Towards Robust Relational Causal Discovery

Sanghack Lee
Causal AI Laboratory
Department of Computer Science
Purdue University
West Lafayette, IN 47907
lee2995@purdue.edu

Vasant Honavar
Artificial Intelligence Research Laboratory
College of Information Sciences and Technology
Pennsylvania State University
University Park, PA 16802
vhonavar@psu.edu

Abstract

We consider the problem of learning causal relationships from relational data. Existing approaches rely on queries to a relational conditional independence (RCI) oracle to establish and orient causal relations in such a setting. In practice, queries to a RCI oracle have to be replaced by reliable tests for RCI against available data. Relational data present several unique challenges in testing for RCI. We study the conditions under which traditional iid-based CI tests yield reliable answers to RCI queries against relational data. We show how to conduct CI tests against relational data to robustly recover the underlying relational causal structure. Results of our experiments demonstrate the effectiveness of our proposed approach.

1 INTRODUCTION

Determining causal effects from observations and experiments is a central concern of all sciences, and increasingly, of artificial intelligence, data sciences, and statistics [Pearl, 2000; Spirtes *et al.*, 2000; Rubin, 1974]. Causal inference allows one to elicit causal effects among variables given partial knowledge or assumptions about the data generating process within a domain of interest, often represented by a *causal graph*, a directed acyclic graph where the nodes represent variables of interest and directed edges denote direct causes. Causal discovery is concerned with obtaining causal knowledge by analyzing data obtained from the system of interest (which can be a model, population, or nature). However, because data provides, at best, only partial information about the underlying system, causal assumptions about the world are essential for causal discovery. Consequently, different algorithms for causal discovery often embody different assumptions about the underlying world [Verma and Pearl, 1990; Spirtes *et al.*, 1995].

Most existing causal discovery algorithms are designed to learn a causal graph over the variables V where data consists of independent and identically distributed (iid) instances, where each instance corresponds to an instantiation v of the variables. Conditional independence relations between variables implicit in the data, a sample of the joint distribution $P(v)$, can partially reveal the underlying causal graph. However, data in many real-world settings violate the iid assumption because they are generated by a system of *interacting* objects e.g., a collaboration network, social network, or entities connected by *relations* stored in relational databases. Hence, there is growing interest in methods for learning causal models from relational data. Maier *et al.* [2010] considered a causal model for relational domains, and devised an algorithm called RPC (Relational PC); Maier *et al.* [2013] introduced the Relational Causal Model (RCM), a revised version of their previous model, and proposed the Relational Causal Discovery algorithm, for learning a RCM from data[1]; Lee and Honavar [2016b] introduced RCD-Light, a more efficient version of the RCD algorithm; The same authors [2016a] proposed RpCD, a relational CI oracle based RCD algorithm, that is sound, and unlike RCD and RCD-Light, also *complete*. Unfortunately, this body of work largely falls short of offering a practical solution to RCD. One main reason has to do with the fact that, in practice, the relational CI (RCI) oracle must be replaced by reliable RCI tests; however, most of the existing CI tests do not account for the relational structure underlying relational data, and hence fail to produce reliable answers for RCI queries. Although several CI tests for some types of non-iid data have been proposed in the literature, e.g., the test proposed by Flaxman *et al.* [2016], which has been shown to work well for temporal,

---

[1]Depending on context, we will use RCD, which stands for *relational causal discovery*, to refer to the problem, or the specific solution proposed by Maier *et al.* [2013]

spatial, or undirected graph-structured data, such tests are not directly applicable to relational data. Lee and Honavar [2017a] proposed KRCIT, a suite of graph kernel based relational CI tests, which can reduce the false positive answers to RCI queries resulting from the violation of the iid assumption when a CI test that designed for iid data is naively applied to relational data, they suffer from low power which can result in failure to detect relational CI.

Contributions We propose a relational causal discovery algorithm that effectively works with the available (necessarily imperfect) relational CI tests. Specifically: 1) We identify the conditions under which CI tests that assume iid data can reliably answer relational CI queries, and show how the resulting insights can be exploited by algorithms for RCD; 2) We examine the consequences of replacing a relational CI oracle by relational CI tests from the perspective of relational causal discovery, and propose ways to increase the robustness RCD algorithms that use imperfect relational CI tests.

2 PRELIMINARIES

The Relational Causal Model (RCM) [Maier *et al.*, 2013] marries a relational schema [Chen, 1976] used by relational databases representing the relational structure of the domain with the causal Bayesian network (CBN) [Pearl, 2000] used to represent the structure and parameters of causal models. We borrow many of the notations introduced in the existing literature on RCMs [Maier *et al.*, 2013; Lee and Honavar, 2016a]. We use an uppercase letter, e.g., X, to denote a variable, and the corresponding lower case letter, e.g., x, to denote its realization. We use bold letters, e.g., $\boldsymbol{X}$ or $\boldsymbol{x}$, to represent sets, and calligraphic letters to represent complex mathematical objects. We use the kinship notation, pa, ch, an, de, for graphical relationships such as parents, children, ancestors, descendants. We express the CI statement that the random variables X and Y are conditionally independent (CI) given Z, i.e. that $P(Y|X,Z)$ can be expressed as $P(Y|Z)$, by $X \perp\!\!\!\perp Y \mid Z$. Throughout the paper, we will make use of examples adapted from [Maier, 2014].

Relational Domain A relational schema $\mathcal{S}$ defines how entities interact within a given domain of interest where $\mathcal{S} = \langle \boldsymbol{E}, \boldsymbol{R}, \boldsymbol{A}, \mathsf{card} \rangle$ — a set of entity classes $\boldsymbol{E}$, relationship classes $\boldsymbol{R}$, attribute classes $\boldsymbol{A}$, and cardinality constraints (on the number of entities that can participate in a relationship, i.e, one, many). See Fig. 1a for a concrete example. In this domain, there are 3 entity classes, Employee, Product, and Business Unit (unit for short), and 2 relationship classes, Develops and Funds with their attribute classes shown using rounded rectangles. We will refer to the item classes using the initial letter of their names (E for Employee etc.): E, P, B, D, and F. Small m

near the line between E and D specifies that an employee can develop many products; and a unit can fund many products but a product can be funded by only one unit.

A relational skeleton, denoted by $\sigma \in \Sigma_{\mathcal{S}}$, is one of possible realizations of the given relational schema $\mathcal{S}$ where $\Sigma_{\mathcal{S}}$ is a set of all possible relational skeletons (realizations) of the schema. We denote by $\sigma(B)$ a set of items in σ corresponding to an item class B. Attribute value x of an item i is denoted by $i.x$. Entities and relationships form a bipartite graph satisfying the constraints imposed by the definition of the relationship classes and the cardinality constraints. If $E \in \boldsymbol{E}$ participates in $R \in \boldsymbol{R}$ with cardinality 'one', then $|\{r \mid (e,r) \in \sigma, r \in \sigma(R)\}| \leq 1$ for every $e \in \sigma(E)$. Relational data is then a tuple of the network structure of the relational skeleton and the values of attributes of the items. With the example schema, see Fig. 1b for a relational skeleton where there are 5 employees, 5 products, and 2 units. The illustration hides relationship items: The edge between e_1 and p_1 represents the existence of a relationship item $d_{e_1,p_1} \in \sigma(D)$, which is connected to e_1 and p_1. The cardinality constraints impact the relational skeleton: Some employees develop multiple products and some products are developed by multiple employees; Every product is funded by at most one business unit. By definition, there is no requirement that an entity must participate in any relationship.

Relational Causal Model Relational Causal Model (RCM) $\mathcal{M} = \langle \mathcal{S}, \boldsymbol{D}, \boldsymbol{F} \rangle$ is defined with respect to a given relational schema $\mathcal{S}$ to represent causal relations among attribute classes related via $\mathcal{S}$. It consists of a set of relational dependencies $\boldsymbol{D}$, which represents causal relationships among variables (defined in the relational space, as described below), and a set of functions $\boldsymbol{F}$, which specify how attribute values are generated from their respective causes. Fig. 1c informally illustrates a set of 5 relational dependencies as curved edges. A relational dependency (RD) from competence to salary means that an employee's salary depends on the employee's competence. The dependency from budget to salary implies that an employee's salary (also) depends on the budget of the unit that funds a product developed by the employee. A (stochastic) function specifies how an employee's salary (say, in dollars) can be obtained from such information.

More formally, a *relational dependency* is of the form $U \to V$ where U and V are *relational variables* such that V represents the attribute class of an item class and U is an attribute class defined *relative* to that item class. A relational variable is of the form e.g., $P.X$. As a brief example, aforementioned dependency is expressed as [EDPFB].Budget $\to$ [E].Salary. Such a sequence of item classes, [EDPFB], appearing in a relational variable is called a *relational path*, which is restricted to a *walk*
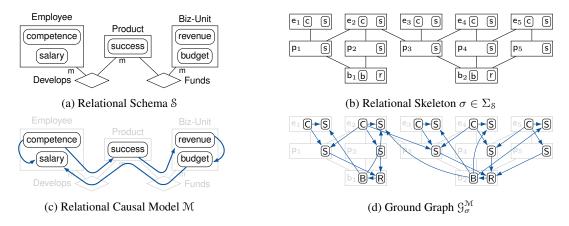
Figure 1: An example of a relational schema, relational skeleton, relational causal model, and ground graph.

(in a graph theoretic sense) on a relational schema from the effect's item class to the cause's item class.[2] A relational path defines the relationship between the attribute classes of item classes that are connected by a relational dependency.[3] The first item class in the path is called a *base* item class (or perspective), and the last item class is called a *terminal* item class. If the relational path of a relational variable is a singleton (which we call *canonical*), we use the following notation, $\mathcal{V}_X = [I_X].X$ where I_X is an item class owning X. A relational dependency, e.g., $P.X \to \mathcal{V}_Y$, implies that the base item class of P is I_Y and the terminal item class is I_X.

A RCM $\mathcal{M}$ is a specification of the causal relationships between the attributes of items of a relational skeleton of a given relational schema. Given a relational skeleton σ, the model $\mathcal{M}$ is instantiated as a *ground graph* $\mathcal{G}_\sigma^{\mathcal{M}}$, which is a CBN made of items' attributes. For instance, there will be a directed edge from $b_2.B$ to $e_2.S$ in Fig. 1d because there exists a path of items $[b_2, f_{p_3,b_2}, p_3, d_{e_2,p_3}, e_2]$ corresponding to the relational path $[BFPDE]$ where f_{p_3,b_2} and d_{e_2,p_3} are implicit in Fig. 1b. Formally, we use $P.X|_i^\sigma$ to denote the multi-set of X of items reachable from item i through a path of items in σ corresponding to P. For instance, $[PDE].C|_{p_3}^\sigma = \{e_2.C, e_3.C, e_4.C\}$. Then, the vertices of $\mathcal{G}_\sigma^{\mathcal{M}}$ correspond to the item attributes and edges are $\{j.X \to i.Y \mid P.X \to \mathcal{V}_Y \in \mathbf{D}, i \in \sigma(I_Y), j \in P|_i^\sigma\}$. A ground graph plays the role of a causal model for the observed relational data. In other words, the attribute values appearing in a relational skeleton corresponds to a single instance sampled from the ground graph. The ground

graph as shown in Fig. 1d is based on Figs. 1b and 1c. Although there are directed edges shown upwards, $\mathcal{G}_\sigma^{\mathcal{M}}$ is acyclic with a topological order as defined by the RCM, i.e., competence, success, revenue, budget, and salary.

Relational Conditional Independence (RCI) RCI (defined below) generalizes CI from the iid setting to the relational setting. Analogous to a CBN embodying a set of CI assertions, a RCM $\mathcal{M}$ embodies a set of RCI assertions. Hence, this set of RCI assertions, if available (e.g., through queries to a RCI oracle or RCI tests against data), allows us to discover the partial structure of the RCM responsible for generating the observed relational data.

Let U, V be relational variables of a given base item class (say B). Let $\mathbf{W}$ be a set of relational variables of the same base item class (B). U and V are said to be RCI given $\mathbf{W}$, denoted by $U \perp\!\!\!\perp V \mid \mathbf{W}$, if and only if $U|_i^\sigma \perp\!\!\!\perp V|_i^\sigma \mid \mathbf{W}|_i^\sigma$ for every relational skeleton $\sigma \in \Sigma_\mathbb{S}$ and every item $i \in \sigma(B)$. By the definition, RCI is a property of a RCM. That is, RCI statement considers a collection of CI statements from *all* possible relational skeletons of a relational schema. However, often, only a single relational skeleton, i.e., a single instance sampled from the ground graph, is available for testing RCI.

The following examples are intended to further illustrate the notion of RCI. The independence between a unit's budget and the unit's employees' competence given the success of the products funded by the unit can be expressed as $[BFPDE].C \perp\!\!\!\perp [B].B \mid [BDP].S$. In contrast, consider a similar statement from a different perspective:

$$[EDPFB].B \not\perp\!\!\!\perp [E].C \mid [EDP].S, \tag{1}$$

which is because we can find a d-connection path, e.g., $e_1.C \to p_1.S \leftarrow e_2.C \to p_2.S \to b_1.R \to b_1.B$ in $\mathcal{G}_\sigma^{\mathcal{M}}$ where $p_1.S$ is a collider. However, as mentioned earlier, it is not feasible to test whether $e_1.C \not\perp\!\!\!\perp b_1.B|p_1.S$ from a single instance of relational data.

---

[2]This formulation is not unlike that found in early probabilistic statistical relational learning literature, e.g., [Koller, 1999].

[3]Restricting the relationship to a path in an underlying skeleton does limit the expressivity of the resulting RCM. However such a restriction simplifies analysis of RCMs and yields a characterization of the equivalence class of a RCM, which in turn leads to a complete RCI-oracle-based RCD algorithm.

3 RELATIONAL CAUSAL DISCOVERY

We first revisit RpCD [Lee and Honavar, 2016a] (See Appendix for the pseudocode), a sound and complete algorithm for learning the structure of a RCM from a given relational schema and access to a RCI oracle. RpCD was inspired by PC [Spirtes *et al.*, 2000] for CBN, and RCD [Maier *et al.*, 2013] for RCM. RpCD consists of two phases where the first phase identifies undirected RDs (i.e., adjacencies) based on answers to RCI queries and the second phase orients (a maximal subset of) the identified RDs based on answers to RCI queries and other known constraints.

Phase I Phase I of RpCD examines undirected RDs of the underlying model. The algorithm starts by enumerating a set of candidate RDs in an undirected form, which is analogous to preparing a complete (undirected) graph for CBN for further processing. Then, it removes undirected RDs of the form $P.Y - \mathcal{V}_X$ from the set of candidate RDs if a separating set S between $P.Y$ and $\mathcal{V}_X$ is found.

Phase II Phase II of RpCD orients a subset of identified undirected RDs based on the answers of queries posed to the RCI oracle. Recall that, in the CBN literature, an undirected path $X - Y - Z$ where X and Z are not adjacent is called an *unshielded triple* (UT). The node Y on the path $X \to Y \leftarrow Z$ is called a collider. If X and Z are not adjacent, Y is called an *unshielded collider*. The PC algorithm orients edges among vertices in an UT $X - Y - Z$ by finding a separating set S between X and Z: $Y \notin S$. Lee and Honavar [2016a] generalized the notion of UTs to the relational setting, and introduced *canonical unshielded triple* (CUT for short) which has testable implications in the underlying RCM:

Definition 1 (Canonical Unshielded Triple). Let $\mathcal{M}$ be a RCM defined on a relational schema $\mathcal{S}$. Suppose $\langle i.X, j.Y, k.Z \rangle$ is an unshielded triple in the ground graph $\mathcal{G}^{\mathcal{M}}_{\sigma}$ for some $\sigma \in \Sigma_{\mathcal{S}}$. There must be two (not necessarily distinct) dependencies $P.Y - \mathcal{V}_X$ and $Q.Z - \mathcal{V}_Y$ of $\mathcal{M}$ (ignoring directions) such that $j \in P|^{\sigma}_i$ and $k \in Q|^{\sigma}_j$. Then, we say that $\langle \mathcal{V}_X, \boldsymbol{P}.Y, R.Z \rangle$ is a *canonical unshielded triple* (CUT) of $\mathcal{M}$ for every $R \in \{T \mid k \in T|^{\sigma}_i\}$ where $\boldsymbol{P} = \{T \mid j \in T|^{\sigma}_i\}$.

If a separating set is found for a CUT through answers RCI queries provided by a RCI oracle, then the edges between the relational variables in a CUT can be oriented in a RCM in a manner analogous to that of UTs in CBN. For example, consider an UT $\langle e_2.C, p_3.S, e_4.C \rangle$ in Fig. 1b based on a relational dependency $[PDE].C \to [P].S$ (where $X = Z$). Then, the corresponding CUT is

$$\langle [E].C, \{[EDP].S\}, [EDPDE].C \rangle. \qquad (2)$$

	[EDPFB].B	[E].C	[EDP].S
e_1	$\{b_1.b\}$	$\{e_1.c\}$	$\{p_1.s\}$
e_2	$\{b_1.b, b_2.b\}$	$\{e_2.c\}$	$\{p_1.s, p_2.s, p_3.s\}$
e_3	$\{b_2.b\}$	$\{e_3.c\}$	$\{p_3.s\}$
e_4	$\{b_2.b\}$	$\{e_4.c\}$	$\{p_3.s, p_4.s\}$
e_5	$\{b_2.b\}$	$\{e_5.c\}$	$\{p_4.s, p_5.s\}$

Table 1: A flattened representation of relational data (Fig. 1b) with respect to a RCI query in Eq. (1).

A separating set S exists such that $[E].C \perp\!\!\!\perp [EDPDE].C \mid S$, and S without $[EDP].S$ indicates $[E].C \to [EDP].S \leftarrow [EDPDE].C$, that is, $[PDE].S \to [E].C$.

RpCD further orients a maximal subset of the rest of the undirected RDs using simple rules that are analogous to those used by PC to orient the edges of a CBN (do not introduce any new unshielded colliders or cycles).

Challenges to be overcome While Lee and Honavar [2016a] showed that RpCD, when given access to a RCM oracle, is guaranteed to yield a correct partially-directed RCM structure, there remain significant hurdles to be overcome before RpCD becomes useful in practice: i) a RCI oracle needs to be replaced with a suitable, sufficiently reliable RCI test. Existing CI tests are either unsuitable for RCI or suffer from low power and hence inability to detect RCI; ii) Even a well-designed RCI test may not be sufficiently reliable when applied to small samples. Incorrect results of RCI tests at early during the execution of the structure learning algorithm may irrecoverably misguide the algorithm. iii) A generic RCI test may fail to account for the specific characteristics of a given relational data, thereby yielding suboptimal results. We address the first challenge in Sec. 4, and the second and third challenges in Sec. 5.

4 TESTING RCI USING A CI TEST

We proceed to consider the implications of using an existing CI test designed for iid data (CI test for short) to reliably answer a RCI query in the context of relational causal discovery using RpCD. Recall that iid data are often stored in a single table where the columns correspond to the variables and rows are populated by (iid) instances. Suppose, given a RCI query, we were to flatten (or propositionalize) the relational data to obtain a RCI query specific single table as follows: To test $P.X \perp\!\!\!\perp Q.Y \mid R.Z$ against relational data, we create a table wherein each row corresponds to a *base item* $i \in \sigma(B)$ of the common base item class B of P, Q, and R and the three columns of the table correspond to $P.X$, $Q.Y$ and $R.Z$ such that the cell for row i and column $P.X$ is a multi-set $P.X|^{\sigma}_i$. Let us call the resulting data flattened data for short. For exam-

ple, Tab. 1 shows a table with three columns constructed to answer a RCI query (Eq. (1)) where the leftmost column corresponds to the row identifier.[4] It is not difficult to observe that the rows of the table constructed using the procedure described above are clearly not independent because, multiple rows of the table, e.g., $P.X|_i^\sigma$ and $P.X|_j^\sigma$, can share the same attributes. Needless to say, flattening does not get rid of the non-iid nature of relational data, which means that, in general, a CI test when applied to the table resulting from the flattening process may incorrectly reject the null hypothesis (independence) although $P.X \perp\!\!\!\perp Q.Y \mid R.Z$. In light of the preceding observation, are there conditions under which a CI test when applied to the table resulting from the RCI query specific flattening process described above is guaranteed to correctly determine whether or not RCI holds? To answer this question, we revisit the Relational Causal Markov Condition (RCMC, Maier [2014]), which states that a canonical relational variable is independent of its non-descendants given its parents. We first recall the definition of non-descendants of a relational variable of a RCM before proceeding to revisit RCMC.

Definition 2. Let RCM $\mathcal{M}$ be defined on a schema $\mathcal{S}$, and let W and $\mathcal{V}_X$ be different relational variables defined on $\mathcal{S}$ sharing a common perspective B. Then, W is *non-descendant* of $\mathcal{V}_X$ if $W|_b^\sigma \cap de\left(b.X; \mathcal{G}_\sigma^\mathcal{M}\right) = \emptyset$ for every $\sigma \in \Sigma_\mathcal{S}$ and $b \in \sigma(B)$.

Definition 3 (Relational Causal Markov Condition). Given a RCM $\mathcal{M}$ defined on a relational schema $\mathcal{S}$, $W|_b^\sigma \perp\!\!\!\perp b.X \mid pa(b.X; \mathcal{G}_\sigma^\mathcal{M})$ for every $X \in \mathbf{A}$, $\sigma \in \Sigma_\mathcal{S}$, and $b \in \sigma(I_X)$ if W is a set of non-descendants of $\mathcal{V}_X$.

RCMC implies that $W \perp\!\!\!\perp \mathcal{V}_X \mid pa(\mathcal{V}_X; \mathcal{M})$. A RCI query of the form $U \perp\!\!\!\perp \mathcal{V}_X \mid \mathbf{Z}$ is said to be RCMC-related if U is non-descendant of $\mathcal{V}_X$, and $\mathbf{Z}$ consists of the parents of $\mathcal{V}_X$ and does not include any non-descendant of $\mathcal{V}_X$. We claim that any RCI query that is RCMC-related can be correctly answered using a CI test (where a random variable can assume values that are multi-sets) applied to the RCI query-specific flattened table constructed using the procedure described above. To see why this claim is true, note that given $pa(\mathcal{V}_X; \mathcal{M})$, attributes of $\mathcal{V}_X$ must be independent and identically distributed, regardless of other conditioned non-descendants. Thus, the variability of $\mathcal{V}_X$ across different conditions arises from external factors that are independent of the non-descendants of $\mathcal{V}_X$. Hence, a traditional CI test applied to the flattened data can accurately answer a RCMC-related RCI query against relational data. For example, consider a generic model where $\mathcal{V}_X \leftarrow f(pa(\mathcal{V}_X; \mathcal{M}), \epsilon)$. Given a fixed value for $pa(\mathcal{V}_X; \mathcal{M})$, $\mathcal{V}_X$ can be viewed

as $g(\epsilon)$ for some function g. Since $g(\epsilon)$ is independent of the non-descendants of $\mathcal{V}_X$, a CI test will correctly assert $W \perp\!\!\!\perp \mathcal{V}_X \mid pa(\mathcal{V}_X; \mathcal{M})$.

Note that although as we showed RCI which is RCMC-related can be correctly answered using a CI test, we can offer no such guarantee in the general case of a RCI query that is not RCMC-related. Specifically, in the general case, such a procedure can fail to establish RCI although RCI holds. Fortunately, however, the violation of iid assumption does not interfere with the CI test rejecting the null hypothesis (independence) when RCI does not in fact hold. Based on this understanding of the conditions under which a CI test can be used to reliably substitute for RCI tests against relational data, we can modify RpCD to substitute RCI oracle with a CI test applied to a RCI-query-specific flattening of the relational data.

5 ROBUST RELATIONAL CAUSAL DISCOVERY

There has been much work on making causal discovery from iid data robust in the presence of limited data or violations of key assumptions [Dash and Druzdzel, 1999; Abellán *et al.*, 2006; Ramsey *et al.*, 2006; Cano *et al.*, 2008; Bromberg and Margaritis, 2009], including on methods that take advantage of recent advances in general-purpose Boolean satisfiability solvers [Hyttinen *et al.*, 2013; Triantafillou and Tsamardinos, 2015; Magliacane *et al.*, 2016]. Hence, in what follows, we focus our discussion primarily on approaches to making causal discovery robust that are specific to the relational (as contrasted with the iid) setting. We proceed to consider the two key phases of RpCD in turn.

5.1 PHASE I: IDENTIFYING ADJACENCIES

Recall that RpCD starts by initializing a set of candidate relational dependencies (RDs) given a user-specified maximum hop length of RDs to be considered. Let $\mathcal{M}'$ be an intermediate RCM at an intermediate step during the execution of phase I of RpCD. In light of the results of the previous section regarding the conditions under which RCI tests against relational data can be reliably substituted by CI tests against an appropriate flattening of the relational data, we can ensure that RpCD first asks RCI queries that match RCMC to eliminate spurious candidate dependencies, while retaining the genuine dependencies. Recall that RpCD performs RCI tests to determine whether a candidate neighbors of a canonical relational variable (CRV) is in fact a genuine neighbor. Since at the outset, the candidate neighbors of a CRV include the genuine parents of the CRV in the RCM, RpCD will eventually test $(P.X \perp\!\!\!\perp \mathcal{V}_Y \mid pa(\mathcal{V}_Y; \mathcal{M}'))$ for any connection

---

[4]If a cell for $P.X$ or $Q.Y$ is empty, we discard the corresponding row from the table.

$P.X - \mathcal{V}_Y$ unless it is disconnected with a separating set other than the parents of the CRV. Note that any incorrect answers to the RCI queries (i.e., incorrect rejections of the null hypothesis (independence) by the CI tests) do not adversely impact the correctness of the algorithm unless the RCI query matches RCMC. However, there is the possibility of incorrectly discarding true RDs. For example, it is possible that $P.X \to \mathcal{V}_Y \in \boldsymbol{D}$ might be discarded from the candidate list due to relatively weak dependence. We proceed to examine a way to contain the deleterious impact of incorrectly discarding true dependencies.

Order-independence Whenever a true RD is discarded, it has a cascading deleterious effect on future steps of the PC algorithm (and its variants) which, as shown by Colombo and Maathuis [2014], can however be avoided by making the necessary modifications to render the algorithms independent of the order in which variables are considered. Such modifications can be directly incorporated into RpCD: Prepare an empty set, store dependencies to be removed in the set instead of removing them immediately, and remove dependencies in the set when the algorithm proceeds to consider larger conditionals.

Asymmetry and Aggregation There exists a notable difference between an edge in a CBN and a RD in a RCM with respect to the test for adjacencies — there can be two different tests for an adjacency.[5] Through Phase I, we seek to ensure that the following holds true $(P.X \not\perp\!\!\!\perp \mathcal{V}_Y \mid \boldsymbol{W})$ for $P.X \to \mathcal{V}_Y \in \boldsymbol{D}$ for a set of RVs $\boldsymbol{W}$ with base item class I_Y and $P.X \notin \boldsymbol{W}$. The test for the same adjacency must be performed from a different perspective I_X, $(\tilde{P}.Y \not\perp\!\!\!\perp \mathcal{V}_X \mid \boldsymbol{R})$ where $\tilde{P}$ corresponds to P reversed. Performing both tests is essential since the algorithm does not know in advance the topological order between X and Y, and which RCI queries are in fact RCMC-related.

Consider a RD, [PDE].Competence $\to$ [P].Success. The dependency between [EDP].S and [E].C can be substantially weaker than the dependency between [PDE].C and [P].S since there is a set of coworkers whose competence affects [EDP].S that is not considered. For instance, e_2 develops p_1, p_2, and p_3 where the success of p_1 and p_3 are also determined by e_1, and e_3 and e_4, respectively, diluting the strength of the relationship between competence of the worker(s) and the success of the product.

To protect against the possibility that a RCI test wrongly failing to reject independence, e.g., $(P.X \perp\!\!\!\perp \mathcal{V}_Y \mid \boldsymbol{W})$, we can perform an *additional* test. We can first apply an aggregating function (e.g., mode, average, median, etc), on $P.X \in adj\,(\mathcal{V}_Y; \mathcal{M}')$, and then conduct an additional

---

[5]Arbour *et al.* [2016] considered asymmetry in relational data to infer the orientation of an underlying dependency while our focus is to improve the power of CI test.

test which (modulo slight abuse of notation) is given by

$$(f\,(P.X) \perp\!\!\!\perp \mathcal{V}_Y \mid \boldsymbol{W})_\sigma$$

where each $P.X|_i^\sigma$ is replaced by $f\,(P.X|_i^\sigma)$. Such aggregation does *not* introduce spurious dependencies. However, in practice, it can help overcome weak RCI tests as the mapping reduces not only the dimensionality of the variables involved in the test but also the variances caused by exogenous variables.

It is worth noting that aggregation is widely used for dealing with RVs in relational machine learning [Perlich and Provost, 2006] as well as relational causal discovery [Maier *et al.*, 2013]. However, an important distinction between their use of aggregation and ours is that we do not apply an aggregate function on the conditionals, for doing so may result in false positive answers to RCI queries. For example, consider $X \to Z \to Y$ where $X \perp\!\!\!\perp Y \mid Z$. If we transform only X using the aggregation function f, we get $f(X) \leftarrow X \to Z \to Y$ where $f(X) \perp\!\!\!\perp Y \mid Z$ still holds true. If Z is transformed through the aggregation function g, we get $X \to Z \to Y$ with $Z \to g(Z)$, and, thus, $X \not\perp\!\!\!\perp Y \mid g(Z)$ nor $f(X) \not\perp\!\!\!\perp Y \mid g(Z)$. Note that aggregation presented here also applies to Phase II of the algorithm described in the next subsection.

5.2 PHASE II: ORIENTING RELATIONAL DEPENDENCIES

A RCI test against a canonical unshielded triple (CUT) can establish the orientation of edges among the vertices in the triple. Since the purpose of the test is to find a separating set, false positives for non-RCMC-related queries are a non-issue. However, weak dependence can lead to false negatives, and hence an invalid separating set, which may include colliders or may exclude non-colliders, resulting in an incorrect orientation of the edges among the vertices in a CUT.

Since RCM assumes acyclicity at an attribute class level, once we perform a RCI test on a CUT $\langle \mathcal{V}_X, \boldsymbol{P}.Y, R.Z \rangle$, assuming that the test is reliable, there is no need to test on other CUTs with matching attribute classes i.e., $\langle X, Y, Z \rangle$ (or $\langle Z, Y, X \rangle$). However, given the possibility of erroneous results from RCI tests, we can perform tests on multiple CUTs to determine the orientation of an edge, e.g., $X \to Y \leftarrow Z$, and make an informed decision by, e.g., majority rule, two-thirds, etc, based on the results of multiple tests. The algorithm may even obtain multiple separating sets against a CUT and check them for consistency (e.g., orientation-faithfulness [Ramsey *et al.*, 2006]). However, as we shall show below, care must be exercised in how these ideas are incorporated into RpCD.

Limitations of CUT-based RCI tests We start by examining why *naively* conducting RCI tests against CUTs is

not the best idea. To see why, consider the flattening of the CUT in Eq. (2) for answering a RCI query using a CI test. Since e_2 develops three products $\{p_1, p_2, p_3\}$, e_2's coworkers are $\{e_1, e_3, e_4\}$. Success of p_2 is, in fact, irrelevant to the competences of the coworkers $\{e_1, e_3, e_4\}$, whereas the success of each product depends only on the competence of the employees who develop it (e.g., e_1 in the case of p_1). (i) Given a CUT $\langle \mathcal{V}_X, \boldsymbol{P}.Y, R.Z \rangle$ and some $P \in \boldsymbol{P}$, the association between $P.Y$ and $R.Z$ seems relatively weaker than that between two RVs connected by a RD. (ii) Further, R can be long, and the average dimensionality of $\{R|_i^\sigma\}_{i \in \sigma(I)}$ can be large. If each employee develops k products, and each product is developed by m employees, then, each employee can have up to $km - 1$ coworkers. (iii) Unlike the unshielded triples of a CBN, absence of a RD between $\mathcal{V}_X$ and $R.Z$ does not imply that there is no connection between $i.X$ and $R.Z|_i^\sigma$ for some item i. If $R'.Z \to \mathcal{V}_X \in \mathbf{D}$, then there will be edges from $i.X$ to $R.Z|_i^\sigma \cap R'.Z|_i^\sigma$.

In light of the preceding observations we consider alternative ways to perform RCI tests that are relevant to CUTs. We can classify RCI tests for orientating the RDs into the following categories: (i) relational bivariate orientation (RBO) which can be further subdivided into split-RBO and pair-RBO, and (ii) non-RBO.[6] We proceed to discuss each of these in turn.

Split-RBO Relational Bivariate Orientation (RBO) is an orientation for a CUT where the two ends share the same attribute class. Given an undirected RD $P.X - \mathcal{V}_Y$ where P is of cardinality 'many', a CUT can be made $\langle \mathcal{V}_X, \tilde{\boldsymbol{P}}.Y, R.X \rangle$ where one can orient $X \to Y$ if it turns out to be a collider or $X \leftarrow Y$, otherwise. See Fig. 2 where an item attribute $i.Y$ has three neighbors for $P.X|_i^\sigma$ in an intermediate undirected ground graph. From the perspective of $i.Y$, $P.X|_i^\sigma$ can be split into two parts: a singleton (one) and the rest of the item attributes (rest). Then, a separating set will be obtained from a subset of neighbors of $\mathcal{V}_X$ in an intermediate model. The corresponding test can be expressed as

$$\text{one}(P.X) \perp\!\!\!\perp \text{rest}(P.X) \mid \boldsymbol{S} \tag{3}$$

where $\boldsymbol{S} \subseteq adj(\mathcal{V}_X; \mathcal{M}')$. To carry out the test, we construct a flattened representation as follows. Among $\sigma(I_Y)$, find items that are connected to at least two I_X items through P in σ. For each element $j.X \in P.X|_i^\sigma$, create a row $\langle j.X, P.X|_i^\sigma \setminus \{j.X\} \rangle$. Then, refine the resulting table to ensure that the $j.X$ column is made of unique item attributes. Columns for $\boldsymbol{S}$ can be added later.

---

[6]RBO is proposed as a special rule in Maier *et al.* [2013] where a non-collider e.g., not $X \to Y \leftarrow X$ implies $Y \to X$. We rather use the term RBO not as an "orientation rule" but as a way to categorize CUT-based RCI tests.

	[PDE].C	[PDE].C $\setminus$ e$_i$.C	e$_i$.C
p$_1$	$\{e_1.c, e_2.c\}$	$\{e_1.c, e_2.c\}$	$e_1.c$
		$\{e_1.c, e_2.c\}$	$e_2.c$
p$_3$	$\{e_2.c, e_3.c, e_4.c\}$	$\{e_2.c, e_3.c, e_4.c\}$	$e_2.c$
		$\{e_2.c, e_3.c, e_4.c\}$	$e_3.c$
		$\{e_2.c, e_3.c, e_4.c\}$	$e_4.c$
p$_4$	$\{e_4.c, e_5.c\}$	$\{e_4.c, e_5.c\}$	$e_4.c$
		$\{e_4.c, e_5.c\}$	$e_5.c$

Table 2: Flattened representation (before deduplication) for split-RBO test ([PDE].C for reference)
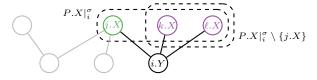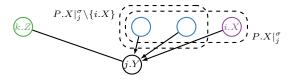

Figure 2: An instance in row-j for split-RBO where three variables are split so as to create three rows. CUT-based tests will include gray vertices.

As an example, consider a representation to be used for split-RBO with respect to [EDP].S $-$ [P].C (Tab. 2). Note that there are multiple employees connected to a product. For each such product, we can list product-specific coworkers who can be split into a singleton and the rest. One might wonder if we can get by with using an employee and only one of his/her coworkers. Consider Fig. 2 again where $P.X$ is the cause of $\mathcal{V}_Y$. Assume, for simplicity, $i.Y = \sum_{j.X \in P.X|_i^\sigma} j.x$. If we were to consider only two singletons instead of one singleton and the rest, the relationship between the two singletons with respect to their common effect will become low.

Pair-RBO Since split-RBO already covered a case where P is of cardinality 'many', we now consider a type of RBO with two different candidate undirected RDs $P.X - \mathcal{V}_Y$ and $Q.X - \mathcal{V}_Y$ where both P and Q are of cardinality 'one'. Due to their cardinalities, P and Q are not intersectable, and every item attribute $j.Y$ for $j \in \sigma(I_Y)$, there can be at most two item attributes of X representing $P.X$ and $Q.X$, i.e., $|P.X|_i^\sigma| \leq 1$ and $|Q.X|_i^\sigma| \leq 1$. Hence, a separating set is obtained for pairs of singletons. If multiple RDs are defined for two attribute classes X and Y, multiple pair-RBO and split-RBO tests can be used to orient them. Techniques described in Colombo and Maathuis [2014] for handling conflicting orientations in a CBN can be adopted to RCM in a relatively straightforward manner.

Non-RBO Now, we seek for a principled approach to orienting RDs when three different attribute classes are involved in a CUT, say $\langle \mathcal{V}_X, \tilde{\boldsymbol{P}}.Y, R.Z \rangle$ with $P.X - \mathcal{V}_Y$ and $Q.Z - \mathcal{V}_Y$ such that $X \neq Z$. Performing RBO tests

Figure 3: Non-RBO with $P.X \to \mathcal{V}_Y$ and $Q.Z - \mathcal{V}_Y$

Aggregation	Order Ind.	Base size	Precision	Recall
False	False	200	98.23	61.91
		500	98.90	77.92
	True	200	98.86	57.95
		500	99.13	76.77
True	False	200	97.75	65.44
		500	98.87	80.31
	True	200	98.75	61.77
		500	99.09	79.03

Table 3: Precision and recall (based on macro-average) for discovering undirected RDs in Phase I.

before performing any non-RBO tests offers several advantages: (i) If RBO-tests between X and Y fail to orient a RD, we can conclude that X and Y exhibit weak dependence in relational data, in which case, it is advisable to avoid non-RBO tests that involve X and Y, which may result in incorrect orientations; and (ii) If some of the RDs are oriented (e.g., $X \prec Z$), then we can limit the tests to be performed to those that seek a separating set only from $\mathcal{V}_Z$. Further, a separating set $\boldsymbol{S}$ will be $pa(\mathcal{V}_Z; \mathcal{M}') \subseteq \boldsymbol{S} \subseteq adj(\mathcal{V}_Z; \mathcal{M}')$, which can reduce the number of tests needed to obtain a separating set. (iii) At least one of P and Q is of cardinality 'one', permitting us to use a CI test (on suitably flattened data) in place of non-RBO RCI tests. If both are of cardinality 'one', non-RBO tests can be done in a manner similar to pair-RBO. (iv) Finally, if Q be of cardinality 'one', then the orientation between X and Y would have already been determined (because a split-RBO test, it would precede a non-RBO test). Further, if $\tilde{P}.Y \to \mathcal{V}_X$ then, no RCI test is required since the CUT is a non-collider. Hence, the only case left for non-RBO test is $P.X \to \mathcal{V}_Y - Q.Z \in \mathcal{M}'$ with P being of cardinality 'many'. Without knowing whether $X \not\prec Z$ nor whether $Z \not\prec X$, we may need to examine separating sets from both I_X and I_Z perspectives. First, from I_Z perspective, RCI test can be performed in a table where row-j for $j \in \sigma(I_Y)$ represents $\langle k.Z, P.X|_j^\sigma, \dots, S^\ell|_k^\sigma, \dots \rangle$ where $\{k\} = Q|_j^\sigma$ and $pa(\mathcal{V}_Z; \mathcal{M}') \subseteq \boldsymbol{S} \subseteq adj(\mathcal{V}_Z; \mathcal{M}')$. With I_X perspective, we should pick one of $P.X|_j^\sigma$, i.e., $\mathsf{one}(P.X)$, for row-j and test against $Q.Z|_j^\sigma$, which is a singleton. Similarly, a separating set $\boldsymbol{S}$ satisfies $pa(\mathcal{V}_X; \mathcal{M}') \subseteq \boldsymbol{S} \subseteq adj(\mathcal{V}_X; \mathcal{M}')$.

Detecting Weak Dependence Ramsey *et al.* [2006] explored techniques for determining whether CI tests against the given data yield consistent orientations of edges in a CBN. Given an unshielded triple $X - Y - Z$, one can examine whether Y appears in every $\boldsymbol{S} \subseteq adj(\{X, Z\}; \mathcal{G})$ such that $X \perp\!\!\!\perp Z \mid \boldsymbol{S}$ and not in $\boldsymbol{W} \subseteq adj(\{X, Z\}; \mathcal{G})$ such that $X \not\perp\!\!\!\perp Z \mid \boldsymbol{W}$. However, because this not only is time consuming and but also yields very conservative results, a more pragmatic alternative is to examine whether both $(X \perp\!\!\!\perp Z \mid \boldsymbol{S})$ and $(X \perp\!\!\!\perp Z \mid \boldsymbol{S} \cup \{Y\})$ result independence for some separating set $\boldsymbol{S}$ which does not include Y. The idea can be incorporated into RpCD where one can check, for example, with conditionals $\boldsymbol{S} \cup \{\tilde{P}.Y\}$ in testing against a CUT $\langle \mathcal{V}_X, \tilde{\boldsymbol{P}}.Y, Q.Z \rangle$ where $\tilde{P} \in \tilde{\boldsymbol{P}}$. In contrast, given our approaches (i.e., split-, pair-, and

non-RBO), a better test can be achievable by limiting the item attributes of Y to only those that are relevant to $P.X$, which is $\mathcal{V}_Y$. For split-RBO with a separating set $\boldsymbol{S}$, the following test can be performed,

$$\mathsf{one}(P.X) \perp\!\!\!\perp \mathsf{rest}(P.X) \mid \boldsymbol{S} \cup \{\mathcal{V}_Y\}.$$

This detection mechanism can be applied to pair-RBO and non-RBO cases.

6 EXPERIMENTS AND RESULTS

Experiments We conducted experiments with synthetic data generated from known RCMs to assess how the proposed approaches to replacing RCI oracle with RCI tests against relational data impact the performance of RpCD. In our implementation, we used two kernel-based CI tests, HSIC [Gretton *et al.*, 2005] for marginal, and SDCIT [Lee and Honavar, 2017b] for conditional RCI queries. In our kernel-based CI test for multi-set valued random variables, we used, following Haussler [1999], $K'(\boldsymbol{x}, \boldsymbol{y}; \theta) = \sum_{x \in \boldsymbol{x}} \sum_{y \in \boldsymbol{y}} K(x, y; \theta)$. In the case of real-valued data, K is chosen to be a RBF kernel whose parameter θ is chosen using the median heuristic [Gretton *et al.*, 2007]. The resulting kernel matrices are normalized, e.g., $\boldsymbol{K}_{a,b} \leftarrow \boldsymbol{K}_{a,b} / \sqrt{\boldsymbol{K}_{a,a} \boldsymbol{K}_{b,b}}$.

We tested the performance of RpCD with the improvements proposed in this paper and the baseline (RpCD using the CI test for RCI with no other changes) on 300 randomly generated RCMs of varying complexity. For each RCM, we randomly generated different sizes of relational data with $n = 200$ to 500 resulting in approximately n items per entity class and $2n$ relationships per relationship class. We parametrized the RCM using an adaptation of a linear Gaussian model to a relational setting. We used Average as the aggregation function. Additional details about the experimental setup and results are provided in the Appendix.[7]

---

[7]Code is available at `https://github.com/sanghack81/RRCD`

	200			500		
	Acc.	C	NC	Acc.	C	NC
CUT (RBO)	54.0	46.1	60.7	54.1	46.4	60.6
P+S	68.7	64.7	72.6	75.6	73.4	77.9
w/ detection	71.2	68.4	74.0	77.4	74.7	80.0
CUT	65.5	77.0	61.4	74.1	77.8	72.8
P+S+N	74.0	73.2	74.7	80.4	78.2	82.2
w/ detection	80.9	75.0	85.4	85.7	82.6	88.2

Table 4: Accuracies for orientation tests (overall (Acc.), collider (C), and non-colliders (NC)), for CUT-based tests, proposed tests, and with the weak dependency detection mechanism enabled. P, S, and N stands for Pair-RBO, Split-RBO, and Non-RBO, respectively.

Phase I Experimental Results We find that (see Tab. 3), as the size of relational data increases, the performance of Phase I improves as expected. Order-independence mitigates the effect of early false negative RCI test results, perhaps at the expense of a slightly reduced recall of undirected relational dependencies. Aggregation improves the power of the test at the expense of a slight increase in the false positive rate of the test. Note that the high precision and relatively low recall implies that errors of RCI tests are mainly false negatives.

We additionally investigated the types of queries where relational conditional independence is correctly found to not hold by the additional aggregation-based tests. Aggregation was especially effective in reducing the false negative rate of the tests of independence between a canonical RV and its child (tests in a reverse direction) while rarely producing false positives (see Appendix).

Phase II Experimental Results Given the correct set of dependencies (which correspond to perfect Phase I results), we first performed experiments to measure the effectiveness of split-, pair-, and non-RBO tests relative to the CUT-based tests. Tab. 4 shows the performance based on the first smallest separating set found. CUT-based tests do not perform well even with larger relational data (which also makes relational structure more complicated) for RBO cases. Proposed RBO tests outperform CUT-based tests regardless of the type of orientations, colliders or non-colliders, and show improvements with larger data. Additional weak dependency detection mechanism helps refining false negatives. A similar trend is observed when we also considered non-RBO cases. Note that, unlike Phase I, false negatives in Phase II might cause wrong orientations, thus, affecting both precision and recall.

Next, we compared our approach against a naive CUT-based approach by measuring the average precision and recall for the final orientations with respect to the true

CPRCM[8] instead of the RCM. The final orientations for our approach were determined as follows: i) a majority vote rule is used to determine the orientation of each attribute class triple (local)[9]; ii) the maximal non-conflicting local orientations are obtained (global). The baseline with CUT-based tests used the same majority rule but each orientation is accepted in a sequential manner if it does not cause conflicts with the already accepted orientations. Given the perfect Phase I results, the precision and recall for Phase II based on our approach are 93.5% and 75.4%, respectively ($n = 500$), as compared to 75.3% and 69.7%, respectively, for the CUT-based Phase II. Thus, we see substantial improvements over the baseline.

7 SUMMARY AND DISCUSSION

We introduced a robust algorithm for learning the structure of a relational causal model from the given relational data. We showed how a conditional independence test designed for iid data can be used to effectively test for relational conditional independence against relational data. The relational causal Markov condition, a relational variable being independent of its non-descendants given its direct causes, allows the test to correctly establish relational conditional independence, whereas the non-iid-ness of relational data helps the test to reject independence when independence does not hold. We introduced several techniques to improve the robustness of the algorithm, and empirically demonstrated their effectiveness. Despite these promising results, there is much room for further improvement, through better methods for testing independence of variables whose values are multi-sets, kernels optimized for the given relational data, as well as improved tests for relational conditional independence.

---

[8]A CPRCM is a maximally-oriented RCM, which represents the Markov equivalence class of a RCM.

[9]Once a separating set of size k is found, then we further examine whether there are other separating sets with size k, while a naive approach only makes use of RCI results based on the first separating set found.

References

J. Abellán, M. Gómez-Olmedo, and S. Moral. Some variations on the PC algorithm. In *Proceedings of the Third European Workshop on Probabilistic Graphical Models (PGM' 06)*, pages 1–8, 2006.

David Arbour, Katerina Marazopoulou, and David Jensen. Inferring causal direction from relational data. In *Proceedings of the 32nd Conference on Uncertainty in Artificial Intelligence*, pages 12–21, 2016.

Facundo Bromberg and Dimitris Margaritis. Improving the reliability of causal discovery from small data sets using argumentation. *Journal of Machine Learning Research*, 10:301–340, 2009.

Andrés Cano, Manuel Gómez-Olmedo, and Serafín Moral. A score based ranking of the edges for the PC algorithm. In *Proceedings of the Fourth European Workshop on Probabilistic Graphical Models (PGM 2008)*, pages 41–48, 2008.

Peter Pin-Shan Chen. The entity-relationship model – toward a unified view of data. *ACM Transactions on Database Systems (TODS)*, 1(1):9–36, 1976.

Diego Colombo and Marloes H Maathuis. Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 15:3921–3962, 2014.

Denver H Dash and Marek J Druzdzel. A hybrid anytime algorithm for the construction of causal models from sparse data. In *Proceedings of the Fifteenth Annual conference on Uncertainty in Artificial Intelligence*, pages 142–149, 1999.

Seth R Flaxman, Daniel B. Neill, and Alexander J. Smola. Gaussian processes for independence tests with non-iid data in causal inference. *ACM Transactions on Intelligent Systems and Technology*, 7(2):1–23, nov 2016.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *Algorithmic Learning Theory. ALT 2005*, pages 63–77. Springer, 2005.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A kernel method for the two-sample problem. In *Advances in Neural Information Processing Systems 19*, pages 513–520. MIT Press, 2007.

David Haussler. Convolution kernels on discrete structures. Technical report, University of California, Santa Cruz, 1999.

A. Hyttinen, P. O. Hoyer, F. Eberhardt, and M. Järvisalo. Discovering cyclic causal models with latent variables: A general SAT-based procedure. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 301–310. AUAI Press, 2013.

Daphne Koller. Probabilistic Relational Models. In *Proceedings of the 9th International Workshop on Inductive Logic Programming*, pages 3–13, 1999.

Sanghack Lee and Vasant Honavar. A characterization of Markov equivalence classes of relational causal models under path semantics. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, pages 387–396, Corvallis, Oregon, 2016. AUAI Press.

Sanghack Lee and Vasant Honavar. On learning causal models from relational data. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 3263–3270, Palo Alto, CA, 2016. AAAI Press.

Sanghack Lee and Vasant Honavar. A kernel conditional independence test for relational data. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*, Corvallis, Oregon, 2017. AUAI Press.

Sanghack Lee and Vasant Honavar. Self-discrepancy conditional independence test. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*, Corvallis, Oregon, 2017. AUAI Press.

S. Magliacane, T. Claassen, and J.M. Mooij. Ancestral causal inference. In *Advances In Neural Information Processing Systems 29*, pages 4466–4474. Curran Associates, Inc., 2016.

Marc Maier, Brian Taylor, Hüseyin Oktay, and David Jensen. Learning causal models of relational domains. In *Proceedings of the Twenty-Fourth National Conference on Artificial Intelligence*, pages 531–538, 2010.

Marc Maier, Katerina Marazopoulou, David Arbour, and David Jensen. A sound and complete algorithm for learning causal models from relational data. In *Proceedings of the Twenty-ninth Conference on Uncertainty in Artificial Intelligence*, pages 371–380, Bellevue, WA, July 2013. AUAI Press.

Marc E Maier. *Causal Discovery for Relational Domains: Representation, Reasoning, and Learning*. PhD thesis, University of Massachusetts Amherst, 2014.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

Claudia Perlich and Foster Provost. Distribution-based aggregation for relational learning with identifier attributes. *Machine Learning*, 62(1-2):65–105, 2006.

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

Donald B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66(5):688–701, 1974.

P. Spirtes, C. Meek, and T. Richardson. Causal inference in the presence of latent variables and selection bias. In P. Besnard and S. Hanks, editors, *Uncertainty in Artificial Intelligence 11*, pages 499–506. Morgan Kaufmann, San Francisco, 1995.

Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, second edition, 2000.

S. Triantafillou and I. Tsamardinos. Constraint-based causal discovery from multiple interventions over overlapping variable sets. *Journal of Machine Learning Research*, 16:2147–2205, 2015.

T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence (UAI 1990)*, pages 220–227, Cambridge, MA, July 1990.

Appendix

RPCD ALGORITHM

We present RpCD algorithm in Alg. 1.

Algorithm 1 RpCD [Lee and Honavar, 2016a]

Input: $\mathbb{S}$ relational schema, h hop threshold

1: initialize $\boldsymbol{D}$ with candidate RDs up to h hops.
2: initialize an undirected graph $\mathcal{M}'$ with undirected $\boldsymbol{D}$.
3: $\ell \leftarrow 0$
4: **repeat**
5:   **for** $(P.Y, \mathcal{V}_X)$ **s.t.** $P.Y - \mathcal{V}_X \in \mathcal{M}'$ **do**
6:     **for** every $\boldsymbol{S} \subseteq ne(\mathcal{V}_X; \mathcal{M}') \setminus \{P.Y\}$ **s.t.** $|\boldsymbol{S}| = \ell$ **do**
7:       **if** $P.Y \perp\!\!\!\perp \mathcal{V}_X \mid \boldsymbol{S}$ **then**
8:         remove $\{P.Y - \mathcal{V}_X, \tilde{P}.X - \mathcal{V}_Y\}$ from $\mathcal{M}'$.
9:         **break**
10:   $\ell \leftarrow \ell + 1$
11: **until** $|ne(\mathcal{V}_X; \mathcal{M}')| - 1 < \ell$ for every $X \in \boldsymbol{A}$

12: initialize $\mathcal{U}$ with CUTs from $\mathcal{M}'$.
13: $\mathcal{N} \leftarrow \emptyset, \mathcal{H} \leftarrow \langle \boldsymbol{A}, \{X - Y \mid P.Y - \mathcal{V}_X \in \mathcal{M}'\}\rangle$
14: **for** every $\langle \mathcal{V}_X, \boldsymbol{P}.Y, R.Z\rangle \in \mathcal{U}$ **do**
15:   **if** $\langle X, Y, Z\rangle \in \mathcal{N}$ **or** $\{X, Z\} \cap ne(Y; \mathcal{H}) = \emptyset$ **or** $\{X, Z\} \cap ch(Y; \mathcal{H}) \neq \emptyset$ **then**
16:     **continue**
17:   **if** exists $\boldsymbol{S} \subseteq adj(\mathcal{V}_X; \mathcal{M}')$ **s.t.** $R.Z \perp\!\!\!\perp \mathcal{V}_X \mid \boldsymbol{S}$ **then**
18:     **if** $\boldsymbol{S} \cap \boldsymbol{P}.Y = \emptyset$ **then** orient $X \rightarrow Y \leftarrow Z$ in $\mathcal{H}$
19:     **else if** $X = Z$ **then** orient $Y \rightarrow X$ in $\mathcal{H}$
20:     **else** add $\langle X, Y, Z\rangle$ to $\mathcal{N}$
21:   orient edges in $\mathcal{H}$ with sound rules with $\mathcal{N}$.

22: $completes\,(\mathcal{H}, \mathcal{N})$

23: **return** $\bigcup_{P.Y - \mathcal{V}_X \in \mathcal{M}'} \begin{cases} P.Y \rightarrow \mathcal{V}_X & Y \rightarrow X \in \mathcal{H} \\ P.Y - \mathcal{V}_X & Y - X \in \mathcal{H} \end{cases}$

RELATIONAL DATA

We randomly generated 300 relational schemas of 3 (50%), 4 (25%), and 5 (25%) entity classes with specified probabilities. Two to five relationship classes are randomly generated to connect a pair (i.e., binary relationship) of entity classes or a triple with 75% and 25% probability, respectively. Cardinalities are selected uniformly. One to three attribute classes are generated for each entity class, and zero or one attribute class is generated for each relationship class. Finally, created relational schemas that do not satisfy following rules are excluded: (i) all item classes are connected and (ii) the total number of attribute classes are less than or equal to 8.

RCMs are also generated randomly. Given a relational schema $\mathbb{S}$, max hop length h is selected uniformly between 2 to 4. The number of dependencies is determined by $\lfloor \frac{3|\boldsymbol{A}|}{2} \rfloor$ and uniformly selected among *all* relational dependencies within the given h. We limit the maximum number of parents of a canonical relational variable by

3. We reject generated RCMs if there exists an isolated attribute class that does not involve any relational dependency. Further, if the CPRCM (a maximally-oriented RCM representing the Markov equivalence class of a RCM) of the generated RCM has no directed dependencies, that is, the orientation of relational dependencies is impossible in theory. We adopt a linear model with additive Gaussian noise using average aggregators:

$$i.X \leftarrow \sum_{P.Y \in pa(\mathcal{V}_X; \mathcal{M})} \frac{\beta_{P.Y, \mathcal{V}_X}}{|P.Y|_i^\sigma} \left(\sum_{j.Y \in P.Y|_i^\sigma} j.Y \right) + \epsilon$$

where $\beta_{P.Y, \mathcal{V}_X} = 1 + |\gamma|$ where $\gamma \sim \mathcal{N}(0, 0.1^2)$ for every $P.Y \in pa(\mathcal{V}_X; \mathcal{M})$ for every $X \in \boldsymbol{A}$. $\epsilon \sim \mathcal{N}(0, 0.1^2)$. The set of parameters will likely yield a relational data less hostile for our learning algorithm given that $\beta \geq 1$ and the variance of noise is relatively small. This fulfills our intention to assess the behavior of learning algorithm across different settings. If we wanted to exploit the fact that the generated RCMs are based on an average aggregator, we could incorporate this into the choice of kernel so that R-convolution kernel is not necessary but a simple RBF kernel on averaged values is sufficient.

Random relational skeletons are generated with a user-specified *base* size n. Given n, the number of relationships (i.e., relationship instances or relationship items) for each relationship class is the twice of the base size if the cardinality is 'many' for every its participating entity class and the same as base size, otherwise. The number of entities per entity class can be computed as $\lfloor 1.2^k \cdot n \rfloor$ where k is the number of related relationship classes with all-'one' cardinalities. For each RCM, we generate 4 relational skeletons corresponding to base size from 200 to 500, increased by 100.

ROBUSTIFICATION of RPCD VS. NAIVE RPCD

For the robust RpCD, we adopted features mentioned in the main paper: order-independence for Phase-I, and split-RBO, pair-RBO, non-RBO tests, and weak dependence detection for Phase-II. Aggregation-based additional tests are applied to both phases. Separating sets are sought from the smallest size of conditionals to the largest. If a separating set is found with size k, the algorithm checks the existence of other separating sets of the same size, which we call 'minimal separating sets'. Then, the orientation of relational dependencies is based on a majority vote for the orientation of each pair of attribute classes. At the end of the algorithm, different orientations are combined to yield a partially-oriented RCM (PRCM) which maximally satisfies obtained test results. If there are multiple candidate PRCMs matching the same number of test

Aggregation	Order Ind.	Base size	precision	recall
True	False	200	97.70%	63.71%
		300	97.61%	70.93%
		400	98.42%	76.21%
		500	98.81%	78.74%
	True	200	98.64%	60.12%
		300	98.99%	69.27%
		400	99.00%	74.52%
		500	99.04%	77.32%
False	False	200	98.02%	60.39%
		300	98.02%	68.40%
		400	98.32%	73.77%
		500	98.82%	76.29%
	True	200	98.76%	56.45%
		300	98.94%	66.23%
		400	98.86%	71.83%
		500	99.06%	75.03%

Table 5: Performance based on micro-average of Phase-I.

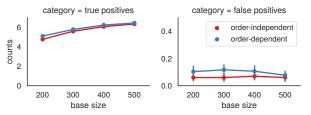| Base Size	Aggregation	Order Ind.	TP	FP
200	False	True	4.843	0.060
		False	5.143	0.090
	True	True	5.103	0.067
		False	5.350	0.130
500	False	True	6.373	0.057
		False	6.493	0.093
	True	True	6.523	0.063
		False	6.633	0.090

Table 6: Performance of Phase-I with average number of true positives (TP) and of false positives (FP)/ FPs are reduced to about two thirds by adopting order-independence.

results, then we choose a PRCM, which has the most common orientations with other competitors.
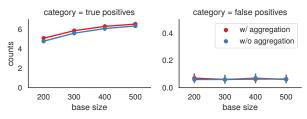
PHASE-I

We first report the performance of Robust RpCD for Phase-I. Micro-averaged precision and recall for undirected dependencies are reported (see Tab. 5). As the size of data increases, more accurate RCMs are discovered since RCI tests can better catch genuine dependencies. We observe relatively high precision in general even with a small-sized relational data, which implies that the main problem of the structure learning is false negatives due to weak dependencies.

Order-independence Fig. 4a depicts plots of performance with and without order-independence — the average number of true and false positives without additional aggregation-based tests. First, order-dependence can yield

(a) Performance on order-independence without aggregation

(b) Effect of aggregation with order-independence
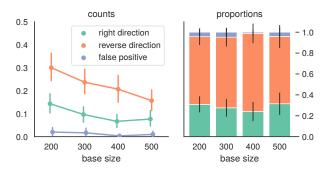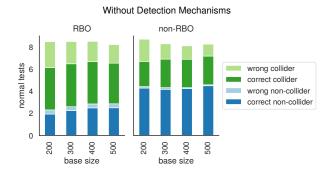
Figure 4: Phase-I

Figure 5: RCI query saved by aggregation-based tests

a higher number of both true and false positives. We can observe that order-independence reduces the number of false positives (see Tab. 6).

Aggregation In Fig. 4b, aggregation-based CI tests yield higher true positives without increasing false positives much. Since the non-aggregated test and its corresponding aggregated test are correlated, doubling the test does not significantly increase the false positive rate.

We explored which types of RCI queries are 'saved' by aggregated tests, i.e., $(U \perp\!\!\!\perp V \mid \mathbf{W}) \wedge (f(U) \not\perp\!\!\!\perp V \mid \mathbf{W})$ such that U is adjacent to V at the end of Phase I. We report three cases: i) false positive, $U \notin adj(V; \mathcal{M})$; ii) right direction, $U \in pa(V; \mathcal{M})$; and iii) reverse direction, $U \in ch(V; \mathcal{M})$. We expected that the aggregation-based test is particularly useful when $U \in ch(V; \mathcal{M})$ since V affects each of item attribute in U 'individually'. Then, averaging values might help reducing noises. In Fig. 5, we illustrate the average number of saved dependencies in the three categories and their proportions. Note that, an
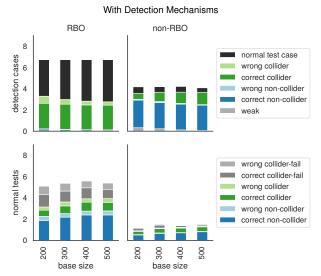
Figure 6: Effect of detection mechanism

Size	Precision	Recall	F-measure
200	65.8	61.0	63.3
300	74.2	67.8	70.8
400	71.9	66.6	69.1
500	75.3	69.7	72.4

Table 7: Orientation performance of a naive approach with CUT-based RCI tests.

Agg.	Size	Detection	Prec.	Recall	F
False	200	False	79.0	64.5	71.0
	300		84.7	68.7	75.8
	400		88.1	72.8	79.7
	500		87.8	73.6	80.1
False	200	True	88.6	69.4	77.8
	300		92.4	73.0	81.5
	400		94.2	76.2	84.2
	500		93.6	75.9	83.8
True	200	True	88.3	70.1	78.2
	300		91.5	73.6	81.6
	400		93.8	76.6	84.3
	500		93.5	75.4	83.5

Table 8: Orientation performance with our proposed approach using RCI tests.

We report micro-average for precision and recall in Tabs. 7 and 8 for a naive approach (i.e., CUT-based RCI tests with a majority vote rule and a simple sequential strategy to resolve conflicts among orientations.) and our approach (i.e., the proposed RCI tests with the weak dependence detection mechanism, the majority vote rule and a maximal non-conflicting orientations strategy), respectively. The differences in both precision and recall between the two approaches are due to the effectiveness of our proposed RCI tests (as shown in the main text) and the fact that finding a maximally non-conflicting orientations works as a majority vote rule for final orientations of relational dependencies.

DETECTING CONFLICTS FOR RBO AND NON-RBO We investigate how weak dependency detection mechanisms for RBO and non-RBO work. In Fig. 6, we illustrate the average number of RCI tests which turned out to be colliders or non-colliders, and whether the RCI test results were right or wrong.

Without detection (the top row), we observe that there exists a non-negligible amount of wrong collider test results. This implies that a set of conditionals without blocking $\tilde{P}.Y$ (or $\tilde{Q}.Y$) yields wrong independence. This, again, suggests how false negatives dominate the performance of the learning algorithm.

With the detection mechanism enabled, the middle row

adjacency $P.X - \mathcal{V}_Y$, which is also $\tilde{P}.Y - \mathcal{V}_X$, can be counted twice. We can first observe that the total number of saved relational dependencies decreases as data size increases since the original (i.e., non-aggregation-based) test will catch weak dependencies better. RCI tests in a reverse direction, e.g., $U \in ch(V; \mathcal{M})$, are mostly saved by aggregation. The use of aggregation will become more useful as the relationships in a relational skeleton become more complicated.

PHASE-II

We first overview how each feature affects the performance of orientation in terms of precision and recall assuming perfect Phase-I, which allows us to judge better how different features work. More specifically, 'correctly directed' relational dependencies lie in the intersection of oriented relational dependencies through Phase-II and true relational dependencies. Then, precision and recall are the proportion of correctly directed relational dependencies among directed relational dependencies through Phase-II, and among directed relational dependencies in the corresponding CPRCM, respectively.

in the figure shows the average orientation results only when an empty set as a separating set is considered. Black bars represent cases where a pair of tests turned out to be dependent, that is, an orientation was not determined. Gray bars (nearly invisible) show cases where both tests returned independence. We can clearly see that the mechanism catches colliders better than without it.

The last row in the figure illustrates orientation results for the undetermined in the previous case (black bars). Note that, since the algorithm seeks for more than one separating set, the lengths of bars in the last row are longer than the lengths of black bars in the middle row. Collider-fail represents a condition where the detection mechanism rejects a collider since both tests yield independence. More than a half of cases, the mechanism correctly rejected false colliders, yielding a relatively low false collider rate.