression model by using a distance-matching regularizer that regularizes regression parameters by assuming the similarity in parameters correspond to the similarity in features of instances. A Bayesian version of this method can be adapted and integrated into a CBN structure learning search to learn instance-specific CBN structures.

6.3 Instant-Specific Causal Discovery with Modeling Latent Confounding

Chapter 5 introduces an instance-specific PAG-learning algorithm, called IGFCI, that combines the BSC and IGES methods to learn an instance-specific PAG structure for a given instance T by utilizing the attributes of T as well as the training samples. The experimental results on simulation studies indicate that IGFCI outperformed its population-wide counterpart (i.e., GFCI) in terms of adjacency and arrowhead precision, and S-SHD, when the sample size was sufficiently large. On the other hand, GFCI performed relatively better in terms of adjacency and arrowhead recall, L-SHD, and A-SHD. The experiments on real-world biomedical datasets show that the PAG structures learned by the IGFCI algorithm are different from the PAGs learned by the GFCI algorithm. These results partially support the third hypothesis: *the combination of instance-specific modeling and Bayesian scoring of constraints will perform CBN structure learning better than either method alone, in terms of discrimination.*

The IGFCI method can be extended in the following ways:

- Develop other instance-specific PAG learning algorithms by combining the instance-specific BSC with other constraint-based methods such as FCI [Spirtes et al., 2000] or RFCI [Colombo et al., 2012].
- Develop an instance-specific method to learn PAG structures that contain other types of variables (e.g., continuous or a mixture of continuous and discrete variables).
- In terms of theoretical work, it would be interesting to attempt to prove that IGFCI is guaranteed to find the data-generating instance-specific PAG for a given test instance in the large sample limit.

Despite the limitations, this dissertation provides support that the instance-specific CBN structure learning methods are promising approaches to discover a CBN structure that better models the relationships among variables of a given instance T, rather than a population-wide model. The results suggest that further investigation of the approach is warranted both in the form of extensions of the methods to improve recall while maintaining precision, and in the form of expanding theoretical and experimental findings.

Appendix A Additional Results from Chapter 4

In this appendix, I report the average results for the full experiments that are done in simulations of Chapter 4.

Table 53: Adjacency precision (P) and recall (R) results for $N = 200$ training cases.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGES ($\kappa = 0.001$)	0.69 ± 0.19	0.94 ± 0.09	0.90 ± 0.09	0.41 ± 0.22	0.36 ± 0.14	0.37 ± 0.08
		IGES ($\kappa = 0.1$)	0.73 ± 0.14	0.94 ± 0.10	0.88 ± 0.09	0.43 ± 0.14	0.42 ± 0.12	0.42 ± 0.10
		IGES ($\kappa = 0.5$)	0.70 ± 0.17	0.94 ± 0.07	0.88 ± 0.08	0.42 ± 0.16	0.43 ± 0.11	0.42 ± 0.10
		IGES ($\kappa = 0.9$)	0.67 ± 0.20	0.92 ± 0.09	0.85 ± 0.09	0.41 ± 0.13	0.45 ± 0.13	0.42 ± 0.09
		GES	0.75 ± 0.16	0.97 ± 0.05	0.89 ± 0.07	0.60 ± 0.18	0.50 ± 0.11	0.54 ± 0.08
	40	IGES ($\kappa = 0.001$)	0.75 ± 0.10	0.87 ± 0.15	0.86 ± 0.08	0.26 ± 0.08	0.25 ± 0.10	0.25 ± 0.08
		IGES ($\kappa = 0.1$)	0.79 ± 0.10	0.89 ± 0.10	0.87 ± 0.06	0.29 ± 0.07	0.26 ± 0.08	0.27 ± 0.07
		IGES ($\kappa = 0.5$)	0.77 ± 0.12	0.91 ± 0.10	0.85 ± 0.07	0.31 ± 0.07	0.29 ± 0.08	0.29 ± 0.06
		IGES ($\kappa = 0.9$)	0.75 ± 0.13	0.89 ± 0.10	0.83 ± 0.09	0.32 ± 0.06	0.29 ± 0.08	0.30 ± 0.06
		GES	0.76 ± 0.13	0.90 ± 0.12	0.84 ± 0.11	0.35 ± 0.06	0.32 ± 0.07	0.33 ± 0.05
	60	IGES ($\kappa = 0.001$)	0.80 ± 0.13	0.91 ± 0.15	0.93 ± 0.06	0.35 ± 0.11	0.23 ± 0.11	0.28 ± 0.10
		IGES ($\kappa = 0.1$)	0.85 ± 0.08	0.92 ± 0.13	0.94 ± 0.05	0.38 ± 0.11	0.23 ± 0.08	0.30 ± 0.09
		IGES ($\kappa = 0.5$)	0.84 ± 0.07	0.93 ± 0.13	0.92 ± 0.06	0.40 ± 0.10	0.24 ± 0.08	0.31 ± 0.09
		IGES ($\kappa = 0.9$)	0.84 ± 0.07	0.94 ± 0.11	0.91 ± 0.04	0.41 ± 0.08	0.27 ± 0.08	0.32 ± 0.08
		GES	0.85 ± 0.09	0.99 ± 0.02	0.92 ± 0.07	0.43 ± 0.09	0.29 ± 0.07	0.34 ± 0.06
20	40	IGES ($\kappa = 0.001$)	0.79 ± 0.15	0.97 ± 0.04	0.92 ± 0.06	0.38 ± 0.14	0.31 ± 0.11	0.33 ± 0.11
		IGES ($\kappa = 0.1$)	0.83 ± 0.10	0.95 ± 0.07	0.89 ± 0.07	0.44 ± 0.14	0.37 ± 0.06	0.39 ± 0.08
		IGES ($\kappa = 0.5$)	0.76 ± 0.10	0.86 ± 0.09	0.82 ± 0.05	0.52 ± 0.12	0.39 ± 0.08	0.44 ± 0.08
		IGES ($\kappa = 0.9$)	0.71 ± 0.10	0.79 ± 0.08	0.75 ± 0.08	0.50 ± 0.12	0.40 ± 0.06	0.43 ± 0.06
		GES	0.81 ± 0.08	0.95 ± 0.07	0.89 ± 0.07	0.53 ± 0.12	0.43 ± 0.11	0.47 ± 0.06
	80	IGES ($\kappa = 0.001$)	0.84 ± 0.11	0.90 ± 0.16	0.89 ± 0.07	0.22 ± 0.08	0.18 ± 0.08	0.21 ± 0.05
		IGES ($\kappa = 0.1$)	0.87 ± 0.07	0.92 ± 0.12	0.90 ± 0.05	0.35 ± 0.07	0.24 ± 0.08	0.29 ± 0.07
		IGES ($\kappa = 0.5$)	0.81 ± 0.05	0.92 ± 0.05	0.85 ± 0.04	0.34 ± 0.07	0.29 ± 0.03	0.31 ± 0.04
		IGES ($\kappa = 0.9$)	0.72 ± 0.09	0.89 ± 0.08	0.79 ± 0.08	0.36 ± 0.10	0.28 ± 0.06	0.32 ± 0.06
		GES	0.86 ± 0.05	0.93 ± 0.15	0.90 ± 0.04	0.40 ± 0.08	0.29 ± 0.09	0.34 ± 0.06
	120	IGES ($\kappa = 0.001$)	0.85 ± 0.08	0.90 ± 0.12	0.89 ± 0.06	0.23 ± 0.06	0.15 ± 0.07	0.19 ± 0.06
		IGES ($\kappa = 0.1$)	0.89 ± 0.09	0.96 ± 0.04	0.93 ± 0.06	0.28 ± 0.07	0.19 ± 0.07	0.23 ± 0.07
		IGES ($\kappa = 0.5$)	0.86 ± 0.07	0.94 ± 0.05	0.89 ± 0.06	0.30 ± 0.04	0.19 ± 0.04	0.25 ± 0.04
		IGES ($\kappa = 0.9$)	0.70 ± 0.11	0.87 ± 0.06	0.76 ± 0.07	0.29 ± 0.04	0.19 ± 0.05	0.24 ± 0.03
		GES	0.86 ± 0.09	0.96 ± 0.05	0.90 ± 0.06	0.30 ± 0.05	0.19 ± 0.06	0.24 ± 0.03
50	100	IGES ($\kappa = 0.001$)	0.90 ± 0.07	1.00 ± 0.01	0.95 ± 0.04	0.39 ± 0.10	0.30 ± 0.09	0.33 ± 0.09
		IGES ($\kappa = 0.1$)	0.85 ± 0.04	0.92 ± 0.05	0.88 ± 0.04	0.43 ± 0.07	0.35 ± 0.05	0.38 ± 0.05
		IGES ($\kappa = 0.5$)	0.76 ± 0.07	0.82 ± 0.04	0.79 ± 0.04	0.45 ± 0.08	0.39 ± 0.06	0.41 ± 0.06
		IGES ($\kappa = 0.9$)	0.59 ± 0.06	0.66 ± 0.07	0.62 ± 0.05	0.50 ± 0.06	0.47 ± 0.08	0.48 ± 0.06
		GES	0.85 ± 0.04	0.98 ± 0.03	0.92 ± 0.03	0.47 ± 0.06	0.41 ± 0.06	0.43 ± 0.05
	200	IGES ($\kappa = 0.001$)	0.89 ± 0.05	0.98 ± 0.03	0.93 ± 0.04	0.26 ± 0.07	0.18 ± 0.05	0.21 ± 0.05
		IGES ($\kappa = 0.1$)	0.86 ± 0.05	0.93 ± 0.03	0.89 ± 0.04	0.31 ± 0.04	0.20 ± 0.03	0.24 ± 0.03
		IGES ($\kappa = 0.5$)	0.76 ± 0.05	0.85 ± 0.07	0.80 ± 0.05	0.31 ± 0.04	0.21 ± 0.03	0.25 ± 0.03
		IGES ($\kappa = 0.9$)	0.62 ± 0.09	0.75 ± 0.06	0.67 ± 0.07	0.34 ± 0.06	0.23 ± 0.04	0.28 ± 0.05
		GES	0.87 ± 0.05	0.97 ± 0.03	0.91 ± 0.04	0.33 ± 0.06	0.22 ± 0.04	0.27 ± 0.05
	300	IGES ($\kappa = 0.001$)	0.92 ± 0.04	0.98 ± 0.03	0.94 ± 0.02	0.19 ± 0.06	0.12 ± 0.05	0.15 ± 0.05
		IGES ($\kappa = 0.1$)	0.89 ± 0.04	0.97 ± 0.03	0.92 ± 0.03	0.25 ± 0.07	0.16 ± 0.03	0.19 ± 0.05
		IGES ($\kappa = 0.5$)	0.75 ± 0.05	0.86 ± 0.05	0.80 ± 0.04	0.29 ± 0.05	0.19 ± 0.04	0.24 ± 0.04
		IGES ($\kappa = 0.9$)	0.66 ± 0.06	0.80 ± 0.07	0.71 ± 0.06	0.32 ± 0.04	0.21 ± 0.04	0.26 ± 0.04
		GES	0.89 ± 0.04	0.99 ± 0.02	0.93 ± 0.03	0.29 ± 0.06	0.22 ± 0.03	0.25 ± 0.04

Table 54: Adjacency precision (P) and recall (R) results for $N = 1000$ training cases.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGES ($\kappa = 0.001$)	0.78 ± 0.13	0.93 ± 0.07	0.88 ± 0.07	0.74 ± 0.13	0.66 ± 0.14	0.68 ± 0.13
		IGES ($\kappa = 0.1$)	0.82 ± 0.10	0.93 ± 0.06	0.89 ± 0.06	0.72 ± 0.13	0.63 ± 0.12	0.65 ± 0.11
		IGES ($\kappa = 0.5$)	0.82 ± 0.10	0.92 ± 0.06	0.88 ± 0.06	0.72 ± 0.14	0.62 ± 0.12	0.65 ± 0.12
		IGES ($\kappa = 0.9$)	0.80 ± 0.12	0.91 ± 0.05	0.86 ± 0.06	0.72 ± 0.13	0.62 ± 0.11	0.64 ± 0.11
		GES	0.72 ± 0.11	0.93 ± 0.06	0.83 ± 0.07	0.83 ± 0.11	0.70 ± 0.14	0.73 ± 0.12
	40	IGES ($\kappa = 0.001$)	0.83 ± 0.09	0.96 ± 0.05	0.88 ± 0.07	0.54 ± 0.11	0.41 ± 0.10	0.47 ± 0.11
		IGES ($\kappa = 0.1$)	0.83 ± 0.07	0.97 ± 0.04	0.88 ± 0.05	0.53 ± 0.12	0.40 ± 0.06	0.46 ± 0.09
		IGES ($\kappa = 0.5$)	0.84 ± 0.07	0.96 ± 0.04	0.89 ± 0.05	0.55 ± 0.11	0.43 ± 0.05	0.49 ± 0.08
		IGES ($\kappa = 0.9$)	0.83 ± 0.08	0.96 ± 0.03	0.88 ± 0.06	0.55 ± 0.10	0.43 ± 0.04	0.49 ± 0.07
		GES	0.77 ± 0.08	0.98 ± 0.03	0.85 ± 0.05	0.60 ± 0.07	0.47 ± 0.10	0.53 ± 0.07
	60	IGES ($\kappa = 0.001$)	0.81 ± 0.10	0.95 ± 0.13	0.90 ± 0.07	0.41 ± 0.08	0.38 ± 0.12	0.39 ± 0.09
		IGES ($\kappa = 0.1$)	0.78 ± 0.10	0.96 ± 0.08	0.87 ± 0.07	0.43 ± 0.10	0.36 ± 0.09	0.38 ± 0.09
		IGES ($\kappa = 0.5$)	0.79 ± 0.10	0.96 ± 0.08	0.87 ± 0.07	0.45 ± 0.09	0.36 ± 0.09	0.39 ± 0.09
		IGES ($\kappa = 0.9$)	0.79 ± 0.09	0.95 ± 0.07	0.87 ± 0.07	0.44 ± 0.09	0.35 ± 0.08	0.39 ± 0.08
		GES	0.72 ± 0.12	0.96 ± 0.08	0.84 ± 0.09	0.48 ± 0.10	0.43 ± 0.11	0.45 ± 0.09
20	40	IGES ($\kappa = 0.001$)	0.82 ± 0.09	0.97 ± 0.04	0.90 ± 0.06	0.70 ± 0.08	0.64 ± 0.09	0.66 ± 0.07
		IGES ($\kappa = 0.1$)	0.88 ± 0.09	0.93 ± 0.06	0.91 ± 0.07	0.69 ± 0.08	0.58 ± 0.07	0.61 ± 0.06
		IGES ($\kappa = 0.5$)	0.85 ± 0.05	0.94 ± 0.02	0.89 ± 0.03	0.73 ± 0.09	0.65 ± 0.07	0.68 ± 0.08
		IGES ($\kappa = 0.9$)	0.75 ± 0.09	0.87 ± 0.06	0.82 ± 0.07	0.72 ± 0.08	0.67 ± 0.05	0.69 ± 0.04
		GES	0.70 ± 0.06	0.98 ± 0.03	0.84 ± 0.04	0.74 ± 0.06	0.67 ± 0.08	0.70 ± 0.05
	80	IGES ($\kappa = 0.001$)	0.78 ± 0.05	0.95 ± 0.06	0.84 ± 0.04	0.51 ± 0.09	0.41 ± 0.06	0.46 ± 0.08
		IGES ($\kappa = 0.1$)	0.88 ± 0.06	0.96 ± 0.05	0.91 ± 0.05	0.50 ± 0.05	0.40 ± 0.07	0.45 ± 0.05
		IGES ($\kappa = 0.5$)	0.82 ± 0.04	0.91 ± 0.05	0.86 ± 0.04	0.51 ± 0.06	0.42 ± 0.04	0.46 ± 0.04
		IGES ($\kappa = 0.9$)	0.78 ± 0.06	0.90 ± 0.04	0.83 ± 0.04	0.52 ± 0.07	0.43 ± 0.06	0.47 ± 0.06
		GES	0.73 ± 0.05	0.98 ± 0.03	0.83 ± 0.03	0.55 ± 0.06	0.43 ± 0.03	0.48 ± 0.04
	120	IGES ($\kappa = 0.001$)	0.83 ± 0.08	0.96 ± 0.04	0.88 ± 0.05	0.48 ± 0.08	0.37 ± 0.06	0.42 ± 0.06
		IGES ($\kappa = 0.1$)	0.85 ± 0.06	0.92 ± 0.06	0.88 ± 0.06	0.46 ± 0.05	0.32 ± 0.06	0.39 ± 0.06
		IGES ($\kappa = 0.5$)	0.83 ± 0.05	0.90 ± 0.05	0.86 ± 0.04	0.45 ± 0.07	0.35 ± 0.08	0.40 ± 0.07
		IGES ($\kappa = 0.9$)	0.80 ± 0.07	0.89 ± 0.03	0.83 ± 0.05	0.49 ± 0.04	0.36 ± 0.08	0.43 ± 0.05
		GES	0.76 ± 0.08	0.97 ± 0.04	0.83 ± 0.04	0.49 ± 0.07	0.37 ± 0.06	0.43 ± 0.05
50	100	IGES ($\kappa = 0.001$)	0.86 ± 0.05	0.98 ± 0.02	0.93 ± 0.03	0.66 ± 0.07	0.63 ± 0.07	0.64 ± 0.06
		IGES ($\kappa = 0.1$)	0.88 ± 0.05	0.96 ± 0.01	0.93 ± 0.02	0.68 ± 0.05	0.61 ± 0.07	0.64 ± 0.05
		IGES ($\kappa = 0.5$)	0.79 ± 0.06	0.89 ± 0.02	0.84 ± 0.03	0.69 ± 0.05	0.66 ± 0.06	0.67 ± 0.04
		IGES ($\kappa = 0.9$)	0.74 ± 0.07	0.79 ± 0.06	0.77 ± 0.05	0.72 ± 0.04	0.64 ± 0.04	0.67 ± 0.02
		GES	0.74 ± 0.06	0.99 ± 0.01	0.86 ± 0.03	0.75 ± 0.05	0.61 ± 0.07	0.66 ± 0.04
	200	IGES ($\kappa = 0.001$)	0.86 ± 0.07	0.97 ± 0.03	0.91 ± 0.05	0.47 ± 0.07	0.34 ± 0.06	0.39 ± 0.06
		IGES ($\kappa = 0.1$)	0.88 ± 0.03	0.93 ± 0.04	0.91 ± 0.03	0.52 ± 0.06	0.38 ± 0.06	0.44 ± 0.06
		IGES ($\kappa = 0.5$)	0.79 ± 0.06	0.89 ± 0.04	0.84 ± 0.04	0.49 ± 0.05	0.38 ± 0.05	0.43 ± 0.05
		IGES ($\kappa = 0.9$)	0.73 ± 0.06	0.86 ± 0.02	0.79 ± 0.03	0.52 ± 0.03	0.40 ± 0.07	0.45 ± 0.05
		GES	0.76 ± 0.05	0.97 ± 0.03	0.86 ± 0.03	0.53 ± 0.03	0.38 ± 0.05	0.44 ± 0.04
	300	IGES ($\kappa = 0.001$)	0.86 ± 0.03	0.96 ± 0.03	0.90 ± 0.03	0.42 ± 0.05	0.29 ± 0.04	0.35 ± 0.05
		IGES ($\kappa = 0.1$)	0.88 ± 0.03	0.95 ± 0.02	0.91 ± 0.02	0.46 ± 0.04	0.31 ± 0.05	0.37 ± 0.04
		IGES ($\kappa = 0.5$)	0.82 ± 0.03	0.92 ± 0.03	0.87 ± 0.02	0.45 ± 0.04	0.33 ± 0.03	0.39 ± 0.04
		IGES ($\kappa = 0.9$)	0.72 ± 0.04	0.86 ± 0.03	0.78 ± 0.03	0.45 ± 0.03	0.34 ± 0.03	0.39 ± 0.03
		GES	0.77 ± 0.02	0.96 ± 0.02	0.85 ± 0.02	0.43 ± 0.03	0.31 ± 0.03	0.36 ± 0.03

Table 55: Adjacency precision (P) and recall (R) results for $N = 5000$ training cases.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGES ($\kappa = 0.001$)	0.82 ± 0.09	0.93 ± 0.06	0.89 ± 0.06	0.84 ± 0.08	0.86 ± 0.07	0.85 ± 0.03
		IGES ($\kappa = 0.1$)	0.86 ± 0.10	0.93 ± 0.07	0.90 ± 0.07	0.83 ± 0.06	0.83 ± 0.07	0.83 ± 0.05
		IGES ($\kappa = 0.5$)	0.86 ± 0.11	0.91 ± 0.08	0.90 ± 0.07	0.81 ± 0.07	0.81 ± 0.08	0.81 ± 0.06
		IGES ($\kappa = 0.9$)	0.86 ± 0.11	0.90 ± 0.08	0.89 ± 0.08	0.79 ± 0.07	0.80 ± 0.10	0.80 ± 0.07
		GES	0.56 ± 0.08	0.93 ± 0.07	0.75 ± 0.06	0.92 ± 0.08	0.90 ± 0.05	0.90 ± 0.05
	40	IGES ($\kappa = 0.001$)	0.78 ± 0.13	0.97 ± 0.05	0.86 ± 0.09	0.72 ± 0.09	0.64 ± 0.10	0.67 ± 0.08
		IGES ($\kappa = 0.1$)	0.81 ± 0.11	0.96 ± 0.05	0.87 ± 0.07	0.69 ± 0.08	0.59 ± 0.07	0.63 ± 0.07
		IGES ($\kappa = 0.5$)	0.81 ± 0.10	0.96 ± 0.05	0.87 ± 0.07	0.67 ± 0.06	0.54 ± 0.06	0.59 ± 0.06
		IGES ($\kappa = 0.9$)	0.81 ± 0.10	0.95 ± 0.04	0.86 ± 0.07	0.66 ± 0.05	0.54 ± 0.06	0.59 ± 0.06
		GES	0.60 ± 0.10	0.96 ± 0.07	0.75 ± 0.08	0.80 ± 0.11	0.74 ± 0.10	0.76 ± 0.09
	60	IGES ($\kappa = 0.001$)	0.72 ± 0.12	0.98 ± 0.04	0.84 ± 0.08	0.61 ± 0.08	0.57 ± 0.09	0.59 ± 0.08
		IGES ($\kappa = 0.1$)	0.76 ± 0.11	0.97 ± 0.04	0.85 ± 0.07	0.65 ± 0.08	0.55 ± 0.07	0.59 ± 0.06
		IGES ($\kappa = 0.5$)	0.72 ± 0.10	0.97 ± 0.04	0.84 ± 0.07	0.59 ± 0.06	0.53 ± 0.06	0.56 ± 0.04
		IGES ($\kappa = 0.9$)	0.76 ± 0.11	0.94 ± 0.06	0.84 ± 0.07	0.60 ± 0.05	0.51 ± 0.08	0.55 ± 0.04
		GES	0.59 ± 0.11	0.98 ± 0.05	0.74 ± 0.07	0.69 ± 0.06	0.60 ± 0.08	0.64 ± 0.04
20	40	IGES ($\kappa = 0.001$)	0.84 ± 0.12	0.94 ± 0.06	0.89 ± 0.08	0.86 ± 0.06	0.82 ± 0.05	0.84 ± 0.04
		IGES ($\kappa = 0.1$)	0.84 ± 0.07	0.93 ± 0.07	0.89 ± 0.06	0.82 ± 0.08	0.79 ± 0.05	0.80 ± 0.05
		IGES ($\kappa = 0.5$)	0.81 ± 0.08	0.89 ± 0.07	0.86 ± 0.06	0.82 ± 0.05	0.80 ± 0.04	0.80 ± 0.04
		IGES ($\kappa = 0.9$)	0.79 ± 0.10	0.87 ± 0.07	0.83 ± 0.07	0.76 ± 0.07	0.77 ± 0.09	0.76 ± 0.07
		GES	0.60 ± 0.06	0.93 ± 0.07	0.78 ± 0.05	0.86 ± 0.08	0.82 ± 0.12	0.84 ± 0.07
	80	IGES ($\kappa = 0.001$)	0.76 ± 0.07	0.88 ± 0.06	0.82 ± 0.06	0.64 ± 0.07	0.59 ± 0.10	0.62 ± 0.07
		IGES ($\kappa = 0.1$)	0.84 ± 0.06	0.92 ± 0.05	0.88 ± 0.05	0.64 ± 0.05	0.58 ± 0.07	0.62 ± 0.04
		IGES ($\kappa = 0.5$)	0.81 ± 0.05	0.89 ± 0.06	0.84 ± 0.05	0.68 ± 0.04	0.58 ± 0.05	0.63 ± 0.05
		IGES ($\kappa = 0.9$)	0.78 ± 0.04	0.89 ± 0.04	0.82 ± 0.03	0.66 ± 0.05	0.57 ± 0.04	0.62 ± 0.03
		GES	0.61 ± 0.10	0.94 ± 0.06	0.73 ± 0.07	0.72 ± 0.05	0.63 ± 0.07	0.67 ± 0.04
	120	IGES ($\kappa = 0.001$)	0.81 ± 0.07	0.93 ± 0.04	0.85 ± 0.04	0.59 ± 0.07	0.50 ± 0.09	0.55 ± 0.08
		IGES ($\kappa = 0.1$)	0.81 ± 0.04	0.89 ± 0.05	0.85 ± 0.04	0.58 ± 0.07	0.47 ± 0.06	0.52 ± 0.06
		IGES ($\kappa = 0.5$)	0.82 ± 0.05	0.90 ± 0.05	0.85 ± 0.05	0.60 ± 0.07	0.51 ± 0.07	0.55 ± 0.07
		IGES ($\kappa = 0.9$)	0.77 ± 0.09	0.90 ± 0.04	0.83 ± 0.06	0.60 ± 0.07	0.51 ± 0.05	0.55 ± 0.05
		GES	0.61 ± 0.05	0.94 ± 0.05	0.75 ± 0.05	0.62 ± 0.06	0.49 ± 0.07	0.55 ± 0.04
50	100	IGES ($\kappa = 0.001$)	0.86 ± 0.05	0.97 ± 0.03	0.93 ± 0.03	0.81 ± 0.06	0.80 ± 0.04	0.80 ± 0.04
		IGES ($\kappa = 0.1$)	0.87 ± 0.06	0.94 ± 0.03	0.90 ± 0.04	0.81 ± 0.03	0.79 ± 0.05	0.80 ± 0.03
		IGES ($\kappa = 0.5$)	0.82 ± 0.06	0.91 ± 0.04	0.88 ± 0.05	0.82 ± 0.05	0.80 ± 0.04	0.81 ± 0.04
		IGES ($\kappa = 0.9$)	0.79 ± 0.07	0.86 ± 0.05	0.83 ± 0.06	0.82 ± 0.04	0.80 ± 0.04	0.81 ± 0.04
		GES	0.62 ± 0.05	0.96 ± 0.05	0.79 ± 0.04	0.86 ± 0.05	0.79 ± 0.05	0.82 ± 0.04
	200	IGES ($\kappa = 0.001$)	0.85 ± 0.03	0.95 ± 0.03	0.90 ± 0.02	0.64 ± 0.05	0.55 ± 0.05	0.59 ± 0.05
		IGES ($\kappa = 0.1$)	0.85 ± 0.02	0.93 ± 0.02	0.89 ± 0.02	0.65 ± 0.04	0.55 ± 0.05	0.59 ± 0.05
		IGES ($\kappa = 0.5$)	0.83 ± 0.04	0.91 ± 0.03	0.87 ± 0.03	0.65 ± 0.05	0.56 ± 0.04	0.60 ± 0.04
		IGES ($\kappa = 0.9$)	0.79 ± 0.05	0.89 ± 0.02	0.84 ± 0.03	0.66 ± 0.04	0.56 ± 0.04	0.60 ± 0.03
		GES	0.63 ± 0.04	0.96 ± 0.03	0.76 ± 0.04	0.66 ± 0.03	0.53 ± 0.05	0.58 ± 0.03
	300	IGES ($\kappa = 0.001$)	0.86 ± 0.05	0.95 ± 0.03	0.90 ± 0.04	0.56 ± 0.03	0.46 ± 0.03	0.50 ± 0.03
		IGES ($\kappa = 0.1$)	0.88 ± 0.04	0.92 ± 0.03	0.90 ± 0.03	0.59 ± 0.04	0.49 ± 0.04	0.53 ± 0.04
		IGES ($\kappa = 0.5$)	0.82 ± 0.03	0.90 ± 0.03	0.86 ± 0.03	0.58 ± 0.04	0.48 ± 0.05	0.52 ± 0.04
		IGES ($\kappa = 0.9$)	0.78 ± 0.03	0.87 ± 0.03	0.82 ± 0.03	0.60 ± 0.05	0.50 ± 0.04	0.54 ± 0.04
		GES	0.66 ± 0.05	0.95 ± 0.03	0.77 ± 0.04	0.59 ± 0.04	0.47 ± 0.04	0.52 ± 0.04

Table 56: Arrowhead precision (P) and recall (R) results for $N = 200$ training cases.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGES ($\kappa = 0.001$)	0.36 ± 0.18	0.37 ± 0.37	0.33 ± 0.36	0.39 ± 0.18	0.17 ± 0.18	0.14 ± 0.14
		IGES ($\kappa = 0.1$)	0.40 ± 0.16	0.30 ± 0.35	0.34 ± 0.36	0.41 ± 0.15	0.14 ± 0.14	0.14 ± 0.13
		IGES ($\kappa = 0.5$)	0.40 ± 0.16	0.32 ± 0.34	0.34 ± 0.33	0.41 ± 0.16	0.15 ± 0.13	0.14 ± 0.12
		IGES ($\kappa = 0.9$)	0.39 ± 0.17	0.31 ± 0.33	0.33 ± 0.32	0.39 ± 0.16	0.16 ± 0.14	0.15 ± 0.13
		GES	0.37 ± 0.18	0.37 ± 0.37	0.33 ± 0.36	0.41 ± 0.17	0.17 ± 0.18	0.14 ± 0.14
	40	IGES ($\kappa = 0.001$)	0.24 ± 0.16	0.12 ± 0.14	0.16 ± 0.17	0.20 ± 0.14	0.04 ± 0.04	0.05 ± 0.05
		IGES ($\kappa = 0.1$)	0.30 ± 0.18	0.14 ± 0.14	0.23 ± 0.19	0.25 ± 0.16	0.04 ± 0.04	0.07 ± 0.06
		IGES ($\kappa = 0.5$)	0.31 ± 0.16	0.19 ± 0.16	0.24 ± 0.19	0.26 ± 0.14	0.07 ± 0.05	0.09 ± 0.06
		IGES ($\kappa = 0.9$)	0.28 ± 0.13	0.19 ± 0.14	0.23 ± 0.17	0.24 ± 0.11	0.07 ± 0.05	0.09 ± 0.06
		GES	0.22 ± 0.18	0.12 ± 0.14	0.14 ± 0.17	0.18 ± 0.15	0.04 ± 0.03	0.05 ± 0.05
	60	IGES ($\kappa = 0.001$)	0.42 ± 0.24	0.32 ± 0.24	0.39 ± 0.32	0.35 ± 0.17	0.08 ± 0.04	0.10 ± 0.07
		IGES ($\kappa = 0.1$)	0.45 ± 0.23	0.34 ± 0.22	0.40 ± 0.33	0.38 ± 0.15	0.09 ± 0.04	0.11 ± 0.08
		IGES ($\kappa = 0.5$)	0.42 ± 0.23	0.33 ± 0.18	0.39 ± 0.30	0.37 ± 0.14	0.10 ± 0.04	0.12 ± 0.08
		IGES ($\kappa = 0.9$)	0.42 ± 0.21	0.33 ± 0.17	0.37 ± 0.27	0.36 ± 0.15	0.11 ± 0.04	0.12 ± 0.07
		GES	0.40 ± 0.24	0.27 ± 0.26	0.37 ± 0.33	0.33 ± 0.18	0.07 ± 0.04	0.09 ± 0.07
20	40	IGES ($\kappa = 0.001$)	0.24 ± 0.23	0.40 ± 0.31	0.40 ± 0.30	0.20 ± 0.19	0.12 ± 0.10	0.12 ± 0.11
		IGES ($\kappa = 0.1$)	0.19 ± 0.13	0.52 ± 0.35	0.49 ± 0.32	0.17 ± 0.14	0.12 ± 0.09	0.13 ± 0.08
		IGES ($\kappa = 0.5$)	0.29 ± 0.14	0.45 ± 0.10	0.43 ± 0.09	0.29 ± 0.25	0.17 ± 0.06	0.19 ± 0.08
		IGES ($\kappa = 0.9$)	0.30 ± 0.13	0.38 ± 0.11	0.37 ± 0.09	0.30 ± 0.14	0.16 ± 0.06	0.17 ± 0.07
		GES	0.26 ± 0.22	0.45 ± 0.34	0.38 ± 0.30	0.23 ± 0.20	0.10 ± 0.09	0.11 ± 0.11
	80	IGES ($\kappa = 0.001$)	0.23 ± 0.19	0.45 ± 0.31	0.42 ± 0.25	0.10 ± 0.08	0.05 ± 0.04	0.06 ± 0.05
		IGES ($\kappa = 0.1$)	0.32 ± 0.16	0.44 ± 0.17	0.46 ± 0.18	0.22 ± 0.14	0.09 ± 0.06	0.11 ± 0.07
		IGES ($\kappa = 0.5$)	0.43 ± 0.21	0.63 ± 0.21	0.60 ± 0.18	0.30 ± 0.16	0.14 ± 0.04	0.16 ± 0.06
		IGES ($\kappa = 0.9$)	0.37 ± 0.19	0.44 ± 0.10	0.43 ± 0.10	0.29 ± 0.12	0.11 ± 0.05	0.14 ± 0.06
		GES	0.37 ± 0.26	0.31 ± 0.30	0.33 ± 0.28	0.21 ± 0.15	0.08 ± 0.08	0.09 ± 0.08
	120	IGES ($\kappa = 0.001$)	0.26 ± 0.21	0.37 ± 0.18	0.35 ± 0.15	0.16 ± 0.15	0.06 ± 0.04	0.07 ± 0.05
		IGES ($\kappa = 0.1$)	0.38 ± 0.22	0.45 ± 0.29	0.44 ± 0.25	0.22 ± 0.15	0.08 ± 0.05	0.10 ± 0.06
		IGES ($\kappa = 0.5$)	0.42 ± 0.22	0.52 ± 0.18	0.52 ± 0.16	0.23 ± 0.13	0.08 ± 0.04	0.11 ± 0.05
		IGES ($\kappa = 0.9$)	0.28 ± 0.09	0.48 ± 0.14	0.41 ± 0.11	0.22 ± 0.10	0.08 ± 0.02	0.11 ± 0.04
		GES	0.23 ± 0.16	0.41 ± 0.28	0.37 ± 0.27	0.12 ± 0.11	0.05 ± 0.04	0.06 ± 0.05
50	100	IGES ($\kappa = 0.001$)	0.36 ± 0.21	0.78 ± 0.25	0.72 ± 0.18	0.21 ± 0.19	0.14 ± 0.11	0.15 ± 0.11
		IGES ($\kappa = 0.1$)	0.34 ± 0.21	0.62 ± 0.25	0.58 ± 0.23	0.14 ± 0.09	0.13 ± 0.05	0.14 ± 0.05
		IGES ($\kappa = 0.5$)	0.33 ± 0.11	0.54 ± 0.07	0.51 ± 0.06	0.20 ± 0.12	0.17 ± 0.07	0.18 ± 0.07
		IGES ($\kappa = 0.9$)	0.30 ± 0.09	0.36 ± 0.07	0.34 ± 0.06	0.39 ± 0.13	0.26 ± 0.09	0.28 ± 0.09
		GES	0.41 ± 0.13	0.74 ± 0.26	0.64 ± 0.16	0.18 ± 0.10	0.13 ± 0.09	0.14 ± 0.09
	200	IGES ($\kappa = 0.001$)	0.47 ± 0.25	0.58 ± 0.27	0.55 ± 0.24	0.17 ± 0.13	0.07 ± 0.04	0.08 ± 0.05
		IGES ($\kappa = 0.1$)	0.53 ± 0.12	0.63 ± 0.17	0.61 ± 0.13	0.25 ± 0.09	0.08 ± 0.03	0.11 ± 0.04
		IGES ($\kappa = 0.5$)	0.40 ± 0.14	0.54 ± 0.08	0.49 ± 0.09	0.21 ± 0.08	0.08 ± 0.02	0.10 ± 0.03
		IGES ($\kappa = 0.9$)	0.33 ± 0.06	0.43 ± 0.05	0.39 ± 0.04	0.32 ± 0.05	0.12 ± 0.04	0.15 ± 0.04
		GES	0.43 ± 0.13	0.69 ± 0.22	0.58 ± 0.15	0.17 ± 0.08	0.07 ± 0.03	0.09 ± 0.04
	300	IGES ($\kappa = 0.001$)	0.52 ± 0.23	0.60 ± 0.31	0.59 ± 0.29	0.13 ± 0.09	0.04 ± 0.03	0.05 ± 0.04
		IGES ($\kappa = 0.1$)	0.57 ± 0.14	0.59 ± 0.18	0.66 ± 0.12	0.16 ± 0.12	0.04 ± 0.03	0.06 ± 0.05
		IGES ($\kappa = 0.5$)	0.45 ± 0.07	0.56 ± 0.08	0.51 ± 0.06	0.27 ± 0.10	0.08 ± 0.04	0.11 ± 0.05
		IGES ($\kappa = 0.9$)	0.43 ± 0.08	0.43 ± 0.10	0.43 ± 0.08	0.36 ± 0.11	0.09 ± 0.03	0.13 ± 0.05
		GES	0.48 ± 0.23	0.65 ± 0.28	0.57 ± 0.23	0.13 ± 0.07	0.05 ± 0.03	0.06 ± 0.03

Table 57: Arrowhead precision (P) and recall (R) results for $N = 1000$ training cases.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGES ($\kappa = 0.001$)	0.48 ± 0.27	0.53 ± 0.26	0.50 ± 0.20	0.53 ± 0.24	0.38 ± 0.15	0.42 ± 0.15
		IGES ($\kappa = 0.1$)	0.52 ± 0.21	0.57 ± 0.21	0.55 ± 0.17	0.53 ± 0.20	0.38 ± 0.09	0.40 ± 0.10
		IGES ($\kappa = 0.5$)	0.54 ± 0.20	0.59 ± 0.20	0.57 ± 0.14	0.55 ± 0.19	0.38 ± 0.08	0.41 ± 0.09
		IGES ($\kappa = 0.9$)	0.57 ± 0.18	0.56 ± 0.21	0.57 ± 0.16	0.59 ± 0.19	0.38 ± 0.07	0.42 ± 0.08
		GES	0.36 ± 0.22	0.49 ± 0.24	0.43 ± 0.13	0.42 ± 0.22	0.37 ± 0.15	0.40 ± 0.14
	40	IGES ($\kappa = 0.001$)	0.42 ± 0.20	0.49 ± 0.17	0.45 ± 0.15	0.49 ± 0.19	0.24 ± 0.08	0.28 ± 0.09
		IGES ($\kappa = 0.1$)	0.42 ± 0.20	0.52 ± 0.15	0.47 ± 0.16	0.47 ± 0.20	0.27 ± 0.09	0.29 ± 0.10
		IGES ($\kappa = 0.5$)	0.42 ± 0.16	0.54 ± 0.20	0.48 ± 0.18	0.49 ± 0.13	0.27 ± 0.10	0.30 ± 0.11
		IGES ($\kappa = 0.9$)	0.42 ± 0.15	0.52 ± 0.20	0.47 ± 0.17	0.47 ± 0.11	0.27 ± 0.09	0.30 ± 0.09
		GES	0.38 ± 0.20	0.44 ± 0.21	0.44 ± 0.20	0.46 ± 0.18	0.21 ± 0.10	0.25 ± 0.11
	60	IGES ($\kappa = 0.001$)	0.27 ± 0.14	0.37 ± 0.17	0.34 ± 0.18	0.28 ± 0.17	0.17 ± 0.10	0.18 ± 0.12
		IGES ($\kappa = 0.1$)	0.31 ± 0.17	0.41 ± 0.17	0.39 ± 0.17	0.30 ± 0.18	0.17 ± 0.10	0.19 ± 0.11
		IGES ($\kappa = 0.5$)	0.32 ± 0.15	0.41 ± 0.15	0.39 ± 0.15	0.30 ± 0.18	0.17 ± 0.10	0.19 ± 0.10
		IGES ($\kappa = 0.9$)	0.33 ± 0.16	0.40 ± 0.13	0.39 ± 0.15	0.30 ± 0.17	0.17 ± 0.08	0.19 ± 0.09
		GES	0.23 ± 0.12	0.36 ± 0.18	0.32 ± 0.19	0.29 ± 0.20	0.15 ± 0.10	0.16 ± 0.12
20	40	IGES ($\kappa = 0.001$)	0.47 ± 0.21	0.78 ± 0.22	0.70 ± 0.19	0.58 ± 0.25	0.49 ± 0.13	0.51 ± 0.15
		IGES ($\kappa = 0.1$)	0.66 ± 0.15	0.68 ± 0.15	0.68 ± 0.14	0.59 ± 0.17	0.41 ± 0.12	0.44 ± 0.12
		IGES ($\kappa = 0.5$)	0.61 ± 0.20	0.79 ± 0.10	0.75 ± 0.12	0.61 ± 0.19	0.52 ± 0.12	0.53 ± 0.11
		IGES ($\kappa = 0.9$)	0.47 ± 0.21	0.63 ± 0.13	0.58 ± 0.14	0.55 ± 0.22	0.51 ± 0.10	0.52 ± 0.09
		GES	0.32 ± 0.12	0.71 ± 0.21	0.57 ± 0.15	0.53 ± 0.17	0.47 ± 0.11	0.48 ± 0.10
	80	IGES ($\kappa = 0.001$)	0.48 ± 0.08	0.76 ± 0.19	0.65 ± 0.14	0.56 ± 0.12	0.28 ± 0.09	0.33 ± 0.10
		IGES ($\kappa = 0.1$)	0.53 ± 0.13	0.79 ± 0.12	0.73 ± 0.11	0.53 ± 0.17	0.27 ± 0.07	0.32 ± 0.07
		IGES ($\kappa = 0.5$)	0.53 ± 0.15	0.64 ± 0.11	0.60 ± 0.11	0.45 ± 0.08	0.27 ± 0.05	0.30 ± 0.05
		IGES ($\kappa = 0.9$)	0.50 ± 0.14	0.70 ± 0.11	0.63 ± 0.11	0.56 ± 0.13	0.32 ± 0.08	0.36 ± 0.08
		GES	0.38 ± 0.11	0.71 ± 0.15	0.55 ± 0.11	0.58 ± 0.15	0.28 ± 0.07	0.33 ± 0.08
	120	IGES ($\kappa = 0.001$)	0.53 ± 0.15	0.71 ± 0.17	0.64 ± 0.11	0.61 ± 0.17	0.25 ± 0.05	0.30 ± 0.06
		IGES ($\kappa = 0.1$)	0.43 ± 0.10	0.66 ± 0.13	0.58 ± 0.10	0.45 ± 0.12	0.20 ± 0.05	0.24 ± 0.05
		IGES ($\kappa = 0.5$)	0.53 ± 0.10	0.65 ± 0.15	0.61 ± 0.11	0.52 ± 0.17	0.22 ± 0.06	0.27 ± 0.07
		IGES ($\kappa = 0.9$)	0.56 ± 0.13	0.70 ± 0.08	0.65 ± 0.10	0.57 ± 0.12	0.27 ± 0.07	0.32 ± 0.07
		GES	0.46 ± 0.13	0.72 ± 0.13	0.62 ± 0.09	0.55 ± 0.20	0.23 ± 0.08	0.29 ± 0.10
50	100	IGES ($\kappa = 0.001$)	0.51 ± 0.18	0.84 ± 0.08	0.74 ± 0.10	0.60 ± 0.10	0.50 ± 0.09	0.51 ± 0.08
		IGES ($\kappa = 0.1$)	0.60 ± 0.12	0.84 ± 0.10	0.79 ± 0.07	0.49 ± 0.14	0.43 ± 0.11	0.44 ± 0.10
		IGES ($\kappa = 0.5$)	0.52 ± 0.07	0.74 ± 0.07	0.69 ± 0.05	0.58 ± 0.09	0.51 ± 0.07	0.52 ± 0.06
		IGES ($\kappa = 0.9$)	0.45 ± 0.13	0.62 ± 0.08	0.57 ± 0.07	0.64 ± 0.10	0.49 ± 0.05	0.51 ± 0.04
		GES	0.34 ± 0.06	0.87 ± 0.10	0.65 ± 0.07	0.66 ± 0.10	0.43 ± 0.09	0.46 ± 0.08
	200	IGES ($\kappa = 0.001$)	0.62 ± 0.16	0.81 ± 0.12	0.73 ± 0.11	0.54 ± 0.12	0.24 ± 0.06	0.29 ± 0.07
		IGES ($\kappa = 0.1$)	0.64 ± 0.16	0.77 ± 0.10	0.73 ± 0.11	0.57 ± 0.09	0.28 ± 0.04	0.33 ± 0.05
		IGES ($\kappa = 0.5$)	0.56 ± 0.12	0.69 ± 0.09	0.64 ± 0.09	0.53 ± 0.13	0.25 ± 0.05	0.29 ± 0.06
		IGES ($\kappa = 0.9$)	0.51 ± 0.08	0.66 ± 0.04	0.61 ± 0.04	0.59 ± 0.09	0.28 ± 0.06	0.33 ± 0.07
		GES	0.45 ± 0.11	0.81 ± 0.09	0.65 ± 0.08	0.51 ± 0.13	0.23 ± 0.05	0.28 ± 0.06
	300	IGES ($\kappa = 0.001$)	0.53 ± 0.07	0.73 ± 0.09	0.66 ± 0.07	0.51 ± 0.12	0.18 ± 0.05	0.23 ± 0.06
		IGES ($\kappa = 0.1$)	0.66 ± 0.06	0.77 ± 0.08	0.73 ± 0.07	0.54 ± 0.12	0.21 ± 0.05	0.26 ± 0.06
		IGES ($\kappa = 0.5$)	0.61 ± 0.05	0.78 ± 0.03	0.71 ± 0.04	0.58 ± 0.06	0.22 ± 0.03	0.27 ± 0.04
		IGES ($\kappa = 0.9$)	0.50 ± 0.08	0.64 ± 0.08	0.59 ± 0.08	0.61 ± 0.05	0.22 ± 0.04	0.28 ± 0.04
		GES	0.46 ± 0.09	0.78 ± 0.15	0.62 ± 0.10	0.53 ± 0.09	0.17 ± 0.05	0.23 ± 0.05

Table 58: Arrowhead precision (P) and recall (R) results for $N = 5000$ training cases.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGES ($\kappa = 0.001$)	0.53 ± 0.23	0.72 ± 0.26	0.65 ± 0.21	0.61 ± 0.22	0.75 ± 0.14	0.72 ± 0.14
		IGES ($\kappa = 0.1$)	0.56 ± 0.22	0.70 ± 0.25	0.65 ± 0.21	0.58 ± 0.21	0.71 ± 0.11	0.68 ± 0.13
		IGES ($\kappa = 0.5$)	0.59 ± 0.23	0.69 ± 0.23	0.66 ± 0.22	0.62 ± 0.23	0.70 ± 0.11	0.67 ± 0.13
		IGES ($\kappa = 0.9$)	0.58 ± 0.23	0.65 ± 0.27	0.64 ± 0.23	0.59 ± 0.24	0.66 ± 0.17	0.64 ± 0.18
		GES	0.23 ± 0.18	0.69 ± 0.26	0.49 ± 0.14	0.52 ± 0.30	0.74 ± 0.14	0.72 ± 0.16
	40	IGES ($\kappa = 0.001$)	0.50 ± 0.25	0.56 ± 0.23	0.53 ± 0.22	0.60 ± 0.27	0.44 ± 0.19	0.48 ± 0.19
		IGES ($\kappa = 0.1$)	0.53 ± 0.26	0.56 ± 0.22	0.54 ± 0.21	0.60 ± 0.28	0.41 ± 0.16	0.45 ± 0.16
		IGES ($\kappa = 0.5$)	0.53 ± 0.25	0.57 ± 0.20	0.55 ± 0.19	0.57 ± 0.24	0.39 ± 0.15	0.43 ± 0.15
		IGES ($\kappa = 0.9$)	0.52 ± 0.24	0.57 ± 0.19	0.55 ± 0.19	0.56 ± 0.23	0.40 ± 0.14	0.43 ± 0.14
		GES	0.27 ± 0.16	0.54 ± 0.28	0.42 ± 0.17	0.58 ± 0.23	0.43 ± 0.22	0.49 ± 0.20
	60	IGES ($\kappa = 0.001$)	0.28 ± 0.19	0.43 ± 0.21	0.38 ± 0.18	0.44 ± 0.23	0.30 ± 0.10	0.35 ± 0.11
		IGES ($\kappa = 0.1$)	0.32 ± 0.16	0.44 ± 0.19	0.41 ± 0.18	0.45 ± 0.15	0.30 ± 0.07	0.32 ± 0.08
		IGES ($\kappa = 0.5$)	0.30 ± 0.16	0.45 ± 0.21	0.40 ± 0.19	0.42 ± 0.15	0.29 ± 0.09	0.31 ± 0.07
		IGES ($\kappa = 0.9$)	0.29 ± 0.12	0.47 ± 0.20	0.41 ± 0.17	0.40 ± 0.11	0.30 ± 0.09	0.32 ± 0.08
		GES	0.15 ± 0.08	0.43 ± 0.22	0.33 ± 0.15	0.40 ± 0.20	0.29 ± 0.06	0.33 ± 0.06
20	40	IGES ($\kappa = 0.001$)	0.60 ± 0.22	0.76 ± 0.17	0.71 ± 0.17	0.70 ± 0.18	0.70 ± 0.12	0.70 ± 0.12
		IGES ($\kappa = 0.1$)	0.59 ± 0.25	0.75 ± 0.17	0.70 ± 0.16	0.72 ± 0.17	0.66 ± 0.10	0.68 ± 0.09
		IGES ($\kappa = 0.5$)	0.51 ± 0.18	0.69 ± 0.12	0.66 ± 0.11	0.62 ± 0.12	0.66 ± 0.07	0.66 ± 0.06
		IGES ($\kappa = 0.9$)	0.58 ± 0.21	0.64 ± 0.15	0.63 ± 0.16	0.63 ± 0.21	0.59 ± 0.09	0.61 ± 0.11
		GES	0.29 ± 0.09	0.68 ± 0.17	0.52 ± 0.08	0.64 ± 0.18	0.57 ± 0.11	0.60 ± 0.11
	80	IGES ($\kappa = 0.001$)	0.51 ± 0.18	0.68 ± 0.15	0.62 ± 0.14	0.65 ± 0.20	0.47 ± 0.08	0.50 ± 0.09
		IGES ($\kappa = 0.1$)	0.53 ± 0.14	0.75 ± 0.13	0.68 ± 0.11	0.68 ± 0.12	0.47 ± 0.06	0.50 ± 0.06
		IGES ($\kappa = 0.5$)	0.47 ± 0.14	0.77 ± 0.15	0.66 ± 0.13	0.74 ± 0.07	0.50 ± 0.09	0.53 ± 0.08
		IGES ($\kappa = 0.9$)	0.43 ± 0.16	0.72 ± 0.09	0.63 ± 0.09	0.62 ± 0.16	0.47 ± 0.05	0.50 ± 0.07
		GES	0.26 ± 0.06	0.73 ± 0.09	0.52 ± 0.05	0.74 ± 0.15	0.48 ± 0.06	0.52 ± 0.07
	120	IGES ($\kappa = 0.001$)	0.47 ± 0.18	0.75 ± 0.13	0.64 ± 0.13	0.63 ± 0.17	0.38 ± 0.08	0.42 ± 0.08
		IGES ($\kappa = 0.1$)	0.52 ± 0.15	0.66 ± 0.13	0.61 ± 0.10	0.58 ± 0.09	0.35 ± 0.04	0.39 ± 0.05
		IGES ($\kappa = 0.5$)	0.58 ± 0.12	0.72 ± 0.11	0.66 ± 0.09	0.69 ± 0.11	0.39 ± 0.09	0.44 ± 0.09
		IGES ($\kappa = 0.9$)	0.51 ± 0.17	0.72 ± 0.14	0.65 ± 0.14	0.69 ± 0.11	0.41 ± 0.07	0.45 ± 0.07
		GES	0.32 ± 0.12	0.71 ± 0.17	0.54 ± 0.15	0.72 ± 0.15	0.37 ± 0.07	0.42 ± 0.08
50	100	IGES ($\kappa = 0.001$)	0.57 ± 0.11	0.84 ± 0.07	0.78 ± 0.06	0.74 ± 0.07	0.71 ± 0.08	0.71 ± 0.07
		IGES ($\kappa = 0.1$)	0.63 ± 0.16	0.81 ± 0.08	0.76 ± 0.08	0.77 ± 0.08	0.69 ± 0.07	0.71 ± 0.07
		IGES ($\kappa = 0.5$)	0.57 ± 0.19	0.77 ± 0.10	0.72 ± 0.12	0.72 ± 0.18	0.70 ± 0.09	0.70 ± 0.10
		IGES ($\kappa = 0.9$)	0.53 ± 0.15	0.74 ± 0.14	0.69 ± 0.13	0.75 ± 0.13	0.72 ± 0.08	0.72 ± 0.08
		GES	0.28 ± 0.07	0.81 ± 0.16	0.59 ± 0.10	0.79 ± 0.14	0.69 ± 0.09	0.70 ± 0.10
	200	IGES ($\kappa = 0.001$)	0.57 ± 0.08	0.83 ± 0.05	0.75 ± 0.06	0.72 ± 0.11	0.45 ± 0.06	0.49 ± 0.06
		IGES ($\kappa = 0.1$)	0.59 ± 0.10	0.80 ± 0.06	0.74 ± 0.06	0.79 ± 0.06	0.47 ± 0.05	0.51 ± 0.05
		IGES ($\kappa = 0.5$)	0.60 ± 0.07	0.80 ± 0.06	0.74 ± 0.05	0.77 ± 0.09	0.47 ± 0.04	0.51 ± 0.04
		IGES ($\kappa = 0.9$)	0.61 ± 0.08	0.78 ± 0.06	0.73 ± 0.06	0.79 ± 0.06	0.48 ± 0.04	0.53 ± 0.04
		GES	0.35 ± 0.07	0.86 ± 0.10	0.61 ± 0.07	0.80 ± 0.06	0.43 ± 0.04	0.49 ± 0.04
	300	IGES ($\kappa = 0.001$)	0.59 ± 0.11	0.78 ± 0.10	0.72 ± 0.11	0.76 ± 0.06	0.35 ± 0.02	0.41 ± 0.02
		IGES ($\kappa = 0.1$)	0.67 ± 0.10	0.76 ± 0.09	0.73 ± 0.09	0.76 ± 0.08	0.39 ± 0.05	0.44 ± 0.05
		IGES ($\kappa = 0.5$)	0.58 ± 0.09	0.76 ± 0.07	0.71 ± 0.07	0.76 ± 0.07	0.38 ± 0.05	0.44 ± 0.06
		IGES ($\kappa = 0.9$)	0.54 ± 0.10	0.73 ± 0.06	0.67 ± 0.06	0.77 ± 0.05	0.40 ± 0.04	0.45 ± 0.04
		GES	0.35 ± 0.05	0.78 ± 0.08	0.58 ± 0.04	0.77 ± 0.04	0.35 ± 0.04	0.42 ± 0.05

Table 59: Adjacency and strict SHD (A-SHD and S-SHD) for $N = 200$ training cases.

			Added			Deleted			Reoriented				
# Variables	# Edges	Method	IS	Other	Overall	IS	Other	Overall	IS	Other	Overall	A-SHD	S-SHD
10	20	IGES ($\kappa = 0.001$)	0.68	0.01	0.69	3.08	5.95	9.03	0.95	1.25	2.20	9.72	11.92
		IGES ($\kappa = 0.1$)	0.76	0.15	0.91	2.87	5.56	8.43	0.95	1.44	2.40	9.34	11.73
		IGES ($\kappa = 0.5$)	0.77	0.20	0.97	2.91	5.45	8.36	1.00	1.39	2.38	9.33	11.71
		IGES ($\kappa = 0.9$)	0.93	0.37	1.29	3.03	5.31	8.34	0.87	1.41	2.27	9.63	11.91
		GES	0.91	0.13	1.04	2.03	4.79	6.83	1.53	1.86	3.39	7.86	11.25
	40	IGES ($\kappa = 0.001$)	0.74	0.28	1.02	7.08	10.78	17.86	1.59	2.32	3.92	18.89	22.81
		IGES ($\kappa = 0.1$)	0.69	0.31	0.99	6.78	10.60	17.38	1.62	2.31	3.93	18.37	22.30
		IGES ($\kappa = 0.5$)	0.95	0.30	1.25	6.57	10.22	16.79	1.68	2.37	4.05	18.04	22.09
		IGES ($\kappa = 0.9$)	1.15	0.39	1.54	6.44	10.23	16.66	1.78	2.42	4.20	18.21	22.41
		GES	1.16	0.43	1.59	6.12	9.85	15.97	2.32	2.95	5.27	17.55	22.82
	60	IGES ($\kappa = 0.001$)	0.59	0.03	0.62	6.80	12.20	19.00	1.66	2.08	3.74	19.62	23.36
		IGES ($\kappa = 0.1$)	0.52	0.03	0.55	6.53	12.14	18.66	1.75	1.95	3.70	19.22	22.92
		IGES ($\kappa = 0.5$)	0.67	0.04	0.71	6.33	12.00	18.33	1.77	2.02	3.79	19.04	22.84
		IGES ($\kappa = 0.9$)	0.83	0.05	0.88	6.27	11.74	18.00	1.84	2.13	3.97	18.88	22.85
		GES	0.80	0.03	0.83	6.02	11.45	17.47	2.22	2.58	4.79	18.30	23.10
20	40	IGES ($\kappa = 0.001$)	0.82	0.13	0.95	5.80	13.07	18.87	1.73	2.14	3.86	19.82	23.68
		IGES ($\kappa = 0.1$)	1.04	0.31	1.36	5.82	11.47	17.29	2.13	2.64	4.77	18.65	23.42
		IGES ($\kappa = 0.5$)	1.62	1.23	2.84	4.63	11.44	16.07	2.41	3.44	5.85	18.91	24.76
		IGES ($\kappa = 0.9$)	2.36	2.09	4.45	5.34	10.77	16.11	2.30	3.24	5.54	20.56	26.10
		GES	1.31	0.48	1.79	4.83	10.35	15.18	2.97	3.49	6.46	16.97	23.43
	80	IGES ($\kappa = 0.001$)	1.02	0.27	1.28	17.38	21.73	39.12	2.64	2.25	4.88	40.40	45.28
		IGES ($\kappa = 0.1$)	1.32	0.30	1.62	15.13	18.44	33.57	3.78	2.94	6.71	35.20	41.91
		IGES ($\kappa = 0.5$)	1.93	0.64	2.56	14.87	17.15	32.02	2.65	2.39	5.04	34.58	39.61
		IGES ($\kappa = 0.9$)	3.30	0.86	4.16	14.48	17.22	31.70	3.70	2.82	6.52	35.86	42.39
		GES	1.46	0.18	1.64	13.80	17.01	30.81	4.52	3.74	8.26	32.45	40.72
	120	IGES ($\kappa = 0.001$)	1.22	0.16	1.38	19.93	24.91	44.83	3.38	2.37	5.74	46.21	51.95
		IGES ($\kappa = 0.1$)	1.12	0.11	1.23	17.88	25.95	43.83	4.04	2.49	6.53	45.07	51.60
		IGES ($\kappa = 0.5$)	1.45	0.37	1.81	19.08	24.42	43.50	3.96	2.67	6.62	45.31	51.94
		IGES ($\kappa = 0.9$)	3.42	0.86	4.28	18.78	23.52	42.30	4.08	2.58	6.67	46.58	53.25
		GES	1.30	0.16	1.45	18.37	23.64	42.01	4.86	2.99	7.85	43.46	51.31
50	100	IGES ($\kappa = 0.001$)	1.35	0.09	1.44	15.91	31.04	46.96	4.11	4.77	8.88	48.40	57.28
		IGES ($\kappa = 0.1$)	2.17	1.38	3.56	14.97	27.35	42.32	4.76	6.85	11.61	45.88	57.49
		IGES ($\kappa = 0.5$)	3.89	3.96	7.85	13.99	27.80	41.79	5.31	7.03	12.34	49.64	61.98
		IGES ($\kappa = 0.9$)	10.07	10.50	20.56	13.39	22.33	35.73	5.50	7.61	13.12	56.29	69.41
		GES	2.26	0.49	2.75	14.21	24.51	38.72	4.64	7.71	12.34	41.47	53.81
	200	IGES ($\kappa = 0.001$)	1.85	0.23	2.08	38.67	58.49	97.16	6.12	5.42	11.54	99.25	110.79
		IGES ($\kappa = 0.1$)	2.82	1.05	3.86	35.52	60.02	95.53	7.34	6.08	13.43	99.40	112.82
		IGES ($\kappa = 0.5$)	5.22	2.75	7.97	35.69	58.79	94.48	7.42	6.18	13.60	102.45	116.05
		IGES ($\kappa = 0.9$)	11.84	5.62	17.47	35.53	54.00	89.53	7.58	6.64	14.22	107.00	121.22
		GES	2.77	0.36	3.14	35.68	54.79	90.47	9.68	7.85	17.53	93.61	111.14
	300	IGES ($\kappa = 0.001$)	1.07	0.17	1.24	49.28	73.60	122.87	5.39	4.24	9.63	124.11	133.75
		IGES ($\kappa = 0.1$)	2.04	0.47	2.51	46.35	68.70	115.05	6.91	5.98	12.89	117.56	130.45
		IGES ($\kappa = 0.5$)	6.15	2.66	8.81	44.19	66.35	110.54	8.08	5.80	13.89	119.36	133.25
		IGES ($\kappa = 0.9$)	10.63	4.20	14.83	43.34	61.65	104.99	8.27	6.86	15.13	119.81	134.94
		GES	2.28	0.24	2.52	44.77	61.12	105.89	10.78	7.93	18.71	108.41	127.12

Table 60: Adjacency and strict SHD (A-SHD and S-SHD) for $N = 1000$ training cases.

			Added			Deleted			Reoriented				
# Variables	# Edges	Method	IS	Other	Overall	IS	Other	Overall	IS	Other	Overall	A-SHD	S-SHD
10	20	IGES ($\kappa = 0.001$)	0.97	0.50	1.47	1.20	3.61	4.81	1.31	2.87	4.18	6.27	10.45
		IGES ($\kappa = 0.1$)	0.72	0.54	1.27	1.41	3.84	5.25	1.08	2.52	3.59	6.52	10.11
		IGES ($\kappa = 0.5$)	0.78	0.61	1.40	1.40	3.93	5.32	1.01	2.37	3.37	6.72	10.09
		IGES ($\kappa = 0.9$)	0.99	0.73	1.72	1.42	3.92	5.33	0.89	2.41	3.30	7.06	10.36
		GES	1.73	0.53	2.26	0.87	3.20	4.06	1.80	3.18	4.98	6.32	11.30
	40	IGES ($\kappa = 0.001$)	1.46	0.13	1.59	4.76	7.91	12.67	2.31	2.33	4.63	14.26	18.89
		IGES ($\kappa = 0.1$)	1.38	0.12	1.50	4.85	7.89	12.73	1.99	2.08	4.07	14.23	18.30
		IGES ($\kappa = 0.5$)	1.36	0.17	1.54	4.65	7.54	12.18	2.05	2.33	4.38	13.72	18.10
		IGES ($\kappa = 0.9$)	1.41	0.21	1.62	4.69	7.47	12.16	2.11	2.40	4.51	13.79	18.30
		GES	2.08	0.09	2.17	4.19	7.24	11.42	2.64	2.84	5.49	13.60	19.09
	60	IGES ($\kappa = 0.001$)	1.07	0.03	1.10	6.08	10.38	16.46	1.84	3.33	5.17	17.56	22.73
		IGES ($\kappa = 0.1$)	1.31	0.08	1.38	5.88	10.57	16.45	1.79	3.10	4.89	17.84	22.73
		IGES ($\kappa = 0.5$)	1.32	0.10	1.41	5.73	10.63	16.36	1.86	3.05	4.91	17.77	22.69
		IGES ($\kappa = 0.9$)	1.36	0.11	1.47	5.80	10.61	16.41	1.84	3.17	5.01	17.88	22.89
		GES	2.11	0.09	2.21	5.38	9.53	14.92	2.33	3.81	6.14	17.12	23.26
20	40	IGES ($\kappa = 0.001$)	1.84	0.39	2.23	3.04	6.84	9.88	2.08	2.98	5.06	12.10	17.16
		IGES ($\kappa = 0.1$)	1.06	0.94	2.00	3.34	8.06	11.40	1.90	2.73	4.62	13.40	18.03
		IGES ($\kappa = 0.5$)	1.43	0.87	2.30	2.64	6.54	9.17	1.54	1.96	3.50	11.48	14.98
		IGES ($\kappa = 0.9$)	2.71	1.76	4.48	2.81	5.88	8.68	2.31	2.96	5.28	13.16	18.44
		GES	3.31	0.21	3.52	2.62	5.94	8.56	2.78	3.46	6.24	12.08	18.32
	80	IGES ($\kappa = 0.001$)	3.49	0.58	4.07	11.09	15.42	26.51	3.55	2.48	6.03	30.58	36.61
		IGES ($\kappa = 0.1$)	1.90	0.43	2.33	12.32	14.62	26.95	3.55	1.84	5.39	29.27	34.67
		IGES ($\kappa = 0.5$)	2.67	1.09	3.76	11.23	14.72	25.95	3.79	3.16	6.95	29.71	36.66
		IGES ($\kappa = 0.9$)	3.34	1.33	4.67	10.43	15.47	25.90	3.09	2.31	5.39	30.57	35.97
		GES	4.52	0.25	4.77	9.86	15.49	25.34	4.22	3.30	7.52	30.11	37.64
	120	IGES ($\kappa = 0.001$)	2.60	0.48	3.08	12.50	19.24	31.75	3.62	3.25	6.87	34.82	41.69
		IGES ($\kappa = 0.1$)	2.19	0.76	2.95	14.64	20.45	35.09	5.26	3.19	8.45	38.03	46.48
		IGES ($\kappa = 0.5$)	2.57	1.14	3.71	14.67	19.80	34.47	3.94	3.13	7.07	38.18	45.25
		IGES ($\kappa = 0.9$)	3.48	1.23	4.71	13.71	17.99	31.71	3.59	2.67	6.25	36.42	42.67
		GES	4.26	0.41	4.67	13.72	17.77	31.49	4.51	3.23	7.73	36.16	43.89
50	100	IGES ($\kappa = 0.001$)	3.04	0.54	3.58	9.03	15.78	24.81	4.65	4.83	9.47	28.40	37.87
		IGES ($\kappa = 0.1$)	2.54	0.97	3.51	8.31	15.65	23.96	3.60	5.12	8.72	27.47	36.19
		IGES ($\kappa = 0.5$)	5.05	3.79	8.85	7.78	15.64	23.43	3.68	5.74	9.41	32.27	41.69
		IGES ($\kappa = 0.9$)	6.95	7.52	14.47	7.41	15.86	23.27	5.15	6.21	11.36	37.74	49.10
		GES	6.97	0.35	7.33	6.64	16.98	23.62	7.22	5.67	12.90	30.95	43.85
	200	IGES ($\kappa = 0.001$)	4.49	0.81	5.30	27.14	48.38	75.52	6.21	5.79	11.99	80.82	92.82
		IGES ($\kappa = 0.1$)	3.74	1.88	5.62	25.55	43.66	69.21	7.46	6.72	14.18	74.83	89.01
		IGES ($\kappa = 0.5$)	7.23	3.56	10.79	26.43	45.57	72.00	6.75	7.51	14.26	82.79	97.05
		IGES ($\kappa = 0.9$)	10.01	4.78	14.79	24.21	42.13	66.34	6.94	6.64	13.58	81.13	94.71
		GES	8.11	0.71	8.83	23.80	43.70	67.50	8.41	7.56	15.97	76.33	92.30
	300	IGES ($\kappa = 0.001$)	4.56	0.97	5.54	36.22	58.93	95.16	9.62	6.43	16.06	100.69	116.75
		IGES ($\kappa = 0.1$)	3.76	1.57	5.33	32.50	59.91	92.42	7.22	6.44	13.65	97.74	111.40
		IGES ($\kappa = 0.5$)	6.16	2.27	8.43	34.26	51.96	86.22	6.64	4.99	11.63	94.64	106.27
		IGES ($\kappa = 0.9$)	11.25	4.47	15.73	33.91	53.78	87.69	7.25	7.41	14.66	103.41	118.08
		GES	8.15	0.95	9.11	35.19	56.03	91.21	9.11	8.00	17.11	100.32	117.43

Table 61: Adjacency and strict SHD (A-SHD and S-SHD) for $N = 5000$ training cases.

			Added			Deleted			Reoriented				
# Variables	# Edges	Method	IS	Other	Overall	IS	Other	Overall	IS	Other	Overall	A-SHD	S-SHD
10	20	IGES ($\kappa = 0.001$)	1.09	0.64	1.73	0.74	1.37	2.11	1.29	1.94	3.23	3.84	7.07
		IGES ($\kappa = 0.1$)	0.88	0.67	1.55	0.81	1.63	2.44	1.18	1.92	3.09	3.99	7.08
		IGES ($\kappa = 0.5$)	0.80	0.76	1.56	0.93	1.81	2.74	1.08	1.79	2.87	4.30	7.17
		IGES ($\kappa = 0.9$)	0.81	0.82	1.64	0.99	1.92	2.91	1.09	1.79	2.89	4.54	7.43
		GES	3.82	0.64	4.46	0.44	0.99	1.43	1.93	2.12	4.04	5.89	9.93
	40	IGES ($\kappa = 0.001$)	2.45	0.28	2.73	2.75	5.26	8.01	2.66	3.42	6.09	10.74	16.82
		IGES ($\kappa = 0.1$)	1.96	0.32	2.27	3.02	5.83	8.85	2.50	3.12	5.63	11.12	16.75
		IGES ($\kappa = 0.5$)	1.74	0.33	2.06	3.30	6.40	9.69	2.33	2.79	5.13	11.75	16.88
		IGES ($\kappa = 0.9$)	1.82	0.36	2.18	3.35	6.48	9.84	2.32	2.75	5.08	12.01	17.09
		GES	5.41	0.40	5.80	1.90	3.86	5.76	3.68	4.09	7.77	11.56	19.33
	60	IGES ($\kappa = 0.001$)	3.04	0.20	3.24	4.10	7.24	11.34	2.86	5.47	8.33	14.58	22.91
		IGES ($\kappa = 0.1$)	2.66	0.26	2.92	3.83	7.59	11.42	3.05	5.60	8.65	14.34	22.99
		IGES ($\kappa = 0.5$)	2.64	0.30	2.94	4.34	7.79	12.12	2.80	5.04	7.84	15.06	22.91
		IGES ($\kappa = 0.9$)	2.56	0.34	2.90	4.65	7.35	12.00	3.22	4.07	7.29	14.90	22.19
		GES	5.72	0.18	5.90	3.70	6.08	9.78	4.12	5.22	9.35	15.68	25.03
20	40	IGES ($\kappa = 0.001$)	2.13	1.02	3.15	1.22	3.53	4.75	2.16	2.82	4.98	7.90	12.88
		IGES ($\kappa = 0.1$)	1.82	1.10	2.92	1.87	3.74	5.61	1.92	2.69	4.61	8.53	13.14
		IGES ($\kappa = 0.5$)	2.20	2.01	4.20	1.88	3.68	5.56	2.23	2.55	4.79	9.77	14.55
		IGES ($\kappa = 0.9$)	2.23	2.29	4.51	2.47	4.20	6.67	1.61	3.40	5.01	11.18	16.19
		GES	5.59	1.17	6.76	1.39	3.24	4.63	3.12	3.77	6.89	11.40	18.29
	80	IGES ($\kappa = 0.001$)	4.72	2.02	6.74	7.67	11.27	18.94	3.70	3.22	6.92	25.68	32.60
		IGES ($\kappa = 0.1$)	3.10	1.30	4.40	8.41	10.08	18.49	4.15	2.78	6.93	22.89	29.82
		IGES ($\kappa = 0.5$)	4.12	1.68	5.79	7.90	10.46	18.36	4.17	2.04	6.21	24.15	30.36
		IGES ($\kappa = 0.9$)	4.70	1.89	6.59	8.25	10.75	18.99	3.91	2.59	6.50	25.58	32.09
		GES	10.80	1.16	11.96	6.77	9.58	16.35	5.67	3.19	8.86	28.31	37.17
	120	IGES ($\kappa = 0.001$)	3.80	1.07	4.87	10.92	14.70	25.62	4.74	2.91	7.65	30.49	38.13
		IGES ($\kappa = 0.1$)	3.51	1.61	5.13	10.68	15.76	26.44	4.42	3.68	8.10	31.56	39.66
		IGES ($\kappa = 0.5$)	3.62	1.62	5.24	9.96	15.09	25.05	3.99	3.29	7.28	30.29	37.57
		IGES ($\kappa = 0.9$)	4.79	2.04	6.82	10.19	16.55	26.74	3.59	3.61	7.21	33.56	40.77
		GES	9.79	1.00	10.79	9.79	16.80	26.60	4.97	4.06	9.03	37.39	46.42
50	100	IGES ($\kappa = 0.001$)	3.41	1.23	4.64	4.90	8.77	13.67	3.81	4.57	8.38	18.32	26.70
		IGES ($\kappa = 0.1$)	3.66	2.27	5.93	5.24	8.86	14.10	3.69	4.53	8.23	20.03	28.26
		IGES ($\kappa = 0.5$)	4.81	3.42	8.24	4.60	9.10	13.70	4.06	5.11	9.16	21.94	31.10
		IGES ($\kappa = 0.9$)	6.23	5.25	11.48	4.60	8.80	13.40	4.63	4.18	8.81	24.89	33.70
		GES	14.23	1.26	15.49	3.68	8.96	12.65	7.78	5.13	12.91	28.13	41.05
	200	IGES ($\kappa = 0.001$)	5.98	2.00	7.98	19.21	31.06	50.28	7.28	4.69	11.96	58.26	70.22
		IGES ($\kappa = 0.1$)	6.11	3.03	9.14	18.56	32.86	51.43	6.50	5.79	12.29	60.57	72.86
		IGES ($\kappa = 0.5$)	7.14	4.05	11.19	17.87	31.79	49.67	5.54	5.22	10.75	60.85	71.61
		IGES ($\kappa = 0.9$)	9.59	4.88	14.47	17.88	31.15	49.02	5.14	4.57	9.71	63.50	73.21
		GES	20.90	1.65	22.55	17.83	33.72	51.55	8.64	4.02	12.66	74.10	86.76
	300	IGES ($\kappa = 0.001$)	6.39	1.93	8.31	28.75	43.33	72.08	8.18	7.23	15.41	80.39	95.80
		IGES ($\kappa = 0.1$)	5.27	3.28	8.55	25.02	40.92	65.95	7.06	6.96	14.02	74.50	88.52
		IGES ($\kappa = 0.5$)	7.84	4.64	12.48	25.25	44.91	70.15	7.41	6.43	13.84	82.64	96.47
		IGES ($\kappa = 0.9$)	10.79	6.04	16.83	25.56	40.79	66.35	7.87	6.26	14.13	83.18	97.31
		GES	20.11	1.97	22.08	25.53	43.04	68.57	10.72	7.34	18.06	90.65	108.71

Appendix B Additional Results from Chapter 5

In this appendix, I report the average results for the full experiments that are done in simulations of Chapter 5. Omitted rows in the tables represent the settings that failed to return a result in under 72 hours.

Table 62: Adjacency precision (P) and recall (R) results for $N = 200$ training cases.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGFCI ($\kappa = 0.001$)	0.93 ± 0.17	1.00 ± 0.00	0.97 ± 0.05	0.21 ± 0.10	0.25 ± 0.11	0.23 ± 0.09
		IGFCI ($\kappa = 0.1$)	0.90 ± 0.20	1.00 ± 0.01	0.95 ± 0.08	0.30 ± 0.16	0.30 ± 0.12	0.30 ± 0.12
		IGFCI ($\kappa = 0.5$)	0.82 ± 0.27	0.99 ± 0.01	0.90 ± 0.14	0.31 ± 0.15	0.33 ± 0.11	0.32 ± 0.10
		IGFCI ($\kappa = 0.9$)	0.85 ± 0.20	0.91 ± 0.14	0.90 ± 0.09	0.34 ± 0.15	0.36 ± 0.12	0.35 ± 0.11
		GFCI	0.88 ± 0.20	1.00 ± 0.01	0.95 ± 0.07	0.40 ± 0.14	0.43 ± 0.14	0.42 ± 0.12
	40	IGFCI ($\kappa = 0.001$)	0.99 ± 0.02	1.00 ± 0.00	0.99 ± 0.02	0.15 ± 0.11	0.15 ± 0.09	0.14 ± 0.08
		IGFCI ($\kappa = 0.1$)	0.84 ± 0.25	1.00 ± 0.00	0.90 ± 0.16	0.22 ± 0.12	0.19 ± 0.09	0.19 ± 0.09
		IGFCI ($\kappa = 0.5$)	0.83 ± 0.18	1.00 ± 0.00	0.89 ± 0.15	0.26 ± 0.11	0.20 ± 0.08	0.22 ± 0.08
		IGFCI ($\kappa = 0.9$)	0.83 ± 0.15	1.00 ± 0.00	0.89 ± 0.12	0.29 ± 0.11	0.21 ± 0.08	0.23 ± 0.09
		GFCI	0.90 ± 0.13	1.00 ± 0.00	0.93 ± 0.11	0.30 ± 0.12	0.25 ± 0.11	0.27 ± 0.10
	60	IGFCI ($\kappa = 0.001$)	0.95 ± 0.06	1.00 ± 0.00	0.97 ± 0.05	0.20 ± 0.13	0.11 ± 0.07	0.16 ± 0.08
		IGFCI ($\kappa = 0.1$)	0.95 ± 0.07	1.00 ± 0.00	0.97 ± 0.05	0.25 ± 0.12	0.16 ± 0.08	0.21 ± 0.06
		IGFCI ($\kappa = 0.5$)	0.92 ± 0.09	1.00 ± 0.00	0.95 ± 0.05	0.25 ± 0.12	0.18 ± 0.08	0.23 ± 0.06
		IGFCI ($\kappa = 0.9$)	0.92 ± 0.08	1.00 ± 0.00	0.95 ± 0.04	0.27 ± 0.13	0.20 ± 0.09	0.24 ± 0.07
		GFCI	0.90 ± 0.10	1.00 ± 0.00	0.94 ± 0.06	0.34 ± 0.14	0.23 ± 0.12	0.30 ± 0.06
20	40	IGFCI ($\kappa = 0.001$)	0.82 ± 0.23	1.00 ± 0.00	0.93 ± 0.06	0.22 ± 0.10	0.20 ± 0.09	0.20 ± 0.08
		IGFCI ($\kappa = 0.1$)	0.86 ± 0.11	1.00 ± 0.01	0.94 ± 0.04	0.27 ± 0.10	0.24 ± 0.08	0.24 ± 0.07
		IGFCI ($\kappa = 0.5$)	0.80 ± 0.08	0.92 ± 0.06	0.86 ± 0.05	0.29 ± 0.09	0.27 ± 0.09	0.26 ± 0.07
		IGFCI ($\kappa = 0.9$)	0.78 ± 0.06	0.90 ± 0.06	0.84 ± 0.05	0.30 ± 0.09	0.27 ± 0.08	0.27 ± 0.07
		GFCI	0.80 ± 0.12	1.00 ± 0.00	0.92 ± 0.04	0.33 ± 0.13	0.30 ± 0.11	0.30 ± 0.09
	80	IGFCI ($\kappa = 0.001$)	0.94 ± 0.07	0.99 ± 0.02	0.96 ± 0.05	0.14 ± 0.06	0.08 ± 0.04	0.11 ± 0.04
		IGFCI ($\kappa = 0.1$)	0.92 ± 0.09	0.99 ± 0.02	0.94 ± 0.07	0.18 ± 0.07	0.12 ± 0.03	0.15 ± 0.05
		IGFCI ($\kappa = 0.5$)	0.88 ± 0.08	0.98 ± 0.02	0.92 ± 0.06	0.21 ± 0.06	0.14 ± 0.04	0.17 ± 0.04
		IGFCI ($\kappa = 0.9$)	0.87 ± 0.08	0.96 ± 0.05	0.90 ± 0.06	0.22 ± 0.06	0.14 ± 0.04	0.17 ± 0.03
		GFCI	0.90 ± 0.07	0.99 ± 0.02	0.93 ± 0.04	0.23 ± 0.06	0.15 ± 0.04	0.19 ± 0.04
	120	IGFCI ($\kappa = 0.001$)	0.91 ± 0.10	0.98 ± 0.07	0.94 ± 0.07	0.13 ± 0.06	0.07 ± 0.04	0.10 ± 0.05
		IGFCI ($\kappa = 0.1$)	0.90 ± 0.06	0.97 ± 0.08	0.93 ± 0.06	0.16 ± 0.06	0.09 ± 0.04	0.12 ± 0.05
		IGFCI ($\kappa = 0.5$)	0.88 ± 0.07	0.95 ± 0.10	0.90 ± 0.07	0.18 ± 0.06	0.10 ± 0.04	0.14 ± 0.05
		IGFCI ($\kappa = 0.9$)	0.85 ± 0.06	0.93 ± 0.09	0.87 ± 0.06	0.18 ± 0.07	0.10 ± 0.03	0.14 ± 0.05
		GFCI	0.89 ± 0.07	0.98 ± 0.07	0.92 ± 0.06	0.21 ± 0.07	0.13 ± 0.04	0.17 ± 0.05
50	100	IGFCI ($\kappa = 0.001$)	0.89 ± 0.09	1.00 ± 0.00	0.95 ± 0.03	0.19 ± 0.08	0.17 ± 0.05	0.18 ± 0.05
		IGFCI ($\kappa = 0.1$)	0.87 ± 0.05	0.97 ± 0.05	0.93 ± 0.03	0.24 ± 0.08	0.22 ± 0.05	0.22 ± 0.05
		IGFCI ($\kappa = 0.5$)	0.76 ± 0.05	0.85 ± 0.08	0.81 ± 0.06	0.29 ± 0.08	0.26 ± 0.05	0.26 ± 0.06
		IGFCI ($\kappa = 0.9$)	0.66 ± 0.08	0.77 ± 0.10	0.72 ± 0.08	0.28 ± 0.08	0.26 ± 0.05	0.26 ± 0.05
		GFCI	0.86 ± 0.06	0.99 ± 0.03	0.94 ± 0.03	0.29 ± 0.09	0.26 ± 0.06	0.27 ± 0.07

Table 63: Adjacency precision (P) and recall (R) results for $N = 1000$ training cases.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGFCI ($\kappa = 0.001$)	0.90 ± 0.12	0.99 ± 0.04	0.95 ± 0.07	0.47 ± 0.16	0.44 ± 0.16	0.45 ± 0.13
		IGFCI ($\kappa = 0.1$)	0.91 ± 0.11	0.99 ± 0.02	0.96 ± 0.05	0.47 ± 0.15	0.43 ± 0.14	0.44 ± 0.12
		IGFCI ($\kappa = 0.5$)	0.91 ± 0.11	0.97 ± 0.07	0.94 ± 0.06	0.48 ± 0.14	0.46 ± 0.12	0.46 ± 0.10
		IGFCI ($\kappa = 0.9$)	0.90 ± 0.12	0.96 ± 0.10	0.93 ± 0.08	0.48 ± 0.14	0.46 ± 0.10	0.46 ± 0.08
		GFCI	0.82 ± 0.14	0.97 ± 0.07	0.91 ± 0.09	0.59 ± 0.21	0.54 ± 0.18	0.55 ± 0.15
	40	IGFCI ($\kappa = 0.001$)	0.96 ± 0.04	1.00 ± 0.00	0.97 ± 0.03	0.34 ± 0.07	0.25 ± 0.09	0.29 ± 0.06
		IGFCI ($\kappa = 0.1$)	0.95 ± 0.05	1.00 ± 0.00	0.97 ± 0.03	0.38 ± 0.07	0.26 ± 0.07	0.31 ± 0.06
		IGFCI ($\kappa = 0.5$)	0.94 ± 0.06	1.00 ± 0.00	0.96 ± 0.03	0.37 ± 0.07	0.25 ± 0.06	0.30 ± 0.06
		IGFCI ($\kappa = 0.9$)	0.93 ± 0.06	1.00 ± 0.01	0.96 ± 0.03	0.36 ± 0.08	0.25 ± 0.06	0.30 ± 0.06
		GFCI	0.89 ± 0.07	1.00 ± 0.00	0.93 ± 0.05	0.46 ± 0.09	0.34 ± 0.10	0.39 ± 0.08
	60	IGFCI ($\kappa = 0.001$)	0.91 ± 0.06	1.00 ± 0.00	0.94 ± 0.04	0.33 ± 0.16	0.25 ± 0.09	0.31 ± 0.10
		IGFCI ($\kappa = 0.1$)	0.90 ± 0.04	1.00 ± 0.00	0.93 ± 0.04	0.35 ± 0.14	0.26 ± 0.10	0.32 ± 0.07
		IGFCI ($\kappa = 0.5$)	0.91 ± 0.05	1.00 ± 0.00	0.93 ± 0.04	0.35 ± 0.14	0.26 ± 0.10	0.31 ± 0.08
		IGFCI ($\kappa = 0.9$)	0.89 ± 0.04	1.00 ± 0.00	0.93 ± 0.04	0.35 ± 0.14	0.25 ± 0.10	0.31 ± 0.08
		GFCI	0.84 ± 0.10	1.00 ± 0.00	0.90 ± 0.07	0.44 ± 0.13	0.35 ± 0.12	0.42 ± 0.06
20	40	IGFCI ($\kappa = 0.001$)	0.82 ± 0.08	0.99 ± 0.02	0.92 ± 0.04	0.38 ± 0.11	0.37 ± 0.10	0.36 ± 0.07
		IGFCI ($\kappa = 0.1$)	0.85 ± 0.07	0.96 ± 0.03	0.91 ± 0.04	0.39 ± 0.11	0.38 ± 0.09	0.37 ± 0.07
		IGFCI ($\kappa = 0.5$)	0.83 ± 0.08	0.94 ± 0.03	0.89 ± 0.04	0.39 ± 0.12	0.38 ± 0.10	0.37 ± 0.07
		IGFCI ($\kappa = 0.9$)	0.80 ± 0.09	0.93 ± 0.04	0.87 ± 0.05	0.39 ± 0.12	0.38 ± 0.09	0.37 ± 0.08
		GFCI	0.71 ± 0.06	0.98 ± 0.03	0.85 ± 0.02	0.46 ± 0.10	0.42 ± 0.10	0.42 ± 0.08
	80	IGFCI ($\kappa = 0.001$)	0.88 ± 0.09	0.99 ± 0.02	0.92 ± 0.06	0.31 ± 0.09	0.21 ± 0.05	0.25 ± 0.05
		IGFCI ($\kappa = 0.1$)	0.89 ± 0.05	0.97 ± 0.04	0.92 ± 0.04	0.29 ± 0.10	0.20 ± 0.05	0.24 ± 0.05
		IGFCI ($\kappa = 0.5$)	0.87 ± 0.04	0.97 ± 0.04	0.91 ± 0.03	0.30 ± 0.09	0.21 ± 0.05	0.25 ± 0.05
		IGFCI ($\kappa = 0.9$)	0.86 ± 0.05	0.96 ± 0.05	0.90 ± 0.04	0.29 ± 0.09	0.21 ± 0.04	0.24 ± 0.05
		GFCI	0.82 ± 0.07	0.99 ± 0.01	0.89 ± 0.06	0.35 ± 0.08	0.24 ± 0.06	0.29 ± 0.04
	120	IGFCI ($\kappa = 0.001$)	0.90 ± 0.06	0.95 ± 0.09	0.92 ± 0.06	0.26 ± 0.07	0.16 ± 0.04	0.20 ± 0.05
		IGFCI ($\kappa = 0.1$)	0.92 ± 0.04	0.97 ± 0.07	0.93 ± 0.05	0.25 ± 0.07	0.16 ± 0.04	0.20 ± 0.05
		IGFCI ($\kappa = 0.5$)	0.90 ± 0.03	0.97 ± 0.07	0.92 ± 0.04	0.26 ± 0.08	0.16 ± 0.04	0.21 ± 0.05
		IGFCI ($\kappa = 0.9$)	0.89 ± 0.03	0.97 ± 0.05	0.92 ± 0.03	0.27 ± 0.08	0.15 ± 0.04	0.21 ± 0.05
		GFCI	0.85 ± 0.07	0.97 ± 0.07	0.89 ± 0.07	0.31 ± 0.07	0.20 ± 0.04	0.25 ± 0.05
50	100	IGFCI ($\kappa = 0.001$)	0.82 ± 0.03	0.99 ± 0.01	0.92 ± 0.02	0.39 ± 0.11	0.36 ± 0.07	0.37 ± 0.08
		IGFCI ($\kappa = 0.1$)	0.84 ± 0.06	0.98 ± 0.02	0.92 ± 0.03	0.40 ± 0.09	0.38 ± 0.07	0.38 ± 0.08
		IGFCI ($\kappa = 0.5$)	0.81 ± 0.04	0.93 ± 0.04	0.88 ± 0.02	0.41 ± 0.09	0.38 ± 0.06	0.39 ± 0.07
		IGFCI ($\kappa = 0.9$)	0.76 ± 0.03	0.90 ± 0.06	0.83 ± 0.04	0.41 ± 0.09	0.38 ± 0.06	0.38 ± 0.06
		GFCI	0.73 ± 0.03	0.99 ± 0.02	0.87 ± 0.03	0.45 ± 0.11	0.42 ± 0.07	0.42 ± 0.08

Table 64: Adjacency precision (P) and recall (R) results for $N = 5000$ training cases.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGFCI ($\kappa = 0.001$)	0.88 ± 0.11	0.98 ± 0.05	0.93 ± 0.06	0.59 ± 0.13	0.61 ± 0.10	0.59 ± 0.07
		IGFCI ($\kappa = 0.1$)	0.89 ± 0.12	0.98 ± 0.04	0.94 ± 0.06	0.59 ± 0.14	0.64 ± 0.15	0.60 ± 0.11
		IGFCI ($\kappa = 0.5$)	0.86 ± 0.12	0.97 ± 0.07	0.93 ± 0.05	0.58 ± 0.13	0.63 ± 0.15	0.60 ± 0.11
		IGFCI ($\kappa = 0.9$)	0.87 ± 0.13	0.97 ± 0.07	0.93 ± 0.06	0.58 ± 0.13	0.63 ± 0.15	0.60 ± 0.11
		GFCI	0.75 ± 0.14	1.00 ± 0.00	0.86 ± 0.09	0.73 ± 0.14	0.75 ± 0.13	0.73 ± 0.11
	40	IGFCI ($\kappa = 0.001$)	0.91 ± 0.04	1.00 ± 0.00	0.94 ± 0.03	0.42 ± 0.10	0.32 ± 0.06	0.36 ± 0.06
		IGFCI ($\kappa = 0.1$)	0.91 ± 0.07	1.00 ± 0.00	0.95 ± 0.04	0.42 ± 0.09	0.31 ± 0.06	0.36 ± 0.05
		IGFCI ($\kappa = 0.5$)	0.90 ± 0.05	1.00 ± 0.01	0.94 ± 0.03	0.41 ± 0.09	0.31 ± 0.08	0.36 ± 0.07
		IGFCI ($\kappa = 0.9$)	0.90 ± 0.05	0.99 ± 0.01	0.94 ± 0.03	0.41 ± 0.08	0.31 ± 0.07	0.35 ± 0.06
		GFCI	0.77 ± 0.08	1.00 ± 0.00	0.86 ± 0.05	0.62 ± 0.09	0.53 ± 0.09	0.57 ± 0.05
	60	IGFCI ($\kappa = 0.001$)	0.89 ± 0.07	1.00 ± 0.00	0.93 ± 0.05	0.44 ± 0.16	0.35 ± 0.09	0.40 ± 0.08
		IGFCI ($\kappa = 0.1$)	0.92 ± 0.07	1.00 ± 0.00	0.94 ± 0.05	0.42 ± 0.15	0.36 ± 0.10	0.40 ± 0.10
		IGFCI ($\kappa = 0.5$)	0.90 ± 0.08	1.00 ± 0.00	0.93 ± 0.06	0.40 ± 0.15	0.33 ± 0.09	0.38 ± 0.10
		IGFCI ($\kappa = 0.9$)	0.90 ± 0.08	1.00 ± 0.00	0.93 ± 0.06	0.40 ± 0.15	0.33 ± 0.09	0.37 ± 0.09
		GFCI	0.78 ± 0.09	1.00 ± 0.00	0.85 ± 0.05	0.65 ± 0.13	0.51 ± 0.13	0.60 ± 0.09
20	40	IGFCI ($\kappa = 0.001$)	0.85 ± 0.10	0.98 ± 0.02	0.93 ± 0.05	0.48 ± 0.12	0.51 ± 0.14	0.48 ± 0.10
		IGFCI ($\kappa = 0.1$)	0.86 ± 0.10	0.98 ± 0.02	0.93 ± 0.05	0.48 ± 0.11	0.51 ± 0.14	0.48 ± 0.10
		IGFCI ($\kappa = 0.5$)	0.85 ± 0.10	0.98 ± 0.03	0.93 ± 0.05	0.46 ± 0.11	0.50 ± 0.14	0.47 ± 0.11
		IGFCI ($\kappa = 0.9$)	0.84 ± 0.10	0.97 ± 0.03	0.92 ± 0.05	0.46 ± 0.11	0.49 ± 0.14	0.47 ± 0.11
		GFCI	0.66 ± 0.07	0.99 ± 0.02	0.83 ± 0.04	0.53 ± 0.11	0.55 ± 0.12	0.53 ± 0.09
	80	IGFCI ($\kappa = 0.001$)	0.86 ± 0.08	0.98 ± 0.03	0.91 ± 0.06	0.34 ± 0.09	0.24 ± 0.06	0.28 ± 0.07
		IGFCI ($\kappa = 0.1$)	0.89 ± 0.04	0.97 ± 0.04	0.92 ± 0.04	0.33 ± 0.09	0.25 ± 0.06	0.28 ± 0.06
		IGFCI ($\kappa = 0.5$)	0.89 ± 0.05	0.96 ± 0.04	0.92 ± 0.04	0.34 ± 0.10	0.25 ± 0.06	0.28 ± 0.06
		IGFCI ($\kappa = 0.9$)	0.88 ± 0.05	0.96 ± 0.04	0.91 ± 0.04	0.33 ± 0.10	0.23 ± 0.05	0.27 ± 0.06
		GFCI	0.76 ± 0.05	0.98 ± 0.02	0.85 ± 0.04	0.42 ± 0.08	0.30 ± 0.08	0.36 ± 0.06
	120	IGFCI ($\kappa = 0.001$)	0.88 ± 0.04	0.98 ± 0.04	0.91 ± 0.03	0.30 ± 0.07	0.19 ± 0.06	0.24 ± 0.06
		IGFCI ($\kappa = 0.1$)	0.91 ± 0.05	0.98 ± 0.03	0.93 ± 0.03	0.30 ± 0.08	0.18 ± 0.05	0.24 ± 0.05
		IGFCI ($\kappa = 0.5$)	0.90 ± 0.04	0.98 ± 0.03	0.93 ± 0.03	0.30 ± 0.08	0.18 ± 0.04	0.24 ± 0.05
		IGFCI ($\kappa = 0.9$)	0.89 ± 0.04	0.99 ± 0.01	0.93 ± 0.03	0.28 ± 0.08	0.18 ± 0.04	0.23 ± 0.05
		GFCI	0.77 ± 0.04	1.00 ± 0.00	0.85 ± 0.03	0.37 ± 0.07	0.26 ± 0.05	0.31 ± 0.05
50	100	IGFCI ($\kappa = 0.001$)	0.81 ± 0.03	0.98 ± 0.02	0.91 ± 0.02	0.49 ± 0.11	0.48 ± 0.07	0.48 ± 0.08
		IGFCI ($\kappa = 0.1$)	0.82 ± 0.06	0.97 ± 0.02	0.90 ± 0.04	0.50 ± 0.11	0.49 ± 0.08	0.49 ± 0.09
		IGFCI ($\kappa = 0.5$)	0.81 ± 0.05	0.95 ± 0.04	0.89 ± 0.04	0.50 ± 0.10	0.48 ± 0.08	0.48 ± 0.08
		IGFCI ($\kappa = 0.9$)	0.78 ± 0.05	0.93 ± 0.04	0.87 ± 0.04	0.50 ± 0.09	0.47 ± 0.08	0.47 ± 0.08
		GFCI	0.66 ± 0.05	0.99 ± 0.01	0.84 ± 0.04	0.54 ± 0.10	0.51 ± 0.08	0.52 ± 0.09

Table 65: Arrowhead precision (P) and recall (R) results for $N = 200$ training cases.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGFCI ($\kappa = 0.001$)	0.34 ± 0.43	0.21 ± 0.29	0.33 ± 0.37	0.02 ± 0.05	0.02 ± 0.04	0.02 ± 0.03
		IGFCI ($\kappa = 0.1$)	0.38 ± 0.40	0.22 ± 0.29	0.36 ± 0.39	0.02 ± 0.05	0.02 ± 0.04	0.02 ± 0.03
		IGFCI ($\kappa = 0.5$)	0.31 ± 0.37	0.30 ± 0.37	0.26 ± 0.35	0.02 ± 0.05	0.03 ± 0.05	0.03 ± 0.04
		IGFCI ($\kappa = 0.9$)	0.20 ± 0.35	0.24 ± 0.33	0.25 ± 0.34	0.03 ± 0.05	0.04 ± 0.04	0.03 ± 0.04
		GFCI	0.23 ± 0.19	0.31 ± 0.39	0.28 ± 0.32	0.02 ± 0.04	0.04 ± 0.10	0.03 ± 0.06
	40	IGFCI ($\kappa = 0.001$)	0.47 ± 0.22	0.38 ± 0.37	0.36 ± 0.31	0.06 ± 0.10	0.02 ± 0.05	0.03 ± 0.05
		IGFCI ($\kappa = 0.1$)	0.41 ± 0.36	0.37 ± 0.28	0.36 ± 0.26	0.07 ± 0.09	0.02 ± 0.02	0.03 ± 0.03
		IGFCI ($\kappa = 0.5$)	0.25 ± 0.34	0.34 ± 0.20	0.29 ± 0.23	0.04 ± 0.07	0.02 ± 0.02	0.03 ± 0.02
		IGFCI ($\kappa = 0.9$)	0.24 ± 0.30	0.38 ± 0.27	0.28 ± 0.24	0.06 ± 0.08	0.03 ± 0.03	0.04 ± 0.03
		GFCI	0.34 ± 0.12	0.50 ± 0.35	0.46 ± 0.30	0.05 ± 0.08	0.02 ± 0.05	0.03 ± 0.05
	60	IGFCI ($\kappa = 0.001$)	0.03 ± 0.05	0.27 ± 0.28	0.11 ± 0.10	0.03 ± 0.07	0.01 ± 0.03	0.02 ± 0.03
		IGFCI ($\kappa = 0.1$)	0.03 ± 0.05	0.15 ± 0.21	0.06 ± 0.09	0.04 ± 0.07	0.01 ± 0.03	0.02 ± 0.03
		IGFCI ($\kappa = 0.5$)	0.04 ± 0.09	0.15 ± 0.32	0.06 ± 0.13	0.03 ± 0.06	0.01 ± 0.02	0.01 ± 0.02
		IGFCI ($\kappa = 0.9$)	0.16 ± 0.33	0.22 ± 0.30	0.24 ± 0.32	0.11 ± 0.24	0.02 ± 0.03	0.05 ± 0.10
		GFCI	0.04 ± 0.06	0.36 ± 0.37	0.13 ± 0.14	0.03 ± 0.07	0.02 ± 0.04	0.02 ± 0.04
20	40	IGFCI ($\kappa = 0.001$)	0.53 ± 0.28	0.86 ± 0.35	0.68 ± 0.32	0.11 ± 0.13	0.06 ± 0.08	0.06 ± 0.07
		IGFCI ($\kappa = 0.1$)	0.55 ± 0.32	0.73 ± 0.39	0.65 ± 0.36	0.11 ± 0.13	0.06 ± 0.07	0.06 ± 0.06
		IGFCI ($\kappa = 0.5$)	0.62 ± 0.21	0.55 ± 0.35	0.57 ± 0.28	0.12 ± 0.13	0.07 ± 0.07	0.08 ± 0.06
		IGFCI ($\kappa = 0.9$)	0.52 ± 0.17	0.53 ± 0.28	0.50 ± 0.24	0.12 ± 0.13	0.07 ± 0.05	0.07 ± 0.05
		GFCI	0.49 ± 0.27	0.86 ± 0.35	0.62 ± 0.29	0.12 ± 0.13	0.06 ± 0.08	0.07 ± 0.07
	80	IGFCI ($\kappa = 0.001$)	0.52 ± 0.38	0.71 ± 0.32	0.67 ± 0.30	0.07 ± 0.10	0.03 ± 0.03	0.03 ± 0.03
		IGFCI ($\kappa = 0.1$)	0.48 ± 0.36	0.70 ± 0.29	0.61 ± 0.28	0.09 ± 0.10	0.03 ± 0.03	0.04 ± 0.03
		IGFCI ($\kappa = 0.5$)	0.42 ± 0.29	0.61 ± 0.29	0.53 ± 0.27	0.10 ± 0.08	0.04 ± 0.03	0.04 ± 0.03
		IGFCI ($\kappa = 0.9$)	0.43 ± 0.25	0.54 ± 0.21	0.51 ± 0.23	0.10 ± 0.08	0.04 ± 0.03	0.05 ± 0.03
		GFCI	0.46 ± 0.27	0.74 ± 0.27	0.63 ± 0.21	0.07 ± 0.12	0.03 ± 0.03	0.03 ± 0.03
	120	IGFCI ($\kappa = 0.001$)	0.60 ± 0.33	0.61 ± 0.34	0.62 ± 0.29	0.06 ± 0.07	0.01 ± 0.01	0.02 ± 0.02
		IGFCI ($\kappa = 0.1$)	0.62 ± 0.31	0.56 ± 0.32	0.57 ± 0.30	0.07 ± 0.06	0.01 ± 0.01	0.02 ± 0.02
		IGFCI ($\kappa = 0.5$)	0.50 ± 0.27	0.52 ± 0.26	0.53 ± 0.25	0.08 ± 0.05	0.02 ± 0.02	0.03 ± 0.02
		IGFCI ($\kappa = 0.9$)	0.43 ± 0.21	0.45 ± 0.25	0.45 ± 0.23	0.10 ± 0.07	0.03 ± 0.02	0.04 ± 0.02
		GFCI	0.44 ± 0.31	0.58 ± 0.37	0.50 ± 0.29	0.07 ± 0.09	0.01 ± 0.01	0.02 ± 0.03
50	100	IGFCI ($\kappa = 0.001$)	0.21 ± 0.20	0.91 ± 0.20	0.71 ± 0.23	0.03 ± 0.05	0.04 ± 0.03	0.05 ± 0.03
		IGFCI ($\kappa = 0.1$)	0.33 ± 0.29	0.81 ± 0.16	0.76 ± 0.19	0.02 ± 0.03	0.06 ± 0.02	0.05 ± 0.02
		IGFCI ($\kappa = 0.5$)	0.23 ± 0.13	0.61 ± 0.15	0.55 ± 0.13	0.07 ± 0.06	0.08 ± 0.04	0.08 ± 0.03
		IGFCI ($\kappa = 0.9$)	0.26 ± 0.10	0.46 ± 0.15	0.41 ± 0.12	0.09 ± 0.05	0.08 ± 0.03	0.08 ± 0.03
		GFCI	0.14 ± 0.18	0.91 ± 0.20	0.66 ± 0.18	0.03 ± 0.05	0.05 ± 0.03	0.05 ± 0.02

Table 66: Arrowhead precision (P) and recall (R) results for $N = 1000$ training cases.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGFCI ($\kappa = 0.001$)	0.30 ± 0.26	0.62 ± 0.33	0.43 ± 0.26	0.07 ± 0.09	0.18 ± 0.16	0.14 ± 0.12
		IGFCI ($\kappa = 0.1$)	0.29 ± 0.24	0.50 ± 0.38	0.35 ± 0.30	0.07 ± 0.09	0.14 ± 0.12	0.11 ± 0.10
		IGFCI ($\kappa = 0.5$)	0.36 ± 0.27	0.35 ± 0.35	0.36 ± 0.33	0.09 ± 0.09	0.14 ± 0.12	0.12 ± 0.10
		IGFCI ($\kappa = 0.9$)	0.30 ± 0.29	0.39 ± 0.33	0.34 ± 0.31	0.11 ± 0.10	0.14 ± 0.12	0.12 ± 0.09
		GFCI	0.18 ± 0.19	0.65 ± 0.37	0.37 ± 0.25	0.08 ± 0.10	0.20 ± 0.18	0.15 ± 0.12
	40	IGFCI ($\kappa = 0.001$)	0.20 ± 0.31	0.30 ± 0.32	0.24 ± 0.26	0.13 ± 0.18	0.06 ± 0.07	0.07 ± 0.08
		IGFCI ($\kappa = 0.1$)	0.14 ± 0.15	0.37 ± 0.32	0.33 ± 0.27	0.11 ± 0.19	0.07 ± 0.07	0.08 ± 0.07
		IGFCI ($\kappa = 0.5$)	0.25 ± 0.21	0.35 ± 0.30	0.32 ± 0.25	0.14 ± 0.18	0.07 ± 0.06	0.09 ± 0.07
		IGFCI ($\kappa = 0.9$)	0.26 ± 0.23	0.35 ± 0.29	0.31 ± 0.25	0.12 ± 0.17	0.07 ± 0.06	0.08 ± 0.07
		GFCI	0.06 ± 0.09	0.24 ± 0.31	0.15 ± 0.18	0.17 ± 0.26	0.09 ± 0.11	0.10 ± 0.13
	60	IGFCI ($\kappa = 0.001$)	0.23 ± 0.31	0.26 ± 0.19	0.27 ± 0.23	0.20 ± 0.26	0.11 ± 0.08	0.13 ± 0.16
		IGFCI ($\kappa = 0.1$)	0.22 ± 0.26	0.27 ± 0.18	0.28 ± 0.21	0.21 ± 0.26	0.09 ± 0.07	0.12 ± 0.14
		IGFCI ($\kappa = 0.5$)	0.21 ± 0.26	0.30 ± 0.21	0.28 ± 0.22	0.19 ± 0.26	0.11 ± 0.07	0.14 ± 0.15
		IGFCI ($\kappa = 0.9$)	0.18 ± 0.23	0.29 ± 0.21	0.25 ± 0.20	0.18 ± 0.26	0.11 ± 0.08	0.13 ± 0.15
		GFCI	0.19 ± 0.24	0.26 ± 0.18	0.25 ± 0.20	0.23 ± 0.26	0.13 ± 0.06	0.16 ± 0.14
20	40	IGFCI ($\kappa = 0.001$)	0.52 ± 0.25	0.77 ± 0.22	0.70 ± 0.21	0.24 ± 0.17	0.18 ± 0.09	0.18 ± 0.09
		IGFCI ($\kappa = 0.1$)	0.68 ± 0.24	0.74 ± 0.20	0.74 ± 0.19	0.24 ± 0.17	0.19 ± 0.08	0.19 ± 0.08
		IGFCI ($\kappa = 0.5$)	0.59 ± 0.15	0.72 ± 0.20	0.70 ± 0.18	0.23 ± 0.17	0.18 ± 0.08	0.19 ± 0.09
		IGFCI ($\kappa = 0.9$)	0.55 ± 0.12	0.70 ± 0.19	0.68 ± 0.16	0.23 ± 0.17	0.18 ± 0.08	0.18 ± 0.09
		GFCI	0.38 ± 0.18	0.73 ± 0.23	0.60 ± 0.11	0.34 ± 0.19	0.20 ± 0.09	0.21 ± 0.09
	80	IGFCI ($\kappa = 0.001$)	0.54 ± 0.23	0.70 ± 0.18	0.63 ± 0.19	0.32 ± 0.18	0.12 ± 0.05	0.14 ± 0.06
		IGFCI ($\kappa = 0.1$)	0.62 ± 0.21	0.75 ± 0.19	0.68 ± 0.20	0.25 ± 0.15	0.10 ± 0.05	0.12 ± 0.05
		IGFCI ($\kappa = 0.5$)	0.58 ± 0.18	0.74 ± 0.16	0.68 ± 0.16	0.28 ± 0.15	0.11 ± 0.05	0.13 ± 0.05
		IGFCI ($\kappa = 0.9$)	0.57 ± 0.17	0.72 ± 0.17	0.65 ± 0.16	0.27 ± 0.16	0.11 ± 0.04	0.13 ± 0.05
		GFCI	0.46 ± 0.16	0.63 ± 0.22	0.56 ± 0.17	0.34 ± 0.17	0.13 ± 0.06	0.15 ± 0.06
	120	IGFCI ($\kappa = 0.001$)	0.56 ± 0.28	0.52 ± 0.30	0.54 ± 0.26	0.24 ± 0.10	0.07 ± 0.04	0.09 ± 0.04
		IGFCI ($\kappa = 0.1$)	0.59 ± 0.27	0.64 ± 0.22	0.62 ± 0.20	0.20 ± 0.07	0.07 ± 0.03	0.09 ± 0.03
		IGFCI ($\kappa = 0.5$)	0.52 ± 0.24	0.62 ± 0.18	0.58 ± 0.16	0.20 ± 0.08	0.07 ± 0.03	0.09 ± 0.03
		IGFCI ($\kappa = 0.9$)	0.53 ± 0.23	0.58 ± 0.16	0.57 ± 0.15	0.22 ± 0.10	0.07 ± 0.03	0.09 ± 0.03
		GFCI	0.50 ± 0.28	0.50 ± 0.35	0.48 ± 0.27	0.26 ± 0.14	0.08 ± 0.05	0.11 ± 0.06
50	100	IGFCI ($\kappa = 0.001$)	0.48 ± 0.11	0.83 ± 0.13	0.74 ± 0.08	0.27 ± 0.12	0.21 ± 0.08	0.21 ± 0.08
		IGFCI ($\kappa = 0.1$)	0.66 ± 0.20	0.82 ± 0.12	0.79 ± 0.12	0.25 ± 0.11	0.20 ± 0.07	0.20 ± 0.07
		IGFCI ($\kappa = 0.5$)	0.60 ± 0.11	0.76 ± 0.12	0.72 ± 0.10	0.29 ± 0.11	0.20 ± 0.06	0.21 ± 0.07
		IGFCI ($\kappa = 0.9$)	0.52 ± 0.12	0.70 ± 0.11	0.66 ± 0.09	0.29 ± 0.10	0.20 ± 0.06	0.21 ± 0.06
		GFCI	0.36 ± 0.15	0.79 ± 0.14	0.64 ± 0.08	0.30 ± 0.16	0.21 ± 0.08	0.22 ± 0.09

Table 67: Arrowhead precision (P) and recall (R) results for $N = 5000$ training cases.

# Variables	# Edges	Method	P_{IS}	P_{other}	P	R_{IS}	R_{other}	R
10	20	IGFCI ($\kappa = 0.001$)	0.35 ± 0.25	0.66 ± 0.28	0.54 ± 0.24	0.30 ± 0.27	0.32 ± 0.21	0.30 ± 0.14
		IGFCI ($\kappa = 0.1$)	0.41 ± 0.21	0.65 ± 0.30	0.60 ± 0.24	0.32 ± 0.31	0.32 ± 0.20	0.30 ± 0.14
		IGFCI ($\kappa = 0.5$)	0.47 ± 0.26	0.66 ± 0.31	0.66 ± 0.27	0.38 ± 0.31	0.32 ± 0.20	0.30 ± 0.14
		IGFCI ($\kappa = 0.9$)	0.49 ± 0.30	0.66 ± 0.31	0.67 ± 0.28	0.36 ± 0.26	0.32 ± 0.20	0.29 ± 0.14
		GFCI	0.29 ± 0.28	0.64 ± 0.31	0.50 ± 0.22	0.40 ± 0.39	0.35 ± 0.21	0.35 ± 0.18
	40	IGFCI ($\kappa = 0.001$)	0.32 ± 0.31	0.42 ± 0.28	0.40 ± 0.22	0.26 ± 0.18	0.17 ± 0.14	0.19 ± 0.12
		IGFCI ($\kappa = 0.1$)	0.32 ± 0.31	0.44 ± 0.26	0.40 ± 0.22	0.27 ± 0.17	0.17 ± 0.12	0.19 ± 0.11
		IGFCI ($\kappa = 0.5$)	0.28 ± 0.24	0.44 ± 0.25	0.39 ± 0.21	0.27 ± 0.15	0.18 ± 0.11	0.20 ± 0.09
		IGFCI ($\kappa = 0.9$)	0.29 ± 0.24	0.45 ± 0.26	0.40 ± 0.20	0.28 ± 0.15	0.18 ± 0.11	0.20 ± 0.10
		GFCI	0.19 ± 0.18	0.35 ± 0.16	0.26 ± 0.09	0.53 ± 0.21	0.32 ± 0.15	0.35 ± 0.13
	60	IGFCI ($\kappa = 0.001$)	0.16 ± 0.16	0.29 ± 0.23	0.25 ± 0.16	0.35 ± 0.24	0.19 ± 0.13	0.21 ± 0.18
		IGFCI ($\kappa = 0.1$)	0.16 ± 0.16	0.30 ± 0.24	0.25 ± 0.16	0.30 ± 0.23	0.18 ± 0.12	0.20 ± 0.18
		IGFCI ($\kappa = 0.5$)	0.16 ± 0.16	0.30 ± 0.24	0.25 ± 0.17	0.27 ± 0.25	0.18 ± 0.12	0.20 ± 0.18
		IGFCI ($\kappa = 0.9$)	0.16 ± 0.16	0.30 ± 0.24	0.26 ± 0.17	0.24 ± 0.25	0.17 ± 0.12	0.20 ± 0.18
		GFCI	0.07 ± 0.07	0.33 ± 0.23	0.20 ± 0.12	0.46 ± 0.26	0.36 ± 0.10	0.36 ± 0.16
20	40	IGFCI ($\kappa = 0.001$)	0.67 ± 0.25	0.80 ± 0.24	0.76 ± 0.22	0.39 ± 0.23	0.33 ± 0.15	0.32 ± 0.14
		IGFCI ($\kappa = 0.1$)	0.72 ± 0.21	0.81 ± 0.24	0.79 ± 0.22	0.39 ± 0.22	0.32 ± 0.14	0.32 ± 0.14
		IGFCI ($\kappa = 0.5$)	0.72 ± 0.20	0.82 ± 0.23	0.79 ± 0.21	0.37 ± 0.20	0.31 ± 0.13	0.31 ± 0.13
		IGFCI ($\kappa = 0.9$)	0.70 ± 0.22	0.82 ± 0.22	0.78 ± 0.21	0.37 ± 0.20	0.31 ± 0.13	0.30 ± 0.12
		GFCI	0.35 ± 0.07	0.78 ± 0.21	0.62 ± 0.13	0.55 ± 0.25	0.36 ± 0.13	0.37 ± 0.12
	80	IGFCI ($\kappa = 0.001$)	0.58 ± 0.21	0.72 ± 0.14	0.66 ± 0.14	0.37 ± 0.20	0.15 ± 0.05	0.18 ± 0.07
		IGFCI ($\kappa = 0.1$)	0.65 ± 0.17	0.73 ± 0.13	0.69 ± 0.13	0.32 ± 0.16	0.15 ± 0.05	0.17 ± 0.05
		IGFCI ($\kappa = 0.5$)	0.63 ± 0.18	0.73 ± 0.12	0.69 ± 0.12	0.31 ± 0.14	0.15 ± 0.05	0.17 ± 0.05
		IGFCI ($\kappa = 0.9$)	0.60 ± 0.16	0.72 ± 0.11	0.68 ± 0.11	0.29 ± 0.13	0.14 ± 0.04	0.16 ± 0.04
		GFCI	0.38 ± 0.13	0.57 ± 0.13	0.48 ± 0.12	0.50 ± 0.20	0.20 ± 0.07	0.23 ± 0.07
	120	IGFCI ($\kappa = 0.001$)	0.47 ± 0.24	0.68 ± 0.08	0.59 ± 0.13	0.25 ± 0.10	0.11 ± 0.06	0.13 ± 0.05
		IGFCI ($\kappa = 0.1$)	0.57 ± 0.20	0.67 ± 0.11	0.63 ± 0.10	0.26 ± 0.07	0.11 ± 0.05	0.13 ± 0.05
		IGFCI ($\kappa = 0.5$)	0.58 ± 0.20	0.69 ± 0.10	0.64 ± 0.10	0.27 ± 0.06	0.11 ± 0.04	0.13 ± 0.04
		IGFCI ($\kappa = 0.9$)	0.57 ± 0.19	0.69 ± 0.10	0.64 ± 0.10	0.25 ± 0.07	0.11 ± 0.04	0.13 ± 0.04
		GFCI	0.34 ± 0.10	0.54 ± 0.15	0.45 ± 0.10	0.43 ± 0.10	0.15 ± 0.06	0.18 ± 0.05
50	100	IGFCI ($\kappa = 0.001$)	0.52 ± 0.10	0.82 ± 0.15	0.74 ± 0.12	0.37 ± 0.15	0.30 ± 0.10	0.30 ± 0.09
		IGFCI ($\kappa = 0.1$)	0.59 ± 0.16	0.79 ± 0.17	0.74 ± 0.14	0.38 ± 0.14	0.29 ± 0.09	0.30 ± 0.09
		IGFCI ($\kappa = 0.5$)	0.57 ± 0.13	0.76 ± 0.17	0.71 ± 0.14	0.38 ± 0.14	0.29 ± 0.09	0.30 ± 0.09
		IGFCI ($\kappa = 0.9$)	0.53 ± 0.12	0.75 ± 0.15	0.69 ± 0.12	0.39 ± 0.13	0.28 ± 0.09	0.29 ± 0.08
		GFCI	0.30 ± 0.05	0.79 ± 0.15	0.61 ± 0.08	0.44 ± 0.16	0.32 ± 0.09	0.33 ± 0.09

Table 68: Strict SHD (S-SHD), lenient SHD (L-SHD), and adjacency SHD (A-SHD) and for $N = 200$ training cases.

			Added			Deleted			Reoriented					
# Variables	# Edges	Method	IS	Other	Overall	IS	Other	Overall	IS	Other	Overall	S-SHD	L-SHD	A-SHD
10	20	IGFCI ($\kappa = 0.001$)	0.09	0.00	0.09	3.66	4.61	8.27	0.35	0.77	1.11	9.48	8.46	8.37
		IGFCI ($\kappa = 0.1$)	0.20	0.01	0.21	3.27	4.29	7.57	0.46	1.00	1.46	9.24	7.88	7.78
		IGFCI ($\kappa = 0.5$)	0.35	0.03	0.37	3.23	4.16	7.39	0.49	1.10	1.59	9.35	7.89	7.76
		IGFCI ($\kappa = 0.9$)	0.32	0.22	0.53	3.15	4.07	7.22	0.74	1.03	1.78	9.53	7.91	7.75
		GFCI	0.34	0.01	0.36	2.86	3.45	6.31	0.73	1.37	2.10	8.76	6.85	6.66
	40	IGFCI ($\kappa = 0.001$)	0.04	0.00	0.04	8.15	8.82	16.97	0.48	0.37	0.84	17.84	17.13	17.00
		IGFCI ($\kappa = 0.1$)	0.18	0.00	0.18	7.60	8.51	16.12	0.61	0.52	1.13	17.43	16.44	16.30
		IGFCI ($\kappa = 0.5$)	0.37	0.00	0.37	7.23	8.29	15.53	0.93	0.71	1.64	17.54	16.07	15.90
		IGFCI ($\kappa = 0.9$)	0.40	0.00	0.40	6.99	8.26	15.25	1.14	0.76	1.90	17.54	15.89	15.65
		GFCI	0.31	0.00	0.31	6.78	7.87	14.65	0.75	0.87	1.62	16.58	15.09	14.96
	60	IGFCI ($\kappa = 0.001$)	0.16	0.00	0.16	8.66	7.85	16.51	0.89	0.47	1.36	18.04	16.72	16.67
		IGFCI ($\kappa = 0.1$)	0.17	0.00	0.17	8.10	7.40	15.50	1.23	0.69	1.93	17.60	15.74	15.67
		IGFCI ($\kappa = 0.5$)	0.23	0.00	0.23	8.09	7.25	15.35	1.27	0.75	2.02	17.60	15.67	15.58
		IGFCI ($\kappa = 0.9$)	0.22	0.00	0.22	7.98	7.06	15.04	1.30	0.93	2.22	17.49	15.38	15.27
		GFCI	0.41	0.00	0.41	7.22	6.72	13.94	1.58	1.05	2.64	16.99	14.42	14.35
20	40	IGFCI ($\kappa = 0.001$)	0.45	0.00	0.45	8.52	15.49	24.01	1.01	1.41	2.42	26.88	24.54	24.46
		IGFCI ($\kappa = 0.1$)	0.44	0.01	0.45	8.10	14.77	22.87	1.12	1.96	3.09	26.41	23.43	23.32
		IGFCI ($\kappa = 0.5$)	0.79	0.41	1.20	7.81	14.34	22.15	1.28	2.31	3.59	26.93	23.62	23.34
		IGFCI ($\kappa = 0.9$)	0.92	0.55	1.47	7.76	14.22	21.98	1.40	2.53	3.94	27.39	23.78	23.46
		GFCI	0.77	0.00	0.77	7.48	13.76	21.24	1.58	2.65	4.23	26.24	22.08	22.01
	80	IGFCI ($\kappa = 0.001$)	0.25	0.02	0.27	23.95	26.19	50.14	1.90	1.11	3.01	53.42	50.76	50.41
		IGFCI ($\kappa = 0.1$)	0.45	0.05	0.50	22.83	25.25	48.09	2.72	1.71	4.43	53.03	48.97	48.59
		IGFCI ($\kappa = 0.5$)	0.74	0.08	0.83	22.03	24.67	46.70	3.40	1.97	5.37	52.90	48.00	47.53
		IGFCI ($\kappa = 0.9$)	0.87	0.21	1.09	21.96	24.66	46.62	3.59	2.13	5.72	53.43	48.25	47.70
		GFCI	0.66	0.02	0.68	21.68	24.24	45.92	3.68	2.29	5.97	52.56	46.93	46.60
	120	IGFCI ($\kappa = 0.001$)	0.37	0.11	0.48	26.43	28.97	55.40	2.28	1.29	3.57	59.44	56.22	55.88
		IGFCI ($\kappa = 0.1$)	0.50	0.14	0.64	25.49	28.34	53.84	2.96	1.69	4.65	59.13	54.82	54.48
		IGFCI ($\kappa = 0.5$)	0.77	0.19	0.96	25.03	28.09	53.12	3.20	1.92	5.12	59.19	54.42	54.08
		IGFCI ($\kappa = 0.9$)	1.01	0.27	1.27	24.96	27.95	52.92	3.18	2.00	5.18	59.37	54.65	54.19
		GFCI	0.69	0.11	0.80	24.03	27.34	51.38	4.22	2.63	6.85	59.03	52.56	52.17
50	100	IGFCI ($\kappa = 0.001$)	0.60	0.00	0.60	22.60	43.02	65.62	2.59	3.44	6.03	72.26	66.35	66.22
		IGFCI ($\kappa = 0.1$)	1.01	0.35	1.36	21.12	41.02	62.13	3.26	4.95	8.21	71.71	63.88	63.49
		IGFCI ($\kappa = 0.5$)	2.60	2.22	4.82	20.16	39.17	59.33	3.91	6.27	10.17	74.32	65.03	64.15
		IGFCI ($\kappa = 0.9$)	4.05	3.87	7.92	20.25	38.84	59.09	4.03	7.13	11.17	78.18	68.03	67.01
		GFCI	1.15	0.11	1.26	19.94	38.92	58.87	4.02	6.61	10.63	70.76	60.26	60.13

Table 69: Strict SHD (S-SHD), lenient SHD (L-SHD), and adjacency SHD (A-SHD) and for $N = 1000$ training cases.

			Added			Deleted			Reoriented					
# Variables	# Edges	Method	IS	Other	Overall	IS	Other	Overall	IS	Other	Overall	S-SHD	L-SHD	A-SHD
10	20	IGFCI ($\kappa = 0.001$)	0.27	0.07	0.33	2.58	3.38	5.96	1.07	1.59	2.66	8.96	6.46	6.30
		IGFCI ($\kappa = 0.1$)	0.24	0.04	0.27	2.59	3.48	6.07	1.07	1.53	2.59	8.94	6.52	6.35
		IGFCI ($\kappa = 0.5$)	0.24	0.12	0.36	2.62	3.33	5.95	0.94	1.54	2.48	8.78	6.54	6.31
		IGFCI ($\kappa = 0.9$)	0.28	0.19	0.47	2.62	3.31	5.94	0.91	1.54	2.45	8.86	6.70	6.41
		GFCI	0.65	0.10	0.75	2.17	2.94	5.12	1.42	1.72	3.14	9.01	6.04	5.87
	40	IGFCI ($\kappa = 0.001$)	0.19	0.00	0.19	6.29	7.75	14.04	1.68	1.34	3.02	17.25	14.52	14.23
		IGFCI ($\kappa = 0.1$)	0.17	0.00	0.17	5.97	7.68	13.65	1.82	1.32	3.14	16.96	14.13	13.82
		IGFCI ($\kappa = 0.5$)	0.20	0.00	0.20	6.10	7.78	13.88	1.67	1.26	2.92	17.01	14.38	14.08
		IGFCI ($\kappa = 0.9$)	0.22	0.01	0.22	6.20	7.79	13.99	1.57	1.29	2.86	17.07	14.55	14.21
		GFCI	0.65	0.00	0.65	5.18	6.88	12.06	3.01	2.02	5.03	17.74	13.26	12.71
	60	IGFCI ($\kappa = 0.001$)	0.38	0.00	0.38	7.35	6.51	13.86	1.72	1.61	3.33	17.57	14.57	14.24
		IGFCI ($\kappa = 0.1$)	0.43	0.00	0.43	7.18	6.46	13.64	1.70	1.58	3.28	17.35	14.50	14.07
		IGFCI ($\kappa = 0.5$)	0.42	0.00	0.42	7.29	6.52	13.81	1.54	1.53	3.06	17.30	14.67	14.23
		IGFCI ($\kappa = 0.9$)	0.48	0.00	0.48	7.31	6.57	13.88	1.61	1.44	3.05	17.42	14.83	14.37
		GFCI	0.97	0.00	0.97	6.24	5.45	11.68	2.46	2.48	4.94	17.59	13.24	12.65
20	40	IGFCI ($\kappa = 0.001$)	0.84	0.06	0.90	6.86	12.44	19.30	1.86	2.73	4.60	24.79	20.59	20.20
		IGFCI ($\kappa = 0.1$)	0.70	0.27	0.97	6.79	12.11	18.90	1.71	2.86	4.57	24.44	20.34	19.87
		IGFCI ($\kappa = 0.5$)	0.82	0.42	1.24	6.86	12.11	18.96	1.79	2.97	4.76	24.97	20.67	20.21
		IGFCI ($\kappa = 0.9$)	1.00	0.50	1.50	6.84	12.21	19.04	1.84	3.08	4.92	25.46	21.05	20.54
		GFCI	1.83	0.19	2.02	6.08	11.46	17.54	2.71	3.30	6.00	25.56	20.07	19.56
	80	IGFCI ($\kappa = 0.001$)	1.07	0.08	1.15	19.59	22.74	42.33	4.22	2.92	7.13	50.62	44.36	43.48
		IGFCI ($\kappa = 0.1$)	0.95	0.14	1.09	20.11	22.97	43.08	4.14	2.61	6.75	50.92	44.92	44.17
		IGFCI ($\kappa = 0.5$)	1.13	0.18	1.30	20.10	22.75	42.85	4.17	2.84	7.01	51.16	44.92	44.15
		IGFCI ($\kappa = 0.9$)	1.29	0.25	1.53	20.16	22.81	42.97	4.16	2.93	7.08	51.58	45.26	44.50
		GFCI	1.92	0.05	1.97	18.41	21.92	40.33	5.23	3.62	8.85	51.15	43.52	42.30
	120	IGFCI ($\kappa = 0.001$)	0.93	0.20	1.13	22.77	26.38	49.15	4.54	3.25	7.79	58.07	51.09	50.28
		IGFCI ($\kappa = 0.1$)	0.69	0.17	0.86	22.84	26.25	49.08	4.31	3.14	7.46	57.40	50.52	49.94
		IGFCI ($\kappa = 0.5$)	0.89	0.18	1.07	22.56	26.24	48.80	4.43	3.08	7.51	57.37	50.52	49.87
		IGFCI ($\kappa = 0.9$)	0.93	0.14	1.06	22.53	26.32	48.85	4.43	3.04	7.47	57.38	50.79	49.91
		GFCI	1.59	0.22	1.81	21.25	24.95	46.20	6.01	4.61	10.62	58.63	49.05	48.01
50	100	IGFCI ($\kappa = 0.001$)	2.47	0.11	2.58	17.23	33.85	51.08	4.80	6.19	10.99	64.64	54.58	53.66
		IGFCI ($\kappa = 0.1$)	2.11	0.41	2.52	16.92	33.34	50.26	4.24	7.11	11.35	64.14	53.61	52.78
		IGFCI ($\kappa = 0.5$)	2.77	1.38	4.15	16.65	32.81	49.46	4.31	7.48	11.79	65.40	54.41	53.61
		IGFCI ($\kappa = 0.9$)	3.65	2.12	5.77	16.75	33.05	49.80	4.66	7.85	12.51	68.09	56.61	55.57
		GFCI	4.37	0.18	4.55	15.76	31.03	46.79	6.56	8.03	14.59	65.92	52.34	51.34

Table 70: Strict SHD (S-SHD), lenient SHD (L-SHD), and adjacency SHD (A-SHD) and for $N = 5000$ training cases.

			Added			Deleted			Reoriented					
# Variables	# Edges	Method	IS	Other	Overall	IS	Other	Overall	IS	Other	Overall	S-SHD	L-SHD	A-SHD
10	20	IGFCI ($\kappa = 0.001$)	0.40	0.04	0.45	2.04	2.57	4.61	0.95	1.75	2.70	7.76	5.35	5.06
		IGFCI ($\kappa = 0.1$)	0.42	0.05	0.48	2.07	2.51	4.58	0.95	1.57	2.52	7.58	5.35	5.06
		IGFCI ($\kappa = 0.5$)	0.47	0.07	0.54	2.06	2.54	4.60	0.86	1.58	2.44	7.58	5.43	5.14
		IGFCI ($\kappa = 0.9$)	0.44	0.07	0.51	2.08	2.54	4.61	0.88	1.58	2.46	7.59	5.42	5.13
		GFCI	1.26	0.00	1.26	1.44	1.74	3.18	1.46	2.15	3.61	8.04	4.70	4.43
	40	IGFCI ($\kappa = 0.001$)	0.40	0.00	0.40	5.63	7.01	12.64	2.24	1.95	4.19	17.24	13.41	13.05
		IGFCI ($\kappa = 0.1$)	0.38	0.00	0.38	5.62	7.09	12.71	2.13	1.88	4.01	17.10	13.50	13.10
		IGFCI ($\kappa = 0.5$)	0.38	0.01	0.39	5.65	7.09	12.73	2.11	1.84	3.95	17.07	13.64	13.12
		IGFCI ($\kappa = 0.9$)	0.38	0.02	0.40	5.69	7.12	12.82	2.05	1.79	3.83	17.05	13.68	13.22
		GFCI	1.78	0.00	1.78	3.70	4.82	8.52	4.77	4.78	9.55	19.85	11.33	10.30
	60	IGFCI ($\kappa = 0.001$)	0.65	0.00	0.65	6.37	5.54	11.92	2.70	2.40	5.10	17.67	13.30	12.56
		IGFCI ($\kappa = 0.1$)	0.48	0.00	0.48	6.55	5.55	12.10	2.51	2.42	4.92	17.51	13.33	12.59
		IGFCI ($\kappa = 0.5$)	0.51	0.00	0.51	6.71	5.80	12.52	2.33	2.18	4.52	17.54	13.74	13.03
		IGFCI ($\kappa = 0.9$)	0.51	0.00	0.51	6.76	5.83	12.60	2.32	2.15	4.47	17.57	13.81	13.10
		GFCI	2.08	0.00	2.08	4.17	3.95	8.11	5.58	4.09	9.67	19.86	11.00	10.19
20	40	IGFCI ($\kappa = 0.001$)	0.89	0.15	1.04	5.85	10.16	16.00	1.97	2.90	4.86	21.91	17.71	17.04
		IGFCI ($\kappa = 0.1$)	0.83	0.14	0.97	5.83	10.10	15.93	1.86	2.94	4.80	21.70	17.55	16.90
		IGFCI ($\kappa = 0.5$)	0.83	0.18	1.02	6.03	10.31	16.34	1.65	2.90	4.55	21.91	18.01	17.36
		IGFCI ($\kappa = 0.9$)	0.88	0.24	1.12	6.07	10.37	16.44	1.65	2.86	4.52	22.08	18.21	17.56
		GFCI	2.78	0.17	2.95	5.26	9.19	14.45	3.19	3.29	6.47	23.88	18.07	17.40
	80	IGFCI ($\kappa = 0.001$)	1.42	0.12	1.53	18.87	22.29	41.17	4.39	3.18	7.57	50.27	43.79	42.70
		IGFCI ($\kappa = 0.1$)	1.08	0.22	1.30	19.16	22.07	41.22	4.17	3.43	7.60	50.13	43.58	42.53
		IGFCI ($\kappa = 0.5$)	1.12	0.24	1.36	19.02	22.08	41.10	4.34	3.42	7.76	50.22	43.54	42.46
		IGFCI ($\kappa = 0.9$)	1.22	0.25	1.46	19.21	22.45	41.66	4.35	3.31	7.65	50.78	44.19	43.12
		GFCI	3.33	0.09	3.42	16.69	20.29	36.98	7.07	5.58	12.65	53.05	42.63	40.40
	120	IGFCI ($\kappa = 0.001$)	1.18	0.14	1.31	21.36	25.66	47.02	5.09	3.52	8.61	56.95	49.70	48.34
		IGFCI ($\kappa = 0.1$)	0.85	0.13	0.99	21.57	25.86	47.43	4.53	3.23	7.76	56.18	49.53	48.42
		IGFCI ($\kappa = 0.5$)	0.99	0.12	1.11	21.50	25.83	47.33	4.53	3.21	7.74	56.18	49.55	48.44
		IGFCI ($\kappa = 0.9$)	0.99	0.06	1.05	21.93	25.85	47.78	4.30	3.16	7.46	56.29	49.91	48.83
		GFCI	3.13	0.01	3.13	19.25	23.41	42.66	7.21	5.65	12.85	58.64	48.10	45.79
50	100	IGFCI ($\kappa = 0.001$)	3.12	0.42	3.54	14.65	28.26	42.92	5.43	8.61	14.04	60.49	48.56	46.46
		IGFCI ($\kappa = 0.1$)	3.09	0.70	3.79	14.40	27.91	42.31	5.38	9.23	14.61	60.71	48.16	46.09
		IGFCI ($\kappa = 0.5$)	3.41	1.18	4.58	14.44	28.19	42.63	5.33	9.17	14.49	61.71	49.28	47.21
		IGFCI ($\kappa = 0.9$)	3.83	1.51	5.34	14.55	28.63	43.17	5.50	9.12	14.61	63.12	50.49	48.51
		GFCI	7.39	0.17	7.56	13.37	26.52	39.90	7.71	9.56	17.27	64.74	50.18	47.46

Bibliography

[Abramson et al., 1996] Abramson, B., Brown, J., Edwards, W., Murphy, A., and Winkler, R. L. (1996). Hailfinder: A Bayesian system for forecasting severe weather. *International Journal of Forecasting*, 12(1):57–71.

[Ali et al., 2009] Ali, R. A., Richardson, T. S., Spirtes, P., et al. (2009). Markov equivalence for ancestral graphs. *The Annals of Statistics*, 37(5B):2808–2837.

[American Cancer Society, 2020] American Cancer Society (2020). *Cancer Facts & Figures 2020*. Atlanta: American Cancer Society. `https://www.cancer.org/content/dam/cancer-org/research/cancer-facts-and-statistics/annual-cancer-facts-and-figures/2020/cancer-facts-and-figures-2020.pdf`.

[American Thoracic Society, 2019] American Thoracic Society (2019). *Top 20 Pneumonia Facts — 2019*. American Thoracic Society. `https://www.thoracic.org/patients/patient-resources/resources/top-pneumonia-facts.pdf`.

[Andrews et al., 2018] Andrews, B., Ramsey, J., and Cooper, G. F. (2018). Scoring Bayesian networks of mixed variables. *International Journal of Data Science and Analytics*, 6(1):1–16.

[Andrews et al., 2019] Andrews, B., Ramsey, J., and Cooper, G. F. (2019). Learning high-dimensional directed acyclic graphs with mixed data-types. *Proceedings of Machine Learning Research*, 104:4–21.

[Bartlett and Cussens, 2013] Bartlett, M. and Cussens, J. (2013). Advances in Bayesian network learning using integer programming. In *Proceedings of the 29th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 182–191. AUAI Press.

[Beal and Zoubin, 2003] Beal, M. J. and Zoubin, G. (2003). The variational Bayesian EM algorithm for incomplete data: With application to scoring graphical model structures. *Bayesian Statistics*, 7:453–464.

[Beinlich et al., 1989] Beinlich, I. A., Suermondt, H. J., Chavez, R. M., and Cooper, G. F. (1989). The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the 2nd European Conference on Artificial Intelligence in Medicine (AIME)*, pages 247–256. Springer.

[Bishop, 2006] Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

[Blum and Roli, 2003] Blum, C. and Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, 35(3):268–308.

[Borchani et al., 2006] Borchani, H., Amor, N. B., and Mellouli, K. (2006). Learning Bayesian network equivalence classes from incomplete data. In *International Conference on Discovery Science*, pages 291–295. Springer.

[Boutilier et al., 1996] Boutilier, C., Friedman, N., Goldszmidt, M., and Koller, D. (1996). Context-specific independence in Bayesian networks. In *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 115–123. Morgan Kaufmann Publishers.

[Bray et al., 2018] Bray, F., Ferlay, J., Soerjomataram, I., Siegel, R. L., Torre, L. A., and Jemal, A. (2018). Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA: A Cancer Journal for Clinicians*, 68(6):394–424.

[Cai et al., 2019] Cai, C., Cooper, G. F., Lu, K. N., Ma, X., Xu, S., Zhao, Z., Chen, X., Xue, Y., Lee, A. V., Clark, N., et al. (2019). Systematic discovery of the functional impact of somatic genome alterations in individual tumors through tumor-specific causal inference. *PLoS Computational Biology*, 15(7):e1007088.

[Chickering, 1995] Chickering, D. M. (1995). A transformational characterization of equivalent Bayesian network structures. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 87–98. Morgan Kaufmann Publishers.

[Chickering, 1996] Chickering, D. M. (1996). Learning Bayesian networks is NP-complete. In *Learning from Data*, pages 121–130. Springer.

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

[Chickering, 2020] Chickering, D. M. (2020). Statistically efficient greedy equivalence search. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 241–249. PMLR.

[Chickering et al., 1997] Chickering, D. M., Heckerman, D., and Meek, C. (1997). A Bayesian approach to learning Bayesian networks with local structure. In *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 80–89. Morgan Kaufmann Publishers.

[Chickering and Meek, 2015] Chickering, D. M. and Meek, C. (2015). Selective greedy equivalence search: Finding optimal Bayesian networks using a polynomial number of score evaluations. In *Proceedings of the 31st Conference Uncertainty in Artificial Intelligence (UAI)*, pages 211–219. AUAI Press.

[Choi et al., 2011] Choi, M. J., Tan, V. Y., Anandkumar, A., and Willsky, A. S. (2011). Learning latent tree graphical models. *Journal of Machine Learning Research*, 12(May):1771–1812.

[Claassen and Heskes, 2012] Claassen, T. and Heskes, T. (2012). A Bayesian approach to constraint-based causal inference. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 207–216. AUAI Press.

[Cleveland, 1979] Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836.

[Cleveland and Devlin, 1988] Cleveland, W. S. and Devlin, S. J. (1988). Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association*, 83(403):596–610.

[Colombo et al., 2012] Colombo, D., Maathuis, M., Kalisch, M., and Richardson, T. (2012). Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, 40(1):294–321.

[Cooper, 1995] Cooper, G. (1995). Causal discovery from data in the presence of selection bias. In *Proceedings of the 5th International Workshop on Artificial Intelligence and Statistics*, pages 140–150.

[Cooper and Herskovits, 1992] Cooper, G. F. and Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9(4):309–347.

[Corander et al., 2019] Corander, J., Hyttinen, A., Kontinen, J., Pensar, J., and Väänänen, J. (2019). A logical approach to context-specific independence. *Annals of Pure and Applied Logic*, 170(9):975–992.

[Cover, 1999] Cover, T. M. (1999). *Elements of information theory*. John Wiley & Sons.

[Daly et al., 2011] Daly, R., Shen, Q., and Aitken, S. (2011). Learning Bayesian networks: Approaches and issues. *The Knowledge Engineering Review*, 26(2):99–157.

[Dasarathy, 1991] Dasarathy, B. V. (1991). *Nearest neighbor (NN) norms: NN pattern classification techniques*. IEEE Computer Society Press.

[Dash and Druzdzel, 1999] Dash, D. and Druzdzel, M. J. (1999). A hybrid anytime algorithm for the construction of causal models from sparse data. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 142–149. Morgan Kaufmann Publishers.

[De Campos et al., 2009] De Campos, C. P., Zeng, Z., and Ji, Q. (2009). Structure learning of Bayesian networks using constraints. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 113–120.

[De Campos et al., 2003] De Campos, L. M., Fernández-Luna, J. M., and Puerta, J. M. (2003). An iterated local search algorithm for learning Bayesian networks with restarts based on conditional independence tests. *International Journal of Intelligent Systems*, 18(2):221–235.

[DeLong et al., 1988] DeLong, E. R., DeLong, D. M., and Clarke-Pearson, D. L. (1988). Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. *Biometrics*, 44(3):837–845.

[Efron and Tibshirani, 1994] Efron, B. and Tibshirani, R. J. (1994). *An introduction to the bootstrap*. CRC press.

[Elidan and Friedman, 2005] Elidan, G. and Friedman, N. (2005). Learning hidden variable networks: The information bottleneck approach. *Journal of Machine Learning Research*, 6(Jan):81–127.

[Ettinger et al., 2017] Ettinger, D. S., Wood, D. E., Aisner, D. L., Akerley, W., Bauman, J., Chirieac, L. R., D'Amico, T. A., DeCamp, M. M., Dilling, T. J., Dobelbower, M., et al. (2017). Non-small cell lung cancer, version 5.2017, NCCN clinical practice guidelines in oncology. *Journal of the National Comprehensive Cancer Network*, 15(4):504–535.

[Ferreira et al., 2013] Ferreira, A., Cooper, G. F., and Visweswaran, S. (2013). Decision path models for patient-specific modeling of patient outcomes. In *AMIA Annual Symposium Proceedings*, pages 413–421. American Medical Informatics Association.

[Fine et al., 1997] Fine, M. J. et al. (1997). A prediction rule to identify low-risk patients with community-acquired pneumonia. *New England Journal of Medicine*, 336(4):243–250.

[Friedman, 1998] Friedman, N. (1998). The Bayesian structural EM algorithm. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 129–138. Morgan Kaufmann Publishers.

[Friedman and Goldszmidt, 1998] Friedman, N. and Goldszmidt, M. (1998). Learning Bayesian networks with local structure. In *Learning in Graphical Models*, pages 421–459. Springer.

[Geiger and Heckerman, 1996] Geiger, D. and Heckerman, D. (1996). Knowledge representation and inference in similarity networks and Bayesian multinets. *Artificial Intelligence*, 82(1-2):45–74.

[Heckerman, 1998] Heckerman, D. (1998). A tutorial on learning with Bayesian networks. In *Learning in Graphical Models*, pages 301–354. Springer.

[Heckerman et al., 1995] Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243.

[Heckerman et al., 1999] Heckerman, D., Meek, C., and Cooper, G. F. (1999). A Bayesian approach to causal discovery. In *Computation, Causation, and Discovery*, pages 141–165. MIT Press.

[Herskovits, 1991] Herskovits, E. (1991). *Computer-based probabilistic-network construction.* PhD thesis, Stanford University, USA.

[Huang et al., 2019] Huang, B., Zhang, K., Xie, P., Gong, M., Xing, E. P., and Glymour, C. (2019). Specific and shared causal relation modeling and mechanism-based clustering. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 13510–13521.

[Hyttinen et al., 2014] Hyttinen, A., Eberhardt, F., and Järvisalo, M. (2014). Constraint-based causal discovery: Conflict resolution with answer set programming. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 340–349. AUAI Press.

[Hyttinen et al., 2018] Hyttinen, A., Pensar, J., Kontinen, J., and Corander, J. (2018). Structure learning for Bayesian networks over labeled DAGs. In *International Conference on Probabilistic Graphical Models (PGM)*, pages 133–144.

[Illari et al., 2011] Illari, P. M., Russo, F., and Williamson, J. (2011). *Causality in the Sciences*. Oxford University Press.

[Jaakkola et al., 2010] Jaakkola, T., Sontag, D., Globerson, A., and Meila, M. (2010). Learning Bayesian network structure using LP relaxations. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 358–365.

[Jabbari and Cooper, 2020] Jabbari, F. and Cooper, G. F. (2020). An instance-specific algorithm for learning the structure of causal Bayesian networks containing latent variables. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, pages 433–441.

[Jabbari et al., 2017a] Jabbari, F., Naeini, M. P., and Cooper, G. F. (2017a). Obtaining accurate probabilistic causal inference by post-processing calibration. *arXiv preprint arXiv:1712.08626*.

[Jabbari et al., 2017b] Jabbari, F., Ramsey, J., Spirtes, P., and Cooper, G. (2017b). Discovery of causal models that contain latent variables through Bayesian scoring of independence constraints. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*, pages 142–157. Springer.

[Jabbari et al., 2020] Jabbari, F., Villaruz, L. C., Davis, M., and Cooper, G. F. (2020). Lung cancer survival prediction using instance-specific Bayesian networks. In *International Conference on Artificial Intelligence in Medicine (AIME)*, pages 149–159. Springer.

[Jabbari et al., 2018] Jabbari, F., Visweswaran, S., and Cooper, G. F. (2018). Instance-specific Bayesian network structure learning. *Proceedings of Machine Learning Research*, 72:169–180.

[Jabbari et al., 2019] Jabbari, F., Visweswaran, S., and Cooper, G. F. (2019). An empirical investigation of instance-specific causal Bayesian network learning. In *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 2582–2585. IEEE.

[Kellum et al., 2007] Kellum, J. A. et al. (2007). Understanding the inflammatory cytokine response in pneumonia and sepsis: Results of the genetic and inflammatory markers of sepsis (GenIMS) study. *Archives of Internal Medicine*, 167(15):1655–1663.

[Kim and Pearl, 1983] Kim, J. and Pearl, J. (1983). A computational model for causal and diagnostic reasoning in inference systems. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 190–193.

[Koivisto, 2012] Koivisto, M. (2012). Advances in exact Bayesian structure discovery in Bayesian networks. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 241–248. AUAI Press.

[Koivisto and Sood, 2004] Koivisto, M. and Sood, K. (2004). Exact Bayesian structure discovery in Bayesian networks. *Journal of Machine Learning Research*, 5:549–573.

[Koski and Noble, 2012] Koski, T. J. and Noble, J. (2012). A review of Bayesian networks and structure learning. *Mathematica Applicanda*, 40(1):51–103.

[Kris et al., 2014] Kris, M. G., Johnson, B. E., Berry, L. D., Kwiatkowski, D. J., Iafrate, A. J., Wistuba, I. I., Varella-Garcia, M., Franklin, W. A., Aronson, S. L., Su, P.-F., et al. (2014). Using multiplexed assays of oncogenic drivers in lung cancers to select targeted drugs. *Journal of American Medical Association*, 311(19):1998–2006.

[Lazic et al., 2013] Lazic, N., Bishop, C., and Winn, J. (2013). Structural expectation propagation (SEP): Bayesian structure learning for networks with latent variables. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 379–387.

[Lengerich et al., 2019] Lengerich, B., Aragam, B., and Xing, E. P. (2019). Learning sample-specific models with low-rank personalized regression. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3575–3585.

[Lengerich et al., 2018] Lengerich, B. J., Aragam, B., and Xing, E. P. (2018). Personalized regression enables sample-specific pan-cancer analysis. *Bioinformatics*, 34(13):i178–i186.

[Liu et al., 2016] Liu, X., Wang, Y., Ji, H., Aihara, K., and Chen, L. (2016). Personalized characterization of diseases using sample-specific networks. *Nucleic Acids Research*, 44(22):e164–e164.

[Magliacane et al., 2016] Magliacane, S., Claassen, T., and Mooij, J. M. (2016). Ancestral causal inference. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4466–4474.

[Naeini et al., 2015] Naeini, M. P., Cooper, G., and Hauskrecht, M. (2015). Obtaining well calibrated probabilities using Bayesian binning. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 2901–2907.

[Nandy et al., 2018] Nandy, P., Hauser, A., Maathuis, M. H., et al. (2018). High-dimensional consistency in score-based and hybrid structure learning. *The Annals of Statistics*, 46(6A):3151–3183.

[Neapolitan et al., 2004] Neapolitan, R. E. et al. (2004). *Learning Bayesian networks*, volume 38. Pearson Prentice Hall Upper Saddle River, NJ.

[Network et al., 2014] Network, C. G. A. R. et al. (2014). Comprehensive molecular profiling of lung adenocarcinoma. *Nature*, 511(7511):543–550.

[Oates et al., 2016] Oates, C. J., Smith, J. Q., Mukherjee, S., and Cussens, J. (2016). Exact estimation of multiple directed acyclic graphs. *Statistics and Computing*, 26(4):797–811.

[Ogarrio et al., 2016] Ogarrio, J. M., Spirtes, P., and Ramsey, J. (2016). A hybrid causal search algorithm for latent variable models. In *International Conference on Probabilistic Graphical Models (PGM)*, pages 368–379.

[Onisko, 2003] Onisko, A. (2003). *Probabilistic causal models in medicine: Application to diagnosis in liver disorders.* PhD thesis, Institute of Biocybernetics and Biomedical Engineering, Polish Academy of Science, Warsaw.

[Parviainen and Koivisto, 2011] Parviainen, P. and Koivisto, M. (2011). Ancestor relations in the presence of unobserved variables. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*, pages 581–596. Springer.

[Pearl, 2003] Pearl, J. (2003). Causality: Models, reasoning, and inference. *Econometric Theory*, 19(675-685):46.

[Pearl, 2009] Pearl, J. (2009). Causal inference in statistics: An overview. *Statistics Surveys*, 3:96–146.

[Pensar et al., 2015] Pensar, J., Nyman, H., Koski, T., and Corander, J. (2015). Labeled directed acyclic graphs: A generalization of context-specific independence in directed graphical models. *Data Mining and Knowledge Discovery*, 29(2):503–533.

[Peters et al., 2017] Peters, J., Janzing, D., and Schölkopf, B. (2017). *Elements of causal inference*. MIT Press.

[Peters et al., 2012] Peters, J., Mooij, J., Janzing, D., and Schölkopf, B. (2012). Identifiability of causal graphs using functional models. In *Proceedings of the 27th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 589–598. AUAI Press.

[Qu and Gotman, 1997] Qu, H. and Gotman, J. (1997). A patient-specific algorithm for the detection of seizure onset in long-term eeg monitoring: Possible use as a warning device. *IEEE Transactions on Biomedical Engineering*, 44(2):115–122.

[Ramsey et al., 2017] Ramsey, J., Glymour, M., Sanchez-Romero, R., and Glymour, C. (2017). A million variables and more: The fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. *International Journal of Data Science and Analytics*, 3(2):121–129.

[Ramsey, 2015] Ramsey, J. D. (2015). Scaling up greedy equivalence search for continuous variables. *arXiv preprint arXiv:1507.07749*.

[Richardson et al., 2002] Richardson, T., Spirtes, P., et al. (2002). Ancestral graph Markov models. *The Annals of Statistics*, 30(4):962–1030.

[Rissanen, 1978] Rissanen, J. (1978). Modeling by shortest data description. *Automatica*, 14(5):465–471.

[Schulam and Saria, 2015] Schulam, P. and Saria, S. (2015). A framework for individualizing predictions of disease trajectories by exploiting multi-resolution structure. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 748–756.

[Schulam and Saria, 2016] Schulam, P. and Saria, S. (2016). Integrative analysis using coupled latent variable models for individualizing prognoses. *The Journal of Machine Learning Research*, 17(1):8244–8278.

[Schulam et al., 2015] Schulam, P., Wigley, F., and Saria, S. (2015). Clustering longitudinal clinical marker trajectories from electronic health data: Applications to phenotyping and endotype discovery. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 2956–2964.

[Schwarz, 1978] Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

[Scutari, 2010] Scutari, M. (2010). Learning Bayesian networks with the bnlearn R package. *Journal of Statistical Software*, 35(3):1–22.

[Sheiner et al., 1979] Sheiner, L. B., Beal, S., Rosenberg, B., and Marathe, V. V. (1979). Forecasting individual pharmacokinetics. *Clinical Pharmacology & Therapeutics*, 26(3):294–305.

[Shoeb et al., 2004] Shoeb, A., Edwards, H., Connolly, J., Bourgeois, B., Treves, S. T., and Guttag, J. (2004). Patient-specific seizure onset detection. *Epilepsy & Behavior*, 5(4):483–498.

[Silander and Myllymaki, 2012] Silander, T. and Myllymaki, P. (2012). A simple approach for finding the globally optimal Bayesian network structure. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 445–452. AUAI Press.

[Silva et al., 2006] Silva, R., Scheines, R., Glymour, C., and Spirtes, P. (2006). Learning the structure of linear latent variable models. *Journal of Machine Learning Research*, 7:191–246.

[Singer et al., 2016] Singer, M., Deutschman, C. S., Seymour, C. W., Shankar-Hari, M., Annane, D., Bauer, M., Bellomo, R., Bernard, G. R., Chiche, J.-D., Coopersmith, C. M., et al. (2016). The third international consensus definitions for sepsis and septic shock (Sepsis-3). *Journal of American Medical Association*, 315(8):801–810.

[Singh and Moore, 2005] Singh, A. P. and Moore, A. W. (2005). *Finding optimal Bayesian networks by dynamic programming*. Carnegie Mellon University.

[Singh and Valtorta, 1995] Singh, M. and Valtorta, M. (1995). Construction of Bayesian network structures from data: A brief survey and an efficient algorithm. *International Journal of Approximate Reasoning*, 12(2):111–131.

[Spirtes et al., 2000] Spirtes, P., Glymour, C. N., and Scheines, R. (2000). *Causation, prediction, and search*. MIT Press.

[Spirtes et al., 1995] Spirtes, P., Meek, C., and Richardson, T. (1995). Causal inference in the presence of latent variables and selection bias. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 499–506. Morgan Kaufmann Publishers.

[Studenỳ and Haws, 2014] Studenỳ, M. and Haws, D. (2014). Learning Bayesian network structure: Towards the essential graph by integer linear programming tools. *International Journal of Approximate Reasoning*, 55(4):1043–1071.

[Tibshirani, 1996] Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288.

[Travis et al., 2011] Travis, W. D., Brambilla, E., Noguchi, M., Nicholson, A. G., Geisinger, K. R., Yatabe, Y., Beer, D. G., Powell, C. A., Riely, G. J., Van Schil, P. E., et al. (2011). International association for the study of lung cancer/American Thoracic Society/European Respiratory Society international multidisciplinary classification of lung adenocarcinoma. *Journal of Thoracic Oncology*, 6(2):244–285.

[Triantafillou et al., 2014] Triantafillou, S., Tsamardinos, I., and Roumpelaki, A. (2014). Learning neighborhoods of high confidence in constraint-based causal discovery. In *European Workshop on Probabilistic Graphical Models*, pages 487–502. Springer.

[Tsamardinos et al., 2006] Tsamardinos, I., Brown, L. E., and Aliferis, C. F. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78.

[Verma and Pearl, 1990] Verma, T. and Pearl, J. (1990). Equivalence and synthesis of causal models. In *Proceedings of the 6th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 255–268. Elsevier Science.

[Visweswaran and Cooper, 2010] Visweswaran, S. and Cooper, G. F. (2010). Learning instance-specific predictive models. *Journal of Machine Learning Research*, 11(Dec):3333–3369.

[Zhang, 2008] Zhang, J. (2008). On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16-17):1873–1896.

[Zheng and Webb, 2000] Zheng, Z. and Webb, G. I. (2000). Lazy learning of Bayesian rules. *Machine Learning*, 41(1):53–84.

[Zou et al., 2017] Zou, Y., Pensar, J., and Roos, T. (2017). Representing local structure in Bayesian networks by Boolean functions. *Pattern Recognition Letters*, 95:73–77.