Constraint-based Causal Discovery from a Collection of Conditioning Sets

Kenneth Lee[1] **Bruno Ribeiro**[2] **Murat Kocaoglu**[1]

[1]Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA
[2]Department of Computer Science, Purdue University, West Lafayette, IN, USA

Abstract

In constraint-based causal discovery, the existing algorithms systematically use a series of conditional independence (CI) relations observed in the data to recover an equivalence class of causal graphs in the large sample limit. One limitation of these algorithms is that CI tests lose statistical power as conditioning set size increases with finite samples. Recent research proposes to limit the conditioning set size for robust causal discovery. However, the existing algorithms require exhaustive testing of all CI relations with conditioning set sizes up to a certain integer k. This becomes problematic in practice when variables with large support are present, as it makes CI tests less reliable due to near-deterministic relationships, thereby violating the faithfulness assumption. To address this issue, we propose a causal discovery algorithm that only uses CI tests where the conditioning sets are restricted to a given set of conditioning sets including the empty set $\mathcal{C}$. We call such set of CI relations $\mathcal{I}_\mathcal{C}$ *conditionally closed*. We define the notion of $\mathcal{C}$-Markov equivalence: two causal graphs are $\mathcal{C}$-Markov equivalent if they entail the same set of CI constraints from $\mathcal{I}_\mathcal{C}$. We propose a graphical representation of $\mathcal{C}$-Markov equivalence and characterize such equivalence between two causal graphs. Our proposed algorithm called the $\mathcal{C}$-PC algorithm is sound for learning the $\mathcal{C}$-Markov equivalence class. We demonstrate the utility of the proposed algorithm via synthetic and real-world experiments in scenarios where variables with large support or high correlation are present in the data. Our source code is available online at github.com/kenneth-lee-ch/cpc.

1 INTRODUCTION

Causal inference has played a vital role in many scientific fields such as biology (Chang et al., 2014; Pearl and Mackenzie, 2018), economics (Varian, 2016; Hair Jr and Sarstedt, 2021), epidemiology (VanderWeele and Robinson, 2014; Reinhart et al., 2021) and medical sciences (Rizzi and Pedersen, 1992; Feuerriegel et al., 2024). A flurry of work has been proposed in machine learning and artificial intelligence literature to address issues such as fairness (Kusner et al., 2017; Chiappa, 2019; Wu et al., 2019), generalization (Peters et al., 2016; Subbaswamy et al., 2019; Ilse et al., 2021; Von Kügelgen et al., 2021), interpretability (Jung et al., 2022; Rohekar et al., 2024), and data privacy (Binkyte et al., 2024) by formalizing the data-generating process as a structural causal model (SCM) (Pearl, 2009a). Directed acyclic graphs (DAGs), also known as causal graphs, are used to model causal relations among a set of random variables. Particularly, an SCM induces a causal graph by assigning a set of endogenous variables as the parents for each observed variable. The significance of causal inference lies in its ability to elucidate the effects of interventions or treatments (Pearl, 1995; Shpitser and Pearl, 2008; **?**; Bareinboim and Pearl, 2016). Estimating these effects from observational data hinges on the causal relationships among variables within a system, necessitating a causal graph for effectively addressing most causal inference tasks.

There are many ways to discover a causal graph about a system. The golden standard is to conduct randomized controlled trials (Keum et al., 2019; Cloyd et al., 2020; Prete et al., 2018). However, randomized experiments may not be feasible or may even be unethical. For instance, there may not be enough interventional samples (Harris et al., 2006). Some experiments may pose risks of concern; for instance, death, irreversible damage, and temporary disability (Miller and Brody, 2002). For a small and well-established domain, expert knowledge may still be feasible. However, solely relying on domain knowledge can be prohibitively expensive in large and complex systems (Yang et al., 2023).

Also, observational data is usually more accessible than experimental data in many domains. Hence, learning a causal graph about the variables in a system from observational data is important.

Related work. Causal discovery aims to recover causal graphs from observational data. Constraint-based causal discovery algorithms utilize a series of conditional independence (CI) tests with graphical orientation rules to learn a representation known as the Markov equivalence class, which represents a set of causal graphs that have the same CI constraints in the large sample limit (Spirtes et al., 2001; Zhang and Spirtes, 2003; Cai et al., 2022). Alternatively, other methods are available, such as exploiting asymmetry in the distributions (Shimizu et al., 2006, 2011), learning exogenous variables with minimum entropy (Kocaoglu et al., 2017; Compton et al., 2020), and score-based methods that greedily search the space of possible DAGs. The score-based methods output a DAG within a Markov equivalence class by optimizing a regularized score function (Chickering, 2002, 2020; Claassen and Bucur, 2022). Additionally, hybrid algorithms blend constraint-based and score-based approaches (Tsamardinos et al., 2006; Gasse et al., 2014; Raskutti and Uhler, 2018; Lam et al., 2022; Guo et al., 2022), while gradient-based methods formulate causal discovery as a continuous optimization problem (Zheng et al., 2018; Yu et al., 2019; Ng et al., 2020, 2022). Order-based methods cast causal discovery as learning the optimal ordering of variables to reduce search space (Schmidt et al., 2007; Xiang and Kim, 2013; Bühlmann et al., 2014; Wang et al., 2021). Each type of method has its limitations. For instance, a significant issue with score-based approaches is that the search space can grow exponentially with the number of variables, especially if the models computed are complex. Meanwhile, gradient-based methods struggle with solving nonconvex optimization problems. Constraint-based methods struggle with limited data as CI tests are prone to have a high false positive rate, especially when the conditioning set is large (Wille and Bühlmann, 2006; Shah and Peters, 2020). Due to the sensitivity of CI tests to sample noise, several ideas have been proposed to enhance the accuracy of the algorithm outputs, including ensuring the output's independence from the sequence of CI tests conducted (Colombo et al., 2014), relaxing model assumptions (Ramsey et al., 2006), ensuring the consistency of conditioning sets used in CI tests (Li et al., 2019), and characterizing and learning of causal graphs based on small conditioning sets (Wienöbst and Liskiewicz, 2020; Kocaoglu, 2023). Building on previous research that explores learning causal graphs with small conditioning sets, our paper aims to further relax the requirement of exhausting all CI relations up to degree k. We achieve this by employing a set of *conditionally closed* CI tests, which is based on a specified collection of conditioning sets $\mathcal{C}$, including the empty set.

To underscore the importance of learning causal graphs from a flexibly chosen set of CI relations, we note the prevalence of scenarios where variables with large support and high correlation are present in the data. For example, when discrete variables in the observed data have large support (e.g., numerous states), performing CI tests on the entire dataset essentially *slices* the dataset according to the states of the variables in the conditioning set. This fragmentation can rapidly diminish statistical power, making some CI tests unreliable. Another issue in using CI tests for learning causal graphs is the problem of high correlation. For instance, when multicollinearity is present in the data, conducting a Fisher's Z test on X and Y given Z where X and Z are highly correlated can lead to incorrect conclusions about the dependence between X and Y. Learning causal graphs with a specific set of conditioning sets allows the flexibility to bypass unreliable CI tests, which can be anticipated by examining the correlations or the number of states of the variables in the data beforehand. Motivated by these examples, this paper aims to characterize and learn causal graphs using a collection of conditioning sets.

Our results. In this paper, we propose learning causal graphs using CI tests based on a collection of conditioning sets $\mathcal{C}$, allowing us to flexibly ignore some CI tests. This flexibility helps address unreliable CI tests in finite samples or with variables having large support, particularly when near-deterministic relationships in the data may violate the faithfulness assumption. We call CI constraints where the conditioning sets are restricted to $\mathcal{C}$ the conditionally closed CI constraints. Two causal graphs are said to be $\mathcal{C}$-Markov equivalent if they entail the same conditionally closed CI constraints. We propose the *$\mathcal{C}$-closure graphs* that entail the same conditionally closed CI constraints and preserve the ancestral causal relations as the causal graph. We propose a sound constraint-based causal discovery algorithm for learning the $\mathcal{C}$-Markov equivalence class from observational data. We present a graphical representation called $\mathcal{C}$-essential graph to represent this equivalence class. We demonstrate the utility of the proposed algorithm via both synthetic and real-world experiments in scenarios where variables with large support and high correlation are present.

2 BACKGROUND

In this section, we give the most relevant definitions and relevant assumptions from graphical causal models literature (Pearl, 2009b). We leave the basic graph terminology in Appendix B. We use boldface capital letters to denote a set of variables. We assume that there is no latent confounder or selection bias relative to the set of observed variables.

Definition 2.1 (d-separation)**.** In a DAG D, a path p between vertices X and Y is *d-connecting (active)* relative to a set of vertices $\mathbf{Z}(X, Y \notin \mathbf{Z})$ if (i) every non-collider on p is not in $\mathbf{Z}$ and (ii) every collider on p is an ancestor of some $Z \in \mathbf{Z}$. If there is no d-connecting path between X and Y

relative to $\mathbf{Z}$, we say X and Y are *d-separated* relative to $\mathbf{Z}$, denoted as $(X \perp\!\!\!\perp Y|\mathbf{Z})_D$ and all paths between X and Y are *blocked* by $\mathbf{Z}$.

Let P be a joint probability distribution over a set of random variables $\mathbf{V}$. Let $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ be subsets of $\mathbf{V}$. We use the notation $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}\,|\mathbf{Z})_P$ to denote that $\mathbf{X}$ is conditionally independent of $\mathbf{Y}$ given $\mathbf{Z}$ in P.

Assumption 2.2 (Causal Markov condition)**.** Given a set of variables whose causal structure can be represented by a DAG D, every variable is probablistically independent of its non-descendants conditional on its parents in D.

Assumptions 2.2 give us the following relation between d-separation and conditional independence (Pearl, 2009b; Spirtes et al., 2001; Heckerman et al., 1999).

Proposition 2.3 (Pearl (2009b))**.** *Let P be a joint distribution over a set of variables $\mathbf{V}$ on a causal graph D. For any subsets $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathbf{V}$, $\mathbf{X}$ and $\mathbf{Y}$ are d-separated relative to $\mathbf{Z}$ in D implies $\mathbf{X}$ is conditionally independent of $\mathbf{Y}$ given $\mathbf{Z}$ in P, i.e.*

$$(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}\,|\mathbf{Z})_D \Rightarrow (\mathbf{X} \perp\!\!\!\perp \mathbf{Y}\,|\mathbf{Z})_P \tag{1}$$

Assumption 2.4 (Causal faithfulness assumption)**.** A distribution P is called faithful to a causal graph $D = (\mathbf{V}, \mathbf{E})$ if and only if for disjoint subsets $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subset \mathbf{V}$

$$(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}\,|\mathbf{Z})_P \Rightarrow (\mathbf{X} \perp\!\!\!\perp \mathbf{Y}\,|\mathbf{Z})_D \tag{2}$$

Assumption 2.4 is a typical assumption in constraint-based causal discovery. It assumes only the CI relations induced by d-separation are observed from data. Some existing algorithms have relaxed this assumption (Raskutti and Uhler, 2018; Wienöbst and Liskiewicz, 2020; Kocaoglu, 2023). Later, we will also weaken this assumption as we are only given a collection of conditioning sets for learning causal graphs. In general, constraint-based algorithms can only learn up to an equivalence class of models, a set of DAGs that induce the same conditional independencies via d-separation, which gives the following definition.

Definition 2.5 (Markov equivalence)**.** Two DAGs D_1, D_2 with the same set of vertices are *Markov equivalent* if for any three disjoint set of vertices $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$, $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in D_1 if and only if $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in D_2. The set of DAGs that encode the same set of conditional independence induced only by the causal Markov assumption is called the *Markov equivalence class*.

Kocaoglu (2023) extended the notion of Markov equivalence to k-Markov equivalence by restricting the cardinality of the conditioning set $\mathbf{Z}$ up to some integer k, i.e., $|\mathbf{Z}| \leq k$. The set of DAGs that belong to the same k-Markov equivalence class entails the same set of CI statements up to degree k. Kocaoglu (2023) proposed an algorithm called k-PC to learn the k-Markov equivalence class of DAGs. They showed that this problem is equivalent to learning the Markov equivalence class of a related graphical representation known as k-closure graphs. At a high level, a k-closure graph is a mixed graph constructed from a DAG to capture CI statements of order up to k. The Markov equivalence class of k-closure graphs is represented by a k-essential graph. Since our work builds on k-PC, we briefly review its core ideas. k-PC begins with a complete undirected graph with circle marks on both ends of each edge. Each circle mark represents that there exists a k-closure graph with a tail mark and another k-closure graph with an arrowhead. k-PC then proceeds to test every conditional independence relation up to order k. Note that k-closure graph is a maximal ancestral graph (MAG) according to Lemma 3.8 in Kocaoglu (2023). Then, it orients the unshielded collider for all unshielded triples whenever the middle node Z in the triple is not found in any conditioning set $\mathbf{Z}$ of the CI statements that involve the other two variables X, Y e.g., $Z \notin \mathbf{Z}$ for any $(X \perp\!\!\!\perp Y|\mathbf{Z})_P$. Then, k-PC will apply some of the orientation rules that are used in the well-celebrated FCI algorithm Zhang (2008b). Finally, k-PC applies some new orientation rules that are specifically designed for learning a k-essential graph, as shown by Algorithm 4 in Appendix A.

Q1	Q2	Q3	Q4	Q5
0	10	1	5	1
1	10	1	5	0
1	10	0	5	0
1	10	0	5	1
0	10	1	5	0
0	10	1	5	0
...	...	...	...	...

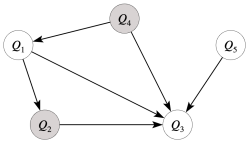


Figure 1: An example where learning causal graphs from a collection of conditioning sets is desired. A survey dataset consists of some categorical variables, e.g. $Q2, Q4$, that show most participants respond in the same way, creating highly correlated variables that pose challenges in testing conditional independence.

3 LEARNING CAUSAL GRAPHS FROM A CONDITIONALLY CLOSED SET OF CI RELATIONS

We begin by presenting an example to showcase the utility of learning causal graphs from a conditionally closed set of CI relations. Consider a scenario in which survey results have been collected to analyze the causal relationships among observed variables. As shown by Figure 1, the survey primarily consists of categorical variables. Some questions, such as $Q2, Q4$, involve many choices for participants to choose from and the majority of the participants tend to

select the same answer. This results in a dataset that consists of variables with large support and high correlation, posing challenges for testing conditional independence. Similar challenges arise in other domains, such as bioinformatics (Ahmed et al., 2018) and finance (Ledoit and Wolf, 2003). Hence, it is desirable to choose which variable to condition on for constraint-based causal discovery.

To take the first step in tackling this challenge, we define a set of CI relations being conditionally closed by restricting the conditioning sets to a collection of conditioning sets $\mathcal{C}$.

Definition 3.1 (Conditionally Closed Sets of CIs)**.** For a DAG $D = (\mathbf{V}, \mathbf{E})$, let $\mathcal{I} = \{I_i\}_{i\in[k]}$ be a set of CI statements of the form $I_i = (X, \mathbf{C}, Y)_i, i.e., (X \perp\!\!\!\perp Y|\mathbf{C})$ or $(X \not\perp\!\!\!\perp Y|\mathbf{C})$, where $X, Y \in \mathbf{V}, \mathbf{C} \subset \mathbf{V}, k \in \mathbb{N}$. $\mathcal{I}$ is *conditionally closed* relative to $\mathcal{C}$, denoted as $\mathcal{I}_\mathcal{C}$, if there exists a set $\mathcal{C} \subset \mathcal{P}(\mathbf{V})$, where $\mathcal{P}(\mathbf{V})$ is the power set of $\mathbf{V}$, that satisfies the following

1. $\emptyset \in \mathcal{C}$ and
2. If there exists $X, Y \in \mathbf{V}$ such that $(X, \mathbf{C}, Y) \in \mathcal{I}$, then $(A, \mathbf{C}, B) \in \mathcal{I}$ for all $A, B \in \mathbf{V}$ and for all $\mathbf{C} \in \mathcal{C}$.

Intuition of Definition 3.1. For any conditioning set $\mathbf{C}$ in $\mathcal{C}$, all CI relations with the conditioning set $\mathbf{C}$ must be fully observed from data. $\mathcal{C}$ must at least include the empty set.

Throughout this work, we use $\mathcal{C}$ to denote some sets that satisfy the conditions $1, 2$ in Definition 3.1. We also define a set of DAGs that are Markov equivalent based on the set of CI relations restricted to $\mathcal{C}$.

Definition 3.2 ($\mathcal{C}$-Markov equivalence)**.** Two DAGs D_1, D_2 are *$\mathcal{C}$-Markov equivalent* if for any three disjoint subsets $\mathbf{X} \subset \mathbf{V}$, $\mathbf{Y} \subset \mathbf{V}, \mathbf{C} \in \mathcal{C}$, $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{C}$ in D_1 if and only if $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{C}$ in D_2. We denote two DAGs D_1, D_2 that are $\mathcal{C}$-Markov equivalent as $D_1 \sim_\mathcal{C} D_2$.The set of DAGs that encode the same $\mathcal{I}_\mathcal{C}$ is called the *$\mathcal{C}$-Markov equivalence class*.

3.1 DIFFERENCES BETWEEN A MARKOV EQUIVALENCE CLASS AND A $\mathcal{C}$-MARKOV EQUIVALENCE CLASS

Verma (1991) has shown that two DAGs are Markov equivalent if and only if they share the same skeleton and unshielded colliders (see Theorem B.9 in Appendix B). We now give an example to show that two DAGs that share the same d-separation statements restricted to $\mathcal{C}$ do not have the same skeleton and the same set of unshielded colliders. We use Figure 2 to illustrate the difference between the Markov equivalence class and the $\mathcal{C}$-Markov equivalence class. As shown by Figures 2(a) and 2(b), both D and D' have different skeletons and different sets of unshielded colliders. Nonetheless, D and D' are $\mathcal{C}$-Markov equivalent when $\mathcal{C} = \{\emptyset, \{Y\}\}$ because both D and D' entail the same CI statement of degree 0 and the same CI statements with conditioning set $\{Y\}$.

Our main contribution is to extend an existing algorithm called k-PC (Kocaoglu, 2023) from learning causal graphs based on CI relations with conditioning set sizes up to k[1] to learning causal graphs based on $\mathcal{I}_\mathcal{C}$ (see Appendix A for the pseudocode of k-PC). All proofs can be found in Appendix C. Following a similar framework to (Kocaoglu, 2023), we have the following auxiliary representation to facilitate the characterization of $\mathcal{C}$-Markov equivalence class. We call this $\mathcal{C}$-closure graph, which is similar to ancestral graphs but without latent variables (Richardson and Spirtes, 2002).

Definition 3.3 ($\mathcal{C}$-covered)**.** Given a DAG $D = (\mathbf{V}, \mathbf{E})$ and a set $\mathcal{C}$, a pair of variables X, Y is said to be $\mathcal{C}$-covered if there exists no separating set $\mathbf{C}$ in $\mathcal{C}$ to d-separate X and Y in D, i.e., $\nexists \mathbf{C} \in \mathcal{C}$ s.t. $(X \perp\!\!\!\perp Y|\mathbf{C})_D$.

Definition 3.4 ($\mathcal{C}$-closure)**.** For a DAG D and a set $\mathcal{C}$, the $\mathcal{C}$-closure of D, denoted as $\mathcal{S}_\mathcal{C}(D)$, is a mixed graph that has the following properties:

1. If: X, Y are $\mathcal{C}$-covered in D
   (i) if $X \in An_D(Y)$, then $X \to Y$ in $\mathcal{S}_\mathcal{C}(D)$, (ii) if $Y \in An_D(X)$, then $Y \to X$ in $\mathcal{S}_\mathcal{C}(D)$, (iii) else $X \leftrightarrow Y$ in $\mathcal{S}_\mathcal{C}(D)$.
2. Else: X, Y are not adjacent in $\mathcal{S}_\mathcal{C}(D)$.

Intuition of Definition 3.4. When two variables X, Y are $\mathcal{C}$-covered and none of them is an ancestor of another, it is as if there exists a confounder between them that is not observed, resulting in $X \leftrightarrow Y$. When one is an ancestor of another, $\mathcal{C}$-closure graphs preserve this ancestral relationship. Figures 2(c) and 2(d) show an example of $\mathcal{C}$-closure graphs. As $\mathcal{C}$-closure graphs are ADMGs, they have the same probabilistic interpretation as ancestral graphs via m-separation. The following lemma gives the relationship between the CI statements entailed by a DAG and the CI statements entailed by a $\mathcal{C}$-closure graph.

Lemma 3.5. *$\mathcal{C}$-closure graph $\mathcal{S}_\mathcal{C}(D)$ of a DAG D entails the same d-separation statements conditioned on any $\mathbf{C} \in \mathcal{C}$ as the DAG, i.e.,* $(X \perp\!\!\!\perp Y|\mathbf{C})_D \Leftrightarrow (X \perp\!\!\!\perp Y|\mathbf{C})_{\mathcal{S}_\mathcal{C}(D)}, \forall \mathbf{C} \in \mathcal{C}$.

As shown by Figures 2(a) and 2(b), given a conditional closed set $\mathcal{C} = \{\emptyset, \{Y\}\}$, we see that W and Q are d-separated by Y in both D and D'. This d-separation statement also holds in their respective $\mathcal{C}$-closure graphs $\mathcal{S}_\mathcal{C}(D)$ and $\mathcal{S}_\mathcal{C}(D')$ as shown by Figures 2(c) and 2(d). All pairwise marginal dependencies also hold in both D and D' and

[1]The first algorithm that learns causal DAGs from CIs up to order k with a full analysis of its correctness is called LOCI (Wienöbst and Liskiewicz, 2020).

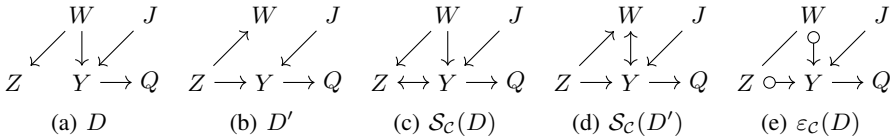


Figure 2: An example shows the difference between a Markov equivalence class and a $\mathcal{C}$-Markov equivalence class, where $\mathcal{C} = \{\emptyset, \{Y\}\}$. (a)-(b): two $\mathcal{C}$-Markov equivalent DAGs, i.e., $D \sim_{\mathcal{C}} D'$. (c)-(d): two Markov equivalent $\mathcal{C}$-closure graphs $\mathcal{S}_{\mathcal{C}}(D), \mathcal{S}_{\mathcal{C}}(D')$. (e): the $\mathcal{C}$-essential graph of D.

Algorithm 1 $\mathcal{C}$-PC

input Observational data $\mathbf{V}$, a conditionally closed set $\mathcal{C}$, CI tester

Initiate a complete graph M among the set of observed variables with circle edge o—o.
Find separating sets $S_{X,Y}$ for every pair $X, Y \in V$ by conditioning on $C \in \mathcal{C}$.
Update M by removing the edges between pairs that are separable.
Orient unshielded colliders of M: For any induced subgraph Xo—oZo—oY or $Xo \rightarrow Zo$—oY or Xo—$oZ \leftarrow oY$, set $Xo \rightarrow Z \leftarrow oY$ for any non-adjacent pair X, Y where $S_{X,Y}$ does not contain Z.
$M \leftarrow$ **FCI_Orient**(M) {Algorithm 5 in Appendix A}
$M \leftarrow$ **kPC_Orient**(M) {Algorithm 4 in Appendix A}
return M

Algorithm 2 Heuristic Search on $\mathcal{C}$

input Observational data $\mathbf{V}$, number of samples per entry in the contingency table k (default is 5)

Discretize the dataset. Let $\mathcal{C} = \{\{\emptyset\}\}$
for $\mathbf{Z} \subseteq \mathbf{V}, |\mathbf{Z}| > 0$ **do**
  for $X, Y \in \mathbf{V} \setminus \mathbf{Z}$ **do**
    Construct contingency tables for X, Y given $\mathbf{Z}$
    if All entries have more than k samples in each table **then**
      Mark $\mathbf{Z}$ as "reliable".
  if $\mathbf{Z}$ is "reliable" for more than half of the pairs X, Y **then**
    Add $\mathbf{Z}$ to $\mathcal{C}$
return $\mathcal{C}$

their respective $\mathcal{C}$-closure graphs. Similar to the result given by Theorem B.9, we can also check whether two DAGs are $\mathcal{C}$-Markov equivalent by verifying whether their respective $\mathcal{C}$-closure graphs share the same set of skeletons and unshielded colliders.

Theorem 3.6. *Two DAGs D_1, D_2 are $\mathcal{C}$-Markov equivalent if and only if $\mathcal{S}_{\mathcal{C}}(D_1)$ and $\mathcal{S}_{\mathcal{C}}(D_2)$ are Markov equivalent.*

From Figures 2(a) and 2(b), given $\mathcal{C} = \{\emptyset, \{Y\}\}$, we see that D_1 and D_2 are $\mathcal{C}$-Markov equivalent and $\mathcal{S}_{\mathcal{C}}(D_1)$ and $\mathcal{S}_{\mathcal{C}}(D_2)$ are Markov equivalent due to Theorem 3.6. Similar to the idea of using all CI statements to learn a Markov equivalence class of DAGs (Spirtes et al., 2001), our goal is to conduct CI tests only by using $\mathcal{I}_{\mathcal{C}}$ to learn the $\mathcal{C}$-Markov equivalence class of DAGs, that is, the Markov equivalence class of $\mathcal{C}$-closure graphs. We do so by using the following edge union operation to characterize the Markov equivalence class of $\mathcal{C}$-closure graphs.

Definition 3.7 (edge unions: —, o—o, $o \rightarrow$ (Kocaoglu, 2023))**.** The edge union operations of a set of $\mathcal{C}$-closure graphs are defined as: (i) X — $Y := X \rightarrow Y \cup X \leftarrow Y$, (ii) $X\, o$—$o\, Y := X \rightarrow Y \cup X \leftarrow Y \cup X \leftrightarrow Y$, (iii) $X\, o\!\rightarrow Y := X \rightarrow Y \cup X \leftrightarrow Y$.

Definition 3.8 ($\mathcal{C}$-essential graph)**.** For any DAG D, the edge union of all $\mathcal{C}$-closure graphs that are Markov equivalent to $\mathcal{S}_{\mathcal{C}}(D)$ is called the $\mathcal{C}$-essential graph of D, denoted as $\varepsilon_{\mathcal{C}}(D)$.

To understand the edge union operation in Definition 3.7 and its connection to the construction of the $\mathcal{C}$-essential graph, consider two Markov equivalent $\mathcal{C}$-closure graphs $\mathcal{S}_{\mathcal{C}}(D), \mathcal{S}_{\mathcal{C}}(D')$ shown by Figures 2(c) and 2(d). We can take the union of the edge $Z \leftrightarrow Y$ in $\mathcal{S}_{\mathcal{C}}(D)$ and the edge $Z \rightarrow Y$ in $\mathcal{S}_{\mathcal{C}}(D')$ to derive the edge $Zo\!\rightarrow Y$ in the $\mathcal{C}$-essential graph shown in Figure 2(e). From both Definition 3.7 and Definition 3.8, we see that a directed edge $\rightarrow$ appears in a $\mathcal{C}$-essential graph only if such directed edge appears in all $\mathcal{C}$-closure graphs that are Markov equivalent. It is important to know the difference between the edges o—o and — from a causal viewpoint. The former says that there exists a $\mathcal{C}$-closure graph in the equivalence class where two variables cannot be a cause of each other. The latter indicates that there exists no $\mathcal{C}$-closure graph in the equivalence class where a bidirected edge exists between the two. Note that any edge that appears in the $\mathcal{C}$-essential graph may not be present in the true DAG, regardless of the circle marks. It is because some separating sets that are required to d-separate two variables may not appear in $\mathcal{C}$.

Assumption 3.9 ($\mathcal{C}$-faithfulness)**.** For a given set $\mathcal{C}$ and a DAG $D = (\mathbf{V}, \mathbf{E})$, for any $X, Y \in \mathbf{V}$ and any $\mathbf{C} \in \mathcal{C}$, X and Y are conditionally independent given $\mathbf{C}$ if and only if they are d-separated given $\mathbf{C}$ in D.

Under Assumption 3.9, we present an algorithm for learning

causal graphs from a conditionally closed set of CIs. We call this algorithm $\mathcal{C}$-PC and it is shown in Algorithm 1. Next, in Theorem 3.11, we prove that $\mathcal{C}$-PC is sound for learning $\mathcal{C}$-essential graphs. We show that the properties of k-PC (Kocaoglu, 2023) can be extended to the setting where the conditioning sets are restricted to those that are given by a set $\mathcal{C}$. For details, please see Appendix C. We also show that $\mathcal{C}$-PC is sound and complete for learning the partial ancestral graph (PAG) of the $\mathcal{C}$-closure graph of any DAG D in Corollary 3.10.

Corollary 3.10. *$\mathcal{C}$-PC without Step 6 is sound and complete for learning $PAG(\mathcal{S}_{\mathcal{C}}(D))$ of any DAG D.*

Theorem 3.11. *$\mathcal{C}$-PC algorithm is sound for learning $\mathcal{C}$-essential graph given a conditional independence oracle under the causal Markov and $\mathcal{C}$-faithfulness assumptions, i.e., if $\mathcal{C}$-PC returns M, we have $\varepsilon_{\mathcal{C}}(D) \subseteq M \subseteq PAG(\mathcal{S}_{\mathcal{C}}(D))$*

3.2 DISCUSSION ON FINDING THE CONDITIONALLY-CLOSED SET $\mathcal{C}$

We present a heuristic to find the conditionally-closed set $\mathcal{C}$ in Algorithm 2. The intuition behind Algorithm 2 is as follows. If each entry in the contingency table constructed for a pair of variables is greater than 5 given $\mathbf{C}$ for all $\mathbf{c}$ such that $\mathbf{C} = \mathbf{c}$, we call $\mathbf{C}$ reliable. If $\mathbf{C}$ is reliable for more than a half of all pairs of variables, we include $\mathbf{C}$ in the conditionally-closed set. This heuristic is inspired by Agresti (2012) for the rule of expected frequencies being at least 5 as a guideline to ensure the accuracy of the chi-square approximation, especially for small sample sizes. We note that our main algorithm, $\mathcal{C}$-PC (see Algorithm 1), is not limited to a specific CI test. This heuristic potentially *skips* over some conditioning sets due to inefficient samples for us to rely on the CI test result.

3.3 AN EXAMPLE OF THE EXECUTION OF $\mathcal{C}$-PC

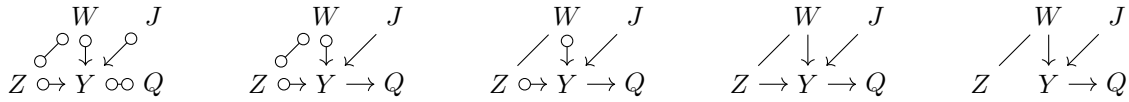
In this section, we provide an example of the execution of Algorithm 1. Suppose the ground truth is a DAG D shown in Figure 2(a). We will continue to let $\mathcal{C} = \{\emptyset, \{Y\}\}$. After Step 4, we see that Algorithm 1 orients unshielded colliders according to $\mathcal{C}$ as shown by Figure 3(a). Then, $\mathcal{C}$-PC follows the same orientation rules in k-PC shown by Algorithm 2 in the Appendix A. Note that these orientation rules do not depend on k in k-PC. By Step 5, $\mathcal{C}$-PC applies $\mathcal{R}1$ as in FCI orientation rules (Zhang, 2008a) to orient Yo—oQ as $Y \to Q$ (see Step 1 in Algorithm 2 in Appendix A). It also removes the circle mark at J due to $\mathcal{R}11$ in Algorithm 2 since Y is not adjacent to any member with which J has an edge o—o connected. Finally, $\mathcal{C}$-PC applies $\mathcal{R}12$ in Algorithm 2 to the edge Wo—oZ and orient it as W—Z, giving the final output shown in Figure 3(c), which is consistent with the $\mathcal{C}$-essential graph of D in Figure 2(e).

3.4 DIFFERENCES BETWEEN PC, k-PC, AND $\mathcal{C}$-PC

A natural question arises: how is $\mathcal{C}$-PC different from the PC algorithm (Spirtes et al., 2001) if one can assume conditional dependence given a subset that is not in $\mathcal{C}$? In general, PC (Spirtes et al., 2001), k-PC (Kocaoglu, 2023), and the proposed $\mathcal{C}$-PC begin with a complete graph, then for any two adjacent variables A, B, the PC algorithm will test whether A and B are conditionally independent given $\mathbf{Z}$ for any subset $\mathbf{Z}$ in the adjacency set of A or in the adjacency set of B. For $k = 1$, k-PC will test whether A and B are conditionally independent given any conditioning set up to size 1. Suppose Y is in one of the adjacency sets. Given $\mathcal{C} = \{\emptyset, \{Y\}\}$, the $\mathcal{C}$-PC will test whether $(A \perp\!\!\!\perp B)_P$ and $(A \perp\!\!\!\perp B|Y)_P$. In terms of characteristics, k-PC subsumes PC and $\mathcal{C}$-PC subsumes both k-PC and PC by design as one can specify $\mathcal{C}$ to include all conditioning sets of sizes up to k as well.

Here, we will use the same example shown by Figure 3 to illustrate their differences. When PC assumes conditional dependence for the conditioning sets that are not in $\mathcal{C}$. It will output the result shown in Figure 3(d). Note that PC will orient the edge (Z, Y) as $Z \to Y$. This is incorrect as Z does not cause Y in the ground truth D in Figure 2(a). Unlike k-PC, $\mathcal{C}$-PC does not need to exhaust all CI statements of order up to k for any $k > 0$. As shown by Figure 3(e), k-PC (Algorithm 3 in Appendix A) removes the edge (Z, Y) by Step 3 since it also conducts CI tests with the conditioning set $\{W\}$ for $k = 1$.

We observe that constraint-based algorithms often face challenges with false discovery rates (FDR) due to sequential hypothesis testing Li and Wang (2009). We can compare PC, k-PC and $\mathcal{C}$-PC in terms of the number of CI tests with the same example based on Figure 3. Since we restrict PC to only conduct CI tests based on the conditionally-closed set $\mathcal{C}$, both PC and $\mathcal{C}$-PC will have the same number of CI tests in this case, which is a total of 14 since there will be $\binom{5}{2}$ marginal tests plus 4 more CI tests: $(Z \perp\!\!\!\perp W|Y), (Z \perp\!\!\!\perp Q|Y), (W \perp\!\!\!\perp Q|Y), (J \perp\!\!\!\perp Q|Y)$. If we do not restrict PC to conduct CI tests based on $\mathcal{C}$, PC algorithm will have a much larger number of CI tests as it will conduct CI tests based on conditioning sets of size from 0 up to at most $|\mathbf{V}| - 2$ by design. Specifically, for the best case in PC in the example with faithfulness assumption, it will conduct all $\binom{5}{2} = 10$ marginal tests, with $(Z \perp\!\!\!\perp Y|W), (Z \perp\!\!\!\perp Q|W), (J \perp\!\!\!\perp Q|Y), (W \perp\!\!\!\perp Q|Y)$. Then, there will be 4 edges left: (Z, W), (W, Y), (J, Y), (Y, Q). For each edge, PC will conduct the following number of tests based on the cardinality of the adjacency sets of the pair: $(Z, W) : 1, (W, Y) : 7, (J, Y) : 3, (Q, Y) : 3$. Therefore, the total number of CI tests for PC is 28 tests in the best case. Regarding k-PC ($k = 1$), for the best case (as there is randomness due to the order of CI tests), it will con-



(a) M after Step 4 (b) M after Step 5 with $\mathcal{R}11$ applied to J (c) M after Step 5 with $\mathcal{R}12$ with applied to W (d) PC (Spirtes et al., 2001) output with CI tests based on $\mathcal{C}$ (e) k-PC (Kocaoglu, 2023) output with $k = 1$

Figure 3: (a)-(c): Given $\mathcal{C} = \{\emptyset, \{Y\}\}$, this is an example of the execution of Algorithm 1. Particularly, 3(c) shows the output of $\mathcal{C}$-PC for learning the ground truth in Figure 2(a). (d): The output of PC algorithm when the conditioning sets are restricted to $\mathcal{C}$ and assume conditional dependence whenever the conditioning sets are not in $\mathcal{C}$ for learning the ground truth in Figure 2(a). (e): The output of k-PC after using all CI tests with conditioning sets of size up to 1 for learning the ground truth in Figure 2(a).

duct all marginal tests $\binom{5}{2} = 10$, with $(Z \perp\!\!\!\perp Y|W), (Z \perp\!\!\!\perp Q|W), (J \perp\!\!\!\perp Q|Y), (W \perp\!\!\!\perp Q|Y)$, plus another 12 tests as there will be 4 edges left in the learned skeleton and k-PC will test each pair conditioning on each of the 3 nodes. Thus, the total is 26 CI tests.

Since we assume only $\mathcal{C}$-faithfulness, we assume all CI conducted based on $\mathcal{C}$ are correct. Therefore, k-PC, $(k = 1)$ may remove more edges as it faces the 4 remaining edges in the learned skeleton after conducting the first 14 tests mentioned previously due to unreliable CI tests with the conditioning sets that are not in $\mathcal{C}$. By reducing the number of hypotheses, our method demonstrates strong potential for effective FDR control in practical applications. We leave this interesting direction for future work.

When the conditionally closed set $\mathcal{C}$ is defined to be small, the proposed algorithm $\mathcal{C}$-PC run time complexity is lower than that of the PC algorithm as it does not condition on all subsets for testing conditional independence. $\mathcal{C}$-PC can also achieve the same complexity as k-PC for $k = 0$, which only conducts $\binom{n}{2}$ many CI tests. Note that our algorithm allows the selection of a particular conditioning set beyond the empty set, significantly improving its time complexity compared to k-PC for any $k > 0$, which requires exhausting all conditioning sets of size up to k.

3.5 TIME COMPLEXITY OF C-PC

We now discuss the complexity of $\mathcal{C}$-PC. Let n be the number of nodes and m be the number of variables that we do not want to condition on. Here, we give the complexity of $\mathcal{C}$-PC based on the purpose of setting $\mathcal{C}$. For example, if $\{V_1\} \notin \mathcal{C}$, then V_1 should not be in any member of $\mathcal{C}$ either. Hence, the largest plausible conditioning set size is $n - m - 2$ in $\mathcal{C}$. Given our results, we can test whether two DAGs are $\mathcal{C}$-Markov equivalence by constructing two $\mathcal{C}$-closure graphs and checking whether they are Markov equivalent. It would take $\mathcal{O}(n^{n-m-2})$ to determine whether two variables are adjacent based on the complexity of searching through $\begin{pmatrix} n-2 \\ n-m-2 \end{pmatrix}$ possible conditioning sets. Hence, we can roughly bound the complexity of $\mathcal{C}$-PC by $\mathcal{O}(n^{2(n-m-2)})$ since we will not be searching for separating set beyond the $\mathcal{O}(n^{n-m-2})$ subsets of size at most $n - m - 2$.

3.6 PC-LIKE ADJACENCY SEARCH STRATEGY VIOLATES THE SOUNDNESS OF $\mathcal{C}$-PC

It is natural to consider applying the adjacency search strategy from the PC algorithm to $\mathcal{C}$-PC. However, we present an example where directly adopting a PC-like adjacency search strategy Spirtes et al. (2001) leads to a violation of the soundness of $\mathcal{C}$-PC. The adjacency search strategy is, for each adjacent pair (X, Y), the algorithm checks each subset of the adjacency set of X and the adjacency set of Y for finding any separating set of size up to k. Here, we use Figure 4 as an example to show why this strategy is not sound. In this example, suppose that $\{E, C, X\}$ and $\{G, U, V\}$ are in $\mathcal{C}$. Note that if $(A \perp\!\!\!\perp G|E, C, X)_P$ is observed before testing $(A \perp\!\!\!\perp X|G, U, V)_P$, using the PC-like adjacency search strategy in $\mathcal{C}$-PC will orient $X \to A \leftarrow B$, which in turn will orient $A \leftrightarrow B$ since $(X \perp\!\!\!\perp B|G, U, V)_P$. Note that if $A \leftrightarrow B$ is oriented, that implies that A cannot be a parent of B nor B can be a parent of A in the true DAG. Therefore, using this PC-like adjacency search strategy will violate the soundness of $\mathcal{C}$-PC.

3.7 DOWNSTREAM TASKS FOR USING $\mathcal{C}$-ESSENTIAL GRAPHS

In this section, we discuss potential downstream applications of $\mathcal{C}$-essential graphs. One promising direction is exploring the identifiability of causal effects under $\mathcal{C}$-Markov equivalence, potentially by extending results from partial identification Jaber et al. (2019). Other applications include identifying invariant predictors under partial structures for distribution shifts Subbaswamy and Saria (2020) and performing root cause analysis. Additionally, the extra edge orientations provided by $\mathcal{C}$-PC could improve the efficiency of IDA-type algorithms Fang and He (2020).

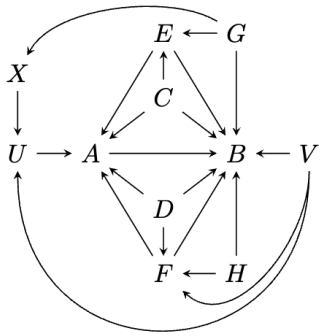


Figure 4: This is a DAG taken from Figure 1b in Schubert et al. (2025). If $(A \perp\!\!\!\perp G|E, C, X)_P$ is observed before testing $(A \perp\!\!\!\perp X|G, U, V)_P$, using the PC-like adjacency search strategy in $\mathcal{C}$-PC will orient $X \to A \leftarrow B$, which in turn will orient $A \leftrightarrow B$ since $(X \perp\!\!\!\perp B|G, U, V)_P$. Note that if $A \leftrightarrow B$ is oriented, that implies that A cannot be parent of B nor B can be a parent of A in the true DAG, which violates the soundness of $\mathcal{C}$-PC.

4 EXPERIMENTS

In this section, we present a synthetic experiment and a real-world experiment that involves missing data. The purpose of these experiments is to demonstrate how $\mathcal{C}$-PC can avoid relying on some CI tests that are likely to be unreliable due to finite sample under large support of the conditioning set, which can violate the faithfulness assumption in practice.

4.1 SYNTHETIC EXPERIMENT: CONDITIONING ON VARIABLES WITH LARGE SUPPORT

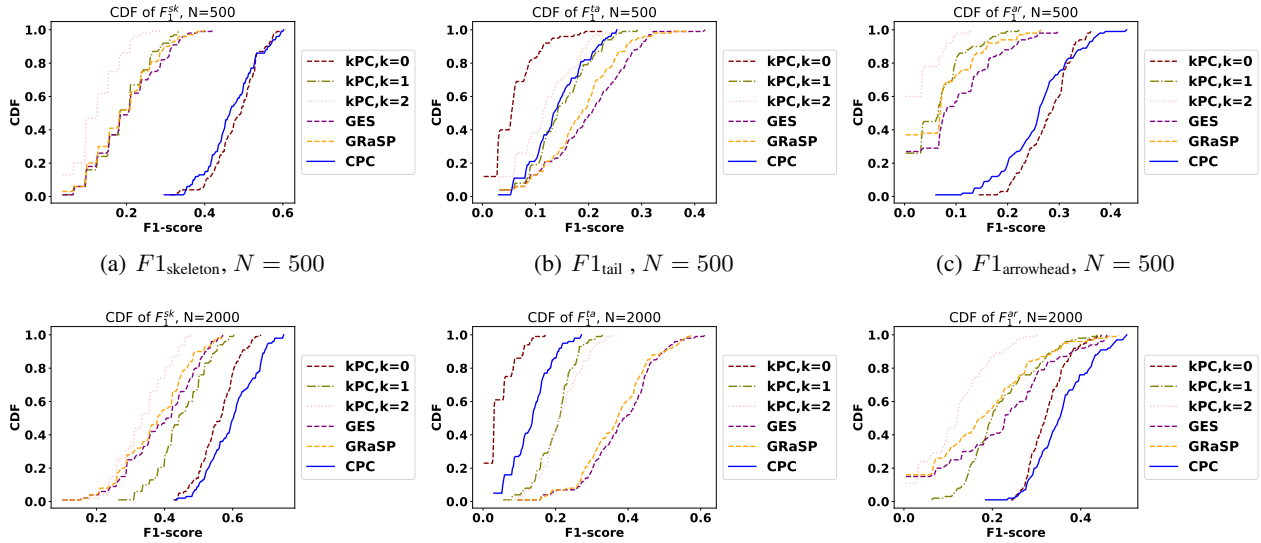
Experimental Setup. We randomly generate 100 causal DAGs of size 30. Each DAG has 60 randomly assigned edges. For each DAG, we sample the dataset three times based on a conditional probability table that is randomly assigned. Each discrete variable is randomly assigned to have either 2 or 30 states, with probabilities of 0.7 and 0.3, respectively. We include the following baselines: k-PC (Kocaoglu, 2023), GES (Chickering, 2002), GRaSP (Lam et al., 2022). Both GES and GRaSP are set to use BDeu scores. For k-PC and $\mathcal{C}$-PC, we use chi-squared tests and set $\alpha = 0.05$. For $\mathcal{C}$-PC, we employ Algorithm 2 as a heuristic to find the conditionally closed set $\mathcal{C}$ and bound any conditioning set in $\mathcal{C}$ to be of size up to 1 to limit the runtime of Algorithm 2. We repeat the experiment with sample sizes of $\{500, 2000\}$. Performance is evaluated based on the F1 scores for the skeletons $F1_{\text{skeleton}}$, tails $F1_{\text{tail}}$ and arrowheads $F1_{\text{arrowhead}}$ against the true essential graph. We implement GES and GRaSP by using `causal-learn` (Zheng et al., 2024) in Python. The data is generated via `pyAgrum` (Ducamp et al., 2020) in Python. We also provide the two-sample t-test results of the average F1-skeleton, F1-tail and F1-arrowhead scores. We also include an additional result about a variant of $\mathcal{C}$-PC named $\mathcal{C}$-PC-Path in Appendix E.2, which only conducts CI tests for a pair of variables X, Y when at least one member in the conditioning set is on some paths between X and Y in the learned skeleton (Li et al., 2019).

Discussion. As shown by Figure 5, we see that $\mathcal{C}$-PC (blue) improves more quickly in terms of $F1_{\text{skeleton}}$ and $F1_{\text{arrowhead}}$ when the sample size increases and outperforms other baselines when the sample size is 2000. Based on Figure 5(a) and 5(d), we note that k-PC suffers from conditioning on variables that have a high number of states for $k > 0$ while $\mathcal{C}$-PC performs similarly with k-PC with $k = 0$. This is because the conditionally closed set includes an empty set such that $\mathcal{C}$-PC must conduct all marginal CI tests and $\mathcal{C}$-PC is set to perform CI tests of order 1 with the sets found by Algorithm 2. As the sample size increases, from Figures 5(a) and 5(c), we see that the gaps between $\mathcal{C}$-PC and k-PC with $k \in \{1, 2\}$ shrink as expected. A similar pattern also appears in terms of $F1_{\text{arrowhead}}$ as shown by Figures 5(d) and 5(f). From Figure 9 in the Appendix E.2, we also note that $\mathcal{C}$-PC-Path does not necessarily improve $\mathcal{C}$-PC in this experiment. It has been shown that doing so can improve the performance of the PC algorithm for consistency of the separating sets (Li et al., 2019). Here, we show that our heuristic of selecting conditioning sets for $\mathcal{C}$ alone has improved the performance significantly. The two-sample t-tests in the Appendix E.2 also show that the difference between the averages of $F1_{\text{skeleton}}$ and $F1_{\text{arrowhead}}$ between $\mathcal{C}$-PC and all other baselines are statistically significant with 0.95 significance level when $N = 2000$.

4.2 REAL-WORLD EXPERIMENT: THE COGNITION AND AGING USA (COGUSA) STUDY

Experimental Setup. We want to explore the utility of $\mathcal{C}$-PC in the presence of missing data where one would perform test-wise deletion CI tests for learning causal graphs. We use the publicly available longitudinal dataset from the Cognition and Aging USA (CogUSA) study (McArdle John and Robert, 2007-2009). The dataset contains 16 discrete variables that are related to neuropsychological activities in the study with a total of 1514 samples. Out of 16 variables, 8 variables have missing values. We report the variables and their respective number of states and missing samples in the form (variable, number of states, number of missing values) here: (`w1_D_dr`, 11, 118), (`w1_D_ir`, 11, 117), (`w1_D_scis`, 2, 1), (`w2_D_bc`, 2, 285), (`w2_D_bc`, 10, 317), (`w2_D_ir`, 11, 315), (`w2_D_num`, 4, 285), (`w2_D_s7`, 6, 285).

We compare $\mathcal{C}$-PC with three baselines: PCMCI (Runge et al., 2019), MVPC (Tu et al., 2019), tPC (Bang et al., 2024), and k-PC ($k = 1$). We use `causal-learn` (Zheng et al., 2024) to implement MVPC and tPC and use `tigramite` to implement PCMCI in Python. We set $\mathcal{C}$-PC to only conduct CI tests up to order 1 without conditioning on those



(a) $F1_{\text{skeleton}}$, $N = 500$ (b) $F1_{\text{tail}}$, $N = 500$ (c) $F1_{\text{arrowhead}}$, $N = 500$

(d) $F1_{\text{skeleton}}$, $N = 2000$ (e) $F1_{\text{tail}}$, $N = 2000$ (f) $F1_{\text{arrowhead}}$, $N = 2000$

Figure 5: Empirical cumulative distribution function of various F1 scores on 100 random DAGs with 30 nodes where the number of states is assigned to be 2 or 30 with probability 0.7 and 0.3 respectively. The lower and farther from the left side the better. **Top**: Skeleton F1 scores $F1_{\text{skeleton}}$, tail F1 scores $F1_{\text{tail}}$ and arrowhead F1 scores $F1_{\text{tail}}$ for sample size of $N = 500$. **Bottom**: Skeleton F1 scores $F1_{\text{skeleton}}$, tail F1 scores $F1_{\text{tail}}$ and arrowhead F1 scores $F1_{\text{tail}}$ for sample size of $N = 2000$. Overall, we see that $\mathcal{C}$-PC (blue) outperforms other baselines at $N = 2000$ in terms of $F1_{\text{skeleton}}$, $F1_{\text{arrowhead}}$ with a small expense in $F1_{\text{tail}}$. When $N = 500$, $\mathcal{C}$-PC performs comparably with k-PC ($k = 0$) as expected as $\mathcal{C}$-PC also conducts all marginal independence tests.

8 variables that have missing values. All algorithms use test-wise deletion chi-squared tests with $\alpha = 0.05$. Here, test-wise deletion involves removing samples that contain missing values for the variables involved in conditional independence (CI) testing, ensuring that these variables have the same number of observations for the CI test. We use the same evaluation metrics as in (Strobl et al., 2018) to evaluate all the algorithms, that is, no variable with its name starting from `w2` (variables that are collected from week 2 of the study) can cause a variable with a name starting from `w1` (variables that are collected from week 1 of the study). Also, the variable (`w1_D_bc`) can only have a directed edge to the variable that represents the mental status of the participants (`w1_D_ms`). We leave the results in Appendix G due to space limit.

Discussion. From Figures 11, we see that $\mathcal{C}$-PC does not output incorrect causal relationships and can recover the directed edge from `w1_D_bc` to `w1_D_ms` highlighted in green. Although MVPC and k-PC can also recover the directed edge from `w1_D_bc` to `w1_D_ms`, we note that all of them also produce incorrect causal relationships highlighted in red. Based on Figure 13, while PCMCI does not orient any variables from `w2` to `w1`, it also fails to recover the directed edge `w1_D_bc` $\rightarrow$ `w1_D_ms`. This is expected as PCMCI involves high-order CI tests, which are more prone to errors when testing conditional independence, in contrast with our proposed algorithm $\mathcal{C}$-PC. As we can see from Figure 14, k-PC with $k = 1$ removes the edge between `w2_D_dr` and `w1_D_ir` such that it orients the unshielded triple $\langle$`w2_D_dr`, `w1_D_dr`, `w1_D_ir`$\rangle$ as a unshielded collider, which gives an incorrect relationship. From Figure 15, we observe that although tPC does not orient any variable from week 2 to week 1 due to background knowledge, tPC fails to capture the directed edge from `w1_D_bc` to `w1_D_ms`. $\mathcal{C}$-PC effectively prevents this by not conditioning on the variables that have missing values since test-wise deletion can be prone to inducing conditional independence when the support of the conditioning set is large. In contrast, $\mathcal{C}$-PC only gives a bidirected edge (`w2_D_dr`) $\leftrightarrow$ (`w1_D_ir`) as shown by Figure 11.

5 CONCLUSION

We propose a sound algorithm called $\mathcal{C}$-PC for learning causal graphs from a collection of conditioning sets known as conditionally closed sets. We extend an existing algorithm called k-PC that exhausts all CI tests of order up to some integer k to a setting where CI tests are restricted to a collection of conditioning sets. We perform experiments to show the utility of the proposed method in the presence of variables with large support under limited samples. One can view $\mathcal{C}$-PC as a more generalized version of k-PC as the conditionally closed set can flexibly include any conditioning set of size greater than zero. For future work, we want to investigate whether one can systematically leverage arbitrary CI statements for learning causal graphs.

Acknowledgements

This research has been supported in part by NSF CAREER 2239375, IIS 2348717, Amazon Research Award and Adobe Research. This work was also funded in part by the National Science Foundation (NSF) awards, CCF-1918483, CAREER IIS-1943364 and CNS-2212160, Ford, AMD, NVidia, CISCO, and Amazon.

References

Alan Agresti. *Categorical data analysis*, volume 792. John Wiley & Sons, 2012.

Syed Sazzad Ahmed, Swarup Roy, and Jugal Kalita. Assessing the effectiveness of causality inference methods for gene regulatory networks. *IEEE/ACM transactions on computational biology and bioinformatics*, 17(1):56–70, 2018.

Christine W Bang, Janine Witte, Ronja Foraita, and Vanessa Didelez. Improving finite sample performance of causal discovery by exploiting temporal structure. *arXiv preprint arXiv:2406.19503*, 2024.

Elias Bareinboim and Judea Pearl. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113(27):7345–7352, 2016.

Ruta Binkyte, Carlos Antonio Pinzón, Szilvia Lestyán, Kangsoo Jung, Héber Hwang Arcolezi, and Catuscia Palamidessi. Causal discovery under local privacy. In *Causal Learning and Reasoning*, pages 325–383. PMLR, 2024.

Peter Bühlmann, Jonas Peters, and Jan Ernest. Cam: Causal additive models, high-dimensional order search and penalized regression. 2014.

Zhanrui Cai, Dong Xi, Xuan Zhu, and Runze Li. Causal discoveries for high dimensional mixed data. *Statistics in Medicine*, 41(24):4924–4940, 2022.

Rui Chang, Jonathan R Karr, and Eric E Schadt. Causal inference in biology networks with integrated belief propagation. In *Pacific Symposium on Biocomputing Co-Chairs*, pages 359–370. World Scientific, 2014.

Silvia Chiappa. Path-specific counterfactual fairness. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7801–7808, 2019.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

Max Chickering. Statistically efficient greedy equivalence search. In *Conference on Uncertainty in Artificial Intelligence*, pages 241–249. Pmlr, 2020.

Tom Claassen and Ioan G Bucur. Greedy equivalence search in the presence of latent confounders. In *Uncertainty in Artificial Intelligence*, pages 443–452. Pmlr, 2022.

Jordan M Cloyd, Victor Heh, Timothy M Pawlik, Aslam Ejaz, Mary Dillhoff, Allan Tsung, Terence Williams, Laith Abushahin, John FP Bridges, and Heena Santry. Neoadjuvant therapy for resectable and borderline resectable pancreatic cancer: a meta-analysis of randomized controlled trials. *Journal of clinical medicine*, 9(4): 1129, 2020.

Diego Colombo, Marloes H Maathuis, et al. Order-independent constraint-based causal structure learning. *J. Mach. Learn. Res.*, 15(1):3741–3782, 2014.

Spencer Compton, Murat Kocaoglu, Kristjan Greenewald, and Dmitriy Katz. Entropic causal inference: Identifiability and finite sample results. *Advances in Neural Information Processing Systems*, 33:14772–14782, 2020.

Gaspard Ducamp, Christophe Gonzales, and Pierre-Henri Wuillemin, pages = 609-612. agrum/pyagrum : a toolbox to build models and algorithms for probabilistic graphical models in python. In Manfred Jaeger and Thomas Dyhre Nielsen, editors, *Proceedings of the 10th International Conference on Probabilistic Graphical Models*, volume 138 of *Proceedings of Machine Learning Research*. PMLR, 23–25 Sep 2020. URL `https://proceedings.mlr.press/v138/ducamp20a.html`.

Zhuangyan Fang and Yangbo He. Ida with background knowledge. In *Conference on Uncertainty in Artificial Intelligence*, pages 270–279. PMLR, 2020.

Stefan Feuerriegel, Dennis Frauen, Valentyn Melnychuk, Jonas Schweisthal, Konstantin Hess, Alicia Curth, Stefan Bauer, Niki Kilbertus, Isaac S Kohane, and Mihaela van der Schaar. Causal machine learning for predicting treatment outcomes. *Nature Medicine*, 30(4):958–968, 2024.

Maxime Gasse, Alex Aussem, and Haytham Elghazel. A hybrid algorithm for bayesian network structure learning with application to multi-label learning. *Expert Systems with Applications*, 41(15):6755–6772, 2014.

Xianjie Guo, Yujie Wang, Xiaoling Huang, Shuai Yang, and Kui Yu. Bootstrap-based causal structure learning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 656–665, 2022.

Joseph F Hair Jr and Marko Sarstedt. Data, measurement, and causal inferences in machine learning: opportunities and challenges for marketing. *Journal of Marketing Theory and Practice*, 29(1):65–77, 2021.

Anthony D Harris, Jessina C McGregor, Eli N Perencevich, Jon P Furuno, Jingkun Zhu, Dan E Peterson, and Joseph Finkelstein. The use and interpretation of quasi-experimental studies in medical informatics. *Journal of the American Medical Informatics Association*, 13(1): 16–23, 2006.

D Heckerman, C Meek, and G Cooper. A bayesian approach to casual discovery: Computation causation and discovery, 1999.

Antti Hyttinen, Patrik O Hoyer, Frederick Eberhardt, and Matti Järvisalo. Discovering cyclic causal models with latent variables: a general sat-based procedure. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 301–310, 2013.

Maximilian Ilse, Jakub M Tomczak, and Patrick Forré. Selecting data augmentation for simulating interventions. In *International conference on machine learning*, pages 4555–4562. PMLR, 2021.

Amin Jaber, Jiji Zhang, and Elias Bareinboim. Causal identification under markov equivalence: Completeness results. In *International Conference on Machine Learning*, pages 2981–2989. PMLR, 2019.

Yonghan Jung, Shiva Kasiviswanathan, Jin Tian, Dominik Janzing, Patrick Blöbaum, and Elias Bareinboim. On measuring causal contributions via do-interventions. In *International Conference on Machine Learning*, pages 10476–10501. PMLR, 2022.

N Keum, DH Lee, DC Greenwood, JE Manson, and E Giovannucci. Vitamin d supplementation and total cancer incidence and mortality: a meta-analysis of randomized controlled trials. *Annals of Oncology*, 30(5):733–743, 2019.

Murat Kocaoglu. Characterization and learning of causal graphs with small conditioning sets. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 74140–74179. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/eaef3b49866b942041a34bb8da397eb7-Paper-Conference.pdf.

Murat Kocaoglu, Alexandros Dimakis, Sriram Vishwanath, and Babak Hassibi. Entropic causal inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *Advances in neural information processing systems*, 30, 2017.

Wai-Yin Lam, Bryan Andrews, and Joseph Ramsey. Greedy relaxations of the sparsest permutation algorithm. In *Uncertainty in Artificial Intelligence*, pages 1052–1062. PMLR, 2022.

Olivier Ledoit and Michael Wolf. Improved estimation of the covariance matrix of stock returns with an application to portfolio selection. *Journal of empirical finance*, 10 (5):603–621, 2003.

Honghao Li, Vincent Cabeli, Nadir Sella, and Hervé Isambert. Constraint-based causal structure learning with consistent separating sets. *Advances in Neural Information Processing Systems*, 32, 2019.

Junning Li and Z Jane Wang. Controlling the false discovery rate of the association/causality structure learned with the pc algorithm. *Journal of Machine Learning Research*, 10 (2), 2009.

Meng Ma, Jingmin Xu, Yuan Wang, Pengfei Chen, Zonghua Zhang, and Ping Wang. Automap: Diagnose your microservice-based web applications automatically. In *Proceedings of The Web Conference 2020*, pages 246–258, 2020.

Rodgers Willard McArdle John and Willis Robert. Cognition and aging in the usa (cogusa), 2007-2009.

Franklin G Miller and Howard Brody. What makes placebo-controlled trials unethical? *The American Journal of Bioethics*, 2(2):3–9, 2002.

Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. On the role of sparsity and dag constraints for learning linear dags. *Advances in Neural Information Processing Systems*, 33:17943–17954, 2020.

Ignavier Ng, Shengyu Zhu, Zhuangyan Fang, Haoyang Li, Zhitang Chen, and Jun Wang. Masked gradient-based causal structure learning. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, pages 424–432. SIAM, 2022.

Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

Judea Pearl. Causal inference in statistics: An overview. 2009a.

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

Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic books, 2018.

Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 78 (5):947–1012, 2016.

Francesco Paolo Prete, Angela Pezzolla, Fernando Prete, Mario Testini, Rinaldo Marzaioli, Alberto Patriti, Rosa Maria Jimenez-Rodriguez, Angela Gurrado, and Giovanni FM Strippoli. Robotic versus laparoscopic minimally invasive surgery for rectal cancer: a systematic review and meta-analysis of randomized controlled trials. *Annals of surgery*, 267(6):1034–1046, 2018.

Joseph Ramsey, Peter Spirtes, and Jiji Zhang. Adjacency-faithfulness and conservative causal inference. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, pages 401–408, 2006.

Garvesh Raskutti and Caroline Uhler. Learning directed acyclic graph models based on sparsest permutations. *Stat*, 7(1):e183, 2018.

Alex Reinhart, Logan Brooks, Maria Jahja, Aaron Rumack, Jingjing Tang, Sumit Agrawal, Wael Al Saeed, Taylor Arnold, Amartya Basu, Jacob Bien, et al. An open repository of real-time covid-19 indicators. *Proceedings of the National Academy of Sciences*, 118(51):e2111452118, 2021.

Thomas Richardson and Peter Spirtes. Ancestral graph markov models. *The Annals of Statistics*, 30(4):962–1030, 2002.

Dominick A Rizzi and Stig Andur Pedersen. Causality in medicine: towards a theory and terminology. *Theoretical Medicine*, 13:233–254, 1992.

Raanan Y Rohekar, Yaniv Gurwicz, and Shami Nisimov. Causal interpretation of self-attention in pre-trained transformers. *Advances in Neural Information Processing Systems*, 36, 2024.

Jakob Runge, Peer Nowack, Marlene Kretschmer, Seth Flaxman, and Dino Sejdinovic. Detecting and quantifying causal associations in large nonlinear time series datasets. *Science advances*, 5(11):eaau4996, 2019.

Mark Schmidt, Alexandru Niculescu-Mizil, Kevin Murphy, et al. Learning graphical model structure using l1-regularization paths. In *AAAI*, volume 7, pages 1278–1283, 2007.

Mátyás Schubert, Tom Claassen, and Sara Magliacane. SNAP: Sequential non-ancestor pruning for targeted causal effect estimation with an unknown graph. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025. URL https://openreview.net/forum?id=0gEjlLdjK9.

Rajen D Shah and Jonas Peters. The hardness of conditional independence testing and the generalised covariance measure. 2020.

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

Shohei Shimizu, Takanori Inazumi, Yasuhiro Sogawa, Aapo Hyvarinen, Yoshinobu Kawahara, Takashi Washio, Patrik O Hoyer, Kenneth Bollen, and Patrik Hoyer. Directlingam: A direct method for learning a linear non-gaussian structural equation model. *Journal of Machine Learning Research-JMLR*, 12(Apr):1225–1248, 2011.

Ilya Shpitser and Judea Pearl. Complete identification methods for the causal hierarchy. 2008.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2001.

Eric V Strobl, Shyam Visweswaran, and Peter L Spirtes. Fast causal inference with non-random missingness by test-wise deletion. *International journal of data science and analytics*, 6:47–62, 2018.

Adarsh Subbaswamy and Suchi Saria. I-spec: An end-to-end framework for learning transportable, shift-stable models. *arXiv preprint arXiv:2002.08948*, 2020.

Adarsh Subbaswamy, Peter Schulam, and Suchi Saria. Preventing failures due to dataset shift: Learning predictive models that transport. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 3118–3127. PMLR, 2019.

Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1): 31–78, 2006.

Ruibo Tu, Cheng Zhang, Paul Ackermann, Karthika Mohan, Hedvig Kjellström, and Kun Zhang. Causal discovery in the presence of missing data. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1762–1770. Pmlr, 2019.

Tyler J VanderWeele and Whitney R Robinson. On the causal interpretation of race in regressions adjusting for confounding and mediating variables. *Epidemiology*, 25 (4):473–484, 2014.

Hal R Varian. Causal inference in economics and marketing. *Proceedings of the National Academy of Sciences*, 113 (27):7310–7315, 2016.

T Verma. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence, 1991*. Elsevier, 1991.

Julius Von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data

augmentations provably isolates content from style. *Advances in neural information processing systems*, 34: 16451–16467, 2021.

X Wang, Y Du, S Zhu, L Ke, Z Chen, J Hao, and J Wang. Ordering-based causal discovery with reinforcement learning. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*, pages 3566–3573. IJCAI International Joint Conferences on Artificial Intelligence Organization, 2021.

Marcel Wienöbst and Maciej Liskiewicz. Recovering causal structures from low-order conditional independencies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10302–10309, 2020.

Anja Wille and Peter Bühlmann. Low-order conditional independence graphs for inferring genetic networks. *Statistical applications in genetics and molecular biology*, 5 (1), 2006.

Yongkai Wu, Lu Zhang, Xintao Wu, and Hanghang Tong. Pc-fairness: A unified framework for measuring causality-based fairness. *Advances in neural information processing systems*, 32, 2019.

Jing Xiang and Seyoung Kim. A* lasso for learning a sparse bayesian network structure for continuous variables. *Advances in Neural Information Processing Systems*, 26, 2013.

Yu Yang, Sthitie Bom, and Xiaotong Shen. A hierarchical ensemble causal structure learning approach for wafer manufacturing. *Journal of Intelligent Manufacturing*, pages 1–18, 2023.

Yue Yu, Jie Chen, Tian Gao, and Mo Yu. Dag-gnn: Dag structure learning with graph neural networks. In *International Conference on Machine Learning*, pages 7154–7163. PMLR, 2019.

Jiji Zhang. Causal reasoning with ancestral graphs. *Journal of Machine Learning Research*, 9(7), 2008a.

Jiji Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16-17): 1873–1896, 2008b.

Jiji Zhang and Peter L Spirtes. Strong faithfulness and uniform consistency in causal inference. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 632–639, 2003.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in neural information processing systems*, 31, 2018.

Yujia Zheng, Biwei Huang, Wei Chen, Joseph Ramsey, Mingming Gong, Ruichu Cai, Shohei Shimizu, Peter Spirtes, and Kun Zhang. Causal-learn: Causal discovery in python. *Journal of Machine Learning Research*, 25(60):1–8, 2024.

Constraint-based Causal Discovery from a Collection of Conditioning Sets (Supplementary Material)

Kenneth Lee[1] **Bruno Ribeiro**[2] **Murat Kocaoglu**[1]

[1]Elmore Family School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA
[2]Department of Computer Science, Purdue University, West Lafayette, IN, USA

A ALGORITHMS

Algorithm 3 k-PC (Kocaoglu, 2023)

input Observational data $\mathbf{V}$, k, CI tester

Initiate a complete graph M among the set of observed variables with circle edge o—o.
Find separating sets $S_{X,Y}$ for every pair $X, Y \in \mathbf{V}$ by conditioning on subsets $\mathbf{Z} \subset \mathbf{V}$ of size at most k.
Update M by removing the edges between pairs that are separable.
Orient unshielded colliders of M: For every unshielded triple: Xo—oZo—oY, set $Xo\rightarrow Z \leftarrow oY$ where $S_{X,Y}$ does not contain Z.
$M \leftarrow$ **FCI_Orient**(M) {See Algorithm 5}
$M \leftarrow$ **kPC_Orient**(M) {See Algorithm 4}
return M

Algorithm 4 kPC_Orient (Kocaoglu, 2023)

input Mixed graph M

For any variable X that has no incoming edges, construct the sets $\mathcal{B}, \mathcal{Q}$:

$$\mathcal{B} = \{Y \in Ne(X) : Xo\rightarrow Y\}, \mathcal{Q} = \{Z \in Ne(X) : Xo\text{—}oZ\}$$

and define sets $\mathcal{B}^\star$ as the set of variables that are non-adjacent to any of the nodes in $\mathcal{Q}$ and $\mathcal{Q}^\star$ as the set of variables that are non-adjacent to other variables in $\mathcal{Q}$:

$$\mathcal{B}^\star = \{Y \in \mathcal{B} : Y, Z \text{ are non-adjacent } \forall Z \in \mathcal{Q}\}, \mathcal{Q}^\star = \{Z' \in \mathcal{Q} : Z', Z \text{ are non-adjacent } \forall Z' \neq Z, Z' \in \mathcal{Q}\}$$

$\mathcal{R}11$: Orient $Xo\rightarrow Y$ as $X \rightarrow Y, \forall Y \in \mathcal{B}^\star$
$\mathcal{R}12$: Orient Xo—oY as X—$Y, \forall Z \in \mathcal{Q}^\star$
return M

Algorithm 5 FCI_Orient (Zhang, 2008a)

input Mixed graph M

Apply the orientation rules of $\mathcal{R}1, \mathcal{R}2, \mathcal{R}3$ of Zhang (2008a) to M until none applies.
Apply the orientation rules of $\mathcal{R}8, \mathcal{R}9, \mathcal{R}10$ of Zhang (2008a)
return M

B GRAPH BASICS

Definition B.1. A graph $D = (\mathbf{V}, \mathbf{E})$ consists of a set of nodes (variables) $\mathbf{V}$ and a set of edges $\mathbf{E}$. We use (X, Y) to denote an edge between a variable X and another variable Y in D. We consider graphs that may contain directed ($\rightarrow$), undirected ($-$) edges, and bidirected ($\leftrightarrow$) edges. A directed graph has only directed edges. A partially directed graph may have both undirected and directed edges. A mixed graph can contain directed, undirected, and bidirected edges. A graph $D' = (\mathbf{V}', \mathbf{E}')$ is a *subgraph* of $D = (\mathbf{V}, \mathbf{E})$ if $\mathbf{V}' \subseteq \mathbf{V}$ and $\mathbf{E}' \subseteq \mathbf{E}$. A graph $D' = (\mathbf{V}', \mathbf{E}')$ is a *subgraph* of $D = (\mathbf{V}, \mathbf{E})$ if $\mathbf{V}' \subseteq \mathbf{V}$ and $\mathbf{E}' \subseteq \mathbf{E}$. D' is an *induced subgraph* of D if $\mathbf{E}'$ are all edges in $\mathbf{E}$ between nodes in $\mathbf{V}'$.

Definition B.2 (Path)**.** Two vertices in a graph are said to be *adjacent* if there is an edge between them. Given a partially directed graph D, a *path* from V_0 to V_n in D is a sequence of distinct vertices $\langle V_0, V_1, \ldots, V_n \rangle$ such that for $0 \leq i \leq n-1$, V_i and V_{i+1} are adjacent. It is called a *causal* (or *directed*) path from V_0 to V_n in D if V_i is a parent of V_{i+1} for $0 \leq i \leq n-1$.

Definition B.3 (Colliders)**.** A consecutive triple of nodes $\langle X, Y, Z \rangle$ on a path is called a *collider* if both the edge between X and Y and the edge between Y and Z have arrowheads pointing to Y. If additionally X and Z are not adjacent, it is called *unshielded collider*. Any other consecutive triple is called a *non-collider*. If additionally, the two end vertices of the triple are not adjacent, it is called a *unshielded non-collider*.

Definition B.4 (Ancestrality)**.** In a graph D, for any two nodes X, Y in D, if there is a directed edge $X \rightarrow Y$, then X is a *parent* of Y and Y is a *child* of X in D. If there is a causal path from X to Y, then X is called an *ancestor* of Y and Y is called a *descendant* of X. We denote a set of parents of X, a set of children of X, a set of ancestors of X, a set of descendants of X and a set of non-descendants of X in D as $Pa_D(X)$, $Ch_D(X)$, $An_D(X)$, $De_D(X)$ and $NDe_D(X)$ respectively. By convention, X is both an ancestor and a descendant of X in D. Furthermore, if $X \leftrightarrow Y$ is in G, then X is a *spouse* of Y and Y is a *spouse* of X.

Definition B.5 (DAGs, ADMGs)**.** For any two nodes X, Y in D, if there is a causal path from X to Y and an edge from Y to X, there is a *directed cycle* in D. An *almost directed cycle* occurs when there is a bidirected edge between X and Y and $X \in An_D(Y)$. A directed graph without directed cycles is called a *directed acyclic graph* (DAG) or a causal graph. An *acyclic directed mixed graph* (ADMG) is a graph that contains both bidirected edges and directed edges without directed cycles.

Definition B.6 (Ancestral graphs)**.** A mixed graph is *ancestral* if there is no directed cycle, no almost directed cyclic, and for any undirected edges $X - Y$, X and Y have no parents or spouses.

Definition B.7 (m-separation)**.** In a mixed graph, a path p between X and Y is *m-connecting* relative to $\mathbf{Z}$ if (i) every non-collider on p is not a member of $\mathbf{Z}$ and (ii) every collider on p has a descendant in $\mathbf{Z}$. $\mathbf{X}$ and $\mathbf{Y}$ are *m-separated* if there is no m-connecting path between any vertex in $\mathbf{X}$ and any vertex in $\mathbf{Y}$.

The probabilistic interpretation of ancestral graphs follows from its Markov property that if $\mathbf{X}$ and $\mathbf{Y}$ are m-separated given $\mathbf{Z}$, then $\mathbf{X}$ and $\mathbf{Y}$ are conditionally independent given $\mathbf{Z}$. In the case of DAGs, m-separation reduces to d-separation (Richardson and Spirtes, 2002).

Definition B.8 (Skeleton)**.** The skeleton of a causal graph D is an undirected graph obtained by replacing every directed edge of D with an undirected edge.

Pearl and Verma have proved the following characterization of DAGs that belong to the same Markov equivalence class.

Theorem B.9 (Verma (1991))**.** *Two DAGs are Markov equivalent if and only if they have the same skeleton and the same set of unshielded colliders.*

C PROOFS

In this section, we will show that all the lemmas and theorems in (Kocaoglu, 2023) can be extended to the setting where the conditioning sets are restricted to $\mathcal{C}$ instead of all conditioning sets of size up to some integer k.

C.1 PROOF OF LEMMA 3.5

To prove the lemma 3.5, we will first prove a few lemmas shown below.

Lemma C.1. *Consider a DAG D where $X \notin An_D(Y)$ and $Y \notin An_D(X)$, X, Y are non-adjacent and $\mathcal{C}$-covered. Then conditioned on any member of a conditionally closed set, i.e. $\mathbf{C} \in \mathcal{C}$, there exists a d-connecting path between X, Y that has an arrowhead at both X and Y.*

Proof. For the sake of contradiction suppose, conditioned on some $\mathbf{C} \in \mathcal{C}$, there is no d-connecting path with an arrow into X and an arrow into Y. Since neither X is an ancestor of Y nor Y is an ancestor of X and assuming there is no d-connecting path with an arrow into X and an arrow into Y, it must be that all d-connecting paths have colliders on them. All such colliders must be ancestors of $\mathbf{C}$.

Consider such a path p where the edge adjacent to X has a tail at X. Let K be the collider that is closest to X.

Thus, we have

$$X \to U_1 \to \ldots \to U_m \to K \leftarrow V \ldots Y \tag{3}$$

for some $\{U_i\}_{i \in m}, V$. However, this is a contradiction since X and Y are $\mathcal{C}$-covered, that is, X and Y must be d-connecting given an empty set on the path p, which is impossible. Due to symmetry between X and Y, the supposition that the only d-connecting paths must have a tail adjacent to either endpoint must be wrong. This proves the lemma. □

Lemma C.2. *Consider a DAG D where $X \notin An_D(Y)$, X, Y are non-adjacent and $\mathcal{C}$-covered. Then conditioned on any member in $\mathcal{C}$, i.e. $\forall \mathbf{C} \in \mathcal{C}$, there exists a d-connecting path between X and Y that starts with an arrow into X.*

Proof. For the sake of contradiction, suppose otherwise. Given any $\mathbf{C}$, there is no d-connecting path between X and Y that starts with an arrow into X, that is, all the d-connecting paths start with a tail at X. We will show that we can find some member $\mathbf{C}' \in \mathcal{C}$ that d-separates X, Y, which leads to a contradiction since X and Y are given to be $\mathcal{C}$-covered.

Consider any path p between X and Y that is d-connecting given $\mathbf{C}$ which starts with a tail at X. p must be either directed or there must be at least one collider along the path. Since $X \notin An_D(Y)$, it must be that the path has at least one collider. Let K be the collider that is closest to X on p as follows

$$X \to U_1 \to \ldots \to U_m \to K \leftarrow V \ldots Y \tag{4}$$

for some $\{U_i\}_i, V$. Since the path is active, this collider cannot block the path p. It must be that $K \in An_D(\mathbf{C})$. However, this is a contradiction since X and Y are $\mathcal{C}$-covered, that is, X and Y must be d-connecting given an empty set on the path p, which is impossible. Thus, the assumption that all the d-connecting paths must have a tail at X must be wrong, which proves the lemma. □

Lemma C.3. *(Kocaoglu, 2023) In a DAG D, let p be an active path between X and Y given Z and q be an active path between Y and Q given Z, where $X, Y, Q \notin Z$. If X and Q are d-separated given Z, then*

1. *Paths p, q must have no overlapping nodes and*
2. *Y must be a collider along the concatenated path and $Q \notin An_D(Z)$*

The following lemma shows that colliders that are closed in D must also be closed in the $\mathcal{C}$-closure graph $\mathcal{S}_{\mathcal{C}}(D)$.

Lemma C.4. *If a collider is blocked in D conditioned on some members $\mathbf{C}$ of a set $\mathcal{C}$, then it must also be blocked in $\mathcal{S}_{\mathcal{C}}(D)$ conditioned on $\mathbf{C}$.*

Proof. Suppose $(X \to Z \leftarrow Y)_D$ is a collider that is blocked given $\mathbf{C}$. Thus, it must be that $Z \notin An_D(\mathbf{C})$. For the sake of contradiction, assume that this collider is unblocked in $\mathcal{S}_{\mathcal{C}}(D)$. Thus, it must be the case $Z \in An_{\mathcal{S}_{\mathcal{C}}(D)}(\mathbf{C})$. This means there is a new directed path from Z to $\mathbf{C}$ in $\mathcal{S}_{\mathcal{C}}(D)$. If this path existed in D, the collider would be unblocked, which is a contradiction. Thus, at least one of the edges along this path must have been added during the construction of $\mathcal{S}_{\mathcal{C}}(D)$. Consider the collection of edges on this path that does not exist in D. Note that by construction of $\mathcal{S}_{\mathcal{C}}(D)$, a directed edge $\alpha \to \beta$ is added between a $\mathcal{C}$-covered pair α, β only if there is a directed path from α to β in D. Consider the path obtained by replacing the directed edge between any $\mathcal{C}$-covered pair along this path with the corresponding directed path in D. The resulting directed path must be in D. This shows that there was at least one path already in D that implied $Z \in An_D(\mathbf{C})$, which is a contradiction. Therefore, any collider on p that is unblocked in $\mathcal{S}_{\mathcal{C}}(D)$ must also be unblocked in D. □

Lemma C.5. *The set of ancestors of any set Z of nodes in D is identical to the set of ancestors of Z in $\mathcal{S}_{\mathcal{C}}(D)$.*

Proof. Adding edges to a graph, directed or bidirected, cannot decrease the set of ancestors of any node. We only need to show that the set of ancestors in $\mathcal{S}_{\mathcal{C}}(D)$ is not larger than the set of ancestors in D.

Suppose otherwise: A node $X \in An_{\mathcal{S}_{\mathcal{C}}}(Z)$ but $X \notin An_D(Z)$. This can only happen if a collection of edges added during the construction of $\mathcal{S}_{\mathcal{C}}(D)$ renders X an ancestor of Z. However, each such edge is added only if there is a directed path between its endpoints in D. Consider the path obtained by replacing each such added edge along the path that renders X an ancestor of Z in $\mathcal{S}_{\mathcal{C}}(D)$ with the corresponding directed paths in D. This directed path must be in D, which means that X was an ancestor of Z in D as well, which is a contradiction. □

Proof of Lemma 3.5

Lemma. $\mathcal{C}$-closure graph $\mathcal{S}_{\mathcal{C}}(D)$ of a DAG D entails the same d-separation statements conditioned any $\mathbf{C} \in \mathcal{C}$ as the DAG, i.e., $(X \perp\!\!\!\perp Y|\mathbf{C})_D \Leftrightarrow (X \perp\!\!\!\perp Y|\mathbf{C})_{\mathcal{S}_{\mathcal{C}}(D)}, \forall \mathbf{C} \in \mathcal{C}$.

Proof. Since no edge is removed during the construction of the $\mathcal{C}$-closure graph, one direction immediately follows: If $(X \perp\!\!\!\perp Y|\mathbf{C})_{\mathcal{S}_{\mathcal{C}}(D)}$, then $(X \perp\!\!\!\perp Y|\mathbf{C})_D$ for any $\mathbf{C} \in \mathcal{C}$. Therefore, we will show the other direction.

Suppose $(X \perp\!\!\!\perp Y|\mathbf{C})_D$ for any $\mathbf{C} \in \mathcal{C}$. We will show that $(X \perp\!\!\!\perp Y|\mathbf{C})_{\mathcal{S}_{\mathcal{C}}(D)}$. For the sake of contradiction, suppose otherwise. Then, there must be a d-connecting path p between X and Y given $\mathbf{C}$ in $\mathcal{S}_{\mathcal{C}}(D)$. The length of any such path must be greater than 1 since otherwise, whether the edge already existed in D or it was added during the construction of $\mathcal{S}_{\mathcal{C}}(D), X, Y$ must have been dependent given $\mathbf{C}$ in D, which is a contradiction. Since the orientation of the existing edges in D did not change in $\mathcal{S}_{\mathcal{C}}(D)$, either this path did not exist in D or that it existed but it was blocked by some collider that is not in $An_D(\mathbf{C})$. The latter is not possible due to Lemma C.4, since any unblocked collider in $\mathcal{S}_{\mathcal{C}}(D)$ must also be unblocked in D. Thus, it must be that this d-connecting path did not exist in D.

Suppose p ***does not exist in*** D. At least one edge must have been added to form this path in $\mathcal{S}_{\mathcal{C}}(D)$ during the construction of $\mathcal{S}_{\mathcal{C}}(D)$.

For any added edge $U \to V$, the following is true: Since $U \to V$ was added in $\mathcal{S}_{\mathcal{C}}(D)$, it must be the case that $V \notin An_D(U)$ since otherwise, there would be a cycle. By Lemma C.2, conditioned on $\mathbf{C}$, there exists a d-connecting path between U and V where the edge adjacent to V is into V. For any added edge $U \leftrightarrow V$, the following is true: since $U \leftrightarrow V$ was added in $\mathcal{S}_{\mathcal{C}}(D)$, it must be the case that $U \notin An_D(V)$ and $V \notin An_D(U)$. By Lemma C.1, conditioned on $\mathbf{C}$, there exists a d-connecting path between U and V, where the edge adjacent to U is into U and the edge adjacent to V is into V. Call any such path implied by these lemmas a replacement path. Note that a replacement path might be directed or not.

Consider a path q in D that is obtained from p by switching the edges added during the construction of $\mathcal{S}_{\mathcal{C}}(D)$ with the replacement paths using the following policy: Suppose $U \to V$ in $\mathcal{S}_{\mathcal{C}}(D)$ for some $\mathcal{C}$-covered pair U, V. If a directed path is open given $\mathbf{C}$ in D, use that path as the replacement path for the edge $X \to Y$. If not, use any other path. This means that either the path that replaces an edge $U \to V$ is directed or that both the endpoints have an arrowhead and that $U \in An_D(\mathbf{C})$.

Observe that each replacement path is d-connecting and the subpaths of p that remain intact in q must be d-connecting since p is d-connecting. By Lemma C.3, any two paths – whether it is a pair of replacement paths or a replacement path and a subpath of p – have overlapping nodes, then their concatenation must be d-connecting. Since we assumed that q was not d-connecting, it must be that one of the endpoints of one of the added edges must be blocking q. We investigate each such node to verify that q is indeed d-connecting to arrive at a contradiction.

In other words, the collider status of some of these nodes must have changed due to replacing some edges with replacement paths. Specifically due to Lemma C.3, one of the endpoints of replacement paths must be a collider and not an ancestor of $\mathbf{C}$. Since ancestrality status cannot change from D to $\mathcal{S}_{\mathcal{C}}(D)$ due to Lemma C.5, the only way for q to not be d-connecting is if some node that is not an ancestor of $\mathbf{C}$ changes status from being a non-collider along p to being a collider along q.

Note that for an edge $U \leftrightarrow V$, the nodes U and V are adjacent to an arrowhead in the replacement path. Thus, if some node T changes collider status in q compared to p, it cannot be due to bidirected edges along p.

Now consider the directed edges $U \to V$. If the replacement path is directed from U to V, similarly U is adjacent to a tail and V is adjacent to an arrowhead on the replacement path. Therefore, such edges cannot alter the collider status of nodes at the junction of different paths. Finally, consider the directed edges $U \to V$ where U and V are both adjacent to an arrowhead on the replacement path. Observe that this edge cannot change the collider status of V. We now focus on U. If the other edge adjacent to U along q is a tail, U remains a non-collider and cannot block q. Now suppose the other edge adjacent to U along q is an arrowhead. This makes U a collider in q whereas U was a non-collider along p since we had

$U \to V$ along p. However, by construction of q, as we ended up adding a path with arrowheads at both endpoints, it must be that the directed path between U, V (which exists since $U \to V$ was added during the construction of $\mathcal{S}_{\mathcal{C}}(D)$ must be blocked via conditioning. This means $U \in An_D(\mathbf{C})$ which means that although the status of U changes from non-collider to collider, it must be that this collider does not block q since it is an ancestor of $\mathbf{C}$. Therefore, no replacement path can alter the status of a node to block the path q, and q must be d-connecting, which contradicts with the assumption that the path was not d-connecting in D.

Thus, any d-connecting path in $\mathcal{S}_{\mathcal{C}}(D)$ is also d-connecting in D, which establishes that if $(X \not\perp\!\!\!\perp Y|\mathbf{C})_{\mathcal{S}_{\mathcal{C}}(D)}$, then $(X \not\perp\!\!\!\perp Y|\mathbf{C})_D$, which proves the lemma. □

C.2 PROOF OF THEOREM 3.6

We proceed by first proving a corollary that follows from a theorem proven by (Richardson and Spirtes, 2002).

Theorem C.6. *(Richardson and Spirtes, 2002) Two MAGs M_1, M_2 are Markov equivalent if and only if i) They have the same skeleton, ii) They have the same unshielded colliders, iii) For any node Y for which there is a discriminating path p, Y has the same collider status on p in M_1, M_2.*

Corollary C.7. *Two $\mathcal{C}$-closure graphs $\mathcal{S}_1, \mathcal{S}_2$ are Markov equivalent if and only if they have the same skeleton and they have the same unshielded colliders.*

Proof. ($\Rightarrow$)If they are Markov equivalent, then they are two Markov equivalent MAGs by Lemma C.12. Therefore, by Theorem C.6, they have the same skeleton and the same unshielded colliders.

($\Leftarrow$) If they have the same skeleton and the same unshielded colliders, then by Lemma C.18, they must have the same colliders along discriminating paths. Thus, by Theorem C.6, they are equivalent. □

Next, we prove two more lemmas which we will use later in the main proof.

Lemma C.8. *In a $\mathcal{C}$-closure graph $\mathcal{S}_{\mathcal{C}}(D)$, two nodes X, Y are non-adjacent iff $(X \perp\!\!\!\perp Y|\mathbf{C})_{\mathcal{S}_{\mathcal{C}}(D)}$ for some $\mathbf{C} \subset V$.*

Proof. ($\Rightarrow$) Suppose X, Y are non-adjacent in $\mathcal{S}_{\mathcal{C}}(D)$. Thus, it must be the case that $(X \perp\!\!\!\perp Y|\mathbf{C})_D$ and $\mathbf{C} \in \mathcal{C}$, where $\mathcal{C}$ is a conditionally closed set. Otherwise, X, Y would have been made adjacent by the construction of $\mathcal{S}_{\mathcal{C}}(D)$. By Lemma 3.5, this implies $(X \perp\!\!\!\perp Y|\mathbf{C})_{\mathcal{S}_{\mathcal{C}}(D)}$.

($\Leftarrow$) Suppose $(X \perp\!\!\!\perp Y|\mathbf{C})_D$. By definition of d-separation, adjacent nodes cannot be d-separated, and therefore, X, Y must be non-adjacent in $\mathcal{S}_{\mathcal{C}}(D)$. □

Lemma C.9. *In a $\mathcal{C}$-closure graph $\mathcal{S}_{\mathcal{C}}(D)$, any pair of non-adjacent nodes X, Y are separable by some members in the conditionally closed set i.e., $\exists \mathbf{C} \in \mathcal{C}, (X, \perp\!\!\!\perp Y|\mathbf{C})_{\mathcal{S}_{\mathcal{C}}(D)}$.*

Proof. Suppose that there exists some non-adjacent pair X, Y, all the d-separating sets in $\mathcal{S}_{\mathcal{C}}(D)$ are not in $\mathcal{C}$. Let S be the subset that makes X, Y d-separated, i.e., $(X \perp\!\!\!\perp Y|S)_{\mathcal{S}_{\mathcal{C}}(D)}$. S is guaranteed to exist due to Lemma C.8. By definition of $\mathcal{C}$-closure graphs, the non-adjacency of X and Y in $\mathcal{S}_{\mathcal{C}}(D)$ implies X, Y are separable in D given some $\mathbf{C} \in \mathcal{C}$, where $\mathcal{C}$ is a conditionally closed set, i.e., $(X \perp\!\!\!\perp Y|\mathbf{C})_D$. By Lemma 3.5, D and $\mathcal{S}_{\mathcal{C}}(D)$ entail the same d-separation statement restricted to the conditioning sets from a conditionally closed set. Therefore, $(X \perp\!\!\!\perp Y|\mathbf{C})_{\mathcal{S}_{\mathcal{C}}(D)}$. □

Proof of Theorem 3.6

Proof. Suppose D_1, D_2 are $\mathcal{C}$-Markov equivalent. For the sake of contradiction, assume that $\mathcal{S}_{\mathcal{C}}(D_1)$ and $\mathcal{S}_{\mathcal{C}}(D_2)$ are not Markov equivalent. By Corollary C.7, this occurs when either they have different skeletons or different unshielded colliders. Thus, there are two cases: (i) $\mathcal{C}$-closure graphs have different skeletons and (ii) $\mathcal{C}$-closure graphs have different unshielded colliders.

Case (i) $\mathcal{C}$-closure graphs have different skeletons: $\mathcal{S}_{\mathcal{C}}(D_1)$ and $\mathcal{S}_{\mathcal{C}}(D_2)$ have different skeletons. Without loss of generality, suppose that $\mathcal{S}_{\mathcal{C}}(D_1)$ has an extra edge, i.e., X, Y are adjacent in $\mathcal{S}_{\mathcal{C}}(D_1)$ but not in $\mathcal{S}_{\mathcal{C}}(D_2)$. This can only happen if there exists $\mathbf{C} \in \mathcal{C}$ such that $(X \perp\!\!\!\perp Y|\mathbf{C})_{D_2}$, while there is no such separating set in D_1, implying that $(X \not\perp\!\!\!\perp Y|\mathbf{C})_{D_1}$. This is a

contradiction with the supposition that D_1, D_2 are $\mathcal{C}$-Markov equivalent. Thus, $\mathcal{S}_{\mathcal{C}}(D_1)$ and $\mathcal{S}_{\mathcal{C}}(D_2)$ must have the same skeleton.

Case (ii) $\mathcal{C}$-closure graphs have different unshielded colliders: Without loss of generality, assume that there exists an unshielded collider $(X*\!\!\rightarrow Z \leftarrow\!\!* Y)$ in $\mathcal{S}_{\mathcal{C}}(D_1)$ but the unshielded triple $\langle X, Z, Y\rangle$ is a non-collider in $\mathcal{S}_{\mathcal{C}}(D_2)$, where $*$ denotes a wildcard that can be a tail, an arrowhead, or a circle mark.

Since X, Y are non-adjacent in both $\mathcal{S}_{\mathcal{C}}(D_1)$ and $\mathcal{S}_{\mathcal{C}}(D_2)$, by Lemma C.9, there exists two subsets $\mathbf{C}_1, \mathbf{C}_2$ that are members of $\mathcal{C}$ such that

$$(X \perp\!\!\!\perp Y|\mathbf{C}_1)_{\mathcal{S}_{\mathcal{C}}(D_1)}, (X \perp\!\!\!\perp Y|\mathbf{C}_2)_{\mathcal{S}_{\mathcal{C}}(D_2)} \tag{5}$$

. So, we have $Z \in \mathbf{C}_2$ and $Z \notin \mathbf{C}_1$ as Z is a collider between X, Y in $\mathcal{S}_{\mathcal{C}}(D_1)$ and a non-collider in $\mathcal{S}_{\mathcal{C}}(D_2)$. If we switch the conditioning sets, due to different collider status of Z in both graphs, the d-separation statements will switch to d-connection statements:

$$(X \not\perp\!\!\!\perp Y|\mathbf{C}_1)_{\mathcal{S}_{\mathcal{C}}(D_2)}, (X \not\perp\!\!\!\perp Y|\mathbf{C}_2)_{\mathcal{S}_{\mathcal{C}}(D_1)} \tag{6}$$

. Since $\mathbf{C}_1, \mathbf{C}_2$ are in $\mathcal{C}$, by Lemma 3.5, we have that

$$(X \perp\!\!\!\perp Y|\mathbf{C}_1)_{D_1}, (X \not\perp\!\!\!\perp Y|\mathbf{C}_1)_{D_2} \tag{7}$$

. This implies that D_1, D_2 are not $\mathcal{C}$-Markov equivalent, which is a contradiction. Thus, if D_1, D_2 are $\mathcal{C}$-Markov equivalent, then $\mathcal{S}_{\mathcal{C}}(D_1), \mathcal{S}_{\mathcal{C}}(D_2)$ must have the same skeleton and the same unshielded colliders. By Corollary C.7, $\mathcal{S}_{\mathcal{C}}(D_1), \mathcal{S}_{\mathcal{C}}(D_2)$ are Markov equivalent.

($\Leftarrow$) Suppose that $\mathcal{S}_{\mathcal{C}}(D_1)$ and $\mathcal{S}_{\mathcal{C}}(D_2)$ are Markov equivalent. Then, they impose the same d-separation statements. Thus, they impose the same d-separation statements when the conditioning set is restricted to the members of $\mathcal{C}$. By Lemma 3.5, this implies that D_1, D_2 must also impose the same d-separation statements for conditioning sets in $\mathcal{C}$. This establishes that D_1, D_2 are $\mathcal{C}$-Markov equivalent. □

C.3 PROOF OF COROLLARY 3.10

Kocaoglu (2023) shows that the well-celebrated FCI algorithm (Zhang, 2008b) is sound but not complete for learning k-essential graphs. A similar result follows that the FCI algorithm is also sound but not complete for learning $\mathcal{C}$-essential graphs. Corollary 3.10 is a corollary that follows the lemma below, which shows that discriminating paths do not carry extra information about the underlying causal structure. We will first prove the lemma and proceed to the proof of the corollary.

Lemma C.1. *In any $\mathcal{C}$-closure graph, if there is a discriminating path p for $\langle X, Y, Z\rangle$ and $X \leftrightarrow Y \leftarrow\!\!* Z$ is a collider along p, then the orientation $X*\!\!\rightarrow Y$ and $Y \leftarrow\!\!* Z$ can be learned by first finding all unshielded colliders, and then applying the orientation rules $\mathcal{R}_1$ and $\mathcal{R}_2$ of FCI.*

Proof. Suppose in the $\mathcal{C}$-closure graph $\mathcal{S}_{\mathcal{C}}(D)$ for some DAG D, we have a discriminating path p for Y between the nodes A and V of the form

$$A*\!\!\rightarrow\leftrightarrow \ldots \leftrightarrow U \leftrightarrow Y \leftrightarrow V \tag{8}$$

. By definition of the discriminating path, U must be a collider along p, and $U \rightarrow V$. If $Y \leftarrow V$, then we would have an almost directed cycle $U \rightarrow V \rightarrow Y \leftrightarrow U$. Thus, we have $U \leftrightarrow Y \leftrightarrow V$.

First, we will show that the arrowhead at Y of the edge $Y \leftrightarrow V$ can be learned by first orienting unshielded colliders and then applying $\mathcal{R}1$ and $\mathcal{R}2$. Consider the bidirected edge $U \leftrightarrow Y$. By definition of the discriminating path A, V must be non-adjacent. By Lemma C.9, they must be separable by some $\mathbf{C} \in \mathcal{C}$. Thus, we have a set $\mathbf{C} \in \mathcal{C}$ such that $(A \perp\!\!\!\perp V|\mathbf{C})_{\mathcal{S}_{\mathcal{C}}(D)}$. By the definition of the discriminating path, every non-endpoint vertices along p is a parent of V. Therefore, it must be that every collider along p including U must be in $\mathbf{C}$. Otherwise, there would be a d-connecting path from A to V given $\mathbf{C}$. By Lemma C.2, conditioned on any set in $\mathcal{C}$, we have a d-connecting path that starts with an arrowhead at Y. Consider the shortest such path q. Let X be the node immediately before Y along q. Since this path exists in the DAG by lemma, we have $X \rightarrow Y$. If X and V are non-adjacent, then $X \rightarrow Y \leftrightarrow V$ would be an unshielded collider and we are done.

Suppose X and V are adjacent. Note that conditioned on $\mathbf{C}$, A and X are d-connected. If X is a non-collider along the path obtained by concatenating the subpath of q between A, X and the edge between X, V, then A, V would be d-connected given $\mathbf{C}$, which is a contradiction. Thus, X must be a collider along this path. Therefore, we have $X \leftrightarrow V$. Note that we

cannot have $X \leftarrow V$ since this would create an almost directed cycle in $\mathcal{S}_{\mathcal{C}}(D)$. Let J be the node immediately before X along q. Thus, we have $J \rightarrow X \leftrightarrow V$ (not $J \leftrightarrow X$ since this edge exists in D).

Suppose J and V are not adjacent. Then, the collider $J \rightarrow X \leftarrow\!* V$ is unshielded and therefore can be learned. Furthermore, J and Y must be non-adjacent since otherwise, we must have $J \rightarrow Y$ to avoid a cycle and there would be a path that is shorter than q, which "skip over" the node X along q. Thus, we can learn that $X \rightarrow Y$ from $\mathcal{R}1$. Finally, since now we have learned $V\!* \!\rightarrow X \rightarrow Y$ and that Y, V are adjacent. We must have $Y \leftarrow\!* V$ by using $\mathcal{R}2$. Thus, the arrowhead mark at Y of the edge $Y \leftrightarrow V$ can be learned, and we are done.

Suppose J and V are adjacent. Following a similar argument, we either have some unshielded collider that can be propagated using the argument above to orient $V\!* \!\rightarrow Y$, or we can continue covering unshielded colliders, which would imply the previous nodes are always parents along q. But this implies that U has a directed path to Y, which cannot happen since we have $U \leftrightarrow Y$, similarly consider the shortest d-connecting path q between Y and V in D given $\mathbf{C}$ that starts with an arrowhead at Y. The argument follows similarly that either there would be a directed path from V to Y, which is a contradiction with the existence of the edge $Y \leftrightarrow V$ in the $\mathcal{C}$-closure graph, or that there exists an unshielded collider along q that can be learned by orienting unshielded colliders, which be propagated to learn $U\!* \!\rightarrow Y$ using $\mathcal{R}1$ and $\mathcal{R}2$. This proves the lemma. □

Corollary 3.10. $\mathcal{C}$-PC without Step 5 is sound and complete for learning $PAG(\mathcal{S}_{\mathcal{C}}(D))$ of any DAG D.

Proof. By Lemma C.9, any non-adjacent pair are separable by a member in $\mathcal{C}$. Thus, a valid separating set for any non-adjacent pair will be found in Step 1, and will be used to learn the skeleton in Step 2, and to orient all unshielded colliders in Step 3. Zhang (2008a) proved arrowhead and tail completeness of FCI with orientation rules $\mathcal{R}1$ to $\mathcal{R}10$. By Lemma C.1, colliders on discriminating paths will be oriented at the end of Step 4. Thus, $\mathcal{R}4$ that is concerned with discriminating path colliders is not applicable. Similarly, $\mathcal{R}5, \mathcal{R}6, \mathcal{R}7$ are only applicable in graphs with selection bias. Thus, they are not applicable. Since the rules that we omit are never applicable during the execution of the algorithm, Step 4 correctly returns the $PAG(\mathcal{S}_{\mathcal{C}}(D))$ since the algorithm at that point is identical to the FCI algorithm for learning PAGs. □

C.4 PROOF OF THEOREM 3.11

Theorem 3.11 . $\mathcal{C}$-PC algorithm is sound for learning $\mathcal{C}$-essential graph given a conditional independence oracle under the causal Markov and $\mathcal{C}$-faithfulness assumptions, i.e., if $\mathcal{C}$-PC returns M, we have $\varepsilon_{\mathcal{C}}(D) \subseteq M \subseteq PAG(\mathcal{S}_{\mathcal{C}}(D))$

Proof. We will prove soundness of the two orientation rules $\mathcal{R}11, \mathcal{R}12$ with the following lemmas:

Lemma C.10. *Let M be a mixed graph that is sandwiched between $\varepsilon_{\mathcal{C}}(D)$ and $PAG(\mathcal{S}_{\mathcal{C}}(D))$, i.e., $\varepsilon_{\mathcal{C}}(D) \subseteq M \subseteq PAG(\mathcal{S}_{\mathcal{C}}(D))$. $\mathcal{R}11$ is sound on M for learning the $\mathcal{C}$-essential graph, i.e., if $M' = \mathcal{R}11(M)$, then $\varepsilon_{\mathcal{C}}(D) \subseteq M' \subseteq M$.*

Proof. For the sake of contradiction, suppose otherwise that $\mathcal{R}11$ orients an edge $Xo\!\rightarrow Y$ in M as $X \rightarrow Y$, and there is a DAG D' with a $\mathcal{C}$-closure graph $\mathcal{S}_{\mathcal{C}}(D')$ that is Markov equivalent to $\mathcal{S}_{\mathcal{C}}(D)$ and is consistent with M where $X \leftrightarrow Y$. This means X and Y are $\mathcal{C}$-covered in D'. Then, by Lemma C.1, conditioned on any member in $\mathcal{C}$, there must be a d-connecting path that starts with an arrowhead both at X and at Y in D. By the construction of the $\mathcal{C}$-closure graph, this path must also exist in $\mathcal{S}_{\mathcal{C}}(D')$. Therefore, there must be some nodes W such that $X \leftarrow W$. Since X has no incoming edges, it must be the case that $W \in \mathcal{C}$. However, Y is chosen so that Y is non-adjacent to any node in $\mathcal{C}$. Therefore, Y must be non-adjacent to W in M. However, this creates the unshielded collider $W \rightarrow X \leftrightarrow Y$. Note that $Wo\!—\!oXo\!\rightarrow Y$ in $PAG(\mathcal{S}_{\mathcal{C}}(D))$. Hence, $\langle W, X, Y\rangle$ is a non-collider in $\mathcal{S}_{\mathcal{C}}(D)$. Therefore, $\mathcal{S}_{\mathcal{C}}(D')$ cannot be Markov equivalent to $\mathcal{S}_{\mathcal{C}}(D)$, which is a contradiction. □

Lemma C.11. *Let M be a mixed graph that is sandwiched between $\varepsilon_{\mathcal{C}}(D)$ and $PAG(\mathcal{S}_{\mathcal{C}}(D))$, i.e., $\varepsilon_{\mathcal{C}}(D) \subseteq M \subseteq PAG(\mathcal{S}_{\mathcal{C}}(D))$. $\mathcal{R}12$ is sound on M for learning the $\mathcal{C}$-essential graph, i.e., if $M' = \mathcal{R}12(M)$, then $\varepsilon_{\mathcal{C}}(D) \subseteq M' \subseteq M$.*

Proof. For the sake of contradiction, suppose otherwise that $\mathcal{R}12$ orients an edge $Xo\!—\!oZ$ in M as $X - Z$, and there is a DAG D' with a $\mathcal{C}$-closure graph $\mathcal{S}_{\mathcal{C}}(D')$ that is Markov equivalent to $\mathcal{S}_{\mathcal{C}}(D)$ and is consistent with M where $X \leftrightarrow Z$. This means X, Z are $\mathcal{C}$-covered in D'. Then, by Lemma C.1, conditioned on any member in $\mathcal{C}$, there must be a d-connecting path that starts with an arrowhead both at X and Z in D. By construction of the $\mathcal{C}$-closure graph, this path must also be in $\mathcal{S}_{\mathcal{C}}(D')$. Therefore, there must be some node W such that $X \leftarrow W$. Since X has no incoming edges, it must be the case that $X \in \mathcal{C}$. However, Z is chosen so that Z is non-adjacent to any other node in $\mathcal{C}$. Therefore, Z must be non-adjacent to W in

M. However, note that Wo—oXo—oZ in $PAG(\mathcal{S}_{\mathcal{C}}(D))$ and therefore, $\langle W, X, Z\rangle$ is a non-collider in $\mathcal{S}_{\mathcal{C}}(D)$. Therefore, $\mathcal{S}_{\mathcal{C}}(D')$ cannot be Markov equivalent to $\mathcal{S}_{\mathcal{C}}(D)$, which is a contradiction. □

Now, consider the execution of the algorithm $\mathcal{C}$-PC. When the algorithm completes Step 4, from Corollary 3.10, we have that $M = PAG(\mathcal{S}_{\mathcal{C}}(D))$. Since we start Step 5 with $M = PAG(\mathcal{S}_{\mathcal{C}}(D))$, from Lemma C.10 and C.11, any arrowhead and tail orientation of the M obtained at the end of Step 5 must be consistent with the $\mathcal{C}$-essential graph of D. Therefore, we have that $\varepsilon_{\mathcal{C}}(D) \subseteq M$. □

We present more results below that are not explicitly described in the main paper.

C.5 PROOF OF LEMMA C.12

Lemma C.12. *For any DAG D, the $\mathcal{C}$-closure graph $\mathcal{S}_{\mathcal{C}}(D)$ is a maximal ancestral graph (MAG).*

For a mixed graph to be a maximal ancestral graph, we need to show that it does not have directed or almost directed cycles and that any non-adjacent pair of nodes can be made conditionally independent by conditioning on some subset of observed variables (Zhang, 2008a). We proceed by proving the following lemma.

Lemma C.13. *For any DAG D and a set $\mathcal{C}$, $\mathcal{S}_{\mathcal{C}}(D)$ does not have directed or almost directed cycles.*

Proof. Suppose, for the sake of contradiction that there is a directed cycle in $\mathcal{S}_{\mathcal{C}}(D)$. Since each edge $X \to Y$ in $\mathcal{S}_{\mathcal{C}}(D)$ either exists in D or for each such edge in $\mathcal{S}_{\mathcal{C}}(D)$, there is a directed path from X to Y in D, the existence of a directed cycle in $\mathcal{S}_{\mathcal{C}}(D)$ would imply a directed cycle in D, which contradicts with the DAG assumption of D.

Suppose, for the sake of contradiction that there is an almost directed cycle in $\mathcal{S}_{\mathcal{C}}(D)$, i.e., we have a directed path from A to B for two nodes $A \leftrightarrow B$. Since $A \leftrightarrow B$ is added during the construction of $\mathcal{S}_{\mathcal{C}}(D)$, it must be the case that neither A nor B are ancestors of each other. However, from the above argument, there must be a directed path from A to B in D, which is a contradiction. Thus, $\mathcal{S}_{\mathcal{C}}(D)$ cannot have almost directed cycles.

The other condition for a mixed graph to be a maximal ancestral graph is that for any non-adjacent pair of nodes, there exists a subset of the observed variables that make them conditionally independent. For the $\mathcal{C}$-closure graphs, this simply follows by construction: Any pair of nodes that are non-adjacent in $\mathcal{S}_{\mathcal{C}}(D)$ can be made conditionally independent given some members in the conditionally closed set, i.e., $\mathbf{C} \in \mathcal{C}$ in D by construction of $\mathcal{S}_{\mathcal{C}}(D)$. From Lemma 3.5, this conditional independence relation must be retained in $\mathcal{S}_{\mathcal{C}}(D)$. Thus any non-adjacent pair of nodes in $\mathcal{S}_{\mathcal{C}}(D)$ can be d-separated in $\mathcal{S}_{\mathcal{C}}(D)$ by some members in a conditionally closed set. This establishes the claim. □

C.6 PROOF OF THEOREM C.16

The following lemmas show that for any mixed graph M that satisfies the constraints in Theorem C.16, i.e., those that are MAGs and that satisfy the condition that for any bidirected edge $X \leftrightarrow Y$, X, Y are $\mathcal{C}$-covered in the graph $M - \{X \leftrightarrow Y\}$.

Lemma C.14. *Consider a bidirected edge $X \leftrightarrow Y$ in a mixed graph M. Suppose conditioned on any member $\mathbf{C}$ of $\mathcal{C}$, $(X \not\perp\!\!\!\perp Y|\mathbf{C})_{M-(X\leftrightarrow Y)}$. Then conditioned on any $\mathbf{C} \in \mathcal{C}$, there exists a d-connecting path between X and Y that starts with an arrow into X and an arrow into Y.*

Proof. For the sake of contradiction suppose, conditioned on some $\mathbf{C} \in \mathcal{C}$, there is no d-connecting path with an arrow into X and an arrow into Y. Since neither X is an ancestor of Y nor Y is an ancestor of X and assuming there is no d-connecting path with an arrow into X and an arrow into Y, it must be that all d-connecting paths have colliders on them. All such colliders must be ancestors of $\mathbf{C}$.

Consider such a path p where the edge adjacent to X has a tail at X. Let K be the collider that is closest to X.

Thus, we have

$$X \to U_1 \to \ldots \to U_m \to K \leftarrow * V \ldots Y \tag{9}$$

for some $\{U_i\}_{i\in m}, V$. However, this is a contradiction since we have $(X \not\perp\!\!\!\perp Y)_{M-(X\leftrightarrow Y)}$, which is impossible. Due to symmetry between X and Y, the supposition that the only d-connecting paths must have a tail adjacent to either endpoint must be wrong. This proves the lemma. □

Lemma C.15. *Let M be a mixed graph that satisfies the conditions in Theorem C.16. Let M' be the graph obtained by removing all the bidirected edges from M. Then,*

1. *M' is a DAG and*
2. *$M' \sim_{\mathcal{C}} M$*

Proof. Since the only difference between M' and M is the removal of bidirected edges, any directed cycle that exists in M' would also have existed in M, which contradicts with the assumption that M is a MAG. This establishes that M has no directed cycles.

Clearly, any independence statement in M holds in M', since it is obtained from M by removing edges. Thus any d-separation relation restricted to $\mathcal{C}$ that holds in M also holds in M'. Therefore, we only need to show that for any $\mathbf{C} \in \mathcal{C}$, $(A \not\perp\!\!\!\perp B|\mathbf{C})_M$ implies $(A \not\perp\!\!\!\perp B|\mathbf{C})_{M'}$.

Suppose for the sake of contradiction that $(A \not\perp\!\!\!\perp B|\mathbf{C})_M$ but $(A \perp\!\!\!\perp B|\mathbf{C})_{M'}$. Let p be a d-connecting path between A, B given $\mathbf{C}$ in M. This path must be closed in M'. Since the only difference between the two graphs is the removal of bidirected edges, ancestrality relations cannot be different. Thus, it cannot be the case that a collider that was open in M is now closed in M' and is closing the path p. Any collider that was open must still be open. Thus, the only way for p to be closed in M' is if some bidirected edge $X \leftrightarrow Y$ along p is removed. However, by Lemma C.14, for any such bidirected edge in M, and for any conditioning set $\mathbf{C} \in \mathcal{C}$, we have a d-connecting path called a replacement path with an incoming edge to both X and Y. Consider the path q obtained by replacing every bidirected edge along p with a corresponding replacement path. Since A, B are d-separated by assumption, this path cannot be open. As this path is a concatenation of several d-connecting paths – either sub-paths of p, which must be open, or replacement paths which must be open, by Lemma C.3, they must have no overlapping nodes, and some node at the junction of these paths must be a collider and non-ancestor of $\mathbf{C}$. However, since we replaced bidirected edges $X \leftrightarrow Y$ with paths of the form $X \leftarrow * \ldots * \rightarrow Y$, both X and Y must have the same collider status on both p and q. Thus, they cannot be blocking q since they are not blocking p. This means that q is d-connecting in M', which is a contradiction. This proves the lemma that M and M' must entail the same d-separation relations restricted to $\mathcal{C}$, which implies they are $\mathcal{C}$-Markov equivalent.

□

Theorem C.16. *A mixed graph $M = (V, E)$ is a $\mathcal{C}$-closure graph if and only if it is a maximal ancestral graph and for any bidirected edge $X \leftrightarrow Y \in E$ the following is true:*

- *$\nexists \mathbf{C} \in \mathcal{C}, (X \perp\!\!\!\perp Y|\mathbf{C})_{M'}$, where $M' = (V, E - \{X \leftrightarrow Y\})$*

Proof. The only if direction: Suppose a mixed graph is a $\mathcal{C}$-closure graph, i.e. $M = \mathcal{S}_{\mathcal{C}}(D)$ for some DAG D and has the edge $A \leftrightarrow B$. Suppose for the sake of contradiction that A, B are not $\mathcal{C}$-covered in $M - (A \leftrightarrow B)$. Let M' be the graph obtained from M by removing all the bidirected edges. Note that M' is a DAG since M has no directed cycles. Also, note that all edges in D must appear in M' by construction of $\mathcal{C}$-closure graphs. D can therefore be obtained from M by removing edges. Thus, any d-separation statement in M must also hold in D. Therefore, A, B must be conditionally independent given some $\mathbf{C} \in \mathcal{C}$ in D. This means A, B are adjacent in M and M cannot be the $\mathcal{C}$-closure graph of D, which is a contradiction.

If direction: Suppose a mixed graph M satisfies the conditions in Theorem C.16. By Lemma C.15, for any such mixed graph M, there is a DAG whose $\mathcal{C}$-closure is M, which shows that any such M is a valid $\mathcal{C}$-closure graph, proving the theorem.

□

Definition C.17. For two partial mixed graphs M_1, M_2 with the same skeleton, M_1 is said to be a subset of M_2 iff the following conditions hold for any pair X, Y

1. $(Y * — X)_{M_2} \Rightarrow (Y * — X)_{M_1}$, 2. $(Y * \rightarrow X)_{M_2} \Rightarrow (Y * \rightarrow X)_{M_1}$

, where $*$ denotes a wildcard that can be a tail, an arrowhead, or a circle mark.

Lemma C.2. $\varepsilon_{\mathcal{C}}(D) \subseteq PAG(\mathcal{S}_{\mathcal{C}}(D))$

Proof. By Theorem C.16, every $\mathcal{C}$-closure graph is a MAG whereas every MAG is not a $\mathcal{C}$-closure graph. Hence, the set of Markov equivalent $\mathcal{C}$-closure graphs form a subset of the set of Markov equivalent MAGs. Therefore, the tails and arrowheads that appear in all Markov equivalent MAGs must also appear in all the Markov equivalent $\mathcal{C}$-closure graphs. The result follows from Definition C.17. □

C.7 PROOF OF LEMMA C.18

Lemma C.18. *Suppose two $\mathcal{C}$-closure graphs $\mathcal{S}_1, \mathcal{S}_2$ have the same skeleton and unshielded colliders. Then, along any discriminating paths p for a node Y, Y has the same collider status in $\mathcal{S}_1$ and $\mathcal{S}_2$*

Proof. Let $\mathcal{S}_1 = \mathcal{S}_{\mathcal{C}}(D_1), \mathcal{S}_2 = \mathcal{S}_{\mathcal{C}}(D_2)$ be two $\mathcal{C}$-closure graphs with the same skeleton and unshielded colliders. Suppose for the sake of contradiction that there is a path p that is discriminating for a triple $\langle X, Y, Z\rangle$ in both such that Y is a collider along p in $\mathcal{S}_1$ and a non-collider in $\mathcal{S}_2$. Thus, in $\mathcal{S}_1$, we have the path p as

$$A{*\!\!\rightarrow}\, Q_1 \leftrightarrow Q_2 \ldots \leftrightarrow Q_m \leftrightarrow X \leftrightarrow Y \leftrightarrow Z \tag{10}$$

where $Q_i \rightarrow Z$ for all i and $X \rightarrow Z$ and A and Z are not adjacent. Note that we cannot have $Y \leftrightarrow Z$ instead of $Y \leftrightarrow Z$ since this would create the almost directed cycle $X \rightarrow Z \rightarrow Y \leftrightarrow X$. The same path with Y as a non-collider can take two configurations in $\mathcal{S}_2$, either as

$$A{*\!\!\rightarrow}\, Q_1 \leftrightarrow Q_2 \ldots \leftrightarrow Q_m \leftrightarrow X \leftrightarrow Y \rightarrow Z \tag{11}$$

or as

$$A{*\!\!\rightarrow}\, Q_1 \leftrightarrow Q_2 \ldots \leftrightarrow Q_m \leftrightarrow X \leftarrow Y \rightarrow Z \tag{12}$$

. Other paths where Y is a non-collider would either render X a non-collider, which cannot happen by the definition of a discriminating path, or create a directed or almost directed cycle. Since A, Z are not adjacent by the definition of a discriminating path, there must be some $\mathbf{C} \in \mathcal{C}$ where $(A \perp\!\!\!\perp Z|\mathbf{C})_{\mathcal{S}_1}$. Note that $\mathbf{C}$ must include Q_i's and X and not include Y since otherwise there would be d-connecting paths between A and Z in $\mathcal{S}_1$ due to the discriminating path. This implies that $(A \not\perp\!\!\!\perp Z|\mathbf{C})_{\mathcal{S}_2}$

Since $X \leftrightarrow Y$ is in $\mathcal{S}_1$, by Lemma C.1, there must be a d-connecting path between X and Z in D_1 conditioned on $\mathbf{C}$ that has an arrowhead at Y. By construction, this path must also appear in $\mathcal{S}_1$. Since the path is inherited from D_1, it does not have bidirected edges. Consider the shortest of all such d-connecting paths, call this path q. Let X be the node adjacent to Y along q. Thus, q has the form

$$X \leftrightarrow \ldots \rightarrow R \rightarrow Y. \tag{13}$$

We have that $R \rightarrow Y$ in both D_1 and $\mathcal{S}_1$. In $\mathcal{S}_1$, we have $R \rightarrow Y \leftrightarrow Z$. Since the edge between Y, Z has a tail at Y in $\mathcal{S}_2$, this collider cannot exist in $\mathcal{S}_2$. Thus, it must be the case that this collider is shielded in $\mathcal{S}_1$, i.e., R and Z are adjacent in $\mathcal{S}_1$. Since $\mathcal{S}_1, \mathcal{S}_2$ have the same skeleton, they must also be adjacent in $\mathcal{S}_2$.

Now consider the path obtained by concatenating the subpath of p $A{*\!\!\rightarrow}\, \ldots X$, and the subpath of q between X and R and the edge between R and Z in $\mathcal{S}_1$. We call this path r. Note that the subpath of q is d-connecting given $\mathbf{C}$, as well as the subpath of p since Q_i's and X are in $\mathbf{C}$. Thus, unless R is a collider on it, the path r between A and Z will be open, which would lead to a contradiction since A and Z are d-separated given $\mathbf{C}$ in $\mathcal{S}_1$. Thus, the edge between R and Z must have an arrowhead at R. Let W be the node before R along q. Thus, we have $W \rightarrow R \leftrightarrow Z$ in $\mathcal{S}_1$. Note that $R \leftarrow Z$ is not possible since this would create an almost directed cycle $R \rightarrow Y \leftrightarrow Z \rightarrow R$ in $\mathcal{S}_1$.

Suppose this collider is unshielded and appears in $\mathcal{S}_2$ as well: $W{*\!\!\rightarrow}\, R {\leftarrow\!\!*} Z$ in $\mathcal{S}_2$. Thus, in $\mathcal{S}_2$, we have $Y \rightarrow Z{*\!\!\rightarrow}\, R \leftarrow W$. Since R, Y are adjacent, it must be that $R \leftarrow Y$ or $R \leftrightarrow Y$ to avoid a directed or almost directed cycle. Thus, in $\mathcal{S}_2$, we have $R {\leftarrow\!\!*} Y$. However, this creates the collider $W{*\!\!\rightarrow}\, R \leftarrow *Y$ in $\mathcal{S}_2$. Note that this collider must be shielded, meaning that W, Y must be adjacent and both in $\mathcal{S}_2$ and $\mathcal{S}_1$. Since we have $W \rightarrow R \rightarrow Y$ in $\mathcal{S}_1$, the edge must be $W \rightarrow Y$ in $\mathcal{S}_1$. Furthermore, similar to R, W cannot be in the conditioning set since this would block the path q. This means there is a d-connecting path that has an arrowhead at Y that is shorter than q, which is a contradiction.

Thus, the collider $W \rightarrow R \leftrightarrow Z$ in $\mathcal{S}_1$ must be shielded. Similar to the above argument, W must be a collider along the path constructed by concatenating the subpath $A{*\!\!\rightarrow}\, \ldots X$ of p, and the subpath of q between X and W, and the edge between W and Z since otherwise this path would be open, which would contradict with $(A \perp\!\!\!\perp Z|\mathbf{C})_{\mathcal{S}_1}$. Let T be the node next to W along q. Thus, we have $T \rightarrow W \rightarrow R$ along q and $T \rightarrow W {\leftarrow\!\!*} Z$ is a collider in $\mathcal{S}_1$. In fact, it must be that $W \leftrightarrow Z$ since otherwise there would be an almost directed cycle $Z \rightarrow W \rightarrow R \rightarrow Y \leftrightarrow Z$ in $\mathcal{S}_1$.

Suppose the collider $T \to W \leftrightarrow Z$ in $\mathcal{S}_1$ is unshielded and also appears in $\mathcal{S}_2$. Note that if T, R were adjacent in $\mathcal{S}_1$, the orientation would have to be $T{*\!\!\to} R$ since otherwise there would be a directed cycle $T \to W \to R \to T$ in $\mathcal{S}_1$. But this would imply that there is a path shorter than q that connects X, Y and has an arrow into Y. Thus, T, R must be non-adjacent in $\mathcal{S}_1$ and hence in $\mathcal{S}_2$. Thus, $\langle T, W, R\rangle$ is an unshielded non-collider in $\mathcal{S}_1$ and must also be in $\mathcal{S}_2$. Thus, it must be that $W \to R$ in $\mathcal{S}_2$. Since $Z{*\!\!\to} W \to R$ is in $\mathcal{S}_2$, it must be that $Z{*\!\!\to} R$ in $\mathcal{S}_2$ to avoid a directed or almost directed cycle. Since $Y \to Z{*\!\!\to} R$ is in $\mathcal{S}_2$, it must be that $R {\leftarrow\!\!*} Y$ to avoid a cycle or almost directed cycle in $\mathcal{S}_2$. However, now we have a collider $W \to R {\leftarrow\!\!*} Y$ in $\mathcal{S}_2$ that is a non-collider in $\mathcal{S}_1$ since in $\mathcal{S}_1$ we have $R \to Y$. Thus, this collider must be shielded, i.e., W, Y must be adjacent in $\mathcal{S}_2$. Thus, they must also be adjacent in $\mathcal{S}_1$. Since $W \to R \to Y$ in $\mathcal{S}_1$, it must be that $W \to Y$ is in $\mathcal{S}_1$ to avoid a cycle. But this implies there is a shorter d-connecting path between X, Y given $\mathbf{C}$ with an arrowhead at Y, which is a contradiction.

Therefore, the collider $T \to W \leftrightarrow Z$ must be shielded in $\mathcal{S}_1$. We can repeat the above argument as many times as needed continuing from the parent of T along q. As we keep shielding more and more colliders in $\mathcal{S}_1$, eventually when we shield the first node along q next to X, we will end up with a directed path from X to Y. However, this is a contradiction since bidirected edge was added between X, Y which implies that X is not an ancestor of Y.

Therefore, if two $\mathcal{C}$-closure graphs $\mathcal{S}_1, \mathcal{S}_2$ have the same skeleton and unshielded colliders, then they cannot have different colliders along discriminating paths, which proves the lemma. □

D RELATIONSHIPS BETWEEN $\mathcal{C}$-FAITHFULNESS, ADJACENCY FAITHFULNESS AND ORIENTATION FAITHFULNESS

One can view our method as a conservative approach to constraint-based causal discovery. Ramsey et al. (2006) has proposed a sound algorithm under orientation unfaithfulness with the assumption of adjacency faithfulness. We provide one example to show when adjacency faithfulness is a weaker assumption and another example to show when $\mathcal{C}$-faithfulness is weaker than adjacency faithfulness and orientation faithfulness.

Example 1. This example shows that adjacency faithfulness can be weaker than $\mathcal{C}$-faithfulness. Consider $A \to B \to C$, where $(A \perp\!\!\!\perp C)_P$ and $(A \perp\!\!\!\perp C|B)_P$. Suppose the conditionally closed set $\mathcal{C}$ for $\mathcal{C}$-PC is $\{\emptyset\}$, we see that $(A \perp\!\!\!\perp C)_P$ violates $\mathcal{C}$-faithfulness., but adjacency faithfulness is not violated.

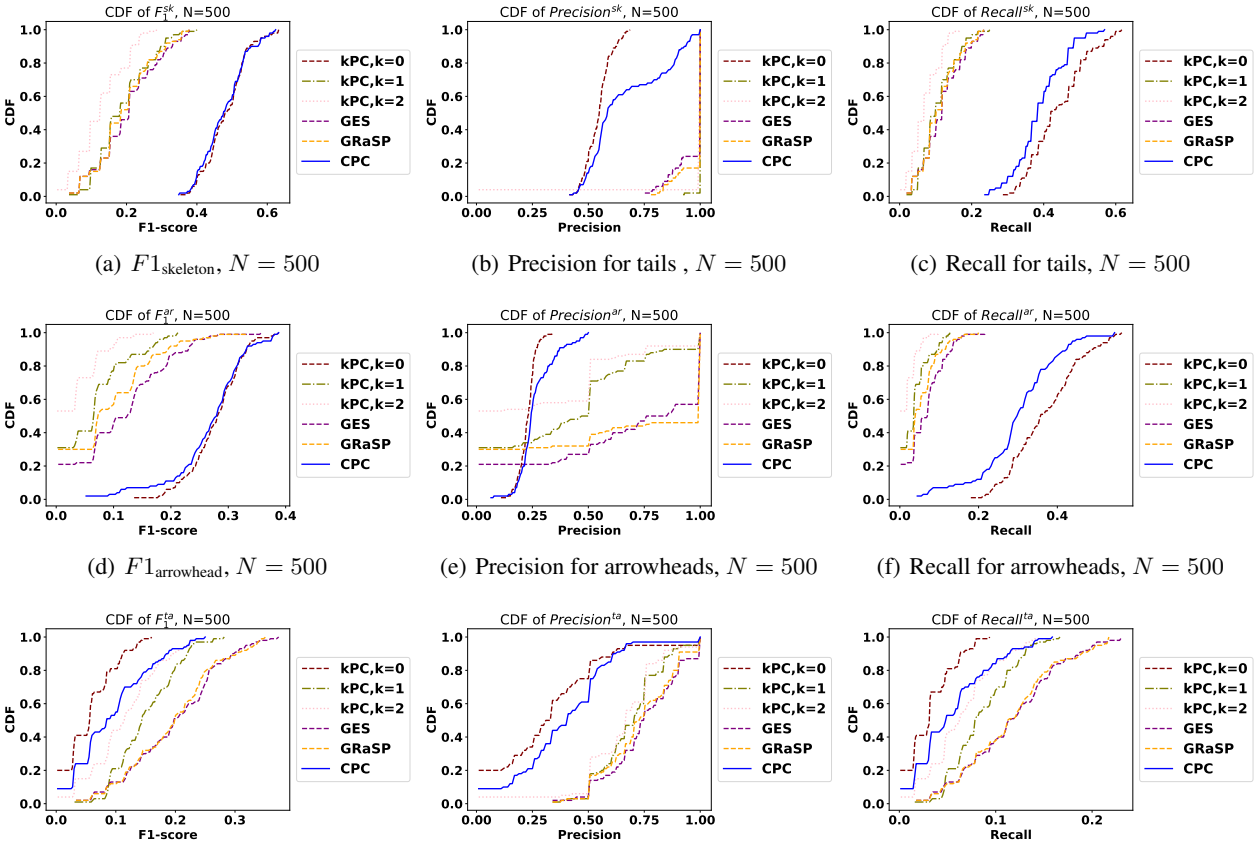
Example 2. This example shows that $\mathcal{C}$-faithfulness can be weaker than adjacency faithfulness and orientation faithfulness. Consider $A \to B \leftarrow C$ with $(A \perp\!\!\!\perp C)_P, (A \perp\!\!\!\perp C|B)_P, (B \perp\!\!\!\perp C|A)_P$. Suppose the conditionally closed set for $\mathcal{C}$-PC is $\{\emptyset\}$, we see that $\mathcal{C}$-faithfulness is not violated while adjacency faithfulness is violated by $(B \perp\!\!\!\perp C|A)_P$ and orientation faithfulness is violated by $(A \perp\!\!\!\perp C)_P$ and $(A \perp\!\!\!\perp C|B)_P$.

E ADDITIONAL EXPERIMENT RESULTS FOR THE EXPERIMENT IN SECTION 4.1

E.1 PRECISION AND RECALL BREAKDOWN

We supplement the experimental result in 4.1 with the corresponding precision and recall for the case where the size of the DAG is 30 with sample sizes of 500 and 2000. The result is shown in Figures 6-7. Based on the precision curve in Figures 6-7, we can see that $\mathcal{C}$-PC indeed exhibits a higher false positive rate across all three metrics: skeleton, tail and arrowheads compared to other baselines, except for k-PC ($k = 0$). However, in terms of arrowheads and skeletons, $\mathcal{C}$-PC enjoys a higher recall rate, indicating that $\mathcal{C}$-PC has fewer false negatives.

We also provide results for the case where the size of the DAG is 20. The result shown in Figure 8 is similar to the case with 30 nodes and 60 edge cases for 500 samples shown in Figure 6, but with a slightly lower score. This is expected as the graph gets denser, and the sample efficiency for CI tests also decreases. While $\mathcal{C}$-PC enjoys similar performance with k-PC ($k = 0$), the other baselines significantly again suffer from low recall rates in terms of arrowheads and skeletons.

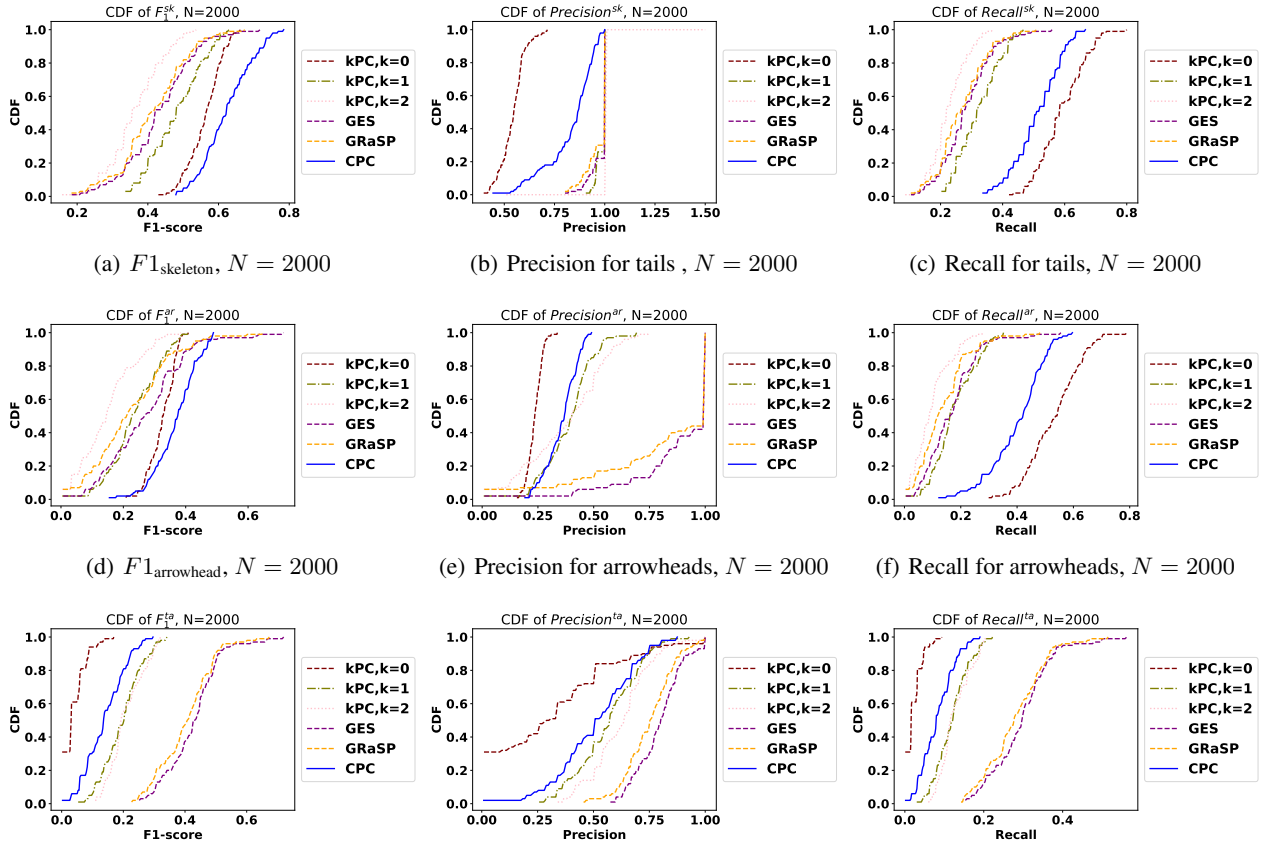


(a) $F1_{\text{skeleton}}$, $N = 500$ (b) Precision for tails , $N = 500$ (c) Recall for tails, $N = 500$

(d) $F1_{\text{arrowhead}}$, $N = 500$ (e) Precision for arrowheads, $N = 500$ (f) Recall for arrowheads, $N = 500$

(g) $F1_{\text{tail}}$, $N = 500$ (h) Precision for tails, $N = 500$ (i) Recall for arrowheads, $N = 500$

Figure 6: Empirical cumulative distribution function of various F1, precision, and recall scores on 100 random DAGs with 30 nodes and 60 edges where the number of states is assigned to be 2 or 30 with probability 0.7 and 0.3 respectively. The lower and farther from the left side the better. **Top**: Skeleton with f1, precision, and recall scores for sample size of $N = 500$. **Middle**: Arrowheads with f1, precision, and recall scores for sample size of $N = 500$. **Bottom**: Tails with f1, precision, and recall scores for sample size of $N = 500$.

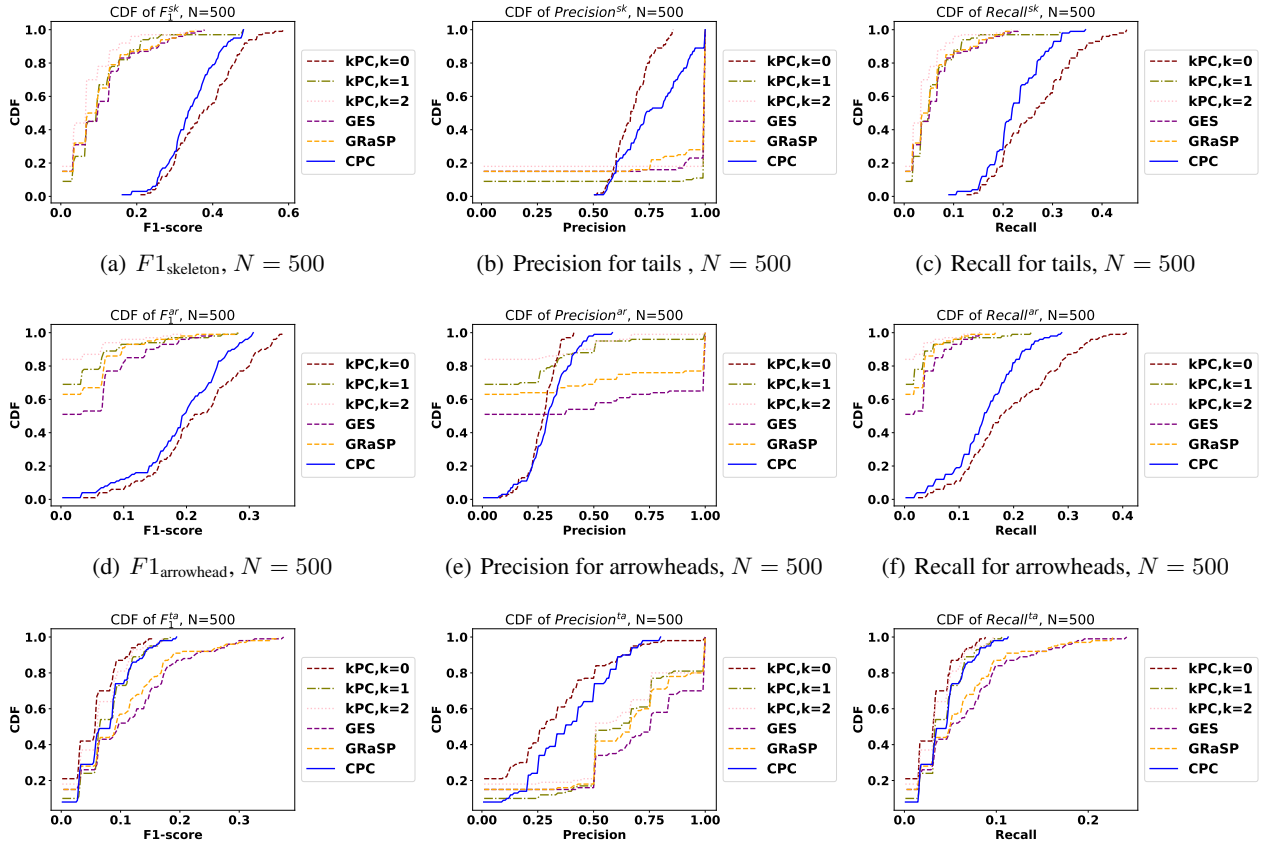


(a) $F1_{\text{skeleton}}$, $N = 2000$ (b) Precision for tails , $N = 2000$ (c) Recall for tails, $N = 2000$

(d) $F1_{\text{arrowhead}}$, $N = 2000$ (e) Precision for arrowheads, $N = 2000$ (f) Recall for arrowheads, $N = 2000$

(g) $F1_{\text{tail}}$, $N = 2000$ (h) Precision for tails, $N = 2000$ (i) Recall for arrowheads, $N = 500$

Figure 7: Empirical cumulative distribution function of various F1, precision, and recall scores on 100 random DAGs with 30 nodes and 60 edges where the number of states is assigned to be 2 or 30 with probability 0.7 and 0.3 respectively. The lower and farther from the left side the better. **Top**: Skeleton with f1, precision, and recall scores for sample size of $N = 2000$. **Middle**: Arrowheads with f1, precision, and recall scores for sample size of $N = 2000$. **Bottom**: Tails with f1, precision, and recall scores for sample size of $N = 2000$.



(a) $F1_{\text{skeleton}}$, $N = 500$ (b) Precision for tails , $N = 500$ (c) Recall for tails, $N = 500$

(d) $F1_{\text{arrowhead}}$, $N = 500$ (e) Precision for arrowheads, $N = 500$ (f) Recall for arrowheads, $N = 500$

(g) $F1_{\text{tail}}$, $N = 500$ (h) Precision for tails, $N = 500$ (i) Recall for arrowheads, $N = 500$

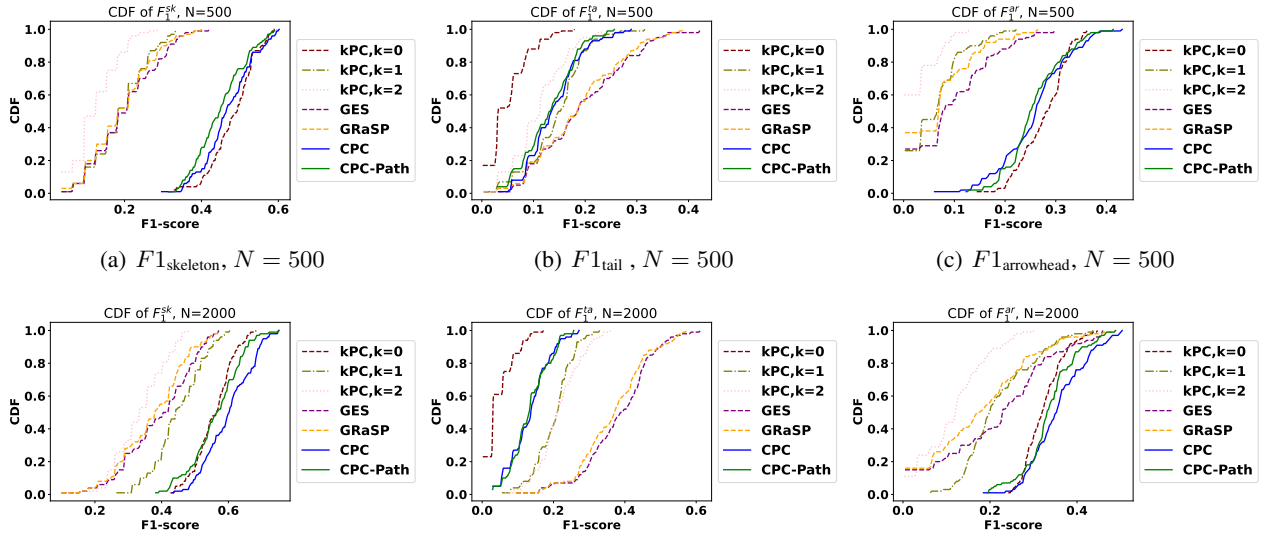
Figure 8: Empirical cumulative distribution function of various F1, precision, and recall scores on 100 random DAGs with 20 nodes with 60 edges where the number of states is assigned to be 2 or 30 with probability 0.7 and 0.3 respectively. The lower and farther from the left side the better. **Top**: Skeleton with f1, precision, and recall scores for sample size of $N = 500$. **Middle**: Arrowheads with f1, precision, and recall scores for sample size of $N = 500$. **Bottom**: Tails with f1, precision, and recall scores for sample size of $N = 500$.

E.2 EXTENDED EXPERIMENT WITH $\mathcal{C}$-PC-PATH AND TWO SAMPLE T-TESTS RESULTS ON EXPERIMENT IN SECTION 4.1

Sample Size	**500**	**2000**
$\mathcal{C}$-PC	0.464 ± 0.004	$\mathbf{0.604 \pm 0.004}$
$\mathcal{C}$PC-Path	0.450 ± 0.004	$\mathbf{0.566 \pm 0.004}$
k-PC, $(k = 0)$	$\mathbf{0.476 \pm 0.004}$	0.562 ± 0.003
k-PC, $(k = 1)$	0.180 ± 0.004	0.458 ± 0.005
k-PC, $(k = 2)$	0.113 ± 0.003	0.341 ± 0.005
GES	0.194 ± 0.005	0.399 ± 0.007
GRaSP	0.185 ± 0.005	0.381 ± 0.007

Table 1: Average F1-skeleton scores with standard errors rounded up to 3 decimal places

Next, we provide the experimental results in section 4.1 with $\mathcal{C}$-PC-Path, which is a variant of $\mathcal{C}$-PC and it only conducts the CI tests for a pair of variables X, Y with conditioning sets $\mathbf{Z}$ in $\mathcal{C}$ when $\mathbf{Z}$ has members that are along with the path between X and Y in the learned skeleton. The results are shown in Figure 9. We see that choosing only the conditioning sets that have at least one member being along with a path between X and Y when testing conditional independence relation does not necessarily improve the performance. Combining the heuristic of defining $\mathcal{C}$ (see Algorithm 2) with $\mathcal{C}$-PC alone has significantly improved k-PC performance where $k = 1$. We also present the two-sample t-test results in Tables 1, 2, 3, 4 for



(a) $F1_{\text{skeleton}}$, $N = 500$ (b) $F1_{\text{tail}}$, $N = 500$ (c) $F1_{\text{arrowhead}}$, $N = 500$

(d) $F1_{\text{skeleton}}$, $N = 2000$ (e) $F1_{\text{tail}}$, $N = 2000$ (f) $F1_{\text{arrowhead}}$, $N = 2000$

Figure 9: Empirical cumulative distribution function of various F1 scores on 100 random DAGs with 30 nodes where the number of states is assigned to be 2 or 30 with probability 0.7 and 0.3 respectively. The lower and farther from the left side the better. This is the same experiment conducted in section 4.1 with an additional baseline named $\mathcal{C}$-PC-Path. **Top**: Skeleton F1 scores $F1_{\text{skeleton}}$, tail F1 scores $F1_{\text{tail}}$ and arrowhead F1 scores $F1_{\text{tail}}$ for sample size of $N = 500$. **Bottom**: Skeleton F1 scores $F1_{\text{skeleton}}$, tail F1 scores $F1_{\text{tail}}$ and arrowhead F1 scores $F1_{\text{tail}}$ for sample size of $N = 2000$. Overall, we see that $\mathcal{C}$-PC (blue) outperforms other baselines at $N = 2000$ in terms of $F1_{\text{skeleton}}$, $F1_{\text{arrowhead}}$ with a small expense in $F1_{\text{tail}}$. When $N = 500$, $\mathcal{C}$-PC performs comparably with k-PC ($k = 0$) as expected as $\mathcal{C}$-PC also conducts all marginal independence tests.

Sample Size	**500**	**2000**
$\mathcal{C}$PC-Path	**(0.017, -2.128)**	**(2.982e-10, -6.295)**
k-PC, $(k = 0)$	(0.982, 2.092)	**(2.590e-15, -8.030)**
k-PC, $(k = 1)$	**(1.010e-198, -45.965)**	**(6.265e-80, -22.111)**
k-PC, $(k = 2)$	**(1.960e-263, -62.154)**	**(7.521e-175, -40.716)**
GES	**(1.639e-164, -38.539)**	**(4.870e-98, -25.520)**
GRaSP	**(2.080e-175, -40.835)**	**(7.346e-114, -28.511)**

Table 2: One-sided two sample t-test result with format (p-value, t-value) under null hypothesis $\mathcal{H}_0 : \mu_{baseline} \geq \mu_{\mathcal{C}PC}$, where $\mu_{baseline}$ is the average F1-skeleton scores of the baseline and $\mu_{\mathcal{C}PC}$ is the average F1 skeleton scores of $\mathcal{C}$-PC. The degree of freedom is 598.

Sample Size	**500**	**2000**
$\mathcal{C}$PC	0.244 ± 0.004	$\mathbf{0.352 \pm 0.004}$
$\mathcal{C}$PC-Path	0.249 ± 0.003	0.335 ± 0.004
k-PC, $(k = 0)$	$\mathbf{0.269 \pm 0.003}$	0.324 ± 0.003
k-PC, $(k = 1)$	0.049 ± 0.003	0.217 ± 0.005
k-PC, $(k = 2)$	0.019 ± 0.002	0.121 ± 0.004
GES	0.092 ± 0.005	0.229 ± 0.008
GRaSP	0.069 ± 0.004	0.190 ± 0.007

Table 3: Average F1-arrowhead scores with standard errors rounded up to 3 decimal places

the experiment in the main paper section 4.1 for testing significance of the difference in $F1_{\text{skeleton}}$ and $F1_{\text{arrowhead}}$ between $\mathcal{C}$-PC and other baselines.

Sample Size	500	2000
$\mathcal{C}$**PC-Path**	$(0.791, 0.810)$	**(4.769e-4, -3.320)**
k**-PC,** $(k = 0)$	$(0.999, 4.646)$	**(7.748e-10, -6.135)**
k**PC,** $(k = 1)$	**(1.214e-154, -36.509)**	**(3.813e-80, -22.152)**
k**-PC,** $(k = 2)$	**(2.078e-204, -47.266)**	**(1.949e-174, -40.628)**
GES	**(2.093e-88, -23.708)**	**(3.233e-39, -14.042)**
GRaSP	**(6.823e-120, -29.663)**	**(5.385e-66, -19.456)**

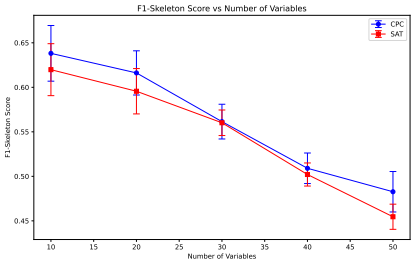
Table 4: One-sided two sample t-test result with format (p-value, t-value) under null hypothesis $\mathcal{H}_0 : \mu_{baseline} \geq \mu_{\mathcal{C}PC}$, where $\mu_{baseline}$ is the average F1-arrowhead scores of the baseline and $\mu_{\mathcal{C}PC}$ is the average F1 arrowhead scores of $\mathcal{C}$-PC. The degree of freedom is 598.

F ADDITIONAL EXPERIMENTS ON COMPARING $\mathcal{C}$-PC WITH SAT-BASED DISCOVERY ALGORITHM

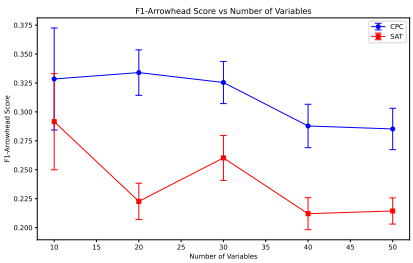
In this section, we will briefly discuss how a solver-based approach would perform as compared to $\mathcal{C}$-PC given an arbitrary selection of independence and dependence constraints based on a conditionally-closed set. We ran an experiment similar to our synthetic experiment in section 4.1 to compare an implementation of an SAT-based discovery algorithm Hyttinen et al. (2013) and -PC. We excluded the bidirected edges from the implementation as we assume causal sufficiency. In this experiment, we randomly generate DAGs of size $10, 20, 30, 40, 50$ with the number of edges of $20, 40, 60, 80, 100$. We fix the sample size to be 1000. Each variable is assigned either 2 states or 30 states with probabilities 0.7 and 0.3. We repeat this experiment 10 times for the number of variables only due to time constraints. We give both $\mathcal{C}$-PC and the SAT-based algorithm, the same CI constraints are found by the heuristic search. The result is shown in Figure 10. Based on the results, we have two observations. First, $\mathcal{C}$-PC outperforms the SAT-based method in terms of F1-arrowhead. They share roughly the same F1-skeleton score. The SAT-based algorithm has a higher F1-tail score. We attribute the higher F1-arrowhead and the lower F1-tail to the formulation of -essential graphs as it uses bidirected edges to represent some pairs of variables that may not be d-separated given any set in the conditionally-closed set. Secondly, we see that as the number of variables increases, the run time of the SAT-based discovery algorithm can far exceed that of $\mathcal{C}$-PC. This is expected as the number of CIs increases exponentially along with the size of the graph. This is important as there are downstream applications such as root cause analysis in microservices where PC-like algorithms are used with hundreds of variables Ma et al. (2020). Although PC-based procedures are more error-prone in general due to the orientation rules, they also allow us to orient the graph more efficiently when there are enough correct CI statements.

G RESULTS ON REAL-WORLD EXPERIMENT: THE COGNITION AND AGING USA (COGUSA) STUDY

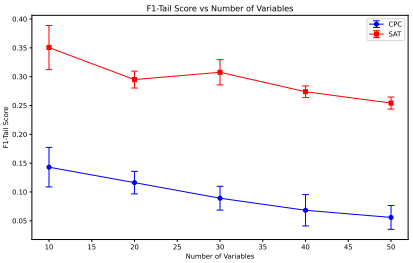
In this section, we present the output of each algorithm we used in the real-world experiment in section 4.2. The result is shown in Figures 11- 15. The dataset contains 16 discrete variables that are related to neuropsychological activities in the study with a total of 1514 samples. 8 variables have missing values. We set $\mathcal{C}$-PC to only conduct CI tests up to order 1 without conditioning on those 8 variables. All algorithms use test-wise deletion chi-squared tests with $\alpha = 0.05$. We use the same evaluation metrics as in (Strobl et al., 2018) to evaluate all the algorithms. Correct and incorrect relationships identified by expert knowledge are highlighted in green and red, respectively. $\mathcal{C}$-PC is able to capture the correct causal relationship between *backwards counting* (`w1_D_bc`) and *mental status* (`w1_D_ms`). MVPC has output incorrect relationships by having variables in the future causing variables in the past, e.g., (`w2_D_s7`)→(`w1_D_ms`). PCMCI does not output the incorrect relationship but it also fails to capture the correct relationship e.g (`w1_D_bc`) → (`w1_D_ms`). Additionally, We ran tPC algorithm Bang et al. (2024) via the Python library named `causal-learn` with $\alpha = 0.05$. As shown by Figure 15, we observe that although tPC does not orient any variable from week 2 to week 1 due to background knowledge, tPC fails to capture the directed edge from `w1_D_bc` to `w1_D_ms`.



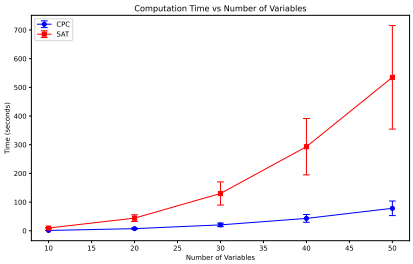
(a) $F1_{\text{skeleton}}$, $N = 1000$



(b) $F1_{\text{arrowhead}}$, $N = 1000$



(c) $F1_{\text{tail}}$, $N = 1000$



(d) Run time comparison

Figure 10: Comparison between $\mathcal{C}$-PC **(blue)** and a SAT-based discovery algorithm **(red)**. In this experiment, we randomly generate DAGs of size 10, 20, 30, 40, 50 with the number of edges of 20, 40, 60, 80, 100. We fix the sample size to be 1000. Each variable is assigned either 2 states or 30 states with probability 0.7 and 0.3. We repeat this experiment 10 times per the number of variables only due to time constraints. We give both $\mathcal{C}$-PC and the SAT-based algorithm the same CI constraints found by the heuristic search.

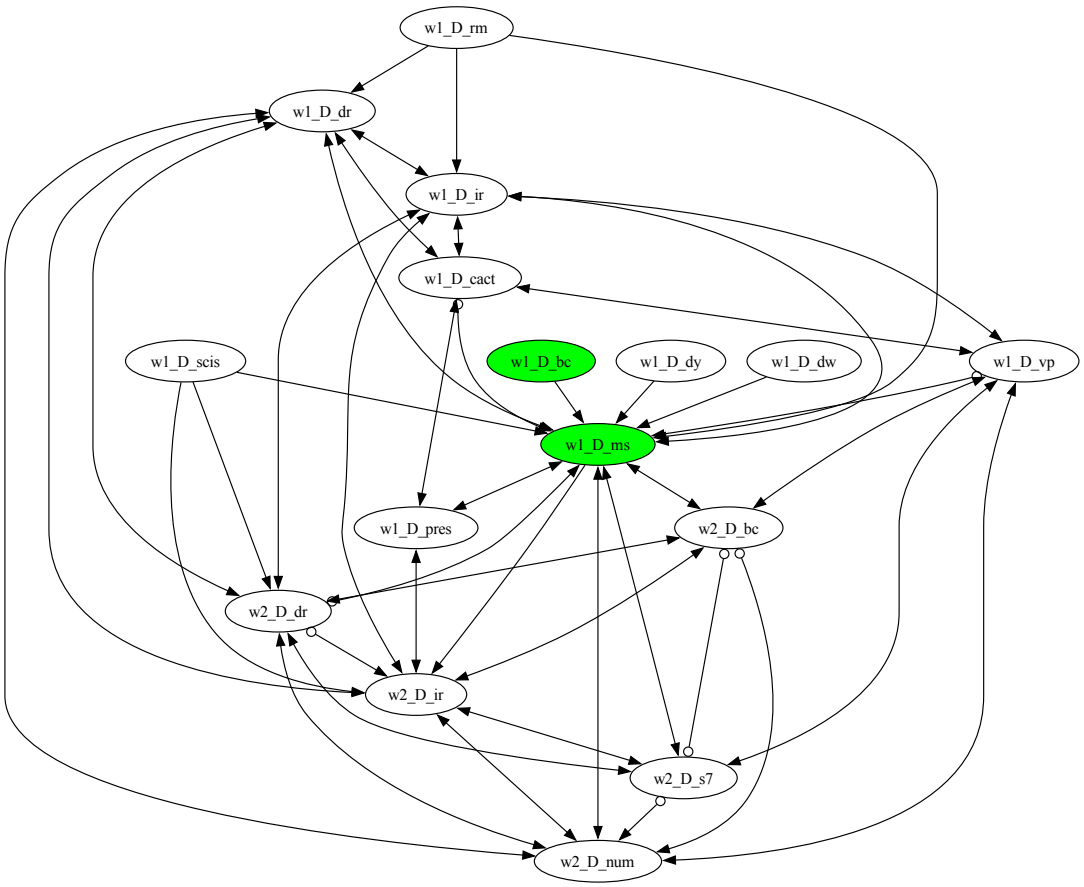


Figure 11: $\mathcal{C}$-PC's output from CogUSA data

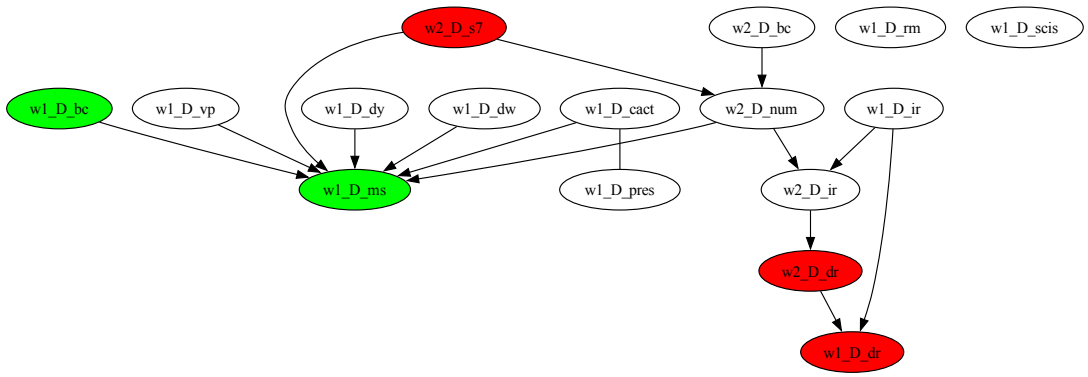


Figure 12: MVPC's output from CogUSA data

Figure 13: PCMCI output from CogUSA data

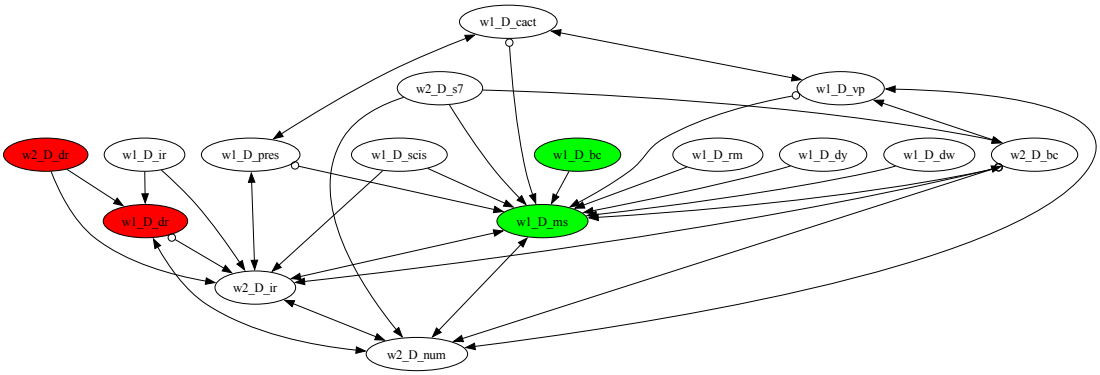


Figure 14: k-PC's output from CogUSA data, where $k = 1$

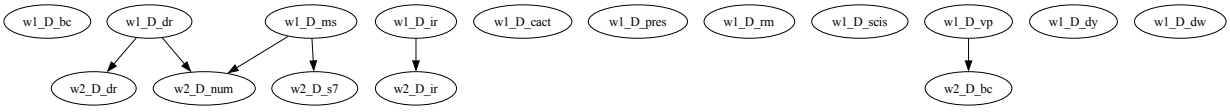


Figure 15: tPC output on CogUSA data