Journal of Machine Learning Research 26 (2025) 1-62 Submitted 1/23; Revised 1/25; Published 3/25

DAGs as Minimal $\mathcal{I}$-maps for the Induced Models of Causal Bayesian Networks under Conditioning

Xiangdong Xie XIEXD569@NENU.EDU.CN
KLAS and School of Mathematics & Statistics
Northeast Normal University
Changchun, China

Jianhua Guo [*] JHGUO@BTBU.EDU.CN
School of Mathematics and Statistics
Beijing Technology and Business University
Beijing, China

Yi Sun BRIAN@XJU.EDU.CN
School of Mathematics and Systems Science
Xinjiang University
Urumqi, China

Editor: Ilya Shpitser

Abstract

Bayesian networks (BNs) are a powerful tool for knowledge representation and reasoning, especially for complex systems. A critical task in the applications of BNs is conditional inference or inference in the presence of selection bias. However, post-conditioning, the conditional distribution family of a BN can become complex for analysis, and the corresponding induced subgraph may not accurately encode the conditional independencies for the remaining variables. In this work, we first investigate the conditions under which a BN remains closed under conditioning, meaning that the induced subgraph is consistent with the structural information of conditional distributions. Conversely, when a BN is not closed, we aim to construct a new directed acyclic graph (DAG) as a minimal $\mathcal{I}$-map for the conditional model by incorporating directed edges into the original induced graph. We present an equivalent characterization of this minimal $\mathcal{I}$-map and develop an efficient algorithm for its identification. The proposed framework improves the efficiency of conditional inference of a BN. Additionally, the DAG minimal $\mathcal{I}$-map offers graphical criteria for the safe integration of knowledge from diverse sources (subpopulations/conditional distributions), facilitating correct parameter estimation. Both theoretical analysis and simulation studies demonstrate that using a DAG minimal $\mathcal{I}$-map for conditional inference is more effective than traditional methods based on the joint distribution of the original BN.

Keywords: causal Bayesian network, conditioning, conditional inference, DAG minimal $\mathcal{I}$-map, selection bias

1. Introduction

Bayesian networks (BNs; Maathuis et al., 2018; Neapolitan, 2004; Pearl, 1988), also known as directed acyclic graph (DAG) models or recursive graphical models (Evans, 2016; Kiiveri

[*]. Corresponding author

et al., 1984; Wermuth and Lauritzen, 1983), are a popular type of graphical models used to represent conditional independence relations or causal relations among a set of variables. They have been extensively studied and have found increasing applications in various fields, including causal inference (Drton and Maathuis, 2017; Glymour et al., 2019; Heinze-Deml et al., 2018; Pearl, 2009; Spirtes et al., 2001), machine learning (Xie et al., 2006; Yu et al., 2019), and engineering (Cowell et al., 2007; Vila-Francés et al., 2013).

A BN is built upon a DAG, which is a tuple $\vec{G} = (V, \vec{E})$ consisting of a vertex set V and a directed edge set $\vec{E}$. Suppose X_V is a random vector with its components indexed by the vertices in V, and X_V takes values in the product space $\mathcal{X}_V \equiv \times_{v\in V}\mathcal{X}_v$, where each $\mathcal{X}_v$ is either a finite-dimensional vector space (e.g., the real line $\mathbb{R}$) or a finite discrete space. We use X_A to denote a subvector of X_V indexed by $A(\subseteq V)$ with a range $\mathcal{X}_A \equiv \times_{v\in A}\mathcal{X}_v$. A distribution P over X_V is *compatible* with $\vec{G}$ (or *factorizes* according to $\vec{G}$) if it can be written as

$$P(X_V) = \prod_{w\in V} P(X_w | X_{\mathbf{pa}_{\vec{G}}(w)}). \tag{1}$$

In words, for any component X_w in X_V, its conditional distribution depends only on the subvector $X_{\mathbf{pa}_{\vec{G}}(w)}$, indexed by the parent set $\mathbf{pa}_{\vec{G}}(w)$ of w in $\vec{G}$. A BN is a pair consisting of a DAG $\vec{G}$ and a distribution family $\mathcal{P}(\vec{G})$, denoted by $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$, such that any distribution within $\mathcal{P}(\vec{G})$ is compatible with $\vec{G}$. The graph $\vec{G}$ is known as the $\mathcal{I}$-map of $\mathcal{P}(\vec{G})$, which means that the conditional independencies among the components in X_V (under any distribution within $\mathcal{P}(\vec{G})$) hold whenever the corresponding d-separation assertions among the vertices in V are true in $\vec{G}$.

Motivations. A BN is usually exploited to perform efficient joint distribution inference, using the conditional independence structure to reduce the number of parameters and lower computational complexity. In various applications of BNs, a key task is conditional inference. However, developing systems that can respond efficiently and accurately to conditional queries is a challenging research problem.

In practice, data is often only available from a conditional distribution rather than a joint distribution. For instance, when analyzing the location of bullet holes on fighter jets returning from the battlefield, the analysis is conditioned on the fact that the aircraft was not shot down. In medical diagnosis data analysis, the collected data exclusively pertains to individuals who have sought medical care, while those who have not received medical attention are not observed. Similarly, conducting social surveys at universities is conditioned on the fact that the respondents have attended the university. These examples underscore the importance of adapting BNs to accurately answer probabilistic queries in such conditional settings.

In addition, even in cases where data is available from the joint distribution, a BN system may face frequent probability queries regarding a specific subpopulation. For example, when studying specific subgroups, such as males or females aged 25 to 35 years, we can generate numerous related queries regarding the probability of certain diseases after observing several specific symptoms. These diseases and symptoms can be selected in various combinations, representing subsets of all the possible conditions and symptoms. Adapting BNs for efficient and accurate computation in these scenarios is crucial for handling such subpopulation-specific queries.

Furthermore, we are also interested in conditional queries in the presence of do-intervention in a causal BN. The do-intervention, denoted as $do(X_A = x_A)$, refers to an action of setting the random vector X_A to a specific value x_A, overriding the probabilistic mechanisms of X_A defined by the original BN via Equation (1). For a detailed distinction between conditioning and intervention, refer to Glymour and Meek (1994), Koller and Friedman (2009, pp.1009–1010), Pearl (2009) and Hitchcock (2016). Specially, we are interested in the conditional intervention distribution $P(X_{R\setminus A}|do(X_A = x_A), X_C = x_C)$, which corresponds to the post-interventional conditional distribution of $X_{R\setminus A}$ given $X_C = x_C$ (Tian, 2004; Shpitser and Pearl, 2006; Kivva et al., 2023). Essentially, this concerns conditional inferences on the intervened DAG. After performing do-intervention, we can view $P(x_{V\setminus A}|do(X_A = x_A))$ as a "joint distribution" over $X_{V\setminus A}$ of the DAG $\vec{G}_{V\setminus A}$. Our goal is to perform further conditional inference based on the intervened BN with DAG $\vec{G}_{V\setminus A}$ and distribution $P(x_{V\setminus A}|do(X_A = x_A))$.

Conditional queries in the presence of do-interventions are also of practical relevance. The following example is based on the one presented in Koller and Friedman (2009). If a patient takes a specific medication and subsequently develops coughing symptoms, what is their chance of recovery? This question can be formulated as the conditional probability $P(X_H \mid do(X_M = x_M), X_C = x_C)$, where X_H represents the patient's health status, $X_M = x_M$ indicates that the patient has taken the medication, and $X_C = x_C$ reflects the presence of coughing symptoms. It is important to note that this query differs from $P(X_H \mid X_M = x_M, X_C = x_C)$. For example, patients who choose to take medication on their own may tend to be more health-conscious and thus have a generally higher likelihood of recovery. As a result, $P(X_H \mid X_M = x_M, X_C = x_C)$ may overestimate the chances of recovery compared to $P(X_H \mid do(X_M = x_M), X_C = x_C)$, which isolates the effect of the intervention.

Solution via the Minimal $\mathcal{I}$-map. Unlike the common approach of calculating conditional distributions using the ratio of joint and marginal distributions, we construct a new BN for efficient conditional inference based on the notion of a minimal $\mathcal{I}$-map (Pearl, 1988; Koller and Friedman, 2009). Suppose V is partitioned into two subsets R and C, and we have the corresponding X_R and X_C as subvectors of X_V. It is well known that conditioning on X_C tends to induce dependence among the random components in X_R. This means $\vec{G}_R$ is generally not a valid $\mathcal{I}$-map for the conditional distribution $P(X_R|X_C = x_C)$, where $\vec{G}_R$ is the subgraph of $\vec{G}$ induced by removing the vertices in C and the related edges. We aim to construct a DAG $\vec{G}^*_R$ over R that encodes all induced dependencies for $P(X_R|X_C = x_C)$, allowing for redundancy but ensuring that none are omitted. The DAG $\vec{G}^*_R$ is constructed by adding a minimal set of directed edges to the original subgraph $\vec{G}_R$, rendering $\vec{G}^*_R$ as a minimal $\mathcal{I}$-map for the conditional distributions of the BN $\mathcal{B}$. For the conditional model of a causal BN over X_V, this approach retains as much conditional independence information as possible to improve the efficiency of conditional inference, while maintaining the original causal interpretability. To distinguish the added non-causal edges from the original causal edges, we use dashed arrows to represent the added non-causal edges, and solid arrows to represent the original causal edges.

Our exploration of the minimal $\mathcal{I}$-map is accompanied by our examination of the closure properties of BNs under conditioning. Let $\mathcal{P}(\vec{G})^C$ denote the collection of conditional distributions $P(X_R|X_C = x_C)$ computed from any $P \in \mathcal{P}(\vec{G})$. Due to changes in the dependence structure, the pair $(\vec{G}_R, \mathcal{P}(\vec{G})^C)$ generally does not form a BN, indicating that the original BN $\mathcal{B}$ is not closed under conditioning on X_C. In this work, we investigate the conditions under which $\mathcal{B}$ is closed under conditioning. In scenarios where $\mathcal{B}$ is not closed, we propose to redirect our focus to another BN over X_R, $\mathcal{B}_R^* = (\vec{G}_R^*, \mathcal{P}^C(\vec{G}_R^*))$, where $\vec{G}_R^*$ is the minimal $\mathcal{I}$-map and $\mathcal{P}^C(\vec{G}_R^*)$ encompasses all conditional distributions $P(X_R|X_C = x_C)$ that are compatible with $\vec{G}_R^*$. To address statistical inference for $\mathcal{P}(\vec{G})^C$, practitioners can employ efficient inference techniques suitable for the constructed BN $\mathcal{B}_R^*$. Due to the inclusion relationship $\mathcal{P}(\vec{G})^C \subseteq \mathcal{P}^C(\vec{G}_R^*)$, the procedure leads to accurate and consistent inference results for any distribution within $\mathcal{P}(\vec{G})^C$.

Illustrative Example. We illustrate the utility of the DAG minimal $\mathcal{I}$-map on conditional probability queries through the example below, which serves purely for demonstration purposes. In the example, the number of vertices in $\vec{G}$ is small and X_C is univariate. More complicated examples are provided in Section 7 to demonstrate the effectiveness of our proposed algorithm and theoretical results. In some scenarios, for A, B, C as disjoint subsets of V (B is possibly empty), we are interested in computing the conditional probability $P(X_A|X_B, X_C = x_C)$, for some *fixed* x_C and *varying* values of X_A, X_B. The standard approach is based on the following conditional formula:

$$P(X_A|X_B, X_C = x_C) = \frac{P(X_A, X_B, X_C = x_C)}{P(X_B, X_C = x_C)}, \tag{2}$$

where the marginal distribution of the denominator on the right hand side is computed by applying existing procedures to the original BN $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$, e.g., see Chapters 9–10 of Koller and Friedman (2009). This can be computationally challenging, especially when the dimensionality of the related variables is relatively high (see Section 7 for details). Fortunately, finding the minimal $\mathcal{I}$-map and constructing the BN $\mathcal{B}_R^* = (\vec{G}_R^*, \mathcal{P}^C(\vec{G}_R^*))$ help compute $P(X_A \mid X_B, X_C = x_C)$. This is because we can exploit the structure of $\mathcal{B}_R^*$ for efficient and accurate computation. Here, we illustrate it using an example based on the one presented in Druzdzel and Díez (2003).

Example 1 *Consider an expert system whose DAG, shown in Figure 1(a), has been constructed for two diseases D_1, D_2, three symptoms S_1, S_2, S_3 and a vertex variable H indicating whether a patient is admitted to the hospital. We aim to construct a decision system for patients admitted to the hospital ($X_H = 1$). Based on our algorithm (see Section 5.5), we find a minimal $\mathcal{I}$-map $\vec{G}_R^*$ for the remaining vertices $R = \{D_1, D_2, S_1, S_2, S_3\}$ as shown in Figure 1(b). For the new BN $\mathcal{B}_R^* = (\vec{G}_R^*, \mathcal{P}^C(\vec{G}_R^*))$, any member within $\mathcal{P}(\vec{G})^C$ factors according to the minimal $\mathcal{I}$-map $\vec{G}_R^*$ (see our Theorem 19):*

$$\begin{aligned} &P(X_{D_1}, X_{D_2}, X_{S_1}, X_{S_2}, X_{S_3}|X_H = 1) \\ &= P(X_{S_1}|X_{D_1}, X_{S_2}, X_{S_3}, X_H = 1)P(X_{S_2}|X_{D_1}, X_{S_3}, X_H = 1) \\ &\quad \times P(X_{S_3}|X_{D_2}, X_H = 1)P(X_{D_1}|X_{S_3}, X_H = 1)P(X_{D_2}|X_H = 1). \end{aligned} \tag{3}$$

We can estimate each factor on the right side of (3) *based on the data collected from the hospitalized patients. Instead of* (2), *$P(X_{D_1}, X_{D_2}, X_{S_1}, X_{S_2}, X_{S_3}|X_H = 1)$ can be evaluated more efficiently based on the factorization* (3). *Other conditional probability queries can* *also be computed efficiently. Consider, for example, the conditional query*

$$P(X_{D_1}, X_{D_2}|X_{S_1}, X_{S_2}, X_{S_3}, X_H = 1) = \frac{P(X_{D_1}, X_{D_2}, X_{S_1}, X_{S_2}, X_{S_3}|X_H = 1)}{P(X_{S_1}, X_{S_2}, X_{S_3}|X_H = 1)}.$$

To compute the marginal distribution in the above denominator, we can employ the existing procedure for the new BN $\mathcal{B}_R^* = (\vec{G}_R^*, \mathcal{P}^C(\vec{G}_R^*))$.

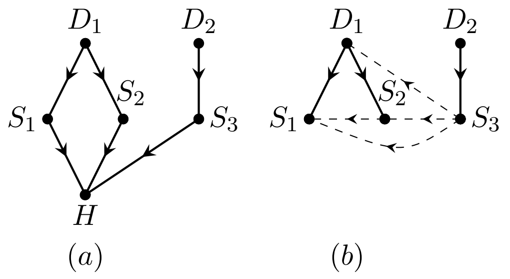


Figure 1: (a) The causal DAG $\vec{G}$ related to diseases D_1, D_2 with H conditioned on; (b) A minimal $\mathcal{I}$-map $\vec{G}_R^*$ of $P(X_{D_1}, X_{D_2}, X_{S_1}, X_{S_2}, X_{S_3}|X_H = 1)$ obtained from $\vec{G}$ in (a) by our algorithm.

Our Contributions. In this work, we approach the conditional inference of a BN $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$ via a DAG minimal $\mathcal{I}$-map. The minimal $\mathcal{I}$-map $\vec{G}_R^*$ identifies a minimal set of edges for potential dependence structure change of $P(X_R|X_C)$. The statistical inference for $P(X_R|X_C)$ is facilitated by the constructed BN $(\vec{G}_R^*, \mathcal{P}^C(\vec{G}_R^*))$, where $\mathcal{P}^C(\vec{G}_R^*)$ is a set of conditional distributions over X_R. Druzdzel and Díez (2003) considered the topic when X_C is univariate. The current work considers a multivariate conditional vector X_C, and is a more systematic investigation from both algorithmic and theoretical perspectives. Our contributions are fourfold as follows:

- We examine the closure property of a BN under conditioning in Section 4. An efficient algorithm is developed to check the closure property of a BN in Section 4.1. Equivalent characterizations of the closure property are derived from several perspectives in Section 6.
- We introduce a concept called minimal filling edge set, and connect it with the minimal $\mathcal{I}$-maps for the conditional distributions of BNs in Section 5.2. We further characterize the minimal filling edge set by vertex pairs called terminal connecting pairs. These pairs can be found by applying a converging vertex sweeping operator to the paths. See Section 5.4 for details.
- We develop a low-order polynomial time complexity algorithm to identify minimal $\mathcal{I}$-maps in Section 5.5. For a sparse causal DAG, the algorithm has linear computational complexity.

- In Section 5.3, we discuss the benefits of employing the minimal $\mathcal{I}$-map $\vec{G}_R^*$, especially using graphical criteria to determine whether knowledge or data sets from different sources can be safely combined into the conditional model. We use simulation studies to demonstrate the advantages of employing a DAG minimal $\mathcal{I}$-map to conduct conditional inference. Simulation studies also shows its advantages in repeated conditional queries with different conditioning sets and conditional inference in the presence of do-intervention. For more details, refer to Section 7.

Related Works. In the literature, various methods exist for capturing the conditional independence structure of a BN under conditioning. A commonly used method involves moralizing DAGs into undirected graphs (UGs) that are closed under conditioning (Lauritzen, 1996). Alternatively, extended graph classes such as summary graphs (Cox and Wermuth, 1996; Wermuth, 2011), MC graphs (Koster, 2002), ancestral graphs (Richardson and Spirtes, 2002), and chain mixed graphs (Sadeghi, 2016) have been explored. These classes incorporate various types of edges, not just directed edges. Although one can find an optimal graph representation of the conditional model of a given BN from these graph classes in their own frameworks, these graph classes often lack straightforward parameterizations of their associated distributions, which complicates quantitative inference. This is a significant drawback, because a primary goal of studying graphical models is to effectively address distribution-related problems.

Compared with the above, our approach to modifying the original DAGs to obtain DAG minimal $\mathcal{I}$-maps offers several advantages. First, a BN has a relatively simpler definition of the Markov property than other graphical models, as its joint distribution can be factored into a series of local conditional distributions for individual variables, rather than potential functions. Second, the number of parameters required to describe a BN grows linearly with the size of the local conditional distributions, while that of the joint distribution itself increases exponentially. Luckily, a more detailed representation of the local conditional distributions, such as noisy-or models (Peng and Reggia, 1986), soft CPD (Lerner et al., 2001), tree-CPDs (Boutilier et al., 1996), DAG-structured CPDs (Chickering et al., 1997), and neural networks (Monti and Cooper, 1996), can achieve further savings.

A Remark on the Prerequisite. In this work, a prerequisite is that the original DAG $\vec{G}$ over all variables is known. Our analysis focuses on the structural changes among the remaining variables after conditioning on a subset of random variables, and the minimal $\mathcal{I}$-map is obtained from $\vec{G}$. In practice, the DAG $\vec{G}$ may be pre-established in scenarios such as the one illustrated in Example 1. It is either hypothesized or constructed based on expert knowledge. Our proposed method is also relevant when the entire BN $\mathcal{B}$ has been estimated, but there is interest in constructing a new BN $\mathcal{B}'_R$ for efficient conditional computation. For example, a system may encounter frequent probability queries regarding a specific subpopulation. In these scenarios, our algorithm aids in identifying the minimal $\mathcal{I}$-map $\vec{G}_R^*$ from the original $\vec{G}$, and facilitates further statistical inference (see Section 7 for details).

For scenarios where the original DAG $\vec{G}$ is unavailable, it becomes compelling to estimate the minimal $\mathcal{I}$-map $\vec{G}_R^*$ directly from data sets collected under specific conditions (i.e., where $X_C = x_C$). Exploring BN structure learning from the data affected by selection bias is a promising direction for future research. For further discussion, see Section 8.

The remainder of this paper is organized as follows. Section 2 reviews the notations and terminologies of DAGs. Section 3 introduces several concepts related to a BN, its minimal $\mathcal{I}$-map and conditional model. Section 4 studies the closure of a BN under conditioning and its equivalent characterizations. In Section 5, we examine the DAG minimal $\mathcal{I}$-map when a BN is not closed under conditioning. Its equivalent characterizations and an efficient algorithm for its identification are developed. Section 6 proposes additional equivalent characterizations for the closure of a BN under conditioning. Section 7 demonstrates the benefits of performing conditional inference using a DAG minimal $\mathcal{I}$-map through simulations. Finally, we conclude with a discussion in Section 8, and provide technical proofs and additional results in the appendices. A table of notations introduced in this work can be found in Appendix A.

2. Review of Directed Acyclic Graph

In this section, we review the basic concepts and notations for DAGs. Interested readers can refer to Neapolitan (2004) for a comprehensive introduction to this topic. A DAG $\vec{G} = (V, \vec{E})$ is characterized by its set of vertices V and its set of directed edges $\vec{E}$. This structure is defined such that there are no directed cycles; that is, it is impossible to start at any vertex $v \in V$ and follow a consistent directional path that eventually returns to v. We employ (u, v) or $u \to v$ to represent a directed edge from u to v. If $(u, v) \in \vec{E}$ or $(v, u) \in \vec{E}$, we say that u, v are *adjacent* in $\vec{G}$, denoted by $u \overset{\vec{G}}{\sim} v$. Otherwise, we say that u, v are *nonadjacent* in $\vec{G}$, denoted by $u \overset{\vec{G}}{\nsim} v$. We denote by $\vec{G}_A = (A, \vec{E}_A)$ as an *induced subgraph* of $\vec{G} = (V, \vec{E})$ over A, i.e., $A \subseteq V$ and $\vec{E}_A = \vec{E} \cap (A \times A)$. The DAG $\vec{G}$ is assumed to be simple throughout this work, which means that it has no multiple edges and self loops.

A *topological order* α of a DAG $\vec{G} = (V, \vec{E})$ is a mapping α from V to $[p]$ such that $\alpha(v_i) < \alpha(v_j)$, if there exists an edge (v_i, v_j) in $\vec{G}$, where $[p] = \{1, 2, \ldots, p = |V|\}$ and $i, j \in [p]$. If all the directed edges in a DAG $\vec{G}$ represent directed causal relationships, we call $\vec{G}$ a *causal DAG*. For a set F of edges, we use $G + F$ to represent the graph $(V, \vec{E} \cup F)$, consisting of the original vertex set but with more edges included. If F consists of a single edge (x, y), we simplify $G + \{(x, y)\}$ to $G + (x, y)$. Similarly, we use $G - F$ and $G - (x, y)$ for the removal of edge(s) from G. To emphasize that we connect x and y by their topological orders, we apply $x \overset{\alpha}{\sim} y$ to denote the newly added edge (x, y) if $\alpha(x) < \alpha(y)$, and (y, x) alternatively. It is evident that $\vec{G} + x \overset{\alpha}{\sim} y$ is also a DAG.

We define a *path* connecting x and y, denoted by l_{xy}, as a sequence of distinct vertices $(x = v_1, v_2, \ldots, v_{k-1}, v_k = y)$ such that $v_i \to v_{i+1}$ or $v_i \leftarrow v_{i+1}$ for $i = 1, 2, \ldots, k-1$ in $\vec{G}$. The length of a path l_{xy} is denoted by $\rho(l_{xy})$, which is the number of edges contained in it. In particular, a single vertex x is considered a path of length zero. We also represent a path l_{xy} as $l_{xy} = \{x = v_1 \overset{\vec{G}}{\sim} v_2 \overset{\vec{G}}{\sim} \cdots \overset{\vec{G}}{\sim} v_{k-1} \overset{\vec{G}}{\sim} v_k = y\}$ to emphasize that l_{xy} is a path in $\vec{G}$. We define $V^s(l_{xy}) = \{v \in V : \{u \to v \to w\} \subseteq l_{xy} \text{ or } \{u \leftarrow v \leftarrow w\} \subseteq l_{xy}\}$, $V^d(l_{xy}) = \{v \in V : \{u \leftarrow v \to w\} \subseteq l_{xy}\}$ and $V^c(l_{xy}) = \{v \in V : \{u \to v \leftarrow w\} \subseteq l_{xy}\}$. We state that $V^s(l_{xy}), V^d(l_{xy})$, and $V^c(l_{xy})$ are the sets of the *serial, diverging*, and *converging connection vertices*, respectively, on the path l_{xy}. We use $V^o(l_{xy}) = V^s(l_{xy}) \cup V^d(l_{xy}) \cup V^c(l_{xy})$ to represent the set of the *interior vertices* of l_{xy}, and $V(l_{xy})$ the set of all the vertices of l_{xy}, i.e., $V(l_{xy}) = V^o(l_{xy}) \cup \{x, y\}$. A path l_{uv} is a *v-path* if $V^c(l_{uv}) \neq \emptyset$; otherwise we call it a *trek* (Sullivant et al., 2010). For a path $l_{xy} = (x = v_1, v_2, \ldots, v_{k-1}, v_k = y)$ in $\vec{G}$, if it satisfies $v_i \to v_{i+1}$ for $i = 1, 2, \ldots, k-1$, we refer to it a *directed path* or an *s-path* from x to y in $\vec{G}$, denoted by $\vec{l}_{xy}$. The path l_{xy} is called an *o-o path* if $x \to v_2$ and $v_{k-1} \leftarrow y$. Here "o" represents the outgoing direction of x or y on the path l_{xy}.

We define the sets of *parents, children, ancestors, descendants* and *non-descendants* of a vertex u in $\vec{G}$ as follows

$$\begin{aligned}
&\mathbf{pa}_{\vec{G}}(u) = \{w \in V \setminus u : (w, u) \in \vec{E}\}, && \mathbf{ch}_{\vec{G}}(u) = \{w \in V \setminus u : (u, w) \in \vec{E}\},\\
&\mathbf{an}_{\vec{G}}(u) = \{w \in V \setminus u : \exists\ \vec{l}_{wu} \text{ in } \vec{G}\}, && \mathbf{de}_{\vec{G}}(u) = \{w \in V \setminus u : \exists\ \vec{l}_{uw} \text{ in } \vec{G}\},\\
&\mathbf{nd}_{\vec{G}}(u) = \{w \in V \setminus u : w \notin \mathbf{de}_{\vec{G}}(u)\}.
\end{aligned}$$

For a subset $A \subseteq V$, we define the sets of parents and ancestors of A in $\vec{G}$ to be $\mathbf{pa}_{\vec{G}}(A) = \bigcup_{u \in A} \mathbf{pa}_{\vec{G}}(u) \setminus A$ and $\mathbf{an}_{\vec{G}}(A) = \bigcup_{u \in A} \mathbf{an}_{\vec{G}}(u) \setminus A$, respectively. Further, we use $\mathbf{An}_{\vec{G}}(A)$ to denote $\mathbf{an}_{\vec{G}}(A) \cup A$. We say that a vertex u is a *root vertex* in $\vec{G}$ if $\mathbf{pa}_{\vec{G}}(u) = \emptyset$, and a set of vertices A is a *root set* in $\vec{G}$ if $\mathbf{pa}_{\vec{G}}(A) = \emptyset$.

3. Bayesian Network, Its Minimal $\mathcal{I}$-map and Conditional Model

Recall that a BN $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$ consists of a DAG $\vec{G}$ and a distribution family $\mathcal{P}(\vec{G})$ that is compatible with $\vec{G}$ (see (1)). Suppose A, B, and S as pairwise disjoint subsets of V, the two sets A and B are *d-separated* by S in $\vec{G}$ if and only if every path l_{ab} connecting any $a (\in A)$ and any $b (\in B)$ in $\vec{G}$ is *blocked* by S, i.e., $(V^s(l_{ab}) \cup V^d(l_{ab})) \cap S \neq \emptyset$ or $V^c(l_{xy}) \not\subseteq \mathbf{An}_{\vec{G}}(S)$.

For any distribution P over X_V, we define

$$\mathcal{I}(P) = \{\langle U, W | S\rangle : X_U \perp\!\!\!\perp X_W | X_S [P] \text{ with pairwise disjoint subsets } U, W, S \subseteq V\}$$

as the set of triplets $\langle U, W|S\rangle$ such that X_U and X_W are conditionally independent given X_S under P. If $P \in \mathcal{P}(\vec{G})$, the corresponding conditional independence relations in P hold whenever the d-separation assertions are true in $\vec{G}$.

Based on the d-separation assertions, we define the *independence model* induced by the DAG $\vec{G}$ as follows:

Definition 1 (Independence model) *The independence model induced by $\vec{G}$ is the set*

$$\mathcal{I}(\vec{G}) = \{\langle U, W|S\rangle : U \perp\!\!\!\perp W | S[\vec{G}] \text{ with pairwise disjoint subsets } U, W, S \subseteq V\},$$

where $U \perp\!\!\!\perp W|S[\vec{G}]$ denotes that U, W are d-separated by S in $\vec{G}$.

For a BN $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$ and $P \in \mathcal{P}(\vec{G})$, it holds that $\mathcal{I}(\vec{G}) \subseteq \mathcal{I}(P)$ (Lauritzen, 1996). This means that for any $P \in \mathcal{P}(\vec{G})$, we can read the conditional independence relations in P from the d-separations in $\vec{G}$. Consequently, the DAG $\vec{G}$ is called an *$\mathcal{I}$-map* of $\mathcal{P}(\vec{G})$. In fact, $\vec{G}$ is recognized as a *minimal $\mathcal{I}$-map* for the entire distribution family $\mathcal{P}(\vec{G})$ (Pearl et al., 1989).

Definition 2 (Pearl, 1988) *Let $\vec{G} = (V, \vec{E})$ be a DAG and $\mathcal{P}$ be a family of probability distributions over X_V. The graph $\vec{G}$ is called a minimal $\mathcal{I}$-map of $\mathcal{P}$ if $\vec{G}$ is an $\mathcal{I}$-map, though $\vec{G} - (x, y)$ is not an $\mathcal{I}$-map of $\mathcal{P}$ for any $(x, y) \in \vec{E}$.*

The property of $\vec{G}$ being a minimal $\mathcal{I}$-map for $\mathcal{P}(\vec{G})$ implies that the removal of any edge from $\vec{G}$ induces a conditional independence relation that does not align with at least one distribution within $\mathcal{P}(\vec{G})$. Consequently, $\mathcal{I}(\vec{G})$ serves as an optimal representation of the independence structure encapsulated by $\mathcal{P}(\vec{G})$. When a specific distribution $P^* \in \mathcal{P}(\vec{G})$ exists such that $\mathcal{I}(\vec{G}) = \mathcal{I}(P^*)$, $\mathcal{P}(\vec{G})$ is known to be *faithful* to $\vec{G}$. This faithfulness indicates a precise correspondence between the structural dependencies in $\vec{G}$ and the dependence relations observed in some $P^* \in \mathcal{P}(\vec{G})$.

When a BN $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$ is conditioned on a subvector X_C of X_V, both its distribution model $\mathcal{P}(\vec{G})$ and independence model $\mathcal{I}(\vec{G})$ are altered. To study the changes due to conditioning, we define their conditional counterparts.

Definition 3 (Conditional distribution model) *Let $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$ be a BN with a DAG $\vec{G} = (V, \vec{E})$, and let C be a subset of V. The conditional distribution model (CDM) $\mathcal{P}(\vec{G})^C$ of $\mathcal{P}(\vec{G})$ consists of all conditional distribution $P(X_{V\setminus C}|X_C = x_C)$, where the distribution $P(X_{V\setminus C}|X_C)$ is determined by the following relation*

$$\int_F P(X_{V\setminus C} \in E|x_C)dP(x_C) = P(X_{V\setminus C} \in E, X_C \in F) \text{ for some } P \in \mathcal{P}(\vec{G}),$$

for any measurable subsets E and F of $\mathcal{X}_{V\setminus C}$ and $\mathcal{X}_C$, respectively.

In the rest of this work, we use R to represent $V \setminus C$, i.e., $R \equiv V \setminus C$. For two disjoint subsets $A, B \subseteq R$, we will use the shorthand notation $P^C(X_A|X_B) \equiv P(X_A|X_B, X_C = x_C)$ to represent the conditional distribution $P(X_A|X_B, X_C = x_C)$ for some fixed $x_C \in \mathcal{X}_C$, if no confusion arises. Specifically, $P^C(X_R) \equiv P(X_R|X_C = x_C)$.

Definition 4 (Conditional independence model) *Let $\vec{G} = (V, \vec{E})$ be a DAG and C be a subset of V. We define $\mathcal{I}(\vec{G})^C$ as the conditional independence model (CIM) of the independence model $\mathcal{I}(\vec{G})$ under conditioning on C, i.e.,*

$$\mathcal{I}(\vec{G})^C = \big\{\langle U, W|S\rangle : U \perp\!\!\!\perp W|(S, C)[\vec{G}] \text{ with pairwise disjoint subsets } U, W, S \subseteq R\big\}.$$

We can compare $\mathcal{I}(\vec{G})^C$ with $\mathcal{I}(\vec{G}_R)$. The latter is the (*unconditional*) independence model induced by the subgraph $\vec{G}_R$. Obviously, it holds that $\mathcal{I}(\vec{G})^C \subseteq \mathcal{I}(\vec{G}_R)$, while the converse is not true. Meanwhile, we can consider the relationship between the CDM $\mathcal{P}(\vec{G})^C$ and the CIM $\mathcal{I}(\vec{G})^C$, which is entailed by Proposition 5 below.

Proposition 5 (Lossless representation) *Let $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$ be a BN with a DAG $\vec{G} = (V, \vec{E})$. Suppose that $\mathcal{P}(\vec{G})$ is faithful to $\vec{G}$. For any subset C of V, we have*

(a) $\mathcal{I}(\vec{G})^C \subseteq \mathcal{I}(P^C)$ for all $P^C \in \mathcal{P}(\vec{G})^C$;

(b) there exists a conditional distribution $P^C \in \mathcal{P}(\vec{G})^C$ such that $\mathcal{I}(\vec{G})^C = \mathcal{I}(P^C)$.

Remark 6 *The concept of faithfulness is originally defined for the purpose of causal inference (Pearl, 1988; Spirtes et al., 2001), which is a key property in both structure learning and estimation procedures for graphical models. It is known that there are distributions that are faithful to the DAG concerned for discrete and normal distributions according to Meek (1995). In general, we can also determine the validity of the faithfulness assumption to a DAG for an arbitrary distribution by applying Corollary 34 in Sadeghi (2017).*

When $\mathcal{P}(\vec{G})$ is faithful to $\vec{G}$, Proposition 5 implies that the conditional independence model $\mathcal{I}(\vec{G})^C$ continues to maintain the property of being an optimal representation of the conditional independence relations described by distributions within $\mathcal{P}(\vec{G})^C$. This relationship parallels that of their unconditional counterparts, where $\mathcal{I}(\vec{G})$ acts as the optimal representation for the conditional independence structure of distributions within $\mathcal{P}(\vec{G})$.

Although $\mathcal{I}(\vec{G})^C$ is an optimal representation of the conditional independence structure of the distributions within $\mathcal{P}(\vec{G})^C$, it may not be intuitive for us to understand $\mathcal{I}(\vec{G})^C$ since its d-separation assertions may not be readily read off from some DAG over R. One may attempt to use $\vec{G}_R$ as a minimal $\mathcal{I}$-map of $\mathcal{P}(\vec{G})^C$. However, it is possible that $\mathcal{I}(\vec{G})^C \subseteq \mathcal{I}(\vec{G}_R)$ with strict inclusion, which means that the subgraph $\vec{G}_R$ can fail to serve as a minimal $\mathcal{I}$-map of $\mathcal{P}(\vec{G})^C$ in general. We explore the conditions under which $\vec{G}_R$ maintains a minimal $\mathcal{I}$-map of $\mathcal{P}(\vec{G})^C$ in Section 4. Additionally, when $\vec{G}_R$ fails to meet the criteria of being a minimal $\mathcal{I}$-map for $\mathcal{P}(\vec{G})^C$, we investigate the modifications required to satisfy this criterion in Section 5.

4. Closure of Bayesian Network under Conditioning

In this section, we begin by introducing a few concepts of closure for a BN under conditioning, and then explore their equivalent characterizations. An algorithm to determine their closure is proposed and a special case is studied when the conditioning set C is a root set.

We have discussed that, for a BN $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$ conditional on $X_C = x_C$ for some fixed x_C, a distribution within $\mathcal{P}(\vec{G})^C$ may no longer be compatible with the corresponding subgraph $\vec{G}_R$. We let $\mathcal{P}^C(\vec{G}_R)$ be the set of conditional distributions that are compatible with $\vec{G}_R$. In general, it holds that $\mathcal{P}^C(\vec{G}_R) \subseteq \mathcal{P}(\vec{G})^C$ (see Proposition 37). If the equality holds, we say that $\mathcal{P}(\vec{G})$ is closed under conditioning on C.

Definition 7 (Closure of a distribution model) *Let $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$ be a BN with a DAG $\vec{G} = (V, \vec{E})$, and C be a subset of V. The distribution model $\mathcal{P}(\vec{G})$ is closed under conditioning on C if $\mathcal{P}(\vec{G})^C = \mathcal{P}^C(\vec{G}_R)$.*

The concept of closure has several implications. First, we can see that $(\vec{G}_R, \mathcal{P}^C(\vec{G}_R))$ is a Bayesian network by the definition of a BN. If $\mathcal{P}(\vec{G})$ is closed under conditioning, this directly implies that $(\vec{G}_R, \mathcal{P}(\vec{G})^C)$ also forms a BN. For scenarios such as that in Example 1, if we know the closure property holds, we can safely conclude that there are no structural changes among the remaining vertices caused by the operation of conditioning. Second, any conditional distribution $P^C(x_R) \in \mathcal{P}(\vec{G})^C$ can be represented as

$$P^C(x_R) = \prod_{w \in R} P^C\big(x_w | x_{\mathbf{pa}_{\vec{G}_R}(w)}\big), \tag{4}$$

where the product is factored exactly according to the subgraph $\vec{G}_R$. As explained in Section 1, the factorization (4) facilitates computation and parameter estimation (see Example 1 and the discussion afterward).

We can also define the concept of closure for the conditional independence model $\mathcal{I}(\vec{G})^C$.

Definition 8 (Closure of an independence model) *Let $\vec{G} = (V, \vec{E})$ be a DAG and C be a subset of V. We define that the independence model $\mathcal{I}(\vec{G})$ is closed under conditioning on C if $\mathcal{I}(\vec{G})^C = \mathcal{I}(\vec{G}_R)$.*

In Section 3, we stated that it holds that $\mathcal{I}(\vec{G})^C \subseteq \mathcal{I}(\vec{G}_R)$ in general. The closure of an independence model implies that the d-separation assertions within $\mathcal{I}(\vec{G})^C$ are exactly those described by $\vec{G}_R$. By Proposition 5, under faithfulness, the closure of the independence model $\mathcal{I}(\vec{G})$ under conditioning implies that $\vec{G}_R$ is a minimal $\mathcal{I}$-map of $\mathcal{P}(\vec{G})^C$.

A BN is closed under conditioning on C if both the conditional distribution model and the conditional independence model are closed under conditioning.

Definition 9 (Closure of a Bayesian network) *Let $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$ be a BN with a DAG $\vec{G} = (V, \vec{E})$, and C be a subset of V. The BN $\mathcal{B}$ is closed under conditioning on C if both $\mathcal{P}(\vec{G})^C = \mathcal{P}^C(\vec{G}_R)$ and $\mathcal{I}(\vec{G})^C = \mathcal{I}(\vec{G}_R)$ hold.*

For a BN, if $\mathcal{P}(\vec{G})$ is faithful to $\vec{G}$, the closure of $\mathcal{P}(\vec{G})$ and the closure of $\mathcal{I}(\vec{G})$ under conditioning are equivalent.

Theorem 10 (Transformation theorem I) *Let $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$ be a BN with a DAG $\vec{G} = (V, \vec{E})$ and C be a subset of V. Suppose that $\mathcal{P}(\vec{G})$ is faithful to $\vec{G}$. Then $\mathcal{P}(\vec{G})$ is closed under conditioning on C if and only if $\mathcal{I}(\vec{G})$ is closed under conditioning on C, i.e.,*

$$\mathcal{P}(\vec{G})^C = \mathcal{P}^C(\vec{G}_R) \Leftrightarrow \mathcal{I}(\vec{G})^C = \mathcal{I}(\vec{G}_R).$$

Remark 11 *If $\mathcal{I}(\vec{G})^C = \mathcal{I}(\vec{G}_R)$, it becomes evident that $\mathcal{P}(\vec{G})^C$ is a subset of $\mathcal{P}^C(\vec{G}_R)$. This is because $\vec{G}_R$ serves as an $\mathcal{I}$-map for $\mathcal{P}(\vec{G})^C$. Furthermore, let $N_C = (C, \emptyset)$ be a null graph over C and $\mathcal{P}_C(N_C)$ be the set of distributions that are compatible with N_C. Then we can choose a joint distribution as $P(x_R)P(x_C)$ in $\mathcal{P}(\vec{G})$, where $P(x_R) \in \mathcal{P}^C(\vec{G}_R)$ and $P(x_C) \in \mathcal{P}_C(N_C)$. Because any probability distribution within $\mathcal{P}^C(\vec{G}_R)$ can be achieved by conditioning on any value of X_C, it follows that $\mathcal{P}^C(\vec{G}_R)$ is contained in $\mathcal{P}(\vec{G})^C$.*

Based on Theorem 10, rather than examining the distributions in $\mathcal{P}(\vec{G})$, we can analyze the graph $\vec{G}$ to verify the closure property under conditioning, which is more accessible. Building on this insight, in the subsequent subsection, we develop several equivalent conditions and propose an algorithm to aid practitioners in checking the closure property of a BN.

4.1 Characterization by C-configuration

We continue to determine the equivalent conditions for $\mathcal{I}(\vec{G})^C = \mathcal{I}(\vec{G}_R)$. Recall in Example 1 that, the non-closure of a BN can occur if the conditional vertex is a collider, i.e., it is an interior vertex of a v-structure. How can we generalize these well-known results to general settings where the conditioning set C contains multiple vertices? Our investigation examines the set $\mathcal{V}(\vec{G})$ of all v-structures in $\vec{G}$, defined as $\mathcal{V}(\vec{G}) = \big\{\{x \to w \leftarrow y\} \subseteq \vec{G} : x \overset{\vec{G}}{\nsim} y\big\}$. Our key observation is that not only do v-structures with endpoints in $V \setminus C$ matter, but all v-structures with their colliders in $\mathbf{An}_{\vec{G}}(C)$ also play an important role. Specifically, we refer to these v-structures as C-configurations.

Definition 12 (C-configuration) *The conditioning configuration (C-configuration) with respect to (w.r.t.) C in $\vec{G}$ is a v-structure in the set $\mathcal{AV}_{\vec{G}}(C)$, which is defined as*

$$\mathcal{AV}_{\vec{G}}(C) = \big\{\{x \to w \leftarrow y\} \in \mathcal{V}(\vec{G}) : x, y \in V \setminus C, w \in \mathbf{An}_{\vec{G}}(C)\big\}.$$

Note that in the above, w is in $\mathbf{An}_{\vec{G}}(C)$ whereas x, y are out of C. It is reasonable to consider x, y in the outside of C, since conditioning on C can only change the structure between the vertex pairs in $V \setminus C$. Our next result verifies the equivalence between $\mathcal{AV}_{\vec{G}}(C)$ being empty and the closure of the independence model $\mathcal{I}(\vec{G})$ under conditioning.

Proposition 13 *Let $\vec{G} = (V, \vec{E})$ be a DAG and C be a subset of V. The following statements are equivalent:*

(a) $\mathcal{I}(\vec{G})^C = \mathcal{I}(\vec{G}_R)$;

(b) for any $x, y \in R$ with $x \overset{\vec{G}}{\nsim} y$, there exists an $S_0 \subseteq R \setminus \{x, y\}$ such that $x \perp\!\!\!\perp y | (S_0, C)[\vec{G}]$;

(c) $\mathcal{AV}_{\vec{G}}(C) = \emptyset$, i.e., there are no C-configurations in $\vec{G}$.

It should be noted that the equivalence of (a) and (c) in Proposition 13 can be implied from the existing work, see Borboudakis and Tsamardinos (2015), Richardson and Spirtes (2002). The condition (b) is a characterization of (a) from the local d-separation viewpoint of nonadjacent vertex pairs in $\vec{G}_R$.

Remark 14 *It enjoys a clear rationale for the equivalence between $\mathcal{AV}_{\vec{G}}(C)$ being empty and the closure of the independence model $\mathcal{I}(\vec{G})$ under conditioning. Suppose that there exists a C-configuration $\{x \to w \leftarrow y\}$ in $\vec{G}$. Conditioning on C and any subset $Z \subseteq V \setminus (C \cup \{x, y\})$, the nonadjacent vertex pair $\{x, y\}$ cannot be d-separated in $\vec{G}$. This implies $\langle x, y | Z\rangle \notin \mathcal{I}(\vec{G})^C$ for any $Z \subseteq V \setminus (C \cup \{x, y\})$. On the other hand, there exists a subset $Z_0 \subseteq R \setminus \{x, y\}$ (which is possibly empty) such that the nonadjacent vertex pair $\{x, y\}$ is d-separated in $\vec{G}_R$, which means $\langle x, y | Z_0\rangle \in \mathcal{I}(\vec{G}_R)$. The above reasoning indicates that, the existence of C-configurations in $\vec{G}$ leads to $\mathcal{I}(\vec{G})^C \neq \mathcal{I}(\vec{G}_R)$.*

Remark 15 *Proposition 13 indicates that the closure of an independence model can be verified by examining whether $\mathcal{AV}_{\vec{G}}(C) = \emptyset$ holds. According to Theorem 10, when the faithful condition holds, the closure of an independence model implies the closure of the corresponding BN under conditioning.*

Based on Proposition 13 (c), we develop Algorithm 1 to check the closure of an independence model under conditioning. Algorithm 1 iterates through all the vertices in $\mathcal{AV}_{\vec{G}}(C)$

and checks whether they form a v-structure. The algorithm has a complexity at most $O(m^2|\mathbf{An}_{\vec{G}}(C)|)$, where m is the maximal number of parents of every vertex in $\mathbf{An}_{\vec{G}}(C)$. When $\vec{G}$ is a causal DAG, the number of parents of each vertex is usually small (therefore, k is small) and our algorithm is linear in the number of ancestors of the conditioning set. In general, the complexity of our algorithm is at most cubic in the number of ancestors of the conditioning set.

Algorithm 1 Check the closure of an independence model under conditioning

```
Input: A DAG $\vec{G} = (V, \vec{E})$, a topological order $\alpha$ of $\vec{G}$, and a subset $C$ of $V$.
Output: $Z = 0$ if $\mathcal{I}(\vec{G})$ is closed under conditioning on $C$; $Z = 1$ otherwise.
1: Initialize $A = \mathbf{An}_{\vec{G}}(C)$ and $Z = 0$;
2: for $u \in A$ do
3:     if $\mathbf{pa}_{\vec{G}}(u) \setminus C$ is not complete then
4:         $Z = 1$ and break;
5:     end if
6: end for
7: return $Z$.
```

4.2 Conditioning on a Root Set

For the simple case where C is a root set in the DAG of a BN $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$, it can be demonstrated that the closure of $\mathcal{B}$ under conditioning on C, i.e., $\mathcal{I}(\vec{G})$ and $\mathcal{P}(\vec{G})$ are both closed under conditioning on C. This result is presented in Proposition 16.

Proposition 16 *Let $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$ be a BN with a DAG $\vec{G} = (V, \vec{E})$ and C be a subset of V. If C is a root set in $\vec{G}$, then*

(a) $\mathcal{I}(\vec{G})^C = \mathcal{I}(\vec{G}_R)$;

(b) $\mathcal{P}(\vec{G})^C = \mathcal{P}^C(\vec{G}_R)$.

It is worth noting that (a) in Proposition 16 can be derived directly from Proposition 13; and if the faithfulness assumption is additionally imposed, (b) can be derived from Theorem 10. We employ an alternative technical proof approach to remove the necessity of the faithfulness assumption in Proposition 16.

Note that, when C is a root set in $\vec{G}$, the corresponding conditional model is called a conditional BN in Koller and Friedman (2009). Conditional BNs are useful in several settings and can be used to define encapsulated conditional probability distributions, which can significantly simplify the model from a cognitive perspective (Koller and Friedman, 2009; Langseth and Bangsø, 2001; Marella and Vicard, 2013; Mortera et al., 2013; Srinivas, 1994).

5. Minimal Filling Edge Set for Non-Closure Bayesian Network

In most practical applications, Bayesian networks are not closed under conditioning. In scenarios where the closure property is not satisfied, $\vec{G}_R$ is no longer a minimal $\mathcal{I}$-map of $\mathcal{P}(\vec{G})^C$. Our approach involves adding the fewest possible directed edges to $\vec{G}_R$ to transform it into a DAG minimal $\mathcal{I}$-map of $\mathcal{P}(\vec{G})^C$. In this section, we start with an illustrative example to show how edges can be added. In Section 5.2, we formalize this concept by defining the minimal filling edge set and demonstrate that this set leads to the DAG minimal $\mathcal{I}$-map. In Section 5.3, we discuss the benefits of the proposed DAG minimal $\mathcal{I}$-map for conditional inference of a BN. Some characterizations of the minimal filling edge set are presented in Section 5.4. Finally, an efficient algorithm, termed reverse order search (ROS) algorithm, is proposed to find the DAG minimal filling edge set in Section 5.5.

5.1 An Example for the Main Idea of Finding a Minimal $\mathcal{I}$-map

We begin with a simple example to illustrate our main idea of finding a DAG minimal $\mathcal{I}$-map by adding new edges.

Example 2 *Consider a causal BN $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$ with a causal DAG $\vec{G} = (V, \vec{E})$, as shown in Figure 2. Let $C = \{e\} \subseteq V$. Suppose that we have two BNs $(\vec{G}, \mathcal{P}(\vec{G}))$ and $(\vec{G}_R, \mathcal{P}^C(\vec{G}_R))$. Then, we have the independence relation $X_{\{a,c\}} \perp\!\!\!\perp X_{\{b,d\}}$ in the joint probability distribution P, for any $P \in \mathcal{P}(\vec{G})$. However, given $X_C = x_C$, we have $X_c \not\perp\!\!\!\perp X_d$ in some conditional distribution $P^C \in \mathcal{P}(\vec{G})^C$. This implies that $\vec{G}_R$ is not an $\mathcal{I}$-map of $\mathcal{P}(\vec{G})^C$. To maintain $\vec{G}_R$ as an $\mathcal{I}$-map of $\mathcal{P}(\vec{G})^C$, we need to add a directed edge (c, d) into $\vec{G}$, while keeping the existing directed edges unchanged. However, this operation introduces a new incompatibility $X_b \not\perp\!\!\!\perp X_c[P^C]$ for some $P^C \in \mathcal{P}(\vec{G} + (c, d))^C$. Therefore, it is necessary to add a directed edge (b, c) into $\vec{G} + (c, d)$. Similarly, we need to add a directed edge (a, b) into $\vec{G} + (c, d) + (b, c)$. Let $\vec{G}' = \vec{G} + (c, d) + (b, c) + (a, b)$. We can verify that $\vec{G}'_R$ is a minimal $\mathcal{I}$-map of $\mathcal{P}(\vec{G})^C$ under the faithfulness assumption.*

The procedure in Example 2 is in fact the process of eliminating C-configurations in $\vec{G}$. Recall from Proposition 13 that $\mathcal{I}(\vec{G})^C = \mathcal{I}(\vec{G}_R)$ if and only if $\mathcal{AV}_{\vec{G}}(C) = \emptyset$. This means that the existence of C-configurations in $\vec{G}$ leads to the "wrong representation" of the conditional independence relations for some conditional distribution $P^C \in \mathcal{P}(\vec{G})^C$. To eliminate the "wrong representation", we add edges between the endpoints of every C-configuration in $\vec{G}$. Given a topological order α, the edge directions can be assigned such that each edge points toward the vertex with a higher topological order. Meanwhile, after each directed edge being added into $\vec{G}$, there may emerge new C-configurations. Thus, we need to add more edges recursively until there are no C-configurations in the final DAG.

5.2 Minimal Filling Edge Set

Based on the discussion in the previous subsection, we can see that a set of edges can be added to eliminate C-configurations in the final DAG. We refer to the set of minimally added edges as the minimal filling edge set. In the following, we formalize the concept of minimal filling edge set and then examine its relationship with minimal $\mathcal{I}$-map.

Given a DAG $\vec{G}$ with a topological order α, we define $\vec{E}_c^\alpha(R) = \{x \overset{\alpha}{\sim} y : x, y \in R, x \overset{\vec{G}}{\nsim} y\}$ to be the complementary set of $\vec{E}(\vec{G}_R)$ with respect to a complete graph $\vec{K}_R$ including $\vec{G}_R$, i.e., $\vec{K}_R = \vec{G}_R + \vec{E}_c^\alpha(R)$.

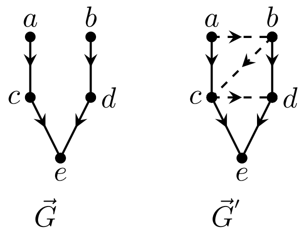


Figure 2: In the left part, $\vec{G}$ is a causal DAG over $V = \{a, b, c, d, e\}$ with a topological order α such that $\alpha(a) < \alpha(b) < \cdots < \alpha(e)$, and $C = \{e\} \subseteq V$. Under faithfulness, the DAG $\vec{G}'_R$ (in the right part) is obtained by adding edges to $\vec{G}_R$ such that $\vec{G}'_R$ is a minimal $\mathcal{I}$-map of $\mathcal{P}(\vec{G})^C$ while keeping the existing directed edges in $\vec{G}$ unchanged.

Definition 17 (Minimal filling edge set) *Let $\vec{G} = (V, \vec{E})$ be a DAG with a topological order α, and C be a subset of V. The edge set $\triangle^{\alpha}_{\vec{G}}(R)$ ($\subseteq \vec{E}^{\alpha}_{c}(R)$) is called a minimal filling edge set of $\vec{G}_R$ with respect to (w.r.t.) α if it satisfies that for any nonadjacent vertices x, y in $\vec{G}$, $x \overset{\alpha}{\sim} y \in \triangle^{\alpha}_{\vec{G}}(R) \Leftrightarrow x, y$ are the endpoints of a C-configuration w.r.t. C in $\vec{G} + \triangle^{\alpha}_{\vec{G}}(R) - x \overset{\alpha}{\sim} y$.*

Definition 17 formalizes the added edges in Example 2. Our intention is to "moralize" the parents by iteratively adding directed edges between the two endpoints of the C-configurations under the acyclic constraint, until no C-configurations remain in the final DAG $\vec{G} + \triangle^{\alpha}_{\vec{G}}(R)$. Indeed, we can verify the close relationship between the minimal filling edge set and minimal $\mathcal{I}$-map.

Theorem 18 (Transformation theorem II) *Let $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$ be a BN with a DAG $\vec{G} = (V, \vec{E})$ and C be a subset of V. Suppose that $\mathcal{P}(\vec{G})$ is faithful to $\vec{G}$ and $\triangle^{\alpha}_{\vec{G}}(R) \subseteq \vec{E}^{\alpha}_{c}(R)$, where α is a topological order of $\vec{G}$. Then $\vec{G}_R + \triangle^{\alpha}_{\vec{G}}(R)$ is a minimal $\mathcal{I}$-map of $\mathcal{P}(\vec{G})^C$ if and only if $\triangle^{\alpha}_{\vec{G}}(R)$ is a minimal filling edge set of $\vec{G}_R$ w.r.t. α.*

Theorem 18 presents an equivalent characterization for a DAG minimal $\mathcal{I}$-map of $\mathcal{P}(\vec{G})^C$. This implies that, under faithfulness, finding a DAG minimal $\mathcal{I}$-map for $\mathcal{P}(\vec{G})^C$ by adding edges in $\vec{G}_R$ is equivalent to finding a minimal filling edge set $\triangle^{\alpha}_{\vec{G}}(R)$. This result is of significance. Finding a DAG-minimal $\mathcal{I}$-map for the conditional distribution model $\mathcal{P}(\vec{G})^C$ is challenging, as it requires analyzing all conditional distributions within $\mathcal{P}(\vec{G})^C$ and systematically mapping their independence relations to a corresponding graph structure. On the other hand, finding a minimal filling edge set is much more intuitive and operable by looking into the graph structure alone. Theorem 18 establishes the link between the task of finding a minimal $\mathcal{I}$-map and that of finding a minimal filling edge set.

5.3 Benefits of DAG Minimal $\mathcal{I}$-map

In Section 3, we discuss the various advantages of BNs being closed under conditioning. When a BN is *not* closed under conditioning, our proposal is to identify a minimal filling edge set $\triangle^{\alpha}_{\vec{G}}(R)$ and to construct $\vec{G}^{\alpha}_{R} = \vec{G}_R + \triangle^{\alpha}_{\vec{G}}(R)$ as a DAG minimal $\mathcal{I}$-map for $\mathcal{P}(\vec{G})^C$. This approach offers practical benefits similar to those observed when BNs are closed under conditioning.

First, even if the original BN $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$ is not closed under conditioning, we can shift our analysis to the BN $\mathcal{B}^{\alpha} = (\vec{G}^{\alpha}, \mathcal{P}(\vec{G}^{\alpha}))$, where $\vec{G}^{\alpha} = \vec{G} + \triangle^{\alpha}_{\vec{G}}(R)$. In fact, we know that $\vec{G}^{\alpha}$ is a DAG which has fewest edges added to satisfy that $\vec{G} \subseteq \vec{G}^{\alpha}$ and that $\mathcal{I}(\vec{G}^{\alpha})$ is closed under conditioning on C (see Corollary 43 in Appendix D.1). Since $\mathcal{P}(\vec{G})^C \subseteq \mathcal{P}(\vec{G}^{\alpha})^C = \mathcal{P}^C(\vec{G}^{\alpha}_R)$, practitioners can employ existing efficient inference procedure for the BN $(\vec{G}^{\alpha}_R, \mathcal{P}^C(\vec{G}^{\alpha}_R))$, which are applicable to the conditional model $\mathcal{P}(\vec{G})^C$ due to the inclusion relationship. Furthermore, the induced BN $(\vec{G}^{\alpha}_R, \mathcal{P}^C(\vec{G}^{\alpha}_R))$ itself can also be used for the efficient computation of conditional probabilities, as shown in Theorem 19.

In addition, when developing medical diagnostic systems in independently conducted projects or other similar scenarioes, a causal graph is typically constructed based on expert knowledge, and the parameters are obtained from data, textbooks, and expert estimates. As the data in practice is often from a conditional distribution/subpopulation, we are interested in profiting from the subpopulation data for parameter estimation. The BN $\mathcal{B}^{\alpha}_R = (\vec{G}^{\alpha}_R, \mathcal{P}^C(\vec{G}^{\alpha}_R))$ offers a powerful framework for integrating evidence from different sources (Druzdzel and Díez, 2003; Flores et al., 2011; Lappenschaar et al., 2013; Yet et al., 2014). This capability is crucial for improving the accuracy of parameter estimation by combining data from different sources, especially for the cases where each source (subpopulation) has a small sample size.

The next result indicates that: 1) the conditional probabilities within $\mathcal{P}(\vec{G})^C$ can be factored as a product of the local conditional probabilities according to the graph structure of $\vec{G}^{\alpha}_R$, and 2) how the minimal $\mathcal{I}$-map $\vec{G}^{\alpha}_R$ provides graphical criteria to determine whether knowledge or data sets from different sources can be safely combined into the conditional distribution model $\mathcal{P}^C(\vec{G}^{\alpha}_R)$.

Theorem 19 *Let $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$ be a BN with a DAG $\vec{G} = (V, \vec{E})$ and C be a subset of V. Suppose that $\mathcal{P}(\vec{G})$ is faithful to $\vec{G}$. Let $\vec{G}^{\alpha} = \vec{G} + \triangle^{\alpha}_{\vec{G}}(R)$ and $\vec{G}^{\alpha}_R = \vec{G}_R + \triangle^{\alpha}_{\vec{G}}(R)$, where α is a topological order of $\vec{G}$ and $\triangle^{\alpha}_{\vec{G}}(R)$ is the minimal filling edge set of $\vec{G}_R$ w.r.t. α. Then we find*

(a) $\forall P^C \in \mathcal{P}(\vec{G})^C$, it holds that $P^C(x_R) = \prod_{w \in R} P^C(x_w | x_{\mathbf{pa}_{\vec{G}^{\alpha}_R}(w)})$;

(b) $\forall P^C \in \mathcal{P}(\vec{G})^C$, it follows that

$$P^C(x_R) = \phi^C(x_A) \times \prod_{w \in R \setminus (\mathbf{an}_{\vec{G}}(C) \cup \mathbf{ch}_{\vec{G}}(C))} P(x_w | x_{\mathbf{pa}_{\vec{G}_R}(w)}),$$

where $\phi^C(x_A) = \prod_{w \in \mathbf{an}_{\vec{G}}(C) \cup \mathbf{ch}_{\vec{G}}(C)} P(x_w | x_{\mathbf{pa}_{\vec{G}^{\alpha}}(w)}, x_{C \cap \mathbf{de}^{o}_{\vec{G}^{\alpha}}(w)})$ and $\mathbf{de}^{o}_{\vec{G}^{\alpha}}(w) = \{u \in V \setminus w : \exists\, \vec{l}_{wu} \subseteq \vec{G}^{\alpha}, s.t.\ V^{o}(\vec{l}_{wu}) \cap C = \emptyset\}$.

Theorem 19 provides valuable insights by using the structural information of conditional distributions. First, result (a) shows that one can obtain the conditional distribution of interest without relying on the joint distribution. This is important as what we are interested in is conditional probability query, rather than the joint probability query.

Result (b) of Theorem 19 provides graphical criteria for determining whether knowledge or data sets from different sources can be safely combined into the conditional distribution model $\mathcal{P}^C(\vec{G}_R^\alpha)$. This capability is crucial for enhancing the efficiency of model parameter estimations across diverse applications. Specifically, the local conditional parameters in $\{P(x_w|x_{\mathbf{pa}_{\vec{G}_R}(w)}) : w \in R \setminus (\mathbf{an}_{\vec{G}}(C) \cup \mathbf{ch}_{\vec{G}}(C))\}$ are not affected by conditioning on X_C and can be estimated by combining conditional data sets that share same structure but have different setting values of conditional random vector X_C. Similarly, the local conditional parameters in $\{P(x_w|x_{\mathbf{pa}_{\vec{G}^\alpha}(w)}, x_{C\cap\mathbf{de}^o_{\vec{G}^\alpha}(w)}) : w \in \mathbf{an}_{\vec{G}}(C) \cup \mathbf{ch}_{\vec{G}}(C)\}$ can be estimated by combining conditional data sets that share same structure but have different setting values of $X_{\mathbf{pa}_{\vec{G}_R^\alpha}(w)}$, in the context $X_{C\cap\mathbf{pa}_{\vec{G}^\alpha}(w)} = x_{C\cap\mathbf{pa}_{\vec{G}^\alpha}(w)}$ and $X_{C\cap\mathbf{de}^o_{\vec{G}^\alpha}(w)} = x_{C\cap\mathbf{de}^o_{\vec{G}^\alpha}(w)}$, which still has more samples than that in the context of $X_C = x_C$.

When $\vec{G}$ is a causal DAG, our approach has a particular benefit of retaining the original causal interpretability as much as possible, because we find a DAG minimal $\mathcal{I}$-map of $\mathcal{P}(\vec{G})^C$ under the constraint of keeping the existing directed edges in the original DAG unchanged. For scenarios such as that in Example 1, our approach can help researchers to identify a minimal number of potential structure changes due to selection bias. Our aim is to find a DAG minimal $\mathcal{I}$-map of the conditional model of a causal BN by adding directed edges, and the added edges generally do not have a causal interpretation.

5.4 Characterizing Minimal Filling Edge Set by Terminal Connecting Pair

This section characterizes the minimal filling edge set from the perspective of paths in the graph $\vec{G}$. Our main idea is to develop an operator called the converging vertex sweeping operator, which maps a path into a union of representation paths. Adding the new edges of the representation paths to the original graph helps eliminate v-structures on the graph. By repeatedly applying the operator to a specific set of paths, we identify a set of vertex pairs called terminal connecting pairs. It turns out the terminal connecting pairs are exactly the vertex pairs to be connected for the minimal filling edge set.

5.4.1 Representation path

Consider a DAG $\vec{G} = (V, \vec{E})$ with a topological order α, and a path $l_{xy} = (v_1, v_2, \ldots, v_k)$ in $\vec{G}$. Based on the diverging connection vertices $V^d(l_{xy}) = \{d_1, d_2, \ldots, d_m\}$ along the path l_{xy}, we can split the path l_{xy} into several subpaths:

$$l_{xy} = (x = d_0, \ldots, d_1, \ldots, d_m, \ldots, d_{m+1} = y) = \bigcup_{i=0}^{m} l_{d_i d_{i+1}}, \tag{5}$$

Note that if $V^d(l_{xy}) = \emptyset$, we let $m = 0$. For simplicity, we call a path l_{uv} a *unit path* in $\vec{G}$ if l_{uv} is an s-path or an o-o path with $|V^c(l_{uv})| = 1$ in $\vec{G}$. Obviously, each subpath $l_{d_i d_{i+1}}$ in (5) is a unit path in $\vec{G}$ (for $i \in \{0, 1, 2, \ldots, m\}$). From these discussions, we can see that a path in $\vec{G}$ can be split into several unit paths. For each unit path, we define its representation path as follows.

For a unit path l_{uv} in $\vec{G}$, let

$$\widetilde{V}(l_{uv}) = \begin{cases} V(l_{uv}) \setminus \{u\} & \text{if } \alpha(u) < \alpha(v), \\ V(l_{uv}) \setminus \{v\} & \text{if } \alpha(v) < \alpha(u), \end{cases}$$

and define $w = \arg\min_{z \in \widetilde{V}(l_{uv})} \{\alpha(z)\}$ as the *representation vertex* of l_{uv}. Without loss of generality we assume that $\alpha(u) < \alpha(v)$; otherwise, in the subsequent procedure, we can simply swap the positions of "u" and "v". If w is an interior vertex of l_{uv}, we define the *representation path* l^r_{uv} of the unit path l_{uv} as

$$l^r_{uv} \equiv \vec{l}_{uw} \cup \vec{l}_{wv}.$$

In the above equation, $\vec{l}_{uw}$ and $\vec{l}_{wv}$ are defined as follows. If l_{uv} is an s-path in $\vec{G}$, $\vec{l}_{uw}$ and $\vec{l}_{wv}$ are both simply the original subpaths of l_{uv} split by w. If l_{uv} is not an s-path in $\vec{G}$, $\vec{l}_{uw}$ is exactly the original subpath of l_{uv} and $\vec{l}_{wv}$ is a newly added edge "$w \to v$". Taking the path $l_{v_1v_5} = (v_1, v_3, v_7, v_6, v_5)$ in $\vec{G}$ (as shown in Figure 3) for an example, we can verify that the representation vertex of $l_{v_1v_5}$ is v_3, and hence $l^r_{v_1v_5} = \vec{l}_{v_1v_3} \cup (v_3, v_5) = (v_1, v_3) \cup (v_3, v_5) = (v_1, v_3, v_5)$, where (v_3, v_5) is a newly added edge.

Consider the case when the representation vertex w is one of the two endpoints of the path l_{uv}. Again, without loss of generality, we assume $\alpha(u) < \alpha(v)$ and it follows $w = v$. We define $\vec{l}_{vv} = (v, v)$ as a degenerate directed edge or a degenerate path of length zero. Then we specify that $\vec{l}_{uv} \cup (v, v) \equiv \vec{l}_{uv}$. In this case, $l_{uv} = l^r_{uv} = \vec{l}_{uv}$ represents the original s-path.

5.4.2 Sweeping operator

We now define the converging vertex sweeping operator $\mathcal{T}$ for a path l_{xy} in $\vec{G}$. Based on the path split in (5), $\mathcal{T}$ maps each unit path $l_{d_id_{i+1}}$ to its representation path $l^r_{d_id_{i+1}}$, and $\mathcal{T}(l_{xy})$ is the union of these representation paths.

Definition 20 (Sweeping operator) *Suppose that $\vec{G} = (V, \vec{E})$ is a DAG with a topological order α. For any path $l_{xy} = \bigcup_{i=0}^{m} l_{d_id_{i+1}}$ in $\vec{G}$, we define the (converging vertex) sweeping operator $\mathcal{T}$ as*

$$\mathcal{T}(l_{xy}) = \bigcup_{i=0}^{m} \mathcal{T}(l_{d_id_{i+1}}) = \bigcup_{i=0}^{m} l^r_{d_id_{i+1}}.$$

Furthermore, we define $\mathcal{T}^k(l_{xy}) = \mathcal{T}(\mathcal{T}^{k-1}(l_{xy}))$ for any integer $k \geq 1$, where $\mathcal{T}^0(l_{xy}) = l_{xy}$.

By the definition of the sweeping operator, $\mathcal{T}$ maps a trek l_{xy} to itself, i.e., $\mathcal{T}(l_{xy}) = l_{xy}$ if l_{xy} is not a v-path. In other words, $\mathcal{T}$ is not an identical mapping for v-paths only. For a path l_{xy} in $\vec{G}$, we can see that $V(\mathcal{T}^k(l_{xy})) \subseteq V(l_{xy})$ and $\rho(\mathcal{T}^k(l_{xy})) \leq \rho(l_{xy})$ for any integer $k \geq 0$. Given a topological order α of $\vec{G}$, there exists a minimum integer $N_0 \geq 0$ such that $V^c(\mathcal{T}^{N_0}(l_{xy})) = \emptyset$, for any path l_{xy} in $\vec{G}$. Equivalently, we have $N_0 = \min\{j : \mathcal{T}^j(l_{xy}) = \mathcal{T}^{j+1}(l_{xy})\}$. We denote $\mathcal{T}^{N_0}(l_{xy})$ by $\mathcal{T}^\infty(l_{xy})$ for simplicity in the rest of the text. In other words, by repeated applications of $\mathcal{T}$ on l_{xy}, we get a path l^*_{xy} with $V^c(l^*_{xy}) = \emptyset$. On the whole graph level, if we repeatedly apply the operator to each path l_{xy} in $\vec{G}$, no v-structures remain in the finally obtained DAG $\vec{G}' = \vec{G} + \vec{E}^*$, where $\vec{G}'$ has the edges added from all possible representation paths $\vec{E}^* = \{(u,v) \in \bigcup_{a,b \in V} Z_{ab} : Z_{ab} = \bigcup_{k=1}^{N_0} \vec{E}(\mathcal{T}^k(l_{ab})) \backslash \vec{E}, l_{ab} \text{ is a path in } \vec{G}\}$. This is why we call $\mathcal{T}$ a converging vertex sweeping operator.

We now provide an intuitive illustration of the sweeping operator $\mathcal{T}$ using Example 3.

Example 3 *Let $\vec{G} = (V, \vec{E})$ be a DAG with a topological order α such that $\alpha(v_1) < \alpha(v_2) < \cdots < \alpha(v_9)$, as shown in Figure 3. For the o-o path $l_{v_3v_6} = \{v_3 \overset{\vec{G}}{\sim} v_7 \overset{\vec{G}}{\sim} v_6\}$ in $\vec{G}$, it holds that $\mathcal{T}(l_{v_3v_6}) = \{v_3 \overset{\alpha}{\sim} v_6\}$ and $\mathcal{T}^2(l_{v_3v_6}) = \mathcal{T}(l_{v_3v_3})$. Thus, for the path $l_{v_3v_6}$, it holds $N_0 = 1$.*

Similarly, for the o-o path $l_{v_1v_4} = \{v_1 \overset{\vec{G}}{\sim} v_3 \overset{\vec{G}}{\sim} v_7 \overset{\vec{G}}{\sim} v_6 \overset{\vec{G}}{\sim} v_5 \overset{\vec{G}}{\sim} v_8 \overset{\vec{G}}{\sim} v_4\}$, we have $\mathcal{T}(l_{v_1v_4}) = \mathcal{T}(l_{v_1v_5}) \cup \mathcal{T}(l_{v_5v_4}) = \{v_1 \overset{\vec{G}}{\sim} v_3 \overset{\alpha}{\sim} v_5 \overset{\alpha}{\sim} v_4\}$, i.e., v_1, v_3 are adjacent in $\vec{G}$, $v_3 \to v_5$ and $v_4 \to v_5$ are connected according to α owing to the action of the sweeping operator, where $l_{v_1v_5} = \{v_1 \overset{\vec{G}}{\sim} v_3 \overset{\vec{G}}{\sim} v_7 \overset{\vec{G}}{\sim} v_6 \overset{\vec{G}}{\sim} v_5\}$ and $l_{v_5v_4} = \{v_5 \overset{\vec{G}}{\sim} v_8 \overset{\vec{G}}{\sim} v_4\}$. Moreover, $\mathcal{T}^2(l_{v_1v_4}) = \mathcal{T}(\mathcal{T}(l_{v_1v_4})) = \{v_1 \overset{\vec{G}}{\sim} v_3 \overset{\alpha}{\sim} v_4\}$ and $\mathcal{T}^3(l_{v_1v_4}) = \mathcal{T}^2(l_{v_1v_4})$, implying $N_0 = 2$ for path $l_{v_1v_4}$.

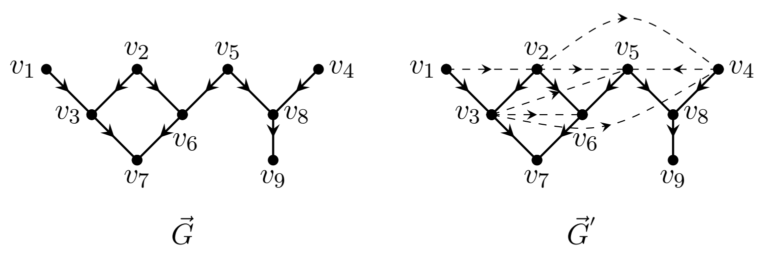


Figure 3: A DAG $\vec{G}$ with a topological order α, such that $\alpha(v_1) < \alpha(v_2) < \cdots < \alpha(v_9)$, and $\vec{G}'$ is the corresponding DAG obtained from $\vec{G}$ by repeated applications of the operator $\mathcal{T}$ on paths until $\mathcal{V}(\vec{G}') = \emptyset$. The directed, dashed edges represent newly added edges in $\vec{G}$.

We now introduce the terminal connecting property of paths.

Definition 21 (Terminal connecting property) *For a path l_{xy} in $\vec{G}$ with $x \overset{\vec{G}}{\nsim} y$, the path l_{xy} has the terminal connecting property with respect to α (TCP w.r.t. α) if $\mathcal{T}^\infty(l_{xy}) = \{x \overset{\alpha}{\sim} y\}$. Otherwise, it has the non-TCP w.r.t. α.*

Recall the cases in Example 3, it shows that the path $l_{v_3v_6}$ has the TCP w.r.t. α and the path $l_{v_1v_4}$ has the non-TCP w.r.t. α. Note that only o-o paths can have the TCP, and the TCP depends on the given topological order of $\vec{G}$. Proposition 22 provides an equivalent characterization of the TCP.

Proposition 22 *Let $\vec{G} = (V, \vec{E})$ be a DAG with a topological order α, and x, y be two non-adjacent vertices such that $\alpha(x) < \alpha(y)$ in $\vec{G}$. For any path l_{xy} in $\vec{G}$, l_{xy} has the TCP w.r.t. α if and only if y is the representation vertex of l_{xy}, i.e., $y = \arg\min_{w \in V(l_{xy}) \setminus \{x\}} \{\alpha(w)\}$.*

5.4.3 Characterizing Minimal Filling Edge Set

Next, we shall show the relation between the TCP property and the minimal filling edge set. First, we introduce the concept of active o-o paths, which are exactly the paths we need only to consider when finding a minimal filling edge set via the terminal connecting property of paths.

Definition 23 (Active o-o path) *Let $\vec{G} = (V, \vec{E})$ be a DAG and C be a subset of V. For an o-o path l_{xy} such that $x, y \in R$ in $\vec{G}$, the path l_{xy} is called an active o-o path conditioning on C in $\vec{G}$ if $V^o(l_{xy}) \cap C \subseteq V^c(l_{xy}) \subseteq \mathbf{An}_{\vec{G}}(C)$.*

As shown in Figure 3, the path $l_{v_1 v_4} = (v_1, v_3, v_2, v_6, v_5, v_8, v_4)$ is an active o-o path conditioning on $C = \{v_7, v_9\}$ in $\vec{G}$. For convenience, let

$$\mathcal{AP}^o_{\vec{G}}(C) = \{l_{xy} : l_{xy} \text{ is an active o-o path conditioning on C in } \vec{G}\}$$

denote the set of all active o-o paths conditioning on C in $\vec{G}$. Particularly, C-configurations aforementioned are special active o-o paths, which are exactly v-structures satisfying certain conditions.

Building on the concepts of terminal connecting property and active o-o paths, we introduce the notion of a tc-pair to characterize the minimal filling edge set. This characterization is based on a specific property of a particular type of paths in DAGs under conditioning.

Definition 24 (Tc-pair) *Let $\vec{G} = (V, \vec{E})$ be a DAG with a topological order α and C be a subset of V. For any $x, y \in R$ such that $x \overset{\vec{G}}{\not\sim} y$, the pair $\{x, y\}$ is called a terminal connecting pair under conditioning on C with respect to (tc-pair w.r.t.) α in $\vec{G}$ if there exists a path $l_{xy} \in \mathcal{AP}^o_{\vec{G}}(C)$ such that it has the TCP w.r.t. α in $\vec{G}$.*

We denote the set of all tc-pairs under conditioning on C w.r.t. α in $\vec{G}$ as

$$\mathcal{PS}^{tc}_{\vec{G}}(C; \alpha) \equiv \{\{x, y\} : \{x, y\} \text{ is a tc-pair w.r.t. } \alpha \text{ in } \vec{G}\},$$

and we denote the corresponding filling edge set of tc-pairs under conditioning on C w.r.t. α in $\vec{G}$ as

$$\triangle^{tc}_{\vec{G}}(C; \alpha) \equiv \{x \overset{\alpha}{\sim} y : \{x, y\} \in \mathcal{PS}^{tc}_{\vec{G}}(C; \alpha)\}.$$

We then find that it is exactly the minimal filling edge set of $\vec{G}_R$ w.r.t. α, as shown in the following proposition.

Proposition 25 *Let $\vec{G} = (V, \vec{E})$ be a DAG with a topological order α and C be a subset of V. Then $\triangle^{tc}_{\vec{G}}(C; \alpha)$ is a minimal filling edge set of $\vec{G}_R$ w.r.t. α.*

Combining Propositions 22 and 25, we know that $x \overset{\alpha}{\sim} y \in \triangle^{tc}_{\vec{G}}(C;\alpha)$ if and only if there exists a path $l_{xy} \in \mathcal{AP}^{o}_{\vec{G}}(C)$ such that $\alpha(w) > \max\{\alpha(x), \alpha(y)\}$ for any $w \in V^{o}(l_{xy})$. It is an intuitive characterization for a minimal filling edge set from the paths with the property of TCP, which implies a minimal filling edge set can be determined by considering only the paths in $\mathcal{AP}^{o}_{\vec{G}}(C)$. However, finding a minimal filling edge set by searching all active o-o paths with the property of TCP is not an efficient strategy. This motivates us to design an efficient algorithm in the following subsection.

5.5 Reverse Order Search Algorithm

In this section we will devise an efficient reverse order search (ROS) algorithm (see Algorithm 2) to find the minimal filling edge set $\triangle^{\alpha}_{\vec{G}}(R)$ and the minimal $\mathcal{I}$-map $\vec{G}_R + \triangle^{\alpha}_{\vec{G}}(R)$ for a given DAG $\vec{G}$ with a topological order α. Recall in Example 2, we have stated our main idea of the algorithm in the case where the conditional set is a single vertex. We formally describe the algorithm in a general setting in this subsection.

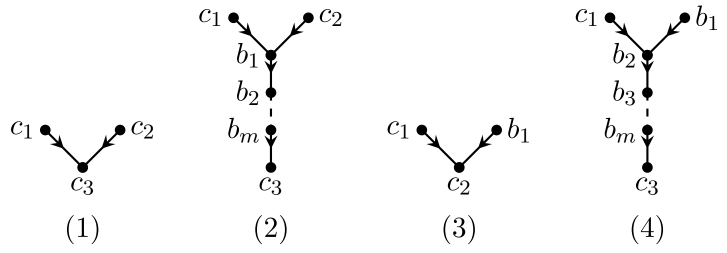


Figure 4: Four structures that are not allowed to be considered when we search for v-structures layer by layer, where $b_i \in \mathbf{an}_{\vec{G}}(C)$ for $i \in [m], m \geq 1$ and $c_j \in C$ for $j = 1, 2, 3$.

Before presenting the algorithm, we consider the four types of v-structures in Figure 4. The v-structures satisfy that $w \in \mathbf{An}_{\vec{G}}(C)$ in $\vec{G}$ but do not form C-configurations. Thus, our algorithm should not join x and y to form a minimal $\mathcal{I}$-map. In these cases, the four types of v-structure are redundant for our minimal $\mathcal{I}$-map searching algorithm. To improve the algorithm efficiency, we can eliminate these types of v-structures by removing the related edges, which start from a vertex in C. For this purpose, we derive Theorem 26.

Theorem 26 (Reduction invariance) *Suppose that $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$ is a BN with a DAG $\vec{G} = (V, \vec{E})$ and C is a subset of V. For any subset C' of C, it holds that*

(a) $\mathcal{I}(\vec{G}^{C'})^{C} = \mathcal{I}(\vec{G})^{C}$;

(b) $\mathcal{P}(\vec{G}^{C'})^{C} = \mathcal{P}(\vec{G})^{C}$.

In the above, $\vec{E}^{C} \equiv \{(x,y) \in \vec{E} : x \in C, y \in V\}$ and $\vec{G}^{C} \equiv \vec{G} - \vec{E}^{C}$.

In particular, when $C' = C$, we discover that $\mathcal{I}(\vec{G}^C)^C = \mathcal{I}(\vec{G})^C$ and $\mathcal{P}(\vec{G}^C)^C = \mathcal{P}(\vec{G})^C$, which reflects that we can simplify the analysis for a BN under conditioning. The lemma below shows that the edge set $\vec{E}^C$ is redundant in $\vec{G}$ for finding the minimal filling edge set $\triangle^{tc}_{\vec{G}}(C;\alpha)$. The result helps in the algorithm design to find the minimal filling edge set $\triangle^{tc}_{\vec{G}}(C;\alpha)$.

Lemma 27 *Suppose that $\vec{G} = (V, \vec{E})$ is a DAG with a topological order α and C is a subset of V. Then $\triangle^{tc}_{\vec{G}}(C;\alpha) = \triangle^{tc}_{\vec{G}^C}(C;\alpha)$.*

In what follows, we formally present the ROS algorithm (Algorithm 2) for determining a minimal filling edge set. Compared with the primary idea in Example 2, Algorithm 2 includes two additional steps: Steps 1 and 2. To avoid searching the four cases in Figure 4, it is sufficient to initialize $\vec{G}_0$ as $\vec{G}_A$ in Step 1 of Algorithm 2, where $A = \mathbf{An}_{\vec{G}^C}(C)$. Step 2 of Algorithm 2 is also beneficial as we can reduce the graph searching complexity according to a reverse order of the topological orders of converging connection vertices (colliders) in the v-structures concerned. Overall, the ROS algorithm is executed locally by completing the $\mathbf{pa}_{\vec{G}^C}(w)$ for all $w \in \mathbf{An}_{\vec{G}^C}(C)$ consecutively, according to the reverse order of the topological orders of the elements in $\mathbf{An}_{\vec{G}^C}(C)$.

Note the results from the ROS algorithm depends on the given topological order α in general. However, a special case is when $\mathcal{I}(\vec{G})$ is closed under conditioning on C, and the output of the ROS algorithm does not depend on the given topological order α. This is because $\vec{G}_R$ is exactly the minimal $\mathcal{I}$-map for $\mathcal{P}(\vec{G})^C$ in such a situation. In essence, our algorithm aims to find a minimal DAG $\vec{G}^*$ over V in the sense that $\vec{G}^*$ is a minimal DAG including $\vec{G}$ such that $\mathcal{I}(\vec{G}^*)^C = \mathcal{I}(\vec{G}^*_R)$. It should be noticed that the tentative algorithm described in Example 2 and the formal Algorithm 2 generate the same result, while Algorithm 2 exhibits lower computational complexity.

Now we illustrate the proposed algorithm using Example 4 as follows.

Example 4 *Let $\vec{G} = (V, \vec{E})$ be a DAG with a topological order α such that $\alpha(a) < \alpha(b) < \cdots < \alpha(j)$ and $C = \{d, f, i\}$, as shown in Figure 5. Using the ROS algorithm, we can find the minimal filling edge set of $\vec{G}_R$ w.r.t. α, denoted as $\vec{E}^\alpha_R = \{(a, b)\}$, and the corresponding minimal $\mathcal{I}$-map for the CDM $\mathcal{P}(\vec{G})^C$ under α, denoted as $\vec{G}^\alpha_R = \vec{G}_R + \vec{E}^\alpha_R$. In fact, the obtained $\vec{E}^\alpha_R$ by the algorithm is exactly the minimum edge set that is added to $\vec{G}$ to eliminate all the C-configurations by adding edges in the dynamically updated graphs $\vec{G}_j$ for $j \in [5]$.*

Next we shall discuss the correctness and complexity of the proposed ROS algorithm. Proposition 28 demonstrates the correctness of Algorithm 2. Proposition 29 shows that for a given causal DAG, the proposed algorithm is quite efficient duo to its linear time complexity, as m is usually small for a causal DAG. The simulation results in Table 1 also support this point. The six real-world BNs listed in Table 1 are available in the bnlearn repository (Scutari, 2024). The topological order of each network is arbitrarily given and the conditional vertex set C includes two vertices.The runtime in the table is the average value of 100 repetitions and all the values are rounded to three decimal places.

Proposition 28 *Suppose that $\vec{G} = (V, \vec{E})$ is a DAG with a topological order α and C is a subset of V. Then $\vec{E}^\alpha_R = \triangle^{tc}_{\vec{G}}(C;\alpha)$, where $\vec{E}^\alpha_R$ is the edge set obtained by the ROS algorithm under α.*

Algorithm 2 ROS Algorithm for Finding a Minimal $\mathcal{I}$-map (ROS_Alg($\vec{G}, \alpha, C$))

```
Input: A DAG $\vec{G} = (V, \vec{E})$, a topological order $\alpha$ of $\vec{G}$, and a subset $C$ of $V$.
Output: A minimal filling edge set: $\vec{E}_R^\alpha$ and a minimal $\mathcal{I}$-map: $\vec{G}_R^\alpha$.
1: Initialize $\vec{G}_R^\alpha = \vec{G}_R$, $\vec{E}_0 = \emptyset$ and $\vec{G}_0 = \vec{G}_A$, where $A = \mathbf{An}_{\vec{G}^C}(C)$;
2: Rearrange vertices in $\vec{G}_0$ as $\{u_j, j \in [k] : \alpha(u_j) > \alpha(u_{j+1}), \forall j \in [k-1]\}$, where
   $k$ is the number of vertices in $\vec{G}_0$;
3: if $k \geq 3$ then
4:     for $j = 1$ to $k-2$ do
5:         repeat
6:             Find v-structure $\{x \to u_j \leftarrow y\}$ in $\vec{G}_{j-1}$ and then join $x$ and $y$ as $x \to y$ if
               $\alpha(x) < \alpha(y)$ or $x \leftarrow y$ if $\alpha(x) > \alpha(y)$;
7:             Update $\vec{E}_j$ by $\vec{E}_{j-1} \cup \{x \overset{\alpha}{\sim} y\}$ and $\vec{G}_{j-1}$ by $\vec{G}_{j-1} + \{x \overset{\alpha}{\sim} y\}$;
8:         until no more edges to join;
9:         Update $\vec{G}_j$ by $\vec{G}_{j-1} - \{u_j\}$;
10:    end for
11: end if
12: return $\vec{E}_R^\alpha = \vec{E}_j$ and $\vec{G}_R^\alpha = \vec{G}_R + \vec{E}_R^\alpha$.
```

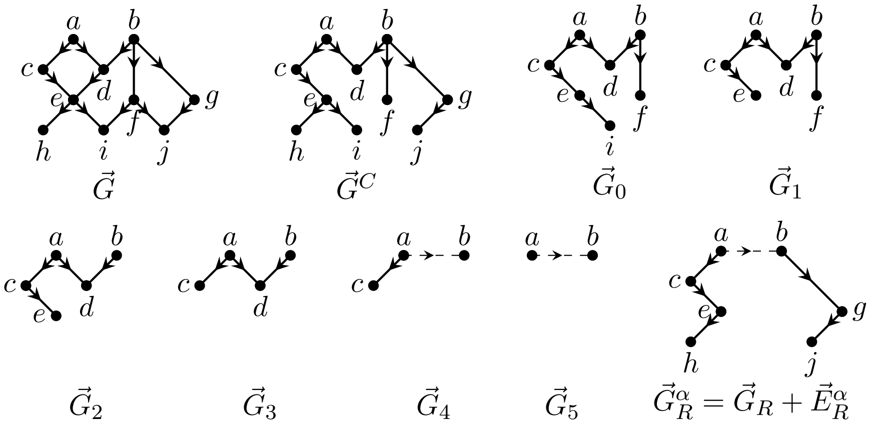


Figure 5: $\vec{G}$: An original DAG with a topological order α such that $\alpha(a) < \alpha(b) < \cdots < \alpha(j)$; $\vec{G}^C$: The obtained graph from $\vec{G}$ by removing the edge set $\vec{E}^C = \{(d, e), (f, i), (f, j)\}$; $\vec{G}_0$: The induced subgraph over $\mathbf{An}_{\vec{G}^C}(C)$ of $\vec{G}$; $\vec{G}_i$: The obtained graph from $\vec{G}_{i-1}$ by adding the edge set $\vec{E}_i \setminus \vec{E}_{i-1}$ and then removing w and the edges incident with w in $\vec{G}_{i-1} + \vec{E}_i \setminus \vec{E}_{i-1}$ for $i \in [5]$ and $w \in \{i, f, e, d, c\}$; $\vec{G}_R^\alpha$: The obtained graph from $\vec{G}_R$ by adding edge set $\vec{E}_R^\alpha = \vec{E}_5 = \{(a, b)\}$.

Proof. By Definition 17, we know $\vec{E}_R^\alpha = \triangle_{\vec{G}^C}^{tc}(C; \alpha)$. Lemma 27 yields that $\triangle_{\vec{G}^C}^{tc}(C; \alpha) = \triangle_{\vec{G}}^{tc}(C; \alpha)$, which gives $\vec{E}_R^\alpha = \triangle_{\vec{G}}^{tc}(C; \alpha)$. ■

Proposition 29 *Algorithm 2 has a complexity at most $O(m^2|k-2|)$, where m is the maximal number of parents of every vertex in $\mathbf{An}_{\vec{G}^C}(C)$ and k is the number of vertices in $\mathbf{An}_{\vec{G}^C}(C)$.*

Network	Asia	Insurance	Alarm	Win95pts	Andes	Munin
$\lvert V\rvert$	8	27	37	76	223	1041
$\lvert\mathbf{An}_{\vec{G}}(C)\rvert$	8	23	26	42	101	125
Runtime (s)	0.004	0.044	0.047	0.359	3.669	4.849

Table 1: Average runtime of ROS algorithm for different DAGs. The runtime in the table is measured in seconds.

6. Other Characterizations of Closure

In Section 4, we have provided a necessary and sufficient condition for the the closure of the independence models of BNs under conditioning by Proposition 13. In this section, we will discuss other several equivalent characterizations.

Recall Proposition 25, which states that the edge $x \overset{\alpha}{\sim} y$ is needed for a minimal $\mathcal{I}$-map if and only if there exists a path $l_{xy} \in \mathcal{AP}^o_{\vec{G}}(C)$ such that l_{xy} has the TCP w.r.t. a given topological order α. We can equivalently develop a characterization from the opposite perspective. In other words, we can consider the situations where the additional edge $x \overset{\alpha}{\sim} y$ is not required for a minimal $\mathcal{I}$-map. For any $x, y \in \mathbf{an}_{\vec{G}}(C)$ such that x, y are nonadjacent in $\vec{G}$, we call $\{x, y\}$ a *conditioning permissible pair* (cp-pair) under conditioning on C with respect to (w.r.t.) α in $\vec{G}$ if l_{xy} has the non-TCP w.r.t. α in $\vec{G}$ for any $l_{xy} \in \mathcal{AP}^o_{\vec{G}}(C)$. In fact, any nonadjacent vertices $\{x, y\} \subseteq \mathbf{an}_{\vec{G}}(C)$ without terminal connecting property form a conditioning permissible pair in $\vec{G}$. We denote the set of all cp-pairs under conditioning on C w.r.t. α in $\vec{G}$ as

$$\mathcal{PS}^{cp}_{\vec{G}}(C; \alpha) \equiv \big\{\{x, y\} : \{x, y\} \text{ is a cp-pair under conditioning on C w.r.t. } \alpha \text{ in } \vec{G}\big\}.$$

Now, we define an important concept of C-completeness through cp-pairs.

Definition 30 (C-completeness) *Suppose that $\vec{G} = (V, \vec{E})$ is a DAG with a topological order α and C is a subset of V. Let A be a subset of $\mathbf{an}_{\vec{G}}(C)$. The set A is C-complete with respect to (w.r.t.) α in $\vec{G}$ (where "C" means "conditioning") if for any $x, y \in A$, x, y are adjacent in $\vec{G}$ or $\{x, y\}$ is a cp-pair under conditioning on C w.r.t. α in $\vec{G}$.*

The concept of C-completeness can be viewed as an extension of the completeness in undirected graphs. It states that all vertex pairs should be adjacent, except those satisfying the non-terminal connecting property (non-TCP). See Corollary 53 and the related statements in Appendix E.1 for more details. Proposition 31 demonstrates the equivalence between C-completeness and the closure of $\mathcal{I}(\vec{G})$ under conditioning, indicating that the concept of C-completeness is independent of the topological order of $\vec{G}$.

Proposition 31 *Let $\vec{G} = (V, \vec{E})$ be a DAG with a topological order α and C be a subset of V. Then $\mathcal{I}(\vec{G})$ is closed under conditioning on C if and only if $\mathbf{an}_{\vec{G}}(C)$ is C-complete w.r.t. α in $\vec{G}$.*

To gain a better understanding of the concept of C-completeness, we can recall the DAG $\vec{G}'$ in the right panel of Figure 3. Suppose that $\vec{G}'$ has the same topological order α as $\vec{G}$ in Figure 3 and $C = \{v_7, v_9\}$. We can verify that $\mathbf{an}_{\vec{G}'}(C)$ is C-complete w.r.t. α in $\vec{G}'$. For example, v_1, v_2 are adjacent in $\vec{G}'$, there are no active o-o path conditioning on C with the endpoints v_6, v_8, and $l_{v_1v_4}$ has the non-TCP w.r.t. α in $\vec{G}'$ for the path $l_{v_1v_4} \in \mathcal{AP}^{o}_{\vec{G}'}(C)$. The C-completeness implies $\mathcal{I}(\vec{G}')$ is closed under conditioning.

We proceed to characterize the closure of independence models of BNs from several other perspectives. First, we propose a concept called covered set, which can be viewed as another description of the property that there is no C-configuration w.r.t. C in $\vec{G}$.

Definition 32 (Covered set) *Let $\vec{G} = (V, \vec{E})$ be a DAG and C be a subset of V. The set C is called a covered set in $\vec{G}$ if $\mathbf{pa}_{\vec{G}}(w) \setminus C$ is complete in $\vec{G}$ for any $w \in \mathbf{An}_{\vec{G}}(C)$. Specifically, for a single vertex $u \in V$, we call u a covered vertex in $\vec{G}$ if $\mathbf{pa}_{\vec{G}}(w)$ is complete in $\vec{G}$ for any $w \in \mathbf{An}_{\vec{G}}(u)$.*

It is evident that a root set is a covered set, but the converse is not true. We also observe the fact that C is a covered set in $\vec{G}$ if and only if C is a covered set in $\vec{G}^C$ (see Lemma 54 in Appendix E.1). This insight sheds light on the rationality of considering $\vec{G}^C$ in the designing of the ROS algorithm from a different perspective. We will show in Proposition 33 that the closure of the independence model is equivalent to C being a covered set.

Moreover, we can also characterize the closure of independence models from the viewpoint of the Markov equivalent class. For two DAGs $\vec{G}_1 = (V, \vec{E}_1)$ and $\vec{G}_2 = (V, \vec{E}_2)$, we say that they are $\mathcal{I}$-equivalent (or Markov equivalent), denoted by $\vec{G}_1 \approx \vec{G}_2$, if $\mathcal{I}(\vec{G}_1) = \mathcal{I}(\vec{G}_2)$. We call the set of all DAGs which are $\mathcal{I}$-equivalent to $\vec{G}$ an $\mathcal{I}$-equivalent class of $\vec{G}$, denoted by $\mathcal{M}(\vec{G})$, i.e., $\mathcal{M}(\vec{G}) = \{\vec{G}' : \vec{G}' \approx \vec{G}\}$. Now we summarize the different characterizations for the closure of independence models of BNs under conditioning as follows.

Proposition 33 *Let $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$ be a BN with a DAG $\vec{G} = (V, \vec{E})$ and α be a topological order of $\vec{G}$. Suppose that $\mathcal{P}(\vec{G})$ is faithful to $\vec{G}$. For any $C \subseteq V$, the following statements are equivalent:*

(a) $\mathcal{I}(\vec{G})$ is closed under conditioning on C, i.e., $\mathcal{I}(\vec{G})^C = \mathcal{I}(\vec{G}_R)$;

(b) $\vec{G}_R$ is a minimal $\mathcal{I}$-map of $\mathcal{P}(\vec{G})^C$;

(c) $\vec{E}^{\alpha}_R = \emptyset$, where $\vec{E}^{\alpha}_R$ is the edge set obtained by the ROS algorithm under α;

(d) $\vec{E}^{\beta}_R = \emptyset$ for any topological order β of $\vec{G}$, where $\vec{E}^{\beta}_R$ is the edge set obtained by the ROS algorithm under β;

(e) C is a covered set in $\vec{G}$;

(f) for any $w \in \mathbf{An}_{\vec{G}^C}(C)$, there exists a DAG $\vec{G}' \in \mathcal{M}(\vec{G}^C)$ such that w is a root vertex in $\vec{G}'$;

(g) for any $w \in \mathbf{An}_{\vec{G}^C}(C)$, there exists a DAG $\vec{G}'' \in \mathcal{M}(\vec{G}^C)$ such that $\{e_1, e_2, \dots, e_m\}$ is sequentially reversible in $\vec{G}''$, where $\mathbf{pa}_{\vec{G}^C}(w) = \{u_1, u_2, \dots, u_m\}$ such that $\alpha(u_1) > \alpha(u_2) > \cdots > \alpha(u_m)$ and $e_i = (u_i, w)$ for $i \in [m]$;

(h) for any $w \in \mathbf{An}_{\vec{G}^C}(C)$, there exists a DAG $\vec{G}^* \in \mathcal{M}(\vec{G}^C)$ such that $\mathbf{pa}_{\vec{G}^*}(w)$ is complete and $\mathbf{pa}_{\vec{G}^*}(w) = \mathbf{an}_{\vec{G}^*}(w)$.

Under the faithfulness assumption, either condition (a) or (b) of Proposition 33 implies that the pair $(\vec{G}_R, \mathcal{P}(\vec{G})^C)$ is also a BN, which is independent of its topological order. This fact is also indicated by condition (d). The condition (c) describes the avenue to judge the closure of $\mathcal{I}(\vec{G})$ based on the output $\vec{E}_R^\alpha$ of Algorithm 2, while the condition (e) can help us judge whether $\mathcal{I}(\vec{G})$ is closed under conditioning in an intuitive way. Furthermore, the conditions (f), (g), and (h) provide certain equivalent characterizations for when $\mathcal{I}(\vec{G})$ is closed under conditioning, from the viewpoint of $\mathcal{I}$-equivalent class.

7. Numerical Examples

In this section, via simulation studies, we will illustrate the benefits of using a DAG minimal $\mathcal{I}$-map to perform conditional inference for a BN.

7.1 Computing Conditional Distribution

We begin by introducing three methods for performing conditional inference of a BN, via Example 5

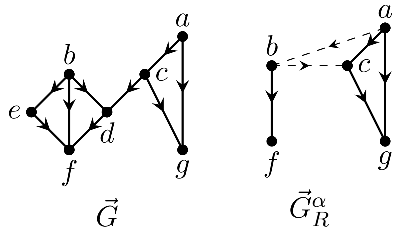


Figure 6: $\vec{G}$: a DAG with a topological order α such that $\alpha(a) < \alpha(b) < \cdots < \alpha(g)$; $\vec{G}_R^\alpha$: a DAG over R obtained by adding edges (a, b) and (b, c) to $\vec{G}_R$.

Example 5 *Consider a DAG $\vec{G} = (V, \vec{E})$ with a topological order α such that $\alpha(a) < \alpha(b) < \cdots < \alpha(g)$, as shown in Figure 6. Let $C = \{d, e\}$ and $R = \{a, b, c, f, g\}$. Applying the ROS algorithm, we can obtain a minimal filling edge set of $\vec{G}_R$ w.r.t. α, and a minimal $\mathcal{I}$-map for the CDM $\mathcal{P}(\vec{G})^C$ w.r.t. α, denoted by $\vec{E}_R^\alpha = \{(a, b), (b, c)\}$ and $\vec{G}_R^\alpha = \vec{G}_R + (a, b) + (b, c)$,* *respectively. Suppose that the true distribution of the related problem is $P(x_V) \in \mathcal{P}(\vec{G})$, and we are interested in $P(x_R|x_C) = P(x_R|x_d, x_e)$. Without loss of generality, we assume that the distributions in question are discrete and $P(x_C) > 0$.*

We can compute $P(x_R|x_C)$ using the J-Method, which takes advantage of the structural information of the joint distribution $P(x_V)$ and calculates the conditional distribution of interest using the ratio of the joint distribution $P(x_V)$ to marginal distribution $P(x_C)$, i.e.,

$$P(x_R|x_C) = \frac{P(x_R, x_C)}{P(x_C)} = \frac{P(x_R, x_C)}{\sum_{x_R} P(x_R, x_C)} = \frac{P(x_R, x_d, x_e)}{\sum_{x_R} P(x_R, x_d, x_e)},$$

where $P(x_R, x_d, x_e) = P(x_a)P(x_b)P(x_c|x_a)P(x_d|x_b, x_c)P(x_e|x_b)P(x_f|x_b, x_d, x_e)P(x_g|x_a, x_c)$.

Alternatively, we consider a method called C-Method as follows. we can calculate it directly according to the corresponding conditional distribution $P^C(x_R)$ of the true distribution $P(x_V)$ in the CDM $\mathcal{P}(\vec{G})^C \subseteq \mathcal{P}(\vec{G}^\alpha)^C = \mathcal{P}^C(\vec{G}^\alpha_R)$. That is, taking advantage of the structural information of the conditional distribution $P^C(x_R)$ directly, i.e.,

$$P(x_R|x_C) = P^C(x_R) = P^C(x_a)P^C(x_b|x_a)P^C(x_c|x_a, x_b)P^C(x_f|x_b)P^C(x_g|x_a, x_c).$$

Regardless of the J-Method or C-Method, we can always obtain the correct conditional probability distribution $P(x_R|x_C)$ when the available data sets are from the joint distribution $P(x_V)$. However, in practice, data sets may only be available from a conditional distribution instead of a joint one. In such cases, we cannot use the J-Method to obtain the conditional probability.

Also, one may adopt the IC-Method in practice, which ignores the changes of the structure over R caused by conditioning on C and calculates $P(x_R|x_C)$ using directly the structure of the subgraph of the original graph over R. That is,

$$P(x_R|x_C) = P^C(x_R) = P^C(x_a)P^C(x_b)P^C(x_c|x_a)P^C(x_f|x_b)P^C(x_g|x_a, x_c).$$

However, this naive approach is not necessarily correct, as $\vec{G}_R$ may not be an $\mathcal{I}$-map of $P(x_R|x_C)$.

Example 5 presents three methods to compute the global conditional distribution interested using the structural information of the distribution. It is important to note that the edges in $\vec{E}(\vec{G}_R)$ represent the dependencies induced by the causal influences, while the edges added in $\vec{E}^\alpha_R = \{(a, b), (b, c)\}$ represent the dependencies induced by the selection mechanism. It is crucial to understand this when performing causal discovery from data with selection bias.

Conditioning on $X_C = x_C$ restricts the sample space of all the random variables in X_R under the context $X_C = x_C$. On the other hand, the conditioning on $X_C = x_C$ may also change the structure of variables in X_R. This is an important consideration for local parameter estimation problems. Ignoring the changes in the structure of the remaining variables under conditioning can lead to completely wrong conclusions. Druzdzel and Díez (2003) showed this phenomenon for the single conditioning variable in the discrete case, and we will further confirm this assertion for multiple conditioning variables under both discrete and continuous circumstances in the following subsection.

7.2 Conditional Probability Queries

In this subsection, we present numerical experiments to demonstrate the benefits of performing conditional inference by using the minimal $\mathcal{I}$-map of the conditional model, under a given causal DAG. For this purpose, we consider the ASIA (LUNG CANCER) network proposed by Lauritzen and Spiegelhalter (1988) as the causal DAG $\vec{G}$ of a causal BN, as illustrated in Figure 7. For convenience, we use the first letter of each variable name to represent the corresponding variable.

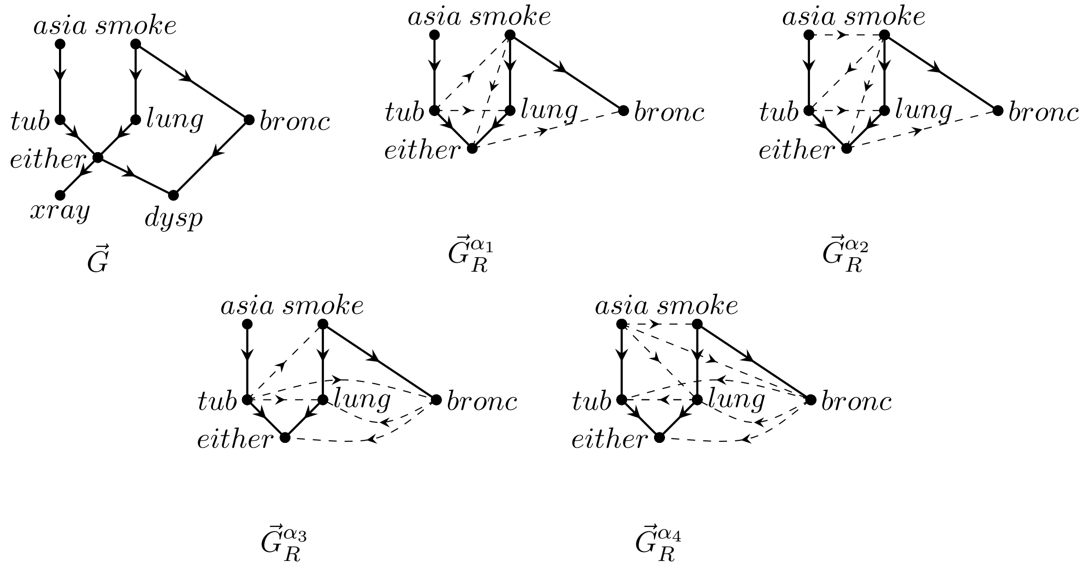


Figure 7: $\vec{G}$: the causal DAG structure of ASIA network, the conditioning set $C = \{xray, dysp\}$ and the remaining set $R = \{asia, tub, smoke, lung, either, bronc\}$; $\vec{G}_R^{\alpha_i}$: the DAG over R obtained by the ROS algorithm w.r.t. α_i, where α_i is a topological order of $\vec{G}$ for $i \in [4]$.

We conduct and analyse two types of simulation studies: one for discrete data and one for continuous data. We generate data sets from multinomial distributions and normal distributions (because the performance is similar to that of discrete cases, in order to save space, we does not provide results for continuous cases in the paper). The code is implemented in R language, and we perform simulations mainly based on **bnlearn** (Scutari and Denis, 2014) and **gRain** (Højsgaard, 2012) R packages. Our code is available at `https://github.com/xiexd569/BNs_under_Conditioning`. The CPU of the computer to conduct experiments in this paper is an AMD Ryzen 7 7840HS processor (8-cores, 3.8-GHZ).

In the discrete data setting, we assume that all variables are binary variables, taking values in $\{yes, no\}$. We generate a joint distribution P from $\mathcal{P}(\vec{G})$ by randomly generating a group of values from the uniform distribution $U(0, 1)$ as the values of its local conditional probability tables. Then, we sample simulation data from P according to its local conditional probability tables. This sampling method is called the forward sampling of the joint distribution. For a distribution $P(a, b, e, s, t, x, d) \in \mathcal{P}(\vec{G})$, assume that we are interested in the conditional probability value $P(a = no, b = no, e = yes, l = yes, s = no, t = yes|x = yes, d = yes)$, simply denoted by $\hat{P}$, we compare the behavior of the J-Method, the C-Method and the IC-Method (as mentioned in Example 5) by computing the conditional probability value of interest, respectively.

We conduct simulations in two scenarios: probabilistic reasoning for the interested conditional probability value $\hat{P}$, and local parameter estimation followed by probabilistic reasoning. To compare the sensitivity of the results under different topological orders, we conduct experiments under four different topological orders α_i for $i \in [4]$ (as mentioned in Figure 7). We use C-Method_i to denote applying C-Method under the ith topological order α_i for $i \in [4]$.

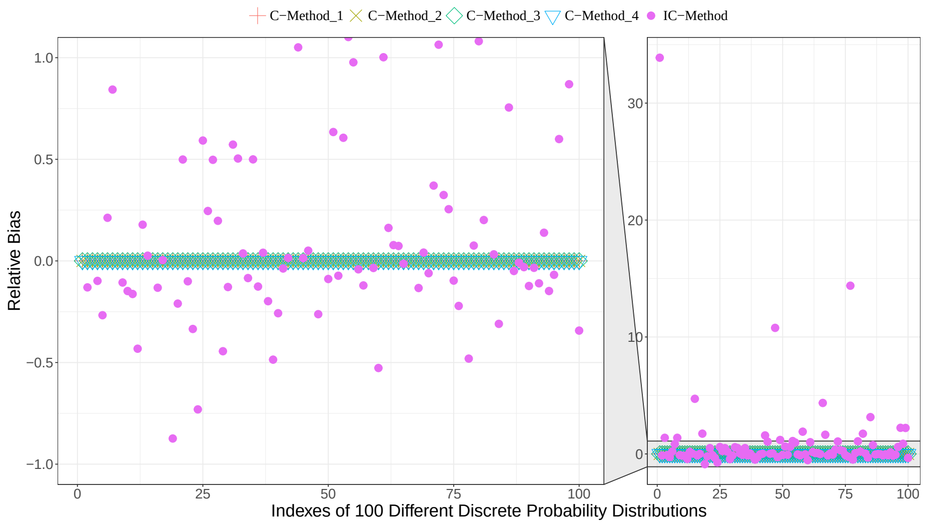
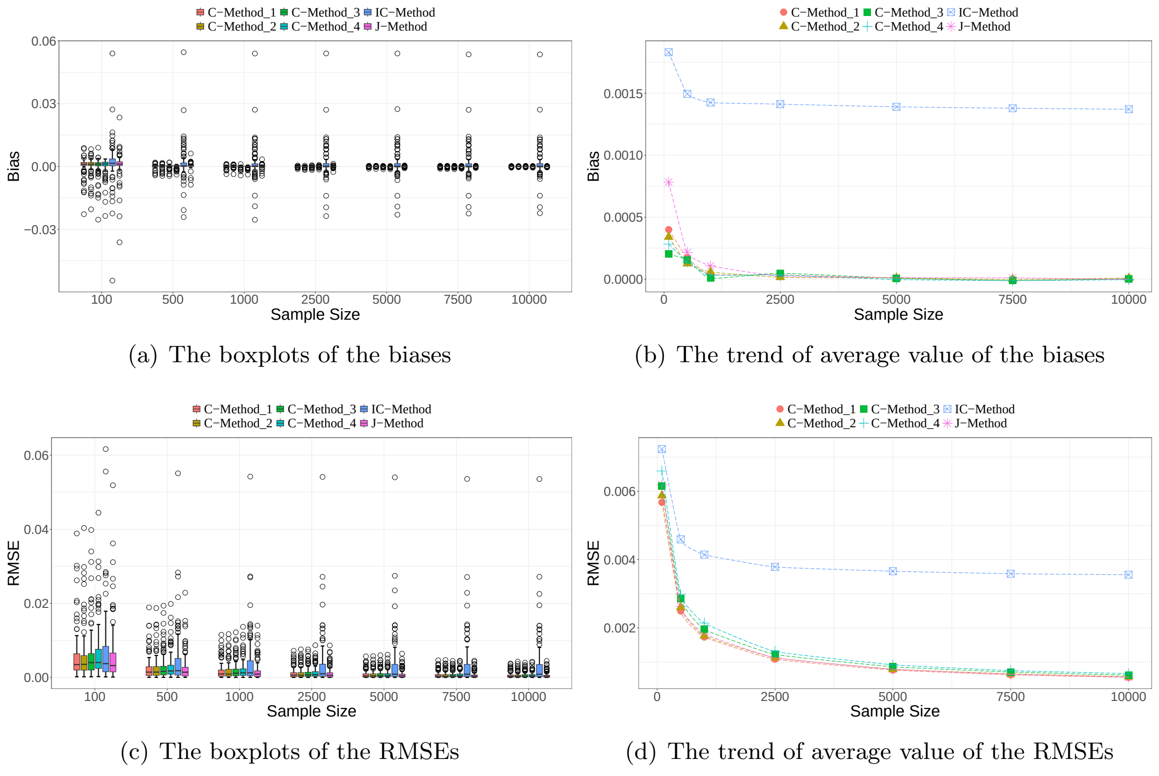


Figure 8: The scatter plot of the relative biases of computing $P(a = no, t = yes, s = no, l = yes, e = yes, b = no|x = yes, d = yes)$ by applying the C-Method and IC-Method in the 100 randomly generated multinomial distributions from $\mathcal{P}(\vec{G})$, under four different topological orders $\alpha_i, i \in [4]$.

In the first scenario, we assume that each local conditional probability value (local parameter) is known and take the conditional probability value calculated by the J-Method as the true value P_0. We then compare the performances of the C-Method and IC-Method. We randomly generate 100 multinomial distributions from $\mathcal{P}(\vec{G})$. For $j \in [100]$ and $w \in \{C, IC\}$, let $\hat{P}_w^j$ represent the conditional probability value obtained by applying w-Method in the jth multinomial distribution, and P_0^j denote the true value corresponding to the jth distribution. Then we calculate the relative bias of applying w-Method in the jth discrete distribution, which can be expressed as

$$RB(w; j) = \frac{bias(w; j)}{P_0^j} = \frac{\hat{P}_w^j - P_0^j}{P_0^j}.$$

Figure 8 shows the scatter plot of relative biases of the C-Method and IC-Method for the 100 multinomial distributions. It can be seen from the figure that the posterior probability reasoning $\hat{P}$ obtained by the C-Method performs as well as that of J-Method (P_0) for $\alpha_i, i = 1, 2, 3, 4$. However, when applying the IC-Method, there are significant biases of the results relative to the true values for the most elements among the 100 distributions, even the relative bias can reach more than 30 times of the true value for some distribution. This indicates that performing conditional inference by using its $\mathcal{I}$-maps can always obtain the same result as that obtained from the joint one in terms of probability reasoning, while it usually leads to a larger bias or even a wrong conclusion if we adopt the incorrect structure information, as the structure involved in the IC-Method may not be an $\mathcal{I}$-map of the CDM.



(a) The boxplots of the biases
(b) The trend of average value of the biases
(c) The boxplots of the RMSEs
(d) The trend of average value of the RMSEs

Figure 9: The biases and RMSEs of computing $P(a = no, b = no, e = yes, l = yes, s = no, t = yes|x = yes, d = yes)$ by applying the J-Method, C-Method and IC-Method for the 100 multinomial distributions with different sample sizes, under four different topological orders $\alpha_i, i \in [4]$.

In the second scenario, we first estimate the local parameters from simulation data and then compare the performances of the three methods under different sample sizes. We randomly generate 100 multinomial distributions from $\mathcal{P}(\vec{G})$, and for each fixed sample size $n = 100, 500, \ldots, 10000$, we obtain the local parameter estimations for the jth discrete distribution by averaging Bayesian estimations over $r = 100$ repetitions for $j \in [100]$. For $i, j \in [100]$ and $w \in \{J, C, IC\}$, let $\hat{P}_w^{i,j}$ denote the conditional probability value obtained by applying w-Method in the ith sample from the jth discrete distribution, and P_0^j represent the true value corresponding to the jth discrete distribution. Then, we can calculate the bias of employing the w-Method in the jth discrete distribution by

$$B(w;j) = \frac{1}{r}\sum_{i=1}^{r}(\hat{P}_w^{i,j} - P_0^j).$$

The root mean square error (RMSE) of applying the w-Method in the jth discrete distribution can be written as

$$RM(w;j) = \sqrt{\frac{1}{r}\sum_{i=1}^{r}(\hat{P}_w^{i,j} - P_0^j)^2}.$$

Figure 9 shows that for both the J-Method and the C-Method, in all 100 multinomial distributions, the biases and RMSEs gradually decrease as the sample size increases under the four different topological orders. The estimates of the two methods are asymptotically unbiased relative to the true values, and have smaller RMSEs than these of the IC-Method. However, there are always some distributions in the 100 distributions where the biases and RMSEs obtained by the IC-Method perform very poorly, and the large biases and RMSEs cannot be eliminated by increasing the sample size. These results demonstrate that if the effect of conditioning on the structure of the remaining variables is ignored, it can sometimes lead to larger biases or incorrect conclusions in the sense of the family of distributions. Furthermore, in smaller samples, the C-Method tends to have smaller biases than these of the J-Method. The reason is that when the sample size is small, there may exist certain bias in each parameter estimation. Since the C-Method has fewer parameters than the J-Method, the results calculated using the C-Method tend to have better accuracy compared to the J-Method when the sample size is small.

Next, we compare the runtime of the different methods. All values are rounded to three decimal places and all values represent the cumulative runtime. Each repetition of the J-Method consists of the three steps outlined below:

- **Step 1.** Use the sample to estimate the parameters.
- **Step 2.** Use the network structure and the estimated parameters to construct the clique tree (Koller and Friedman, 2009, p.346).
- **Step 3.** Use the clique tree and the parameters to calculate the conditional probability value $\hat{P}$.

In particular, we consider three versions of J-method regarding Step 3:

- **J-Method_cd.** Set the evidence on the clique tree according to the values of the conditional variables, and then directly query the conditional probability value $P(a = no, b = no, e = yes, l = yes, s = no, t = yes|x = yes, d = yes)$.
- **J-Method_cr.** First query the joint distribution value $P(a = no, b = no, e = yes, l = yes, s = no, t = yes, x = yes, d = yes)$ using directly the clique tree, then query the marginal distribution value $P(x = yes, d = yes)$ (i.e., the normalization constant), and finally compute the conditional probability by taking the ratio of the two values.

- **J-Method_fr.** Use the clique tree to query each local conditional probability value and normalization constant $P(x = yes, d = yes)$, then calculate the joint distribution value $P(a = no, b = no, e = yes, l = yes, s = no, t = yes, x = yes, d = yes)$ by multiplying the local conditional probabilities according to Equation 1, and finally compute the conditional probability by taking the ratio of the joint distribution value to the normalization constant.

Table 2 displays the cumulative CPU times in seconds for the three implementations of the J-Method, which are called J-Method_cd, J-Method_cr and J-Method_fr. For each fixed sample size n, we generate data from 100 multinomial distributions, and report the *cumulative* runtime of these 100 repetitions. This includes 100 repetitions for the local parameter estimation of a distribution (the "Est.T" table column), 100 repetitions for constructing a clique tree (the "Cli.T" column), and 100 repetitions for other components involved in computing $\hat{P}$ for each distribution (the "Com.T1" columns). Since J-Method_fr is the fastest, it is used to implement J-Method in the simulation for the rest of this paper. Unless otherwise specified, J-Method in this paper refers to J-Method_fr.

n	**Est.T**	**Cli.T**	**Com.T1**		
			J-Method_cd	**J-Method_cr**	**J-Method_fr**
100	7.254	156.536	223.583	941.987	31.268
500	7.394	155.305	225.113	943.185	32.907
1000	7.551	155.914	225.570	947.081	32.901
2500	8.090	156.123	225.839	948.225	32.732
5000	9.140	155.759	225.283	948.112	32.885
7500	10.290	156.083	224.234	948.144	33.075
10000	11.433	155.075	234.087	948.835	33.815

Table 2: Cumulative runtime of three implementations of J-Method (in seconds). **Est.T**: the cumulative time for parameter estimation; **Cli.T**: the cumulative time to construct the clique tree; **Com.T1**: the cumulative total time to compute the normalization constant using the clique tree and calculate the conditional probability values using the normalization constant and parameters.

Next, we compare the cumulative runtime of each component for the J-Method_fr, IC-Method, and C-Method. For the IC-Method, at each fixed sample size, we report the cumulative runtime for the results obtained from 100 multinomial distributions. This includes 100 repetitions for the local parameter estimation of each distribution (the "Est.T" table column), and 100 repetitions of using the estimated parameters to compute $\hat{P}$ (the "Com.T" column). For the C-Method, at each fixed sample size, we also report the cumulative runtime for the 100 repetitions. This includes 100 repetitions for running the ROS algorithm for each distribution (the "ROS.T" table row), 100 repetitions for the local parameter estimation (the "Est.T" table column), and 100 repetitions for using the estimated parameters to compute $\hat{P}$ (the "Com.T" column).

Table 3 shows the cumulative runtime of the J-Method_fr and the IC-Method. Here the cumulative runtime of the J-Method_fr is the same as that of J-Method_fr in Table 2, and we have **Com.T1=Nor.T+Com.T2**. Meanwhile, Table 4 presents the runtime of the C-Method_i, $i \in [4]$. The results illustrate that the C-Method has significantly faster runtime than the J-Method and almost the same runtime as the IC-Method. Importantly, the simulations in Table 4 indicate that the performance of the C-Method is stable for different topological orders.

n	**J-Method_fr**				**IC-Method**	
	Est.T	**Cli.T**	**Nor.T**	**Com.T2**	**Est.T**	**Com.T**
100	7.254	156.536	20.493	10.775	4.003	7.413
500	7.394	155.305	21.652	11.255	4.324	7.430
1000	7.551	155.914	21.419	11.482	4.586	7.571
2500	8.090	156.123	21.160	11.572	5.057	7.302
5000	9.140	155.759	21.464	11.421	5.891	7.735
7500	10.290	156.083	21.639	11.436	6.852	7.754
10000	11.433	155.075	21.689	12.126	8.033	7.618

Table 3: Cumulative runtime of J-Method_fr and IC-Method (in seconds). **Nor.T**: the cumulative time required to compute the normalization constant using the clique tree; **Com.T2**: the cumulative time to calculate the conditional probability values using the normalization constant and estimated parameters; **Com.T**: the cumulative time to calculate the conditional probability values using the estimated parameters.

n	**C-Method_1**		**C-Method_2**		**C-Method_3**		**C-Method_4**	
	Est.T	**Com.T**	**Est.T**	**Com.T**	**Est.T**	**Com.T**	**Est.T**	**Com.T**
100	4.284	7.759	4.383	7.938	4.287	8.014	4.320	8.412
500	4.374	7.979	4.407	8.051	4.394	8.044	4.375	8.504
1000	4.836	7.751	4.567	8.167	4.520	8.262	4.670	8.361
2500	5.438	7.824	5.168	8.209	5.360	7.995	5.208	8.562
5000	6.311	7.984	6.236	8.143	6.235	8.125	6.117	8.628
7500	7.035	8.033	7.184	8.236	6.988	8.441	7.106	8.473
10000	8.181	8.202	8.167	8.636	8.389	8.621	8.402	9.053
ROS.T	23.478		25.421		24.918		29.784	

Table 4: Cumulative runtime of C-Method_i, i$\in$ [4] (in seconds). **ROS.T**: the cumulative time required to run the ROS algorithm based on $\vec{G}$ under the topological order α_i, $i \in [4]$.

7.3 Repeated Conditional Queries of a BN

In the previous subsection, we have discussed the advantages of using the C-Method for handling a single conditional probability query. When it comes to repeated queries with different conditioning sets, it is known that a clique tree is efficient for queries involving repeated probability calculations, as it can reuse computations across different conditioning sets. Meanwhile, the C-Method requires running the ROS algorithm once for each conditioning set to find a DAG minimal $\mathcal{I}$-map, it might seem less efficient than using a clique tree. Through simulation examples, we demonstrate this is not the case. In fact, our method also exhibits competitive strength when dealing with repeated queries involving various sets of conditional variables.

The setup of the simulation is as follows:

1. A distribution over $X_V, V = \{asia, smoke, tub, lung, bronc, either, xray, dysp\}$ is randomly generated, and its corresponding conditional distribution under a conditioning set is then generated. Two data sets, each with a sample size of 10,000, are randomly generated based on this distribution and its conditional distribution, respectively.

2. The cumulative runtime for J-Method_fr and C-Method_1 are reported, with each method running 10,000 times based on the generated data.

Con.set	**J-Method_fr**				**C-Method_1**		
	Est.T	**Cli.T**	**Nor.T**	**Com.T2**	**ROS.T**	**Est.T**	**Com.T**
x	11.843	156.048	5.390	14.117	15.881	9.030	9.832
x,d	-	-	21.548	11.889	24.364	8.294	8.642
x,d,a	-	-	37.201	11.851	26.891	7.155	8.125
x,d,a,s	-	-	63.302	11.847	23.036	6.054	6.262
x,d,a,s,t	-	-	115.825	11.611	22.587	5.022	4.941
x,d,a,s,t,l	-	-	223.608	11.709	22.582	3.867	3.115
x,d,a,s,t,l,b	-	-	448.504	11.617	21.889	3.024	1.506

Table 5: Cumulative runtime of repeated conditional queries (in seconds). **Con.set**: the conditioning sets for conditional queries. The symbol "-" represents that by applying the J-Method_fr, the processes of parameter estimation and clique tree construction are performed only once, after which repeated queries under different conditional sets can be conducted. Therefore, the table lists only the cumulative runtime for parameter estimation and clique tree construction when the conditional set is x for the J-Method_fr.

The cumulative runtime is reported in Table 5. The table shows the cumulative runtime for two methods. Our method is denoted as C-Method_1 and the clique tree algorithm is denoted as J-Method_fr. The first column represents the set of conditional variables. The rest of the columns are organized into two groups. One group is denoted "J-Method_fr" for the clique tree algorithm. This group has four columns reporting four parts of the cumulative running time: parameter estimation (the "Est.T" table column), clique tree construction (the "Cli.T" column), normalization constant calculation (the "Nor.T" column), and the computation of $\hat{P}$ using the estimated parameters and normalization constant (the "Com.T2" column). Note in each replicate of the clique tree algorithm, the steps are only executed once for parameter estimation and clique tree construction. Then, we perform seven queries with different conditioning sets. The other column group for our method is denoted as "C-Method_1". It contains three parts for the cumulative runtime: running ROS algorithm (the "ROS.T" table column), parameter estimation (the "Est.T" column) and using the estimated parameters to compute $\hat{P}$ (the "Com.T" column).

Table 5 shows that using the clique tree is efficient for repeated conditional probability queries when the number of conditional variables is small. However, as the number of conditional variables grows, the clique tree approach gradually becomes time-consuming. For the J-Method_fr, the average time for these seven different conditioning sets, excluding the parameter estimation and the construction of the clique tree, is 142.860 seconds. On the other hand, for the C-Method_1, the average time for these seven different conditioning sets is 34.596 seconds (including the ROS algorithm). This example demonstrates the competitive advantages of our method, even in the cases of repeated queries with various conditioning sets.

From Tables 3, 4 and 5, it can be seen that the main reasons why C-Method is much faster than J-Method_fr are mainly twofold:

- J-Method_fr requires constructing the clique tree.
- J-Method_fr needs to calculate the normalization constant.

Next we will analyze in detail the reason why calculating the normalization constant is usually time-consuming by an example. Without loss of generality, we take the case under the topological order α_1 as an example. By the C-Method, we have

$$P(a,t,s,l,e,b|x=yes,d=yes) = P^C(a)P^C(t|a)P^C(s)P^C(l|s)P^C(e|l,t)P^C(b|s), \tag{6}$$

where $P^C(\cdot|\cdot) = P(\cdot|\cdot, x=yes, d=yes)$. The number of independent parameters needed by the C-Method is $1+2+1+2+4+2=12$.

By the J-Method,

$$P(a,t,s,l,e,b|x=yes,d=yes) = \frac{P(a,t,s,l,e,b,x=yes,d=yes)}{P(x=yes,d=yes)}, \tag{7}$$

where

$$\begin{aligned} &P(a,t,s,l,e,b,x=yes,d=yes)\\ &= P(a)P(t|a)P(s)P(l|s)P(e|l,t)P(b|s)P(x=yes|e)P(d=yes|b,e) \end{aligned} \tag{8}$$

and

$$\begin{aligned} &P(x=yes,d=yes)\\ &= \sum_{a,t,s,l,e,b} P(a,t,s,l,e,b,x=yes,d=yes)\\ &= \sum_{a,t,s,l,e,b} P(a)P(t|a)P(s)P(l|s)P(e|l,t)P(b|s)P(x=yes|e)P(d=yes|b,e). \end{aligned} \tag{9}$$

The number of independent parameters needed by J-Method is 1+2+1+2+4+2+2+4 = 18.

For calculating $p(a = no, t = yes, s = no, l = yes, e = yes, b = no|x = yes, d = yes)$ by C-Method, one only needs to do 5 multiplications according to (6). However, for calculating $p(a = no, t = yes, s = no, l = yes, e = yes, b = no|x = yes, d = yes)$ by the J-Method, one needs to compute first (8) and (9), and then divide $P(a, t, s, l, e, b, x = yes, d = yes)$ by $P(x = yes, d = yes)$. Besides 7 multiplications and 1 division, one also has to compute the normalized constant $P(x = yes, d = yes)$, which is the most complex part of calculating (7) after constructing a clique tree.

In general, if every variable considered is binary, one needs $\sum_{w\in V} 2^{|\mathbf{pa}_{\vec{G}}(w)\setminus C|}$ independent parameters by using the J-Method, while one only needs $\sum_{w\in R} 2^{|\mathbf{pa}_{\vec{G}^{\alpha}_{R}}(w)|} = \sum_{w\in R} 2^{|\mathbf{pa}_{\vec{G}^{\alpha}}(w)\setminus C|}$ by using the C-Method. Thus, one can save the time of calculating at most $(\sum_{w\in C} 2^{|\mathbf{pa}_{\vec{G}}(w)\setminus C|} - 1)$ multiplications and one division, and the time of calculating the normalized constant $P(x_C)$, to compute a query $P(X_R = x_R|X_C = x_C)$ for any fixed $x_R \in \mathcal{X}_R$ and any fixed value $x_C \in \mathcal{X}_C$ by using the C-Method instead of the J-Method.

7.4 Conditional Intervention Queries of a Causal BN

Similar to the case of conditional probability queries, it is also helpful to estimate the conditional distribution based on an interventional distribution. Suppose that $\vec{G} = (V, \vec{E})$ is a causal DAG. Then we have

$$P(x_{V\setminus A}|do(X_A = x_A)) = \prod_{w\in V\setminus A} P(x_w|x_{\mathbf{pa}_{\vec{G}}(w)})I_{\{X_A=x_A\}} \tag{10}$$

Equation (10) is known as the g-formula (Robins, 1986), the manipulated distribution formula (Spirtes et al., 2001) or the truncated factorization formula (Pearl, 2009) in the literature. Then we can view $P(x_{V\setminus A}|do(X_A = x_A))$ as a "joint distribution" over $X_{V\setminus A}$ of the DAG $\vec{G}_{V\setminus A}$ after performing do-intervention. Thus, we can apply our method to compute its conditional distribution $P(X_{R\setminus A}|do(X_A = x_A), X_C = x_C)$, based on the BN $(\vec{G}_{V\setminus A}, P(x_{V\setminus A}|do(X_A = x_A)))$.

For the causal DAG structure of the ASIA network (denoted as $\vec{G}$) depicted in Figure 10, assume that variable "t" has been intervened and set to level "no". In this scenario, we are interested in determining the interventional mean of the variable "e". Let $I_{\{t=no\}}$ be the indicator of the t component of the vector $V = \{a, b, d, e, s, t, x\}$ being equal to no when V takes a group of values $v \in \mathcal{X}_V$.

When t is intervened and set to level no, the distribution of the variables obtained from the conditional model of the BN $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$ under conditioning on $\{x = yes, d = yes\}$ is

$$\begin{aligned}
&P(a, b, e, l, s|do(t = no), x = yes, d = yes)\\
&= \frac{P(a, b, e, l, s, x = yes, d = yes|do(t = no))}{P(x = yes, d = yes|do(t = no))}\\
&= \frac{P(a, b, e, l, s, x = yes, d = yes|do(t = no))}{\sum_{a,b,e,l,s} P(a, b, e, l, s, x = yes, d = yes|do(t = no))},
\end{aligned}$$

where

$$\begin{aligned} & P(a,b,e,l,s,x=yes,d=yes|do(t=no)) \\ & = \prod_{w \in R} P(w|\mathbf{pa}_{\vec{G}}(w)) I_{\{t=no\}} \\ & = I_{\{t=no\}} P(a)P(b|s)P(e|l,t=no)P(l|s)P(s)P(x=yes|e)P(d=yes|b,e), \end{aligned} \tag{11}$$

Then, we can calculate the interventional mean by the J-Method as $E[e|t(no)] = \sum_{a,b,e,l,s} e \cdot P(a,b,e,l,s|x=yes,d=yes,do(t=no)) = \sum_e e \cdot \phi(e)$, where

$$\phi(e) = \frac{\sum_{a,b,l,s} P(a,b,e,l,s,x=yes,d=yes|do(t=no))}{\sum_{a,b,e,l,s} P(a,b,e,l,s,x=yes,d=yes|do(t=no))},$$

and $P(a,b,e,l,s,x=yes,d=yes|do(t=no)) = I_{\{t=no\}}P(a)P(b|s)P(l|s)P(s)P(e|l,t=no)P(x=yes|e)P(d=yes|b,e)$. The number of independent parameters is $1+2+2+1+2+2+4=14$.

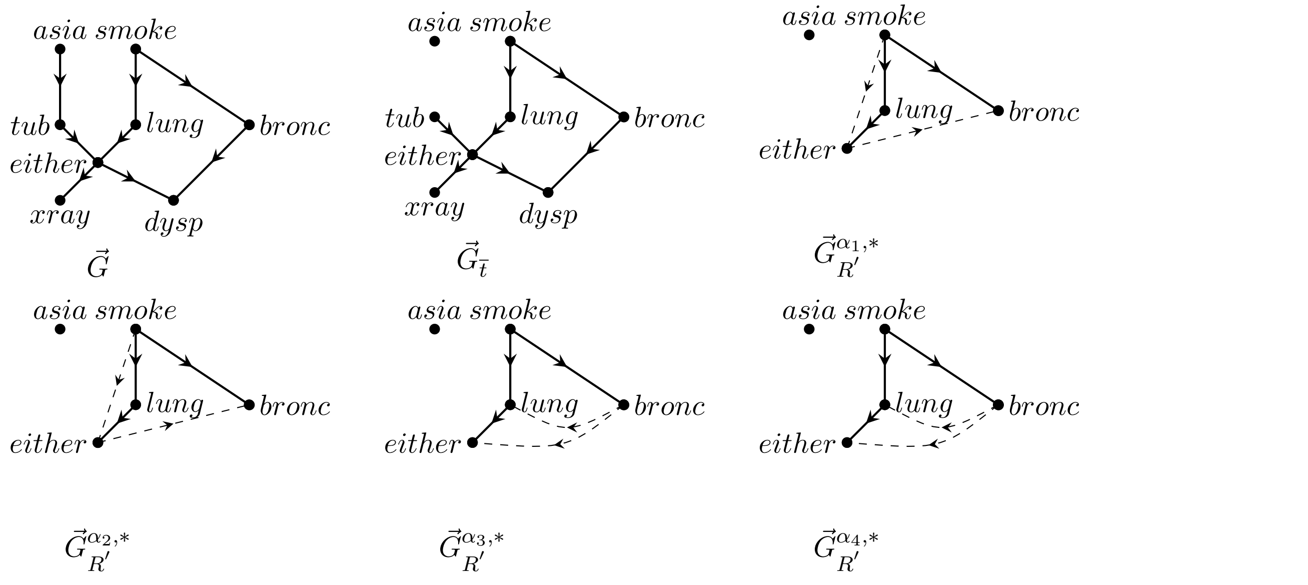

asia
smoke
tub
lung
bronc
either
xray
dysp
$\vec{G}$
$\vec{G}_{\bar{t}}$
$\vec{G}_{R'}^{\alpha_1,*}$
$\vec{G}_{R'}^{\alpha_2,*}$
$\vec{G}_{R'}^{\alpha_3,*}$
$\vec{G}_{R'}^{\alpha_4,*}$

Figure 10: $\vec{G}$: the causal DAG structure of ASIA network, the intervention vertex *tub*, the conditioning set $C = \{xray, dysp\}$ and the remaining set $R' = \{asia, smoke, lung, either, bronc\}$; $\vec{G}_{\bar{t}}$: the corresponding mutilated graph by applying do-intervention on *tub*; $\vec{G}_{R'}^{\alpha_i,*}$: the DAG over R' obtained by the ROS algorithm w.r.t. α_i based on $\vec{G}_{\bar{t}}$, where α_i is a topological order of $\vec{G}$ for $i \in [4]$.

Essentially, the conditional intervention distribution described above is exactly computing a conditional distribution after the do-intervention. Therefore, we can still simplify the calculation by using the DAG minimal $\mathcal{I}$-map. Next we will verify this point through a simulation below. By applying the C-Method under the topological order α_1 based on the mutilated graph $\vec{G}_{\bar{t}}$ in Figure 10, we have

$$\begin{aligned}
&P(a,b,e,l,s|do(t=no),x=yes,d=yes)\\
&=I_{\{t=no\}}P^C(a)P^C(b|e,s)P^C(e|l,s,t=no)P^C(l|s,t=no)P^C(s|t=no)\\
&=P^C(a|t=no)P^C(b|e,s,t=no)P^C(e|l,s,t=no)P^C(l|s,t=no)P^C(s|t=no)\\
&=\prod_{w\in R'}P^{C'}(w|\mathbf{pa}_{\vec{G}^{\alpha_1,*}_{R'}}(w)),
\end{aligned}$$

where $R'=\{asia, bronc, either, lung, smoke\}$, $C'=\{x,d,t\}$ and $\vec{G}^{\alpha_1,*}_{R'}$ is the DAG shown in Figure 10. We can calculate the interventional mean by the C-Method as

$$\begin{aligned}
E[e|t(no)]&=\sum_{a,b,e,l,s}e\cdot P(a,b,e,l,s|x=yes,d=yes,do(t=no))\\
&=\sum_e e\cdot\phi'(e),
\end{aligned}$$

where $\phi'(e)=\sum_{a,b,l,s}I_{\{t=no\}}P^C(a)P^C(b|e,s)P^C(e|l,s,t=no)P^C(l|s,t=no)P^C(s|t=no)$. The number of independent parameters is $1+4+4+2+1=12$.

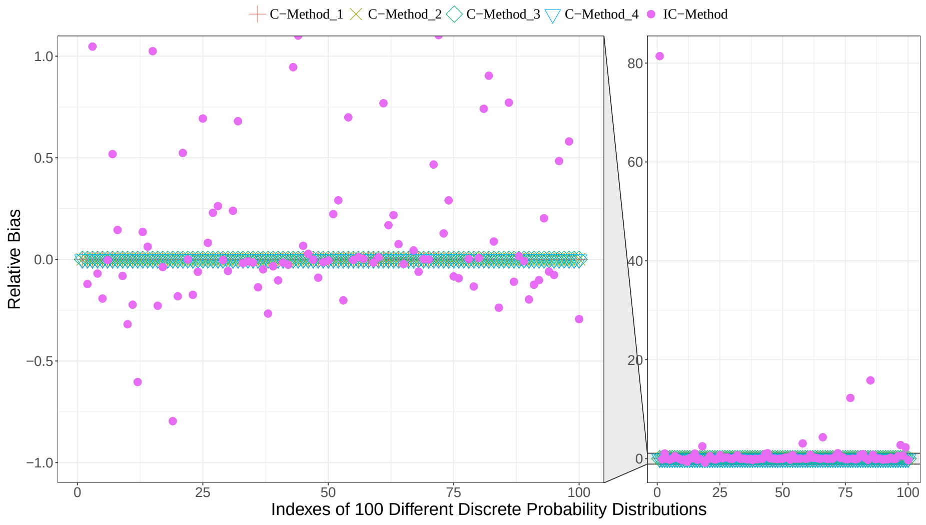


Figure 11: Based on $\vec{G}_{\bar{t}}$, the scatter plot of the relative biases of computing $P(a=no,b=no,l=yes,e=yes,s=no|do(t=no),x=yes,d=yes)$ by applying the C-Method and IC-Method in the 100 multinomial distributions, under four different topological orders $\alpha_i, i\in[4]$.

By the IC-Method based on $\vec{G}_{\bar{t}}$, we can calculate the interventional mean as

$$\begin{aligned} E[e|t(no)] &= \sum_{a,b,e,l,s} e \cdot P(a,b,e,l,s|do(t=no), x=yes, d=yes) \\ &= \sum_{a,b,e,l,s} e \cdot I_{\{t=no\}} P^C(a)P^C(b|s)P^C(e|l,t=no)P^C(l|s)P^C(s) \\ &= \sum_{e} e \cdot \phi''(e), \end{aligned}$$

where

$$\phi''(e) = \sum_{a,b,l,s} I_{\{t=no\}} P^C(a)P^C(b|s)P^C(e|l,t=no)P^C(l|s)P^C(s).$$

The number of independent parameters is $1+2+2+2+1=8$.

For simplicity, we only compare the performances of the J-Method, the C-Method_i for $i \in [4]$, and the IC-Method in computing the conditional intervention distribution value $P(a=no, b=no, l=yes, e=yes, s=no|do(t=no), x=yes, d=yes)$.

According to Figure 11, we can see that when calculating the interventional distribution value $P(a=no, b=no, l=yes, e=yes, s=no|do(t=no), x=yes, d=yes)$, based on the mutilated graph $\vec{G}_{\bar{t}}$, the intervention posterior probability reasoning $P(a=no, b=no, l=yes, e=yes, s=no|do(t=no), x=yes, d=yes)$ obtained by the C-Method performs as well as that of J-Method (base line) for $\alpha_i, i=1,2,3,4$. However, if we apply the IC-Method, there are significant biases of the results relative to the true values for the most elements among the 100 distributions, even the relative bias can reach more than 80 times of the true value for some distribution.

Although it is a single variable intervention, it shows the possibility that this approach is also beneficial for estimating conditional interventional means interested. The importance of this discovery is for saving the computational time of conditional inferences after intervention. This is the main reason why we aim to find a DAG minimal $\mathcal{I}$-map of the conditional model of a causal BN, under the constraint that keeping the directed edges in the original causal DAG unchanged. It retains the causal interpretability among the original edges, and one can make as much use of the structural information of the conditional model as possible to improve the efficiency of conditional inference. Meanwhile, it is important to note that, in the DAG minimal $\mathcal{I}$-map, the extra added edges are not causal.

8. Discussion and Future Work

In this paper, we analyze the conditioning operation of a BN and present ROS, a polynomial-time algorithm that can find a DAG as a minimal $\mathcal{I}$-map for the conditional distributions of a BN. We discuss the multiple benefits of using a DAG minimal $\mathcal{I}$-map to perform conditional statistical inference of a BN. Additionally, we have provided the necessary and sufficient conditions for the closure of conditioning in a BN from multiple perspectives, which can be easily verified by examining the graph structure.

Our idea of adding new edges into the original DAG is to extend the distribution family $\mathcal{P}(\vec{G})^C$ instead of extending the class of DAGs. Either the idea of extending the class of DAGs or the idea of extending the distribution family is to find a graph over R with which the conditional distribution family $\mathcal{P}(\vec{G})^C$ is compatible, as $\vec{G}_R$ may not be an $\mathcal{I}$-map of $\mathcal{P}(\vec{G})^C$. Although the minimal $\mathcal{I}$-map obtained by adding edges in the original DAG may have a loss of the conditional independence (CI) information, it inherits all the advantages of a BN. In this paper, we focus on finding a minimal $\mathcal{I}$-map of a causal BN under conditioning, with the restriction of keeping the existing directed causal edges unchanged. This is useful especially for the practical applications where one also needs as many simple causal explanations as possible besides CIs, such as medical diagnosis, financial analysis and government decision-making.

Investigating the issue of BNs under conditioning within the framework of DAGs is valuable for causal interpretability, causal discovery and inference, in the presence of selection bias. DAGs under conditioning can represent generating mechanisms of selection bias (Borboudakis and Tsamardinos, 2015; Cooper, 2000; Spirtes et al., 1995; Zhang, 2008) and missingness (Daniel et al., 2012; Gain and Shpitser, 2018; Mohan et al., 2013; Liu and Constantinou, 2022; Squires and Uhler, 2022; Strobl et al., 2018; Tu et al., 2019), which enables researchers to reason about spurious correlations and develop strategies to address them by formulating and testing causal hypotheses. However, attempting to analyze data causally without good background knowledge of the subject area and plausible causal mechanisms, including those giving rise to data with selection bias, is unfeasible (Daniel et al., 2012).

Admittedly, there are also some limitations to using DAGs as minimal $\mathcal{I}$-maps of conditional models. First, DAGs cannot be used to deal with problems with feedback mechanisms due to the property of acyclicity. Second, using DAGs as minimal $\mathcal{I}$-maps under conditioning may induce more edges than other types of graphs, and may not be optimal in terms of capturing conditional independencies in the corresponding distributions.

Throughout the current work, the original DAG $\vec{G}$ over all variables is known, and we examine the effect of conditioning. In some applications, it is interesting to estimate the DAG structure based on data with selection bias. Although there are various BN learning algorithms available in the literature, as discussed in recent review papers such as Glymour et al. (2019), Kitson et al. (2023), Vowels et al. (2022) and Zanga et al. (2022), structural learning of DAGs is challenging in the presence of selection bias. Therefore, we have divided this problem into two steps for investigation: (1) Given a DAG over the full variable set, explore what the DAG minimal $\mathcal{I}$-map for the conditional model looks like; (2) When the DAG over the full variable set is unknown and only selection biased data is available, investigate how to learn directly a DAG minimal $\mathcal{I}$-map. We have accomplished the first step in this work. Our theoretical findings can serve as the foundation for DAG structure learning, especially for selection biased data. Regarding the second step, we have obtained some preliminary results and plan to conduct further research in the near future.

One of our referees points out that, given a predetermined variable order (not necessarily a topological order), one can obtain a minimal $\mathcal{I}$-map by checking d-separations, based on a given full DAG or a data set with selection bias. This is similar to the unconditional case (Koller and Friedman, 2009, p.79), but with the selection variables included in the conditioning set during d-separation checking. When the full DAG is known, using the local Markov property to obtain a DAG minimal $\mathcal{I}$-map of the conditional model by d-separation checks is a plausible alternative. When the vertex order is the same as some inverse topological order of the DAG, the DAG minimal $\mathcal{I}$-map obtained by local Markov property is consistent with the one sought by our proposed ROS algorithm. Through an initial exploration, we have found that the proposed ROS algorithm executes more efficiently (see Appendix F for details). This is because our method avoids the exhaustive checking of d-separation assertions for specific subsets within the Markov boundary of each vertex in the updated graph. When the full DAG is unknown and we only have access to a set of selection biased data, learning a DAG minimal $\mathcal{I}$-map with a given order of vertices and by using the local Markov property can be a valuable direction for future research. The referee also suggests that, for the global viewpoint, there are algorithms (e.g., GES, Chickering and Meek, 2002) to learn a minimal $\mathcal{I}$-map from distributions that satisfy the composition property. This is also a valuable direction for future exploration. Due to the limited length of the article, we will not elaborate further here.

The DAG framework offers an avenue for efficient estimation of conditional intervention distribution under causal BNs with selection bias. The previous study in Guo et al. (2022), based on a DAG minimal $\mathcal{I}$-map of the marginal model of a BN, investigated the effective estimation of causal effects. Inspired by their work, in the presence of selection bias, we can explore the effective estimation of conditional causal effects based on a DAG minimal $\mathcal{I}$-map of the conditional model of a BN. Thus, our results in this paper provide a potential foundation for extending this approach to improve the efficiency of conditional causal effect, which is partially supported by some simulation results in Section 7.

It is important to note that the results introduced here only deal with the conditional independence constraints. Under conditioning in a BN, Lauritzen (1998) showed that there exist non-CI factorization constraints, and Evans and Didelez (2015) showed the existence of non-CI, non-factorization constraints in a certain categorical case. In addition, the concept of closure for a BN under conditioning in this paper differs from that described in Richardson and Spirtes (2002), in which the closure of a BN is a more general concept than that in this paper. A potential direction of future work is to investigate the topic based on the general version of the closure of a BN under conditioning. Meanwhile, it is interesting to find minimal $\mathcal{I}$-maps for BNs under both marginalization and conditioning within the DAG framework in the future. Moreover, based on the results obtained in this paper, there are several other possibilities for further research. For example, similar to the undirected graphical model (Liu and Guo, 2012), by combining with the marginalization of a BN, we can investigate the collapsibility of a conditional model of a BN.

Acknowledgments

The authors are grateful to the associate editor and three anonymous reviewers for many valuable comments. We also thank Dr. Shiyuan He's help to polish the writing of the paper. This work was partially supported by the National Natural Science Foundation of China (Grant Numbers 12431009, 11631003, 11690012 and 11726629), the National Key Research and Development Program of China (Grant Numbers 2020YFA0714102, 2020YFA0714100) and Tianyuan Fund for Mathematics of the National Natural Science Foundation of China (Grant Numbers 12426520 and 12426105).

Appendix A. Symbol List

For the readability of the paper, below is a list of newly defined symbols, their explanations, and the specific page numbers where they are defined.

Symbol	Definition	Page
$\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$	Bayesian network	2
$\vec{G} = (V, \vec{E})$	directed acyclic graph over V	7
$\mathcal{P}(\vec{G})^C$	the conditional distribution model	9
R	the set $V \setminus C$	9
$\mathcal{I}(\vec{G})^C$	the conditional independence model	9
$P^C(X_A \vert X_B)$	the conditional distribution $P(X_A \vert X_B, X_C = x_C)$	9
$\mathcal{P}^C(\vec{G}_R)$	the set of conditional distributions that are compatible with $\vec{G}_R$	10
$\mathcal{V}(\vec{G})$	the set of all v-structures in $\vec{G}$	12
$\mathcal{AV}_{\vec{G}}(C)$	the set of all C-configurations w.r.t. C in $\vec{G}$	12
$\vec{E}^{\alpha}_{c}(R)$	the complement set of edge set $\vec{E}(\vec{G}_R)$ under α	14
$\triangle^{\alpha}_{\vec{G}}(R)$	a minimal filling edge set of $\vec{G}_R$ w.r.t. α	15
$\mathcal{AP}^{o}_{\vec{G}}(C)$	the set of all active o-o paths under conditioning on C in $\vec{G}$	20
$\mathcal{PS}^{tc}_{\vec{G}}(C;\alpha)$	the set of all tc-pairs under conditioning on C w.r.t. α in $\vec{G}$	20
$\vec{G}^C$	the graph obtained by removing the outgoing edges of C from $\vec{G}$	21
$\mathcal{PS}^{cp}_{\vec{G}}(C;\alpha)$	the set of all cp-pairs under conditioning on C w.r.t. α in $\vec{G}$	24
$\mathcal{M}(\vec{G})$	the Markov equivalent class of $\vec{G}$	25

Table 6: List of symbols. The last column provides the page numbers where the symbols are defined.

Appendix B. Proof of Claims in Section 3

For further study, we use $X_U \perp\!\!\!\perp X_W \vert (X_S, X_C)[P]$ to represent $X_U \perp\!\!\!\perp X_W \vert (X_S, X_C = x_C)[P]$ for some $x_C \in \mathcal{X}_C$. We define

$$\mathcal{I}(P)^C = \{\langle U, W \vert S\rangle : X_U \perp\!\!\!\perp X_W \vert (X_S, X_C)[P] \text{ with pairwise disjoint subsets } U, W, S \subseteq R\},$$

where $\mathcal{I}(P)^C$ is the set of triplets $\langle U, W \vert S\rangle$ that are entailed by the conditional independencies $X_U \perp\!\!\!\perp X_W \vert (X_S, X_C)$ in a probability distribution P. In order to prove Proposition 5, we first give two lemmas below.

Lemma 34 *Suppose that P is a joint probability distribution over the random variable vector X_V and C is a subset of V. Let P^C be the corresponding conditional probability distribution of P given X_C. Then we have $\mathcal{I}(P)^C = \mathcal{I}(P^C)$.*

Proof. Without loss of generality we assume that the density of P is f and the conditional density of P^C is f^C. For $X_U \perp\!\!\!\perp X_W \vert (X_S, X_C)[P]$, it follows that $f(x_U \vert x_W, x_S, x_C) =$

$f(x_U|x_S, x_C)$. Thus

$$\begin{aligned} f^C(x_U|x_W, x_S) &= \frac{f^C(x_U, x_W, x_S)}{f^C(x_W, x_S)} & &= \frac{f(x_U, x_W, x_S, x_C)/f(x_C)}{f(x_W, x_S, x_C)/f(x_C)} \\ &= \frac{f(x_U, x_W, x_S, x_C)}{f(x_W, x_S, x_C)} & &= f(x_U|x_W, x_S, x_C) \\ &= f(x_U|x_S, x_C) & &= \frac{f(x_U, x_S, x_C)}{f(x_S, x_C)} \\ &= \frac{f(x_U, x_S|x_C)f(x_C)}{f(x_S|x_C)f(x_C)} & &= \frac{f(x_U, x_S|x_C)}{f(x_S|x_C)} \\ &= \frac{f^C(x_U, x_S)}{f^C(x_S)} & &= f^C(x_U|x_S), \end{aligned}$$

which means $X_U \perp\!\!\!\perp X_W|X_S[P^C]$. Since all the equations in the proof above are reversible, we complete this proof. ■

Lemma 35 (Lauritzen, 1996) *Suppose that $\vec{G} = (V, \vec{E})$ is a DAG and $\mathcal{P}$ is a family of probability distributions with strictly positive densities over X_V. For any $P \in \mathcal{P}$, P factorizes according to $\vec{G}$ if and only if $\vec{G}$ is an $\mathcal{I}$-map of P.*

B.1 Proof of Proposition 5

Proof. To prove (a), for any $P \in \mathcal{P}(\vec{G})$, Lemma 35 together with that P factorizes according to $\vec{G}$ yields $\mathcal{I}(\vec{G}) \subseteq \mathcal{I}(P)$. It follows from Lemma 34 that $\mathcal{I}(\vec{G})^C \subseteq \mathcal{I}(P)^C = \mathcal{I}(P^C)$. To prove (b), under the faithfulness assumption, there exists a distribution $P_* \in \mathcal{P}(\vec{G})$ such that $\mathcal{I}(\vec{G}) = \mathcal{I}(P_*)$, implying that there exists a distribution $P_*^C \in P(\vec{G})^C$ such that $\mathcal{I}(\vec{G})^C = \mathcal{I}(P_*)^C = \mathcal{I}(P_*^C)$. ■

Appendix C. Proof of Claims in Section 4

To prove Theorem 10, we must first prove Proposition 37. We begin by considering two BNs that share the same variable set. Our intuition tells us that conditioning on the same variable subset of both networks, when one network's set of edges includes that of the other, should yield a result that the induced models of BNs under conditioning also have an inclusion relationship. This insight forms the foundation for the proof of Proposition 37, which will prove valuable for future proofs.

To prove Proposition 37, we first propose a lemma as follows.

Lemma 36 (Monotonicity) *Suppose that $\mathcal{B}_i = (\vec{G}_i, \mathcal{P}(\vec{G}_i))$ is a BN with a DAG $\vec{G}_i = (V, \vec{E}_i)$ for $i = 1, 2$. For any subset C of V, if $\vec{E}_1 \subseteq \vec{E}_2$, we find that*

(a) $\mathcal{I}(\vec{G}_1)^C \supseteq \mathcal{I}(\vec{G}_2)^C$;

(b) $\mathcal{P}(\vec{G}_1)^C \subseteq \mathcal{P}(\vec{G}_2)^C$.

Proof. For any $\langle X, Y|S\rangle \in \mathcal{I}(\vec{G}_2)^C$, let $l_{xy} \subseteq \vec{G}_1$ be any path connecting x and y for any $x \in X$ and $y \in Y$. From $\vec{E}_1 \subseteq \vec{E}_2$ and $V(\vec{G}_1) = V(\vec{G}_2) = V$, it follows that $l_{xy} \subseteq \vec{G}_2$, implying that l_{xy} is blocked by $S \cup C$ in $\vec{G}_2$. Since $\mathbf{An}_{\vec{G}_1}(S \cup C) \subseteq \mathbf{An}_{\vec{G}_2}(S \cup C)$, it follows also that l_{xy} is blocked by $S \cup C$ in $\vec{G}_1$, proving (a).

To prove (b), for any $w \in V$, let $A_w = \mathbf{pa}_{\vec{G}_2}(w) \setminus \mathbf{pa}_{\vec{G}_1}(w)$. Then it follows from the acyclicity of $\vec{G}_1$ that $A_w \subseteq \mathbf{nd}_{\vec{G}_1}(w) \setminus \mathbf{pa}_{\vec{G}_1}(w)$, which implies $w \perp\!\!\!\perp A_w | \mathbf{pa}_{\vec{G}_1}(w)[\vec{G}_1]$. For any $P(x_V) \in \mathcal{P}(\vec{G}_1)$, we have $P(x_V) = \prod_{w \in V} P(x_w | x_{\mathbf{pa}_{\vec{G}_1}(w)}) = \prod_{w \in V} P(x_w | x_{\mathbf{pa}_{\vec{G}_2}(w)}) \in \mathcal{P}(\vec{G}_2)$. Therefore, $\mathcal{P}(\vec{G}_1) \subseteq \mathcal{P}(\vec{G}_2)$, which implies $\mathcal{P}(\vec{G}_1)^C \subseteq \mathcal{P}(\vec{G}_2)^C$. ■

Proposition 37 (Heritability) *Suppose that $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$ is a BN with a DAG $\vec{G} = (V, \vec{E})$ and C is a subset of V. Then*

(a) $\mathcal{I}(\vec{G})^C \subseteq \mathcal{I}(\vec{G}_R)$;

(b) $\mathcal{P}(\vec{G})^C \supseteq \mathcal{P}^C(\vec{G}_R)$.

Proof. Let $\vec{G}^* = \vec{G}_R \cup N_C$, where $N_C = (C, \vec{E}_C)$ is a null graph over C, i.e., $\vec{E}_C = \emptyset$. From Lemma 36, it follows that $\mathcal{I}(\vec{G})^C \subseteq \mathcal{I}(\vec{G}^*)^C$ and $\mathcal{P}(\vec{G})^C \supseteq \mathcal{P}(\vec{G}^*)^C$, implying $\mathcal{I}(\vec{G})^C \subseteq \mathcal{I}(\vec{G}^*)^C = \mathcal{I}(\vec{G}_R)$. Since we can choose $P(x_V) = P^C(x_R)P^0(x_C)$ to be a distribution in $\mathcal{P}(\vec{G}^*)$ for any $P^C(x_R) \in \mathcal{P}^C(\vec{G}_R)$, where $P^0(x_C)$ is a distribution compatible with N_C, it holds $\mathcal{P}^C(\vec{G}_R) \subseteq \mathcal{P}(\vec{G}^*)^C \subseteq \mathcal{P}(\vec{G})^C$. ■

C.1 Proof of Theorem 10

Proof. From (a) of Proposition 5, it follows that $\mathcal{I}(\vec{G}_R) = \mathcal{I}(\vec{G})^C \subseteq \mathcal{I}(P^C)$ for any $P^C \in \mathcal{P}(\vec{G})^C$. By Lemma 35, we have that $P^C \in \mathcal{P}^C(\vec{G}_R)$, which gives $\mathcal{P}(\vec{G})^C \subseteq \mathcal{P}^C(\vec{G}_R)$. According to (b) of Proposition 37, we can arrive at $\mathcal{P}^C(\vec{G}_R) \subseteq \mathcal{P}(\vec{G})^C$, implying $\mathcal{P}^C(\vec{G}_R) = \mathcal{P}(\vec{G})^C$.

Conversely, for all $P^C \in \mathcal{P}^C(\vec{G}_R)$, it follows that $\mathcal{I}(\vec{G}_R) \subseteq \mathcal{I}(P^C)$ from Lemma 35. Under the faithfulness assumption, there exists a conditional distribution P_*^C in $\mathcal{P}(\vec{G})^C$ such that $\mathcal{I}(\vec{G})^C = \mathcal{I}(P_*^C)$ by (b) of Proposition 5. Consider $\mathcal{P}(\vec{G})^C = \mathcal{P}^C(\vec{G}_R)$, which implies $P_*^C \in \mathcal{P}^C(\vec{G}_R)$ and thus $\mathcal{I}(\vec{G}_R) \subseteq \mathcal{I}(P_*^C)$ follows. Therefore, we can derive that $\mathcal{I}(\vec{G}_R) \subseteq \mathcal{I}(P_*^C) = \mathcal{I}(\vec{G})^C$. This combined with (a) of Proposition 37 yields that $\mathcal{I}(\vec{G})^C = \mathcal{I}(\vec{G}_R)$. ■

For any path l_{xy} in $\vec{G}$, we can write it as $l_{xy} = (x = c_0, \ldots, c_1, \ldots, c_{m+1} = y)$, where $V_<^c(l_{xy}) = \{c_1, c_2, \ldots, c_m\}$, denoting the set of ordered converging connection vertices in the direction from x to y of the path l_{xy} (If $V^c(l_{xy}) = \emptyset$, let $m = 0$). For simplicity, we also use $l_{xy} = \{x \rightarrow w \leftarrow y\}$ to represent a path $l_{xy} = (x, w, y)$ with $(x, w), (y, w)$ being directed edges in $\vec{G}$.

To prove Proposition 13, we first give three lemmas below.

Lemma 38 *Let $\vec{G} = (V, \vec{E})$ be a DAG and C be a subset of V. The following statements are equivalent:*

(a) $\mathcal{AV}_{\vec{G}}(C) = \emptyset$;

(b) $\mathcal{AP}_{\vec{G}}^o(C) = \emptyset$ or for all $x, y \in R$, there exists a trek $l'_{xy} \subseteq \vec{G}_R$ such that $V(l'_{xy}) \subseteq V(l_{xy})$ for any path $l_{xy} \in \mathcal{AP}_{\vec{G}}^o(C)$.

Proof. To prove (a) implies (b), since $\mathcal{AV}_{\vec{G}}(C) \subseteq \mathcal{AP}^o_{\vec{G}}(C)$, it holds naturally $\mathcal{AV}_{\vec{G}}(C) = \emptyset$ if $\mathcal{AP}^o_{\vec{G}}(C) = \emptyset$. For the case $\mathcal{AP}^o_{\vec{G}}(C) \neq \emptyset$, without loss of generality we can assume that $l_{xy} = (x, \ldots, u_1, c_1, v_1, \ldots, u_2, c_2, v_2, \ldots, u_m, c_m, v_m, \ldots, y)$, where $V^c_<(l_{xy}) = \{c_1, c_2, \ldots, c_m\} \subseteq \mathbf{An}_{\vec{G}}(C)$ and $(V(l_{xy}) \setminus V^c(l_{xy})) \cap C = \emptyset$. Note that $w \notin \mathbf{An}_{\vec{G}}(C)$ for all $\{u \to w \leftarrow v\} \in \mathcal{V}(\vec{G})$, it holds that $u_i \overset{\vec{G}}{\sim} v_i$ for $i \in [m]$. Since $V^c(l_{xy}) \subseteq \mathbf{An}_{\vec{G}}(C)$ and $(V(l_{xy}) \setminus V^c(l_{xy})) \cap C = \emptyset$, we have $V(l_{xy}) \setminus V^c(l_{xy}) \subseteq \mathbf{an}_{\vec{G}}(C) \subseteq R$. Thus, $l^1_{xy} = (x, \ldots, u_1, v_1, \ldots, u_2, v_2, \ldots, u_m, v_m, \ldots, y)$ is a path in $\vec{G}_R$ such that $V(l^1_{xy}) \subseteq V(l_{xy})$, which is obtained from l_{xy} by removing the elements in $V^c_<(l_{xy})$. If $V^c(l^1_{xy}) = \emptyset$, we are done. Otherwise, redefine l^1_{xy} as $l^1_{xy} = (x, \ldots, u^1_1, c^1_1, v^1_1, \ldots, u^1_2, c^1_2, v^1_2, \ldots, u^1_{m_1}, c^1_{m_1}, v^1_{m_1}, \ldots, y)$, where $V^c_<(l^1_{xy}) = \{c^1_1, c^1_2, \ldots, c^1_{m_1}\}$. Note that $V^c(l^1_{xy}) = V^c_<(l^1_{xy}) \subseteq \{\{u_i, v_i\} : i \in [m]\} \subseteq \mathbf{an}_{\vec{G}}(C) \subseteq \mathbf{An}_{\vec{G}}(C)$ and $(V^s(l^1_{xy}) \cup V^d(l^1_{xy})) \cap C \subseteq (V^s(l_{xy}) \cup V^d(l_{xy})) \cap C = \emptyset$, it follows that $l^1_{xy} \in \mathcal{AP}^o_{\vec{G}}(C)$. Analogously, we finally get a path l^k_{xy} in $\vec{G}_R$ such that $V^c(l^k_{xy}) = \emptyset$ and $V(l^k_{xy}) \subseteq V(l_{xy})$ for some positive integer k.

To prove (b) implies (a), suppose for contradiction that $\mathcal{AV}_{\vec{G}}(C) \neq \emptyset$. If $\mathcal{AP}^o_{\vec{G}}(C) = \emptyset$, a contradiction. Otherwise, for any $l_{xy} = \{x \to w \leftarrow y\} \in \mathcal{AP}^o_{\vec{G}}(C)$, it holds that x, y are adjacent in $\vec{G}$, also a contradiction. ■

Lemma 39 (Borboudakis and Tsamardinos, 2015) *Let $\vec{G} = (V, \vec{E})$ be a DAG and C be a subset of V. For any $x, y \in R$ such that $x \overset{\vec{G}}{\nsim} y$, the following statements are equivalent:*

(a) $x \not\perp\!\!\!\perp y | (S, C)[\vec{G}]$ for all $S \subseteq R \setminus \{x, y\}$;

(b) there exists $\{x \to w \leftarrow y\} \in \mathcal{AV}_{\vec{G}}(C)$, i.e. there exists a C-configuration with ending points x and y in $\vec{G}$.

Now we are ready to provide a characterization of the closure of the independence model of a BN under conditioning, in terms of conditional independence.

Lemma 40 *Let $\vec{G} = (V, \vec{E})$ be a DAG and C be a subset of V. Then $\mathcal{I}(\vec{G})^C = \mathcal{I}(\vec{G}_R)$ if and only if for any $x, y \in R$ such that $x \overset{\vec{G}}{\nsim} y$, there exists a subset S_0 of $R \setminus \{x, y\}$ such that $x \perp\!\!\!\perp y | (S_0, C)[\vec{G}]$.*

Proof. The necessity is trivial. We will give the proof of sufficiency. It follows from Lemma 39 that $\mathcal{AV}_{\vec{G}}(C) = \emptyset$. It suffices to show that $\langle X, Y | S \rangle \in \mathcal{I}(\vec{G})^C$ for any $\langle X, Y | S \rangle \in \mathcal{I}(\vec{G}_R)$. For any $x \in X$ and any $y \in Y$, it is enough to prove that l_{xy} is blocked by $S \cup C$ in $\vec{G}$ for any $l_{xy} \subseteq \vec{G}$. Suppose that there exists a path l_{xy} in $\vec{G}$ with $x \overset{\vec{G}}{\nsim} y$ such that it cannot be blocked by $S \cup C$ in $\vec{G}$. There are two cases to be considered. For the case $V^o(l_{xy}) \cap C = \emptyset$, consider that $\langle X, Y | S \rangle \in \mathcal{I}(\vec{G}_R)$, from which it follows that $x \perp\!\!\!\perp y | S[\vec{G}_R]$, which gives

$$(V^s(l_{xy}) \cup V^d(l_{xy})) \cap S \neq \emptyset \text{ or } V^c(l_{xy}) \nsubseteq \mathbf{An}_{\vec{G}_R}(S). \tag{12}$$

Based on the hypothesis that l_{xy} cannot be blocked by $S \cup C$ in $\vec{G}$, we know that $(V^s(l_{xy}) \cup V^d(l_{xy})) \cap (S \cup C) = \emptyset$ and $V^c(l_{xy}) \subseteq \mathbf{An}_{\vec{G}}(S \cup C)$. By applying (12), it follows that $\emptyset \neq V^c(l_{xy}) \subseteq \mathbf{An}_{\vec{G}}(C)$. Since $\mathcal{AV}_{\vec{G}}(C) = \emptyset$, there must exist a trek l'_{xy} in $\vec{G}_R$ such that

$V(l'_{xy}) \subseteq V(l_{xy})$ by Lemma 38. Therefore, we obtain a path l'_{xy} which satisfies $(V^s(l'_{xy}) \cup V^d(l'_{xy})) \cap S \neq \emptyset$, a contradiction.

For the case $V^o(l_{xy}) \cap C \neq \emptyset$, if $(V^s(l_{xy}) \cup V^d(l_{xy})) \cap C \neq \emptyset$, it contradicts our hypothesis. Hence, we can derive that $(V^s(l_{xy}) \cup V^d(l_{xy})) \cap C = \emptyset$. Further, if $V^c(l_{xy}) \cap C = \emptyset$, we obtain a contradiction with the hypothesis from (a). Hence, we have $V^c(l_{xy}) \cap C \neq \emptyset$. For simplicity, let $l_{xy} = (x, \ldots, u_1, c_1, v_1, \ldots, u_2, c_2, v_2, \ldots, u_m, c_m, v_m, \ldots, y)$, for which $V^c(l_{xy}) = \{c_1, c_2, \ldots, c_m\} \subseteq \mathbf{An}_{\vec{G}}(S \cup C)$ and $(V(l_{xy}) \setminus V^c(l_{xy})) \cap C = \emptyset$. Without loss of generality, we can assume that $V^c(l_{xy}) \cap C = \{c_1, c_2, \ldots, c_i\}, i \leq m$. Thus, l_{xy} can be rewritten as $l_{xy} = l_{xv_i} \cup l_{v_iy}$, where $l_{xv_i} = (x, \ldots, u_1, c_1, v_1, \ldots, u_2, c_2, v_2, \ldots, u_i, c_i, v_i)$ and $l_{v_iy} = (v_i, \ldots, u_{i+1}, c_{i+1}, v_{i+1}, \ldots, u_m, c_m, v_m, \ldots, y)$ in $\vec{G}_R$. Using Lemma 38 again, we know that there exists a trek $l^*_{u_1v_i}$ in $\vec{G}_R$ such that $V(l^*_{u_1v_i}) \subseteq V(l_{u_1v_i})$. This gives us a path $l^*_{xy} = l_{xu_1} \cup l^*_{u_1v_i} \cup l_{v_iy}$ in $\vec{G}_R$. Consider that $(V^s(l_{xy}) \cup V^d(l_{xy})) \cap (S \cup C) = \emptyset$ and $V^c(l_{xy}) \subseteq \mathbf{An}_{\vec{G}}(S \cup C)$, it holds that $(V^s(l^*_{xy}) \cup V^d(l^*_{xy})) \cap (S \cup C) = \emptyset$ and $V^c(l^*_{xy}) \subseteq \mathbf{An}_{\vec{G}}(S \cup C)$. These statements imply that l^*_{xy} cannot be blocked by S in $\vec{G}_R$, as desired. ■

C.2 Proof of Proposition 13

Proof. It holds obviously by Lemmas 38, 39 and 40. ■

C.3 Proof of Proposition 16

Proof. To prove (a), Proposition 37 yields $\mathcal{I}(\vec{G})^C \subseteq \mathcal{I}(\vec{G}_R)$. It suffices to prove that $\langle X, Y|S\rangle \in \mathcal{I}(\vec{G})^C$ for any $\langle X, Y|S\rangle \in \mathcal{I}(\vec{G}_R)$. For any $x \in X$ and any $y \in Y$, it is enough to show that l_{xy} is blocked by $S \cup C$ in $\vec{G}$ for any path l_{xy} in $\vec{G}$. There are two cases to be considered. If $l_{xy} \subseteq \vec{G}_R$, since $\langle X, Y|S\rangle \in \mathcal{I}(\vec{G}_R)$, it holds that $x \perp\!\!\!\perp y|S[\vec{G}_R]$. That C is a root set in $\vec{G}$ yields $V^c(l_{xy}) \cap \mathbf{An}_{\vec{G}}(C) = \emptyset$. Note that $\mathbf{An}_{\vec{G}_R}(S) \cup \mathbf{An}_{\vec{G}}(C) = \mathbf{An}_{\vec{G}}(S \cup C)$, we can derive that l_{xy} is blocked by $S \cup C$ in $\vec{G}$, as desired. If $l_{xy} \nsubseteq \vec{G}_R$, since $V^c(l_{xy}) \cap C = \emptyset$, it holds that $(V^s(l_{xy}) \cup V^d(l_{xy})) \cap C \neq \emptyset$, which gives that l_{xy} is blocked by $S \cup C$ in $\vec{G}$.

To prove (b), by Proposition 37, it holds that $\mathcal{P}^C(\vec{G}_R) \subseteq \mathcal{P}(\vec{G})^C$. It suffices to show that $P^C \in \mathcal{P}^C(\vec{G}_R)$ for any $P^C \in \mathcal{P}(\vec{G})^C$. From (a) of Proposition 5, it follows that $\mathcal{I}(\vec{G}_R) = \mathcal{I}(\vec{G})^C \subseteq \mathcal{I}(P^C)$ for any $P^C \in \mathcal{P}(\vec{G})^C$. Combining this with Lemma 35, we know that $\mathcal{P}(\vec{G})^C \subseteq \mathcal{P}^C(\vec{G}_R)$. ■

Appendix D. Proofs of Claims in Section 5

Before the proof of Theorem 18, we first need to prove two lemmas below.

Lemma 41 *Let $\vec{G} = (V, \vec{E})$ be a DAG with a topological order α and C be a subset of V. Suppose that $\triangle^\alpha_{\vec{G}}(R) \subseteq \vec{E}^\alpha_c(R)$. Then $\triangle^\alpha_{\vec{G}}(R)$ is a minimal filling edge set of $\vec{G}_R$ w.r.t. α if and only if $\mathcal{I}(\vec{G} + \triangle^\alpha_{\vec{G}}(R))$ is closed under conditioning on C but $\mathcal{I}(\vec{G} + \triangle^\alpha_{\vec{G}}(R) - (x, y))$ is not closed under conditioning on C for any edge $(x, y) \in \triangle^\alpha_{\vec{G}}(R)$.*

Proof. To prove the sufficiency, by contradiction, suppose that $\triangle^\alpha_{\vec{G}}(R)$ is not a minimal filling edge set of $\vec{G}_R$ w.r.t. α. By Proposition 13, we obtain that x, y are the endpoints of a C-configuration in DAG $\vec{G} + \triangle^\alpha_{\vec{G}}(R) - (x, y)$ and there are no C-configurations in $\vec{G} + \triangle^\alpha_{\vec{G}}(R)$.

There are two cases to be considered. If there exist $(x,y) \in \triangle^{\alpha}_{\vec{G}}(R)$ and $S_0 \subseteq R \setminus \{x,y\}$ such that $x \perp\!\!\!\perp y|(S_0, C)[\vec{G} + \triangle^{\alpha}_{\vec{G}}(R) - (x,y)]$, it follows that x or y is not an endpoint of a C-configuration in $\vec{G} + \triangle^{\alpha}_{\vec{G}}(R) - (x,y)$, a contradiction. If there exist $x, y \in R$ with $x \overset{\vec{G}+\triangle^{\alpha}_{\vec{G}}(R)}{\nsim} y$ such that $x \not\perp\!\!\!\perp y|(S,C)[\vec{G}+\triangle^{\alpha}_{\vec{G}}(R) - (x,y)]$ for all $S \subseteq R \setminus \{x,y\}$, implying that x, y are the endpoints of a C-configuration in $\vec{G} + \triangle^{\alpha}_{\vec{G}}(R)$, also a contradiction.

Conversely, we can see that x or y is not an endpoint of a C-configuration in $\vec{G}+\triangle^{\alpha}_{\vec{G}}(R)$ for any $x, y \in R$ with $x \overset{\vec{G}}{\nsim} y$, otherwise a contradiction. This implies that $\mathcal{I}(\vec{G} + \triangle^{\alpha}_{\vec{G}}(R))$ is closed under conditioning on C. For any $(x,y) \in \triangle^{\alpha}_{\vec{G}}(R)$, it follows from the definition of minimal filling edge set that x, y are the endpoints of a C-configuration in $\vec{G}+\triangle^{\alpha}_{\vec{G}}(R)-(x,y)$, which gives that $\mathcal{I}(\vec{G} + \triangle^{\alpha}_{\vec{G}}(R) - (x,y))$ is not closed under conditioning on C for any $(x,y) \in \triangle^{\alpha}_{\vec{G}}(R)$. ■

Lemma 42 *Let $\vec{G} = (V, \vec{E})$ be a DAG with a topological order α, and C be a subset of V. Suppose that $\triangle^{\alpha}_{\vec{G}}(R)$ is a minimal filling edge set of $\vec{G}_R$ w.r.t. α. Then, for any edge $(x,y) \in \triangle^{\alpha}_{\vec{G}}(R)$, we have that $x \not\perp\!\!\!\perp y|(\mathbf{pa}_{\vec{G}_R+\triangle^{\alpha}_{\vec{G}}(R)-(x,y)}(y), C)[\vec{G}]$.*

Proof. Since $(x,y) \in \triangle^{\alpha}_{\vec{G}}(R)$, there exists a C-configuration $\{x \to w \leftarrow y\}$ in $\vec{G}+\triangle^{\alpha}_{\vec{G}}(R) - (x,y)$ by Lemma 39. If $\{x \to w = w_1 \leftarrow y\} \in \mathcal{V}(\vec{G})$, there exists a path $l^1_{xy} = \{x \to w_1 \leftarrow y\}$ in $\mathcal{AP}^o_{\vec{G}}(C)$ such that $\alpha(z) < \alpha(y)$ for any $z \in V^o(l^1_{xy})$. Otherwise, there exists a C-configuration $\{x \to w^1_2 \leftarrow w_1\}$ or $\{y \to w^2_2 \leftarrow w_1\}$ in $\vec{G} + \triangle^{\alpha}_{\vec{G}}(R) - (x, w_1) - (y, w_1)$. This implies that there exists a path $l^2_{xy} = \{x \to w^1_2 \leftarrow w_1 \leftarrow y\}$ or $\{y \to w^2_2 \leftarrow w_1 \leftarrow x\}$ or $\{x \to w^1_2 \leftarrow w_1 \to w^2_2 \leftarrow y\}$ in $\mathcal{AP}^o_{\vec{G}}(C)$ such that $\alpha(z) < \alpha(y)$ for any $z \in V^o(l^2_{xy})$ if $\{x \to w^1_2 \leftarrow w_1\} \in \mathcal{V}(\vec{G})$ or $\{y \to w^2_2 \leftarrow w_1\} \in \mathcal{V}(\vec{G})$.

Analogously, we can derive that there exists a path $l^k_{xy} \in \mathcal{AP}^o_{\vec{G}}(C)$ such that $\alpha(z) < \alpha(y)$ for any $z \in V^o(l^k_{xy})$ for some integer k. By the acyclicity of $\vec{G} + \triangle^{\alpha}_{\vec{G}}(R) - (x,y)$, we arrive at $V^o(l^k_{xy}) \cap (\mathbf{pa}_{\vec{G}_R+\triangle^{\alpha}_{\vec{G}}(R)-(x,y)}(y) \cup C) = \emptyset$. From the definition of $\mathcal{AP}^o_{\vec{G}}(C)$, it follows that $(V^s(l^k_{xy}) \cup V^d(l^k_{xy})) \cap (\mathbf{pa}_{\vec{G}_R+\triangle^{\alpha}_{\vec{G}}(R)-(x,y)}(y) \cup C) = \emptyset$ and $V^c(l^k_{xy}) \subseteq \mathbf{An}_{\vec{G}}(C) \subseteq \mathbf{An}_{\vec{G}}(\mathbf{pa}_{\vec{G}_R+\triangle^{\alpha}_{\vec{G}}(R)-(x,y)}(y) \cup C)$. Thus, there exists a path l^k_{xy} being not blocked by $\mathbf{pa}_{\vec{G}^*_R}(y) \cup C$ in $\vec{G}$, where $\vec{G}^*_R = \vec{G}_R + \triangle^{\alpha}_{\vec{G}}(R) - (x,y)$. We complete the proof. ■

D.1 Proof of Theorem 18

Proof. First, Lemma 41 provides us with $\mathcal{I}(\vec{G}_R + \triangle^{\alpha}_{\vec{G}}(R)) = \mathcal{I}(\vec{G} + \triangle^{\alpha}_{\vec{G}}(R))^C$, which implies that $\vec{G}_R + \triangle^{\alpha}_{\vec{G}}(R)$ is an $\mathcal{I}$-map of $\mathcal{P}(\vec{G})^C$ by Proposition 5. It suffices to show that $\mathcal{I}(\vec{G}_R + \triangle^{\alpha}_{\vec{G}}(R) - (x,y)) \nsubseteq \mathcal{I}(\vec{G})^C$ for any $(x,y) \in \triangle^{\alpha}_{\vec{G}}(R)$. Since x, y are nonadjacent in $\vec{G}_R + \triangle^{\alpha}_{\vec{G}}(R) - (x,y)$ and $\alpha(x) < \alpha(y)$, we have that $x \perp\!\!\!\perp y|\mathbf{pa}_{\vec{G}_R+\triangle^{\alpha}_{\vec{G}}(R)-(x,y)}(y)[\vec{G}_R + \triangle^{\alpha}_{\vec{G}}(R) - (x,y)]$. By Lemma 42, we know that $x \not\perp\!\!\!\perp y|(\mathbf{pa}_{\vec{G}_R+\triangle^{\alpha}_{\vec{G}}(R)-(x,y)}(y), C)[\vec{G}]$, which gives $\mathcal{I}(\vec{G}_R + \triangle^{\alpha}_{\vec{G}}(R) - (x,y)) \nsubseteq \mathcal{I}(\vec{G})^C$, as desired.

Conversely, Proposition 5 yields that $\mathcal{I}(\vec{G}_R + \triangle^{\alpha}_{\vec{G}}(R)) \subseteq \mathcal{I}(\vec{G})^C$ and $\mathcal{I}(\vec{G}_R + \triangle^{\alpha}_{\vec{G}}(R) - (x,y)) \nsubseteq \mathcal{I}(\vec{G})^C$ for any $(x,y) \in \triangle^{\alpha}_{\vec{G}}(R)$. By Lemma 36, we have $\mathcal{I}(\vec{G} + \triangle^{\alpha}_{\vec{G}}(R) - (x,y))^C \subseteq \mathcal{I}(\vec{G})^C$, which implies $\mathcal{I}(\vec{G}_R + \triangle^{\alpha}_{\vec{G}}(R) - (x,y)) \nsubseteq \mathcal{I}(\vec{G} + \triangle^{\alpha}_{\vec{G}}(R) - (x,y))^C$. It follows from Lemma 41 that there exists a minimal filling edge set $\triangle^{'}_{\vec{G}}(R)$ of $\vec{G}_R$ w.r.t. α such that $\triangle^{'}_{\vec{G}}(R) \subseteq \triangle^{\alpha}_{\vec{G}}(R)$. Suppose for contradiction that $\triangle^{\alpha}_{\vec{G}}(R) \neq \triangle^{'}_{\vec{G}}(R)$. Using Lemma 41 again, it holds that $\mathcal{I}(\vec{G}_R + \triangle^{'}_{\vec{G}}(R)) = \mathcal{I}(\vec{G} + \triangle^{'}_{\vec{G}}(R))^C$. This implies that $\vec{G}_R + \triangle^{'}_{\vec{G}}(R)$ is an $\mathcal{I}$-map of $\mathcal{P}(\vec{G})^C$ by Proposition 5, a contradiction with that $\vec{G}_R + \triangle^{\alpha}_{\vec{G}}(R)$ is a minimal $\mathcal{I}$-map of $\mathcal{P}(\vec{G})^C$. ■

Here, we shall discuss a property on a minimal $\mathcal{I}$-map of $\mathcal{P}(\vec{G})^C$ w.r.t. any given topological order α.

Corollary 43 *Let $\mathcal{B} = (\vec{G}, \mathcal{P}(\vec{G}))$ be a BN with a DAG $\vec{G} = (V, \vec{E})$ and C be a subset of V. Suppose that $\mathcal{P}(\vec{G})$ is faithful to $\vec{G}$ and $\triangle^{\alpha}_{\vec{G}}(R) \subseteq \vec{E}^{\alpha}_c(R)$, where α is a topological order of $\vec{G}$. Let $\vec{G}^{\alpha} = \vec{G} + \triangle^{\alpha}_{\vec{G}}(R)$. If $\vec{G}_R + \triangle^{\alpha}_{\vec{G}}(R)$ is a minimal $\mathcal{I}$-map of $\mathcal{P}(\vec{G})^C$, then we find that*

(a) $\mathcal{I}(\vec{G}^{\alpha})^C = \mathcal{I}(\vec{G}^{\alpha}_R)$;

(b) $\mathcal{P}(\vec{G}^{\alpha})^C = \mathcal{P}^C(\vec{G}^{\alpha}_R)$.

Proof. Theorem 18 combined with Lemma 41 yields that $\mathcal{I}(\vec{G}^{\alpha})^C = \mathcal{I}(\vec{G}^{\alpha}_R)$. From Theorem 10 it follows that $\mathcal{P}(\vec{G}^{\alpha})^C = \mathcal{P}^C(\vec{G}^{\alpha}_R)$. ■

D.2 Proof of Theorem 19

Proof. According to Lemma 36, Theorem 18 and Corollary 43, we have that $\mathcal{P}(\vec{G})^C \subseteq \mathcal{P}(\vec{G}^{\alpha})^C = \mathcal{P}^C(\vec{G}^{\alpha}_R)$, implying that (a) holds. To prove (b), since $x, y \in \mathbf{an}_{\vec{G}}(C)$ for any $(x,y) \in \vec{E}^{\alpha}_R$, we know that $\mathbf{an}_{\vec{G}^{\alpha}}(C) = \mathbf{an}_{\vec{G}}(C)$ and $\mathbf{ch}_{\vec{G}^{\alpha}}(C) = \mathbf{ch}_{\vec{G}}(C)$. Thus it suffices to consider the factorization of P^C by the partition $\mathbf{an}_{\vec{G}}(C) \cup \mathbf{ch}_{\vec{G}}(C) \cup Z$ of R for any $P^C \in \mathcal{P}(\vec{G})^C \subseteq \mathcal{P}(\vec{G}^{\alpha})^C$. For any $w \in \mathbf{ch}_{\vec{G}}(C) \setminus \mathbf{an}_{\vec{G}}(C)$, it holds that $C_w \subseteq \mathbf{nd}_{\vec{G}^{\alpha}}(w)$, where $C_w = C \cup \mathbf{pa}_{\vec{G}^{\alpha}_R}(w) \setminus \mathbf{pa}_{\vec{G}^{\alpha}}(w)$. From which we know that $w \perp\!\!\!\perp C_w | \mathbf{pa}_{\vec{G}^{\alpha}}(w)[\vec{G}^{\alpha}]$. Since $\mathcal{I}(\vec{G}^{\alpha}) \subseteq \mathcal{I}(\vec{G}) \subseteq \mathcal{I}(P)$ for any $P \in \mathcal{P}(\vec{G})$, it holds that $X_w \perp\!\!\!\perp X_{C_w} | X_{\mathbf{pa}_{\vec{G}^{\alpha}}(w)}[P]$ for any $w \in \mathbf{ch}_{\vec{G}}(C) \setminus \mathbf{an}_{\vec{G}}(C)$. This implies that for any $w \in \mathbf{ch}_{\vec{G}}(C) \setminus \mathbf{an}_{\vec{G}}(C)$,

$$\begin{aligned} P^C(x_w | x_{\mathbf{pa}_{\vec{G}^{\alpha}_R}(w)}) &= P(x_w | x_{\mathbf{pa}_{\vec{G}^{\alpha}}(w)}, x_{C_w}) \\ &= P(x_w | x_{\mathbf{pa}_{\vec{G}^{\alpha}}(w)}) \\ &= P(x_w | x_{\mathbf{pa}_{\vec{G}^{\alpha}_R}(w)}, x_{C \cap \mathbf{pa}_{\vec{G}^{\alpha}}(w)}). \end{aligned}$$

For any $w \in \mathbf{an}_{\vec{G}}(C) \setminus \mathbf{ch}_{\vec{G}}(C)$, since $\mathbf{pa}_{\vec{G}^{\alpha}}(w) = \mathbf{pa}_{\vec{G}^{\alpha}_R}(w)$, we know that

$$P^C(x_w | x_{\mathbf{pa}_{\vec{G}^{\alpha}_R}(w)}) = P(x_w | x_{\mathbf{pa}_{\vec{G}^{\alpha}_R}(w)}, x_C) = P(x_w | x_{\mathbf{pa}_{\vec{G}^{\alpha}_R}(w)}, x_{C \cap \mathbf{de}_{\vec{G}^{\alpha}}(w)}).$$

For any $w \in \mathbf{an}_{\vec{G}}(C) \cap \mathbf{ch}_{\vec{G}}(C)$, we know that

$$P^C(x_w | x_{\mathbf{pa}_{\vec{G}^{\alpha}_R}(w)}) = P(x_w | x_{\mathbf{pa}_{\vec{G}^{\alpha}_R}(w)}, x_{C \cap \mathbf{pa}_{\vec{G}^{\alpha}}(w)}, x_{C \cap \mathbf{de}_{\vec{G}^{\alpha}}(w)}).$$

Let $\mathbf{de}^o_{\vec{G}^\alpha}(w) = \{u \in V \setminus w : \exists\ \vec{l}_{wu} \subseteq \vec{G}^\alpha, s.t.\ V^o(\vec{l}_{wu}) \cap C = \emptyset\}$. Note that $w \perp\!\!\!\perp \mathbf{de}_{\vec{G}^\alpha}(w) \setminus \mathbf{de}^o_{\vec{G}^\alpha}(w) | \mathbf{de}^o_{\vec{G}^\alpha}(w)[\vec{G}]$, for any $w \in \mathbf{an}_{\vec{G}}(C) \cup \mathbf{ch}_{\vec{G}}(C)$, we have

$$\begin{aligned} P^C(x_w | x_{\mathbf{pa}_{\vec{G}^\alpha_R}(w)}) &= P(x_w | x_{\mathbf{pa}_{\vec{G}^\alpha_R}(w)}, x_{C \cap \mathbf{pa}_{\vec{G}^\alpha}(w)}, x_{C \cap \mathbf{de}^o_{\vec{G}^\alpha}(w)}) \\ &= P(x_w | x_{\mathbf{pa}_{\vec{G}^\alpha}(w)}, x_{C \cap \mathbf{de}^o_{\vec{G}^\alpha}(w)}). \end{aligned}$$

Since $C \subseteq \mathbf{nd}_{\vec{G}^\alpha}(w)$ and $C \cap \mathbf{pa}_{\vec{G}^\alpha}(w) = \emptyset$ for any $w \in Z$, it follows that

$$\prod_{w \in Z} P^C(x_w | x_{\mathbf{pa}_{\vec{G}^\alpha_R}(w)}) = \prod_{w \in Z} P(x_w | x_{\mathbf{pa}_{\vec{G}^\alpha_R}(w)}, x_C) = \prod_{w \in Z} P(x_w | x_{\mathbf{pa}_{\vec{G}_R}(w)}).$$

Combining these with $P^C(x_R) = \prod_{w \in R} P^C(x_w | x_{\mathbf{pa}_{\vec{G}^\alpha_R}}(w))$, we complete the proof. ■

Before the proof of of Proposition 22, we first list certain trivial properties of the converging vertices sweeping operator without proof. Lemma 44 indicates that the sweeping operator works only on a v-path.

Lemma 44 *Suppose that $\vec{G} = (V, \vec{E})$ is a DAG. For any path l_{xy} in $\vec{G}$, the following results are equivalent:*

(a) $V^c(l_{xy}) = \emptyset$;

(b) $\mathcal{T}(l_{xy}) = l_{xy}$;

(c) $\rho(\mathcal{T}(l_{xy})) = \rho(l_{xy})$.

D.3 Proof of Proposition 22

Proof. (if) From Lemma 44 and the definition of $\mathcal{T}$, it follows that $\mathcal{T}^\infty(l_{xy}) = \vec{l}_{xy}$ or $\vec{l}_{wx} \cup \vec{l}_{wy}$ for some $w \in V^s(l_{xy}) \cup V^d(l_{xy})$, where $V(\vec{l}_{xy}) \subseteq V(l_{xy})$ and $V(\vec{l}_{wx} \cup \vec{l}_{wy}) \subseteq V(l_{xy})$. Note that $\alpha(x) < \alpha(y)$ and $y = \arg\min_{w^* \in V(l_{xy}) \setminus \{x\}} \{\alpha(w^*)\}$, it holds that $\mathcal{T}^\infty(l_{xy}) = \vec{l}_{xy}$ and $V^o(l_{xy}) = \emptyset$, which gives $\mathcal{T}^\infty(l_{xy}) = (x, y)$.

(only if) Since l_{xy} has the TCP w.r.t. α and $\alpha(x) < \alpha(y)$, $\mathcal{T}^\infty(l_{xy}) = \mathcal{T}^{N_0}(l_{xy}) = \mathcal{T}^{N_0+1}(l_{xy}) = (x, y)$ holds. This implies that l_{xy} is an o-o path and $\mathcal{T}^{N_0-1}(l_{xy}) = \vec{l}_{xv} \cup \vec{l}_{yv}$ for some $v \in V^s(l_{xy}) \cup V^d(l_{xy})$. Thus $x = \arg\min_{w \in V(\mathcal{T}^{N_0-1}(l_{xy})) \setminus \{y\}} \{\alpha(w)\}$.

Without loss of generality we assume that $l_{xy} = \bigcup_{i=0}^m l_{d_i d_{i+1}}$, where $d_0 = x, d_{m+1} = y$ and $V^d_<(l_{xy}) = \{d_1, d_2, \ldots, d_m\}$. Then $\mathcal{T}(l_{xy}) = \bigcup_{i=0}^m \mathcal{T}(l_{d_i d_{i+1}}) = \bigcup_{i=0}^m l^r_{d_i d_{i+1}}$. From which we can derive that $V(\mathcal{T}(l_{d_i d_{i+1}})) \subseteq V(l_{d_i d_{i+1}})$, implying $V(\mathcal{T}(l_{xy})) \subseteq V(l_{xy})$. By the definition of representation path, we have that $\alpha(u) > \max\{\alpha(d_i), \alpha(d_{i+1})\}$ for any $u^* \in V(\mathcal{T}(l_{d_i d_{i+1}}))$ and $\alpha(u^*) < \max\{\alpha(d_i), \alpha(d_{i+1})\}$ for any $u^* \in V(\mathcal{T}(l_{d_i d_{i+1}}))$. As a result, for any $u^* \in V(\mathcal{T}(l_{d_i d_{i+1}}))$ and any $u \in V(l_{d_i d_{i+1}}) \setminus V(\mathcal{T}(l_{d_i d_{i+1}}))$, we have that $\alpha(u^*) < \alpha(u)$. From which it follows that $\alpha(u^*) < \alpha(u)$ for any $u^* \in V(\mathcal{T}(l_{xy}))$ and any $u \in V(l_{xy}) \setminus V(\mathcal{T}(l_{xy}))$.

Analogously, for any $i \in [N_0 - 1]$, we can obtain that $V(\mathcal{T}^i(l_{xy})) \subseteq V(\mathcal{T}^{i-1}(l_{xy}))$ and $\alpha(u^*) < \alpha(u)$ for any $u^* \in V(\mathcal{T}^i(l_{xy}))$ and any $u \in V(\mathcal{T}^{i-1}(l_{xy})) \setminus V(\mathcal{T}^i(l_{xy}))$. This combined with $y = \arg\min_{w \in V(\mathcal{T}^{N_0-1}(l_{xy})) \setminus \{x\}} \{\alpha(w)\}$ yields that $y = \arg\min_{w \in V(l_{xy}) \setminus \{x\}} \{\alpha(w)\}$. ■

To prove Proposition 25, we first give Lemma 45.

Lemma 45 *Suppose that $\vec{G} = (V, \vec{E})$ is a DAG with a topological order α, and C is a subset of V. For any $x, y \in R$ such that $x \overset{\vec{G}}{\nsim} y$, $x \overset{\alpha}{\sim} y \in \triangle^{tc}_{\vec{G}}(C;\alpha)$ if and only if x, y are the endpoints of a C-configuration in $\vec{G} + \triangle^{tc}_{\vec{G}}(C;\alpha) - x \overset{\alpha}{\sim} y$.*

Proof. Without loss of generality, suppose that $\alpha(x) < \alpha(y)$. (if) Lemma 39 yields that there exists $\{x \to w \leftarrow y\} \in \mathcal{AV}_{\vec{G}^*}(C)$, where $\vec{G}^* = \vec{G} + \triangle^{tc}_{\vec{G}}(C;\alpha) - (x, y)$. By contradiction, suppose that $(x, y) \notin \triangle^{tc}_{\vec{G}}(C;\alpha)$. Then we have $\mathcal{T}^{\infty}(l_{xy}) \neq (x, y)$ for any path $l_{xy} \in \mathcal{AP}^{o}_{\vec{G}}(C)$ with $x \overset{\vec{G}}{\nsim} y$. There are two cases to be considered:

1. If $(x, w) \in \vec{E}$ and $(y, w) \in \vec{E}$, there exists a path $l_{xy} = \{x \to w \leftarrow y\} \in \mathcal{AP}^{o}_{\vec{G}}(C)$. By Proposition 22, $\mathcal{T}^{\infty}(l_{xy}) = (x, y)$ holds, a contradiction.

2. Otherwise, there exists $l_{xw} \in \mathcal{AP}^{o}_{\vec{G}}(C)$ such that $w = \arg\min_{w^* \in V(l_{wx}) \setminus \{x\}}\{\alpha(w^*)\}$ and $\alpha(w) > \alpha(x)$ or there exists $l_{wy} \in \mathcal{AP}^{o}_{\vec{G}}(C)$ such that $w = \arg\min_{w^* \in V(l_{wy}) \setminus \{y\}}\{\alpha(w^*)\}$ and $\alpha(w) > \alpha(y)$. From which we can derive that there exists $l_{xy} = l_{xw} \cup (y, w)$ such that $y = \arg\min_{w \in V(l_{xy}) \setminus \{x\}}\{\alpha(w)\}$ or there exists $l_{xy} = (x, w) \cup l_{wy}$ such that $\alpha(y) < \alpha(w^*)$ for any $w^* \in V(l_{xy}) \setminus \{x\}$ or there exists $l_{xy} = l_{xw} \cup l_{wy} \in \mathcal{AP}^{o}_{\vec{G}}(C)$ such that $\alpha(y) < \alpha(w^*)$ for any $w^* \in V(l_{xy}) \setminus \{x\}$, a contradiction.

(only if) Since $(x, y) \in \triangle^{tc}_{\vec{G}}(C;\alpha)$, we know that there exists l_{xy} in $\mathcal{AP}^{o}_{\vec{G}}(C)$ such that $y = \arg\min_{w \in V(l_{xy}) \setminus \{x\}}\{\alpha(w)\}$. There are two cases to be considered:

1. If $V^o(l_{xy}) = \emptyset$, it follows from Lemma 39 that x, y are the endpoints of a C-configuration in $\vec{G} + \triangle^{tc}_{\vec{G}}(C;\alpha) - (x, y)$.

2. Otherwise, without loss of generality we can assume $u = \arg\min_{w \in V(l_{xy}) \setminus \{x,y\}}\{\alpha(w)\}$. Then there exist $l_{xu} \in \mathcal{AP}^{o}_{\vec{G}}(C)$ such that $u = \arg\min_{w \in V(l_{xu}) \setminus \{x\}}\{\alpha(w)\}$ and $\alpha(u) > \alpha(x)$, and $l_{uy} \in \mathcal{AP}^{o}_{\vec{G}}(C)$ such that $\alpha(z) > \alpha(u) > \alpha(y)$ for any $z \in V^o(l_{uy})$, respectively. This implies that $\mathcal{T}^{\infty}(l_{xu}) = (x, u) \in \vec{E}(\vec{G} + \triangle^{tc}_{\vec{G}}(C;\alpha))$ and $\mathcal{T}^{\infty}(l_{uy}) = (y, u) \in \vec{E}(\vec{G} + \triangle^{tc}_{\vec{G}}(C;\alpha))$, which gives $\{x \to u \leftarrow y\} \in \mathcal{V}(\vec{G}^*)$. Since $u \in \mathbf{An}_{\vec{G}^*}(C)$, we have that x, y are the endpoints of a C-configuration in $\vec{G} + \triangle^{tc}_{\vec{G}}(C;\alpha) - (x, y)$.

■

D.4 Proof of Proposition 25

Proof. It follows directly from Definition 17 and Lemma 45. ■

The following corollary shows that we need only to search in $\mathbf{an}_{\vec{G}}(C)$ in order to find $\triangle^{tc}_{\vec{G}}(C;\alpha)$.

Corollary 46 *Suppose that $\vec{G} = (V, \vec{E})$ is a DAG with a topological order α and C is a subset of V. For any edge $(x, y) \in \triangle^{tc}_{\vec{G}}(C;\alpha)$, we find that $\alpha(x) < \alpha(y)$ and $x, y \in \mathbf{an}_{\vec{G}}(C)$.*

Proof. We need to prove that $x, y \in \mathbf{an}_{\vec{G}}(C)$ since $\alpha(x) < \alpha(y)$ is trivial. By Propositions 22 and 25, there exists a path $l_{xy} \in \mathcal{AP}^{o}_{\vec{G}}(C)$ for any $(x, y) \in \triangle^{tc}_{\vec{G}}(C;\alpha)$. This implies that there exists an o-o path l_{xy} such that $\emptyset \neq V^c(l_{xy}) \subseteq \mathbf{An}_{\vec{G}}(C)$, which gives $x, y \in \mathbf{an}_{\vec{G}}(C)$. ■

Before proving Theorem 26, we first need to present Lemma 47.

Lemma 47 *Let $\vec{G} = (V, \vec{E})$ be a DAG and C be a subset of V. For any subset C' of C and any subset S of $R \setminus \{x, y\}$, we have $\mathbf{An}_{\vec{G}^{C'}}(S \cup C) = \mathbf{An}_{\vec{G}}(S \cup C)$.*

Proof. For any $S \subseteq R \setminus \{x, y\}$, it is trivial that $\mathbf{An}_{\vec{G}^{C'}}(S \cup C) \subseteq \mathbf{An}_{\vec{G}}(S \cup C)$. We need to show that $\mathbf{An}_{\vec{G}}(S \cup C) \subseteq \mathbf{An}_{\vec{G}^{C'}}(S \cup C)$. Suppose that there exists $S_0 \subseteq R \setminus \{x, y\}$ such that $\mathbf{An}_{\vec{G}}(S_0 \cup C) \nsubseteq \mathbf{An}_{\vec{G}^{C'}}(S_0 \cup C)$. Then there exists $w \in \mathbf{An}_{\vec{G}}(S_0 \cup C)$ but $w \notin \mathbf{An}_{\vec{G}^{C'}}(S_0 \cup C)$. Frow which we derive that there exists a directed path $\vec{l}_{wu}$ such that $u \in S_0 \cup C$ and $V^o(\vec{l}_{wu}) \cap C' \neq \emptyset$ in $\vec{G}$, i.e., there exists a directed path $\vec{l}_{wc}$ in $\vec{G}^{C'}$ such that $V(\vec{l}_{wc}) \subseteq V(\vec{l}_{wu})$ and $V^o(\vec{l}_{wc}) \cap C' = \emptyset$, a contradiction with $w \notin \mathbf{An}_{\vec{G}^{C'}}(S_0 \cup C)$. ■

D.5 Proof of Theorem 26

Proof. To prove (a), first of all, Lemma 36 together with $\vec{G}^{C'} \subseteq \vec{G}$ yields $\mathcal{I}(\vec{G})^C \subseteq \mathcal{I}(\vec{G}^{C'})^C$. So, it suffices to prove that $\langle X, Y|S\rangle \in \mathcal{I}(\vec{G})^C$ for any $\langle X, Y|S\rangle \in \mathcal{I}(\vec{G}^{C'})^C$. For any $x \in X$ and $y \in Y$, let l_{xy} be an arbitrary path connecting x and y in $\vec{G}$.

If $l_{xy} \subseteq \vec{G}^{C'}$, it follows that l_{xy} is blocked by $S \cup C$ in $\vec{G}^{C'}$ since $\langle X, Y|S\rangle \in \mathcal{I}(\vec{G}^{C'})^C$. This combined with Lemma 47 yields that l_{xy} is blocked by $S \cup C$ in $\vec{G}$.

Otherwise, there exist $c_1, c_2 \in C'$ such that $(c_1, c_2) \subseteq l_{xy}$ or $c_1 \in C', u \in R$ such that $(c_1, u) \subseteq l_{xy}$. Thus we have $c_1 \in V^s(l_{xy}) \cup V^d(l_{xy})$, which gives that l_{xy} is blocked by $S \cup C$ in $\vec{G}$. As a consequence, $\mathcal{I}(\vec{G}^{C'})^C \subseteq \mathcal{I}(\vec{G})^C$ holds.

To prove (b), it is trivial that $\mathcal{P}(\vec{G}^{C'})^C \subseteq \mathcal{P}(\vec{G})^C$. What we need is to prove $\mathcal{P}(\vec{G})^C \subseteq \mathcal{P}(\vec{G}^{C'})^C$. For any $P(x_R|x_C) \in \mathcal{P}(\vec{G})^C$, without loss of generality we can assume that $P(x_C) > 0$ for any fixed $x_C \in \mathcal{X}_C$, then we have that $P(x_R|X_C = x_C) = \frac{P(x_V)}{P(x_C)} = c_1 \prod_{w\in V} P(x_w|x_{\mathbf{pa}_{\vec{G}}(w)})$, where $c_1 = \frac{1}{P(x_C)}$ is a constant. For any $w \in V$, there are two cases to be considered:

1. If $\{w\} \cup \mathbf{pa}_{\vec{G}}(w) \subseteq C$, let $c_w = P(X_w = x_w|X_{\mathbf{pa}_{\vec{G}}(w)} = x_{\mathbf{pa}_{\vec{G}}(w)})$, where c_w is a constant. Let $A_1 = \{w \in V : \{w\} \cup \mathbf{pa}_{\vec{G}}(w) \subseteq C\}$. It follows that $\prod_{w\in A_1} c_w = \prod_{w\in A_1} P(X_w = x_w|X_{\mathbf{pa}_{\vec{G}}(w)\cap C} = x_{\mathbf{pa}_{\vec{G}}(w)\cap C}) = \prod_{w\in A_1} \phi(X_w = x_w)$.

2. Otherwise, there are three cases to be considered. Let $A_2 = \{w \in V : w \in R, \mathbf{pa}_{\vec{G}}(w) \subseteq C\}$, $A_3 = \{w \in V : w \in R, \mathbf{pa}_{\vec{G}}(w) \nsubseteq C\}$ and $A_4 = \{w \in V : w \in C, \mathbf{pa}_{\vec{G}}(w) \nsubseteq C\}$. Then we have that $\prod_{w\in A_2} P(x_w|X_{\mathbf{pa}_{\vec{G}}(w)} = x_{\mathbf{pa}_{\vec{G}}(w)}) = \prod_{w\in A_2} P(x_w|X_{\mathbf{pa}_{\vec{G}}(w)\cap C} = x_{\mathbf{pa}_{\vec{G}}(w)\cap C}) = \prod_{w\in A_2} \phi(x_w)$,

$$\prod_{w\in A_3} P(x_w|x_{\mathbf{pa}_{\vec{G}^{C'}}(w)}, X_{\mathbf{pa}_{\vec{G}}(w)\cap C} = x_{\mathbf{pa}_{\vec{G}}(w)\cap C}) = \prod_{w\in A_3} \phi(x_w|x_{\mathbf{pa}_{\vec{G}^{C'}}(w)}) \text{ and}$$

$$\prod_{w\in A_4} P(X_w = x_w|x_{\mathbf{pa}_{\vec{G}^{C'}}(w)}, X_{\mathbf{pa}_{\vec{G}}(w)\cap C} = x_{\mathbf{pa}_{\vec{G}}(w)\cap C}) = \prod_{w\in A_4} \phi(X_w = x_w|x_{\mathbf{pa}_{\vec{G}^{C'}}(w)}).$$

Thus, there exists $\phi(x_V) \in \mathcal{P}(\vec{G}^{C'})$ such that $\phi(x_R|X_C = x_C) = P(x_R|X_C = x_C)$, which gives $\mathcal{P}(\vec{G})^C \subseteq \mathcal{P}(\vec{G}^{C'})^C$. ■

D.6 Proof of Lemma 27

Proof. Let $A_1 = \{x, y \in R : x \overset{\vec{G}}{\not\sim} y\}$ and $A_2 = \{x, y \in R : x \overset{\vec{G}^C}{\not\sim} y\}$. It follows from the definition of $\vec{G}^C$ that $A_1 = A_2$ and $\vec{G}^C$ also has a topological order α. By Propositions 22 and 25, it is enough to prove $\mathcal{AP}^o_{\vec{G}}(C) = \mathcal{AP}^o_{\vec{G}^C}(C)$. Using Lemma 49, we complete the proof. ■

D.7 Proof of Proposition 29

Proof. Since the most complex part of the algorithm is Step 5, we can use the complexity of Step 5 to represent the complexity of the whole algorithm. In the worst case, we have that $|P_j| = |\mathbf{pa}_{\vec{G}_{j-1}}(u_j)| = k - j$ for $j \in [k-2]$. Let m be the maximal number of elements in P_j for $j \in [k-2]$. To find $\vec{E}_j$ for all $j \in [k-2]$, we have to check no more than $|k-2|\binom{m}{2}$ vertex pairs except the vertex pairs in C. Hence the complexity of the algorithm can be calculated by $|k-2|\binom{m}{2} - |C|(|C|-1)/2$. Thus we obtain that the complexity of the algorithm is at most $O(m^2|k-2|)$. ■

Appendix E. Proofs of Claims in Section 6

For any $x, y \in \mathbf{an}_{\vec{G}}(C)$ with $x \overset{\vec{G}}{\not\sim} y$, Proposition 48 implies that conditioning on C does not change qualitatively the information transitivity between x and y since $\{x, y\}$ is a cp-pair w.r.t. α in $\vec{G}$. That is, if $x \perp\!\!\!\perp y|(\mathbf{pa}_{\vec{G}_R}(\{x,y\}), C)[\vec{G}]$, we have that $\{x, y\}$ is a cp-pair under conditioning on C w.r.t. α in $\vec{G}$. Thus, we do not need to make such pairs adjacent after conditioning on C for a minimal $\mathcal{I}$-map.

Proposition 48 *Let $\vec{G} = (V, \vec{E})$ be a DAG with a topological order α and C be a subset of V. For any $x, y \in \mathbf{an}_{\vec{G}}(C)$, $\{x, y\}$ is a cp-pair in $\vec{G}$ if $x \perp\!\!\!\perp y|(\mathbf{pa}_{\vec{G}_R}(\{x,y\}), C)[\vec{G}]$.*

Proof. Suppose for contradiction that there exists $l_{xy} \in \mathcal{AP}^o_{\vec{G}}(C)$ such that l_{xy} has the TCP w.r.t. α. From Proposition 22 it follows $\alpha(w) > \max\{\alpha(x), \alpha(y)\}$ for any $w \in V^o(l_{xy})$. Note that $x \perp\!\!\!\perp y|(\mathbf{pa}_{\vec{G}_R}(\{x,y\}), C)[\vec{G}]$, we know that for any path l'_{xy} in $\vec{G}$, $(V^s(l'_{xy}) \cup V^d(l'_{xy})) \cap (\mathbf{pa}_{\vec{G}_R}(\{x,y\}) \cup C) \neq \emptyset$ or $V^c(l'_{xy}) \nsubseteq \mathbf{An}_{\vec{G}}(\mathbf{pa}_{\vec{G}_R}(\{x,y\}) \cup C)$. Since $l_{xy} \in \mathcal{AP}^o_{\vec{G}}(C)$, we have that $(V^s(l_{xy}) \cup V^d(l_{xy})) \cap C = \emptyset$ and $V^c(l_{xy}) \subseteq \mathbf{An}_{\vec{G}}(C)$. Thus, $(V^s(l_{xy}) \cup V^d(l_{xy})) \cap \mathbf{pa}_{\vec{G}_R}(\{x,y\}) \neq \emptyset$, a contradiction with $y = \arg\min_{w \in V(l_{xy}) \setminus \{x\}}\{\alpha(w)\}$. ■

To prove Proposition 50, we first give a lemma below.

Lemma 49 *Let $\vec{G} = (V, \vec{E})$ be a DAG and C be a subset of V. Then $\mathcal{AP}^o_{\vec{G}}(C) = \mathcal{AP}^o_{\vec{G}^C}(C)$.*

Proof. For any l_{xy} in $\mathcal{AP}^o_{\vec{G}}(C)$, we know that l_{xy} is an o-o path such that it is not blocked by C in $\vec{G}$. From the definition of $\vec{E}^C$, it follows that $\mathbf{An}_{\vec{G}}(C) = \mathbf{An}_{\vec{G}^C}(C)$ and $u \in C$ for any $(u, v) \in \vec{E}^C$. Thus $l_{xy} \in \mathcal{AP}^o_{\vec{G}^C}(C)$, which gives $\mathcal{AP}^o_{\vec{G}}(C) \subseteq \mathcal{AP}^o_{\vec{G}^C}(C)$.

Next we will show $\mathcal{AP}^o_{\vec{G}^C}(C) \subseteq \mathcal{AP}^o_{\vec{G}}(C)$. For any $l_{xy} \in \mathcal{AP}^o_{\vec{G}^C}(C)$, it holds that l_{xy} is an o-o path in $\vec{G}$ such that $(V^s(l_{xy}) \cup V^d(l_{xy})) \cap C = \emptyset$ and $V^c(l_{xy}) \subseteq \mathbf{An}_{\vec{G}^C}(C)$. Note that $\mathbf{An}_{\vec{G}^C}(C) = \mathbf{An}_{\vec{G}}(C)$, we have that $l_{xy} \in \mathcal{AP}^o_{\vec{G}}(C)$, implying $\mathcal{AP}^o_{\vec{G}^C}(C) \subseteq \mathcal{AP}^o_{\vec{G}}(C)$. ■

Proposition 50 *Suppose that $\vec{G} = (V, \vec{E})$ is a DAG with a topological order α, and C is a subset of V. Then $\mathcal{PS}^{cp}_{\vec{G}}(C;\alpha) = \mathcal{PS}^{cp}_{\vec{G}^C}(C;\alpha)$.*

Proof. It follows that $\mathcal{AP}^o_{\vec{G}^C}(C) = \mathcal{AP}^o_{\vec{G}}(C)$ from Lemma 49. For any $x, y \in R$, note that $x \overset{\vec{G}}{\sim} y$ if and only if $x \overset{\vec{G}^C}{\sim} y$, we can derive that $\mathcal{PS}^{cp}_{\vec{G}}(C;\alpha) = \mathcal{PS}^{cp}_{\vec{G}^C}(C;\alpha)$ by the definition of cp-pair. ■

Proposition 50 shows us that it is equivalent to considering the cp-pairs in $\vec{G}^C$ instead of those in the original graph $\vec{G}$, which is more convenient for verifying the set of cp-pairs under conditioning on C in $\vec{G}$.

Next, we will give certain simple properties of C-completeness.

Corollary 51 *Let $\vec{G} = (V, \vec{E})$ be a DAG with a topological order α and C be a subset of V. Then $\mathbf{an}_{\vec{G}}(C)$ is C-complete w.r.t. α in $\vec{G}$ if and only if $\mathbf{an}_{\vec{G}^C}(C)$ is C-complete w.r.t. α in $\vec{G}^C$.*

Proof. For any $x, y \in R$, we know that $x \overset{\vec{G}}{\sim} y$ if and only if $x \overset{\vec{G}^C}{\sim} y$. This combined with Proposition 50 completes the proof. ■

Proposition 52 *Let $\vec{G} = (V, \vec{E})$ be a DAG with a topological order α, and C be a subset of V. For $A \subseteq D \subseteq \mathbf{an}_{\vec{G}}(C)$, if D is C-complete w.r.t. α in $\vec{G}$, then A is C-complete w.r.t. α in $\vec{G}$.*

Proof. Suppose for contradiction that A is not C-complete w.r.t. α in $\vec{G}$. Then there exist $x, y \in A$ such that $x \overset{\vec{G}}{\nsim} y$ and $\{x, y\} \notin \mathcal{PS}^{cp}_{\vec{G}}(C;\alpha)$, a contradiction. ■

E.1 Proof of Proposition 31

Proof. To prove the necessity, suppose for contradiction that $\mathbf{an}_{\vec{G}}(C)$ is not C-complete w.r.t. α in $\vec{G}$. Then there exist $x, y \in \mathbf{an}_{\vec{G}}(C)$ with $x \overset{\vec{G}}{\nsim} y$ such that $\{x, y\} \notin \mathcal{PS}^{cp}_{\vec{G}}(C;\alpha)$, implying there exists $l_{xy} \in \mathcal{AP}^o_{\vec{G}}(C)$ such that l_{xy} has the TCP w.r.t. α. Proposition 22 yields that $\alpha(w) > \max\{\alpha(x), \alpha(y)\}$ for any $w \in V^o(l_{xy})$, a contradiction with Lemma 38 and Proposition 13.

Conversely, since $\mathbf{an}_{\vec{G}}(C)$ is C-complete w.r.t. α in $\vec{G}$, it holds that $\mathcal{PS}^{tc}_{\vec{G}}(C;\alpha) = \emptyset$. Combining this with Theorem 18 and Proposition 25, we know that $\mathcal{I}(\vec{G})^C = \mathcal{I}(\vec{G}_R)$. ■

Corollary 53 reminds us of the marginal collapsibility condition of the undirected graph models. Particularly, for an undirected graph model $(G, \mathcal{P}(G))$ with an undirected graph $G = (V, E)$, under the condition that $\mathcal{P}$ is closed under marginalization, i.e., the joint and marginal distributions belong to the same type of distribution. $\mathcal{P}(G)$ or G is collapsible onto $V \setminus A$ if and only if for each connected component A^* of A, the neighbors of A^*, denoted by $\mathbf{ne}_G(A^*)$, is complete in G, that is, $\mathbf{ne}_G(A^*) \setminus A$ is complete in G since $\mathbf{ne}_G(A^*) \cap A = \emptyset$.

Corollary 53 *Let $\vec{G} = (V, \vec{E})$ be a DAG with a topological order α and C be a subset of V. Then the following statements are equivalent:*

(a) $\mathcal{I}(\vec{G})$ is closed under conditioning on C;

(b) for any connected component C^* of C, $\mathbf{an}_{\vec{G}}(C^*) \setminus C$ is C-complete w.r.t. α in $\vec{G}$.

Proof. To prove (a) implies (b), it follows from Proposition 31 that $\mathbf{an}_{\vec{G}}(C)$ is C-complete w.r.t. α in $\vec{G}$. Note that $\mathbf{an}_{\vec{G}}(C^*) \setminus C \subseteq \mathbf{an}_{\vec{G}}(C)$ for any connected component C^* of C, we know that $\mathbf{an}_{\vec{G}}(C^*) \setminus C$ is C-complete w.r.t. α in $\vec{G}$ by Proposition 52.

To prove (b) implies (a), by Proposition 31, it suffices to prove that $\mathbf{an}_{\vec{G}}(C)$ is C-complete w.r.t. α in $\vec{G}$. By contradiction, suppose that $\mathbf{an}_{\vec{G}}(C)$ is not C-complete w.r.t. α in $\vec{G}$. It holds that there exist $x, y \in \mathbf{an}_{\vec{G}}(C)$ with $x \overset{\vec{G}}{\nsim} y$ such that $\{x, y\} \notin \mathcal{PS}^{cp}_{\vec{G}}(C; \alpha)$. This leads to a contradiction with $\mathbf{an}_{\vec{G}}(C) = \bigcup_{C^* \subseteq C} \mathbf{an}_{\vec{G}}(C^*) \setminus C$, where C^* is a connected component of C in $\vec{G}$. ■

To prove Proposition 33, we still need three lemmas as follows. For convenience, let $\mathbf{fa}_{\vec{G}}(u) = \mathbf{pa}_{\vec{G}}(u) \cup \{u\}$. We define $\vec{G}_{r((u_1,v_1))} = \vec{G} - (u_1, v_1) + (v_1, u_1)$, $\vec{G}_{r((u_1,v_1),(u_2,v_2))} = \vec{G}_{r((u_1,v_1))} - (u_2, v_2) + (v_2, u_2), \ldots, \vec{G}_{r((u_1,v_1),\ldots,(u_m,v_m))} = \vec{G}_{r((u_1,v_1),\ldots,(u_{m-1},v_{m-1}))} - (u_m, v_m) + (v_m, u_m)$ for $m \geq 2$.

Lemma 54 *Suppose that $\vec{G} = (V, \vec{E})$ is a DAG and C is a subset of V. Then the following statements are equivalent:*

(a) $\mathcal{I}(\vec{G})$ is closed under conditioning on C;

(b) C is a covered set in $\vec{G}$;

(c) C is a covered set in $\vec{G}^C$;

(d) $\mathcal{V}(\vec{G}^C_T) = \emptyset$, where $T = \mathbf{An}_{\vec{G}^C}(C)$;

(e) for any $w \in \mathbf{An}_{\vec{G}^C}(C)$, w is a covered vertex in $\vec{G}^C$.

Proof. To prove (a) implies (b), it follows that $\mathcal{AV}_{\vec{G}}(C) = \emptyset$ from Proposition 13. This implies that $\mathbf{pa}_{\vec{G}}(w) \setminus C$ is complete for any $w \in \mathbf{An}_{\vec{G}}(C)$, that is, C is a covered set in $\vec{G}$.

To prove (b) implies (c), it holds that $\mathbf{pa}_{\vec{G}}(w) \setminus C$ is complete for any $w \in \mathbf{An}_{\vec{G}}(C)$. By the definition of $\vec{G}^C$, we know that $\mathbf{An}_{\vec{G}^C}(C) = \mathbf{An}_{\vec{G}}(C)$ and $\mathbf{pa}_{\vec{G}^C}(w) = \mathbf{pa}_{\vec{G}}(w) \setminus C$ for any $w \in \mathbf{An}_{\vec{G}^C}(C)$. From which we derive that C is a covered set in $\vec{G}^C$.

To prove (c) implies (d), suppose for contradiction that $\mathcal{V}(\vec{G}^C_T) \neq \emptyset$. Then there exists $w \in \mathbf{An}_{\vec{G}^C}(C)$ such that $\mathbf{pa}_{\vec{G}^C}(w)$ is not complete, a contradiction.

To prove (d) implies (e), by contradiction, suppose that there exists $w \in \mathbf{An}_{\vec{G}^C}(C)$ such that w is not a covered vertex in $\vec{G}^C$. That is, there exists $u \in \mathbf{An}_{\vec{G}^C}(w) \subseteq \mathbf{An}_{\vec{G}^C}(C)$ such that $\mathbf{pa}_{\vec{G}^C}(u)$ is not complete, a contradiction.

To prove (e) implies (a), suppose for contradiction that $\mathcal{I}(\vec{G}_R) \neq \mathcal{I}(\vec{G})^C$. It follows from Proposition 13 that there exists a C-configuration $\{x \to w \leftarrow y\}$ in $\vec{G}$. This implies that there exists $w \in \mathbf{An}_{\vec{G}}(C) = \mathbf{An}_{\vec{G}^C}(C)$ such that w is not a covered vertex in $\vec{G}^C$, which leads to a contradiction. ■

Lemma 55 *Suppose that $\vec{G} = (V, \vec{E})$ is a DAG with a topological order α and $u \in V$. Let $\mathbf{pa}_{\vec{G}}(u) = \{u_1, u_2, \dots, u_m\}$ such that $\alpha(u_1) > \alpha(u_2) > \cdots > \alpha(u_m)$ and $e_i = (u_i, u)$ for $i \in [m]$. Then $\{e_1, e_2, \dots, e_m\}$ is sequentially reversible in $\vec{G}$ if and only if $\mathbf{pa}_{\vec{G}}(u)$ is complete and $\mathbf{pa}_{\vec{G}}(u) = \mathbf{an}_{\vec{G}}(u)$.*

Proof. (if) For any $i \in [m]$, it is obvious that $w \in \mathbf{pa}_G(u)$ for any $w \in \mathbf{pa}_{\vec{G}}(u_i)$. Since $\alpha(u_1) > \alpha(u_2) > \cdots > \alpha(u_m)$, $\mathbf{pa}_{\vec{G}}(u) = \mathbf{fa}_{\vec{G}}(u_1)$ holds. Otherwise, suppose that there exists w such that $w \in \mathbf{pa}_{\vec{G}}(u)$ but $w \notin \mathbf{pa}_{\vec{G}}(u_1)$, a contradiction. Since $\vec{G}_{r(e_1)}$ is a DAG, we have that $\mathbf{pa}_{\vec{G}_{r(e_1)}}(u) = \mathbf{pa}_{\vec{G}}(u) \setminus \{u_1\} = \mathbf{an}_{\vec{G}}(u) \setminus \{u_1\} = \mathbf{an}_{\vec{G}_{r(e_1)}}(u)$ and $\mathbf{pa}_{\vec{G}_{r(e_1)}}(u)$ is complete. Similarly, note that $\alpha(u_2) > \cdots > \alpha(u_m)$ in $\vec{G}_{r(e_1)}$, it holds that $\mathbf{pa}_{\vec{G}_{r(e_1)}}(u) = \mathbf{fa}_{\vec{G}_{r(e_1)}}(u_2)$.

Applying the same logic to the rest, it holds $\mathbf{pa}_{\vec{G}_{r(e_0,e_1,e_2,\dots,e_{i-1})}}(u) = \mathbf{fa}_{\vec{G}_{r(e_0,e_1,e_2,\dots,e_{i-1})}}(u_i)$ for $i \in [m]$, where $\vec{G}_{r(e_0)} = \vec{G}$ and $\vec{G}_{r(e_0,e_1,e_2,\dots,e_i)} = \vec{G}_{r(e_0,e_1,e_2,\dots,e_{i-1})} - (u_i, u) + (u, u_i)$ for $i \in [m]$. By the definition of reversible edge, e_i is reversible in $\vec{G}_{r(e_0,e_1,e_2,\dots,e_{i-1})}$ for $i \in [m]$, implying that $\{e_1, e_2, \dots, e_m\}$ is sequentially reversible in $\vec{G}$.

(only if) It is obvious that $\mathbf{pa}_{\vec{G}_{r(e_0,e_1,e_2,\dots,e_{i-1})}}(u) = \mathbf{fa}_{\vec{G}_{r(e_0,e_1,e_2,\dots,e_{i-1})}}(u_i)$ for $i \in [m]$. By contradiction, suppose that $\mathbf{pa}_{\vec{G}}(u)$ is not complete or $\mathbf{pa}_{\vec{G}}(u) \neq \mathbf{an}_{\vec{G}}(u)$. If $\mathbf{pa}_{\vec{G}}(u)$ is not complete, there exist $k, s \in [m]$ such that $k > s$ and $(u_k, u_s) \notin \vec{E}$. Note that u_s, u_k are in $\mathbf{pa}_{\vec{G}_{r(e_0,e_1,\dots,e_{s-1})}}(u)$, we have $\mathbf{pa}_{\vec{G}_{r(e_0,e_1,\dots,e_{s-1})}}(u) \neq \mathbf{fa}_{\vec{G}_{r(e_0,e_1,\dots,e_{s-1})}}(u_s)$, a contradiction. If $\mathbf{pa}_{\vec{G}}(u) \neq \mathbf{an}_{\vec{G}}(u)$, there must exist some $j \in [m]$ such that $w \in \mathbf{pa}_{\vec{G}}(u_j)$ but $w \notin \mathbf{pa}_{\vec{G}}(u)$, also a contradiction. ■

Lemma 56 *Suppose that $\vec{G} = (V, \vec{E})$ is a DAG with a topological order α and $u \in V$. Let $\mathbf{pa}_{\vec{G}}(u) = \{u_1, u_2, \dots, u_m\}$ such that $\alpha(u_1) > \alpha(u_2) > \cdots > \alpha(u_m)$ and $e_i = (u_i, u)$ for $i \in [m]$. Then the following statements are equivalent:*

(a) u is a covered vertex in $\vec{G}$;

(b) there exists a DAG $\vec{G}' \in \mathcal{M}(\vec{G})$ such that $\mathbf{pa}_{\vec{G}'}(u) = \emptyset$;

(c) there exists a DAG $\vec{G}'' \in \mathcal{M}(\vec{G})$ such that $\{e_1, e_2, \dots, e_m\}$ is sequentially reversible in $\vec{G}''$;

(d) there exists a DAG $\vec{G}^* \in \mathcal{M}(\vec{G})$ such that $\mathbf{pa}_{\vec{G}^*}(u)$ is complete and $\mathbf{pa}_{\vec{G}^*}(u) = \mathbf{an}_{\vec{G}^*}(u)$.

Proof. To prove (a) implies (b), by contradiction, suppose that $\mathbf{pa}_{\vec{G}'}(u) \neq \emptyset$ for any $\vec{G}' \in \mathcal{M}(\vec{G})$. Then there exists an edge $(t, u) \in \vec{E}(\vec{G}')$ for any $\vec{G}' \in \mathcal{M}(\vec{G})$. From which we can derive that there exists a v-structure $\{t \to w \leftarrow s\} \in \mathcal{AV}_{\vec{G}}(u)$, a contradiction.

To prove (b) implies (c), suppose for contradiction that $\{e_1, e_2, \dots, e_m\}$ is not sequentially reversible in $\vec{G}''$ for any $\vec{G}'' \in \mathcal{M}(\vec{G})$. This implies that there exists at least one edge of $\{e_1, e_2, \dots, e_m\}$ being not reversible in $\vec{G}''$. Without loss of generality we assume that e_1 is not reversible in $\vec{G}''$. Consider that we can choose $\vec{G}''$ to be $\vec{G}'$, a contradiction.

To prove (c) implies (d), it holds obviously by Lemma 55. To prove (d) implies (a), suppose for contradiction that u is not a covered vertex in $\vec{G}$. There exists $w \in \mathbf{An}_{\vec{G}}(u)$

such that $\mathbf{pa}_{\vec{G}}(w)$ is not complete. Without loss of generality we assume that $x, y \in \mathbf{pa}_{\vec{G}}(w)$ and $x \overset{\vec{G}}{\nsim} y$. Then it holds that $\{x \to w \leftarrow y\} \in \mathcal{V}(\vec{G})$. Since $\vec{G}^* \in \mathcal{M}(\vec{G})$, we have that $w \in \mathbf{An}_{\vec{G}^*}(u)$ and $\{x \to w \leftarrow y\} \in \mathcal{V}(\vec{G}^*)$, a contradiction. ■

E.2 Proof of Proposition 33

Proof. By Proposition 5, we have that (a) and (b) are equivalent. According to Theorem 18, Propositions 25 and 28, we know that (b) and (c) are equivalent. It is trivial that (d) implies (c). To prove (c) implies (d), suppose for contradiction that there exists a topological order β_0 of $\vec{G}$ such that $\vec{E}_R^{\beta_0} \neq \emptyset$. This implies that $\mathcal{I}(\vec{G})^C \neq \mathcal{I}(\vec{G}_R)$. Since $\vec{E}_R^{\alpha} = \emptyset$, it holds that $\mathcal{I}(\vec{G})^C = \mathcal{I}(\vec{G}_R)$, a contradiction.

Lemma 54 yields that (a) and (e) are equivalent. Combining this with Lemma 56, we know that (e), (f), (g) and (h) are all equivalent. This completes the proof. ■

Appendix F. Additional Experiments

To compare the computational complexity of our approach and the local Markov method relying on d-separation checks, we conduct a simulation study here.

For each network in this experiment, two vertex variables are randomly selected as conditional variables and the topological order is also randomly selected. The average runtime of 100 repetitions is shown in Table 7 below. The column "ROS Time" is the average runtime of our method, and the column "LMP Time" is the average runtime of Algorithm 3 based on local Markov property (LMP). These results demonstrate the efficiency of our algorithm, with running times being negligible in these examples. Our experiment deomonstrates that our ROS algorithm is sufficiently efficient for our purpose.

In this example, the local Markov method takes longer time because it involves exhaustive checking of d-separation assertions for specific subsets within the Markov boundary of each vertex in the updated graph. Nevertheless, we believe that the research direction based on local Markov property holds promising potential for future studies to realize its full capacity.

Network	**Vertex Number**	**ROS Time**	**LMP Time**	**Ratio of Runtime**
Asia	8	0.004	0.007	1.750
Insurance	27	0.044	13.859	314.977
Alarm	37	0.047	15.516	330.128

Table 7: Average runtime for DAG minimal $\mathcal{I}$-maps by two methods (in seconds). The ROS Time is the average runtime of our proposed algorithm, and the LMP time refers to the average runtime of the algorithm based on d-separation checks by using local Markov property. Runtime Ratio is calculated as LMP Time divided by ROS Time. All values are rounded to three decimal places.

Algorithm 3 Finding Minimal $\mathcal{I}$-maps of Conditional Models by Local Markov Property

```
Input: A DAG $\vec{G} = (V, \vec{E})$ with a topological order $\alpha$, conditional set $C \subseteq V$.
Output: A minimal I-map of $\mathcal{P}(\vec{G})^C$ under $\alpha$: $\vec{G}_R^{\alpha,*}$.
1: Reorder the vertices in $R$ according to the reverse of the topological order $\alpha$, i.e.,
   $R = \{r_i, i \in [k] : \alpha(r_i) > \alpha(r_j), i < j, i, j \in [k]\}$, where $k = |R|$;
2: for j=1 to k do
3:     $D_j \leftarrow \mathbf{mb}_{\vec{G}}(r_j) \setminus (\mathbf{pa}_{\vec{G}}(r_j) \cup \mathbf{ch}_{\vec{G}}(r_j) \cup C)$;
4:     if $D_j = \emptyset$ then
5:         $R \leftarrow R \setminus r_j$ and continue;
6:     else
7:         for $F_j \subseteq D_j$ do
8:             $Z_j \leftarrow F_j \cup \mathbf{pa}_{\vec{G}}(r_j)$;
9:             if $r_j \perp\!\!\!\perp R \setminus (Z_j \cup \{r_j\}) | (Z_j, C)[\vec{G}]$ and $F_j \neq \emptyset$ then
10:                for $u \in F_j$ do
11:                    Add $u \rightarrow r_j$ to $\vec{G}$;
12:                end for
13:            end if
14:        end for
15:    end if
16:    $R \leftarrow R \setminus r_j$;
17: end for
18: $\vec{G}_R^{\alpha,*} \leftarrow \vec{G}_R$;
19: return $\vec{G}_R^{\alpha,*}$.
```

References

G. Borboudakis and I. Tsamardinos. Bayesian network learning with discrete case-control data. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, pages 151–160, 2015.

C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in Bayesian networks. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 115–123, 1996.

D. M. Chickering and C. Meek. Finding optimal Bayesian networks. In *Conference on Uncertainty in Artificial Intelligence*, pages 94–102, 2002.

D. M. Chickering, D. Heckerman, and C. Meek. A Bayesian approach to learning Bayesian networks with local structure. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 80–89, 1997.

G. F. Cooper. A Bayesian method for causal modeling and discovery under selection. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 98–106, 2000.

R. Cowell, A. Dawid, S. Lauritzen, and D. Spiegelhalter. *Probabilistic Networks and Expert Systems: Exact Computational Methods for Bayesian Networks.* Springer Publishing Company, Incorporated, 2007.

D. R. Cox and N. Wermuth. *Multivariate Dependencies: Models, Analysis and Interpretation*, volume 67. CRC Press, 1996.

R. M. Daniel, M. G. Kenward, S. N. Cousens, and B. L. D. Stavola. Using causal diagrams to guide analysis in missing data problems. *Statistical Methods in Medical Research*, 21 (3):243–256, 2012.

M. Drton and M. H. Maathuis. Structure learning in graphical modeling. *Annual Review of Statistics and Its Application*, 4(1):365–393, 2017.

M. J. Druzdzel and F. J. Díez. Combining knowledge from different sources in causal probabilistic models. *Journal of Machine Learning Research*, 4(3):295–316, 2003.

R. J. Evans. Graphs for margins of Bayesian networks. *Scandinavian Journal of Statistics*, 43(3):625–648, 2016.

R. J. Evans and V. Didelez. Recovering from selection bias using marginal structure in discrete models. In *Proceedings of the UAI 2015 Conference on Advances in Causal Inference-Volume 1504*, pages 46–55, 2015.

M. J. Flores, A. E. Nicholson, A. Brunskill, K. B. Korb, and S. Mascaro. Incorporating expert knowledge when learning Bayesian network structure: A medical case study. *Artificial Intelligence in Medicine*, 53(3):181–204, 2011.

A. Gain and I. Shpitser. Structure learning under missing data. In *Proceedings of the Ninth International Conference on Probabilistic Graphical Models*, pages 121–132, 2018.

C. Glymour and C. Meek. Conditioning and intervening. *British Journal for the Philosophy of Science*, 45(4):1001–1021, 1994.

C. Glymour, K. Zhang, and P. L. Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, 10(524), 2019.

F. R. Guo, E. Perković, and A. Rotnitzky. Variable elimination, graph reduction and the efficient g-formula. *Biometrika*, 2022. asac062.

C. Heinze-Deml, M. H. Maathuis, and N. Meinshausen. Causal structure learning. *Annual Review of Statistics and Its Application*, 5(1):371–391, 2018.

C. Hitchcock. Conditioning, intervening, and decision. *Synthese*, 193(4):1157–1176, 2016.

S. Højsgaard. Graphical independence networks with the grain package for R. *Journal of Statistical Software*, 46(10):1–26, 2012.

H. Kiiveri, T. P. Speed, and J. B. Carlin. Recursive causal models. *Journal of the Australian Mathematical Society*, 36(1):30–52, 1984.

N. Kitson, A. Constantinou, Z. Guo, Y. Liu, and K. Chobtham. A survey of Bayesian network structure learning. *Artificial Intelligence Review*, 56(8):8721–8814, 2023.

Y. Kivva, J. Etesami, and N. Kiyavash. On identifiability of conditional causal effects. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 1078–1086, 2023.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques.* MIT Press, Cambridge, MA, 2009.

J. T. A. Koster. Marginalizing and conditioning in graphical models. *Bernoulli*, 8(6): 817–840, 2002.

H. Langseth and O. Bangsø. Parameter learning in object-oriented Bayesian networks. *Annals of Mathematics and Artificial Intelligence*, 32(1):221–243, 2001.

M. Lappenschaar, A. J. Hommersom, P. J. Lucas, J. Lagro, S. Visscher, J. C. Korevaar, and F. G. Schellevis. Multilevel temporal Bayesian networks can model longitudinal change in multimorbidity. *Journal of Clinical Epidemiology*, 66(12):1405–1416, 2013.

S. L. Lauritzen. *Graphical Models*, volume 17 of *Oxford Statistical Science Series.* The Clarendon Press, Oxford University Press, New York, 1996.

S. L. Lauritzen. Generating mixed hierarchical interaction models by selection. Technical Report R-98-2009, Dept. Mathematics, Univ. Aalborg, 1998.

S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society: Series B (Statistical Methodological)*, 50(2):157–194, 1988.

U. Lerner, E. Segal, and D. Koller. Exact inference in networks with discrete children of continuous parents. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 319–328, 2001.

B. Liu and J. Guo. Collapsibility of conditional graphical models. *Scandinavian Journal of Statistics*, 40(2):191–203, 2012.

Y. Liu and A. C. Constantinou. Greedy structure learning from data that contain systematic missing values. *Machine Learning*, 111(10):3867–3896, 2022.

M. Maathuis, M. Drton, S. Lauritzen, and M. Wainwright. *Handbook of Graphical Models.* CRC Press, 2018.

D. Marella and P. Vicard. Object-oriented Bayesian networks for modeling the respondent measurement error. *Communications in Statistics—Theory and Methods*, 42(19):3463–3477, 2013.

C. Meek. Strong completeness and faithfulness in Bayesian networks. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 411–418, 1995.

K. Mohan, J. Pearl, and J. Tian. Graphical models for inference with missing data. In *Advances in Neural Information Processing Systems*, pages 1277–1285, 2013.

S. Monti and G. F. Cooper. Learning Bayesian belief networks with neural network estimators. In *Proceedings of the Ninth International Conference on Neural Information Processing Systems*, pages 578–584, 1996.

J. Mortera, P. Vicard, and C. Vergari. Object-oriented Bayesian networks for a decision support system for antitrust enforcement. *The Annals of Applied Statistics*, 7(2):714–738, 2013.

R. E. Neapolitan. *Learning Bayesian Networks*, volume 38. Pearson Prentice Hall Upper Saddle River, 2004.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

J. Pearl. *Causality: Models, Reasoning and Inference, 2nd edn.* Cambridge University Press, Cambridge, UK, 2009.

J. Pearl, D. Geiger, and T. Verma. Conditional independence and its representations. *Kybernetika*, 25(7):33–44, 1989.

Y. Peng and J. A. Reggia. Plausibility of diagnostic hypotheses: the nature of simplicity. In *Proceedings of the Fifth AAAI National Conference on Artificial Intelligence*, pages 140–145, 1986.

T. Richardson and P. Spirtes. Ancestral graph Markov models. *The Annals of Stastics*, 30 (4):962–1030, 2002.

J. M. Robins. A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical Modelling*, 7(9-12):1393–1512, 1986.

K. Sadeghi. Marginalization and conditioning for LWF chain graphs. *The Annals of Stastics*, 44(4):1792–1816, 2016.

K. Sadeghi. Faithfulness of probability distributions and graphs. *Journal of Machine Learning Research*, 18(1):5429–5457, 2017.

M. Scutari. Bayesian network repository, 2024. URL `https://www.bnlearn.com/bnrepository/`. Accessed: 2024-11-20.

M. Scutari and J. B. Denis. *Bayesian Networks with Examples in R.* Chapman and Hall/CRC, 2014.

I. Shpitser and J. Pearl. Identification of conditional interventional distributions. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, pages 437–444, 2006.

P. Spirtes, C. Meek, and T. Richardson. Causal inference in the presence of latent variables and selection bias. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 499–506, 1995.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search, 2nd Edition.* MIT Press, Cambridge, MA, 2001.

C. Squires and C. Uhler. Causal structure learning: a combinatorial perspective. *Foundations of Computational Mathematics*, 23(5):1781–1815, 2022.

S. Srinivas. A probabilistic approach to hierarchical model-based diagnosis. In *Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence*, pages 538–545, 1994.

E. V. Strobl, S. Visweswaran, and P. L. Spirtes. Fast causal inference with non-random missingness by test-wise deletion. *International Journal of Data Science and Analytics*, 6(1):47–62, 2018.

S. Sullivant, K. Talaska, and J. Draisma. Trek separation for Gaussian graphical models. *The Annals of Statistics*, 38(3):1665–1685, 2010.

J. Tian. Identifying conditional causal effects. In *Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence*, pages 561–568, 2004.

R. Tu, C. Zhang, P. Ackermann, K. Mohan, H. Kjellström, and K. Zhang. Causal discovery in the presence of missing data. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pages 1762–1770, 2019.

J. Vila-Francés, J. Sanchis, E. Soria-Olivas, A. J. Serrano, M. Martinez-Sober, C. Bonanad, and S. Ventura. Expert system for predicting unstable angina based on Bayesian networks. *Expert Systems with Applications*, 40(12):5004–5010, 2013.

M. J. Vowels, N. C. Camgoz, and R. Bowden. D'ya like DAGs? A survey on structure learning and causal discovery. *ACM Computing Surveys*, 55(4):1–36, 2022.

N. Wermuth. Probability distributions with summary graph structure. *Bernoulli*, 17(3): 845–879, 2011.

N. Wermuth and S. L. Lauritzen. Graphical and recursive models for contingency tables. *Biometrika*, 70(3):537–552, 1983.

X. Xie, Z. Geng, and Q. Zhao. Decomposition of structural learning about directed acyclic graphs. *Artificial Intelligence*, 170(4-5):422–439, 2006.

B. Yet, Z. B. Perkins, T. E. Rasmussen, N. Tai, and D. W. R. Marsh. Combining data and meta-analysis to build Bayesian networks for clinical decision support. *Journal of Biomedical Informatics*, 52:373–385, 2014.

Y. Yu, J. Chen, T. Gao, and M. Yu. DAG-GNN: DAG structure learning with graph neural networks. In *International Conference on Machine Learning*, pages 7154–7163, 2019.

A. Zanga, E. Ozkirimli, and F. Stella. A survey on causal discovery: Theory and practice. *International Journal of Approximate Reasoning*, 151:101–129, 2022.

J. Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16):1873–1896, 2008.