Relational Causal Discovery with Latent Confounders

Matteo Negro[*1] **Andrea Piras**[*1] **Ragib Ahsan**[2] **David Arbour**[3] **Elena Zheleva**[1]

[1]University of Illinois Chicago, Chicago
[2]Pinterest, Inc., San Francisco
[3]Adobe Research, San Francisco

Abstract

Estimating causal effects from real-world relational data can be challenging when the underlying causal model and potential confounders are unknown. While several causal discovery algorithms exist for learning causal models with latent confounders from data, they assume that the data is independent and identically distributed (i.i.d.) and are not well-suited for learning from relational data. Similarly, existing *relational* causal discovery algorithms assume causal sufficiency, which is unrealistic for many real-world datasets. To address this gap, we propose RelFCI, a sound and complete causal discovery algorithm for relational data with latent confounders. Our work builds upon the Fast Causal Inference (FCI) and Relational Causal Discovery (RCD) algorithms and it defines new graphical models, necessary to support causal discovery in relational domains. We also establish soundness and completeness guarantees for relational d-separation with latent confounders. We present experimental results demonstrating the effectiveness of RelFCI in identifying the correct causal structure in relational causal models with latent confounders.

1 INTRODUCTION

The goal of causal discovery is to reveal causal information by analyzing observational data. Most causal discovery algorithms assume that the data is independent and identically distributed (i.i.d.), and that the data generation is based on a directed acyclic model [Heinze-Deml et al., 2018]. However, many real-world data sources, including biological and social networks, do not meet the i.i.d. assumption and contain entities which interact with each other and exhibit causal dependencies among their attributes. To capture such dependencies and enable causal reasoning in relational data, more expressive classes of directed graphical models [Maier et al., 2014, Lee and Honavar, 2016a, Ahsan et al., 2022] and algorithms for relational causal discovery [Maier et al., 2013, Lee and Honavar, 2016a, 2020, Ahsan et al., 2023] have been developed over the past decade.

Existing relational causal discovery algorithms rely on the strong assumption of causal sufficiency, i.e., all common causes of observed variables have been measured and included in the data. However, this assumption rarely holds for real-world data where the presence of latent confounders can invalidate the causal discovery and causal effect estimation processes. This is especially true in relational domains where capturing latent confounders in causal models is key to separating homophily-based correlations from contagion [Shalizi and Thomas, 2011, Lee and Ogburn, 2021]. While multiple algorithms exist for causal discovery with latent confounders in i.i.d. data (e.g., Spirtes et al. [2000], Colombo et al. [2012]), none address relational data. To facilitate more realistic causal discovery in relational domains, it is necessary to formalize latent confounders in relational causal models and lift the assumption of causal sufficiency.

In this work, we introduce novel graphical models and a novel relational causal discovery algorithm, RelFCI, that can capture latent confounders in relational data. We build upon the representations and algorithms for Fast Causal Inference (FCI) [Spirtes et al., 2000] and Relational Causal Discovery (RCD) [Maier et al., 2013], neither of which is sufficient on its own. FCI performs causal discovery with latent confounders but does not address relational data, whereas RCD performs relational causal discovery through relational *d*-separation but assumes causal sufficiency. We introduce new relational graphical models, *Latent Relational Causal Models* (LRCMs), *Maximal Ancestral Abstract Ground Graphs* (MAAGGs), and *Partial Ancestral Abstract Ground Graphs* (PAAGGs), and provide a set of assumptions necessary for causal discovery with latent variables on relational causal models. These models address the unique challenges of re-

[*]These authors contributed equally to this work.

Proceedings of the 41st Conference on Uncertainty in Artificial Intelligence (UAI 2025), PMLR 244:3123–3154.

lational data, such as variable construction across relational paths and partial observation of entities. We then show that with these new models and under our specified assumptions, the rules of FCI, combined with the rules of RCD and applied to the PAAGGs, yield a sound and complete procedure for relational causal discovery. Specifically, we prove soundness and completeness guarantees of RelFCI up to a bounded hop threshold in the presence of latent variables. We demonstrate the algorithm's correctness on experimental datasets, comparing it to existing algorithms.

2 RELATED WORK

Related work falls broadly into two categories: causal discovery in the presence of latent variables and relational causal discovery. Several causal discovery methods support latent confounders, but only for propositional data. Spirtes et al. [2000] introduce FCI, a generalization of PC algorithm explicitly designed for acyclic causal models with latent confounders. Zhang [2008] augments FCI with an additional set of edge-orienting rules, providing completeness of the resulting algorithm. Mooij and Claassen [2020] show that FCI is sound and complete for cyclic models under σ-separation criteria.

Maier et al. [2014] considered d-separation semantics on relational causal models, using *abstract ground graphs*, a lifted representation. Maier et al. [2014] further provide soundness and completeness of m-separation, an analogue of d-separation on mixed graphs [Richardson and Spirtes, 2002], on abstract ground graphs. Maier et al. [2013] introduce RCD, a sound and complete algorithm for discovery on abstract ground graphs under the assumptions of d-faithfulness, sufficiency, and acyclicity. Lee and Honavar [2016a] develop a more efficient version of RCD, RCD-Light, that requires polynomial time and space to compute. Additionally, using a novel characterization of relational causal models under different path semantics, they present an alternate technique for causal discovery [Lee and Honavar, 2016b]. Our work can be seen as an extension of these works, which relaxes causal sufficiency in order to more closely mirror real-world cases [Rothenhäusler et al., 2015, Strobl, 2019].

3 BACKGROUND

We provide an overview of relational theory, which serves as the foundation for our proposed RelFCI algorithm and its proofs of correctness. We follow the theoretical definitions provided by Maier et al. [2014]. We also go over the theory underlying causal discovery using latent confounders and partial ancestral graphs for Bayesian networks as specified by Spirtes et al. [2000] for the FCI algorithm. Finally, we provide the set of assumptions used in this work for relational causal discovery with latent variables. Appendices A.1 and A.2 contain accompanying figures that illustrate the concepts presented in this section.

3.1 RELATIONAL DATA AND RELATIONAL CAUSAL MODELS

A *Relational Schema* $\mathcal{S} = (\mathcal{E}, \mathcal{R}, \mathcal{A}, card)$ is a collection of a set of entity types $\mathcal{E}$; a set of relationship types $\mathcal{R}$, where $\mathcal{R}_i = \langle E_1^i, ..., E_a^i \rangle \in \mathcal{R}$, with $E_j^i \in \mathcal{E}$ and a the arity of the relation; a set of attribute classes $\mathcal{A}(I)$ for each entity or relationship and a cardinality function *card*: $\mathcal{R} \times \mathcal{E} \to \{\text{ONE, MANY}\}$. As a running example, we will consider a schema with two entity types, USER (U) and POST (P), and the relationship between them, REACTS (R). USER has three attributes (U.Type, U.Activity and U.Sentiment), POST has two attributes (P.Engagement, P.Content), and REACTS has one attribute (R.Frequency).

Given a relational schema, a *Relational Variable* $[I_X ... I_Y].Y$ consists of a *Relational Path* $[I_X ... I_Y]$, an alternating sequence of connected entities and relations, and an attribute Y of the last class reached by said path. The first class I_X of this relational variable is called *perspective*. For example, from the described schema example, $[U, R, P].Engagement$ is a relational variable from the perspective USER, which captures the set of Engagements of all posts that a user reacts to.

A *Relational Dependency* $[I_X ... I_Y].Y \to [I_X].X$ is a pair of two relational variables with a common perspective. Relational paths allow us to model causal dependencies between attributes of different entities, e.g., $[P, R, U].Sentiment \to [P].Engagement$ indicates that the engagement of a post depends on the sentiment of the user reacting to that post. The dependency is called *canonical* if the path of the outcome variable (in the example, $[P].Engagement$) has a path of length 1. A *Relational Causal Model* $\mathcal{M}_\Theta(\mathcal{S}, \mathcal{D})$ is a set of relational dependencies $\mathcal{D}$ defined over schema $\mathcal{S}$, with Θ denoting the set of conditional probability distributions for each attribute $\mathcal{A}(I)$ of every class $I \in \mathcal{E} \cup \mathcal{R}$ over its parents. The arrow corresponds to a relational dependency. The example relational causal model in Figure 1 shows that the user's sentiment and the post's content influence the engagement of the post. A *Relational Skeleton* σ, is an instantiation of the schema for all entities, relationships, and attributes which follows the cardinality requirements specified by *card*. In other words, this is the data realization of the schema. For example, one relational skeleton could have two entities of type USER, Bob and Anna, and one entity of type POST, a food recipe, that Bob and Anna react to. We denote the set of all possible relational skeletons for a schema $\mathcal{S}$ as $\Sigma_\mathcal{S}$.

Each relational causal model $\mathcal{M}_\Theta$ and relational skeleton σ correspond to a *Ground Graph* $GG_{\mathcal{M}\sigma}$. The nodes in this graph are the attributes of all Entities and Relation instances in the skeleton σ, while the edges between instances of variables represent all dependencies in $\mathcal{M}$. Graphical examples of these representations can be seen in Appendix A.1. An *Abstract Ground Graph* $AGG_{\mathcal{MB}h}$, for the relational causal model $\mathcal{M}$, perspective $\mathcal{B}$ and hop-threshold h, is a graph that captures dependencies between relational variables that hold for all possible ground graphs $GG_{\mathcal{M}\sigma}$, with $\sigma \in \Sigma_{\mathcal{S}}$. Abstract ground graphs are defined for each perspective $\mathcal{B}$ and relational path length fixed to h.

They include one node for each relational variable defined over the given perspective, as well as additional *intersection variables*, which represent the potential overlap in instances between two relational variables that share the same attribute class. For instance, if the AGG contains $[U, R, P]$.*Engagement* and $[U, R, P, R, U, R, P]$.*Engagement*, two relational variables with different paths but the same attribute, an intersection node $[U, R, P] \cap [U, R, P, R, U, R, P]$.*Engagement* is added to capture their shared instances. These intersection nodes inherit all incoming and outgoing edges from the original variables, as a single ground random variable may participate in multiple relational variable instances. Including intersection variables is essential for accurately modeling implied dependencies and for correctly applying d-separation, since such shared participation can affect conditional independence reasoning.

$AGG_{\mathcal{MB}h}$ contains edges between relational variables if the instantiations of those relational variables contain a dependent pair in some ground graph. The edges are obtained using the `extend` method [Maier et al., 2014], which constructs relational paths from the current perspective to a dependency's source attribute. Formally, given a relational dependency $[I_Y, \ldots, I_Z].Y \rightarrow [I_X].X$, and a current perspective path $[I_B, \ldots, I_X]$, the method finds all valid pivot points between the reversed path $[I_X, \ldots, I_B]$ and the dependency path, and concatenates them at the pivot to generate new paths of the form $[I_B, \ldots, I_X, \ldots, I_Z]$. The process ensures that dependencies are appropriately lifted to the abstract ground graph, regardless of original perspective. Furthermore, intersection variables inherit the edges from both the variables in the pair.

A single dependency in $\mathcal{M}$, with the extend method, may support multiple edges in $AGG_{\mathcal{MB}h}$. Additionally, a single model $\mathcal{M}$ produces multiple AGGs, one for each perspective. A more complete description of AGGs components and the extend method are provided in Appendix A.1. Maier et al. [2014] showed that d-separation applied to abstract ground graphs (i.e., relational d-separation) allows the identification of conditional independences that hold across all ground graphs.

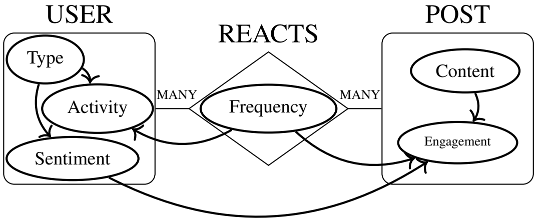


Figure 1: Example of a Relational Causal Model

3.2 PARTIAL ANCESTRAL GRAPH

The variables of a causal graph, $V \in G$, can be divided into three categories: observed (**O**), selection (**S**), and latent (**L**) variables, denoted as G(**O**,**S**,**L**). In this work, we focus on latent variables and assume there are no selection ones, i.e., $\boldsymbol{S} = \emptyset$. We denote **Cond** as the set of conditional independence relations among variables in **O**, and define the equivalence class of graphs that meets the conditional independence *O-Equiv*(**Cond**) as follows: for a graph G(**O**,**L**) belonging to *O-Equiv*(**Cond**), given three sets of variables **X**, **Y** and **Z**, G(**O**,**L**) entails that **X** $\perp\!\!\!\perp$ **Y** | **Z** if and only if **X** $\perp\!\!\!\perp$ **Y** | **Z** $\in$ **Cond**.

An *ancestral graph* [Zhang, 2008] is a causal graph that can be used to represent conditional independence and causal relations of a DAG with latent variables, using only the observed variables. A path p between any two vertices $X, Y \in$ **O** is called an *inducing path relative to* $\langle$**L**$\rangle$ if every non-endpoint vertex on p is either in **L** or a collider, and every collider in p is an ancestor vertex of either X or Y. A path is called primitive when **L** is empty. A *Maximal Ancestral Graph* (MAG) is an ancestral graph having no primitive inducing path between any two non-adjacent vertices.

The FCI algorithm learns a Markov equivalence class of a MAG called *Partial Ancestral Graph* (PAG), with edges ends having three possible marks, ∘, -, >, which indicate the following relationships: [1] $A \rightarrow B$ implies that A causes B; [2] $A \leftrightarrow B$ implies a common latent confounder between the two observed variables. An edge has an arrowhead > or tail - mark between two variables if and only if all DAGs in *O-Equiv*(**Cond**) share the same arrowhead (or tail) mark for those variables, i.e. the mark is *invariant*. On the other hand, if there exist two DAGs with a different edge mark between two variables, the PAG will contain a ∘ mark, i.e. the mark is *variant*. If every circle mark corresponds to an invariant in *O-Equiv*(**Cond**), the PAG is called *maximally informative* for the equivalence class. Examples of MAG and PAG are available in Appendix A.2.

3.3 ASSUMPTIONS FOR RELATIONAL CAUSAL DISCOVERY

In this subsection, we define and discuss some key assumptions used for causal discovery in relational data, including the maximum hop threshold, d-faithfulness, acyclicity, and absence of latent descendants for latent variables.

- Maximum Hop Threshold (h): The maximum hop threshold defines the largest permissible path length (or number of relational hops) between entities in a relational causal model that will be considered when constructing causal dependencies. Setting h limits the computational complexity and ensures that the discovered relationships are both interpretable and relevant. For instance, in a social network, $h = 2$ might capture direct friendships and friends-of-friends relationships while ignoring more distant connections.

- D-Faithfulness: D-faithfulness (Dependency-Faithfulness) posits that any conditional independence observed in the data is also represented in the underlying causal graph, and vice versa. This ensures that the causal relationships inferred from the data align with the observed statistical dependencies in the relational causal model.

- Acyclicity: Acyclicity mandates that the causal graph representing the relationships among variables and entities is a directed acyclic graph (DAG). This means there are no directed cycles in the relational causal model.

- Absence of latent descendants for latent variables: For this work, we assume that latent variables cannot be descendants of each other, i.e. all the parents and children of a latent variable are observed. This assumption is standard in constraint-based latent variable models [Evans, 2016]. Spirtes et al. [1995] note that conclusions about the equivalence class over observed variables remain valid regardless of the causal relations among latent variables.

4 RELATIONAL CAUSAL DISCOVERY WITH LATENT VARIABLES

In this section, we define latent variables in relational causal models, show why existing algorithms cannot perform relational causal discovery with latent variables, define the graphical models necessary for such discovery, and propose an algorithm for it. The full proofs of all theoretical findings in this paper are available in Appendix G.

4.1 LATENT VARIABLES IN RELATIONAL CAUSAL MODELS

To perform causal discovery with latent confounders, we first define them in the relational context. Considering the set of latent variables $\mathbf{L}$, we need to define what constitutes a latent relational variable $[I_X...I_Y].Y \in \mathbf{L}$. We assume that all entities in $\mathcal{E}$ and relationships in $\mathcal{R}$ are observed in the relational schema and the model and, consequently, in the variable's relational path. We define the set of latent attributes in a schema $\mathcal{S}$ as the set $\mathcal{A}_\mathbf{L}$. We can then look at the definition that follows:

Definition 1 (Latent Relational Variable). *A relational variable RV: $[I_X...I_Y].Y$ is considered latent, i.e., $RV \in \mathbf{L}$ if and only if its attribute class $Y \in \mathcal{A}(I_Y)$ is unobserved in the schema, meaning $Y \in \mathcal{A}_\mathbf{L}$.*

Consequently, a relational variable RV: $[I_W...I_Z].Z$ is observed, i.e., $RV \in \mathbf{O}$ if its attribute class $Z \in \mathcal{A}(I_Z)$ is observed, indicating that it is a member of the set of observed attributes classes in the schema, which we respectively define as $\mathcal{A}_\mathbf{O}$. A model's set of relational dependencies $\mathcal{D}$ is thus divided into two groups:

1. Set $\mathcal{D}_\mathbf{O}$ of observed dependencies $RV_1 \rightarrow RV_2$ defined only over observed relational variables i.e., $RV_1, RV_2 \in \mathbf{O}$;
2. Set $\mathcal{D}_\mathbf{L}$ of latent dependencies $RV_1 \rightarrow RV_2$ containing at least one latent relational variable i.e., $RV1 \in \mathbf{L} \vee RV_2 \in \mathbf{L}$;

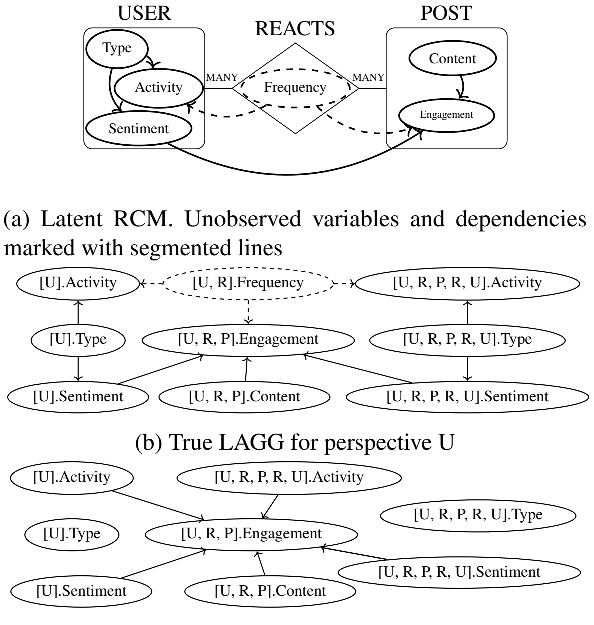
The modified relational causal model can now be defined as follows:

Definition 2 (Latent Relational Causal Model (LRCM)). *A relational causal model with latent variables $\mathcal{M}_{\Theta L}$ consists of two parts:*

1. *The structure $\mathcal{M}_L = (\mathcal{S}, \mathcal{D})$: the schema $\mathcal{S}$, containing a set of latent attributes $\mathcal{A}_\mathbf{L}$; the set of dependencies $\mathcal{D} = \mathcal{D}_\mathbf{O} \cup \mathcal{D}_\mathbf{L}$ defined over all relational variables;*
2. *Parameters Θ: a conditional probability distribution $P([I_j].X \mid parents([I_j].X))$ for all relational variables of the form $[I_j].X$ [Maier et al., 2014].*

An example LRCM can be seen in figure 2a. The latent AGG is constructed from LRCM $\mathcal{M}_L$, similarly to conventional relational causal models [Maier et al., 2014]. The construction divides the edges of the abstract ground graph into observed and unobserved edges, based on whether the underlying dependency from which the edge is yielded belongs to $\mathcal{D}_L$, i.e., is unobserved.

Definition 3 (Latent Abstract Ground Graph (LAGG)). *Given a relational causal model $\mathcal{M}_L$ a maximum hop threshold h, and a perspective $\mathcal{B}$, the $LAGG_{\mathcal{M}_L \mathcal{B} h}$ is the abstract*



(a) Latent RCM. Unobserved variables and dependencies marked with segmented lines

(b) True LAGG for perspective U

(c) RCD output AGG for perspective U

Figure 2: Counterexample that shows RCD does not produce the correct output AGG for LRCM with a faithful oracle

ground graph of the latent relational causal model. It contains both variables in the sets $\mathbf{O}$ *and* $\mathbf{L}$ *over perspective* $\mathcal{B}$*, plus intersection variables divided into observed intersection variables (both participating variables in the intersections are observed), and latent ones (i.e., at least one participating variable is latent). The set of edges* E *yielded from the dependencies in* $\mathcal{D}_{\mathbf{O}}$ *and* $\mathcal{D}_{\mathbf{L}}$*, using the* `extend` *method [Maier et al., 2014], is partitioned respectively into the set of observed* ($E_{\mathbf{O}}$) *and unobserved* ($E_{\mathbf{L}}$) *edges.*

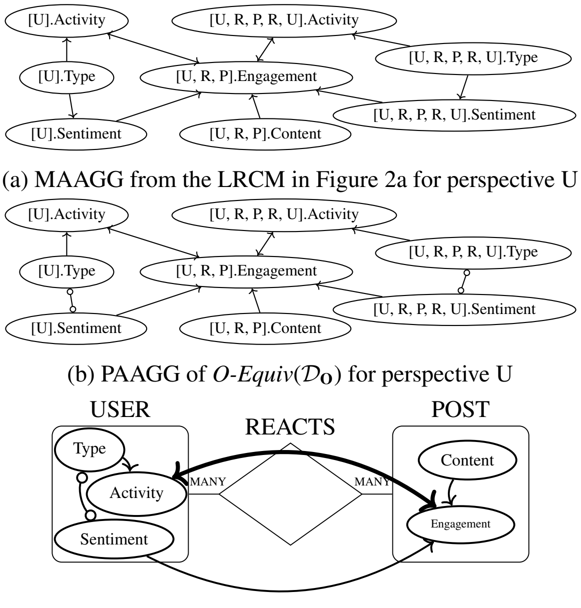
Consider the LRCM shown in figure 2a and a hop threshold $h = 2$. The Frequency attribute for REACT is unobserved. There are six relational dependencies in the model: 1) [U].Type → [U].Activity, 2) [U].Type → [U].Sentiment, 3) [P, R, U].Sentiment → [P].Engagement, 4) [P].Content → [P].Engagement, 5) [U, R].Frequency → [U].Activity, 6) [P, R].Frequency → [P].Engagement. The last two are unobserved dependencies in $\mathcal{D}_{\mathbf{L}}$. The respective LAGG for the described LRCM is shown in figure 2b.

4.2 LATENT RELATIONAL CAUSAL DISCOVERY

The RCD algorithm is the first sound and complete procedure that learns the dependencies of a relational causal model [Maier et al., 2013]. It works under several assumptions, described in detail in Appendix 3.3: maximum hop threshold h, d-faithfulness, acyclicity, and causal sufficiency. Causal sufficiency in particular implies that RCD was not originally designed for models with latent variables. Given that some forms of latent confounding can be detected via simple dependence tests [Arbour et al., 2016], it is natural to ask whether RCD is still sound and complete when the casual sufficiency assumption is lifted. To the best of our knowledge, no prior research has addressed this question in detail. The following counterexample shows that RCD is neither sound nor complete for relational causal discovery with latent variables.

Counterexample Figure 2c shows the output AGG produced by RCD using an oracle faithful to the underlying distribution. As we can see, the actual LAGG in figure 2b contains outgoing edges from [U].Type and from [U, R, P, R, U].Type; however, the output AGG (Fig. 2c lacks these edges. This indicates that RCD fails to identify the relational dependencies 1) and 2). Furthermore, without latent variables, RCD cannot capture and detect the presence of a latent confounder on the AGG using only directed edges. As seen in Figure 2, the fundamental problem that renders RCD neither sound nor complete for LRCM is the lack of identification of latent variables. This suggests that a more expressive representation than AGGs is required for the correct causal discovery in the presence of latent variables.

4.3 PARTIAL ANCESTRAL ABSTRACT GROUND GRAPHS



(a) MAAGG from the LRCM in Figure 2a for perspective U

(b) PAAGG of *O-Equiv*($\mathcal{D}_{\mathbf{O}}$) for perspective U

(c) PARM of *O-Equiv*($\mathcal{D}_{\mathbf{O}}$)

Figure 3: New representations to enable relational causal discovery in LRCMs

To enable causal discovery in latent relational causal models, we first need to define the necessary graphical models. Figure 3 shows an example of the models we introduce in this section. We will do so by considering an extension of the graphical models used in FCI to the relational setting, with a representation that expresses the underlying dependencies $\mathcal{D}$ over a set of observed variables $\mathbf{O}$ (i.e., $\mathcal{D}_{\mathbf{O}}$):

Definition 4 (Maximal Ancestral Abstract Ground Graph (MAAGG))**.** *Given a latent relational causal model $\mathcal{M}_{\boldsymbol{L}}(\mathcal{S},\mathcal{D})$ with a hop threshold h, any perspective $\mathcal{B}$, and the resulting Latent Abstract Ground Graph $LAGG_{\mathcal{M}_L\mathcal{B}h}$, the maximal ancestral abstract ground graph $MAAGG_{\mathcal{M}\mathcal{B}h'}$ is a graph, with a hop threshold $h' \geq h$ comprising:*

- *One node for each relational variable of the LAGG in $\boldsymbol{O}$ and the respective set of observed intersection variables;*
- *Three types of edges: $\rightarrow$, —, and $\leftrightarrow$, which are used to represent the underlying dependencies $\mathcal{D}_{\boldsymbol{O}}$.*

The MAAGG $\boldsymbol{\mathcal{G}}$ defined over the variables in $\mathbf{O}$, following the definition of Zhang [2008], probabilistically represents the respective LAGG defined over $\mathbf{O}$ and $\mathbf{L}$, specifically:

- Two variables $A, B \in \mathbf{O}$ are adjacent in $\boldsymbol{\mathcal{G}}$ if and only if there is an inducing path relative to $\langle L \rangle$ in the true LAGG;
- The orientation entails the same concept of non-causality and ancestry between two variables for MAGs and PAGs.

We now introduce a lemma that is necessary for proving the soundness and completeness of our proposed method:

Lemma 1. *Given a relational causal model structure $\mathcal{M}$ and perspective $\mathcal{B}$, an abstract ground graph $AGG_{\mathcal{M}\mathcal{B}}$ is ancestral if and only if all ground graphs $GG_{\mathcal{M}\sigma}$, with skeleton $\sigma \in \sum_{\mathcal{S}}$, are ancestral.*

Generally, the set underlying dependencies $\mathcal{D}_{\mathbf{O}}$ is not associated with a single MAAGG, but with the class of Markov equivalence defined as *O-Equiv*($\mathcal{D}_{\mathbf{O}}$). Therefore, we define another abstraction, based on PAGs and AGGs, that represents this equivalence class:

Definition 5 (Partial Ancestral Abstract Ground Graph (PAAGG))**.** *Given a relational causal model $\mathcal{M}_{\boldsymbol{L}}(\mathcal{S},\mathcal{D})$ with hop threshold h, the respective $MAAGG_{\mathcal{M}\mathcal{B}_{h'}}$, and its equivalence class O-Equiv($\mathcal{D}_{\boldsymbol{O}}$), and a perspective $\mathcal{B}$, the partial ancestral abstract ground graph $PAAGG_{\boldsymbol{\mathcal{M}}\mathcal{B}_{h'}}$ is a bidirected PAG, with hop threshold $h' \geq h$ comprising:*

- *The same set of nodes and adjacencies as the MAAGG;*
- *Edges containing three kinds of marks: $\circ$, —and $\rightarrow$, which are used to represent the variance and invariance of the equivalence class O-Equiv($\mathcal{D}_{\boldsymbol{O}}$).*

The following proposition provides a description of the soundness and completeness of the new representation:

Proposition 1. *Given a relational causal model $\mathcal{M}_{\boldsymbol{L}}(\mathcal{S},\mathcal{D})$ with hop threshold h, and its respective latent abstract ground graph G, the constructed MAAGG probabilistically and causally represents G and thus the underlying relational causal model. Furthermore, assuming a sound and complete procedure to construct the PAAGG G′, it correctly represents the Markov equivalence class of the produced MAAGG and, therefore, of G and the underlying model $\mathcal{M}_{\boldsymbol{L}}$.*

The equivalence class of MAAGGs represented by the PAAGG corresponds to multiple LRCMs that share the same set of dependencies over $\mathcal{D}_{\mathbf{O}}$. Thus, it is possible to define a new model from the PAAGG, which represents the equivalence class *O-Equiv*($\mathcal{D}_{\mathbf{O}}$):

Definition 6 (Partial Ancestral Relational Model (PARM))**.** *Given a LRCM $\mathcal{M}_{\boldsymbol{L}}(\mathcal{S},\mathcal{D})$ and its respective $PAAGG_{\boldsymbol{\mathcal{M}}\mathcal{B}_{h'}}$ for the equivalence class O-Equiv($\mathcal{D}_{\boldsymbol{O}}$), a partial ancestral model $\boldsymbol{\mathcal{M}}(\mathcal{S}_{\boldsymbol{O}},\mathcal{D}')$ is the relational causal model abstracted by the PAAGG that represents O-Equiv($\mathcal{D}_{\boldsymbol{O}}$). The PARM is defined over a relational schema containing only observed attribute classes $\mathcal{S}_{\boldsymbol{O}} = (\mathcal{E},\mathcal{R},\mathcal{A}_{\boldsymbol{O}},card)$ and a set $\mathcal{D}'$ of dependencies, which are used to represent the causality information for all models in O-Equiv($\mathcal{D}_{\boldsymbol{O}}$).*

The definitions of $MAAGG$ and $PAAGG$ allow a limit on the hop threshold higher than that of the underlying equivalence class of models. This is because the set of possible underlying dependencies with at most the same hop threshold would not capture paths that, in addition to the allowed threshold for observed variables, include unobserved variables. The higher hop threshold implied by definition 2 for $PAAGG$s is required to obtain an abstraction that correctly represents the presence of latent confounders in the underlying model. This is due to the presence of latent confounders (Definition 3) in a relational causal model and to the absence of latent parents and children for latent variables.

Proposition 2. *Given a latent relational causal model $\mathcal{M}_{\boldsymbol{L}}(\mathcal{S},\mathcal{D})$ with hop threshold h and its corresponding PARM $\boldsymbol{\mathcal{M}}$, the hop threshold h' of the $PAAGG_{\boldsymbol{\mathcal{M}}\mathcal{B}}$ for any perspective $\mathcal{B}$ can be at most* $2h$.

4.4 THE RELFCI ALGORITHM

In this section, we present the Relational Fast Causal Inference (RelFCI) algorithm, a sound and complete procedure for determining causal relationships from relational data when unobserved variables are present. RelFCI follows a three-step approach similar to the FCI algorithm for Bayesian networks (Spirtes 2013). RelFCI adapts the FCI procedure to relational causal models, similar to how RCD [Maier et al., 2013] does with the PC algorithm [Spirtes et al., 2000]. Given that FCI is an extension of PC, RelFCI follows the same orientation rules as RCD and also assumes a prior relational skeleton. However, it differs from RCD in two ways: [1] RelFCI uses partial ancestral abstract ground graphs, one for each perspective, as the underlying representation; [2] RelFCI applies seven additional rules from FCI to ensure soundness and completeness with latent relational data.

Algorithm 1 shows the high-level pseudocode for RelFCI, and Appendix C contains the complete algorithm pseudocode. RelFCI, like RCD, computes the set of potential dependencies in canonical form, limited by the $h' = 2h$ threshold. Starting from these dependencies, the algorithm constructs PAAGGs, one for each perspective, all with ∘ edge marks. The first step is to remove potential dependencies using conditional independence tests with conditioning sets of increasing size drawn from the collection of adjacencies of the two nodes considered. After deleting all possible edges, a set of unshielded triples is obtained. The second phase detects colliders while finding potential additional independence relationships between the triples' variables and potentially eliminating the respective edges. Even though RelFCI operates on different graphical models compared to FCI and RCD, it is straightforward to adapt their rules for RelFCI. The third step thus performs edge orientation by applying RBO, KNC, CA, and MK3 rules from RCD first, then rules R4 through R10 from FCI. A detailed description of these rules is provided in Appendix B. In contrast to the first step, the latter two differ from FCI because they apply the RBO rule from RCD and propagate each edge orientation to other PAAGGs. All steps are performed on all PAAGGs to accurately identify additional separation sets for each perspective.

Before demonstrating the soundness and completeness of RelFCI, we first clarify how the algorithm handles relational dependencies and edge orientations in PAAGGs. Since with ∘ marks RelFCI produces an equivalence class rather than a single causal model, certain underlying dependencies remain ambiguous. To address this, we distinguish between *required dependencies*, which must be oriented in a specific direction to respect the PAAGG orientation, and *possible dependencies*, which may have alternative orientations while remaining consistent with the learned PAAGG. With this new distinction, it is then possible to define the propagation of edges orientation across all PAAGGs for every perspective in a given LRCM, following a similar approach to the one described in RCD Maier et al. [2013] for regular AGGs. A detailed explanation of these aspects is provided in Appendices D and E.

4.5 SOUNDNESS

Maier et al. [2013] prove the soundness of CD, KNC, CA, MR3, and the new RBO rule using a proof derived from the soundness definition presented in Meek [1995]. Thus, we will focus on the soundness of the remaining rules R4-R10 adapted from FCI [Zhang, 2008].

Algorithm 1 RelFCI algorithm

```
Input: schema, oracle, threshold
Output: Dependencies
$PDs \leftarrow$ get potential dependencies from the base schema with 2*threshold
$PAAGGs \leftarrow$ construct PAAGGs from set of potential dependencies $PDs$
$S \leftarrow \{\}$
$PAAGGs, S, U \leftarrow$ find all independent variables in the graphs, storing separating sets and unshielded triples

$PAAGGS, S \leftarrow$ orient v-structures using CD, starting from unshielded triples in $U$
$PAAGGs, S \leftarrow$ orient PAAGGs edges using RCD and FCI rules
$Deps \leftarrow$ retrieve underlying dependencies from the edges of oriented PAAGGs
return Deps
```

Theorem 1. *Let G be the partially oriented PAAGG from perspective B, with the correct adjacencies, unshielded colliders correctly orientated through CD and RBO, and as many edges as possible oriented through KNC, CA, MR3, and the purely common cause of RBO. Then, FCI's rules R4-R10 and the orientation propagations are sound.*

The proof is an extension of those presented by Spirtes et al. [2000] for rule R4 and Zhang [2008] for rules R5-R10.

4.6 COMPLETENESS

A set of orientation rules is called complete if it generates a maximally informative graph. In PAAGGs, each circle corresponds to a variation mark in the equivalence class *O-Equiv*($\mathcal{D}_{\mathbf{O}}$) (modified from Zhang [2008]). The rules employed in FCI can be divided into two groups based on their function: those used to identify arrowhead invariants (CD, KNC, CA, MR3, and R4) and those used to identify tail invariants (R5-R10). According to Ali [2005], the first set of rules covers all invariant arrowheads. Lemma 2 shows that PAAGGs have similar arrowhead completeness, which can be used to prove overall rule completeness.

Lemma 2. *Let G be a partially oriented PAAGG with correct adjacencies. Then, exhaustively applying CD, RBO, KNC, CA, and MR3, all with orientation propagation of edges, produces a graph G' in which for every circle mark, there exists a MAAGG in the O-Equiv(*$\mathcal{D}_{\mathbf{O}}$*) class with a corresponding tail mark.*

Following Ali [2005]'s proofs for MAG, we apply the same reasoning for MAAGG and expand it with the RBO rule. The orientation propagation proof is identical to the one offered in Maier et al. [2013]. We now provide tail completeness of the remaining set of rules.

Lemma 3. *Let G' be the partially oriented PAAGG with correct adjacencies and unshielded colliders, and as many edges orientated with KNC, CA, and MR3, consistently applying edge propagation. Then, applying rules R5-R10, along with orientation propagation, provides a graph G" such that for every circle mark, there exists a MAAGG in O-Equiv($\mathcal{D}_{\boldsymbol{O}}$) in which the associated mark is an arrowhead.*

The proof comes from Zhang [2008] tail completeness, establishing that every PAAGG edge ∘—, ∘—∘, ∘→, the circle mark corresponds to an arrowhead in an MAAGG belonging to the equivalence class. With lemmas 2 and 3 in place, completeness follows:

Theorem 2. *Given a partially oriented PAAGG G with the appropriate set of adjacencies, applying rules CD, KNC, CA, MR3, and RBO extensively, followed by orienting any possible edges with rules R4-R10, all with orientation propagation, yields a maximally informative graph G.*

Proof. Lemmas 2 and 3 prove that in the output graph G produced by applying all the rules, every remaining circle mark corresponds to both tail and arrowhead variant marks in the *O-Equiv* ($\mathcal{D}_{\mathbf{O}}$). As such, the circle mark is considered a variation mark. Thus, by definition, the graph G is maximally informative. □

We are now ready to establish the soundness and completeness of RelFCI:

Theorem 3. *Given a schema and a probability distribution P(V) with $\boldsymbol{V} = \boldsymbol{O} \cup \boldsymbol{L} \cup \boldsymbol{S}$, the output of RelFCI is a correct maximally informative PAAGG, and thus a maximally informative PARM $\boldsymbol{\mathcal{M}}$, assuming perfect conditional independence tests and sufficient hop threshold h'.*

5 EXPERIMENTAL RESULTS

5.1 SETUP

Our RelFCI algorithm implementation[*] is based on the RFCI algorithm Colombo et al. [2012] rather than FCI.

RFCI performs significantly fewer conditional independence tests than other FCI variants. While not proven complete, experiments show it achieves similar accuracy in edge orientation. We generate synthetic data using a procedure similar to Maier et al. [2013] but with the addition of introducing latent variables into the schema and model. We generate 1000 random LRCMs from randomly generated schemas for each of the following combinations: number of entities $n \in [2, 4]$; $n - 1$ relationships with randomly selected cardinalities; attributes per item drawn from a Poisson distribution Pois($\lambda = 1$) + 1; and the number of relational dependencies (6, 8, 10, and 12) limited by hop threshold of 2. We additionally require the presence of one or two latent attributes, which are randomly chosen from the set of attributes for relational variables in the LAGG involved in at least two dependencies as the cause variable. The process yields a total of 22,000 synthetic models. We use an oracle to perform conditional independent tests for RelFCI and RCD for all possible perspectives. The results are then averaged over multiple runs for every combination, i.e., averaging over 1000 different LRCMs sharing the same properties.

5.2 EVALUATION

We evaluated our work by comparing the model derived from the algorithm's dependencies to the ground truth. We define the latent relational causal model obtained as ground truth by replacing the latent variable with double arrowhead edges using the same Maximal Ancestral Graph construction approach as presented in Zhang [2008]. We label a missing edge as a false negative, an additional edge as a false positive, and a correct edge as a true positive and compute the precision and recall. Furthermore, to assess the necessity of new rules for relational causal discovery, we also measure the frequency with which each rule was invoked during the RelFCI runs. This last result can be found in Appendix H.

5.3 RESULTS

Figure 4 presents a comparative analysis of RelFCI and RCD regarding precision and recall. An apparent discrepancy can be noticed in the results. This difference arises due to latent variables, which RCD fails to handle effectively. As previously discussed, the influence of hidden confounders violates RCD's core assumptions, significantly degrading its accuracy. In contrast, our proposed method, RelFCI, is designed to be sound and complete in the presence of latent variables. Since the RFCI implementation can sometimes introduce spurious edges or omit true ones, we expect its precision and recall to be slightly below one, as supported by Colombo et al. [2012]. Furthermore, RelFCI exhibits a smaller variance than RCD. This indicates that RelFCI produces more consistent and reliable results across different conditions, reinforcing its robustness in handling latent variables.

Figure 5 further illustrates the performance trends with either one or two latent variables as the number of entities and dependencies increases. A key observation is that while RCD's performances slightly improve as the number of entities and dependencies grows, its precision and recall remain consistently lower than those of RelFCI. This trend is par-

[*] Code available at `https://github.com/edgeslab/RelFCI`.

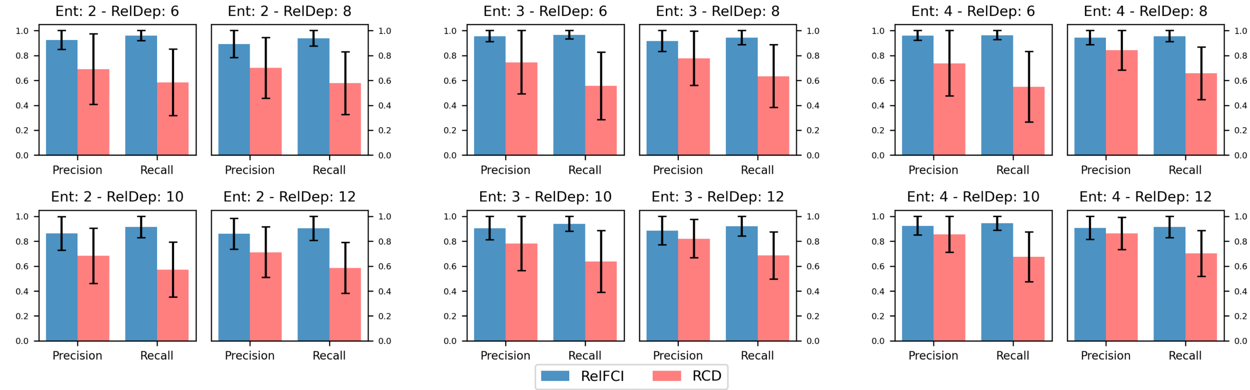


Figure 4: RelFCI and RCD Precision and Recall comparison. Results are combined for both 1 and 2 latent variables. Intervals represent ±1 standard deviation.

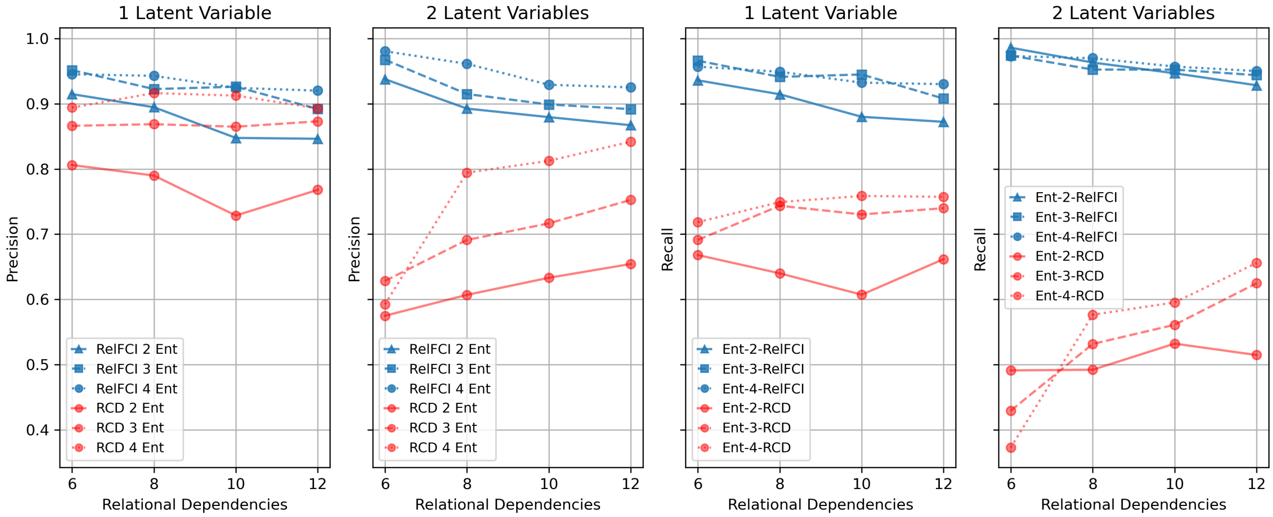


Figure 5: RelFCI and RCD Precision and Recall performance with 1 and 2 latent variables.

ticularly noticeable in recall, suggesting that RCD benefits marginally from increased structural complexity. RelFCI, instead, maintains stable and high precision and recall across all conditions. These findings highlight the robustness of RelFCI in handling relational datasets with latent variables, where RCD struggles to achieve comparable accuracy.

As an additional analysis, we evaluated our new algorithm's rule activation distribution over all synthetic runs. Rules unique to FCI account for approximately one-third of all orientations. It demonstrates that latent confounders impact the entire model structure during the learning process. The plot of the rules distribution is shown in Figure 6.

6 CONCLUSION

In this paper, we provide novel representations for relational causal models with latent confounders. We present a sound and complete algorithm, RelFCI, for detecting causality relationships from relational data with latent confounders, which provides a more comprehensive understanding of relational causal models. To the best of our knowledge, this approach is the first to study relational causal discovery with latent variables. We believe this work will be critical in enabling causal effect estimation in complex relational systems for which the underlying causal model is unknown. Areas of future work include investigating the effects of including selection bias and cycles into latent relational causal models.

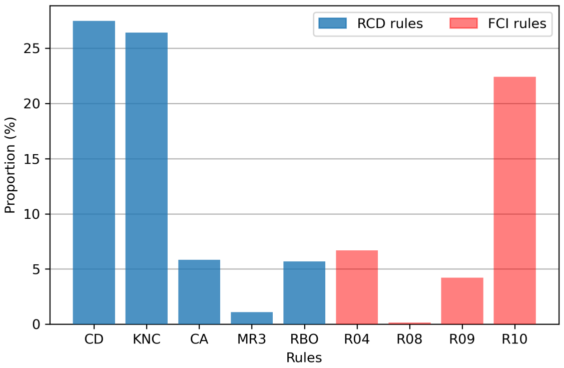


Figure 6: RelFCI's rule distribution of RCD and FCI rules.

Acknowledgements

This research was funded in part by NSF under grant no. 2047899. The authors would like to thank Aleksandr Elifirenko for valuable analysis on the code.

References

Ragib Ahsan, Zahra Fatemi, David Arbour, and Elena Zheleva. Non-parametric inference of relational dependence. In *Uncertainty in Artificial Intelligence*, pages 54–63, 2022.

Ragib Ahsan, David Arbour, and Elena Zheleva. Learning relational causal models with cycles through relational acyclification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 12164–12171, 2023.

Ayesha R Ali. Towards characterizing markov equivalence classes for directed acyclic graph models with latent variables. In *Proc. Conf. on Uncertainty in Artificial Intelligence (UAI-05)*, pages 10–17, 2005.

David Arbour, Katerina Marazopoulou, and David Jensen. Inferring causal direction from relational data. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pages 12–21, 2016.

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.

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

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

Sanghack Lee and Vasant Honavar. On learning causal models from relational data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016a.

Sanghack Lee and Vasant Honavar. A characterization of markov equivalence classes of relational causal models under path semantics. In *32nd Conference on Uncertainty in Artificial Intelligence 2016, UAI 2016*, pages 387–396, 2016b.

Sanghack Lee and Vasant Honavar. Towards robust relational causal discovery. In *Uncertainty in Artificial Intelligence*, pages 345–355, 2020.

Youjin Lee and Elizabeth L Ogburn. Network dependence can lead to spurious associations and invalid inference. *Journal of the American Statistical Association*, 116(535): 1060–1074, 2021.

Marc Maier, Katerina Marazopoulou, David Arbour, and David Jensen. A sound and complete algorithm for learning causal models from relational data. In *Uncertainty in Artificial Intelligence*, page 371, 2013.

Marc Maier, Katerina Marazopoulou, and David Jensen. Reasoning about independence in probabilistic models of relational data, 2014.

Christopher Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 403–410, 1995.

Joris M Mooij and Tom Claassen. Constraint-based causal discovery using partial ancestral graphs in the presence of cycles. In *Conference on Uncertainty in Artificial Intelligence*, pages 1159–1168, 2020.

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

Dominik Rothenhäusler, Christina Heinze, Jonas Peters, and Nicolai Meinshausen. Backshift: Learning causal cyclic graphs from unknown shift interventions. *advances in neural information processing systems*, 28, 2015.

Cosma Rohilla Shalizi and Andrew C Thomas. Homophily and contagion are generically confounded in observational social network studies. *Sociological methods & research*, 40(2):211–239, 2011.

Peter Spirtes, Christopher Meek, and Thomas Richardson. Causal inference in the presence of latent variables and selection bias. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*, pages 499–506, 1995.

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

Eric V Strobl. A constraint-based algorithm for causal discovery with cycles, latent variables and selection bias. *International Journal of Data Science and Analytics*, 8: 33–56, 2019.

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

Relational Causal Discovery with Latent Confounders

(Supplementary Material)

Matteo Negro[*1] **Andrea Piras**[*1] **Ragib Ahsan**[2] **David Arbour**[3] **Elena Zheleva**[1]

[1]University of Illinois Chicago, Chicago
[2]Pinterest, Inc., San Francisco
[3]Adobe Research, San Francisco

A BACKGROUND

A.1 RELATIONAL DATA

In this subsection, we provide possible examples of relational data. Figure 7 shows an example relational schema with two entities, USER (E) and POST (P), and the relationship between them, REACTS (P), with a MANY TO MANY cardinality, meaning users can react to multiple posts and vice versa. The USER type has three attributes: Type, Sentiment, and Activity, while the POST entity type has the attributes Content and Engagement. The relationship type REACTS instead has the attribute Frequency.

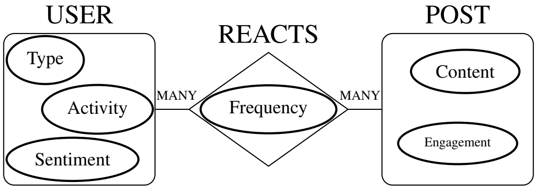


Figure 7: Example of Relational Schema

An example of an instantiation of the depicted relational schema can be seen in figure 8. For simplicity, attributes are left with the original placeholder for each entity and relationship instance. As an example, the skeleton contains three instantiations of the USER entity, Bob, Anna, and Andrea, and four instantiations of the POST entity type, Food recipe, Meme, Poem, and News. Bob and Anna react to the Food recipe and Meme, while Andrea reacts to the Poem and News. It is important to note that this skeleton is coherent with the cardinality requirements (i.e., MANY TO MANY) of the relationship defined in the schema.

Given the relational skeleton provided and the relational dependencies provided in the relational causal model in figure 1, it is possible to obtain the corresponding ground graph, shown in figure 9. The nodes on the ground graph

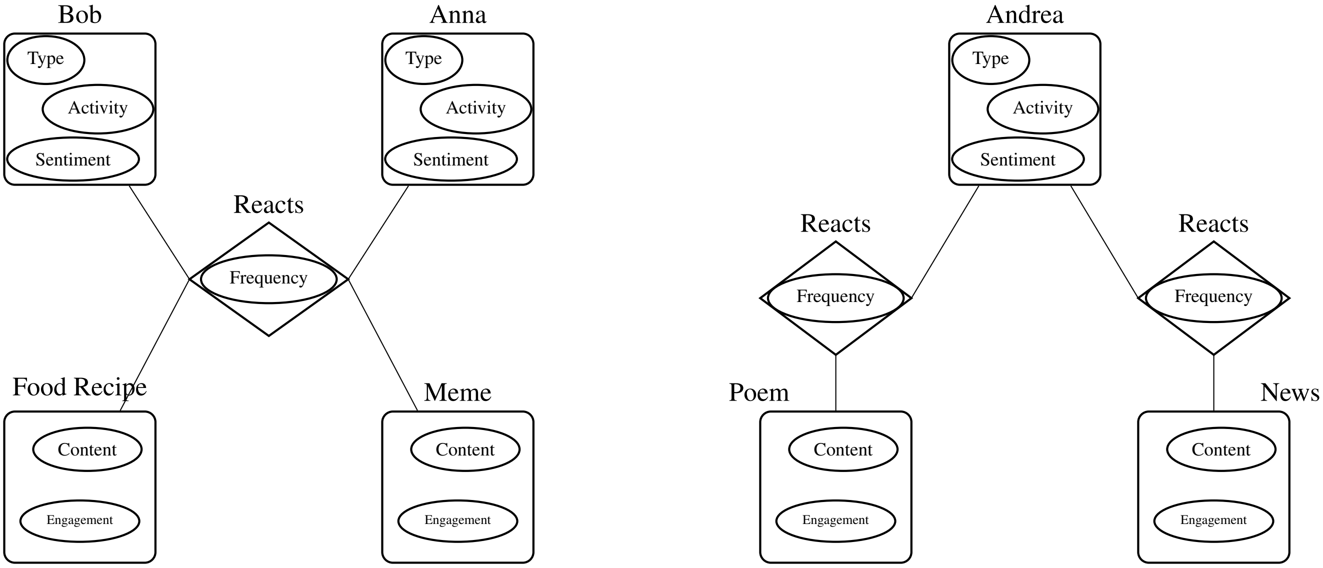


Figure 8: Example of Relational Skeleton

represent the attributes of every single entity and relationship instance in the skeleton. In contrast, the edges represent the dependencies in the relational causal model applied to the attribute instances of the relational skeleton. For example, the relational dependency $[P, R, U].Sentiment \rightarrow [P].Engagement$ in the model, which indicates that a post's engagement depends on the user's reaction to the product, is represented in the ground graph with the following edges: Bob.Sentiment → Food_Recipe.Engagement, Bob.Sentiment → Meme.Engagement, Anna.Sentiment → Food_Recipe.Engagement, Anna.Sentiment → Food_Recipe.Engagement, Andrea.Sentiment → Poem.Engagement, Andrea.Sentiment → News.Engagement.

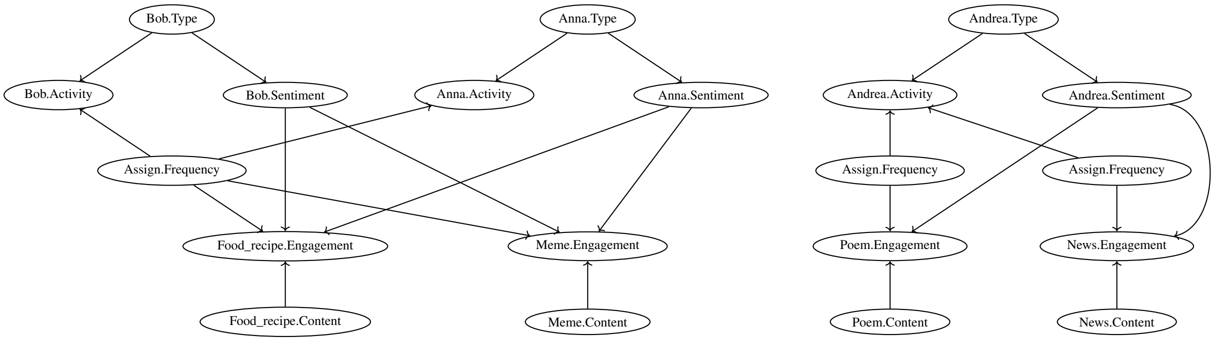


Figure 9: Example of Ground Graph

Beyond the specific instantiation of the ground graph, to perform relational causal discovery, it is necessary to define an abstract ground graph that generalizes the structure of dependencies without referring to particular entities or relationship instances. The abstract ground graph represents the relational dependencies in the relational causal model at a higher level, capturing attribute interactions without being tied to a specific skeleton. In this representation, nodes correspond to attribute types rather than instances, while edges represent the abstract relational dependencies in the model [Maier et al., 2014].

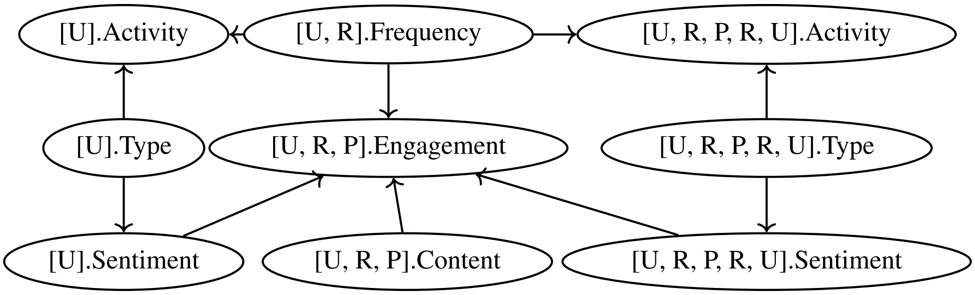


Figure 10: Example of Abstract Ground Graph for perspective USER

To construct the abstract ground graph from a given relational causal model, it is necessary to project the dependencies onto the relevant perspective. The `extend` method devised by Maier et al. [2014] achieves this by mapping underlying relational dependencies into the set of edges in the abstract ground graphs. Below we provide the formula of the `extend` method:

$$\text{extend}(P_{\text{orig}}, P_{\text{ext}}) = \left\{ P = P_{\text{orig}}^{1,n_o-i+1} + P_{\text{ext}}^{i+1,n_e} \;\middle|\; i \in \text{pivots}(\text{reverse}(P_{\text{orig}}), P_{\text{ext}}) \wedge \text{validPath}(P) \right\}$$

$$\text{pivots}(P_1, P_2) = \left\{ i \;\middle|\; P_1^{1,i} = P_2^{1,i} \right\}$$

Where validPath(P) checks that the relational path is valid with the respect to the schema and its relationships' cardinalities. Each abstract ground graph edge $[B, \ldots, I_k].Y \rightarrow [B, \ldots, I_j].X$ is then constructed from the underlying dependency $[I_j, \ldots, I_k].Y \rightarrow [I_j].X$ with the following logic:

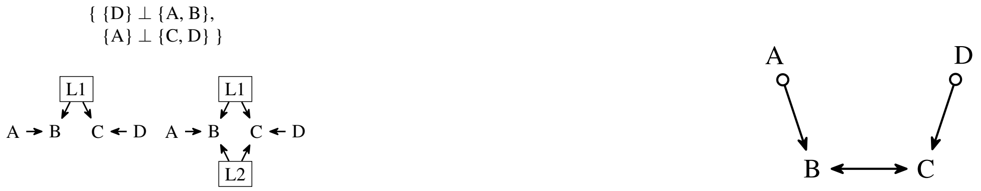
$$\{[B, \ldots, I_k].Y \rightarrow [B, \ldots, I_j].X \mid [I_j, \ldots, I_k].Y \rightarrow [I_j].X \in \mathcal{D} \wedge [B, \ldots, I_k] \in \text{extend}([B, \ldots, I_j], [I_j, \ldots, I_k])\}$$

For example, the relational dependency $[P, R, U].Sentiment \rightarrow [P].Engagement$, which in the ground graph manifests as instance-specific edges (e.g., Bob.Sentiment → Food_Recipe.Engagement), is represented in the abstract ground graph for the perspective USER with the directed edges $[U].Sentiment \rightarrow [U, R, P].Engagement$ and $[U, R, P, R, U].Sentiment \rightarrow [U, R, P].Engagement$.

Similarly, other relational dependencies in the model are reflected as edges between attribute types in the abstract ground graph, providing a compact and generalized view of how information propagates through the relational structure. Analyzing the abstract ground graph makes it possible to reason about potential influences and dependencies at the schema level without requiring explicit enumeration of individual instances.

A.2 MAGS AND PAGS

In this subsection, we provide an example of how more than one MAG can be a member of the same PAG and single equivalency class. Given a collection of observable variables, let **Cond** in figure 11a represent the set of conditional dependencies. It is evident that it is entailed by several DAGs. Figure 11b displays the PAG that was generated for **Cond**. Since they are not mentioned in the conditional set, A and D's edge marks are $\circ$, which could lead to different marks for various DAGs in *O-Equiv*(**Cond**).



(a) DAGs in same O-Equiv(Cond) class

(b) Resulting PAG for O-Equiv(Cond) class

Figure 11: DAGs in the same observational equivalence class under **Cond** (a), and the resulting PAG (b) that captures shared structure and uncertainty in edge directions.

B RELFCI RULES

This section outlines every rule we apply to the new Partial Ancestral Abstract Ground Graph representation to obtain a maximally informative graph and, thus, an underlying model. We introduce the rules in the framework of PAAGGs, where any $\circ$ marks represent unoriented edges and $*$ denotes any edge mark.

B.1 RCD RULES

RCD [Maier et al., 2013] performs relational causal discovery using a similar strategy to the Poem algorithm, extended with the RBO purely common cause rule. The edges of the abstract ground graph are oriented using the following set of rules:

1. Collider Detection (CD): For each triple $\langle \alpha, \beta, \gamma \rangle$, if β is not in the set that separates α and γ, orient it as $\alpha \ast\!\!\rightarrow \beta \leftarrow\!\!\ast \gamma$;

2. Relational Bivariate Orientation (RBO): Let $\mathcal{M}$ be a relational causal model and G a partially directed PAAGG for $\mathcal{M}$ for perspective I_X, and let there be an unshielded triple in G $\alpha \circ\!\!-\!\!\circ \beta \circ\!\!-\!\!\circ \gamma$ with $\alpha = [I_X].X, \beta = [I_X, ..., I_Y].Y, \gamma = [I_X, ..., I_Y, ..., I_X].X$. If *card*$([I_Y, ..., I_X]) = \texttt{MANY}$ and $\alpha \perp\!\!\!\perp \gamma | \mathbf{Z}$, then if $\beta \in \mathbf{Z}$, orient the triple as $\alpha \leftarrow\!\!\circ\ \beta\ \circ\!\!\rightarrow \gamma$;

3. Known Non-Colliders (KNC): If $\alpha \ast\!\!\rightarrow \beta \circ\!\!-\!\!\ast \gamma$, with α, γ not adjacent, orient the triple as $\alpha \ast\!\!\rightarrow \beta \rightarrow \gamma$

4. Cycle Avoidance (CA): If either $\alpha \rightarrow \beta \ast\!\!\rightarrow \gamma$ or $\alpha \ast\!\!\rightarrow \beta \rightarrow \gamma$, with $\alpha \ast\!\!-\!\!\circ \gamma$, orient the latter as $\alpha \ast\!\!\rightarrow \gamma$;

5. Meek Rule 3 (MR3): If both $\alpha \ast\!\!\rightarrow \beta \leftarrow\!\!\ast \gamma$ and $\alpha \ast\!\!-\!\!\circ \theta \circ\!\!-\!\!\ast \gamma$, with α, γ not adjacent and $\theta \ast\!\!-\!\!\circ \beta$, then orient the latter as $\theta \ast\!\!\rightarrow \beta$.

B.2 FCI RULES

FCI [Zhang, 2008] constructs a causal graph starting from a fully connected undirected graph with $\circ$ marks and removes edges between conditionally dependent variables. In the second phase, it orients edges by identifying colliders and "Y" structures. The remaining edges are then oriented according to a set of additional rules:

4. If $u = \langle \theta, ..., \alpha, \beta, \gamma \rangle$ is a discriminating path and $\beta{*}\text{—}\gamma$, if $\beta \in \mathit{SepSet}(\theta, \gamma)$ orient $\beta \rightarrow \gamma$, otherwise orient $\alpha \leftrightarrow \beta \leftrightarrow \gamma$;
5. For every (remaining) $\alpha{\circ}\text{—}{\circ}\beta$, if there is an uncovered path $p = \langle \alpha, \gamma, ..., \theta, \beta \rangle$ s.t. all edges are ${\circ}\text{—}{\circ}$ and α, θ are not adjacent and β, γ are not adjacent, then orient all edges in the path as —;
6. If $\alpha\text{—}\beta{\circ}\text{—}{*}\gamma$, with α, γ either adjacent or not, orient $\beta\text{—}{*}\gamma$;
7. If $\alpha\text{—}{\circ}\beta{\circ}\text{—}{*}\gamma$, and α, γ are not adjacent, orient $\beta\text{—}{*}\gamma$;
8. If $\alpha\text{—}{\circ}\beta \rightarrow \gamma$ or $\alpha\text{—}{\circ}\beta \rightarrow \gamma$, and $\alpha{\circ}\!\!\rightarrow \gamma$, orient $\alpha \rightarrow \gamma$;
9. If $\alpha \circ\!\!\rightarrow \gamma$ and $p = \langle \alpha, \beta, \theta, ..., \gamma \rangle$ is an uncovered path s.t. β and γ are not adjacent, orient $\alpha \rightarrow \gamma$;
10. If $\alpha \circ\!\!\rightarrow \gamma$, $\beta \rightarrow \gamma \leftarrow \theta$, and p_1, p_2 are uncovered p.d. paths from α to β and from α to θ, let μ and ω be the adjacent nodes of α on p_1, p_2. If μ and ω are distinct, orient $\alpha \rightarrow \gamma$.

C ALGORITHMS

The following section provides more detailed pseudocode for each step in the main algorithm. The described algorithm and steps are adapted from the implementation provided in Colombo et al. [2012]. For easy reference, the main RelFCI pseudocode is provided again below in Algorithm 2.

Algorithm 2 RelFCI algorithm

```
Input: schema, oracle,
Parameter: threshold
Output: Dependencies
// Step 1: Graphs initialization
$PDs \leftarrow$ get potential Dependencies from the base schema (with no dependencies) and two times the threshold (2*h)
$PAAGGs \leftarrow$ construct PAAGGs from potential dependencies set $PDs$
$S \leftarrow \{\}$
// Step 1: Independent Variables identification, storing separating sets and unshielded triples
$PAAGGs, S, U \leftarrow$ obtainInitialSkeleton($PAAGGs, S$)
// Step 2: V-structures orientation using CD, starting from unshielded triples in $U$
$PAAGGS, S \leftarrow$ orientVStructures($PAAGGs, S, U$)
// Step 3: edges orientation using rules from RCD and additional ones from FCI
$PAAGGs, S \leftarrow$ performEdgeOrientation($PAAGGs, S$)
$Deps \leftarrow$ retrieve underlying dependencies from oriented PAAGGs edges
return Deps
```

Algorithm 3 obtainInitialSkeleton

Input: Schema, Oracle,
Parameter: threshold, depth
Output: Non-oriented AGGs

```
for $agg$ in AGGs do
    Let $l = 0$
    Let $max\_depth = agg.number\_of\_nodes - 2$
    while $l \leq max\_depth$ do
        for all pair of vertices ($X_i$, $X_j$) in $agg$ do
            Let $C = agg.nodes - \{X_i, X_j\}$
            for all $Y \subseteq C$ do
                if CITest($X_i$, $X_j$, $Y$) then
                    Remove dependencies between ($X_i$, $X_j$)
                    Store $Y$ as $sepSet$ for ($X_i$, $X_j$)
                end if
            end for
        end for
        Let $l = l + 1$
    end while

    for all triple of vertices ($X_k$, $X_j$, $X_m$) in $agg$ do
        if $k < m$ then
            if $agg.has\_edge(X_k, X_j)$ and $agg.has\_edge(X_j, X_m)$ and not $agg.has\_edge(X_k, X_m)$ then
                Append ($X_k$, $X_j$, $X_m$) to $unshieldedTriples[agg]$
            end if
        end if
    end for
end for
```

Algorithm 4 orientVStructures

Input: Schema, Oracle,
Parameter: threshold, depth
Output: Partially oriented AGGs

```
for $agg$ in AGGs do
    while $unshieldedTriples[agg]$ do
        Let $(X_i, X_j, X_k) = unshieldedTriples[agg].pop()$
        Let $Z = sepSet(X_i, X_k) - \{X_j\}$
        if not CITest($X_i$, $X_j$, $Z$) and not CITest($X_j$, $X_k$, $Z$) then
            Append ($X_i$, $X_j$, $X_k$) to $dependentTriples[agg]$
        else
            for $X_r$ in [$X_i$, $X_k$] do
                if CITest($X_r$, $X_j$, $Z$) then
                    Let $Y = findMinimalSepset(X_r, X_j, Z)$
                    Store $Y$ as $sepSets$ for ($X_r$, $X_j$)
                    for all $X_x$ in $agg.nodes$ do
                        if isTriangle($X_{min(r,j)}, \cdot, X_{max(r,j)}$) then
                            Add to $unshieldedTriples[agg]$ the triple
                        end if
                    end for
                    for all triple in $unshieldedTriples[agg]$ do
                        Delete the triple if matches one of the following patterns: $(X_r, X_j, \cdot)$, $(X_j, X_r, \cdot)$, $(\cdot, X_j, X_r)$ and
                        $(\cdot, X_r, X_j)$
                    end for
                    Remove dependencies between ($X_r$, $X_j$)
                end if
            end for
        end if
    end while
    for all triple in $dependentTriples[agg]$ do
        Let $X_i, X_j, X_k = triple$
        if $X_j$ not in $sepSets(X_i, X_k)$ and $agg.has\_edge(X_i, X_j)$ and $agg.has\_edge(X_j, X_k)$ then
            Orient the triple as a collider
        end if
    end for
end for
```

Algorithm 5 performEdgeOrientation

Input: Schema, Oracle,
Parameter: threshold, depth
Output: Maximum oriented AGGs

```
for $agg$ in AGGs do
    while AGG is updated do
        Orient as many edges as possible by applying RBO rule
        Orient as many edges as possible by applying FCI_1 - FCI_3 rules
        for all possible triples do
            Let $X_l, X_j, X_k = triple$
            if $isTriangle(X_l, X_j, X_k)$ and $X_j \circ\!\!-\!\!* X_k$ and $X_l \leftarrow\!\!* X_j$ and $X_l \rightarrow X_k$ then
                Find Minimal Discriminating Path for the triple
                if minimalDiscriminatingPath then
                    for all adjacent couples do
                        Let $X_r, X_q = couple$
                        Let $otherSepSet = sepSets(X_i, X_k) - X_r, X_q$
                        Let $l = -1$
                        while $|otherSepSet| \geq l$ do
                            Let $l = l + 1$
                            for all $Y \subseteq otherSepSet$ and $|Y| = l$ do
                                if CITest($X_r$, $X_q$, $Y$) then
                                    Store $Y$ as $sepSet$ for ($X_r$, $X_q$)
                                    for all $X_x$ in $agg.nodes$ do
                                        if isTriangle($X_{min(r,j)}, \cdot, X_{max(r,j)}$) then
                                            Add to $unshieldedTriples[agg]$ the triple
                                        end if
                                    end for
                                    Remove dependencies between ($X_r$, $X_q$)
                                    Execute Algorithm 2
                                end if
                            end for
                        end while
                    end for
                    if Still adjacent and $X_j$ in $sepSets(X_i, X_k)$ then
                        Orienting $X_j \rightarrow X_k$
                    else if Still adjacent then
                        Orienting $X_l \leftrightarrow X_j \leftrightarrow X_k$
                    end if
                end if
            end if
        end for
        Orient as many edges as possible by applying FCI_5 - FCI_10 rules
    end while
end for
```

D POSSIBLE DEPENDENCIES

The presence of ∘ marks in the edge of $PAAGGs$, and thus in the underlying $PARM$, implies that the *O-Equiv*($\mathcal{D}_{\mathbf{O}}$) class contains different relational causal models. The algorithm's output is not the exact relational causal model that generates the data. RelFCI returns an equivalence class containing the model responsible for the data causal relationships. RelFCI computes conditional independence tests among the variables, thus possibly producing the same result with different underlying topologies e.g., with the independence fact $A \perp\!\!\!\perp C \mid B$, the nodes A, B, and C can be correctly oriented as follows: $A \rightarrow B \rightarrow C$, $A \leftarrow B \rightarrow C$, $A \leftarrow B \leftarrow C$, $A \rightarrow B \leftarrow C$ [Spirtes et al., 2000]. RelFCI works by learning the edges' orientation of each $PAAGG$, which are defined by underlying relational dependencies.

When the algorithm concludes and collects all the information learned to produce the $PARM$, the remaining ∘ marks lose significance in terms of relational dependencies. The definition of relational dependency in canonical form implies a natural orientation, i.e., $[I_X...I_Y].Y \rightarrow [I_X].X$. Orienting dependencies the other way around is an infraction of the definition, i.e., $[I_X].X \nrightarrow [I_X...I_Y].Y$. For this reason, given this formalization of the problem, we differentiate the information the algorithm learns by clearly stating which relational dependencies are required to define the $PARM$ and which are instead allowed. We define the required relational dependencies with a →, i.e., $[I_X...I_Y].Y \rightarrow [I_X].X$ and the ones that are allowed but not necessary with a ⇝, i.e., $[I_X...I_Y].Y \rightsquigarrow [I_X].X$. We will refer to the latter as *Possible Dependencies*.

E PAAGG EDGE ORIENTATION

We apply the four PC rules and the new RBO rule, described in RCD, and further apply the rules of FCI, as defined by Zhang (2008), adapted for the PAAGG representation. A latent relational causal model consists of a set of AGGs, one for each perspective, derived from the same set of relational dependencies $\mathcal{D}$. Similarly, both MAAGGs and PAAGGs are derived from the same collection of observed relational dependencies $\mathcal{D}_{\mathbf{O}}$. In classical AGGs, activating a rule in a certain abstract ground graph involves propagating the orientation of the underlying dependency across all AGGs [Maier et al., 2013].

Consider a PARM $\boldsymbol{\mathcal{M}}$ defined over the set of dependencies $\mathcal{D}_{\mathbf{O}}$ and its corresponding PAAGG G for the perspective $\mathcal{B}$. Let $\alpha = [\mathcal{B}, ..., I_X].X$ and $\gamma = [\mathcal{B}, ..., I_Y].Y$ be two nodes in G, $\alpha - \gamma$ be a bidirected edge in G, and $d_1 = [I_X, ..., I_Y].Y \rightarrow [I_X].X \in \mathcal{D}_{\mathbf{O}}$ be the underlying dependency that yields the left direction of the edge. The FCI rules can orient a PAG edge with three edge marks: ∘, —, and →. We apply these orientations to the PAAGG using the following logic:

- The orientation α∘—γ implies that the underlying dependency d_1 belongs to the set of possible dependencies;
- The orientation $\alpha - \gamma$ implies that the underlying dependency d_1 is not coherent with the edge orientation and, as such, is not existent in the underlying PARM;
- The orientation $\alpha \leftarrow \gamma$ indicates that the underlying dependency d_1 is consistent with the edge orientation and belongs to the category of exact dependencies.

With this logic, the same propagation property applies to new representations that share the same underlying dependencies because exact and potential dependencies are propagated equally.

F EXAMPLE EXECUTION OF RELFCI

To illustrate the functioning of the RelFCI algorithm, we provide a step-by-step execution over an example relational causal model. This walk-through demonstrates the graphical transformations applied to the Partial Ancestral Abstract Ground Graph (PAAGG) across the different phases of the algorithm. Each figure referenced corresponds to a visual depiction of the model after the respective step of the algorithm.

Note: For this example, we focus on a single perspective (in this case, $AB1$). Similar graphs and reasoning are applied to all other perspectives. Rule propagation ensures that orientations in one PAAGG are reflected across others in line with shared underlying dependencies.

INITIAL MODEL AND UNDERLYING GRAPH

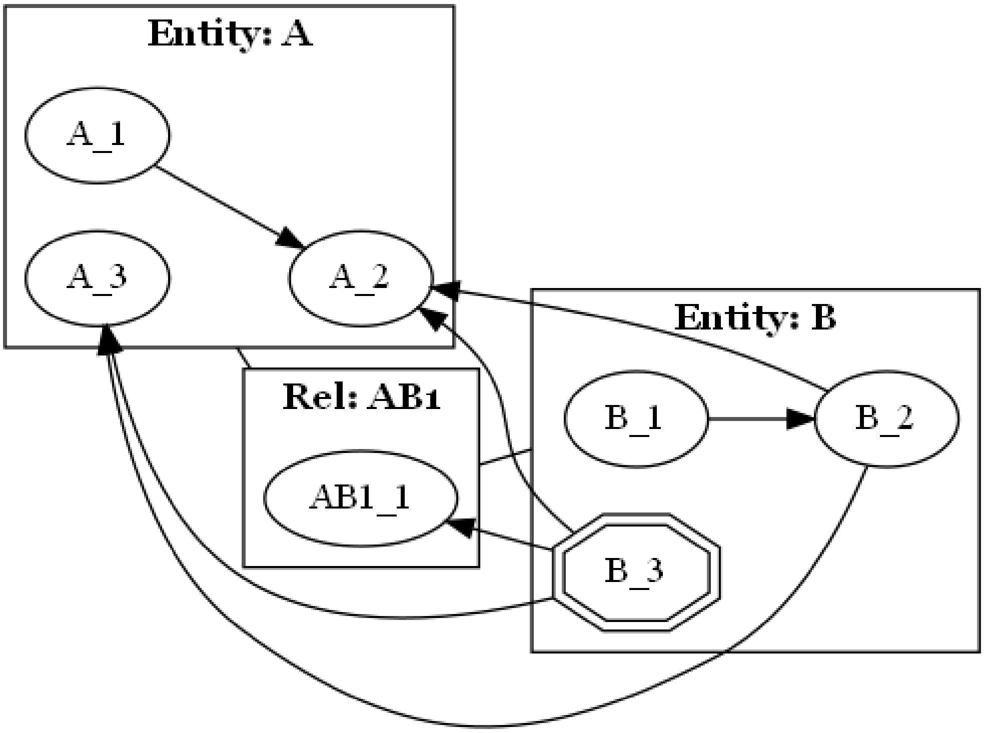


Figure 12: Relational causal model with entities, relationships, and dependencies, including latent variables.

We begin with a relational causal model that includes observed and latent variables. The figure depicts:

- Entities A and B with a relationship $AB1$;

- Attributes A_1, A_2, A_3, B_1, B_2 (observed), and B_3 (latent, represented with a double edges octagon);
- Dependencies between relational variables, considering a hop threshold $h = 2$:
  - Observed dependencies $\in \mathcal{D}_{\boldsymbol{O}}$: $[A].A_1 \rightarrow [A].A_2$, $[A, AB1, B].B_2 \rightarrow [A].A_2$, $[A, AB1, B].B_2 \rightarrow [A].A_3$, $[B].B_1 \rightarrow [B].B_2$].
  - Unobserved dependencies $\in \mathcal{D}_{\boldsymbol{L}}$: $[A, AB1, B].B_3 \rightarrow [A].A_2$, $[A, AB1, B].B_3 \rightarrow [A].A_3$, $[AB1, B].B_3 \rightarrow [AB1].AB1_1$.

PHASE 0 – PAAGG CONSTRUCTION

In this phase, the algorithm constructs the PAAGGs with all possible dependencies:

- A node is created for each relational variable with a path length up to the hop threshold $h' = 2h = 4$.
- Edges are added according to the `extend` method, resulting in a fully connected undirected graph with ∘−∘ marks.
- Intersection variables are included if needed to maintain the closure under intersections. In this example, these variables are excluded from the plots for better readability.

The graph in 13 represents the PAAGG with all potential dependencies for the perspective $AB1$.

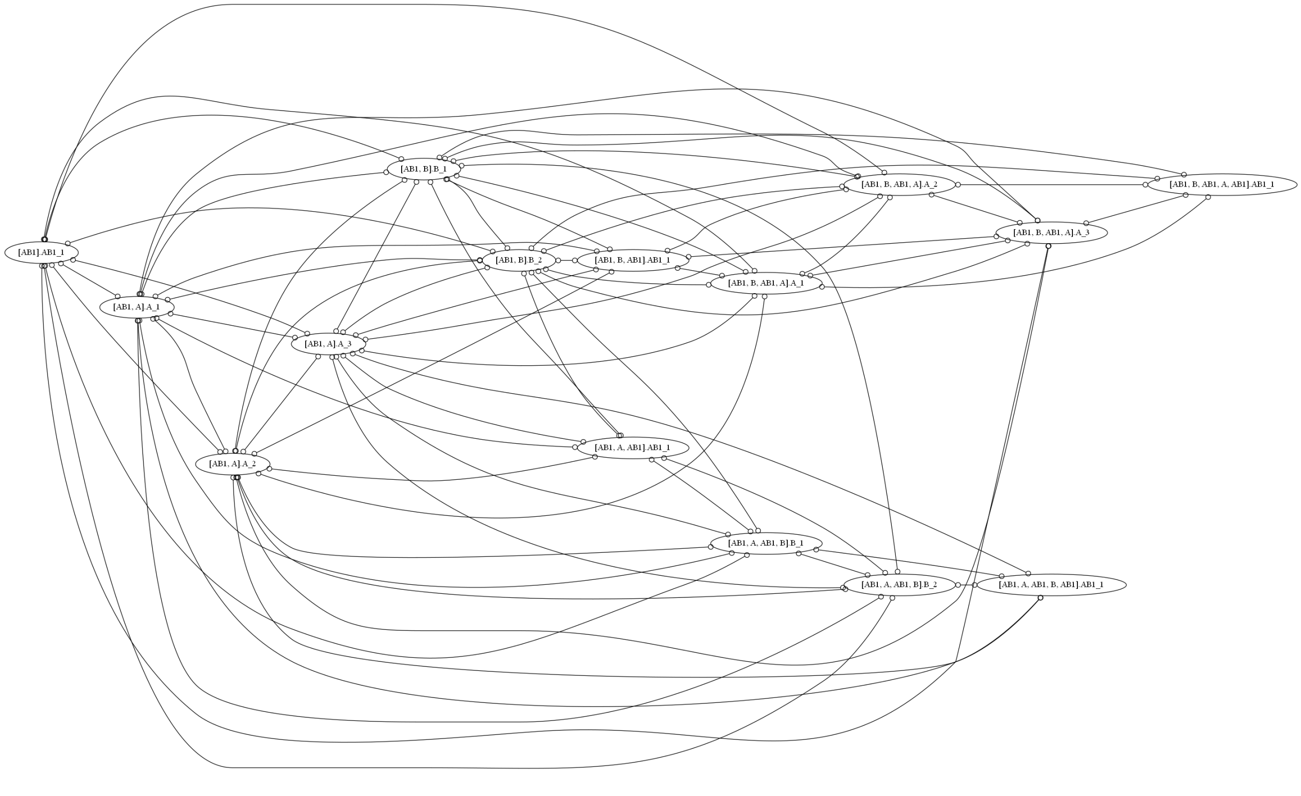


Figure 13: Fully connected PAAGG for perspective $AB1$.

PHASE 1 – INITIAL SKELETON IDENTIFICATION VIA CONDITIONAL INDEPENDENCE TESTING

The algorithm now performs conditional independence tests between every pair of variables, using increasingly bigger separating sets. If the two variables are found to be independent conditioned on the variables in the separating set, the edge is removed, and the set is stored.

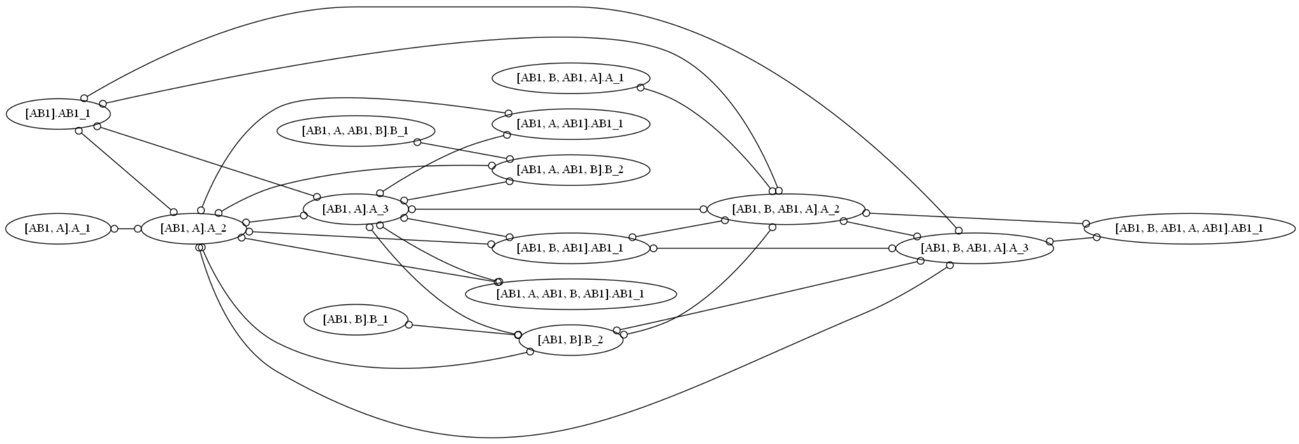


Figure 14: PAAGG after conditional independence testing.

Unshielded triples are also identified at this stage as candidate collider patterns. In this example, the following triples are found:

- $[AB1].AB1_1, [AB1, A].A_2, [AB1, A].A_1$;
- $[AB1].AB1_1, [AB1, A].A_2, [AB1, B].B_2$;
- $[AB1].AB1_1, [AB1, A].A_2, [AB1, A, AB1, B].B_2$;
- $[AB1].AB1_1, [AB1, A].A_3, [AB1, B].B_2$;
- $[AB1].AB1_1, [AB1, A].A_3, [AB1, A, AB1, B].B_2$;
- $[AB1].AB1_1, [AB1, B, AB1, A].A_2, [AB1, B].B_2$;
- $[AB1].AB1_1, [AB1, B, AB1, A].A_2, [AB1, B, AB1, A].A_1$;
- $[AB1].AB1_1, [AB1, B, AB1, A].A_3, [AB1, B].B_2$.

PHASE 2 – COLLIDER DETECTION AND V-STRUCTURE ORIENTATION

This phase introduces the first directed edge orientations in the graph. The algorithm starts by checking whether the unshielded triples are found to be dependent (i.e., for triple X, Y, Z, X, Z and Y, Z are not independent given the separating set of X and Z) or not. For this example, all 7 unshielded triples are identified as dependent. Then, the CD rule is applied to identify and orient colliders among these triples. The PAAGG after CD is applied is shown on figure 15.

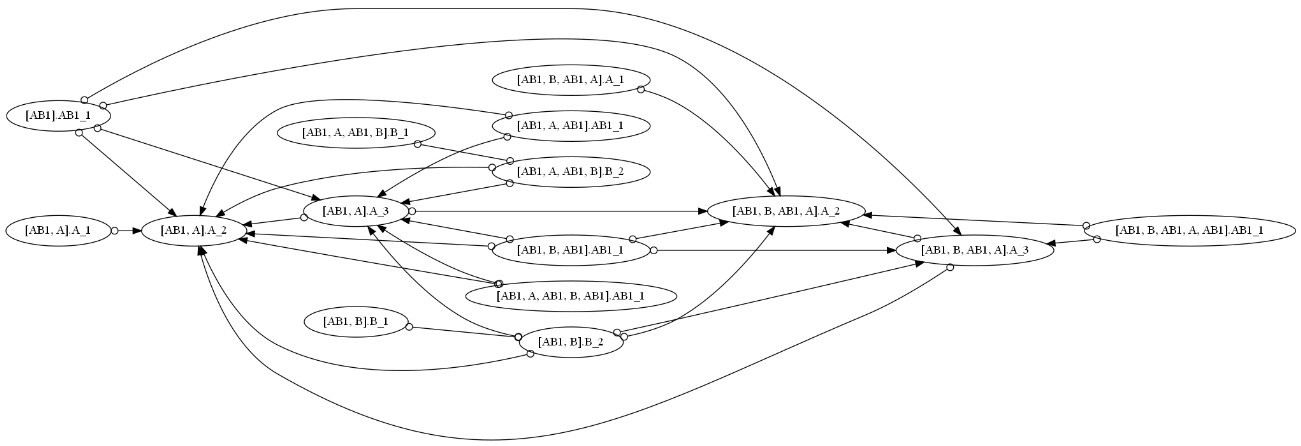


Figure 15: PAAGG after collider orientation via CD.

PHASE 3 – FURTHER ORIENTATION VIA RCD AND FCI RULES

In this step, remaining ambiguous edge marks are refined using the additional RCD (RBO, CA, MR3, and KNC) and FCI rules, repeating this process until no rule can be applied anymore. For this example:

- Rule KNC is activated once to orient the triple $[AB1, A, AB1].AB1_1 \ast\!\!\rightarrow [AB1, A].A_3 \rightarrow [AB1, B, AB1, A].A_2$ and all other triples sharing the same underlying dependencies;
- FCI rule R4 is activated once to orient the triangle $[AB1, A].A_3 \leftrightarrow [AB1].AB1_1 \leftrightarrow [AB1, B, AB1, A].A_2$ and all other triples sharing the same underlying dependencies;
- All other rules are not activated.

After all rule applications and orientation propagation, the resulting PAAGG (Figure 16 is maximally informative: each remaining $\circ$ mark reflects a true ambiguity in the equivalence class O-Equiv(D_O).

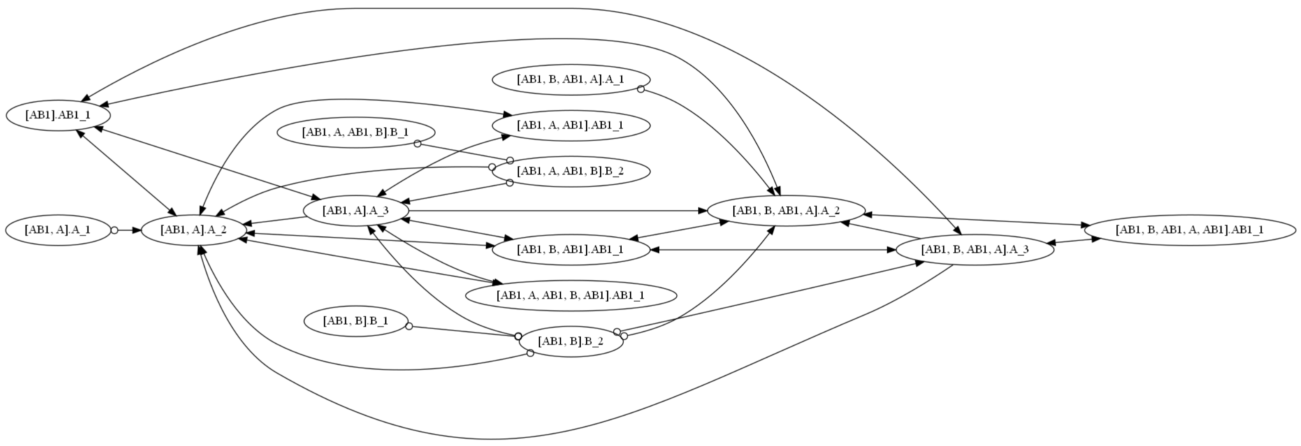


Figure 16: Final PAAGG with maximally informative edge orientations.

OUTPUT – EXTRACTION OF DEPENDENCIES

From the oriented PAAGGs, the algorithm extracts the required and possible underlying dependencies. These define the Partial Ancestral Relational Model, shown in Figure 17.

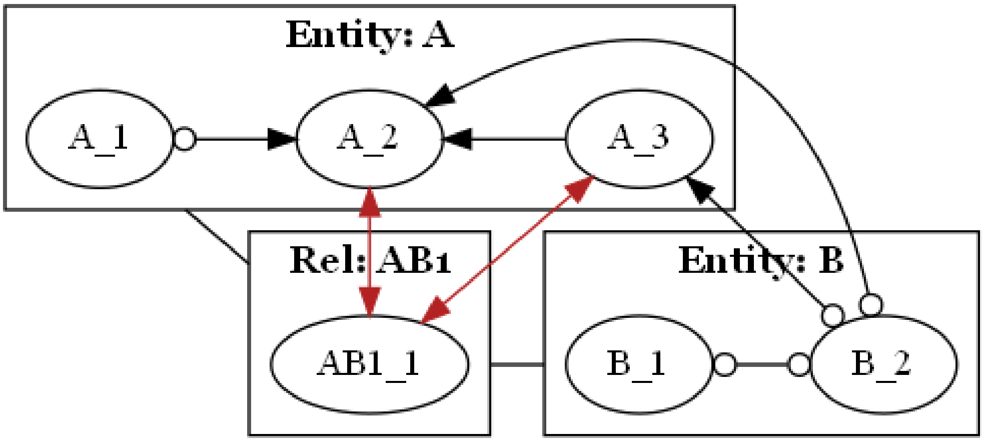


Figure 17: Learned PARM for the example model.

G PROOFS

This section contains complete proofs for all the theoretical results presented in the main paper.

Lemma 1. *Given a relational causal model structure $\mathcal{M}$ and perspective $\mathcal{B}$, if an abstract ground graph $AGG_{\mathcal{M}\mathcal{B}}$ is ancestral, then all ground graphs $GG_{\mathcal{M}\sigma}$, with skeleton $\sigma \in \sum_{\mathcal{S}}$, are ancestral.*

Proof. From the definition of Zhang [2008], a graph is ancestral if:

1. There is no directed cycle, i.e., B→A is in G and A is an ancestor of B (meaning there's a directed path from A to B);
2. There is no almost directed cycle, i.e., B↔A is in G and A is an ancestor of B;
3. For any undirected edge A—B, both A and B have no parent or spouses, i.e., X, Y such that either or both A↔X or B↔Y.

For each of the three conditions, we must demonstrate that if the AGG is ancestral, all GGs must likewise be ancestral to prove this lemma. Given the definition of the abstract ground graph building process in Definition 5.2 and Theorem 5.2 from Maier et al. [2014], we know that the AGG is sound and complete for all ground graphs for a given perspective and hop threshold h. This suggests that the AGG captures every dependent path between two variables in every GG. In the same way, each path of dependence between two variables in the AGG is mirrored in at least one GG. We now verify the lemma for the three conditions of ancestrality:

1. Assume that the AGG is ancestral and that one of the ground graphs, G, has a directed cycle between A and B to provide a contradiction. Consequently, the two dependence paths in G will also be present in the AGG, resulting in a directed cycle. Thus, the maximal ancestral abstract ground can't be ancestral;
2. Similar reasoning can be carried when considering almost directed cycles containing double-arrowed edges (in the case of *Maximal Ancestral Abstract Ground Graphs*), thus verifying the lemma for this condition as well;
3. Given the assumptions of the underlying structure's acyclicity and no selection bias (i.e., no variables are in the set **S**), an undirected edge cannot exist as it corresponds to the presence of selection variables, of which X and Y are the cause Zhang [2008]. Thus, this condition does not apply to AGGs.

□

Lemma 1 guarantees that the theoretical reasoning devised for MAGs and PAGs can also be applied to the relational counterparts we provide in this work, MAAGGs, and PAAGGs. In other words, we know that the ancestrality of these relational lifted representations corresponds to the same ancestrality properties in the underlying ground graphs and, thus, in the underlying latent causal relational causal model we want to learn.

Proposition 3. *Given a relational causal model $\mathcal{M}_L(\mathcal{S}, \mathcal{D})$ with hop threshold h, and its respective latent abstract ground graph $LAGG$:*

I. The constructed MAAGG probabilistically and causally represents $LAGG$ and thus the underlying relational causal model;

II. Assuming a sound and complete procedure to construct the $PAAGG$, it correctly represents the Markov equivalence class of the produced $MAAGG$ and, therefore, of $LAGG$ and the underlying model $\mathcal{M}_L$.

Proof. I. We can demonstrate that the MAAGG, constructed from $LAGG$ by employing the same MAG construction procedure provided in Zhang [2008], probabilistically and causally represents it as a result of theorem 4.18 of Richardson and Spirtes [2002], where they show that the independence model corresponding to the constructed graph coincides with the one obtained by marginalizing and conditioning the model on the original graph ($LAGG$). Furthermore, the MAAGG also represents the model $\boldsymbol{\mathcal{M}}_{\mathbf{L}}$, which follows from Lemma 1.

II. Under the assumption of a sound and complete procedure for generating said representation (i.e., the RelFCI algorithm), the PAAGG represents the Markov equivalence class containing the MAAGG. This proof follows from Zhang [2008]: the PAAGG, constructed from a sound and complete algorithm that outputs a set of graphs which includes all the causal relationships consistent across all MAAGGs, accurately represents the equivalence class. This is because it captures the uncertainty (circle marks) where the data does not provide enough information to distinguish between different causal structures. Finally, from I., we can prove that the PAAGG also represents the equivalence class of $LAAG$ and the underlying model $\boldsymbol{\mathcal{M}}_{\mathbf{L}}$.

□

Proposition 4. *Given a latent relational causal model $\mathcal{M}_{\boldsymbol{L}}(\mathcal{S}, \mathcal{D})$ with hop threshold h and its corresponding PARM $\boldsymbol{\mathcal{M}}$, the hop threshold h' of the $PAAGG_{\boldsymbol{\mathcal{M}}\mathcal{B}}$ for any perspective $\mathcal{B}$ can be at most $2h$.*

Proof. Let us consider a scenario within a relational causal model that allows relational latent variables to be observed and in which the non-dependence of these variables holds (i.e., no latent variable causes another latent variable, which entails there cannot exist a chain of dependencies consisting of multiple consecutive latent variables). For the sake of clarity, we will focus on three entities, A, B, and C, each containing one attribute, respectively $A1$, $B1$, and $C1$, with $B1$ designated as latent as in Figure 18. Suppose we were to connect them, using B1 as the connecting bridge between the other two attributes using the following dependencies: $[A, B].B1 \to [A].A1$ and $[C, B].B1 \to [C].C1$, both of which require a hop threshold of one to be represented. After removing the assumption of having all variables observed, the scenario reverts to one where $B1$ is latent, which means that the dependencies between $B1 - A1$ and $B1 - C1$ are no longer observable. The possible existing dependencies, containing only relational variables with a path of length two (hop threshold equal to one), make the model unable to express the dependencies among the attributes of different entities, e.g., $[A, B, C].C1 \to [A].A1$ and $[C, B, A].A1 \to [C].C1$. To account for the relational dependencies between the two entities, we need a relational path that is long enough to traverse the entities and describe the relationship between the variables expressed by the model, which requires twice the original hop threshold of one. □

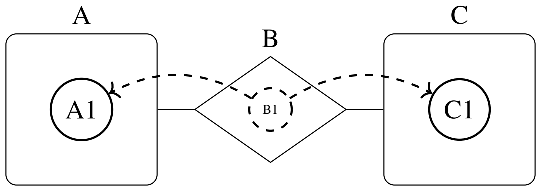


Figure 18: Example of Relational Causal Model with a latent variable

Theorem 4. *Let G be the partially oriented PAAGG from perspective B with the correct set of adjacencies, unshielded colliders oriented correctly through CD and RBO, and as many edges as possible oriented through KNC, CA, MR3, and the purely common cause of RBO. Then, the rules R4-R10 from FCI and the orientation propagations are sound.*

Proof. Given lemma 1, the proof derives from Spirtes et al. [1995] and Zhang [2008]. A rule is sound if the arrows and tails used in the resulting PAAGG are invariant. Therefore, we need to prove that any mixed abstract ground graph G that violates a rule does not belong to the equivalence class *O-Equiv*($\mathcal{D}_{\mathbf{O}}$), that is, it is not ancestral or Markov equivalent to the original MAAGG. The proof for rule R4 is identical to the proof by induction provided in Spirtes et al. [1995], stating that by applying iteratively rule R4 on a PAAGG G oriented using rules CD, CA, KNC, and MR3, the resulting graph G_i at each iteration i maintains its ancestral properties for the equivalence class *O-Equiv*($\mathcal{D}_{\mathbf{O}}$). The proof for the remaining rules is taken from Zhang [2008]:

- R5: The rule states that the path $p = \langle \alpha, \gamma, ..., \theta, \beta, \alpha \rangle$ consists of an uncovered cycles of only circle marks. If we assume instead that a graph G has an arrowhead on this cycle because of KNC, this cycle must be directed to avoid unshielded colliders. But by doing so, the graph is not ancestral;
- R6: Any graph G that contains the opposite orientation than the one stated by the rule, i.e., $\alpha — \beta \leftarrow *\gamma$, is not ancestral;
- R7: Supposed that a graph G has an arrowhead into β as opposed to the rule. Therefore, the triple can be oriented as $\alpha — \beta \leftarrow *\gamma$ or $\alpha \rightarrow \beta \leftarrow *\gamma$. In the former case, G is not ancestral. In the latter, it contains an unshielded collider not present in the original MAAGG;
- R8: If a graph G instead of $\alpha \rightarrow \gamma$ contains $\alpha \leftrightarrow \gamma$, then there is an almost directed cycle or an arrowhead into an undirected edge. In both cases, the graph is not ancestral;
- R9: The same proof for R5 can be applied for this rule;
- R10: The rule states that $\langle \mu, \alpha, \omega, \rangle$ is not a collider in the original MAAGG. Assume that a graph G in the equivalence class contains $\alpha \leftrightarrow \gamma$ instead of the rule specification. Then, for G to be ancestral, one or more edges out of α must be directed. Therefore, to avoid unshielded colliders not in the original MAAGG, p_1 or p_2 must be a directed path, making $alpha$ an ancestor of $gamma$ and thus G not ancestral.

Finally, considering that the rules are proven sound and, as such, all orientations produced are correct, it is straightforward to prove that the respective orientation propagation procedure is sound, following from Maier et al. [2013]. □

The following two lemmas for the arrowhead and tail completeness make use of a representation defined as *chordal graph*, established in Meek [1995] and extended in Maier et al. [2013] for relational data. This representation is an undirected graph where every undirected cycle of length four or more has an edge between two nonconsecutive vertices on the cycle. In chordal graphs, a total order α is consistent with respect to AGG if and only if AGG_α (abstract ground graph in which $A \rightarrow B$ if and only if $A < B$ with respect to α) has no unshielded colliders. Furthermore, for all adjacent vertices A and B, there exists consistent total orderings α and γ such that $A \leftarrow B \in AGG_\alpha$ and $A \rightarrow B \in AGG_\gamma$.

Lemma 2. *Let G be a partially oriented PAAGG with correct adjacencies. Then, exhaustively applying CD, RBO, KNC, CA, MR3, and R4, all with orientation propagation of edges, produces a PAAGG G' in which for every circle mark there exists a MAAGG in the O-Equiv($\mathcal{D}_{\boldsymbol{O}}$) class with a corresponding tail mark.*

Proof. The proof follows from Theorem 4.3 of Ali [2005]. They prove arrowhead completeness for a different graph representation for the Markov equivalence class of MAGs, *Joined Graphs*, which do not distinguish between tail marks and circle marks, provided that the work focused explicitly on arrowhead edge orientations. The same reasoning can be used to ancestral graphs and, with Lemma 1, to PAAGGs. Let G' be the PAAGG with as many edges orientated using CD, RBO, CA, MR3, and R4. For these proofs, we define the edge marker $\otimes$, which corresponds to either a circle or edge mark. There are four steps to prove the arrowhead completeness:

1. Removing any non-directed edge in G' creates a disjoint union of maximal ancestral PAAGGs. Assume for contradiction that the graph G^* obtained by removing undirected edges is not ancestral. Given that G^* does not contain undirected edges, it cannot contain the following configurations: $A\otimes \rightarrow B—C$ or $A* \rightarrow B—C—D \rightarrow A$. Therefore, it contains a partially directed k-cycle such as $X* \rightarrow Y \rightarrow ... \rightarrow Z \rightarrow X$. It can be easily proven that no such cycle can exist without contradiction for $k \geq 3$; therefore, G^* is both ancestral and maximal (Lemma 4.1 of their work that proves that the oriented G' contains only triangles with the following forms:
   (i) $B \ast\!\!\rightarrow A \leftarrow\!\!\ast C* —*B$; (ii) $B* —A —*C* —*B$; or (iii) $Y \ast\!\!\rightarrow A—*C \leftarrow\!\!\ast B$).

2. No replacement of the undirected edges in G' by directed edges will result in non-ancestral structures such as partially directed cycles, unshielded colliders, colliders with order, or inducing paths with non-adjacent endpoints that include an edge oriented by the orientation rules. The absence of these non-ancestral structures is a direct consequence of Lemma 4.1.

3. By removing all directed edges and undirected ones with no parents or spouses from G', the resulting AGG U is a disjoint union of chordal undirected graphs. Assume for contradiction that the orderings of U lead to unshielded colliders. From 2, we know that a replacement of undirected edges could generate a collider with order or inducing paths with non-adjacent endpoints. It's also possible to prove by contradiction that if U is not chordal, then the subgraph U' of the partially oriented PAAGG corresponding to U must contain the same non-chordal properties (i.e., unshielded colliders), which is not possible as U' cannot contain an unshielded collider given the orientation provided by the CD rule. Therefore, U must be chordal.

4. By definition of chordal graph, for every pair (A, B) there are at least two orderings such that $A \rightarrow B$ in one and $A \leftarrow B)$ in the other. Therefore, G' is maximally oriented, and as such, the rules CD, CA, MR3, and R4 are arrowhead complete.

Maier et al. [2013] demonstrates the completeness of the merely common cause rule of RBO, which establishes edge orientation through arrowhead marks only. Consider again the PAAGG G'. Assume by contradiction that there's an edge in G' with a circle mark (without loss of generality, $A \circ\!\!\rightarrow B$), such that there are no MAAGGs in *O-Equiv*($\mathcal{D}_\mathbf{O}$) with a corresponding tail mark for that edge. This requires that the edge mark correspond to an arrowhead in both the equivalence class and the generated PAAGG. Based on the completeness proofs provided above, one of the rules would have orientated that edge mark with an arrowhead. As a result, there must be a MAAGG in *O-Equiv*($\mathcal{D}_\mathbf{O}$) that has the edge $A \rightarrow B$, also known as a tail mark. □

Lemma 3. *Let G' be the partially oriented PAAGG with correct adjacencies and unshielded colliders, and as many edges oriented with KNC, CA, and MR3, all with orientation propagation. Then, applying rules R5-R10, together with orientation propagation, produces a PAAGG G" such that for every circle mark, there exists a MAAGG in O-Equiv($\mathcal{D}_\mathbf{O}$) in which the corresponding mark is an arrowhead.*

Proof. Using Lemma 1, we may follow Zhang [2008] tail completeness proof. We show that any PAAGG edge with a $\circ$ mark (e.g., $\circ\!\!-$, $\circ\!\!-\!\!\circ$, $\circ\!\!\rightarrow$) corresponds to an arrowhead in a MAAGG in the equivalence class. For the first two types of edges ($\circ\!\!-$, $\circ\!\!-\!\!\circ$), we make use of some properties of PAGs, proven in Zhang [2008] and adapted to PAAGGs:

P1 Given a triple A, B, C in a PAAGG, if $A *\!\!\rightarrow B \circ\!\!-\!\!* C$, then there is an edge $A *\!\!\rightarrow C$. In addition, if $A \rightarrow B$, then the edge between A and C cannot be $A \leftrightarrow C$;

P2 Given two vertices, A and B, in a PAAGG, if $A\!-\!\!\circ B$, then there is no edge into A or B;

P3 Given a triple A, B, C in a PAAGG, if $A\!-\!\!\circ B \circ\!\!-\!\!* C$, then there is an edge between A and C. Furthermore, if $A\!-\!\!\circ B \circ\!\!-\!\!\circ C$, then the edge between A and C is $A\!-\!\!\circ C$; if $A\!-\!\!\circ B *\!\!\rightarrow C$, then either $A \rightarrow C$ or $A *\!\!\rightarrow C$;

P4 Given two vertices, A and B, in a PAAGG, if $A\!-\!\!\circ B$, then there is no cycle with the following structure $A\!-\!\!\circ B\!-\!\!\circ \ldots -\!\!\circ A$.

With these properties, it can be proven that:

- For every edge $A \circ\!\!-\!\!\circ B$ in the subgraph obtained by keeping only $\circ\!\!-\!\!\circ$ edges from the PAAGG (which we denote as P^C_{AAGG}), the subgraph can be oriented into two DAGs without unshielded colliders such that $A \rightarrow B$ in one and $A \leftarrow B$ in the other. This is proven by showing that P^C_{AAGG} is chordal: assume by contradiction that there is a non-chordal cycle $\langle X, Y, W, ..., Z\rangle$. This implies that any non-consecutive vertices in the cycle are not adjacent in either P^C_{AAGG} or the original PAAGG, as otherwise they would be connected by a $\circ\!\!-\!\!\circ$ edge (deriving from **P**1 and **P**3) and as such connected in the P^C_{AAGG} as well. Therefore, this non-chordal cycle also appears in the PAAGG, which should have been oriented with rule R5. Therefore the P^C_{AAGG} is chordal.
- Let H be the graph obtained from the following steps applied to the PAAGG:

  1. orient all $\circ\!\!\rightarrow$ and $-\!\!\circ$ edges into directed ones, i.e., $\rightarrow$;
  2. orient the P^C_{AAGG} into a DAG with no unshielded collider.

  Then H belongs to the equivalence class represented by the PAAGG:

  P1-4 ensure that no directed or almost directed cycle is generated after the first step. For step 2, **P**1 and **P**3 ensure that in the P^C_{AAGG} no new directed or almost directed cycles will be generated in H, and furthermore, no new edge into any vertex incident to undirected edges and no inducing paths between any non-adjacent vertices appear. This verifies that H is ancestral and maximal. It is easy to prove then that H belongs to the equivalence class as **P**1-3 guarantee that no new unshielded colliders are created, and as no new bi-directed edges are created also the discriminating path condition for Markov equivalence between H and the PAAGG is verified.

These two theoretical conclusions guarantee that no circle on a PAAGG's $\circ\!\!-$and $\circ\!\!-\!\!\circ$ edges corresponds to an invariant tail. The proof for the $\circ\!\!\rightarrow$ edge comes from Theorem 3 in Zhang [2008], which uses the chordal graph representation established in Meek [1995] and extended in Maier et al. [2013] for relational data. For the PAAGG G'', a proof by contradiction similar to the one provided in Lemma 2 can be carried out for every circle mark corresponding to an arrowhead in at least one MAAGG in the equivalence class *O-Equiv*($\mathcal{D}_{\mathbf{O}}$). □

Theorem 3. *Given a schema and a probability distribution P($\boldsymbol{V}$) with $\boldsymbol{V} = \boldsymbol{O} \cup \boldsymbol{L} \cup \boldsymbol{S}$, the output of RelFCI is a correct maximally informative PAAGG, and thus a maximally informative PARM $\boldsymbol{\mathcal{M}}$, assuming perfect conditional independence tests and sufficient hop threshold h'.*

Proof. The following proof sketch is adapted from Maier et al. [2014]. Given a sufficient h' at least equal to $2h$ (Proposition 4), the set of potential dependencies PDs includes all true dependencies that generate the respective $MAAGG$, which implies the generation of the correct adjacencies, which include the true causes for each relational variable. The unoriented PAAGGs are then constructed using the procedure from Maier et al. [2014]. Assuming perfect conditional independence tests, the algorithm maintains only the correct edges for the PAAGGs. S and U also contain the correct separating sets for every pair of nonadjacent variables and the true unshielded colliders. Next, RelFCI orients all unshielded colliders using either CD or RBO and then, finally, produces a maximally informative PAAGG G and PARM $\boldsymbol{\mathcal{M}}$ as an implication of Theorem 1 and Theorem 2. □

H ADDITIONAL RESULTS

We further evaluated the performance of RelFCI in the absence of latent variables to establish a fair comparison with RCD under causal sufficiency. The experimental setup mirrors that described in Section 5.1, and the results are presented in Figure 19. As shown, RelFCI achieves precision and recall comparable to, and in some configurations slightly exceeding, those of RCD. These results demonstrate that RelFCI maintains high accuracy even when latent confounders are not present, confirming its soundness in recovering the true causal structure in standard relational settings.

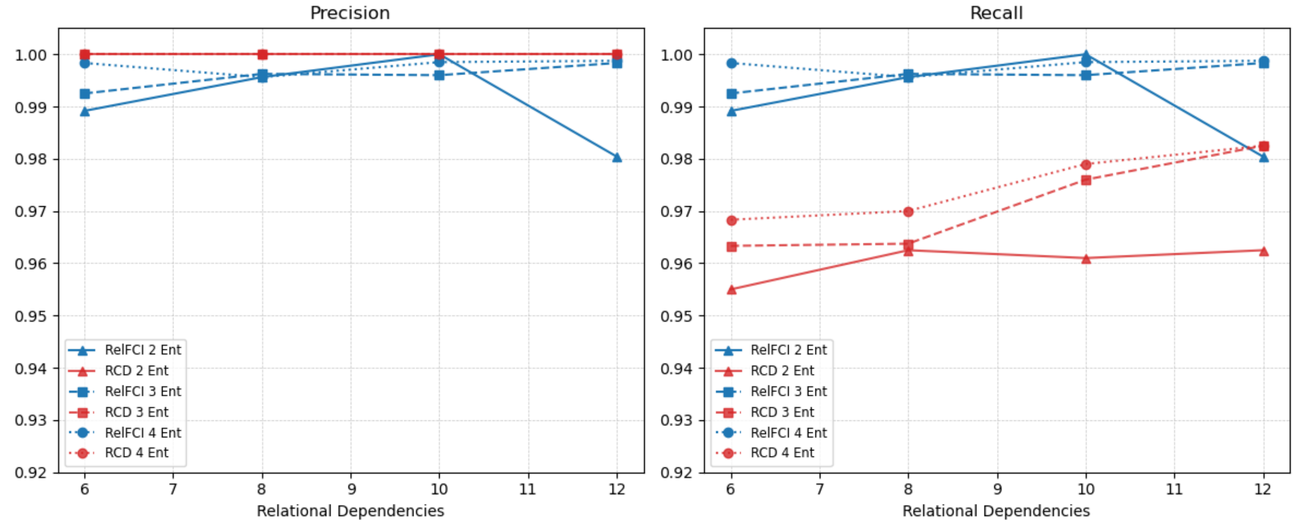


Figure 19: RelFCI Precision and Recall performance with no latent variables.