Constraint-based causal discovery with tiered background knowledge and latent variables in single or overlapping datasets

Christine W Bang BANG@LEIBNIZ-BIPS.COM

Vanessa Didelez DIDELEZ@LEIBNIZ-BIPS.COM
Leibniz Institute for Prevention Research and Epidemiology – BIPS, Bremen, Germany
Faculty of Mathematics and Computer Science, University of Bremen, Bremen, Germany

Editors: Biwei Huang and Mathias Drton

Abstract

In this paper we consider the use of tiered background knowledge within constraint based causal discovery. Our focus is on settings relaxing causal sufficiency, i.e. allowing for latent variables which may arise because relevant information could not be measured at all, or not jointly, as in the case of multiple overlapping datasets. We first present novel insights into the properties of the 'tiered FCI' (tFCI) algorithm. Building on this, we introduce a new extension of the IOD (integrating overlapping datasets) algorithm incorporating tiered background knowledge, the 'tiered IOD' (tIOD) algorithm. We show that under full usage of the tiered background knowledge tFCI and tIOD are sound, while simple versions of the tIOD and tFCI are sound and complete. We further show that the tIOD algorithm can often be expected to be considerably more efficient and informative than the IOD algorithm even beyond the obvious restriction of the Markov equivalence classes. We provide a formal result on the conditions for this gain in efficiency and informativeness. Our results are accompanied by a series of examples illustrating the exact role and usefulness of tiered background knowledge.

Keywords: Causal inference, graphical models, multi-cohort studies, temporal structure

1. Introduction

This work aims at exploiting tiered background knowledge for constraint-based causal discovery. The focus, here, is on relaxing causal sufficiency, i.e. allowing for latent variables which may arise, for instance, because relevant information could not be measured jointly or not at all. The former occurs in cases where datasets stem from e.g. different studies that have some but not all measured variables in common. We then speak of multiple (partially) overlapping datasets.

In classical constraint-based causal discovery algorithms, such as the PC (Spirtes et al., 2000) and FCI algorithms (Spirtes et al., 1999; Zhang, 2008), the entire joint independence structure of the data is available, where the FCI algorithm allows for latent variables, i.e. unmeasured common causes of measured variables. The Integrating Overlapping Datasets (IOD) algorithm (Tillman and Spirtes, 2011) extends the FCI to multiple datasets, i.e. multiple independence structures that are each marginal over those variables in the other datasets that they do not contain. Allowing for latent variables clearly induces a lower degree of identifiability of causal relations. It is therefore important to efficiently use any available background knowledge to improve identifiability. We focus on tiered background knowledge which arises in temporal data, in particular in cohort studies where multiple datasets occur for instance in multi-cohort designs (O'Connor et al., 2022).

As the IOD algorithm builds on the FCI algorithm, we first consider incorporating tiered background knowledge into the FCI and build on this to propose an IOD algorithm exploiting tiered background knowledge; we refer to these as the 'tiered' FCI/IOD, or 'tFCI/tIOD' algorithms. While the tFCI algorithm has been implemented (Scheines et al., 1998; Chen and Malinsky, 2023; Petersen, 2023) and applied (Lee et al., 2022) before, to our knowledge, only a few formal results have been proven. Therefore, we provide a formal description of the tFCI algorithm and, as a step towards showing correctness of the tIOD, we show correctness of the tFCI.

While algorithms like the FCI represent lack of identification in form of non-directed edges, the output of the IOD algorithm is often more complex: Not only does it allow for different edges within an equivalence class (represented as PAG), but it can also comprise of different equivalence classes. Background knowledge is then useful in two regards: To restrict the equivalence classes and to reduce the number of possible equivalence classes. The proposed tIOD algorithm outputs a set of graphs that are all consistent with the information available in all datasets and with the known tiered ordering. Moreover, we show that with tiered background knowledge we can reduce the number of of possible equivalence classes even without orienting any additional edges.

The motivation for our work is to be able to use data from multiple cohort studies (multi-cohorts) to learn causal structures across long time spans. Cohort studies are common in life sciences, such as life course epidemiology (Kuh and Ben-Shlomo, 2004), but the results here are valid for any type of data that has a tiered ordering. Tiered background knowledge is known to not only improve the informativeness of the estimated graphs (Bang and Didelez, 2023), but also the accuracy of discovery algorithms in finite samples (Bang et al., 2024). While these previous results are limited to the case of causal sufficiency and a single dataset, the present work relaxes these assumptions.

1.1. Related work

A precursor of the IOD algorithm was the Integrating Overlapping Networks (ION) algorithm (Tillman et al., 2008). Building on the IOD, Dhir and Lee (2020) propose orienting directed edges in the IOD algorithm using a bivariate causal discovery algorithm, but assume causal sufficiency and a particular data generating model. Huang et al. (2020) introduce two approaches that learn the entire DAG over multiple overlapping datasets under assumptions of linearity and non-Gaussianity. Other work on causal discovery for overlapping datasets first learns the individual models and then combines them using a SAT solver (Triantafillou et al., 2010; Tsamardinos et al., 2012; Triantafillou and Tsamardinos, 2015). We believe that constraint-based approaches are preferable, specifically with cohort data, since the data is likely to have missing values and mixed variable types with non-linear relations, and constraint-based methods can accommodate these issues flexibly (Witte et al., 2022).

1.2. Overview

In Section 2, we give an introduction to the type of data setting that motivates our work. Section 3 formally introduces (tiered) background knowledge and restricted equivalence classes, while an overview and background on the general framework is provided in Appendix A. In Sections 4, 5 and 6 we describe the tFCI and tIOD algorithms, and investigate their properties. In Section 7 we discuss the assumptions and limitations of the proposed approaches. We include pseudo-algorithms in Appendix B and proofs in Appendix D.

2. Overlapping cohort studies

Our work is motivated by the wish to combine different cohort studies, i.e. separate temporally structured datasets, for causal discovery with the ultimate aim of learning causal pathways over long time spans. It has been argued that multi-cohort designs can considerably advance life course research (O'Connor et al., 2022). The hope is to obtain a better understanding of e.g. how modifiable factors in early life and childhood affect health outcomes in adulthood and old age. An illustration is given in in Figure 1. The datasets might overlap in different aspects: Some may have taken measurements during the same period on the same and different variables, as the international and national children cohort studies in Figure 1; or they contain measurements of the same concept (e.g. 'healthy eating') at different time points, as the children's cohort studies and the adult health study with common measurements in adolescence.

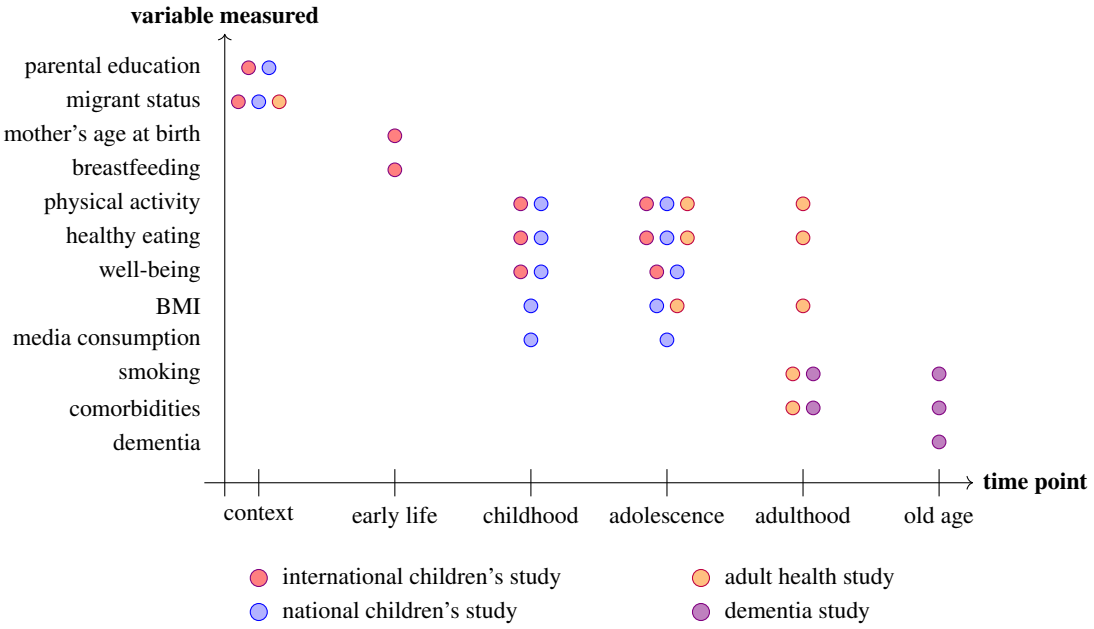

Figure 1: Toy example of how four different cohort studies can overlap in time and variables.

Thus, different cohort studies can overlap in variable sets and in time of measurements. We define a variable as being the same in two different dataset if and only if it is measured at the same time (or, here, life stage). This also means that two variables measuring the same concept, perhaps in the same dataset, are considered different if they do not refer to the same time point or life stage. For example, consider the variable 'physical activity' in Figure 1. This is measured in three different datasets and at three different time points. Physical activity during childhood and adolescence in the international children's cohort study are considered two different variables, while physical activity during adolescence in the international children's cohort study, in the national cohort study, and in the adult health study are all considered the same variable.

When variables are measured in some but not all datasets, this could in principle be viewed as a missing data problem. However, a key difference is that some variables are *never* measured together so that there is no information on their joint distribution. In the above example, breastfeeding, BMI

and dementia status are not measured jointly in any study. As pointed out by Tillman et al. (2008), if we wanted to, say, impute the values of unmeasured variables the correct imputation models cannot be obtained from the measured data and would rely on correct prior knowledge of the (causal) data generating mechanism. Clearly, this is not viable in the situation where determining the causal model is the desired aim.

While the example in Figure 1 only refers to cohort studies, in practice we could have other data structures. Although not required, our setup is most interesting in the case where all datasets to be combined contain at least one variable that is also measured in at least one of the other datasets.

3. Background knowledge

The temporal structure of cohort studies induces tiered background knowledge and, in turn, restricted equivalence classes which we formalise here. We refer to Appendix A for a general introduction to the relevant graphical concepts. We consider maximal ancestral graphs (MAGs) (Richardson and Spirtes, 2002), which allow for latent variables, and partial ancestral graphs (PAGs) encoding the equivalence classes of MAGs (Zhang, 2006).

Remark 1 *General MAGs allow for selection bias, but here, we assume that the data contains no selection bias. Thus, we assume that there is an underlying true MAG and that this MAG does not contain selection bias. This means that in the FCI algorithm, and consequently also in the IOD algorithm, the orientation rules R5-R7 can be omitted (Zhang, 2008).*

We define *background knowledge* $\mathcal{K} = (\mathbf{R}, \mathbf{F})$ relative to a node set $\mathbf{V}$ as a set of *required edges* $\mathbf{R}$ and a set of *forbidden edges* $\mathbf{F}$. A graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ *encodes* $\mathcal{K}$ if $\mathbf{E}$ contains all edges in $\mathbf{R}$ and no edges in $\mathbf{F}$. A PAG $\mathcal{P}$ is *consistent* with $\mathcal{K}$ if there is a MAG $\mathcal{M} \in [\mathcal{P}]$ encoding $\mathcal{K}$. We define the addition of background knowledge to a consistent PAG by orienting edges as in Algorithm 1. The output $\mathcal{P}_\mathcal{K}$ is a partial mixed graph (PMG) which may or may not be ancestral, and may or may not represent a restricted equivalence class. Throughout we assume the background knowledge to be consistent with some true underlying MAG, and in that sense *correct*.

Algorithm 1: Constructing $\mathcal{P}_\mathcal{K}$
input : PAG $\mathcal{P} = (\mathbf{V}, \mathbf{E})$ and consistent background knowledge $\mathcal{K} = (\mathbf{R}, \mathbf{F})$.
output: PMG $\mathcal{P}_\mathcal{K} = (\mathbf{V}, \mathbf{E}')$

$\mathbf{E}' = \mathbf{E}$
forall $\{V_i \circ\!\!-\!\!* V_j\} \in \mathbf{E}$ **do**
  if $\{V_i \to V_j\} \in \mathbf{F}$ **then**
    replace $\{V_i \circ\!\!-\!\!* V_j\}$ with $\{V_i \leftrightarrow V_j\}$ in $\mathbf{E}'$
    else if $\{V_i \to V_j\} \in \mathbf{R}$ **then**
      replace $\{V_i \circ\!\!-\!\!* V_j\}$ with $\{V_i \to V_j\}$ in $\mathbf{E}'$
end

Tiered background knowledge arises from partitioning the variables through a *tiered ordering*, such that variables in later tiers cannot cause variables in earlier tiers.

Definition 2 (Tiered ordering) *Let $\mathbf{V}$ be a node set of size p, and let $T \in \mathbb{N}$, $T \leq p$. A tiered ordering of the nodes in $\mathbf{V}$ is a map $\tau : \mathbf{V} \to \{1, \dots, T\}$ that assigns every node $V \in \mathbf{V}$ to a unique tier $t \in \{1, \dots, T\}$.*

Definition 3 (Tiered background knowledge) *Let τ be a tiered ordering of the node set $\mathbf{V}$, the corresponding tiered background knowledge $\mathcal{K}_\tau = (\mathbf{R}_\tau, \mathbf{F}_\tau)$ is then defined by $\mathbf{R}_\tau = \emptyset$ and $\mathbf{F}_\tau = \{V_i \leftarrow V_j \mid \tau(V_i) < \tau(V_j), V_i, V_j \in \mathbf{V}\}$.*

If tiered background knowledge is consistent with a graph, then we say that the corresponding tiered ordering is consistent with the graph.

In Algorithm 1 (line 4), a tiered ordering τ with $\tau(A) < \tau(B)$ means that an edge $A \ast\!\!-\!\!\circ B$ is oriented as $A \ast\!\!\rightarrow B$ by ruling out $A \leftarrow B$. We have knowledge of this orientation because the edge connects nodes from two different tiers, and we refer to this type of edges as *cross-tier edges*:

Definition 4 (Cross-tier edge) *Let $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ be a graph and let τ be a tiered ordering of the nodes in $\mathbf{V}$. Then, if $\tau(A) < \tau(B)$ the edge $\{A \ast\!\!-\!\!\ast B\} \in \mathbf{E}$ is referred to as a cross-tier edge.*

Importantly, tiered orderings not only inform us of possible edge directions in the form of a set of forbidden edges, they also imply certain restrictions on the separations.

Definition 5 *Let $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ be a graph and τ a tiered ordering of $\mathbf{V}$. Then for some $A \in \mathbf{V}$ we define the past of A in $\mathbf{V}$ relative to τ as $\text{past}^\tau_{\mathbf{V}}(A) = \{V \in \mathbf{V} \mid \tau(V) \leq \tau(A)\}$. For some $\mathbf{A} \subseteq \mathbf{V}$, where $\mathbf{A} = \{A_1, \ldots, A_n\}$, we define the past of $\mathbf{A}$ in $\mathbf{V}$ relative to τ as $\text{past}^\tau_{\mathbf{V}}(\mathbf{A}) = \{V \in \mathbf{V} \mid \tau(V) \leq \max(\tau(A_1), \ldots, \tau(A_n))\}$.*

The next proposition follows from well-known results on the separation properties of DAGs and MAGs, but we state it specifically for the context of tiered background knowledge.

Proposition 6 *Let $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ be a DAG or MAG, τ a tiered ordering of $\mathbf{V}$ consistent with $\mathcal{G}$, and let $A, B \in \mathbf{V}$ be two distinct nodes. Then A and B are separated in $\mathcal{G}$ by some subset of $\mathbf{V}$ if and only if they are separated by a set $\mathbf{S} \subseteq \text{past}^\tau_{\mathbf{V}}(A)$ or $\mathbf{S}' \subseteq \text{past}^\tau_{\mathbf{V}}(B)$.*

The proof of Proposition 6 can be found in Appendix D.

Proposition 6 implies that when checking for conditional independence, we can restrict the potential separating sets to those belonging to the (joint) past. For node pairs in later tiers, this may not have much impact, but for early node pairs, this may reduce the potential separating sets substantially.

4. The tFCI algorithm

For a single dataset and allowing for latent variables, the FCI algorithm Spirtes et al. (1999) constructs a PAG based on conditional independencies. Note that here and in the following sections we assume faithfulness (cf. Appendix A). In brief, the FCI starts with a complete undirected graph, and removes edges between pairs of nodes whenever the corresponding random variables are found to be (conditionally) independent. Afterwards, it orients v-structures using the conditional independencies learned in the previous phase, and finally it applies orientation rules R1-R10 from Zhang (2008) (see Figures 5 and 6 in Appendix A) to obtain a maximally informative PAG.

Tiered orderings are relevant to two aspects of the algorithm: For the conditional independence tests and for edge orientations. When restricting the tests performed by exploiting information from the tiered ordering via Proposition 6, we obtain the *simple* tFCI algorithm. Here, the objective is

still to learn a PAG representing the given independence model without restriction by the background knowledge. However, we show (in Proposition 7 below) that the reduction of conditional independence tests still results in an algorithm that is sound and complete, i.e. that the oracle version outputs the same PAG as the original FCI. This algorithm plays a central role, since the following algorithms build on it.

If we additionally orient arrowheads based on the tiered background knowledge (Algorithm 1 before applying rules R1-R4 and R8-R10), then we obtain the (full) tFCI algorithm. In principle, the objective here is to recover the maximally informative graph that encodes the given independence model and the tiered background knowledge. However, the full tFCI outputs a PMG that may or may not be maximally informative (i.e. the algorithm is not complete). It does represent a superset of the MAGs in the restricted equivalence class, i.e. the algorithm is sound (cf. Proposition 8 below). We provide the tFCI as a pseudo-algorithm (Algorithm 2) in Appendix B, where the simple version is given by omitting lines 30-34.

Versions of the tFCI algorithm have previously been implemented and applied (Scheines et al., 1998; Chen and Malinsky, 2023; Lee et al., 2022; Petersen, 2023). In fact, it is closely related to the SVAR-FCI by Malinsky and Spirtes (2018). While these authors probably took for granted that the tFCI is sound, we are not aware of any formal proofs, so that we provide them here. This also leads up to the corresponding properties for the tIOD algorithm below.

Proposition 7 *The oracle version of the simple tFCI algorithm (Algorithm 2 without lines 30-34) is sound and complete.*

Proposition 8 *The oracle version of the tFCI algorithm (Algorithm 2) is sound.*

The proofs of Propositions 7 and 8 can be found in Appendix D. The proof strategies are analogous to those in Spirtes et al. (1999) and Colombo et al. (2012) with additional use of Proposition 6 and assuming correct tiered background knowledge.

While the simple tFCI carries out fewer conditional independence tests, in the oracle case its output is identical to the one of the FCI and hence is a PAG (we comment on the sample version of the simple tFCI in Section 6 and the discussion). However, it is important to note that, in contrast to the simple tFCI, the full tFCI algorithm might not be complete, and the output might not be a maximally informative PAG. This is because with the addition of tiered background knowledge we might need more orientation rules than R1-R4 and R8-R10. Completeness of the FCI under certain types of background knowledge has been shown, e.g., local background knowledge (Wang et al., 2022), context variables (Mooij et al., 2020), or a restricted kind of tiered background knowledge (Andrews et al., 2020). But for the general case, it is still an open question whether all orientation rules needed to ensure completeness under tiered background knowledge have been found (Venkateswaran and Perković, 2024).

5. The tIOD algorithm

Let us first review the basic IOD algorithm (Tillman and Spirtes, 2011) without background knowledge. It takes as input n datasets $\mathbf{V}_1, \ldots, \mathbf{V}_n$ of which any pair may share some variables[1]. The

1. The extreme cases are that all datasets share all variables (which can then be tackled with the FCI for a unique dataset) or that the datasets have no variables in common, in which case the output set of PAGs will be very large.

underlying assumption of the IOD algorithm is that there is a MAG $\mathcal{M} = (\mathbf{V}, \mathbf{E})$ on the full set of variables $\mathbf{V} = \mathbf{V}_1 \cup \ldots \cup \mathbf{V}_n$ of which the individual datasets represent the marginals corresponding to their respective subset of variables. The IOD algorithm returns a *set* of PAGs for which the marginalised independence models onto the nodes in $\mathbf{V}_1, \ldots, \mathbf{V}_n$ are equal to the independence models learned from the individual datasets. Furthermore, this set is guaranteed (in the oracle case) to contain the PAG of $\mathcal{M}$. The IOD algorithm consists of two parts: The first part learns the independence models from the individual datasets, and uses these models to construct a graph $\mathcal{G}$, which contains a superset of the adjacencies of $\mathcal{M}$, and a subset of the v-structures of $\mathcal{M}$. An individual independence model is necessarily marginal over those variables not included in this dataset and, hence, must allow for latent variables. The second part of the IOD algorithm constructs a graph for each possible combination of edge removals or v-structure orientations encoding an independence model that can be marginalised into all of the n independence models learned from data. We illustrate the basic IOD algorithm with the following example.

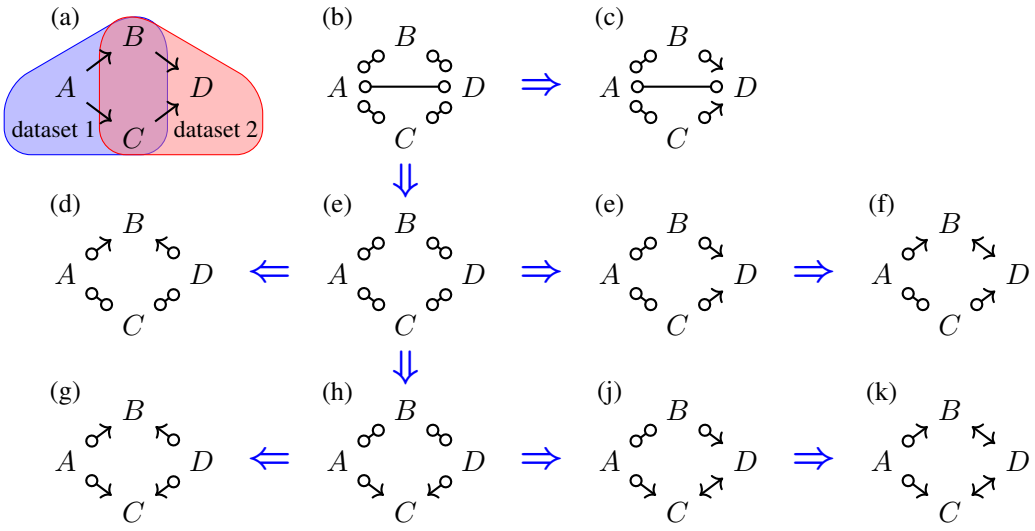

Figure 2: Examples of graphs visited by the IOD algorithm. Here, (e) is the PAG of (a), but more graphs may encode the marginal independence models learned from dataset 1 and dataset 2. Given oracle knowledge of (a) as input, the algorithm considers all graphs where combinations of the edges $A \circ\!\!-\!\!\circ D$, $A \circ\!\!-\!\!\circ B$, $A \circ\!\!-\!\!\circ C$, $B \circ\!\!-\!\!\circ D$ and $C \circ\!\!-\!\!\circ D$ are removed. Here, we illustrate the removal of $A \circ\!\!-\!\!\circ D$. In total, the IOD visits 73 graphs, including graphs based on (b), (c),. . ., (k), and it outputs the eight graphs in Figure 3.

Example 1 (The IOD algorithm) *Figure 2 (a) shows a MAG over the nodes $\{A, B, C, D\}$. Here, X_A, X_B and X_C have been measured in dataset 1, and X_B, X_C and X_D in dataset 2. When learning the causal structure of $\{A, B, C, D\}$ given dataset 1 and dataset 2, the IOD algorithm proceeds as follows: First it constructs a PMG over $\{A, B, C\}$ and one over $\{B, C, D\}$, and while doing so it also constructs a common graph $\mathcal{G}$. At first, $\mathcal{G}$ is a complete undirected graph over $\{A, B, C, D\}$. Then, for every pair of variables that are conditionally independent in dataset 1 or dataset 2, the edge between the corresponding nodes in $\mathcal{G}$ is removed, and we obtain the graph*

in Figure 2 (b-i). At this stage of the IOD algorithm, nodes are adjacent if either they cannot be separated in the marginal graph over $\{A, B, C\}$ or $\{B, C, D\}$, or if they are never measured jointly in the same dataset. Hence, $\mathcal{G}$ contains a superset of the edges of the true underlying graph. Next, the algorithm considers all edges that might not be contained in the true graph, either because the variables were never measured jointly, or because all variables necessary for obtaining conditional independence were never measured jointly. In this example, all edges are candidates for this, and the algorithm considers each combination of edge removals (32 candidate skeletons in total). Consider the case of the removal of edge $A \circ\!\!-\!\!\circ D$: The algorithm proceeds with two possible graphs, Figure 2 (b-i) and (c-i). Now, the IOD algorithm considers all possible v-structures. A v-structure might be missing in Figure 2 (b-i) and (c-i) if for every dataset either the collider is not measured, or there is no separating set measured in the dataset. The triple $\langle B, D, C\rangle$ might be a v-structure, since it cannot be tested whether X_B and X_C are conditionally independent given X_A and X_D – all we know is that they are conditionally independent given X_A alone, but dependent give X_D alone, which at this stage is not sufficient to determine whether $\langle B, D, C\rangle$ is a v-structure. Moreover, if A and D are not adjacent, then $\langle A, B, D\rangle$ or $\langle A, C, D\rangle$ might be v-structures as well: We cannot test this since X_A and X_D are never measured jointly. Given the skeleton in Figure 2 (b-i), the only possible additional v-structure is $\langle B, D, C\rangle$ (Figure 2 (b-ii)), but the skeleton in Figure 2 (c-i) allows for all combinations of $\langle B, D, C\rangle$, $\langle A, B, D\rangle$ and $\langle A, C, D\rangle$ (Figure 2 (c-ii) to (c-iix)), so the algorithm proceeds with all these possible graphs. The algorithm further proceeds by orienting edges according to the orientation rules R1–R10, constructing additional graphs when multiple discriminating paths are consistent with the information (see Tillman and Spirtes (2011) for details). Finally, it verifies that from all constructed graphs, we can construct MAGs that are consistent with the marginal independence models. It then outputs this set of PAGs. In the present example, the IOD algorithm visits a total of 73 graphs and outputs 8 PAGs. See Table 1 in Appendix C for an overview, and Figure 3 in Example 2 for a depiction of the graphical output.

We extend the IOD algorithm to incorporate tiered background knowledge in multiple ways. First, we use it to restrict the conditional independence tests in the first part of the algorithm, analogously to the simple tFCI. Second, we use the tiered background knowledge for additional restrictions involving adjacencies and v-structures so as to exclude some PAGs that are inconsistent with the background knowledge. With these two steps, the objective is to learn the set of all PAGs, and only those PAGs, that are compatible with the marginal independence models and consistent with the tiered background knowledge. As we will see, tiered background knowledge reduces the number of potential edge removals, and the algorithm thus visits fewer potential skeletons. Moreover, it also increases the number of identifiable v-structures compared to no background knowledge. We refer to the combination of these two modifications as the *simple tIOD* algorithm (Algorithm 3 without lines 67-72).

In addition, we may also orient cross-tier edges and further edges as consequences of the former due to rules R1-R4 and R8-R10 (Figures 5 and 6). The algorithm using all three modifications is referred to as the (full) tIOD algorithm (Algorithm 3). Here, the desired objective is to recover all maximally informative graphs that are compatible with the marginal independence models and encode the tiered background knowledge. However, for the same reasons as the tFCI, the algorithm is only sound, and not necessarily complete: The tIOD outputs a set of PMGs that might not represent restricted equivalence classes. However, the algorithm is guaranteed to output at least one PMG that represents a set of MAGs containing the true underlying MAG (Proposition 10).

In summary, we have the following modifications and results.

Modifications defining the simple tIOD algorithm (Algorithm 3)

(i) When testing conditional independence in the first round, only consider separating sets that belong to the past (lines 3-13).

(ii) Only condition on sets of nodes that belong to the past when testing for v-structures, and orient cross-tier edges that constitute v-structures (lines 14-18 and 27-31).

(iii) Only consider possible separating sets that belong to the past when constructing the final skeleton and v-structures (lines 19-26).

(iv) When considering removable edges, restrict possible separating sets to the past (lines 33-37).

(v) Orient v-structures consisting of cross-tier edges before considering possible v-structures (lines 39-47).

(vi) Discard every PAG that is not consistent with the tiered background knowledge (line 65).

Additional modifications defining the full tIOD algorithm (Algorithm 3)

(vi) For each PAG in the final output, orient any remaining cross-tier edges and apply orientation rules R1-R4 and R8-R10 (lines 67-72).

The simple and full tIOD algorithms have the following properties.

Proposition 9 *The oracle version of the simple tIOD algorithm (Algorithm 3 without lines 67-72) is sound and complete for the set of PAGs consistent with the tiered background knowledge.*

Proposition 10 *The oracle version of the full tIOD algorithm (Algorithm 3) is sound.*

The proofs are given in Appendix D. The proof strategies are analogous to those in Tillman and Spirtes (2011), but make use of Proposition 6, and exploit the tiered background knowledge which is assumed to be correct.

6. Efficiency and informativeness

Tiered background knowledge leads to more informative outputs whenever the tiered ordering implies the orientation of cross-tier edges that are not involved in v-structures; this may even imply additional informativeness due to the orientation rules. Thus, the output contains more information than could be obtained from the independence model. In particular, the tFCI and tIOD algorithms then output a set of graphs that contain more information than the independence models alone. This is certain for the oracle case and can be expected for the finite sample case.

Without the orientation of cross-tier edges, the advantages might not be as obvious. The simple tFCI algorithm outputs the same PAG as the FCI algorithm under oracle knowledge. The benefit of the simple tFCI is found, instead, in the sample case, where the output is prone to statistical errors. By skipping unnecessary statistical tests, the output is more robust as was found for the tiered PC algorithm, i.e. in the causally sufficient case (Bang et al., 2024). Analogously, we expect the simple tiered algorithms to be more robust against statistical errors than the original FCI/IOD algorithms.

However, the simple tIOD algorithm has an interesting additional benefit. Even the simple modifications alone, without orienting cross-tier edges, outputs a set of PAGs that should often be more informative than with the original IOD algorithm in the sense that the set contains fewer PAGs, and, hence, represents fewer possible equivalence classes. This is illustrated in Examples 2 and 3. The reason that the simple tIOD algorithm might output fewer PAGs than the IOD algorithm is that it can usually reduce the set of possible edge removals and possible v-structure orientations based on the tiered ordering. In this case, the simple tIOD algorithm is more efficient than the IOD algorithm as it needs to visit fewer graphs, cf. Example 2. Here, by efficiency we refer to the algorithm visiting fewer graphs. Below we provide a proposition that states under what conditions the simple tIOD algorithm is more efficient than the IOD.

Proposition 11 considers sets $\mathrm{pds}_{\mathcal{G}}(A, B)$ and $\mathrm{pds}_{\mathcal{G}}(B, A)$ which are sets of nodes in $\mathcal{G}$ containing a set that m-separates A and B if and only if any subset of the nodes in $\mathcal{G}$ m-separates A and B (Spirtes et al., 1999; Colombo et al., 2012). These sets are used by the algorithms to determine the independence models. See Appendix A for a formal definition.

Proposition 11 *Consider the simple tIOD algorithm over $\mathbf{V} = \mathbf{V}_1 \cup \ldots \cup \mathbf{V}_n$ using τ, where $\tau : V \to \{1, \ldots, k\}$ for $k > 1$. Let $\mathcal{G}$ be the graph obtained in line 32. Then the oracle version of the simple tIOD algorithm visits fewer graphs than the oracle version of the IOD algorithm over $\mathbf{V} = \mathbf{V}_1 \cup \ldots \cup \mathbf{V}_n$, if and only if one of the two following conditions holds:*

(i) Consider a pair of nodes A, B such that $A \in \mathrm{adj}_{\mathcal{G}}(B)$, and for all $i \in \{1, \ldots, n\}$ we have

$$\{A\} \cup \mathrm{adj}_{\mathcal{G}}(A) \cup \mathrm{pds}_{\mathcal{G}}(A, B) \cup \mathrm{pds}_{\mathcal{G}}(B, A) \not\subset \mathbf{V}_i \quad \textit{and}$$

$$\{B\} \cup \mathrm{adj}_{\mathcal{G}}(B) \cup \mathrm{pds}_{\mathcal{G}}(A, B) \cup \mathrm{pds}_{\mathcal{G}}(B, A) \not\subset \mathbf{V}_i.$$

For at least one such pair we require that $\exists\, i$ such that $A, B \in \mathbf{V}_i$ and

$$\big(\mathrm{pds}_{\mathcal{G}}(A, B) \cup \mathrm{pds}_{\mathcal{G}}(B, A)\big) \setminus \mathbf{V}_i \subseteq \mathbf{V} \setminus \mathrm{past}^{\tau}_{\mathbf{V}}(A, B) \quad \textit{and}$$

$$\mathrm{adj}_{\mathcal{G}}(A) \setminus \mathbf{V}_i \subseteq \mathbf{V} \setminus \mathrm{past}^{\tau}_{\mathbf{V}}(A) \;\; \textit{or} \;\; \mathrm{adj}_{\mathcal{G}}(B) \setminus \mathbf{V}_i \subseteq \mathbf{V} \setminus \mathrm{past}^{\tau}_{\mathbf{V}}(B).$$

(ii) Consider a triple of nodes $\langle A, C, B \rangle$ such that (a) $A, B \in \mathrm{adj}_{\mathcal{G}}(C)$, (b) $A \notin \mathrm{adj}_{\mathcal{G}}(B)$, and (c) for all $i \in \{1, \ldots n\}$ either $C \notin \mathbf{V}_i$ or $\mathrm{sepset}(\mathcal{G}_i, \{A, B\})$ is undefined. For at least one such triple we require that $\tau(C) > \max(\tau(A), \tau(B))$.

The proof of Proposition 11 can be found in Appendix D. The proof is based on the fact that the number of visited graphs depends on the number of potential edge removals and potential v-structures. These numbers decrease in the case where tiered background knowledge renders either the potential edge removals impossible, or the v-structures certain. This occurs when for some node pairs all possible conditioning sets that have not been measured in the same dataset lie in the future (condition (i)), or if some v-structures are implied by the tiered ordering (condition (ii)).

Remark 12 *Let $\mathcal{P}$ be a set of PAGs over $\mathbf{V}$ obtained by the oracle version of the IOD algorithm, and $\mathcal{P}_\tau$ a set of PAGs obtained by the oracle version of the simple tIOD algorithm. It then follows from Proposition 11 that $\mathcal{P}_\tau$ is a subset of $\mathcal{P}$ only if at least one of the conditions holds.*

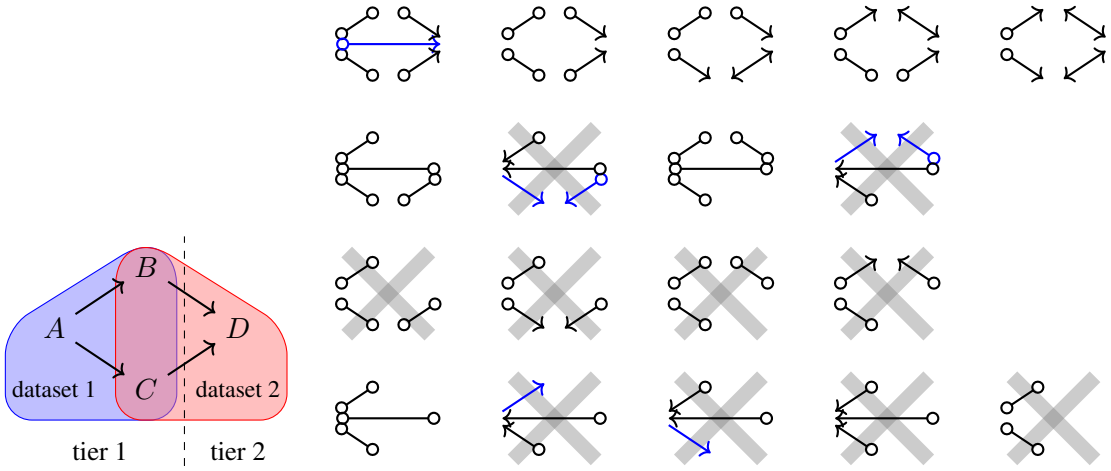

Figure 3: Left: A MAG $\mathcal{M}$ with tiered ordering τ, where the variables are measured in two datasets, dataset 1 and dataset 2. Right: All graphs visited by the tIOD algorithm (Algorithm 3). Black edges are obtained up to line 50, blue edges are obtained from orientation rules R1-R4 and R8-R10. Crossed out graphs do not satisfy the criteria of line 65; all other graphs are output by the tIOD algorithm. In this example, the IOD algorithm visits more graphs but still outputs the same PAGs as the simple tIOD.

Note that the output of the tFCI always has the same skeleton and at most as many circle marks as the output of the FCI. Hence, it is always at least as informative as the FCI. Moreover, the simple tIOD also always outputs a (weak) subset of the output of IOD, thus it is at least as informative.

Remark 13 *Since the simple and full tIOD are identical up until line 66, the full tIOD will consider the same potential skeletons and v-structures as the simple tIOD.*

Example 2 *Consider the MAG $\mathcal{M}$ of Figure 3 (a), which is the same as in Example 1 but with a tiered ordering, such that nodes A, B and C belong to tier 1, and D belongs to tier 2. Then Proposition 6 implies that the only candidate separating set for A and B is C and the only candidate separating set for A and C is B. Since all of the corresponding variables are measured in the same dataset, we see that these are in fact not separating sets, and it is not necessary to consider skeletons without $A \circ\!\!-\!\!\circ B$ and $A \circ\!\!-\!\!\circ C$ in the tIOD algorithm. The tIOD algorithm considers $A \circ\!\!-\!\!\circ D$, $B \circ\!\!-\!\!\circ D$, and $C \circ\!\!-\!\!\circ D$ as removable edges and considers 8 different skeletons (see Table 2 in Appendix C), which is considerably fewer than visited by the IOD algorithm in Example 1.*

In dataset 1 we find that $X_B \perp\!\!\!\perp X_C \mid X_A$. In dataset 2 we find no independencies, in particular $X_B \not\perp\!\!\!\perp X_C \mid X_D$, and in all graphs with both edges $B \circ\!\!-\!\!\circ D$ and $C \circ\!\!-\!\!\circ D$ present, $\langle B, D, C\rangle$ is considered a potential v-structure by the IOD algorithm in Example 1, while the tIOD algorithm orients $\langle B, D, C\rangle$ as a v-structure because D is in a later tier than B and C. In total, the simple tIOD algorithm visits 18 graphs (see Table 2 in Appendix C) and outputs 8 graphs.

Example 3 *Example 2 is a case where the tIOD visits fewer graphs but outputs the same PAGs as the IOD. A key reason for this is that both algorithms require the output to be consistent with the*

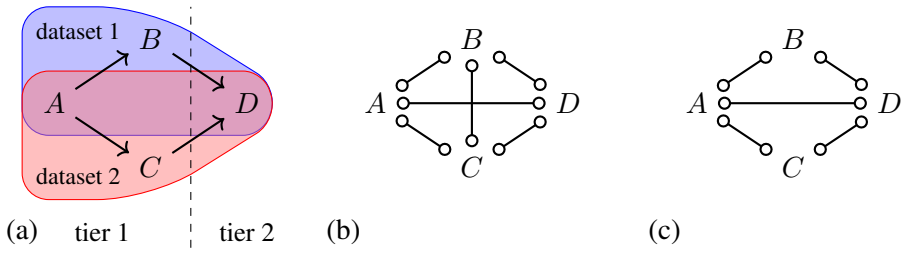

Figure 4: (a) MAG with tiered ordering of the nodes, with measurements in dataset 1 and dataset 2. (b) The intermediate graph $\mathcal{G}$ obtained at line 32 of the tIOD algorithm (Algorithm 3) with oracle knowledge of (a) as input. (c) An example graph output by the IOD algorithm, with (a) as input), that would not have been output by the simple tIOD.

learned (marginal) independence models. The only independence found was $X_B \perp\!\!\!\perp X_C \mid X_A$. *This implicitly requires the graph to have a path between* B *and* C, *that can be separated by* A *without also conditioning on* D *and requires* $\langle B, D, C\rangle$ *to be a v-structure whenever it is an unshielded triple, which much reduces the number of output graphs.*

A different situation is illustrated in Figure 4, where nodes A, B, D *are measured in dataset 1, while* A, C, D *are measured in dataset 2, with the same true MAG as before. No independencies are found in dataset 1 and dataset 2. This leaves a complete graph as the intermediate graph* $\mathcal{G}$, *Figure 4 (b). Hence, the IOD algorithm outputs, among others, Figure 4 (c). However, this graph cannot be output by the tIOD algorithm since it orients* $\langle B, D, C\rangle$ *as a v-structure in every graph where this is an unshielded triple. In this case, the tIOD algorithm is not just more efficient, it also outputs fewer graphs and is thus more informative.*

7. Discussion

We have introduced extensions of constraint-based causal discovery algorithms that exploit tiered background knowledge while allowing for latent variables and multiple overlapping datasets. The simple versions of tFCI and tIOD can be seen as estimating (sets of) equivalence classes whereas the full versions make further use of the background knowledge at the expense of completeness. We have shown that the simple versions of tFCI and tIOD are sound and complete, and we have shown when exactly using the background knowledge reduces the number of graphs visited by the simple tIOD. In consequence, the simple (not just the full) tIOD outputs a set of graphs that can often be expected to be considerably more informative than without the tiered ordering.

The results in this paper apply to the oracle case. In practice, however, conditional independence testing is subject to statistical errors. Here, the simple tFCI and tIOD are attractive as they omit unnecessary conditional independence tests by exploiting Proposition 6. Indeed, for analogous reasons, tiered background knowledge has been shown to improve the statistical properties of the tiered PC algorithm (Bang et al., 2024) – similarly robust finite sample behavior can be expected for the sample versions of the tFCI and tIOD algorithms. A further reason for exploiting tiered background knowledge to detect v-structures, as done by the simple tFCI and simple tIOD, is to

ensure that the estimated equivalence classes are consistent with that background knowledge which is not otherwise guaranteed for finite samples.

We have explicitly assumed one single underlying joint MAG over all variables. This means that the different datasets must inform us about comparable settings. Hence, the marginal equivalence classes are consistent on the overlapping nodes under oracle knowledge, so that it makes sense to combine them. However, in finite samples we may encounter the case that the independence models, if estimated separately with the different datasets, are not consistent on the overlapping variables even under this assumption. This can easily be fixed by combining the statistical tests and / or the datasets on the overlapping variables before inputting the estimated independence models into the algorithm; one such solution is given in (Tillman and Spirtes, 2011).

A number of further extensions could be interesting for future work. For instance, the principles of the tFCI algorithm could be applied to other versions of the FCI algorithm such as the anytime FCI (Spirtes, 2001), the RFCI (Colombo et al., 2012), the FCI+ (Claassen et al., 2013), the order independent or conservative FCI (Colombo and Maathuis, 2014), etc. Future research might explore how the ideas of these algorithms could also be utilised in the tIOD algorithm.

Often, tiered background knowledge is not the only kind of background knowledge available. Typical epidemiological applications, in fact, often rely on fully specified expert causal graphs rather than data driven graph construction (Petersen et al., 2023). More realistically, even experts may only have partial knowledge of the causal structure (Didelez, 2024). It would be an interesting direction for future research to explore how one could use the tIOD algorithm to combine or refine graphs based on a mix of (possibly multiple) experts and (possibly multiple) data sources.

Our work is motivated by multi-cohort studies, which may or may not contain repeated measures, but always contain a tiered structure. Temporal structure is also available with time series data. Extensions of the FCI algorithm to time series exist (Malinsky and Spirtes, 2018; Gerhardus and Runge, 2020). These are based on similar principles, essentially exploiting corresponding versions of Proposition 6, i.e. not conditioning on the future. Additional common assumptions such as stationarity and considerations of autocorrelation are crucial for time series which we did not cover, here. While we are not aware that multiple overlapping datasets have been considered in the literature on causal analysis of time series, it is conceivable that a subvector of the multivariate time series is measured in one dataset while another subvector is measured in another dataset. It may be promising to develop a version of the tIOD algorithm for such settings.

Acknowledgments

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project numbers 281474342/GRK2224/2 and 459360854. We would like to thank the reviewers for their constructive comments, which helped improve the paper.

References

Bryan Andrews, Peter Spirtes, and Gregory F Cooper. On the completeness of causal discovery in the presence of latent confounding with tiered background knowledge. In *International Conference on Artificial Intelligence and Statistics*, pages 4002–4011. PMLR, 2020.

Christine W Bang and Vanessa Didelez. Do we become wiser with time? On causal equivalence with tiered background knowledge. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 119–129. PMLR, 2023.

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.

Qixuan Chen and Daniel Malinsky. tFCI, 2023. URL https://github.com/QixShawnChen/tfci.

Tom Claassen, Joris M Mooij, and Tom Heskes. Learning sparse causal models is not np-hard. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 172–181, 2013.

Diego Colombo and Marloes H Maathuis. Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 15(116):3741–3782, 2014.

Diego Colombo, Marloes H Maathuis, Markus Kalisch, and Thomas S Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, pages 294–321, 2012.

Anish Dhir and Ciarán M Lee. Integrating overlapping datasets using bivariate causal discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3781–3790, 2020.

Vanessa Didelez. Invited commentary: where do the causal DAGs come from? *American Journal of Epidemiology*, page kwae028, 2024.

Andreas Gerhardus and Jakob Runge. High-recall causal discovery for autocorrelated time series with latent confounders. *Advances in Neural Information Processing Systems*, 33:12615–12625, 2020.

Biwei Huang, Kun Zhang, Mingming Gong, and Clark Glymour. Causal discovery from multiple data sets with non-identical variable sets. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 10153–10161, 2020.

Diana Kuh and Yoav Ben-Shlomo. *A life course approach to chronic disease epidemiology*. Oxford university press, 2nd edition, 2004.

Jaron JR Lee, Ranjani Srinivasan, Chin Siang Ong, Diane Alejo, Stefano Schena, Ilya Shpitser, Marc Sussman, Glenn JR Whitman, and Daniel Malinsky. Causal determinants of postoperative length of stay in cardiac surgery using causal graphical learning. *The Journal of Thoracic and Cardiovascular Surgery*, 2022.

Daniel Malinsky and Peter Spirtes. Causal structure learning from multivariate time series in settings with unmeasured confounding. In *Proceedings of 2018 ACM SIGKDD Workshop on Causal Discovery*, pages 23–47. PMLR, 2018.

Joris M Mooij, Sara Magliacane, and Tom Claassen. Joint causal inference from multiple contexts. *Journal of Machine Learning Research*, 21:1–108, 2020.

Meredith O'Connor, Elizabeth Spry, George Patton, Margarita Moreno-Betancur, Sarah Arnup, Marnie Downes, Sharon Goldfeld, David Burgner, and Craig A Olsson. Better together: advancing life course research through multi-cohort analytic approaches. *Advances in Life Course Research*, 53:100499, 2022.

Anne Helby Petersen. *causalDisco: Tools for Causal Discovery on Observational Data*, 2023. URL https://github.com/annennenne/causalDisco. R package version 0.9.2.

Anne Helby Petersen, Claus Thorn Ekstrøm, Peter Spirtes, and Merete Osler. Constructing causal life-course models: Comparative study of data-driven and theory-driven approaches. *American Journal of Epidemiology*, 192(11):1917–1927, 2023.

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

Richard Scheines, Peter Spirtes, Clark Glymour, Christopher Meek, and Thomas Richardson. The tetrad project: Constraint based aids to causal model specification. *Multivariate Behavioral Research*, 33(1):65–117, 1998.

Peter Spirtes. An anytime algorithm for causal inference. In *International Workshop on Artificial Intelligence and Statistics*, pages 278–285. PMLR, 2001.

Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse causal graphs. *Social science computer review*, 9(1):62–72, 1991.

Peter Spirtes, Christopher Meek, and Thomas Richardson. *Computation, Causation, and Discovery*, chapter An algorithm for causal inference in the presence of latent variables and selection bias, pages 211–252. MIT press, 1999.

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

Robert Tillman and Peter Spirtes. Learning equivalence classes of acyclic models with latent and selection variables from multiple datasets with overlapping variables. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 3–15. JMLR Workshop and Conference Proceedings, 2011.

Robert Tillman, David Danks, and Clark Glymour. Integrating locally learned causal structures with overlapping variables. *Advances in Neural Information Processing Systems*, 21, 2008.

Sofia Triantafillou and Ioannis Tsamardinos. Constraint-based causal discovery from multiple interventions over overlapping variable sets. *The Journal of Machine Learning Research*, 16(1): 2147–2205, 2015.

Sofia Triantafillou, Ioannis Tsamardinos, and Ioannis Tollis. Learning causal structure from overlapping variable sets. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 860–867. JMLR Workshop and Conference Proceedings, 2010.

Ioannis Tsamardinos, Sofia Triantafillou, and Vincenzo Lagani. Towards integrative causal analysis of heterogeneous data sets and studies. *The Journal of Machine Learning Research*, 13(1):1097–1157, 2012.

Aparajithan Venkateswaran and Emilija Perković. Towards complete causal explanation with expert knowledge. *arXiv preprint arXiv:2407.07338*, 2024.

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

Tian-Zuo Wang, Tian Qin, and Zhi-Hua Zhou. Sound and complete causal identification with latent variables given local background knowledge. *Advances in Neural Information Processing Systems*, 35:10325–10338, 2022.

Janine Witte, Ronja Foraita, and Vanessa Didelez. Multiple imputation and test-wise deletion for causal discovery with incomplete cohort data. *Statistics in Medicine*, 41(23):4716–4743, 2022.

Jiji Zhang. *Causal inference and reasoning in causally insufficient systems*. PhD thesis, Citeseer, 2006.

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, 2008.

Appendix A. Terminology

A.1. Ancestral graphs

Graphs, nodes and edges A graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ consists of a set of *nodes* $\mathbf{V}$ and *edges* $\mathbf{E}$. We allow edges to have three types of *edge marks*: $\circ$ (*circle mark*), $-$ (*tail*) and $>$ (*arrowhead*), and we allow the following types of edges: $\circ\!\!-\!\!\circ$ (*non-directed*), $\circ\!\!\rightarrow$ (*partially directed*), $\rightarrow$ (*directed*), and $\leftrightarrow$ (bidirected). We use $*\!\!-\!\!*$ (both edge marks can be of any type), $*\!\!\rightarrow$ (left edge mark can be of any type) and $*\!\!-\!\!\circ$ (left edge mark can a circle mark or arrowhead) as placeholders. We say that an edge (or tail) $A \rightarrow B$ is out of A and an edge (or arrowhead) $A*\!\!\rightarrow B$ is into B. Let $A, B \in \mathbf{V}$ and $\{A *\!\!-\!\!* B\} \in \mathbf{E}$, then A and B are *adjacent* in $\mathcal{G}$. If all pairs of nodes in a graph are adjacent, then the graph is *complete*. By $\mathrm{adj}_{\mathcal{G}}(A)$ we denote all the nodes that are adjacent to $A \in \mathbf{V}$ in $\mathcal{G}$. We allow for at most one edge between any pair of nodes, and no node can be adjacent to itself. A graph that only contains directed/bidirected/non-directed edges is a directed/bidirected/non-directed graph.

Graphs and subgraphs Let $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ and $\mathcal{G}' = (\mathbf{V}', \mathbf{E}')$ be distinct graphs. If $\mathbf{V}' \subseteq \mathbf{V}$ and $\mathbf{E}' \subseteq \mathbf{E}$ then $\mathcal{G}'$ is a *subgraph* of $\mathcal{G}$. The *non-directed subgraph* $\mathcal{G}_U = (\mathbf{V}, \mathbf{E}_U)$ of $\mathcal{G}$ is obtained by removing all edges in $\mathbf{E}$ that contain an arrowhead, such that $\mathbf{E}_U \subseteq \mathbf{E}$. The *induced subgraph* of $\mathcal{G}$ over $\mathbf{V}' \subseteq \mathbf{V}$ is defined as $\mathcal{G}_{\mathbf{V}'} = (\mathbf{V}', \mathbf{E}_{\mathbf{V}'})$ where $\mathbf{E}_{\mathbf{V}'} \subseteq \mathbf{E}$ contains all edges between nodes in $\mathbf{V}'$. By $\mathcal{G}_{\mathbf{E}'} = (\mathbf{V}, \mathbf{E}')$ we denote the subgraph of $\mathcal{G}$ obtained by removing all edges not in $\mathbf{E}' \subseteq$. The skeleton of $\mathcal{G}$ is obtained by replacing every $\{A *\!\!-\!\!* B\} \in \mathbf{E}$ with $\{A \circ\!\!-\!\!\circ B\}$.

Paths and cycles Let $\pi = \langle V_1, \ldots, V_n \rangle$ be a sequence of adjacent nodes. If each node only occurs once in π, then this is as a path from V_1 to V_n. If for any $\langle V_{i-1}, V_i, V_{i+1} \rangle$, $1 < i < n$, V_{i-1} and V_{i+1} are not adjacent, then this is an unshielded triple, and if every triple on p is unshielded, then π is an unshielded path. Let $\pi = \langle V_1, \ldots V_n \rangle$ be a path. If $V_{i-1} {*\!\!\rightarrow} V_i \leftarrow\!\!* V_{i+1}$ occurs on π, then V_i is a collider on π and otherwise it is a non-collider on π. If in addition $\langle V_{i-1}, V_i, V_{i+1} \rangle$ is an unshielded triple, then this is a *v-structure*. If on π $V_i \rightarrow V_{i+1}$ for every $1 \leq i < n$, then π is a directed path (from V_1 to V_n). If on π $V_i \circ\!\!-\!\!\circ V_{i+1}$ for every $1 \leq i < n$, then π is a non-directed path (from V_1 to V_n). A path $\langle V_1, \ldots V_n \rangle$ is possibly directed from V_1 to V_n if for every $1 \leq i < n$, the edge between V_i and V_{i+1} is not into V_i or out of V_{i+1}. When needed, we denote a possibly directed path from V_1 to V_n by $V_1 \dashrightarrow V_n$. A (non-) directed path from V_1 to V_n combined with a (non-) directed path from V_n to V_1 is a (non-) directed cycle. A directed path from V_1 to V_n combined with $V_1 \leftrightarrow V_n$ is an almost directed cycle. Let $\mathcal{G}' = (\mathbf{V}, \mathbf{E}')$ be a graph with same skeleton as $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ but possibly $\mathbf{E}' \neq \mathbf{E}$, then for a path π in $\mathcal{G}$ the corresponding path in $\mathcal{G}'$ is the path π' that consists of the same nodes as π. A path $\pi = \langle V_1, \ldots, V_n \rangle$ is an *inducing path* if every node V_i, $1 < i < n$, is a collider on π and an ancestor of V_1 or V_n.

Parents, ancestors and descendants For $\mathcal{G} = (\mathbf{E}, \mathbf{V})$ we define the *parents*, *ancestors*, *possible ancestors*, *descendants* and *possible descendants* of a node A in $\mathcal{G}$ as follows

$$
\begin{aligned}
\mathrm{pa}_{\mathcal{G}}(A) &= \{V \in \mathbf{V} \mid \{V \rightarrow A\} \in \mathbf{E}\} \\
\mathrm{an}_{\mathcal{G}}(A) &= \{V \in \mathbf{V} \mid \mathcal{G} \text{ contains a directed path from } V \text{ to } A\} \\
\mathrm{possan}_{\mathcal{G}}(A) &= \{V \in \mathbf{V} \mid \mathcal{G} \text{ contains a possibly directed path from } V \text{ to } A\} \\
\mathrm{de}_{\mathcal{G}}(A) &= \{V \in \mathbf{V} \mid \mathcal{G} \text{ contains a directed path from } A \text{ to } V\} \\
\mathrm{possde}_{\mathcal{G}}(A) &= \{V \in \mathbf{V} \mid \mathcal{G} \text{ contains a possibly directed path from } A \text{ to } V\}
\end{aligned}
$$

For a set $\mathbf{A}$ we define $\mathrm{an}_{\mathcal{G}}(\mathbf{A}) = \{V \in \mathbf{V} \mid \mathcal{G} \text{ contains a directed path from } V \text{ to some } A \in \mathbf{A}\}$.

Graph types A directed graph that does not contain any directed cycles is called a directed acyclic graph (DAG). A mixed graph (MG) contains both directed and bidirected edges. A graph that contains no directed cycles or almost directed cycles is ancestral. An ancestral mixed graph that does not contain an inducing path between any pair of non-adjacent nodes is a maximal ancestral graph (MAG). A partial mixed graph (PMG) contains directed, bidirected, non-directed and partially directed edges.

A.2. Independence models and Markov equivalence

Definition 14 (m-connecting) *Let $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ be a MAG, $\pi = \langle V_1, \ldots, V_n \rangle$ be a path in $\mathcal{G}$, and let $\mathbf{S} \subseteq \mathbf{V} \backslash \{V_1, V_n\}$. If*

(i) for every collider V on π, either V or a descendant of V is in $\mathbf{S}$, and

(ii) no non-collider on π is in $\mathbf{S}$

then π is m-connecting given $\mathbf{S}$.

For a MAG $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, if there exists a path from a $A \in \mathbf{V}$ to $B \in \mathbf{V}$ where $B \neq A$ that is m-connecting given $\mathbf{S}$, we say that A and B are m-connected given $\mathbf{S}$. If no such path exists,

we say that A and B are m-separated given $\mathbf{S}$ in $\mathcal{G}$ and denote this by $A \perp_{\mathcal{G}} B \mid \mathbf{S}$. We define an independence model $\mathcal{I}(\mathcal{G})$ induced by $\mathcal{G}$ as the collection of all m-separations in $\mathcal{G}$:

$$(A \perp_{\mathcal{G}} B \mid \mathbf{S}) \in \mathcal{I}(\mathcal{G}) \Leftrightarrow A \text{ and } B \text{ are m-separated by } \mathbf{S} \text{ in } \mathcal{G}$$

The same definitions are valid for DAGs, and here m-separation (m-connection) is equivalent to d-separation (d-connection). Occasionally, we omit the "m" or "d" and simply refer to separation in a graph.

Definition 15 ($\mathrm{dsep}_{\mathcal{G}}(A, B)$ **(Spirtes et al., 1999))** *Let $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ be a MAG and let A, B, V be distinct nodes in $\mathbf{V}$. Then $V \in \mathrm{dsep}_{\mathcal{G}}(A, B)$ if and only if there is a path $\langle A = V_0, V_1, \ldots, V_{n+1} = V \rangle$ for which*

(i) $V_i \in \mathrm{an}_{\mathcal{G}}(\{A, B\})$ for every $i \in \{1, \ldots n\}$, and

(ii) $V_i \leftrightarrow V_{i+1}$ for every $i \in \{1, \ldots n-1\}$.

Lemma 16 (Lemma 12 in Spirtes et al. (1999)) *Let $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ be a MAG and $X, Y \in \mathbf{V}$. There exists a set $\mathbf{V}' \subseteq \mathbf{V} \backslash \{A, B\}$ such that A and B are m-separated by $\mathbf{V}'$ if and only if they are m-separated by $\mathrm{dsep}_{\mathcal{G}}(A, B)$.*

Two MAGs (or DAGs) $\mathcal{G}_1$ and $\mathcal{G}_2$ are Markov equivalent if they induce the same independence model: $\mathcal{I}(\mathcal{G}_1) = \mathcal{I}(\mathcal{G}_2)$. A (Markov) equivalence class is a set of Markov equivalent MAGs (or DAGs). A partial ancestral graph (PAG) represents an equivalence class of MAGs. A PAG $\mathcal{P}$ represents an equivalence class of MAGs $[\mathcal{P}]$ such that every arrowhead and every tail present in $\mathcal{P}$ is also present in every $\mathcal{M} \in [\mathcal{P}]$, and we assume that they are maximally informative in the sense that for every circle mark, there is at least one $\mathcal{M}_1 \in [\mathcal{P}]$ where this is an arrowhead, and one $\mathcal{M}_2 \in [\mathcal{P}]$ where this is a tail.

In addition to Markov equivalence, a class of graphs might share more information, and constitute a restricted equivalence class. In general, we can represent the additional background knowledge using a PMG, which may not be ancestral and maximally informative. Given a sufficient set of orientation rules we may represent a restricted equivalence class using a maximally informative PAG. The orientation rules needed depends on the type of background knowledge.

A.3. The FCI algorithm

We assume that the nodes $\mathbf{V}$ in a graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ represent a set of random variables $\mathbf{X}_{\mathbf{V}}$. Throughout, we assume that the *global Markov property* and *faithfulness* hold, which allows us to learn an equivalence class of causal graphs from data. Let $\mathcal{D} = (\mathbf{V}, \mathbf{E})$ be a DAG and let P be a probability distribution over $\mathbf{X}_{\mathbf{V}}$. If P obeys the global Markov property and faithfulness with respect to $\mathcal{D}$ then for any distinct nodes $A, B \in \mathbf{V}$ and set $\mathbf{S} \subseteq \mathbf{V} \backslash \{A, B\}$

$$A \perp_{\mathcal{D}} B \mid \mathbf{S} \Leftrightarrow X_A \perp\!\!\!\perp X_B \mid \mathbf{X}_{\mathbf{S}} \tag{1}$$

where $\perp\!\!\!\perp$ denotes (conditional) independence between random variables.

We can still utilise the global Markov property and faithfulness without causal sufficiency, since d-separation in a DAG corresponds to m-separation in a MAG that has been constructed by marginalising over the latent variables (see Richardson and Spirtes (2002) for details).

The FCI algorithm searches for possible separating sets in the following set:

Definition 17 ($\text{poss.dsep}_{\mathcal{G}}(A, B)$ **(Spirtes et al., 1999))** *Let $\mathcal{M} = (\mathbf{V}, \mathbf{E})$ be a PMG and A and B distinct nodes in $\mathbf{V}$. Then $V \in \mathbf{V}$ is in $\text{poss.dsep}_{\mathcal{G}}(A, B)$ if and only if there is a path $\pi = \langle A = V_1, \ldots, V_n = V \rangle$ between V and A that in $\mathcal{G}$ such that π is not directed, and for every subpath $\langle V_{j-1}, V_j, V_{j+1} \rangle$, $2 \leq j \leq n-1$, of π, either*

(a) V_j is a collider, or

(b) V_j is a definite non-collider and V_{j-1} and V_{j+1} are adjacent.

This set was refined by Colombo et al. (2012) in the following way:

$$\text{pds}_{\mathcal{G}}(A, B) = \{V \in \text{poss.dsep}_{\mathcal{G}}(A, B) \mid V \text{ is on a path between } A \text{ and } B \text{ in } \mathcal{G}\}$$

Definition 18 (Discriminating path (Zhang, 2008)) *Let $\mathcal{M} = (\mathbf{V}, \mathbf{E})$ be a MAG, and let $\pi = \langle A, \ldots, B, C, D \rangle$ be a path between $A \in \mathbf{V}$ and $D \in \mathbf{V}$ in $\mathcal{M}$. Then π is a discriminating path for C if*

(i) π consists of at least three edges, and

(ii) C is adjacent to D in π and is a non-endpoint node on π, and

(iii) $A \notin \text{adj}_{\mathcal{M}}(D)$, and every node between A and C on π is a collider and a parent of D.

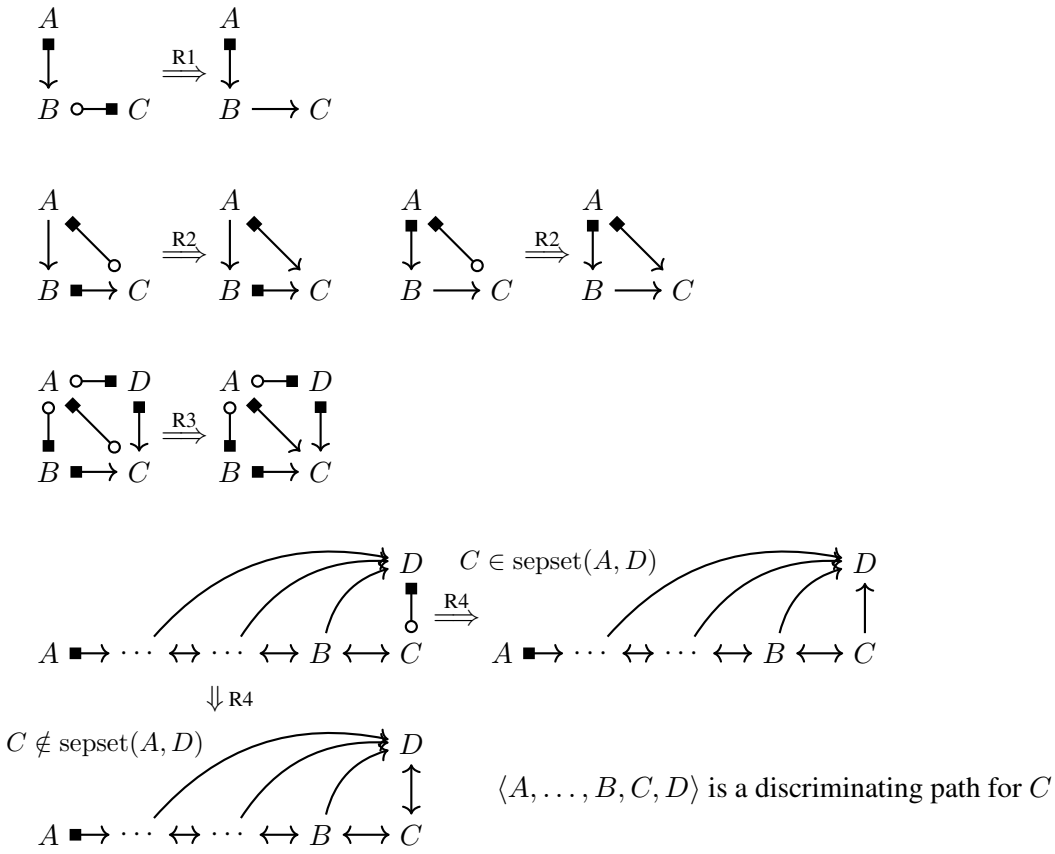

Figure 5: Orientation rules for the FCI algorithm (Spirtes et al., 1999; Zhang, 2008). Here ■ is a placeholder edge mark equivalent to $*$.

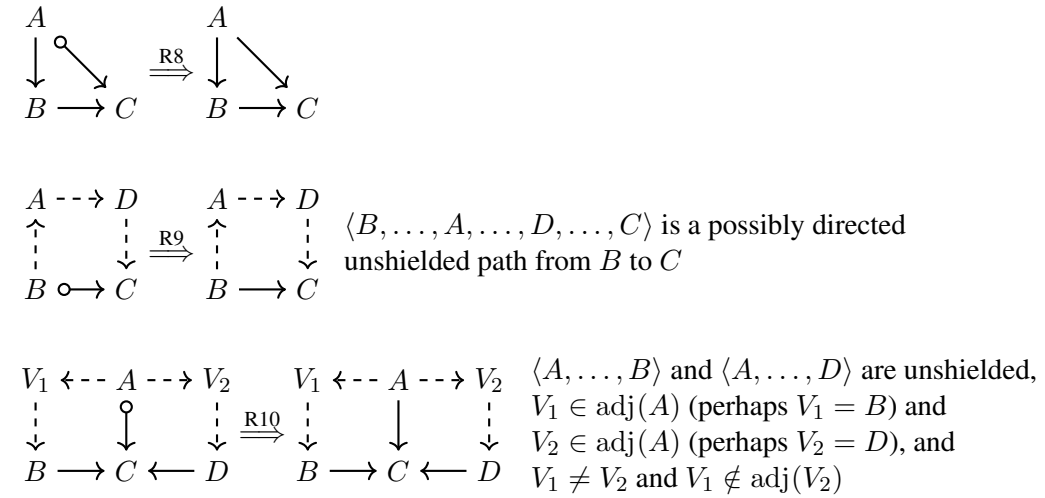

Figure 6: Orientation rules for the FCI algorithm (Zhang, 2008)

Appendix B. Pseudo-algorithms

By $\mathscr{P}(\cdot)$ we denote the power set. By $X \perp\!\!\!\perp Y \mid Z$ we indicate conditional independence between X and Y given Z. All independence facts are obtained through oracle knowledge.

Algorithm 2: The tiered FCI (tFCI) algorithm

input : Node set $\mathbf{V}$, tiered ordering τ, and an independence oracle over $\mathbf{X_V}$

output: partial mixed graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$

1 Let $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ be a complete graph with non-directed edges.
2 $k = 0$
3 **repeat**
4 **for** *each ordered pair of adjacent nodes* V_i *and* V_j **do**
5 **if** *there exists a* $\mathbf{S} \subseteq \mathrm{adj}_{\mathcal{G}}(V_i) \cap \mathrm{past}^{\tau}_{\mathbf{V}}(V_i) \backslash \{V_j\}$ *of size* k *with* $X_{V_i} \perp\!\!\!\perp X_{V_j} \mid \mathbf{X_S}$ **then**
6 remove $\{V_i \circ\!\!-\!\!\circ V_j\}$ from $\mathbf{E}$
7 add $\mathbf{S}$ to $\mathrm{sepset}(V_i, V_j)$
8 **end**
9 **end**
10 $k = k + 1$
11 **until** *there is no ordered pair of adjacent edges* V_i *and* V_j *with* $|\mathrm{adj}_{\mathcal{G}}(V_i) \cap \mathrm{past}^{\tau}_{\mathbf{V}}(V_i) \backslash \{V_j\}| \geq k$;
12 **for** *each unshielded triple* $\langle V_i, V_j, V_k \rangle$ **do**
13 **if** $\tau(V_j) > \max(\tau(V_i), \tau(V_k))$ *or* $V_j \notin \mathrm{sepset}(V_i, V_k)$ **then**
14 orient $V_i \ast\!\!-\!\!\ast V_j \ast\!\!-\!\!\ast V_k$ as $V_i \ast\!\!\rightarrow V_j \leftarrow\!\!\ast V_k$
15 **end**
16 **end**
17 **for** *each ordered pair of adjacent nodes* V_i *and* V_j **do**
18 **if** *there exists a* $\mathbf{S} \subseteq \mathrm{pds}_{\mathcal{G}}(V_i, V_j) \cap \mathrm{past}^{\tau}_{\mathbf{V}}(\{V_i, V_j\}) \backslash \{V_j\}$ *of size* k *with* $X_{V_i} \perp\!\!\!\perp X_{V_j} \mid \mathbf{X_S}$ **then**
19 remove $\{V_i \circ\!\!-\!\!\circ V_j\}$ from $\mathbf{E}$
20 add $\mathbf{S}$ to $\mathrm{sepset}(V_i, V_j)$
21 **end**
22 **end**
23 replace each edge $\{V_i \ast\!\!-\!\!\ast V_j\}$ in $\mathbf{E}$ with $\{V_i \circ\!\!-\!\!\circ V_j\}$
24 **for** *each unshielded triple* $\langle V_i, V_j, V_k \rangle$ **do**
25 **if** $\tau(V_j) > \max(\tau(V_i), \tau(V_k))$ *or* $V_j \notin \mathrm{sepset}(V_i, V_k)$ **then**
26 orient $V_i \ast\!\!-\!\!\ast V_j \ast\!\!-\!\!\ast V_k$ as $V_i \ast\!\!\rightarrow V_j \leftarrow\!\!\ast V_k$
27 **end**
28 **end**
29 `#omit lines 30-34 in the simple tFCI algorithm`
30 **for** *each ordered pair of adjacent nodes* V_i, V_j **do**
31 **if** $\tau(V_i) < \tau(V_j)$ **then**
32 orient $V_i \ast\!\!-\!\!\circ V_j$ as $V_i \ast\!\!\rightarrow V_j$
33 **end**
34 **end**
35 apply orientation rules R1-R4 (Figure 5) and R8-R10 (Figure 6) repeatedly to $\mathcal{G}$ until none applies

Algorithm 3: The tiered IOD (tIOD) algorithm

input : Node sets $\mathbf{V}_1, \ldots, \mathbf{V}_n$, tiered ordering τ, and independence oracles over $\mathbf{X}_{\mathbf{V}_1}, \ldots, \mathbf{X}_{\mathbf{V}_n}$

output : set of partial mixed graphs $\mathcal{G}$

Let $\mathcal{G} = (\mathbf{V}, \mathbf{E})$ be a complete graph with non-directed edges.
for *each* $l \in \{1, \ldots, n\}$ **do**
  Let $\mathcal{G}_l = (\mathbf{V}_l, \mathbf{E}_l)$ be a complete graph with non-directed edges.
  $k = 0$
  repeat
    for *each ordered pair of adjacent nodes* V_i *and* V_j **do**
      if *there exists a set* $\mathbf{S} \subseteq \mathrm{adj}_{\mathcal{G}_l}(V_i) \cap \mathrm{past}^{\tau}_{\mathbf{V}_l}(V_i)$ *of size* k *with* $X_{V_i} \perp\!\!\!\perp X_{V_j} \mid \mathbf{X}_{\mathbf{S}}$ **then**
        remove $\{V_i \circ\!\!-\!\!\circ V_j\}$ from $\mathbf{E}_l$ and from $\mathbf{E}$
        add $\mathbf{S}$ to $\mathrm{sepset}(\mathcal{G}_l, \{V_i, V_j\})$
      end
    end
    $k = k + 1$
  until *there is no ordered pair of adjacent edges* V_i *and* V_j *with* $|\mathrm{adj}_{\mathcal{G}_l}(V_i) \cap \mathrm{past}^{\tau}_{\mathbf{V}_l}(V_i) \backslash \{V_j\}| \geq k$;
  for *each unshielded triple* $\langle V_i, V_j, V_k \rangle$ **do**
    if $\tau(V_j) > \max(\tau(V_i), \tau(V_k))$ *or* $V_j \notin \mathrm{sepset}(\mathcal{G}_l, \{V_i, V_k\})$ **then**
      orient $V_i \ast\!\!-\!\!\ast V_j \ast\!\!-\!\!\ast V_k$ as $V_i \ast\!\!\rightarrow V_j \leftarrow\!\!\ast V_k$
    end
  end
  for *each ordered pair of adjacent nodes* V_i *and* V_j **do**
    if *there exists a set* $\mathbf{S} \subseteq \mathrm{pds}_{\mathcal{G}_l}(V_i, V_j) \cap \mathrm{past}^{\tau}_{\mathbf{V}_l}(\{V_i, V_j\})$ *with* $X_{V_i} \perp\!\!\!\perp X_{V_j} \mid \mathbf{X}_{\mathbf{S}}$ **then**
      remove $\{V_i \circ\!\!-\!\!\circ V_j\}$ from $\mathbf{E}_l$ and from $\mathbf{E}$
      add $\mathbf{S}$ to $\mathrm{sepset}(\mathcal{G}_l, \{V_i, V_j\})$
    else
      add $\langle \{V_i, V_k\}, \mathbf{V}_l \rangle$ to **InducingPaths**
    end
  end
  for *each unshielded triple* $\langle V_i, V_j, V_k \rangle$ *in* $\mathcal{G}_l$ **do**
    if $V_j \in \mathrm{adj}_{\mathcal{G}}(V_i) \cup \mathrm{adj}_{\mathcal{G}}(V_k)$, *and* $\tau(V_j) > \max(\tau(V_i), \tau(V_k))$ *or* $V_j \notin \mathrm{sepset}(\mathcal{G}_l, \{V_i, V_k\})$ **then**
      orient any $V_i \ast\!\!-\!\!\ast V_k$ as $V_i \ast\!\!\rightarrow V_k$, and any $V_k \ast\!\!-\!\!\ast V_j$ to $V_k \leftarrow\!\!\ast V_j$, in $\mathcal{G}$
    end
  end
end
for *each pair of adjacent nodes* V_i *and* V_j *in* $\mathcal{G}$ **do**
  if *for all* $l \in \{1, \ldots, n\}$
  $\{V_i, V_j\} \cup \big(\mathrm{adj}_{\mathcal{G}}(V_i) \cap \mathrm{past}^{\tau}_{\mathbf{V}}(V_i)\big) \cup \big((\mathrm{pds}_{\mathcal{G}}(V_i, V_j) \cup \mathrm{pds}_{\mathcal{G}}(V_j, V_i)) \cap \mathrm{past}^{\tau}_{\mathbf{V}}(\{V_i, V_j\})\big) \not\subset \mathbf{V}_l$ *and*
  $\{V_i, V_j\} \cup \big(\mathrm{adj}_{\mathcal{G}}(V_j) \cap \mathrm{past}^{\tau}_{\mathbf{V}}(V_j)\big) \cup \big((\mathrm{pds}_{\mathcal{G}}(V_i, V_j) \cup \mathrm{pds}_{\mathcal{G}}(V_j, V_i)) \cap \mathrm{past}^{\tau}_{\mathbf{V}}(\{V_i, V_j\})\big) \not\subset \mathbf{V}_l$ **then**
    add $\{V_i \ast\!\!-\!\!\ast V_j\}$ to **RemoveEdges**
  end
end

Algorithm 3: The tiered IOD (tIOD) algorithm (continued)

input : Node sets $\mathbf{V}_1, \ldots, \mathbf{V}_n$, tiered ordering τ, and independence oracles over $\mathbf{X}_{\mathbf{V}_1}, \ldots, \mathbf{X}_{\mathbf{V}_n}$
output : set of partial mixed graphs $\boldsymbol{\mathcal{G}}$

38 **for** *each* $\mathbf{E}' \in \mathscr{P}(\mathbf{RemoveEdge})$ **do**
39 **for** *each* $V_j \in \mathbf{V}$ *and pair of nodes* $V_i, V_k \in \mathrm{adj}_{\mathcal{G}_{\mathbf{E}\setminus\mathbf{E}'}}(V_j)$ **do**
40 **if** *the triple* $\langle V_i, V_j, V_k\rangle$ *can be oriented as a v-structure in* $\mathcal{G}_{\mathbf{E}\setminus\mathbf{E}'}$ *and for every* $l \in \{1, \ldots, n\}$ *either* $V_j \notin \mathbf{V}_l$ *or* $\mathrm{sepset}(\mathcal{G}_l, \{V_i, V_k\})$ *is undefined* **then**
41 **if** $\tau(V_j) > \max(\tau(V_i), \tau(V_j))$ **then**
42 orient $\langle V_i, V_j, V_k\rangle$ as a v-structure
43 **else**
44 add $\langle V_i, V_j, V_k\rangle$ to $\mathbf{OrientVstructure}$
45 **end**
46 **end**
47 **end**
48 **for** *each* $\mathbf{V}' \in \mathscr{P}(\mathbf{OrientVstructure})$ **do**
49 $\mathcal{G}^{\mathbf{V}'}_{\mathbf{E}\setminus\mathbf{E}'} = \mathcal{G}_{\mathbf{E}\setminus\mathbf{E}'}$
50 orient every triple in $\mathbf{V}'$ as a v-structure in $\mathcal{G}^{\mathbf{V}'}_{\mathbf{E}\setminus\mathbf{E}'}$
51 **repeat**
52 apply orientation rules R1-R4 and R8-R10
53 **if** *a discriminating path is found in R4* **then**
54 construct two graphs from $\mathcal{G}^{\mathbf{V}'}_{\mathbf{E}\setminus\mathbf{E}'}$ with each direction of the discriminated collider and continue orienting edges in both graphs
55 **end**
56 **until** *all graphs are closed under orientation rules R1-R4 and R8-R10*;
57 add all graphs to $\mathbf{Possible}\boldsymbol{\mathcal{G}}$
58 **for** *each* $\mathcal{G}' = (\mathbf{V}, \mathbf{E}'') \in \mathbf{Possible}\boldsymbol{\mathcal{G}}$ **do**
59 $\mathcal{M} = \mathcal{G}'$
60 $\boldsymbol{\mathcal{G}}' = \{\mathcal{G}'\}$
61 **for** *each* $\{V_i \circ\!\!\rightarrow V_j\} \in \mathbf{E}''$ **do**
62 orient $V_i \circ\!\!\rightarrow V_j$ as $V_i \rightarrow V_j$ in $\mathcal{M}$
63 **end**
64 orient $\mathcal{M}_U$ as a DAG with no new v-structures and replace every edge $\{V_i \circ\!\!-\!\!\circ V_j\}$ in $\mathcal{M}$ with the corresponding edge from $\mathcal{M}_U$
65 **if** *(i)* $\mathcal{M}$ *is a MAG, and (ii) each set in* sepset *corresponds to an independence in* $\mathcal{I}(\mathcal{M})$*, (iii) for every* l *for each* $\langle\{V_i, V_j\}, \mathbf{V}_l\rangle \in \mathbf{InducingPaths}$ *there is an inducing path between* V_i *and* V_j *with respect to* $\mathbf{V}\setminus\mathbf{V}_l$ *in* $\mathcal{M}$*, and (iv) if for every non-adjacent pair* A *and* B *in* $\mathcal{M}$ *there exists a set* $\mathbf{S}' \subseteq \mathrm{past}^{\tau}_{\mathbf{V}}(A, B)$ *such that* A *and* B *are m-separated by* $\mathbf{S}'$ *in* $\mathcal{M}$ **then**
66 `#omit lines 67-72 in the simple tIOD algorithm`
67 **for** *each ordered pair of adjacent nodes* V_i, V_j *in* $\mathcal{G}'$ **do**
68 **if** $\tau(V_i) < \tau(V_j)$ **then**
69 orient $V_i \ast\!\!\circ V_j$ as $V_i \ast\!\!\rightarrow V_j$
70 **end**
71 **end**
72 apply rules R1-R4 and R8-R10673 to $\mathcal{G}'$ until none applies. If a discriminating path is found then construct two graphs with each direction of the discriminated collider and continue orienting edges in both graphs. Let $\boldsymbol{\mathcal{G}}'$ be the set of these graphs.
73 add every $\mathcal{G}' \in \boldsymbol{\mathcal{G}}'$ to $\boldsymbol{\mathcal{G}}$
74 **end**
75 **end**
76 **end**
77 **end**

Remark 19 *The tIOD algorithm may also be made more efficient by, e.g., not considering nodes in* $\mathrm{adj}_{\mathcal{G}}(A)$ *and* $\mathrm{adj}_{\mathcal{G}}(B)$ *at lines 33-36 of Algorithm 3 when determining whether an edge* $A \ast\!\!-\!\!\ast B$ *is removable. However, we focus on the gain in efficiency by adding tiered background knowledge, and we choose to otherwise follow what is done in the original IOD algorithm.*

Appendix C. Number of graphs output in Example 1 and Example 2

$\mathbf{E}' \in \mathscr{P}(\mathbf{RemoveEdges})$	Possible v-struct.	Graphs considered	Graphs output
$\emptyset$	1	2	1
$\{A \circ\!\!-\!\!\circ D\}$	3	8	4
$\{A \circ\!\!-\!\!\circ B\}$	2	4	0
$\{A \circ\!\!-\!\!\circ C\}$	2	4	0
$\{B \circ\!\!-\!\!\circ D\}$	1	2	1
$\{C \circ\!\!-\!\!\circ D\}$	1	2	1
$\{\{A \circ\!\!-\!\!\circ D\}, \{A \circ\!\!-\!\!\circ B\}\}$	2	4	0
$\{\{A \circ\!\!-\!\!\circ D\}, \{A \circ\!\!-\!\!\circ C\}\}$	2	4	0
$\{\{A \circ\!\!-\!\!\circ D\}, \{B \circ\!\!-\!\!\circ D\}\}$	1	2	0
$\{\{A \circ\!\!-\!\!\circ D\}, \{C \circ\!\!-\!\!\circ D\}\}$	1	2	0
$\{\{A \circ\!\!-\!\!\circ B\}, \{A \circ\!\!-\!\!\circ C\}\}$	3	5	0
$\{\{A \circ\!\!-\!\!\circ B\}, \{B \circ\!\!-\!\!\circ D\}\}$	0	1	0
$\{\{A \circ\!\!-\!\!\circ B\}, \{C \circ\!\!-\!\!\circ D\}\}$	2	4	0
$\{\{A \circ\!\!-\!\!\circ C\}, \{B \circ\!\!-\!\!\circ D\}\}$	2	4	0
$\{\{A \circ\!\!-\!\!\circ C\}, \{C \circ\!\!-\!\!\circ D\}\}$	0	1	0
$\{\{B \circ\!\!-\!\!\circ D\}, \{C \circ\!\!-\!\!\circ D\}\}$	2	4	1
$\{\{A \circ\!\!-\!\!\circ D\}, \{A \circ\!\!-\!\!\circ B\}, \{A \circ\!\!-\!\!\circ C\}\}$	1	2	0
$\{\{A \circ\!\!-\!\!\circ D\}, \{A \circ\!\!-\!\!\circ B\}, \{B \circ\!\!-\!\!\circ D\}\}$	1	2	0
$\{\{A \circ\!\!-\!\!\circ D\}, \{A \circ\!\!-\!\!\circ B\}, \{C \circ\!\!-\!\!\circ D\}\}$	0	1	0
$\{\{A \circ\!\!-\!\!\circ D\}, \{A \circ\!\!-\!\!\circ C\}, \{B \circ\!\!-\!\!\circ D\}\}$	0	1	0
$\{\{A \circ\!\!-\!\!\circ D\}, \{A \circ\!\!-\!\!\circ C\}, \{C \circ\!\!-\!\!\circ D\}\}$	1	2	0
$\{\{A \circ\!\!-\!\!\circ D\}, \{B \circ\!\!-\!\!\circ D\}, \{C \circ\!\!-\!\!\circ D\}\}$	0	1	0
$\{\{A \circ\!\!-\!\!\circ B\}, \{A \circ\!\!-\!\!\circ C\}, \{B \circ\!\!-\!\!\circ D\}\}$	1	2	0
$\{\{A \circ\!\!-\!\!\circ B\}, \{A \circ\!\!-\!\!\circ C\}, \{C \circ\!\!-\!\!\circ D\}\}$	1	2	0
$\{\{A \circ\!\!-\!\!\circ B\}, \{B \circ\!\!-\!\!\circ D\}, \{C \circ\!\!-\!\!\circ D\}\}$	1	2	0
$\{\{A \circ\!\!-\!\!\circ C\}, \{B \circ\!\!-\!\!\circ D\}, \{C \circ\!\!-\!\!\circ D\}\}$	1	2	0
$\{\{A \circ\!\!-\!\!\circ B\}, \{A \circ\!\!-\!\!\circ C\}, \{A \circ\!\!-\!\!\circ D\}, \{B \circ\!\!-\!\!\circ D\}\}$	0	1	0
$\{\{A \circ\!\!-\!\!\circ C\}, \{A \circ\!\!-\!\!\circ D\}, \{B \circ\!\!-\!\!\circ D\}, \{C \circ\!\!-\!\!\circ D\}\}$	0	1	0
$\{\{A \circ\!\!-\!\!\circ D\}, \{B \circ\!\!-\!\!\circ D\}, \{C \circ\!\!-\!\!\circ D\}, \{A \circ\!\!-\!\!\circ B\}\}$	0	1	0
$\{\{B \circ\!\!-\!\!\circ D\}, \{C \circ\!\!-\!\!\circ D\}, \{A \circ\!\!-\!\!\circ B\}, \{A \circ\!\!-\!\!\circ C\}\}$	0	1	0
$\{\{C \circ\!\!-\!\!\circ D\}, \{A \circ\!\!-\!\!\circ B\}, \{A \circ\!\!-\!\!\circ C\}, \{A \circ\!\!-\!\!\circ B\}\}$	0	1	0
$\mathbf{E}$	0	1	0
Total	32	73	8

Table 1: Number of graphs output by the IOD algorithm in Example 1

$\mathbf{E}' \in \mathscr{P}(\mathbf{RemoveEdges})$	Possible v-struct.	Graphs visited	Graphs output
$\emptyset$	0	1	1
$\{A \circ\!\!-\!\!\circ D\}$	2	4	4
$\{B \circ\!\!-\!\!\circ D\}$	1	2	1
$\{C \circ\!\!-\!\!\circ D\}$	1	2	1
$\{\{A \circ\!\!-\!\!\circ D\}, \{B \circ\!\!-\!\!\circ D\}\}$	1	2	0
$\{\{A \circ\!\!-\!\!\circ D\}, \{C \circ\!\!-\!\!\circ D\}\}$	1	2	0
$\{\{B \circ\!\!-\!\!\circ D\}, \{C \circ\!\!-\!\!\circ D\}\}$	2	4	1
$\{\{A \circ\!\!-\!\!\circ D\}, \{B \circ\!\!-\!\!\circ D\}, \{C \circ\!\!-\!\!\circ D\}\}$	0	1	0
Total	8	18	8

Table 2: Number of graphs visited and output by the tIOD algorithm in Example 2

Appendix D. Proofs

Proof of Proposition 6 The "if" part is trivial, we show the "only if" part. Assume $\mathcal{G}$ is a DAG. Then A and B can be d-separated in $\mathcal{G}$ by some $\mathbf{S} \subset \mathbf{V}$ if and only if they can be d-separated by $\mathrm{pa}_{\mathcal{G}}(A)$ or $\mathrm{pa}_{\mathcal{G}}(B)$ (Verma and Pearl, 1990; Spirtes and Glymour, 1991). Since $\mathrm{pa}_{\mathcal{G}}(A) \subseteq \mathrm{past}^{\tau}_{\mathbf{V}}(A)$ and $\mathrm{pa}_{\mathcal{G}}(B) \subseteq \mathrm{past}^{\tau}_{\mathbf{V}}(B)$ for any τ consistent with $\mathcal{G}$, if A and B cannot be d-separated by any $\mathbf{S} \subseteq \mathrm{past}^{\tau}_{\mathbf{V}}(A)$ or $\mathbf{S}' \subseteq \mathrm{past}^{\tau}_{\mathbf{V}}(B)$, then they cannot be d-separated by $\mathrm{pa}_{\mathcal{G}}(A)$ or $\mathrm{pa}_{\mathcal{G}}(B)$, and then no subset of $\mathbf{V}$ d-separates them. Assume instead that $\mathcal{G}$ is a MAG. Then A and B can be m-separated in $\mathcal{G}$ by any $\mathbf{S} \subset \mathbf{V}$ if and only if they can be m-separated by $\mathrm{dsep}_{\mathcal{G}}(A, B)$ or $\mathrm{dsep}_{\mathcal{G}}(B, A)$ by Lemma 16. Recall that $\mathrm{dsep}_{\mathcal{G}}(A, B) \subseteq \mathrm{an}_{\mathcal{G}}(\{A, B\})$ (Definition 15). Note that $\mathrm{an}_{\mathcal{G}}(A) \subseteq \mathrm{past}^{\tau}_{\mathbf{V}}(A)$ and $\mathrm{an}_{\mathcal{G}}(B) \subseteq \mathrm{past}^{\tau}_{\mathbf{V}}(B)$ for any τ consistent with $\mathcal{G}$, and it follows that $\mathrm{dsep}_{\mathcal{G}}(A, B) \subseteq \mathrm{past}^{\tau}_{\mathbf{V}}(A) \cup \mathrm{past}^{\tau}_{\mathbf{V}}(B)$. Assume without loss of generality that $\tau(A) \leq \tau(B)$, then $\mathrm{past}^{\tau}_{\mathbf{V}}(A) \subseteq \mathrm{past}^{\tau}_{\mathbf{V}}(B)$ and then $\mathrm{dsep}_{\mathcal{G}}(A, B) \subseteq \mathrm{past}^{\tau}_{\mathbf{V}}(B)$: If there is no $\mathbf{S}' \subseteq \mathrm{past}^{\tau}_{\mathbf{V}}(B)$ that m-separates A and B, then they are not m-separated by $\mathrm{dsep}_{\mathcal{G}}(A, B)$, and then no subset of $\mathbf{V}$ m-separates them. The same holds for $\mathrm{dsep}_{\mathcal{G}}(B, A)$. ■

Lemma 20 *Let $\mathcal{M} = (\mathbf{V}, \mathbf{E})$ be a MAG and A and B distinct nodes in $\mathbf{V}$. Let $\mathcal{G}$ be the graph obtained at line 22 of the tFCI algorithm (Algorithm 2) when estimating the equivalence class of $\mathcal{M}$ using oracle knowledge. Then A and B are adjacent in $\mathcal{M}$ if and only if they are adjacent in $\mathcal{G}$.*

Proof First, we could have obtained the same graph at line 11 if we had run the algorithm without using background knowledge c.f. Proposition 6, since we have oracle knowledge of the conditional independences and correct background knowledge. Second, note that given oracle knowledge, any v-structure oriented at line 14 based on the tiered ordering would also have been oriented without background knowledge. Since we obtain the same graph at line 16 as we would have without background knowledge, by Lemma 13 in Spirtes et al. (1999) $\mathrm{dsep}_{\mathcal{M}}(A, B) \subseteq \mathrm{poss.dsep}_{\mathcal{G}}(A, B)$, and by Lemma 1.2 in Colombo et al. (2012), A and B can be m-separated by any set in $\mathcal{M}$ if and only if they are m-separated by a set of nodes that all lie on a path between A and B in $\mathcal{M}$, i.e. $\mathrm{dsep}_{\mathcal{M}}(A, B) \subseteq \mathrm{pds}_{\mathcal{G}}(A, B)$. By Proposition 6, A and B can be m-separated in $\mathcal{M}$ if and only if they are m-separated by a subset of $\mathrm{past}^{\tau}_{\mathbf{V}}(\{A, B\})$ for any τ consistent with $\mathcal{M}$. Hence,

$\text{dsep}_{\mathcal{M}}(A,B) \subseteq \text{pds}_{\mathcal{G}}(A,B) \cap \text{past}^{\tau}_{\mathbf{V}}(\{A,B\})$ and every potential separating set is checked. Hence, A and B are adjacent in $\mathcal{M}$ if and only if they are adjacent in $\mathcal{G}$. ■

Proof of Proposition 7 Let $\mathcal{M}$ be a MAG and let $\mathcal{P}$ be the PAG representing the equivalence class of $\mathcal{M}$. We show that if we use oracle knowledge of the conditional independencies implied by $\mathcal{M}$ as input, then the simple tFCI algorithm outputs $\mathcal{P}$. By Lemma 20, the skeleton obtained at line 22 is the skeleton of $\mathcal{M}$. Due to the background knowledge being correct, no v-structure, that could not have been found from the conditional independencies, is oriented by background knowledge. Since we have oracle knowledge of the conditional independencies, every v-structure that cannot be oriented from background knowledge will be oriented by the conditional independencies. The orientation rules R1-R4 and R8-R10 are sound and complete given the correct adjacencies and v-structures (Spirtes et al., 1999; Zhang, 2008), hence the graph $\mathcal{G}$ output by the algorithm is the PAG $\mathcal{P}$ representing the equivalence class of $\mathcal{M}$. ■

Proof of Proposition 8 Let $\mathcal{M}$ be a MAG. We show that if we use oracle knowledge of the conditional independencies implied by $\mathcal{M}$ and correct background knowledge as input the full tFCI algorithm outputs a PMG $\mathcal{G}$ that does not contain any arrowhead or tail that is not in $\mathcal{M}$. By Proposition 7, the graph obtained at line 28 has the same skeleton and v-structures as $\mathcal{M}$. We only need to argue that the additional arrowheads and tails obtained in the full tFCI algorithm are correct. The arrowhead orientation in line 32 is sound due the background knowledge being correct. The orientation rules R1-R4 and R8-R10 are sound in the sense that they prevent the encoded independence model to change, any new cycles to occur, or any orientations that contradict the ancestral relations encoded (Spirtes et al., 1999; Zhang, 2008). Hence, they do not introduce any new information not already known from the conditional independencies and background knowledge, and the graph output after applying these rules does not contain any arrowheads or tails not in $\mathcal{M}$. ■

In order to prove Proposition 9, we need some extensions of the results in Tillman and Spirtes (2011), and we follow their proof strategy of their Theorem 5.1 and 5.2 to a large extend. Here, Corollary 21 (which follows from Proposition 7) is analogous to Theorem 7.2, Corollary 22 extends Corollary 7.1, Lemma 23 extends Lemma 7.1, Lemma 25 extends Lemma 7.6, and Lemma 26 extends Lemma 7.7.

Corollary 21 *Let* sepset *be constructed as in the simple tIOD algorithm (Algorithm 3), and for all* $i \in \{1,\dots,n\}$ *let* $\mathcal{M}_i = (\mathbf{V}_i, \mathbf{E}_i)$ *be a MAG. For every* $i \in \{1,\dots,n\}$ *if for all* $A, B \in \mathbf{V}_i$ *such that* $\text{sepset}(\mathcal{G}_i, \{A,B\})$ *is defined* $A \perp_{\mathcal{M}_i} B \mid \text{sepset}(\mathcal{G}_i, \{A,B\})$*, then for all* $A, B \in \mathbf{V}_i$ *and* $\mathbf{S} \subseteq \mathbf{V}_i \backslash \{A,B\}$*:* $X_A \perp\!\!\!\perp X_B \mid \mathbf{X_S} \Rightarrow A \perp_{\mathcal{M}_i} B \mid \mathbf{S}$

From this we have the following result.

Corollary 22 *Let* sepset *be constructed as in the simple tIOD algorithm (Algorithm 3), and let* $\mathcal{M} = (\mathbf{V}, \mathbf{E})$ *be a MAG. Then for all* $i \in \{1,\dots,n\}$ *if for all* $A, B \in \mathbf{V}_i$ *such that* $\text{sepset}(\mathcal{G}_i, \{A,B\})$ *is defined,* $A \perp_{\mathcal{M}} B \mid \text{sepset}(\mathcal{G}_i, \{A,B\})$*, then for all* $A, B \in \mathbf{V}_i$ *and* $\mathbf{S} \subseteq \mathbf{V}_i \backslash \{A,B\}$*:* $X_A \perp\!\!\!\perp X_B \mid \mathbf{X_S} \Rightarrow A \perp_{\mathcal{M}} B \mid \mathbf{S}$

Proof This proof is identical to the proof of Corollary 7.1 in Tillman and Spirtes (2011). ■

Lemma 23 *Let $\mathcal{G}'$ be a graph that is added to $\boldsymbol{\mathcal{G}}$ at line 73 in the simple tIOD algorithm (Algorithm 3), and let $\mathcal{M}$ be the graph obtained at line 64. Then $\mathcal{M}$ is a MAG and for all $i \in \{1, \ldots, n\}$, $A, B \in \mathbf{V}_i$ and $\mathbf{S} \subseteq \mathbf{V}_i \backslash \{A, B\}$ $X_A \perp\!\!\!\perp X_B \mid \mathbf{X_S} \Leftrightarrow A \perp_{\mathcal{M}} B \mid \mathbf{S}$.*

Proof This proof directly follows from the proof of Lemma 7.1 in Tillman and Spirtes (2011). $\mathcal{M}$ is a MAG by condition (i) at line 65. By condition (ii) at line 65 and Corollary 22, $X_A \perp\!\!\!\perp X_B \mid \mathbf{X_S} \Rightarrow A \perp_{\mathcal{M}} B \mid \mathbf{S}$. Then, by condition (iii) at line 65 and Theorem 3.2 in Tillman and Spirtes (2011) it follows that $X_A \perp\!\!\!\perp X_B \mid \mathbf{X_S} \Leftrightarrow A \perp_{\mathcal{M}} B \mid \mathbf{S}$. ■

Lemma 24 *Let $\mathcal{G}_1, \ldots, \mathcal{G}_n$, $\mathcal{G}$ and* **InducingPaths** *be the graphs and sets of dependent pairs obtained at line 32 of the oracle version of the simple tIOD algorithm (Algorithm 3). Then these are identical to the graphs $\mathcal{G}_1, \ldots, \mathcal{G}_n$, $\mathcal{G}$ and and sets of dependent pairs* **InducingPaths** *obtained after line 30 of Algorithm 2 in Tillman and Spirtes (2011) (the IOD algorithm).*

Proof We refer to Algorithm 3 as the tIOD (algorithm) and Algorithm 2 in Tillman and Spirtes (2011) as the IOD (algorithm).

Consider a fixed $i \in \{1, \ldots, n\}$, let $A, B \in \mathbf{V}_i$ and let $\mathrm{pds}_{\mathcal{G}_i}(A, B)^\tau$ and $\mathrm{pds}_{\mathcal{G}_i}(B, A)^\tau$ be sets of nodes considered at line 20 of the tIOD algorithm. Let $\mathcal{M}_i$ be the true MAG over V_i. Then, A and B are m-separated in $\mathcal{M}_i$ if and only if they are m-separated by $\mathrm{dsep}_{\mathcal{M}_i}(A, B)$ or $\mathrm{dsep}_{\mathcal{M}_i}(B, A)$ (Definition 15) by Lemma 16. Since $\mathrm{dsep}_{\mathcal{M}_i}(A, B) \subseteq \mathrm{pds}_{\mathcal{G}_i}(A, B)^\tau \cap \mathrm{past}^\tau_{\mathbf{V}}(A, B)$ and $\mathrm{dsep}_{\mathcal{M}_i}(B, A) \subseteq \mathrm{pds}_{\mathcal{G}_i}(B, A)^\tau \cap \mathrm{past}^\tau_{\mathbf{V}}(A, B)$ (c.f. proof of Lemma 20), the correct skeleton is obtained a line 26. Let $\mathrm{pds}_{\mathcal{G}_i}(A, B)$ and $\mathrm{pds}_{\mathcal{G}_i}(B, A)$ be sets of nodes considered at line 18 of the IOD algorithm. Then $\mathrm{pds}_{\mathcal{G}_i}(A, B)^\tau \cap \mathrm{past}^\tau_{\mathbf{V}}(A, B) \subseteq \mathrm{pds}_{\mathcal{G}_i}(A, B)$ and $\mathrm{pds}_{\mathcal{G}_i}(B, A)^\tau \cap \mathrm{past}^\tau_{\mathbf{V}}(A, B) \subseteq \mathrm{pds}_{\mathcal{G}_i}(B, A)$. Hence, the tIOD and IOD obtain the same skeleton of $\mathcal{G}_i$, and the same **InducingPaths** for V_i. Since this must be the case for every i, the two algorithms also obtain the same **InducingPaths** and the same skeleton of $\mathcal{G}$ after line 26 of the tIOD algorithm and line 24 of the IOD algorithm.

Next, the algorithms differ at lines 27-32 of the tIOD (line 25-29 in the IOD). In the tIOD, some arrowheads are oriented based on the tiered ordering, and some are oriented based on the conditional independencies. However, since the tiered background knowledge is assumed to be correct, any orientation by background knowledge could have been oriented solely by oracle knowledge of the conditional independencies. Here, the IOD orients arrowheads only based on conditional independencies. However, under oracle knowledge, both algorithms use the same conditional independencies, and since every $\mathcal{G}_i$ has the same skeleton in both algorithms, they orient the same arrowheads in $\mathcal{G}$. ■

Lemma 25 *Let $\mathcal{M}$ be a MAG over $\mathbf{V} = \mathbf{V}_1 \cup \ldots \cup \mathbf{V}_n$ such that for all $i \in \{1, \ldots, n\}$ $A, B \in \mathbf{V}_i$ and $\mathbf{S} \subseteq \mathbf{V}_i \backslash \{A, B\}$ $A \perp_{\mathcal{M}} B \mid \mathbf{S} \Leftrightarrow X_A \perp\!\!\!\perp X_B \mid \mathbf{X_S}$, then some graph considered at line 38 of Algorithm 3 has the same skeleton as $\mathcal{M}$.*

Proof Let $\mathcal{G}$ be the graph considered at line 33 of Algorithm 3. By Lemma 24 it follows from Lemma 7.4 in Tillman and Spirtes (2011) that $\mathcal{G}$ contains a superset of the adjacencies in $\mathcal{M}$. Let A and B be a pair of nodes that are adjacent in $\mathcal{G}$ but not in $\mathcal{M}$. Then, by maximality of $\mathcal{M}$, there is some set $\mathbf{S} \subseteq \mathbf{V} \backslash \{A, B\}$ that m-separates A and B in $\mathcal{M}$. It then follows from Corollary 22

that for all i, no such $\mathbf{S}$ is a subset of $\mathbf{V}_i$. One such $\mathbf{S}$ is in $\text{adj}_{\mathcal{M}}(A) \cap \text{past}^{\tau}_{\mathbf{V}}(A) \subseteq \text{adj}_{\mathcal{G}}(A) \cap \text{past}^{\tau}_{\mathbf{V}}(A)$ or $(\text{pds}_{\mathcal{M}}(A,B) \cup \text{pds}_{\mathcal{M}}(B,A)) \cap \text{past}^{\tau}_{\mathbf{V}}(\{A,B\}) \subseteq \big(\text{pds}_{\mathcal{G}}(A,B) \cup \text{pds}_{\mathcal{G}}(B,A)\big) \cap \text{past}^{\tau}_{\mathbf{V}}(\{A,B\})$, or $\mathbf{S}$ is in $\text{adj}_{\mathcal{M}}(B) \cap \text{past}^{\tau}_{\mathbf{V}}(B) \subseteq \text{adj}_{\mathcal{G}}(B) \cap \text{past}^{\tau}_{\mathbf{V}}(B)$ or $(\text{pds}_{\mathcal{M}}(A,B) \cup \text{pds}_{\mathcal{M}}(B,A)) \cap \text{past}^{\tau}_{\mathbf{V}}(\{A,B\}) \subseteq \big(\text{pds}_{\mathcal{G}}(A,B) \cup \text{pds}_{\mathcal{G}}(B,A)\big) \cap \text{past}^{\tau}_{\mathbf{V}}(\{A,B\})$. Hence, A and B are added to $\mathbf{RemoveEdge}$. Since all combinations of edge removals are considered, one graph must have the same skeleton as $\mathcal{M}$. ■

Lemma 26 *Let $\mathcal{M}$ be a MAG over $\mathbf{V}$ such that for all $i \in \{1,\dots,n\}$ $A, B \in \mathbf{V}_i$ and $\mathbf{S} \subseteq \mathbf{V}_i \backslash \{A,B\}$ $A \perp_{\mathcal{M}} B \mid \mathbf{S} \Leftrightarrow X_A \perp\!\!\!\perp X_B \mid \mathbf{X}_{\mathbf{S}}$, then some graph considered at line 49 of Algorithm 3 has the same v-structures as $\mathcal{M}$.*

Proof Let $\mathcal{G}$ be a graph constructed at line 38 of Algorithm 3 and assume that $\mathcal{G}$ has the same skeleton as $\mathcal{M}$ (such a $\mathcal{G}$ exists by Lemma 25). By Lemma 24 it follows from Lemma 7.5 in Tillman and Spirtes (2011) that $\mathcal{G}$ contains a subset of the v-structures in $\mathcal{M}$. Let $\langle A,B,C\rangle$ be an unshielded triple in $\mathcal{G}$ and $\mathcal{M}$, and assume that this is a v-structure in $\mathcal{M}$ but not in $\mathcal{G}$. If $\tau(B) > \max(\tau(A),\tau(C))$, then every graph considered at line 49 must have $\langle A,B,C\rangle$ oriented as a v-structure. If $\tau(B) \leq \max(\tau(A),\tau(C))$ and there were an i where $B \in \mathbf{V}_i$ and $\text{sepset}(\mathcal{G}_i, \{A,C\})$ were defined, then the v-structure would have been oriented in $\mathcal{G}$ (line 29 of Algorithm 3). Hence, $\langle A,B,C\rangle$ is added to $\mathbf{OrientVstructure}$. Since all combinations of possible v-structures are considered, one graph must have the same v-structures as $\mathcal{M}$. ■

Lemma 27 *Let $\mathcal{G}$ be a PAG. Then $\mathcal{G}$ is consistent with the tiered ordering τ if and only if some $\mathcal{M} \in [\mathcal{G}]$ satisfies criterion (iv) at line 65 in Algorithm 3.*

Proof $\mathcal{G}$ is consistent with τ if there exist an $\mathcal{M} \in [\mathcal{G}]$ that encodes the background knowledge implied by τ. Let $\mathcal{M} \in [\mathcal{G}]$.

"*If*": Let A and B non-adjacent in $\mathcal{M}$. If for every path π between A and B in $\mathcal{M}$ that only goes through nodes in $\mathbf{V} \backslash \text{past}^{\tau}_{\mathbf{V}}(\{A,B\})$, π is not m-connecting given $\text{past}^{\tau}_{\mathbf{V}}(\{A,B\})$, then the orientation of cross-tier edges in $\mathcal{G}$, which is the PAG of $\mathcal{M}$, will not construct any new v-structures.

"*Only if*": Let A and B be nodes in $\mathcal{M}$ and assume that (iv) at line 65 is not satisfied. Let π be a path between A and B that goes through nodes in $\mathbf{V} \backslash \text{past}^{\tau}_{\mathbf{V}}(\{A,B\})$. Assume that π is m-connecting given every $\mathbf{S} \subset \text{past}^{\tau}_{\mathbf{V}}(\{A,B\})$. Then, π will have either a directed edge that is into A, one that is into B, or both. However, τ implies that the edge between A and the next node on π cannot be a directed edge into A (similar for B). I.e. $\mathcal{M}$ does then not encode the tiered background knowledge implied by τ. Since such a path exists for any MAG that is Markov equivalent to $\mathcal{M}$, $\mathcal{G}$ is not consistent with τ. ■

Proof of Proposition 9 Let $\mathcal{G} \in \boldsymbol{\mathcal{G}}$. First, it follows directly from Lemma 7.3 in Tillman and Spirtes (2011) that $\mathcal{G}$ is a PAG. Second, $\mathcal{G}$ is only added to $\boldsymbol{\mathcal{G}}$ if it satisfies criterion (iv) at line 65. Hence, by Lemma 27 it follows that $\mathcal{G}$ is consistent with τ. We need to show the following:

Soundness: For all PAGs $\mathcal{G} \in \boldsymbol{\mathcal{G}}$ and for all $i \in \{1,\dots,n\}$ $A, B \in V_i$ and $\mathbf{S} \subseteq V_i \backslash \{A,B\}$ for all $\mathcal{M} \in [\mathcal{G}]$ $A \perp_{\mathcal{M}} B \mid \mathbf{S} \Leftrightarrow X_A \perp\!\!\!\perp X_B \mid \mathbf{X}_{\mathbf{S}}$.

Completeness: Let $\mathcal{M}$ be a MAG over $\mathbf{V}$ such that the PAG of $\mathcal{M}$ is consistent with τ and for all $i \in \{1, \ldots, n\}$ $A, B \in \mathbf{V}_i$ and $\mathbf{S} \subseteq \mathbf{V}_i \backslash \{A, B\}$ $A \perp_{\mathcal{M}} B \mid \mathbf{S} \Leftrightarrow X_A \perp\!\!\!\perp X_B \mid \mathbf{X_S}$, then $\mathcal{M} \in [\mathcal{G}]$ for some $\mathcal{G} \in \boldsymbol{\mathcal{G}}$.

Soundness: Let $\mathcal{M}$ be the graph obtained at line 62. By Theorem 2 in Zhang (2008) $\mathcal{M}$ is a MAG, and by Lemma 23 it holds that for all $i \in \{1, \ldots, n\}$, $A, B \in \mathbf{V}_i$ and $\mathbf{S} \subseteq \mathbf{V}_i \backslash \{A, B\}$: $X_A \perp\!\!\!\perp X_B \mid \mathbf{X_S} \Leftrightarrow A \perp_{\mathcal{M}} B \mid \mathbf{S}$. Since $\mathcal{M} \in [\mathcal{G}]$, for any other $\mathcal{M}' \in [\mathcal{G}]$ it also holds that $X_A \perp\!\!\!\perp X_B \mid \mathbf{X_S} \Leftrightarrow A \perp_{\mathcal{M}'} B \mid \mathbf{S}$.

Completeness: We follow the proof of Theorem 5.2 in Tillman and Spirtes (2011). By Lemma 26 there is a graph $\mathcal{G}'$ considered at line 50 of Algorithm 3 that has the same skeleton and v-structures as $\mathcal{M}$. Assume that $\mathcal{G}'$ also contains arrowheads or tails that are not in $\mathcal{M}$: Then this has been oriented at line 27. Following the proof of Theorem 5.2 in Tillman and Spirtes (2011), this must be an arrowhead that is into the collider in a v-structure in some MAG $\mathcal{M}_i = (\mathbf{V}_i, \mathbf{E}_i)$, for some $i \in \{1, \ldots, n\}$, such that $\mathcal{M}$ and $\mathcal{M}_i$ encode the same m-separations over $\mathbf{V}_i$, and by Corollary 7.2 in Tillman and Spirtes (2011), this arrowhead must then be contained in the PAG representing the equivalence class of $\mathcal{M}$. Hence, this orientation must also be in $\mathcal{M}$. By the soundness and completeness of the orientation rules R1-R4 and R8-R10 (Zhang, 2008), the PAG $\mathcal{G}$ of $\mathcal{M}$ is contained in $\mathbf{Possible}\boldsymbol{\mathcal{G}}$. By Theorem 2 in Zhang (2008), given $\mathcal{G}$, at line 62 we obtain a MAG that is Markov equivalent to $\mathcal{M}$. For this MAG, criteria (i) and (ii) at line 65 are satisfied, and by Theorem 4.2 in Richardson and Spirtes (2002) criterion (iii) is also satisfied. Hence, the PAG of $\mathcal{M}$ is added to $\boldsymbol{\mathcal{G}}$. ■

Proof of Proposition 10 Let $\mathcal{M}$ be a MAG over $\mathbf{V} = \mathbf{V}_1 \cup \ldots \cup \mathbf{V}_n$ that encodes τ. Consider the full tIOD algorithm using oracle knowledge of the marginal independence models over $\mathbf{V}_1, \ldots, \mathbf{V}_n$ and the background knowledge implied by τ. Let $\mathcal{G}'$ be the graph considered at line 59 and assume that conditions (i), (ii), (iii) and (iv) at line 65 are satisfied. Assume that $\mathcal{G}'$ is the PAG of $\mathcal{M}$ (by Proposition 9 some $\mathcal{G}'$ will be the PAG of $\mathcal{M}$). We will argue that the additional arrowheads and tails obtained in the full tIOD algorithm are correct: The arrowhead orientation at line 69 is sound since we assumed the tiered background knowledge to be correct. The orientation rules R1-R4 and R8-R10 are sound in the sense that they prevent the encoded independence model to change, any new cycles to occur, or any orientations that would contradict the ancestral relations encoded (Spirtes et al., 1999; Zhang, 2008). Hence, they do not introduce any new information not already known from the independence models and background knowledge. Moreover, for every possible orientation implied by R4 we include both graphs. Hence, at least one graph output encodes the independence model of $\mathcal{M}$ and does not contain any arrowheads and tails not in $\mathcal{M}$. ■

Proof of Proposition 11 We refer to Algorithm 3 as the tIOD (algorithm) and Algorithm 2 and 3 in Tillman and Spirtes (2011) as the IOD (algorithm).

In the first part of the algorithms (line 32 of the tIOD and line 30 of Algorithm 2 in Tillman and Spirtes (2011)), the IOD algorithm and simple tIOD obtain the same graph $\mathcal{G}$ over $\mathbf{V}$ (Lemma 24). The only other part later in the simple tIOD algorithm where tiered background knowledge is being used is when constructing $\mathbf{RemoveEdge}$ and $\mathbf{OrientVstructure}$. Like the IOD, the tIOD visits all graphs that can be constructed from $\mathcal{G}$ from all combinations of edge removals in $\mathbf{RemoveEdge}$ and v-structured orientations in $\mathbf{OrientVstructure}$. Let $\mathbf{RemoveEdge}$ and $\mathbf{OrientVstructure}$ be obtained in the IOD algorithm (Algorithm 3 in Tillman and Spirtes (2011)), and let $\mathbf{RemoveEdge}^{\tau}$ and $\mathbf{OrientVstructure}^{\tau}$ be obtained in the tIOD algorithm.

Then the tIOD algorithm visits fewer graphs than the IOD algorithm if and only if either $|\mathbf{RemoveEdge}| > |\mathbf{RemoveEdge}^\tau|$, $|\mathbf{OrientVstructure}| > |\mathbf{OrientVstructure}^\tau|$, or both $|\mathbf{RemoveEdge}| > |\mathbf{RemoveEdge}^\tau|$ and $|\mathbf{OrientVstructure}| > |\mathbf{OrientVstructure}^\tau|$. We show that this is equivalent to conditions (i) and (ii).

First, note that it must always be the case that $|\mathbf{RemoveEdge}^\tau| \leq |\mathbf{RemoveEdge}|$ since for any A, B

$$\{A, B\} \cup \big(\mathrm{adj}_\mathcal{G}(A) \cap \mathrm{past}^\tau_\mathbf{V}(A)\big) \cup \Big(\big(\mathrm{pds}_\mathcal{G}(A, B) \cup \mathrm{pds}_\mathcal{G}(B, A)\big) \cap \mathrm{past}^\tau_\mathbf{V}(A, B)\Big) \not\subset \mathbf{V}_i \Rightarrow \{A, B\} \cup \mathrm{adj}_\mathcal{G}(A) \cup \mathrm{pds}_\mathcal{G}(A, B) \cup \mathrm{pds}_\mathcal{G}(B, A) \not\subset \mathbf{V}_i$$

and

$$\{A, B\} \cup \big(\mathrm{adj}_\mathcal{G}(B) \cap \mathrm{past}^\tau_\mathbf{V}(B)\big) \cup \Big(\big(\mathrm{pds}_\mathcal{G}(A, B) \cup \mathrm{pds}_\mathcal{G}(B, A)\big) \cap \mathrm{past}^\tau_\mathbf{V}(A, B)\Big) \not\subset \mathbf{V}_i \Rightarrow \{A, B\} \cup \mathrm{adj}_\mathcal{G}(B) \cup \mathrm{pds}_\mathcal{G}(A, B) \cup \mathrm{pds}_\mathcal{G}(B, A) \not\subset \mathbf{V}_i$$

Moreover, it also always holds that $|\mathbf{OrientVstructure}^\tau| \leq |\mathbf{OrientVstructure}|$: If a triple is added to $\mathbf{OrientVstructure}^\tau$ it is also added to $\mathbf{OrientVstructure}$ since any unshielded triple obtained from $\mathbf{RemoveEdge}^\tau$ will also be obtained from $\mathbf{RemoveEdge}$.

"If:" Condition (i):: Since A and B have been measured together in at least one dataset, X_A and X_B are not marginally independent, so if the edge between A and B can be removed, it must be because there exists an $\mathbf{S} \subseteq \mathbf{V}$ that separates A and B. The IOD algorithm searches for such a set in $\mathrm{adj}_\mathcal{G}(A) \cup \mathrm{pds}_\mathcal{G}(A, B) \cup \mathrm{pds}_\mathcal{G}(B, A)$ and $\mathrm{adj}_\mathcal{G}(B) \cup \mathrm{pds}_\mathcal{G}(A, B) \cup \mathrm{pds}_\mathcal{G}(B, A)$. If for all i

$$\{A\} \cup \mathrm{adj}_\mathcal{G}(A) \cup \mathrm{pds}_\mathcal{G}(A, B) \cup \mathrm{pds}_\mathcal{G}(B, A) \not\subset \mathbf{V}_i \text{ and}$$
$$\{B\} \cup \mathrm{adj}_\mathcal{G}(B) \cup \mathrm{pds}_\mathcal{G}(A, B) \cup \mathrm{pds}_\mathcal{G}(B, A) \not\subset \mathbf{V}_i$$

then $A *\!\!-\!\!* B$ is added to $\mathbf{RemoveEdge}$. If for some i $A, B \in \mathbf{V}_i$ and

$$\big(\mathrm{pds}_\mathcal{G}(A, B) \cup \mathrm{pds}_\mathcal{G}(B, A)\big) \setminus \mathbf{V}_i \subseteq \mathbf{V} \setminus \mathrm{past}^\tau_\mathbf{V}(A, B) \text{ and } \mathrm{adj}_\mathcal{G}(A) \setminus \mathbf{V}_i \subseteq \mathbf{V} \setminus \mathrm{past}^\tau_\mathbf{V}(A)$$

then

$$\big(\mathrm{pds}_\mathcal{G}(A, B) \cup \mathrm{pds}_\mathcal{G}(B, A)\big) \cap \mathrm{past}^\tau_\mathbf{V}(A, B) \subseteq \mathbf{V}_i \text{ and } \mathrm{adj}_\mathcal{G}(A) \cap \mathrm{past}^\tau_\mathbf{V}(A) \subseteq \mathbf{V}_i$$

and it then follows that

$$\{A, B\} \cup \Big(\big(\mathrm{pds}_\mathcal{G}(A, B) \cup \mathrm{pds}_\mathcal{G}(B, A)\big) \cap \mathrm{past}^\tau_\mathbf{V}(A, B)\Big) \cup \big(\mathrm{adj}_\mathcal{G}(A) \cap \mathrm{past}^\tau_\mathbf{V}(A)\big) \subseteq \mathbf{V}_i$$

and $A *\!\!-\!\!* B$ is not added to $\mathbf{RemoveEdge}^\tau$ and $|\mathbf{RemoveEdge}| > |\mathbf{RemoveEdge}^\tau|$. The same holds for $\mathrm{adj}_\mathcal{G}(B) \setminus \mathbf{V}_i \subseteq \mathbf{V} \setminus \mathrm{past}^\tau_\mathbf{V}(B)$.

Condition (ii): Now, let $\langle A, C, B\rangle$ be an unshielded triple where for all i, either $C \notin \mathbf{V}_i$ or $\mathrm{sepset}(\mathcal{G}_i, \{A, B\})$ is undefined. Then $\langle A, C, B\rangle$ is added to $\mathbf{OrientVstructure}$. However, if $\tau(C) > \max(\tau(A), \tau(B))$, this is oriented as a v-structure in the tIOD algorithm (line 42 in Algorithm 3), and not added to $\mathbf{OrientVstructure}^\tau$ and $|\mathbf{OrientVstructure}| > |\mathbf{OrientVstructure}^\tau|$.

"Only if:" Assume that neither is satisfied. *Condition (i):* Assume that for all adjacent A and B for which it holds that for all i

$$\{A\} \cup \mathrm{adj}_\mathcal{G}(A) \cup \mathrm{pds}_\mathcal{G}(A, B) \cup \mathrm{pds}_\mathcal{G}(B, A) \not\subset \mathbf{V}_i \text{ and} \tag{2}$$

$$\{B\} \cup \mathrm{adj}_{\mathcal{G}}(B) \cup \mathrm{pds}_{\mathcal{G}}(A, B) \cup \mathrm{pds}_{\mathcal{G}}(B, A) \not\subset \mathbf{V}_i$$

it is either the case that i A and B are not in $\mathbf{V}_i$, or

$$\big(\mathrm{pds}_{\mathcal{G}}(A, B) \cup \mathrm{pds}_{\mathcal{G}}(B, A)\big) \setminus \mathbf{V}_i \not\subset \mathbf{V} \setminus \mathrm{past}^{\tau}_{\mathbf{V}}(A, B) \quad \text{or}$$

$$\mathrm{adj}_{\mathcal{G}}(A) \setminus \mathbf{V}_i \not\subset \mathbf{V} \setminus \mathrm{past}^{\tau}_{\mathbf{V}}(A) \;\text{ and }\; \mathrm{adj}_{\mathcal{G}}(B) \setminus \mathbf{V}_i \not\subset \mathbf{V} \setminus \mathrm{past}^{\tau}_{\mathbf{V}}(B)$$

then it follows that

$$\{A, B\} \cup \big(\mathrm{adj}_{\mathcal{G}}(A) \cap \mathrm{past}^{\tau}_{\mathbf{V}}(A)\big) \cup \big((\mathrm{pds}_{\mathcal{G}}(A, B) \cup \mathrm{pds}_{\mathcal{G}}(B, A)) \cap \mathrm{past}^{\tau}_{\mathbf{V}}(\{A, B\})\big) \not\subset \mathbf{V}_i$$

and

$$\{A, B\} \cup \big(\mathrm{adj}_{\mathcal{G}}(B) \cap \mathrm{past}^{\tau}_{\mathbf{V}}(B)\big) \cup \big((\mathrm{pds}_{\mathcal{G}}(A, A) \cup \mathrm{pds}_{\mathcal{G}}(B, A)) \cap \mathrm{past}^{\tau}_{\mathbf{V}}(\{A, B\})\big) \not\subset \mathbf{V}_i$$

Hence, for all pairs A, B where (2) is satisfied $A \ast\!\!-\!\!\ast B$ is then added to both $\mathbf{RemoveEdge}$ and $\mathbf{RemoveEdge}^{\tau}$, and $|\mathbf{RemoveEdge}| = |\mathbf{RemoveEdge}^{\tau}|$

Condition (ii): Assume that for all unshielded triples $\langle A, C, B\rangle$ where for all i, either $C \notin \mathbf{V}_i$ or $\mathrm{sepset}(\mathcal{G}_i, \{A, B\})$ is undefined we have that $\tau(C) \leq \max(\tau(A), \tau(B))$, then all of these triples are added to both $\mathbf{OrientVstructure}$ and $\mathbf{OrientVstructure}^{\tau}$, and $|\mathbf{OrientVstructure}| = |\mathbf{OrientVstructure}^{\tau}|$.

■