Lifted Causal Inference

Malte Luttermann[1,2*], Tanya Braun[3], Ralf Möller[1], Marcel Gehrke[1]

[1]Institute for Humanities-Centered Artificial Intelligence, University of Hamburg, Hamburg, Germany.
[2*]German Research Center for Artificial Intelligence (DFKI), Lübeck, Germany.
[3]Data Science Group, University of Münster, Münster, Germany.

*Corresponding author(s). E-mail(s): malte.luttermann@dfki.de;
Contributing authors: tanya.braun@uni-muenster.de;
ralf.moeller@uni-hamburg.de; marcel.gehrke@uni-hamburg.de;

Abstract

Lifted inference exploits indistinguishabilities in probabilistic graphical models by using a representative for indistinguishable objects, thereby speeding up query answering while maintaining exact answers. In this article, we show how lifting can be applied to efficiently compute causal effects in relational domains. More specifically, we introduce parametric causal factor graphs (PCFGs) to incorporate causal knowledge in lifted models and give a formal semantics of interventions therein. We further present the Lifted Causal Inference (LCI) algorithm to compute causal effects on a lifted level, thereby drastically speeding up causal inference compared to propositional inference, e.g., in causal Bayesian networks. In addition, we present partially directed parametric causal factor graphs (PD-PCFGs) as a generalisation of PCFGs to handle partial causal knowledge and extend LCI to perform lifted causal inference in a PD-PCFG, thereby extending the applicability of lifted causal inference to a broader range of models requiring less prior knowledge about causal relationships.

Keywords: causal inference, lifting, probabilistic relational models

1 Introduction

A fundamental problem in the research field of artificial intelligence for an intelligent agent is to plan and act rationally in a relational domain. To compute the best possible action in a perceived state, the agent considers the available actions and chooses the one with the maximum expected utility. When computing the expected utility of an action performed on a specific variable, it is crucial to deploy the semantics of an intervention instead of a typical conditioning on that variable (Pearl, 2009, Chapter 4). When calculating the effect of an intervention, a specific variable is set to a fixed value and all incoming probabilistic causal influences of this variable must be ignored for the specific query. It is fundamental to deploy the semantics of an intervention instead of the typical conditioning to correctly determine the effect of an action. Otherwise, when treating actions as evidence (by applying a classical conditioning), conclusions might become misleading. For example, assume a scenario in which the severity of fires influences the number of firefighters trying to extinguish the fire, that is, the more severe a fire is, the more firefighters are on duty. Classical conditioning then suggests to reduce the number of firefighters to reduce the severity of fires (because the probability for a severe fire is lower when observing a low number of firefighters on duty). In this article, we apply lifting to efficiently compute causal effects (and hence, the correct effect of actions) in relational domains, where efficient inference refers to inference running in polynomial time with respect to domain sizes.

Over the last years, causal models have become a widely used formalism to answer questions concerning the causal effect of an intervention on a random variable (randvar) on another randvar. A causal model consists of (i) a causal graph representing the causal relationships between the involved randvars, and (ii) a probability distribution over the randvars. There has been a considerable amount of work to perform causal effect estimation in causal models, and most of this work focuses on propositional models (Pearl, 2009; Pearl, Glymour, & Jewell, 2016; Peters, Janzing, & Schölkopf, 2017; Spirtes, Glymour, & Scheines, 2000). Some works extend propositional (undirected) factor graphs (FGs) by adding edge directions to enable the computation of the effect of interventions (Frey, 2003; Winn, 2012). Maier, Marazopoulou, Arbour, and Jensen (2013) introduce so-called relational causal models to express causal dependencies within relational domains. Their work focuses on causal discovery, that is, on learning relational causal models from observed data (Maier, Taylor, Oktay, & Jensen, 2010). Further developments on relational causal models also focus on causal discovery and on reasoning about conditional independence (e.g., Lee & Honavar, 2015, 2016, 2019). Relational causal models provide a lifted representation (that is, a representation that abstracts over individual objects and hence over all instantiations of a relational model) to reason about conditional independence, however, relational causal models do not support lifted causal inference. More recently, relational causal models have also been extended to cover cyclic dependency structures (Ahsan, Arbour, & Zheleva, 2022, 2023). Prior work dealing with the estimation of causal effects in relational domains still applies propositional probabilistic inference (Arbour, Garant, & Jensen, 2016; Salimi et al., 2020). Consequently, there is a lack of efficient algorithms to compute causal effects on a lifted level. In probabilistic inference, lifting exploits indistinguishabilities in a relational model, allowing to carry out query answering more efficiently while maintaining exact answers (Niepert & Van den Broeck, 2014). First introduced by Poole (2003), parametric factor graphs (PFGs) and Lifted Variable Elimination (LVE) allow to perform lifted probabilistic inference, resulting in significant speed-ups for probabilistic query answering in relational domains. Over time, LVE has been refined by many researchers to reach its current form (Braun & Möller, 2018; De Salvo Braz, Amir, & Roth, 2005, 2006; Kisyński & Poole, 2009; Milch, Zettlemoyer, Kersting, Haimes, & Kaelbling, 2008; Taghipour, Fierens, Davis, & Blockeel, 2013). To perform efficient inference in a PFG not only for single queries but also for sets of queries, Braun and Möller (2016) introduce the Lifted Junction Tree (LJT) algorithm. PFGs have been well-studied for many years and have been developed further to incorporate probabilistic inference over time (Gehrke, Braun, & Möller, 2018; Gehrke, Möller, & Braun, 2020), and, among other extensions, to allow for decision making by following the maximum expected utility principle (Braun & Gehrke, 2022; Gehrke, Braun, & Möller, 2019; Gehrke, Braun, Möller, Waschkau, et al., 2019). Markov logic networks are another lifted representation and have been extended to incorporate maximum expected utility as well (Apsel & Brafman, 2012). In this article, we extend PFGs to enable lifted causal inference to correctly determine the effect of actions on a lifted level.

This article is based on and extends the works (Luttermann, Hartwig, Braun, Möller, & Gehrke, 2024) and (Luttermann, Braun, Möller, & Gehrke, 2024). Specifically, we present the introduced models and algorithms for lifted causal inference under a unified view, thereby making the following contributions: First, we give a formal definition of causal factor graphs (CFGs) as an extension of FGs to incorporate causal knowledge on a propositional level. We then provide a unified view on fully directed lifted causal models introduced by Luttermann, Hartwig, et al. (2024) and partially directed lifted causal models introduced by Luttermann, Braun, et al. (2024). In particular, we expose the connection between these models and their corresponding algorithms to perform lifted causal inference therein. We especially highlight the differences in the assumptions made in the two models and exhibit how these assumptions affect their corresponding inference algorithms. Furthermore, we align the model definitions and algorithm descriptions for consistency of terminology and improved clarity. We also extend the theoretical results for fully directed and partially directed lifted causal models and showcase all presented concepts on a full running example.

The remaining part of this article is structured as follows. In Sec. 2, we introduce CFGs and define the notion of an intervention in a CFG to allow for the computation of causal effects therein (on a propositional level). Thereafter, in Sec. 3, we present parametric causal factor graphs (PCFGs) as an extension of PFGs and provide a formal semantics of interventions in PCFGs. By incorporating causal knowledge on a lifted level, a PCFG allows to perform lifted causal inference, thereby enabling efficient decision making in relational domains using the notion of an intervention. Then, in Sec. 4, we elucidate the Lifted Causal Inference (LCI) algorithm, which operates on a PCFG, and show how LCI computes causal effects on a lifted level to avoid grounding the PCFG as much as possible. We then portray partially directed parametric causal factor graphs (PD-PCFGs) as a generalisation of PCFGs in Sec. 5. Afterwards, we investigate how the effect of interventions can be computed in a PD-PCFG in the presence of unknown causal relationships. In Sec. 6, we present the Extended Lifted Causal Inference (ELCI) algorithm as a generalisation of LCI to efficiently compute causal effects in a PD-PCFG before we conclude this article in Sec. 7.

2 Causal Factor Graphs

Similar to a causal Bayesian network (CBN) (Pearl, 1988, 2009), a CFG is a probabilistic graphical model that simultaneously encodes a probability distribution over a set of randvars $\boldsymbol{R}$ and causal relationships between the randvars in $\boldsymbol{R}$. As in non-causal FGs (Frey, Kschischang, Loeliger, & Wiberg, 1997; Kschischang, Frey, & Loeliger, 2001), the full joint probability distribution is encoded as a product of factors, where each factor is a function of a subset of the randvars. The difference between an FG and a CFG is that a CFG contains directed edges instead of undirected edges to represent the causal relationships between the randvars. More specifically, a directed edge from a randvar R_i to another randvar R_j in a CFG indicates that R_i is a direct cause of R_j and thus, the value of R_i influences the value of R_j (Pearl, 2009). Therefore, in any causal graph, it holds that the value of a randvar depends on the values of its parents. We next provide a formal definition of a CFG based on the definition of directed FGs given by Frey (2003). In the following, we denote by $\text{range}(R_i)$ the range of a randvar R_i, that is, the set of possible values that R_i can take.

Definition 1 (Causal Factor Graph) We define a *CFG* as a tuple $M = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{\Phi})$ where $(\boldsymbol{V}, \boldsymbol{E})$ is a directed bipartite graph with node set $\boldsymbol{V} = \boldsymbol{R} \cup \boldsymbol{F}$ and edge set $\boldsymbol{E} \subseteq \boldsymbol{R} \times \boldsymbol{F}$ and $\boldsymbol{\Phi}$ is a set of function definitions. The set of nodes $\boldsymbol{V}$ is divided into a set of randvars $\boldsymbol{R} = \{R_1, \dots, R_n\}$ (variable nodes) and a set of function names (factor nodes) $\boldsymbol{F} = \{f_1, \dots, f_m\}$. Every function name $f_j \in \boldsymbol{F}$ has a function definition (factor, for short) $\phi_j(\mathcal{R}_j) \in \boldsymbol{\Phi}$, where $\phi_j \colon \times_{R \in \mathcal{R}_j} \text{range}(R) \mapsto \mathbb{R}_{\geq 0}$ maps range values of a sequence $\mathcal{R}_j$ of randvars from $\boldsymbol{R}$ to a non-negative real number (potential). For each function definition, there must be at least one sequence of range values that is mapped to a potential which is non-zero. The set of edges $\boldsymbol{E}$ contains two types of edges. For every factor node $f_j \in \boldsymbol{F}$ with corresponding function definition $\phi_j(\mathcal{R}_j)$, there is either an undirected edge $\{R_i, f_j\} \in \boldsymbol{E}$ or a directed edge $(f_j, R_i) \in \boldsymbol{E}$ for every randvar $R_i \in \mathcal{R}_j$. We stipulate that for every factor node $f_j \in \boldsymbol{F}$, there exists exactly one outgoing directed edge $(f_j, R_i) \in \boldsymbol{E}$ among the edges incident to f_j. Each directed edge $\{R_i, f_j\}, (f_j \to R_k)$ from a randvar $R_i \in \boldsymbol{R}$ to a randvar $R_k \in \boldsymbol{R}$ via a factor node $f_j \in \boldsymbol{F}$ corresponds to a direct causal relationship between R_i and R_k. Furthermore, M has to be acyclic, that is, M is required to not contain any directed cycles. The joint potential for an assignment $\boldsymbol{R} = \boldsymbol{r}$ is defined as the product over all factors in the CFG M:

$$\psi_M(\boldsymbol{R} = \boldsymbol{r}) = \prod_{j=1}^{m} \phi_j(\mathcal{R}_j = \boldsymbol{r}_j), \tag{1}$$

where $\boldsymbol{r}_j$ is a projection of $\boldsymbol{r}$ to the argument list of ϕ_j. The full joint probability distribution $P_M(\boldsymbol{R})$ over $\boldsymbol{R}$ encoded by M is then given by the normalised joint potential:

$$P_M(\boldsymbol{R} = \boldsymbol{r}) = \frac{1}{Z} \psi_M(\boldsymbol{R} = \boldsymbol{r}), \tag{2}$$

where the normalisation constant Z is defined as the sum of all joint potentials:

$$Z = \sum_{\boldsymbol{r} \in \text{range}(R_1) \times \ldots \times \text{range}(R_n)} \psi_M(\boldsymbol{R} = \boldsymbol{r}). \tag{3}$$

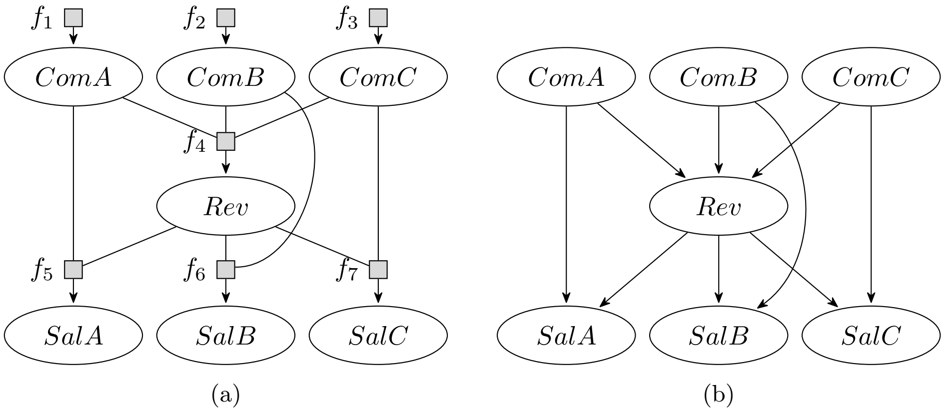

Fig. 1: (a) A CFG modelling the interplay between the competences and salaries of three employees *Alice*, *Bob*, and *Charlie* and the revenue of a company, and (b) the underlying causal graph. We omit the potential tables of the factors for brevity.

Example 1 (Causal Factor Graph) *Consider the CFG $M = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{\Phi})$ depicted in Fig. 1a. M represents the causal relationships between the competences and salaries of three employees Alice, Bob, and Charlie and the revenue of the company they work for. The underlying causal graph is illustrated in Fig. 1b. In set notation, the graph structure of M is given by $(\boldsymbol{V} = \boldsymbol{R} \cup \boldsymbol{F}, \boldsymbol{E})$, where*

$$\begin{aligned}
\boldsymbol{R} =& \{ComA, ComB, ComC, Rev, SalA, SalB, SalC\}, \\
\boldsymbol{F} =& \{f_1, f_2, f_3, f_4, f_5, f_6, f_7\}, \text{ and} \\
\boldsymbol{E} =& \{(f_1, ComA), \{ComA, f_4\}, \{ComA, f_5\}, (f_2, ComB), \{ComB, f_4\}, \{ComB, f_6\}, \\
& (f_3, ComC), \{ComC, f_4\}, \{ComC, f_7\}, (f_4, Rev), \{Rev, f_5\}, \{Rev, f_6\}, \{Rev, f_7\}, \\
& (f_5, SalA), (f_6, SalB), (f_7, SalC)\}.
\end{aligned}$$

Moreover, the set of function definitions is

$$\begin{aligned}
\boldsymbol{\Phi} =& \{\phi_1(ComA), \phi_2(ComB), \phi_3(ComC), \phi_4(ComA, ComB, ComC, Rev), \\
& \phi_5(ComA, Rev, SalA), \phi_6(ComB, Rev, SalB), \phi_7(ComC, Rev, SalC)\},
\end{aligned}$$

where we omit the exact specification of the potential tables for brevity.

If the context is clear, we may omit the subscript M in P_M and simply write P instead. From the definition of the full joint probability distribution P_M in Eq. (2), it becomes clear that the full joint probability distribution P_M encoded by a CFG M is independent of the edge directions in M. In other words, changing the edge directions in M does not affect the probability distribution P_M encoded by M. However, the edge directions impact the effect of an intervention and also the conditional independence statements induced by M, which are implied by separation in M. The separation criteria in a CFG differ from the separation criteria in a FG as the direction of edges influences whether paths are blocked or not. Before we define the separation criteria in a CFG, we introduce the following notations for a CFG $M = (\boldsymbol{R} \cup \boldsymbol{F}, \boldsymbol{E}, \boldsymbol{\Phi})$:

- $\mathrm{Pa}_{\boldsymbol{R}}(M, R) = \{R' \in \boldsymbol{R} \mid \exists f \in \boldsymbol{F} \colon \{R', f\} \in \boldsymbol{E} \wedge (f, R) \in \boldsymbol{E}\}$ denotes the set of parent randvars of a randvar $R \in \boldsymbol{R}$ in M,
- $\mathrm{Pa}(M, f) = \{R \in \boldsymbol{R} \mid \{R, f\} \in \boldsymbol{E}\}$ denotes the set of parent randvars of a factor node $f \in \boldsymbol{F}$ in M,
- $\mathrm{Ch}_{\boldsymbol{R}}(M, R) = \{R' \in \boldsymbol{R} \mid \exists f \in \boldsymbol{F} \colon \{R, f\} \in \boldsymbol{E} \wedge (f, R') \in \boldsymbol{E}\}$ denotes the singleton set of child randvars of a randvar $R \in \boldsymbol{R}$ in M,
- $\mathrm{Ch}(M, f) = \{R \in \boldsymbol{R} \mid (f, R) \in \boldsymbol{E}\}$ denotes the singleton set of child randvars of a factor node $f \in \boldsymbol{F}$ in M,
- $\mathrm{De}_{\boldsymbol{R}}(M, R) = \{R' \in \boldsymbol{R} \mid \exists f_1, \dots, f_k \in \boldsymbol{F}, R_1, \dots, R_{k-1} \in \boldsymbol{R} \colon \{R, f_1\}, (f_1, R_1), \dots, \{R_{k-1}, f_k\}, (f_k, R') \in \boldsymbol{E}\}$ denotes the set of descendant randvars of a randvar $R \in \boldsymbol{R}$ in M (i.e., randvars that can be reached from R via a directed path), and
- $\mathrm{De}(M, f) = \{R' \in \boldsymbol{R} \mid \exists f_1, \dots, f_k \in \boldsymbol{F}, R_1, \dots, R_k \in \boldsymbol{R} \colon (f, R_1), \{R_1, f_1\}, \dots, (f_k, R') \in \boldsymbol{E}\}$ denotes the set of descendant randvars of a factor node $f \in \boldsymbol{F}$ in M (i.e., randvars that can be reached from f via a directed path).

The subscript $\boldsymbol{R}$ indicates that the sets are defined with respect to the randvars in $\boldsymbol{R}$, that is, the sets contain neighbouring randvars (connected via a factor node) instead of directly connected factor nodes. As factor nodes are directly connected to randvars, we omit the subscript $\boldsymbol{R}$ for the sets defined with respect to factor nodes. Moreover, since every factor node has exactly one outgoing directed edge, it holds that $|\mathrm{Ch}(M, f)| = 1$ for every factor node $f \in \boldsymbol{F}$. For the ease of reading, we may write $f \to R$ (or $R \leftarrow f$) to represent a directed edge $(f, R) \in \boldsymbol{E}$ and $R - f$ (or $f - R$) to represent an undirected edge $\{R, f\} \in \boldsymbol{E}$. We next define separation in CFGs based on the definition given by Frey (2003) for directed FGs.

Definition 2 (Separation in Causal Factor Graphs) Let $M = (\boldsymbol{R} \cup \boldsymbol{F}, \boldsymbol{E}, \boldsymbol{\Phi})$ denote a CFG and let $\boldsymbol{R}_i \subseteq \boldsymbol{R}$, $\boldsymbol{R}_j \subseteq \boldsymbol{R}$, and $\boldsymbol{S} \subseteq \boldsymbol{R}$ denote pairwise disjoint sets of randvars. A path from a randvar to another randvar in M is a connected sequence of edges and is not restricted to follow the directions of the edges. Thus, it is also possible for a path to pass from a parenting randvar of a factor to another parenting randvar of the same factor. A path is blocked by $\boldsymbol{S}$ if

1. the path contains the pattern $f_1 \to S \leftarrow f_2$, where $f_1, f_2 \in \boldsymbol{F}$, such that neither S nor any of its descendants are in $\boldsymbol{S}$, or
2. the path passes from f_1 through S to f_2, where $f_1, f_2 \in \boldsymbol{F}$, such that it does not contain the pattern $f_1 \to S \leftarrow f_2$ and S is in $\boldsymbol{S}$, or
3. the path passes from a parent of a factor node $f \in \boldsymbol{F}$ to another parent of f, and neither the child of f nor any of its descendants are in $\boldsymbol{S}$.

M implies $(\boldsymbol{R}_i \perp\!\!\!\perp \boldsymbol{R}_j \mid \boldsymbol{S})$ if $\boldsymbol{S}$ *separates* $\boldsymbol{R}_i$ and $\boldsymbol{R}_j$ in M, that is, if $\boldsymbol{S}$ blocks all paths from a randvar in $\boldsymbol{R}_i$ to a randvar in $\boldsymbol{R}_j$.

The separation criteria for CFGs given in Def. 2 directly correspond to the rules of d-separation in Bayesian networks (BNs) introduced by Pearl (1986).

Example 2 (Separation) *Consider again the CFG M depicted in Fig. 1a. For instance, it holds that $ComA \perp\!\!\!\perp ComB$ (i.e., $\boldsymbol{S} = \emptyset$) as all paths from $ComA$ to $ComB$ are blocked by* *the condition given in Item 3 from Def. 2. However, it holds that $ComA \not\perp\!\!\!\perp ComB \mid \{Rev\}$ as, for instance, the path $ComA - f_4 - ComB$ is not blocked anymore as soon as $Rev \in \boldsymbol{S}$.*

When using a CFG M to encode a probability distribution, it is crucial that the conditional independence statements induced by M actually hold in the probability distribution. The *global Markov property* ensures that separation criteria in a CFG are compliant with the conditional independence statements in a probability distribution.

Definition 3 (Global Markov Property (Lauritzen, 1996)) A probability distribution P satisfies the *global Markov property* for a CFG M if and only if for all disjoint sets of variables $\boldsymbol{R}_i$, $\boldsymbol{R}_j$, and $\boldsymbol{S}$ it holds that if $\boldsymbol{R}_i$ is separated from $\boldsymbol{R}_j$ given $\boldsymbol{S}$ in M, then $\boldsymbol{R}_i$ is conditionally independent from $\boldsymbol{R}_j$ given $\boldsymbol{S}$ in P.

In other words, every conditional independence statement induced by the graph structure of a CFG M also holds in the probability distribution P if the global Markov property is satisfied. Thus, when encoding a probability distribution P using a CFG M, M cannot be chosen arbitrarily but instead must be chosen such that P satisfies the global Markov property with respect to M. In this article, we therefore assume that all distributions P satisfy the global Markov property for the corresponding CFG that is used to encode P. Furthermore, we additionally require that M and P fulfil the causal Markov property (Spirtes et al., 2000), stating that in M, every randvar $R \in \boldsymbol{R}$ is independent of all randvars that are neither effects nor direct causes of R given the set of R's direct causes. By assuming that the causal Markov property holds, we ensure that every directed edge in M accurately represents a causal relationship between the involved randvars (i.e., edge directions are actually causal).

Definition 4 (Causal Markov Property (Spirtes et al., 2000)) Let $M = (\boldsymbol{R} \cup \boldsymbol{F}, \boldsymbol{E}, \boldsymbol{\Phi})$ be a CFG and let P be a probability distribution over the randvars in $\boldsymbol{R}$ generated by the underlying causal structure of M. M and P satisfy the *causal Markov property* if and only if for every randvar $R \in \boldsymbol{R}$, R is independent of its non-descendants given its parents, i.e., $R \perp\!\!\!\perp \boldsymbol{R} \setminus (\mathrm{De}_{\boldsymbol{R}}(M, R) \cup \mathrm{Pa}_{\boldsymbol{R}}(M, R)) \mid \mathrm{Pa}_{\boldsymbol{R}}(M, R)$.

In case a CFG M and a probability distribution P satisfy the causal Markov property, P also satisfies the global Markov property with respect to M, that is, the global Markov property is implied by the causal Markov property. However, it is generally possible for a probability distribution P to satisfy the global Markov property for a directed (non-causal) FG M without M and P satisfying the causal Markov property if M encodes the conditional independence statements in P but does not represent the true underlying causal relationships of P (i.e., the directed edges in M are not causal). In case the causal Markov property is satisfied, direct causes of a randvar R are given by the parents of R and the effects of R are given by the children of R and hence, P factorises as

$$P(R_1 = r_1, \ldots, R_n = r_n) = \prod_{i=1}^{n} P(R_i = r_i \mid \mathrm{Pa}_{\boldsymbol{R}}(M, R_i) = \mathrm{pa}_{\boldsymbol{R}}(M, R_i)), \tag{4}$$

where $\mathrm{pa}_{\boldsymbol{R}}(M, R_i)$ denotes a projection of the assignment $(r_1, \ldots, r_n)$ to the parents of R_i. Moreover, we assume causal sufficiency (Spirtes et al., 2000) in this article. The causal sufficiency assumption states that all common causes of randvars that are included in a model are also included in the model.

Definition 5 (Causal Sufficiency) A set $\boldsymbol{R}$ of randvars is *causally sufficient* if and only if every common cause of any two randvars in $\boldsymbol{R}$ is also in $\boldsymbol{R}$.

To summarise, whenever we deal with a CFG (or any of the causal models introduced in the upcoming sections) M, we make the following assumptions:

1. M is acyclic, i.e., M contains no directed cycles (Def. 1),
2. M and the probability distribution P encoded by M satisfy the causal Markov property (Def. 4), and
3. the set of randvars $\boldsymbol{R}$ in M is causally sufficient (Def. 5).

We next introduce a formal notion of an intervention in a CFG. An intervention $do(R = r)$ sets the value of a randvar R to a fixed value $r \in \mathrm{range}(R)$ and removes all incoming influences on R (Pearl et al., 2016). In the following, we use the notation $do(R_1 = r_1, \ldots, R_k = r_k)$ to denote a joint intervention on the randvars $R_1, \ldots, R_k$, which is an abbreviation of $do(R_1 = r_1), \ldots, do(R_k = r_k)$. When performing an intervention, the underlying probability distribution changes. The next definition formalises the effect of an intervention on the probability distribution encoded by a CFG.

Definition 6 (Interventional Distribution) Let $M = (\boldsymbol{R} \cup \boldsymbol{F}, \boldsymbol{E}, \boldsymbol{\Phi})$ be a CFG with $\boldsymbol{R} = \{R_1, \ldots, R_n\}$. An intervention $do(R'_1 = r'_1, \ldots, R'_k = r'_k)$ on the randvars $R'_1, \ldots, R'_k \in \boldsymbol{R}$ changes the probability distribution P_M encoded by M such that

$$\begin{aligned} & P_M(R_1 = r_1, \ldots, R_n = r_n \mid do(R'_1 = r'_1, \ldots, R'_k = r'_k)) \\ & = \begin{cases} \displaystyle\prod_{R_i \in \{R_1, \ldots, R_n\} \setminus \{R'_1, \ldots, R'_k\}} P(r_i \mid \mathrm{pa}_{\boldsymbol{R}}(M, R_i)) & \text{if } \forall j \in \{1, \ldots, k\} \colon r_j = r'_j \\ 0 & \text{otherwise,} \end{cases} \end{aligned}$$

where $\mathrm{pa}_{\boldsymbol{R}}(M, R_i)$ denotes a projection of the assignment $(r_1, \ldots, r_n)$ to the parents $\mathrm{Pa}_{\boldsymbol{R}}(M, R_i)$ of R_i. $P_M(R_1 = r_1, \ldots, R_n = r_n \mid do(R'_1 = r'_1, \ldots, R'_k = r'_k))$ is called the *interventional distribution* of P_M under the intervention $do(R'_1 = r'_1, \ldots, R'_k = r'_k)$.

The interventional distribution is used to compute the effect of an intervention, that is, to answer an interventional query, which we define as follows.

Definition 7 (Interventional Query) An *interventional query* $P(Q \mid do(R_1 = r_1, \dots, R_k = r_k))$ consists of a query term Q (also called query variable) and a set of interventions $\boldsymbol{I} = \{do(R_1 = r_1), \dots, do(R_k = r_k)\}$ where Q and $R_1, \dots, R_k$ are disjoint randvars. We also refer to the variables $R_1, \dots, R_k$ in $\boldsymbol{I}$ as intervention variables. To query a specific probability instead of a probability distribution, the query term is an event $Q = q$.

To answer an interventional query, we can directly apply Def. 6. In particular, given a CFG $M = (\boldsymbol{R} \cup \boldsymbol{F}, \boldsymbol{E}, \boldsymbol{\Phi})$ with $\boldsymbol{R} = \{R_1, \dots, R_\ell, R'_1, \dots, R'_k\}$, for an intervention $do(R'_1 = r'_1, \dots R'_k = r'_k)$, the joint distribution over $R_1, \dots, R_\ell$ is given as

$$\begin{aligned} &P(R_1 = r_1, \dots, R_\ell = r_\ell \mid do(R'_1 = r'_1, \dots, R'_k = r'_k)) \\ &\quad = \prod_{R_i \in \{R_1, \dots, R_\ell\}} P(R_i = r_i \mid \mathrm{Pa}_{\boldsymbol{R}}(M, R_i) = \mathrm{pa}_{\boldsymbol{R}}(M, R_i)), \end{aligned} \tag{5}$$

where $\mathrm{pa}_{\boldsymbol{R}}(M, R_i)$ is a projection of the assignment $(r_1, \dots, r_\ell, r'_1, \dots, r'_k)$ to the parents $\mathrm{Pa}_{\boldsymbol{R}}(M, R_i)$ of R_i. Equation (5) is also known under the name of the truncated product formula or g-formula (Pearl et al., 2016). If we are not interested in the joint distribution over *all* randvars $R_1, \dots, R_\ell$ that are not intervened on but instead wish to compute the distribution over a subset of $\{R_1, \dots, R_\ell\}$, we have to sum out all randvars from $R_1, \dots, R_\ell$ that are not queried.

Definition 8 (Truncated Product Formula) Let $M = (\boldsymbol{R} \cup \boldsymbol{F}, \boldsymbol{E}, \boldsymbol{\Phi})$ be a CFG. Further, let $\boldsymbol{R} = \{Q, R_1, \dots, R_\ell, R'_1, \dots, R'_k\}$. The result of an interventional query $P(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k))$ is then given by

$$\begin{aligned} &P(Q \mid do(R'_1 = r'_1, \dots R'_k = r'_k)) \\ &\quad = \sum_{r_1 \in \mathrm{range}(R_1)} \dots \sum_{r_\ell \in \mathrm{range}(R_\ell)} P(Q \mid \mathrm{pa}_{\boldsymbol{R}}(M, Q)) \cdot \prod_{R_i \in \{R_1, \dots, R_\ell\}} P(r_i \mid \mathrm{pa}_{\boldsymbol{R}}(M, R_i)), \end{aligned} \tag{6}$$

where r_i is again shorthand for $R_i = r_i$ (analogously for $\mathrm{pa}_{\boldsymbol{R}}$) and $\mathrm{pa}_{\boldsymbol{R}}(M, Q)$ as well as $\mathrm{pa}_{\boldsymbol{R}}(M, R_i)$ denote projections of the assignment $(q, r_1, \dots, r_\ell, r'_1, \dots, r'_k)$ to the parents of Q and R_i, respectively.

So far, we introduced CFGs as propositional causal models to compute the effect of interventions. Next, we combine lifted representations with causal knowledge to allow for lifted causal inference.

3 Parametric Causal Factor Graphs

A PCFG combines a CFG and relational logic (Genesereth & Kao, 2017) (that is, first-order logic with known universes). By incorporating relational logic, a PCFG allows to encode that certain properties hold for all objects in a group (i.e., set) of objects. In a PCFG, parameterised randvars (PRVs) and parametric factors (parfactors) represent sets of randvars and factors, respectively. More specifically, a PRV is parameterised by logical variables (logvars), each having a domain consisting of constants, to represent a set of randvars. Replacing the logvars with constants from their respective domains, called *grounding*, results in classical randvars again. To restrict logvars to specific constants from their respective domains, PRVs are provided with constraints. We first define PRVs and their components and afterwards define parfactors, before we introduce PCFGs and their semantics.

3.1 Definitions

The upcoming definitions are based on the definitions given by Braun (2020) but have been adapted and extended.

Definition 9 (Parameterised Random Variable) Let $\boldsymbol{R}$ be a set of randvar names, $\boldsymbol{L}$ a set of logvar names, and $\boldsymbol{D}$ a set of constants. All sets are finite. Each logvar $L \in \boldsymbol{L}$ has a domain $\mathrm{dom}(L) \subseteq \boldsymbol{D}$. A *constraint* $C = (\mathcal{L}, \boldsymbol{C}_{\mathcal{L}})$ is a tuple of a sequence of logvars $\mathcal{L} = (L_1, \dots, L_n)$ and a set $\boldsymbol{C}_{\mathcal{L}} \subseteq \times_{i=1}^{n} \mathrm{dom}(L_i)$. The symbol $\top$ for C marks that no restrictions apply, i.e., $\boldsymbol{C}_{\mathcal{L}} = \times_{i=1}^{n} \mathrm{dom}(L_i)$. A *substitution* $\boldsymbol{\sigma} = \{L_i \mapsto t_i\}_{i=1}^{n}$ replaces every occurrence of logvar L_i with term $t_i \in \mathrm{dom}(L_i)$ (also called grounding). A *PRV* $R(L_1, \dots, L_n)$, $n \geq 0$, is a syntactical construct of a randvar name $R \in \boldsymbol{R}$ possibly combined with logvars $L_1, \dots, L_n \in \boldsymbol{L}$ to represent a set of randvars. If $n = 0$, the PRV is parameterless and forms a propositional randvar. A PRV A (or logvar L) under constraint C is given by $A_{|C}$ ($L_{|C}$, respectively). We may omit $|\top$ in $A_{|\top}$ or $L_{|\top}$. The term $\mathrm{range}(A)$ denotes the possible values of a PRV A. An *event* $A = a$ denotes the occurrence of PRV A with range value $a \in \mathrm{range}(A)$ and a set of events $\boldsymbol{\Xi} = \{A_1 = a_1, \dots, A_k = a_k\}$ is called *evidence*.

We further denote by $\mathrm{lv}(Y)$ the logvars occurring in Y, where Y may be a PRV or a constraint. The set of all instances of Y (a logvar or PRV) with respect to given constraints is denoted by $\mathrm{gr}(Y)$. An instance (also called grounding) of Y is the result of substituting the logvars in Y with constants from the specified constraints. For a set of elements $\boldsymbol{Y}$ (e.g., logvars or PRVs), we define $\mathrm{lv}(\boldsymbol{Y}) = \bigcup_{Y \in \boldsymbol{Y}} \mathrm{lv}(Y)$ and $\mathrm{gr}(\boldsymbol{Y}) = \bigcup_{Y \in \boldsymbol{Y}} \mathrm{gr}(Y)$. The next example introduces PRVs for our running example.

Example 3 (Parameterised Random Variable) *Consider* $\boldsymbol{R} = \{Com, Rev, Sal\}$ *for competence, revenue, and salary, respectively,* $\boldsymbol{L} = \{E\}$ *with* $\mathrm{dom}(E) = \{Alice, Bob, Charlie\}$ *(employees), and* $\boldsymbol{D} = \{Alice, Bob, Charlie\}$*. Combining Com and Sal with the logvar E, we obtain the PRVs* $Com(E)_{|\top} = Com(E)$ *and* $Sal(E)_{|\top} = Sal(E)$*. Furthermore, Rev is a parameterless PRV. For the sake of the example, let* $\mathrm{range}(Com(E)) = \mathrm{range}(Rev) = \mathrm{range}(Sal(E)) = \{\mathrm{low}, \mathrm{high}\}$*. Applying the substitution* $\boldsymbol{\sigma} = \{E \mapsto Alice\}$ *to* $Com(E)$ *results in* $Com(Alice)$*. The groundings of* $Com(E)$ *are given by* $\mathrm{gr}(Com(E)) = \{Com(Alice), Com(Bob), Com(Charlie)\}$*. Applying the constraint* $C = (E, \{Alice\})$ *to* $Com(E)$ *yields* $Com(E)_{|(E,\{Alice\})}$ *with groundings* $\mathrm{gr}(Com(E)_{|(E,\{Alice\})}) = \{Com(Alice)\}$*.*

We next define parfactors, which represent sets of factors and are used to encode the probability distribution over the randvars. A parfactor describes a function, mapping argument values to positive real numbers (potentials), of which at least one is non-zero.

Definition 10 (Parfactor) Let $\boldsymbol{\Phi}$ denote a set of function definitions, let $\mathcal{A} = (A_1, \dots, A_n)$ denote a sequence of PRVs, and let $(\mathcal{L}, \boldsymbol{C}_{\mathcal{L}})$ denote a constraint on the logvars $\mathcal{L}$ in $\mathcal{A}$. With $\phi\colon \times_{i=1}^{n} \mathrm{range}(A_i) \mapsto \mathbb{R}_{\geq 0}$ being a function from $\boldsymbol{\Phi}$, a *parfactor* is given by $\forall \boldsymbol{l} \in \boldsymbol{C}_{\mathcal{L}}\colon \phi(\mathcal{A})_{|(\mathcal{L},\boldsymbol{C}_{\mathcal{L}})}$, where $\mathcal{L}$ is substituted by $\boldsymbol{l}$ in $\mathcal{A}$. We write $\phi(\mathcal{A})_{|(\mathcal{L},\boldsymbol{C}_{\mathcal{L}})}$ as a shorthand for $\forall \boldsymbol{l} \in \boldsymbol{C}_{\mathcal{L}}\colon \phi(\mathcal{A})_{|(\mathcal{L},\boldsymbol{C}_{\mathcal{L}})}$ (omitting the substitution) and we again may omit $|\top$ in $\phi(\mathcal{A})_{|\top}$.

For a parfactor ϕ, $\mathrm{lv}(\phi)$ again refers to the logvars in ϕ and $\mathrm{gr}(\phi)$ again denotes the set of instances of ϕ. We next introduce parfactors for our running example.

Example 4 (Parfactor) *Take a look at* $\phi_1(Com(E))_{|\top}$ *with* $\mathrm{range}(Com(E)) = \{\mathrm{low}, \mathrm{high}\}$ *and* $\mathrm{dom}(E) = \{Alice, Bob, Charlie\}$*. For* ϕ_1*, we have*

$$\phi_1(Com(E))_{|\top} = \forall e \in \mathrm{dom}(E)\colon \phi_1(Com(e))_{|\top}.$$

It holds that $\mathrm{gr}(\phi_1(Com(E))_{|\top}) = \{\phi_1(Com(Alice)), \phi_1(Com(Bob)), \phi_1(Com(Charlie))\}$*. In this specific example,* $\phi_1(Com(E))_{|\top}$ *thus represents a set of three ground factors.*

Before we are ready to define a PCFG, we need one more concept, namely the concept of a counting randvar (CRV) (Milch et al., 2008), which allows us to compactly encode a factor where it does not matter which specific individual randvars have a certain range value but instead only the number of randvars having particular range values is of interest. The range of a CRV is the space of histograms, i.e., a range value is a histogram indicating how many randvars have a certain value.

Definition 11 (Counting Random Variable) Let $A(\mathcal{L})_{|C}$ denote a PRV under constraint C, where $\mathrm{lv}(\mathcal{L}) = \{L\}$, i.e., either $\mathcal{L}$ consists of only L or the other inputs are constants (meaning $\mathcal{L}$ contains at most one logvar). We denote a *CRV* by $\#_L[A(\mathcal{L})_{|C}]$. Its range is the space of possible histograms. A histogram h is a set of tuples $\{(v_i, n_i)\}_{i=1}^{m}$, $v_i \in \mathrm{range}(A(\mathcal{L}))$, $n_i \in \mathbb{N}$, $m = |\mathrm{range}(A(\mathcal{L}))|$, and $\sum_i n_i = |\mathrm{gr}(L_{|C})|$ for some constraint C over $\mathcal{L}$. A shorthand notation is $[n_1, \dots, n_m]$. Since counting binds the logvar L, $\mathrm{lv}(\#_L[A(\mathcal{L})]) = \mathcal{L} \setminus \{L\}$.

Example 5 (Counting Random Variable) *Let* $\#_E[Com(E)]$ *be a CRV,* $\mathrm{range}(Com(E)) = \{\mathrm{low}, \mathrm{high}\}$ *and* $\mathrm{dom}(E) = \{Alice, Bob, Charlie\}$*. Then, there are* $m = |\mathrm{range}(Com(E))| = 2$ *possible range values and* $n = |\mathrm{gr}(E)| = 3$ *groundings. Hence, the histograms are* $[0, 3]$*,* $[1, 2]$*,* $[2, 1]$*, and* $[3, 0]$ *(corresponding to* $\{(\mathrm{high}, 0), (\mathrm{low}, 3)\}$*,* $\{(\mathrm{high}, 1), (\mathrm{low}, 2)\}$*,* $\{(\mathrm{high}, 2), (\mathrm{low}, 1)\}$ *and* $\{(\mathrm{high}, 3), (\mathrm{low}, 0)\}$ *in set notation, respectively).*

We have now introduced all components of a PCFG, which we define next.

Definition 12 (Parametric Causal Factor Graph) A *PCFG* $M = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{\Phi})$ consists of a directed graph $(\boldsymbol{V}, \boldsymbol{E})$ with node set $\boldsymbol{V} = \boldsymbol{A} \cup \boldsymbol{G}$ and edge set $\boldsymbol{E} \subseteq \boldsymbol{A} \times \boldsymbol{G}$. The set of nodes $\boldsymbol{V} = \boldsymbol{A} \cup \boldsymbol{G}$ is partitioned into a set of PRVs $\boldsymbol{A} = \{A_1, \dots, A_n\}$ and a set of parfactor names (parfactor nodes) $\boldsymbol{G} = \{g_1, \dots, g_m\}$. For every parfactor name $g_j \in \boldsymbol{G}$, there is a function definition (parfactor) $\phi_j(\mathcal{A}_j)_{|C} \in \boldsymbol{\Phi}$ with $\mathcal{A}_j$ being a sequence of PRVs from $\boldsymbol{A}$ and C being a constraint on the logvars of $\mathcal{A}_j$ such that $\phi\colon \times_{A \in \mathcal{A}_j} \mathrm{range}(A) \mapsto \mathbb{R}_{\geq 0}$ maps range values in $\mathcal{A}_j$ to a positive real number (potential). In every function definition, at least one potential has to be non-zero and we again may omit $|\top$ in $\phi_j(\mathcal{A}_j)_{|\top}$. For each parfactor name $g_j \in \boldsymbol{G}$ with corresponding function definition $\phi_j(\mathcal{A}_j)_{|C} \in \boldsymbol{\Phi}$, there is either an undirected edge $\{A_i, g_j\} \in \boldsymbol{E}$ or a directed edge $(g_j, A_i) \in \boldsymbol{E}$ for every PRV $A_i \in \mathcal{A}_j$ (directed edges are only allowed to point from parfactor nodes to PRVs but not vice versa). We stipulate that for every parfactor node $g_j \in \boldsymbol{G}$, there is exactly one outgoing directed edge $(g_j, A_i) \in \boldsymbol{E}$ among the edges incident to g_j. Each directed edge $A_i - g_j \to A_k$ from a PRV $A_i \in \boldsymbol{A}$ to a PRV $A_k \in \boldsymbol{A}$ via a parfactor node $g_j \in \boldsymbol{G}$ corresponds to a direct causal relationship between A_i and A_k. A PCFG is an acyclic graph, that is, $\boldsymbol{E}$ contains no sequence of edges $\{A_1, g_1\}, (g_1, A_2), \ldots, \{A_{k-1}, g_k\}, (g_k, A_1)$ starting from an arbitrary PRV $A_1 \in \boldsymbol{A}$ such that the sequence ends again at A_1 when following the edges in the direction of the arrows. The semantics of M is given by grounding with respect to constraints and building a full joint distribution over $\boldsymbol{R} = \mathrm{gr}(\boldsymbol{A})$. The joint potential for an assignment $\boldsymbol{R} = \boldsymbol{r}$ is

$$\psi_M(\boldsymbol{R} = \boldsymbol{r}) = \prod_{\phi_j \in \boldsymbol{\Phi}} \prod_{\phi_k \in \mathrm{gr}(\phi_j)} \phi_k(\mathcal{R}_k = \boldsymbol{r}_k), \tag{7}$$

where $\boldsymbol{r}_k$ is a projection of $\boldsymbol{r}$ to the argument list $\mathcal{R}_k$ of ϕ_k. The normalised joint potential then yields the full joint probability distribution over $\boldsymbol{R}$ that is encoded by M, that is,

$$P_M(\boldsymbol{R} = \boldsymbol{r}) = \frac{1}{Z} \psi_M(\boldsymbol{R} = \boldsymbol{r}), \tag{8}$$

where Z is the normalisation constant, defined as

$$Z = \sum_{\boldsymbol{r} \in \times_{R \in \boldsymbol{R}} \mathrm{range}(R)} \psi_M(\boldsymbol{R} = \boldsymbol{r}). \tag{9}$$

The definition of a PCFG also implies that every CFG is a PCFG containing only parameterless PRVs (analogously, every factor is a parfactor having only parameterless arguments). Grounding a PCFG thus yields a CFG entailing equivalent semantics (that is, encoding the same full joint probability distribution) as the PCFG. As logvars abstract from individual objects, we refer to PCFGs as *lifted* representations and to CFGs as *propositional* representations (in the same way, we refer to parameterless PRVs as propositional randvars and to parfactors having only parameterless arguments as propositional factors). In the literature, a lifted representation is sometimes also referred to as a first-order representation. Before we take a look at an example, we introduce the following notations for a PCFG $M = (\boldsymbol{A} \cup \boldsymbol{G}, \boldsymbol{E}, \boldsymbol{\Phi})$:

- $\mathrm{Pa}_{\boldsymbol{A}}(M, A) = \{A' \in \boldsymbol{A} \mid \exists g \in \boldsymbol{G} \colon \{A', g\} \in \boldsymbol{E} \wedge (g, A) \in \boldsymbol{E}\}$ denotes the set of parent PRVs of a PRV $A \in \boldsymbol{A}$ in M,
- $\mathrm{Pa}(M, g) = \{A \in \boldsymbol{A} \mid \{A, g\} \in \boldsymbol{E}\}$ denotes the set of parent PRVs of a parfactor node $g \in \boldsymbol{G}$ in M,
- $\mathrm{Ch}_{\boldsymbol{A}}(M, A) = \{A' \in \boldsymbol{A} \mid \exists g \in \boldsymbol{G} \colon \{A, g\} \in \boldsymbol{E} \wedge (g, A') \in \boldsymbol{E}\}$ denotes the singleton set of child PRVs of a PRV $A \in \boldsymbol{A}$ in M,
- $\mathrm{Ch}(M, g) = \{A \in \boldsymbol{A} \mid (g, A) \in \boldsymbol{E}\}$ denotes the singleton set of child PRVs of a parfactor node $g \in \boldsymbol{G}$ in M,
- $\mathrm{De}_{\boldsymbol{A}}(M, A) = \{A' \in \boldsymbol{A} \mid \exists g_1, \ldots, g_k \in \boldsymbol{G}, A_1, \ldots, A_{k-1} \in \boldsymbol{A} \colon \{A, g_1\}, (g_1, A_1), \ldots, \{A_{k-1}, g_k\}, (g_k, A') \in \boldsymbol{E}\}$ is the set of descendant PRVs of a PRV $A \in \boldsymbol{A}$ in M, and
- $\mathrm{De}(M, g) = \{A' \in \boldsymbol{A} \mid \exists g_1, \ldots, g_k \in \boldsymbol{G}, A_1, \ldots, A_k \in \boldsymbol{A} \colon (g, A_1), \{A_1, g_1\}, \ldots, (g_k, A') \in \boldsymbol{E}\}$ is the set of descendant PRVs of a parfactor node $g \in \boldsymbol{G}$ in M.

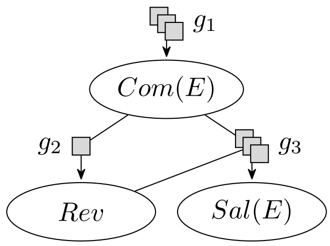

Fig. 2: An illustration of a PCFG for our running example. We omit the potential tables of the (par)factors for brevity.

As before, the subscript $\boldsymbol{A}$ indicates that the sets are defined with respect to the PRVs in $\boldsymbol{A}$, that is, the sets contain neighbouring PRVs that are connected via a parfactor node. Furthermore, as the definition of a PCFG requires every parfactor node to have exactly one outgoing directed edge, it holds that $|\mathrm{Ch}(M, g)| = 1$ for every parfactor node $g \in \boldsymbol{G}$. We next give an example.

Example 6 (Parametric Causal Factor Graph) *Figure 2 displays a PCFG M for our running example. M contains two PRVs $Com(E)$ (for the competence of employees) and $Sal(E)$ (for the salary of employees), as well as a propositional randvar Rev (for the revenue of the company). The ranges of the PRVs are* $\mathrm{range}(Com(E)) = \mathrm{range}(Sal(E)) = \mathrm{range}(Rev) = \{\mathrm{low}, \mathrm{high}\}$ *and the logvar E (representing employees) has the domain* $\mathrm{dom}(E) = \{Alice, Bob, Charlie\}$. *There are three parfactor nodes g_1, g_2, and g_3 with corresponding function definitions $\phi_1(Com(E))$, $\phi_2(\#_E[Com(E)], Rev)$, and $\phi_3(Com(E), Rev, Sal(E))$. We omit the potential tables of the parfactors for brevity. As $Com(E)$ appears count-converted in $\phi_2(\#_E[Com(E)], Rev)$, it holds that $\mathrm{lv}(\phi_2) = \emptyset$ and thus, g_2 is not layered in Fig. 2 while g_1 and g_3 are layered (because $\mathrm{lv}(\phi_1) \neq \emptyset$ and $\mathrm{lv}(\phi_3) \neq \emptyset$). In set notation, $M = (\boldsymbol{A} \cup \boldsymbol{G}, \boldsymbol{E}, \boldsymbol{\Phi})$ is given as*

$$\boldsymbol{R} = \{Com, Rev, Sal\},$$
$$\boldsymbol{L} = \{E\},$$
$$\boldsymbol{D} = \{Alice, Bob, Charlie\},$$
$$\boldsymbol{A} = \{Com(E), Rev, Sal(E)\},$$
$$\boldsymbol{G} = \{g_1, g_2, g_3\},$$
$$\boldsymbol{E} = \{(g_1, Com(E)), \{Com(E), g_2\}, \{Com(E), g_3\}, (g_2, Rev), \{Rev, g_3\}, (g_3, Sal(E))\},$$
$$\boldsymbol{\Phi} = \{\phi_1(Com(E)), \phi_2(\#_E[Com(E)], Rev), \phi_3(Com(E), Rev, Sal(E))\},$$

where $\boldsymbol{R}$ is the set of randvar names, $\boldsymbol{L}$ is the set of logvar names, and $\boldsymbol{D}$ is the set of constants. Grounding M results in the CFG from Ex. 1, where $Com(Alice)$ corresponds to $ComA$, $Com(Bob)$ corresponds to $ComB$, and so on.

We deliberately chose labels $ComA$, $ComB$, and $ComC$ instead of $Com(Alice)$, $Com(Bob)$, and $Com(Charlie)$ in Ex. 1 to emphasise that there is no explicit representation of objects (here employees) in the graph structure of the propositional CFG. In general, node labels can be arbitrary strings of characters. The size of the PCFG (that is, the number of nodes and edges in the graph) remains constant even if the number of employees increases. In the CFG from Fig. 1a, however, the size of the graph increases linearly with the number of employees as every additional employee adds two randvars and two factors to the graph. In general, there might be multiple groups of indistinguishable objects instead of having a single group including all objects, which can be represented by using constraints.

Before we continue to define the semantics of an intervention in a PCFG, we briefly provide the separation criteria in a PCFG, linking its graph structure to conditional independence statements in the underlying probability distribution.

3.2 Conditional Independence in Parametric Causal Factor Graphs

The separation criteria in a PCFG are given on a ground level and hence directly correspond to the separation criteria for a CFG given in Def. 2.

Definition 13 (Separation in Parametric Causal Factor Graphs) Let $M = (\boldsymbol{A} \cup \boldsymbol{G}, \boldsymbol{E}, \boldsymbol{\Phi})$ be a PCFG. Further, let $\boldsymbol{R} = \mathrm{gr}(\boldsymbol{A})$ and let $\boldsymbol{R}_i \subseteq \boldsymbol{R}$, $\boldsymbol{R}_j \subseteq \boldsymbol{R}$, and $\boldsymbol{S} \subseteq \boldsymbol{R}$ be pairwise disjoint sets of ground randvars. We say that $\boldsymbol{S}$ *separates* $\boldsymbol{R}_i$ and $\boldsymbol{R}_j$ in M if $\boldsymbol{S}$ blocks all paths from any randvar in $\boldsymbol{R}_i$ to any randvar in $\boldsymbol{R}_j$ in $\mathrm{gr}(M)$. M implies the conditional independence statement $(\boldsymbol{R}_i \perp\!\!\!\perp \boldsymbol{R}_j \mid \boldsymbol{S})$ if $\boldsymbol{S}$ separates $\boldsymbol{R}_i$ and $\boldsymbol{R}_j$ in M.

While the separation criteria in a PCFG are defined on a ground level, it is nevertheless possible to check for separation on a lifted level without grounding the entire PCFG. The Bayes-Ball algorithm (Shachter, 1998) allows to efficiently check for induced conditional independence statements in a propositional BN (and hence can also be applied to a CFG by taking the underlying causal graph structure into account). Meert, Taghipour, and Blockeel (2010) extend the Bayes-Ball algorithm to the lifted setting, thereby allowing to run it directly on a lifted representation.

It is also possible to check for implied conditional independence statements involving PRVs instead of ground randvars in a highly efficient manner on a lifted level. In a PCFG, every PRV $A(\mathcal{L})_{|C}$ is represented by a variable node and thus, checking for conditional independence statements that involve A can be done by looking at this specific variable node instead of taking into account all groundings of A. In contrast, in a propositional setting (i.e., in a ground model), each ground randvar in $\mathrm{gr}(A)$ is an individual node in the graph and hence must be looked at individually. For instance, to check whether $Com(E) \perp\!\!\!\perp Sal(E) \mid Rev$ is implied by the PCFG depicted in Fig. 2, only three variable nodes are of relevance whereas $2 \cdot |\mathrm{dom}(E)| + 1$ nodes are of relevance in the corresponding ground model shown in Fig. 1a. Here, the conditional independence statement $Com(E) \perp\!\!\!\perp Sal(E) \mid Rev$ (which is not implied by the PCFG from Fig. 2) is a shorthand to refer to a set of conditional independence statements resulting from substituting E with every $e \in \mathrm{dom}(E)$, i.e., $\{Com(e) \perp\!\!\!\perp Sal(e) \mid Rev\}_{e \in \mathrm{dom}(E)}$.

For now, we additionally assume that every PRV A_k in a PCFG has exactly one parent parfactor node g_j such that each corresponding ground function definition $\phi_j(R_1, \dots, R_k) \in \mathrm{gr}(\phi_j(A_1, \dots, A_k))$ represents a conditional probability distribution $P(R_k \mid R_1, \dots, R_{k-1})$. In other words, we assume that the ground CFG represented by a given PCFG directly corresponds to a CBN. This assumption results in a more convenient and often more efficient computation when answering interventional queries in a PCFG but is not necessary to answer such queries in general. Later on, we relax this assumption and show how interventional queries can still be answered on a lifted level. In the next subsection, we apply the notion of an intervention to PCFGs.

3.3 Interventions in Parametric Causal Factor Graphs

An intervention in a PCFG is defined analogously to an intervention in a CFG. The interventional distribution, in particular, is defined on a ground level again.

Definition 14 (Interventional Distribution in a Parametric Causal Factor Graph) Let $M = (\boldsymbol{A} \cup \boldsymbol{G}, \boldsymbol{E}, \boldsymbol{\Phi})$ be a PCFG with $\boldsymbol{R} = \mathrm{gr}(\boldsymbol{A}) = \{R_1, \dots, R_n\}$. Further, let $do(R'_1 = r'_1, \dots, R'_k = r'_k)$ be an intervention on the randvars $R'_1, \dots, R'_k \in \mathrm{gr}(\boldsymbol{A})$. The interventional distribution under the intervention $do(R'_1 = r'_1, \dots, R'_k = r'_k)$ is given by

$$
\begin{aligned}
&P_M(R_1 = r_1, \dots, R_n = r_n \mid do(R'_1 = r'_1, \dots, R'_k = r'_k)) \\
&\quad = \begin{cases} \displaystyle\prod_{R_i \in \{R_1, \dots, R_n\} \setminus \{R'_1, \dots, R'_k\}} P(r_i \mid \mathrm{pa}_{\boldsymbol{R}}(\mathrm{gr}(M), R_i)) & \text{if } \forall j \in \{1, \dots, k\} \colon r_j = r'_j \\ 0 & \text{otherwise,} \end{cases}
\end{aligned}
$$

where $\mathrm{pa}_{\boldsymbol{R}}(\mathrm{gr}(M), R_i)$ denotes a projection of the assignment $(r_1, \dots, r_n)$ to the parents $\mathrm{Pa}_{\boldsymbol{R}}(\mathrm{gr}(M), R_i)$ of R_i in the ground model $\mathrm{gr}(M)$.

Furthermore, we allow for interventions on PRVs. An intervention $do(A(\mathcal{L})_{|C} = a)$ on a PRV A, where $a \in \mathrm{range}(A)$, can be seen as a joint intervention on all ground randvars in $\mathrm{gr}(A_{|C})$. In other words, $do(A(\mathcal{L})_{|C} = a)$ is equivalent to $do(R_1 = a, \dots, R_k = a)$, where $\mathrm{gr}(A_{|C}) = \{R_1, \dots, R_k\}$. From now on, we therefore also allow for interventional queries of the form $P(Q \mid do(A_1 = a_1, \dots, A_k = a_k))$, where $A_1, \dots, A_k$ are PRVs. Since any interventional query involving PRVs can be reduced to an interventional query containing only parameterless randvars, we continue to work with our original definition of an interventional query (Def. 7). To answer an interventional query in a PCFG, we can again apply the truncated product formula.

Definition 15 (Truncated Product Formula in a Parametric Causal Factor Graph) Let $M = (\boldsymbol{A} \cup \boldsymbol{G}, \boldsymbol{E}, \boldsymbol{\Phi})$ be a PCFG and let $\boldsymbol{R} = \mathrm{gr}(\boldsymbol{A}) = \{Q, R_1, \dots, R_\ell, R'_1, \dots, R'_k\}$. The result of an interventional query $P(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k))$ is then given by

$$
\begin{aligned}
P(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k)) = &\sum_{r_1 \in \mathrm{range}(R_1)} \dots \sum_{r_\ell \in \mathrm{range}(R_\ell)} P(Q \mid \mathrm{pa}_{\boldsymbol{R}}(\mathrm{gr}(M), Q)) \\
&\cdot \prod_{R_i \in \{R_1, \dots, R_\ell\}} P(r_i \mid \mathrm{pa}_{\boldsymbol{R}}(\mathrm{gr}(M), R_i)),
\end{aligned} \tag{10}
$$

where $\mathrm{pa}_{\boldsymbol{R}}(\mathrm{gr}(M), Q)$ and $\mathrm{pa}_{\boldsymbol{R}}(\mathrm{gr}(M), R_i)$ denote projections of the assignment $(q, r_1, \dots, r_\ell, r'_1, \dots, r'_k)$ to the parents of Q and R_i in the ground model $\mathrm{gr}(M)$, respectively.

Even though both the interventional distribution and the truncated product formula are defined on a ground level, an interventional query can be answered without grounding the entire PCFG. In particular, Eq. (10) gives us a formula that consists of a set of probabilistic queries, which can be answered on a lifted level. Under the assumption of having a direct correspondence of parfactors to conditional probability distributions, we can further simplify query answering in a PCFG M.

Proposition 1 *Let $M = (\boldsymbol{A} \cup \boldsymbol{G}, \boldsymbol{E}, \boldsymbol{\Phi})$ denote a PCFG with each $\phi_j(R_{j_1}, \ldots, R_{j_z}) \in \mathrm{gr}(\boldsymbol{\Phi})$ representing a conditional probability distribution $P(R_{j_z} \mid R_{j_1}, \ldots, R_{j_{z-1}})$, let $\mathrm{gr}(\boldsymbol{A}) = \{Q, R_1, \ldots, R_\ell, R'_1, \ldots, R'_k\}$, and let $P(Q \mid do(R'_1 = r'_1, \ldots, R'_k = r'_k))$ be an interventional query. Further, let $M' = (\boldsymbol{A} \cup \boldsymbol{G}, \boldsymbol{E}, \boldsymbol{\Phi}')$ be the PCFG obtained by changing $\boldsymbol{\Phi}$ to $\boldsymbol{\Phi}'$ such that every factor $\phi_j(R_{j_1}, \ldots, R_{j_z}) \in \mathrm{gr}(\boldsymbol{\Phi})$ that has a child $R_{j_z} = R'_z$ in $\{R'_1, \ldots, R'_k\}$ is replaced by a factor $\phi'_j(R_{j_1}, \ldots, R_{j_z}) \in \mathrm{gr}(\boldsymbol{\Phi}')$ with*

$$\phi'_j(R_{j_1} = r_{j_1}, \ldots, R_{j_z} = r_{j_z}) = \begin{cases} 1 & \text{if } r_{j_z} = r'_z \\ 0 & \text{if } r_{j_z} \neq r'_z. \end{cases}$$

All factors whose child is not in $\{R'_1, \ldots, R'_k\}$ remain unchanged. The result of the interventional query $P_M(Q \mid do(R'_1 = r'_1, \ldots, R'_k = r'_k))$ in the original model M is then given by the result of the probabilistic query $P_{M'}(Q \mid R'_1 = r'_1, \ldots, R'_k = r'_k)$ in the modified model M'.

Proof For each ground factor $\phi_j(R_{j_1}, \ldots, R_{j_z}) \in \mathrm{gr}(\boldsymbol{\Phi})$, it holds that

$$\phi_j(R_{j_1} = r_{j_1}, \ldots, R_{j_z} = r_{j_z}) = P(R_{j_z} = r_{j_z} \mid R_{j_1} = r_{j_1}, \ldots, R_{j_{z-1}} = r_{j_{z-1}}) \tag{11}$$

for all assignments $(r_{j_1}, \ldots, r_{j_z})$. Entering Eq. (11) into the truncated product formula (Eq. (10)) leaves us with

$$P_M(Q \mid do(R'_1 = r'_1, \ldots, R'_k = r'_k)) = \sum_{r_1 \in \mathrm{range}(R_1)} \cdots \sum_{r_\ell \in \mathrm{range}(R_\ell)} \prod_{\phi_j \in \hat{\mathrm{gr}}(\boldsymbol{\Phi})} \phi_j(\mathcal{R}_j = \boldsymbol{r}_j),$$

where $\hat{\mathrm{gr}}(\boldsymbol{\Phi}) = \{\phi_j(R_{j_1}, \ldots, R_{j_z}) \in \mathrm{gr}(\boldsymbol{\Phi}) \mid R_{j_z} \notin \{R'_1, \ldots, R'_k\}\}$ denotes the set of groundings of $\boldsymbol{\Phi}$ whose child is not in $\{R'_1, \ldots, R'_k\}$ and $\boldsymbol{r}_j$ denotes a projection of the assignment $(q, r_1, \ldots, r_\ell, r'_1, \ldots, r'_k)$ to the argument list $\mathcal{R}_j$ of ϕ_j. Now, consider the modified model M' and the query $P(Q \mid R'_1 = r'_1, \ldots, R'_k = r'_k)$. By entering Eq. (11) into the definition of the full joint probability distribution $P_{M'}$ encoded by M' (Eq. (8)), we end up with

$$P_{M'}(Q \mid R'_1 = r'_1, \ldots, R'_k = r'_k) = \frac{1}{Z'} \sum_{r_1 \in \mathrm{range}(R_1)} \cdots \sum_{r_\ell \in \mathrm{range}(R_\ell)} \prod_{\phi'_j \in \mathrm{gr}(\boldsymbol{\Phi}')} \phi'_j(\mathcal{R}_j = \boldsymbol{r}_j).$$

Furthermore, as every factor ϕ'_j whose child is not in $\{R'_1, \ldots, R'_k\}$ is left unchanged (i.e., $\phi'_j(R_{j_1}, \ldots, R_{j_z}) = \phi_j(R_{j_1}, \ldots, R_{j_z})$ if $R_{j_z} \notin \{R'_1, \ldots, R'_k\}$), it holds that

$$\begin{aligned} \mathrm{gr}(\boldsymbol{\Phi}') &= \{\phi'_j(R_{j_1}, \ldots, R_{j_z}) \in \mathrm{gr}(\boldsymbol{\Phi}') \mid R_{j_z} \notin \{R'_1, \ldots, R'_k\}\} \\ &\quad \cup \{\phi'_j(R_{j_1}, \ldots, R_{j_z}) \in \mathrm{gr}(\boldsymbol{\Phi}') \mid R_{j_z} \in \{R'_1, \ldots, R'_k\}\} \\ &= \hat{\mathrm{gr}}(\boldsymbol{\Phi}) \cup \{\phi'_j(R_{j_1}, \ldots, R_{j_z}) \in \mathrm{gr}(\boldsymbol{\Phi}') \mid R_{j_z} \in \{R'_1, \ldots, R'_k\}\}. \end{aligned}$$

Due to the modifications in M', it holds that every factor $\phi'_j(R_{j_1}, \ldots, R_{j_z}) \in \{\phi'_j(R_{j_1}, \ldots, R_{j_z}) \in \mathrm{gr}(\boldsymbol{\Phi}') \mid R_{j_z} \in \{R'_1, \ldots, R'_k\}\}$ maps any assignment that assigns $R'_1 = r'_1, \ldots, R'_k = r'_k$ to the value one. Consequently, we end up with

$$P_{M'}(Q \mid R'_1 = r'_1, \ldots, R'_k = r'_k) = \frac{1}{Z'} \sum_{r_1 \in \mathrm{range}(R_1)} \cdots \sum_{r_\ell \in \mathrm{range}(R_\ell)} \prod_{\phi'_j \in \mathrm{gr}(\boldsymbol{\Phi}')} \phi'_j(\mathcal{R}_j = \boldsymbol{r}_j)$$

$$= \frac{1}{Z'} \sum_{r_1 \in \mathrm{range}(R_1)} \dots \sum_{r_\ell \in \mathrm{range}(R_\ell)} \prod_{\phi_j \in \hat{\mathrm{gr}}(\boldsymbol{\Phi})} \phi_j(\mathcal{R}_j = \boldsymbol{r}_j).$$

Thus, to conclude the proof of showing that $P_M(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k)) = P_{M'}(Q \mid R'_1 = r'_1, \dots, R'_k = r'_k)$, it remains to be shown that $Z' = 1$. The definition of the normalisation constant Z' (Eq. (9)) is given by

$$Z' = \sum_{(q, r_1, \dots, r_\ell, r'_1, \dots, r'_k) \in \times_{R \in \mathrm{gr}(\boldsymbol{A})} \mathrm{range}(R)} \prod_{\phi'_j \in \mathrm{gr}(\boldsymbol{\Phi}')} \phi'_j(\mathcal{R}_j = \boldsymbol{r}_j).$$

After the modification, every factor $\phi'_j(R_{j_1}, \dots, R_{j_z}) \in \mathrm{gr}(\boldsymbol{\Phi}')$ still represents a valid conditional probability distribution $P(R_{j_z} = r_{j_z} \mid R_{j_1} = r_{j_1}, \dots, R_{j_{z-1}} = r_{j_{z-1}})$ because exactly one assignment of R_{j_z} is mapped to one while all other assignments are mapped to zero, thus ensuring that the sum of all assignments is one. We can therefore again apply Eq. (11) to the definition of the normalisation constant Z' and obtain

$$Z' = \sum_{(q, r_1, \dots, r_\ell, r'_1, \dots, r'_k) \in \times_{R \in \mathrm{gr}(\boldsymbol{A})} \mathrm{range}(R)} \prod_{R_i \in \mathrm{gr}(\boldsymbol{A})} P(r_i \mid \mathrm{pa}_{\boldsymbol{R}}(\mathrm{gr}(M'), R_i)),$$

where r_i is the assigned value for R_i in the assignment $(q, r_1, \dots, r_\ell, r'_1, \dots, r'_k)$ and $\mathrm{pa}_{\boldsymbol{R}}(\mathrm{gr}(M), R_i)$ is a projection of the assignment $(q, r_1, \dots, r_\ell, r'_1, \dots, r'_k)$ to the parents of R_i in the ground model $\mathrm{gr}(M')$. In other words, Z' is a sum over all entries in the full joint probability distribution, and hence, it holds that $Z' = 1$ as all entries in a full joint probability distribution must sum up to one. □

An alternative way of verifying that the normalisation constant is equal to one if every factor $\phi_j(R_{j_1}, \dots, R_{j_z})$ represents a conditional probability distribution $P(R_{j_z} \mid R_{j_1}, \dots, R_{j_{z-1}})$ is to make use of Eq. (11) in the factorisation implied by the causal Markov property (Eq. (4)). The resulting factorisation then is equivalent to the factorisation given in the definition of the semantics of a PCFG (Eq. (8)) for $Z = 1$ and as both factorisations are valid, Z has to be equal to one.

By modifying the original model, an interventional query $P(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k))$ can be answered by computing the result of *a single* probabilistic query in the modified model (however, the modification of the model introduces some overhead). Using the truncated product formula directly on the original model instead, we obtain a set of multiple probabilistic queries that have to be answered. In both cases, a lifted inference algorithm such as LVE or the LJT algorithm (which is specifically advantageous if a set of probabilistic queries needs to be answered as a result of the truncated product formula) can be applied to answer these queries on a lifted level.

Before we give a full algorithm to efficiently answer interventional queries in a PCFG, we explain how the modification of the original model is done such that Prop. 1 can be applied. In particular, as only specific ground factors that have an intervention variable as a child are changed, we have to *split* parfactors such that modified ground factors can be separated from the remaining ground factors. Splitting a parfactor in a PCFG M results in a modified PCFG M' entailing equivalent semantics as M (De Salvo Braz et al., 2005) such that M' forms a valid model on which lifted inference algorithms (such as LVE and the LJT algorithm) can be run. The procedure of splitting a parfactor $\phi(\mathcal{A})_{|C}$ on a specific instance $A(l_1, \dots, l_z) \in \mathrm{gr}(A(\mathcal{L}_A))$, where $A(\mathcal{L}_A) \in \mathcal{A}$ is a PRV in the argument list of $\phi(\mathcal{A})_{|C}$, replaces $\phi(\mathcal{A})_{|C}$ by two parfactors $\phi(\mathcal{A})_{|C_1}$ and $\phi(\mathcal{A})_{|C_2}$. The constraints C_1 and C_2 are chosen such that the inputs of $\phi(\mathcal{A})_{|C_1}$

Algorithm 1 Lifted Causal Inference

Input: A PCFG $M = (\boldsymbol{A} \cup \boldsymbol{G}, \boldsymbol{E}, \boldsymbol{\Phi})$, and an interventional query $P(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k))$ with $\{Q, R'_1, \dots, R'_k\} \subseteq \mathrm{gr}(\boldsymbol{A}) = \{Q, R_1, \dots, R_\ell, R'_1, \dots, R'_k\}$.

Output: The result of the interventional query $P(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k))$.

1: $M' \leftarrow$ PCFG obtained by splitting parfactors in M on each $R'_i \in \{R'_1, \dots, R'_k\}$
2: **for each** $R'_i \in \{R'_1, \dots, R'_k\}$ **do**
3:     **for each** $\phi'_j(R_{j_1}, \dots, R_{j_z}) \in \mathrm{Pa}(M', R'_i)$ **do**
4:         **for each** assignment $(r_{j_1}, \dots, r_{j_z}) \in \mathrm{range}(R_{j_1}) \times \dots \times \mathrm{range}(R_{j_z})$ **do**
5:             Set $\phi'_j(r_{j_1}, \dots, r_{j_z}) = \begin{cases} 1 & \text{if } (r_{j_1}, \dots, r_{j_z}) \text{ assigns } R'_i = r'_i \\ 0 & \text{if } (r_{j_1}, \dots, r_{j_z}) \text{ assigns } R'_i \neq r'_i \end{cases}$
6: $D \leftarrow$ Call LVE on M' and $P(Q \mid R'_1 = r'_1, \dots, R'_k = r'_k)$
7: **return** D

are restricted to all sequences under constraint C that contain $A(l_1, \dots, l_z)$ and the inputs of $\phi(\mathcal{A})_{|C_2}$ are restricted to the remaining input sequences under constraint C.

We next present the LCI algorithm, which efficiently answers interventional queries in a PCFG on a lifted level. The basic idea of LCI is to split parfactors based on the intervention variables such that the parent factors of intervention variables are detached from their respective groups and thus can be changed according to Prop. 1.

4 The Lifted Causal Inference Algorithm

The LCI algorithm solves the problem of efficiently computing the effect of interventions in a PCFG. LCI avoids to fully ground the PCFG if possible to benefit from lifted inference. For instance, consider again the PCFG M illustrated in Fig. 2 and assume we would like to compute the answer to the interventional query $P(Rev \mid do(Com(Bob) = \text{high}))$ in M. As the intervention $do(Com(Bob) = \text{high})$ fixes the value of $Com(Bob)$ to high, we have to treat Bob differently from $Alice$ and $Charlie$, whose competences remain unobserved. In other words, not all employees are indistinguishable anymore. Nevertheless, and this is the crucial point, we can still treat $Alice$ and $Charlie$ as indistinguishable when computing the result of the interventional query $P(Rev \mid do(Com(Bob) = \text{high}))$.

4.1 Algorithm Description

We next describe the LCI algorithm to compute the result of an interventional query $P(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k))$ in a PCFG M where every ground factor represents a conditional probability distribution. Algorithm 1 displays the entire LCI algorithm, which we now discuss in detail. First, in Line 1, LCI splits the parfactors in M on the intervention variables $R'_i \in \{R'_1, \dots, R'_k\}$ to obtain a modified PCFG M'. More specifically, LCI splits every parfactor $\phi \in \boldsymbol{\Phi}$ for which there is an instance $\phi_j \in \mathrm{gr}(\phi)$ such that any intervention variable $R'_i \in \{R'_1, \dots, R'_k\}$ is a child of ϕ_j. After the splitting procedure, the semantics of the model remains unchanged as the set of ground factors in M' is still the same as the set of ground factors of the initial model M. The only difference after splitting is that the ground factors are now arranged differently across the sets of ground instances. Having completed the split of all respective parfactors, LCI next changes the parent parfactors of all intervention variables $R'_i \in \{R'_1, \ldots, R'_k\}$ to modify the underlying probability distribution encoded by M' according to the semantics of the intervention $do(R'_1 = r'_1, \ldots, R'_k = r'_k)$ (Lines 2 to 5). LCI changes the parfactors in M' according to Prop. 1 and thus, after the parfactors have been changed, M' encodes the interventional distribution under the intervention $do(R'_1 = r'_1, \ldots, R'_k = r'_k)$. More specifically, as each intervention variable R'_i is fixed to the value r'_i, all parent parfactors $\phi'_j(R_{j_1}, \ldots, R_{j_z}) \in \mathrm{Pa}(M', R'_i)$ of R'_i are altered such that all input sequences $(r_{j_1}, \ldots, r_{j_z})$ assigning $R'_i = r'_i$ map to the value one while all other input sequences map to zero. Finally, LCI computes the result of the probabilistic query $P(Q \mid R'_1 = r'_1, \ldots, R'_k = r'_k)$ in the modified model M', which is, according to Prop. 1, equivalent to the result of the interventional query $P(Q \mid do(R'_1 = r'_1, \ldots, R'_k = r'_k))$ in the original model M. To compute the result of $P(Q \mid R'_1 = r'_1, \ldots, R'_k = r'_k)$ in M', LCI calls LVE on $P(Q \mid R'_1 = r'_1, \ldots, R'_k = r'_k)$ and M' and then returns the result computed by LVE (Lines 6 and 7). During this step, LVE (which originally operates on a PFG) ignores the edge directions in M'. Since the semantics of the underlying full joint probability distribution encoded by a PFG are defined identically to the semantics of the probability distribution encoded by a PCFG, LVE can also be applied to compute the result of probabilistic queries in a PCFG (alternatively, a different lifted inference algorithm that works on a PFG could be called as well).

Example 7 (Lifted Causal Inference) *Look at the PCFG M shown in Fig. 2 and consider the interventional query $P(Rev \mid do(Com(Bob) = \mathrm{high}))$. In accordance with the previous examples, we assume that $\mathrm{dom}(E) = \{Alice, Bob, Charlie\}$. Since $Com(Bob)$ is a particular instance of $Com(E)$, we have to split the parfactor $\phi_1(Com(E))_{|\top}$, which is a parent parfactor of $Com(E)$. Figure 3 shows the modified PCFG M' obtained after splitting $\phi_1(Com(E))_{|\top}$ on $Com(Bob)$. In M', $\phi_1(Com(E))_{|\top}$ has been replaced by $\phi_1(Com(E))_{|C'}$ (the corresponding parfactor node is g'_1) and $\phi_1(Com(E))_{|C''}$ (the corresponding parfactor node is g''_1), where $C' = (E, \{Bob\})$ and $C'' = (E, \{Alice, Charlie\})$. To incorporate the semantics of the intervention $do(Com(Bob)) = \mathrm{high}$, LCI next modifies the parent parfactors of $Com(Bob)$, i.e., LCI modifies $\phi_1(Com(E))_{|C'}$ in this example (since $\phi_1(Com(E))_{|C''}$ is constrained to Alice and Charlie, $\phi_1(Com(E))_{|C''}$ is not a parent of $Com(Bob)$ and hence not changed). More specifically, $\phi_1(Com(E) = \mathrm{high})_{|C'}$ is set to one and $\phi_1(Com(E) = \mathrm{low})_{|C'}$ is set to zero. Finally, LCI runs LVE to compute $P(Rev \mid Com(Bob) = \mathrm{high})$ in M', which is equivalent to computing $P(Rev \mid do(Com(Bob) = \mathrm{high}))$ in the original model M.*

LCI is able to handle both interventions on a single (ground) randvar as well as interventions on a conjunction of multiple randvars efficiently. In particular, when intervening on multiple indistinguishable randvars at the same time, LCI is able to treat those randvars as a group even after the intervention. For instance, assume that in our running example, we would like to train multiple employees simultaneously, as a training program is mostly offered not only for a single employee but for a group of employees (here, training an employee $e \in \mathrm{dom}(E)$ corresponds to the intervention $do(Com(e) = \mathrm{high})$). Then, it is not necessary to split all trained employees into

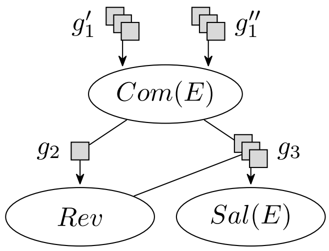

Fig. 3: A visualisation of the modified PCFG obtained after altering the PCFG shown in Fig. 2 by splitting $\phi_1(Com(E))_{|\top}$ on $Com(Bob)$.

separate groups but instead it is sufficient to differentiate between trained employees and all remaining employees. Formally, the intervention $do(R'_1 = r'_1, \dots, R'_k = r'_k)$ on an arbitrary set of randvars $\{R'_1, \dots, R'_k\}$ can thus efficiently be handled by splitting the parfactors in M such that all R'_i that are represented by the same PRV A and that are set to the same value $r'_i \in \text{range}(R'_i)$ remain grouped. Specifically, LCI needs just a single split on the parfactors per group and thus avoids manipulating the parents of each individual randvar separately. In contrast, in a propositional model, every object has to be treated individually and therefore the parents for each randvar need to be manipulated separately. Given the way we specified the semantics of an intervention in a PCFG, it immediately follows that LCI correctly computes the effect of interventions.

Proposition 2 *The result computed by LCI (Alg. 1) is the correct answer to the interventional query $P(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k))$ in the given PCFG M.*

Proof As LCI directly applies Prop. 1 by setting the parent factors of all intervention variables accordingly, the result of the interventional query $P(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k))$ in the original model M is equivalent to the result of the probabilistic query $P(Q \mid R'_1 = r'_1, \dots, R'_k = r'_k)$ in the modified model M' computed by LCI. $\square$

Moreover, by calling LVE, LCI allows for tractable probabilistic inference problems with respect to domain sizes of logvars. Thus, LCI runs in polynomial time with respect to domain sizes of logvars for all PCFGs belonging to the class of domain-liftable models. The class of domain-liftable models includes all PCFGs containing only parfactors with at most two logvars and all PCFGs containing only PRVs having at most one logvar (Van den Broeck, 2011).

Proposition 3 *LCI (Alg. 1) allows for tractable probabilistic inference problems with respect to domain sizes of logvars for the class of domain-liftable models.*

Proof If the input PCFG M for LCI belongs to the class of domain-liftable models, so does the modified model M' obtained after splitting the parfactors in M and changing the parent factors of the intervention variables because newly introduced parfactors contain identical logvars as the original parfactors that were split. Consequently, the input for LVE, which is given by M', belongs to the class of domain-liftable models and as LVE is complete for this model class (Taghipour, Fierens, Van den Broeck, Davis, & Blockeel, 2013) (that is, LVE runs in polynomial time with respect to the domain sizes of the logvars in its input model for all combinations of queries, evidence, and models in this class), the call of LVE allows for tractable probabilistic inference problems with respect to domain sizes of logvars provided that M belongs to the class of domain-liftable models. Furthermore, both the splitting procedure in Line 1 and the loops in Lines 2 to 5 of Alg. 1 do not influence the overall time complexity of LCI (as the loops iterate over potential tables that must be considered anyway during inference). Thus, LCI allows for tractable probabilistic inference problems with respect to domain sizes of logvars for the class of domain-liftable models. □

To summarise, LCI is a simple, yet effective algorithm to perform lifted causal inference. LCI can also handle queries with multiple query variables, provided that the lifted inference algorithm which is called in Line 6 can handle multiple query variables as well. Next, we take a look at our experiments, which highlight the practical performance of LCI to compute the effect of interventions in a PCFG on a lifted level.

4.2 Experiments

In this subsection, we evaluate the runtimes needed to compute the result of interventional queries in CBNs, CFGs, and PCFGs. For our experiments, we use a slightly modified version of the PCFG M given in Fig. 2, whose ground CFG directly corresponds to a CBN. In addition to the PCFG M, we also investigate runtimes for causal inference in the CFG obtained by grounding M, and its equivalent CBN. To obtain the equivalent CBN, we apply the transformation from directed FG to CBN proposed by Frey (2003). Hence, all three models, the PCFG, the CFG, and the CBN, encode the same underlying full joint probability distribution. As a remark, we note that the PCFG used in our experiments to demonstrate the practical efficiency of lifted causal inference is rather small with four parfactors and PRVs, respectively, and the gain we obtain from lifted inference might further increase for models consisting of more PRVs. Our experiments can thus be seen as a proof of concept demonstrating the practical efficiency of lifted causal inference and more extensive experiments on PCFGs with various graph structures and on PCFGs with more PRVs are left for future work.

We test the required runtime to compute the result of an interventional query for each of the three graphical models on different graph sizes by setting the domain size of the employees to $d \in \{8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096\}$, that is, $|\text{dom}(E)| = d$. Figure 4 shows the runtimes needed to compute the result of the probabilistic query in the modified graph when running variable elimination (VE) on the CFG, VE on the CBN, and LVE on the PCFG. As we have seen, VE algorithm is the propositional counterpart of LVE and operates on a propositional (ground) model, such as a CBN or an CFG. Consequently, VE considers every object (e.g., every employee) individually for computations, independent of whether objects are indistinguishable or not. In contrast, LVE treats indistinguishable objects as a group by using a representative for computations instead of considering each of those objects separately. The results emphasise that the LCI algorithm, which internally exploits LVE, overcomes scalability issues for large domain sizes as the runtime of LVE, in contrast to the runtimes of

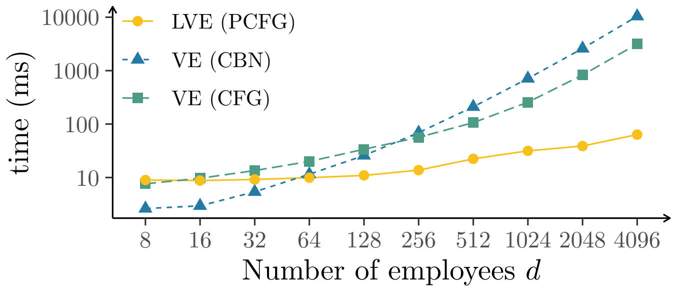

Fig. 4: A comparison of the runtimes required to compute interventional distributions on different graphical models encoding equivalent full joint probability distributions.

VE on the CBN and the CFG, does not exponentially increase for increasing values of d (y-axis is log-scaled). Even though the splitting of parfactors results in a less compressed lifted representation, it becomes evident that the performance of LVE is not significantly affected by the splitting.

While LCI in combination with a given PCFG solves the problem of efficiently computing the effect of interventions on a lifted level, in practice, we often face the problem of not knowing all the underlying causal relationships. Thus, in the upcoming section, we relax the assumption of knowing all causal relationships by allowing for partial causal knowledge. In particular, we introduce PD-PCFGs a generalisation of PCFGs and investigate the implications of not knowing all causal relationships for answering interventional queries. Moreover, we have hitherto assumed that every factor in a PCFG encodes a conditional probability distribution and we also abandon this assumption in the next section.

5 Partially Directed Parametric Causal Factor Graphs

We now move on to define a PD-PCFG as a lifted representation that is able to incorporate partial causal knowledge by combining a PFG with a partially directed graph to model causal relationships. The major advantage of a PD-PCFG over a PCFG is that not all causal relationships between the involved randvars need to be known, thereby reducing the amount of prior knowledge required and thus making the formalism more suitable for many practical settings.

Definition 16 (Partially Directed Parametric Causal Factor Graph) A *PD-PCFG* $M = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{\Phi})$ consists of a partially directed graph $(\boldsymbol{V}, \boldsymbol{E})$ with node set $\boldsymbol{V} = \boldsymbol{A} \cup \boldsymbol{G}$ and edge set $\boldsymbol{E} \subseteq \boldsymbol{A} \times \boldsymbol{G}$. The set of nodes $\boldsymbol{V} = \boldsymbol{A} \cup \boldsymbol{G}$ is partitioned into a set of PRVs $\boldsymbol{A} = \{A_1, \dots, A_n\}$ and a set of parfactor names (parfactor nodes) $\boldsymbol{G} = \{g_1, \dots, g_m\}$. For every parfactor name $g_j \in \boldsymbol{G}$, there is a function definition (parfactor) $\phi_j(\mathcal{A}_j)_{|C} \in \boldsymbol{\Phi}$ with $\mathcal{A}_j$ being a sequence of PRVs from $\boldsymbol{A}$ and C being a constraint on the logvars of $\mathcal{A}_j$ such that $\phi \colon \times_{A \in \mathcal{A}_j} \mathrm{range}(A) \mapsto \mathbb{R}_{\geq 0}$ maps range values in $\mathcal{A}_j$ to a positive real number (potential). As usual, in every function definition, at least one potential has to be non-zero and we may omit $|\top$ in $\phi_j(\mathcal{A}_j)_{|\top}$. For each parfactor name $g_j \in \boldsymbol{G}$ with corresponding function definition

$\phi_j(\mathcal{A}_j)_{|C} \in \boldsymbol{\Phi}$, there is either an undirected edge $\{A_i, g_j\} \in \boldsymbol{E}$ or a directed edge $(g_j, A_i) \in \boldsymbol{E}$ for every PRV $A_i \in \mathcal{A}_j$. We stipulate that for every parfactor node $g_j \in \boldsymbol{G}$, there is at most one outgoing directed edge $(g_j, A_i) \in \boldsymbol{E}$ among the edges incident to g_j. In case a parfactor $\phi_j \in \boldsymbol{\Phi}$ has only a single argument $A \in \boldsymbol{A}$, its corresponding parfactor node $g_j \in \boldsymbol{G}$ is connected to A via a directed edge $(g_j, A) \in \boldsymbol{E}$. Furthermore, each directed edge $A_k - g_j \to A_i$ from a PRV $A_k \in \boldsymbol{A}$ to a PRV $A_i \in \boldsymbol{A}$ via a parfactor node $g_j \in \boldsymbol{G}$ corresponds to a direct causal relationship between A_k and A_i. The directed edges in $\boldsymbol{E}$ are not allowed to form any directed cycles, i.e., $\boldsymbol{E}$ contains no sequence of edges $\{A_1, g_1\}, (g_1, A_2), \ldots, \{A_{k-1}, g_k\}, (g_k, A_1)$ starting from an arbitrary PRV $A_1 \in \boldsymbol{A}$ such that the sequence ends again at A_1 while every second edge in the sequence is directed and the sequence follows the arrow directions of the directed edges. The semantics of M is given by grounding with respect to constraints and building a full joint distribution over $\boldsymbol{R} = \mathrm{gr}(\boldsymbol{A})$. The joint potential for an assignment $\boldsymbol{R} = \boldsymbol{r}$ is defined as

$$\psi_M(\boldsymbol{R} = \boldsymbol{r}) = \prod_{\phi_j \in \boldsymbol{\Phi}} \prod_{\phi_k \in \mathrm{gr}(\phi_j)} \phi_k(\mathcal{R}_k = \boldsymbol{r}_k), \tag{12}$$

where $\boldsymbol{r}_k$ is a projection of $\boldsymbol{r}$ to the argument list $\mathcal{R}_k$ of ϕ_k. The normalised joint potential then yields the full joint probability distribution over $\boldsymbol{R}$ that is encoded by M, that is,

$$P_M(\boldsymbol{R} = \boldsymbol{r}) = \frac{1}{Z} \psi_M(\boldsymbol{R} = \boldsymbol{r}), \tag{13}$$

where Z is the normalisation constant, defined as

$$Z = \sum_{\boldsymbol{r} \in \times_{R \in \boldsymbol{R}} \mathrm{range}(R)} \psi_M(\boldsymbol{R} = \boldsymbol{r}). \tag{14}$$

A PD-PCFG thus offers the possibility to omit edge directions if no information about the underlying causal relationships is available. Grounding a PD-PCFG M yields a partially directed CFG, which encodes the same underlying full joint probability distribution as M. If all causal relationships are known (and hence all parfactor nodes have an outgoing edge), a PD-PCFG is identical to a PCFG. In a PD-PCFG $M = (\boldsymbol{A} \cup \boldsymbol{G}, \boldsymbol{E}, \boldsymbol{\Phi})$, we follow the same notations for the parent PRVs $\mathrm{Pa}_{\boldsymbol{A}}(M, A)$ of a PRV $A \in \boldsymbol{A}$, the parent PRVs $\mathrm{Pa}(M, g)$ of a parfactor node $g \in \boldsymbol{G}$, the child PRVs $\mathrm{Ch}_{\boldsymbol{A}}(M, A)$ of a PRV $A \in \boldsymbol{A}$, the child PRVs $\mathrm{Ch}(M, g)$ of a parfactor node $g \in \boldsymbol{G}$, the descendant PRVs $\mathrm{De}_{\boldsymbol{A}}(M, A)$ of a PRV $A \in \boldsymbol{A}$, and the descendant PRVs $\mathrm{De}(M, g)$ of a parfactor node $g \in \boldsymbol{G}$ as in a PCFG. Additionally, we define the (undirected) neighbour PRVs of a PRV $A \in \boldsymbol{A}$ in M as $\mathrm{Ne}_{\boldsymbol{A}}(M, A) = \{A' \in \boldsymbol{A} \mid \exists g \in \boldsymbol{G} \colon \mathrm{Ch}(M, g) = \emptyset \wedge \{g, A'\} \in \boldsymbol{E} \wedge \{g, A\} \in \boldsymbol{E}\}$. In contrast to a PCFG, the set of child PRVs $\mathrm{Ch}(M, g)$ of a parfactor node $g \in \boldsymbol{G}$ in a PD-PCFG M may be empty. More specifically, as every parfactor node has at most one outgoing edge, it holds that $|\mathrm{Ch}(M, g)| \leq 1$ for every parfactor node $g \in \boldsymbol{G}$. If a parfactor node $g \in \boldsymbol{G}$ corresponds to a parfactor with a single argument, it always holds that $|\mathrm{Ch}(M, g)| = 1$. From now on, we also drop the assumption that every ground factor $\phi_j(R_{j_1}, \ldots, R_{j_z}) \in \mathrm{gr}(\boldsymbol{\Phi})$ in a PD-PCFG M encodes a conditional probability distribution $P(R_{j_z} \mid R_{j_1}, \ldots, R_{j_{z-1}})$. Let us next consider a modified version of our running example, where the underlying causal relationships are only partially known, resulting in a PD-PCFG instead of a fully directed PCFG.

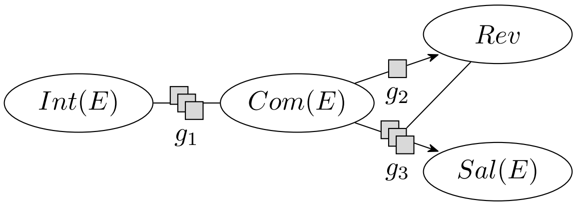

Fig. 5: A PD-PCFG that extends the PCFG depicted in Fig. 2 in the sense that an additional PRV $Int(E)$ has been added to the model. We omit the specification of the potential tables of the (par)factors for brevity.

Example 8 (Partially Directed Parametric Causal Factor Graph) *Figure 5 depicts a PD-PCFG M, which extends the PCFG given in Fig. 2. In particular, there is an additional PRV $Int(E)$ in M, which represents the intelligence of an employee. Moreover, for the sake of the example, there is no information available about the causal relationship between $Int(E)$ and $Com(E)$. As g_1 has no outgoing directed edge, we have $\mathrm{Ne}_{\boldsymbol{A}}(M, Com(E)) = \{Int(E)\}$ and $\mathrm{Ne}_{\boldsymbol{A}}(M, Int(E)) = \{Com(E)\}$, whereas the remaining PRVs have no undirected neighbour PRVs. We omit the set notation of M for brevity.*

Separation in a PD-PCFG is defined as in a PCFG, that is, the conditions specifying when a path is blocked are identical. Again, it is also possible to check whether PRVs (instead of ground randvars) are conditionally independent in a highly efficient manner on a lifted level in a PD-PCFG. In a PD-PCFG, every PRV $A(\mathcal{L})_{|C}$ is represented by a single variable node and thus, checking for conditional independence statements that involve A can be done by looking at this single variable node instead of taking into account all groundings of A individually.

In accordance with our previous assumptions, whenever we deal with a PD-PCFG in this article, we demand that whenever a probability distribution P is modelled using a PD-PCFG M, P satisfies the global Markov property with respect to M. We further stipulate that all directed edges in a PD-PCFG M are causal and hence accurately represent causal relationships between the involved randvars.

We next show how the computation of the effect of interventions can efficiently be realised in a PD-PCFG. An important challenge is that the effect of an intervention might differ depending on the actual causal relationships between the randvars. As there might be multiple possible causal explanations and we do not know the correct one, it is not always possible to uniquely determine the effect of an intervention.

The semantics of an intervention is defined on a fully directed graph. In particular, the interventional distribution is defined as a factorisation over the conditional probability distributions of all randvars, which are no intervention variables, given their parents. A PD-PCFG, however, might contain parfactor nodes without any outgoing directed edges, thereby possibly leading to unknown sets of parents for some randvars. Thus, when computing the effect of an intervention, we have to take all possible parent sets of the intervention variables into account. In general, not all combinations of orienting the undirected edges in a PD-PCFG are consistent with the conditional independence statements holding in the underlying probability distribution. More specifically, every PD-PCFG M represents a set of fully directed PCFGs obtained by orienting the undirected edges in M such that every parfactor node has exactly one outgoing directed edge and the resulting model entails the same conditional independence statements as M. We formalise this concept in the following definition.

Definition 17 (Consistent Extension) Let $M = (\boldsymbol{A} \cup \boldsymbol{G}, \boldsymbol{E}, \boldsymbol{\Phi})$ denote a PD-PCFG. A PCFG $M' = (\boldsymbol{A} \cup \boldsymbol{G}, \boldsymbol{E}', \boldsymbol{\Phi})$ is a *consistent extension* of M if

1. every directed edge $(g, A) \in \boldsymbol{E}$ is also in $\boldsymbol{E}'$,
2. for every parfactor node $g \in \boldsymbol{G}$ with $\mathrm{Ch}(M, g) = \emptyset$, exactly one edge $\{A, g\} \in \boldsymbol{E}$ is replaced by an edge (g, A) in $\boldsymbol{E}'$, and
3. M' entails the same conditional independence statements as M.

We denote the set of all consistent extensions of M as $[M]$.

In other words, a consistent extension of a PD-PCFG M is a PCFG obtained by orienting the undirected edges in M such that every parfactor node has exactly one outgoing directed edge and the implied conditional independence statements in the extension remain the same as in M. The set of consistent extensions $[M]$ of a PD-PCFG M might be empty. If $[M] \neq \emptyset$, we say that M is extendable. The concept of a consistent extension is closely related to the concept of a Markov equivalence class, which is a set of directed acyclic graphs that entail the same conditional independence statements (Andersson, Madigan, & Perlman, 1997; Verma & Pearl, 1990) (and thus, the set of consistent extensions of a partially directed acyclic graph is a subset of a Markov equivalence class).

Example 9 (Consistent Extension) *Consider again the PD-PCFG M depicted in Fig. 5. The set of consistent extensions $[M]$ of M contains two fully directed PD-PCFGs M_1 and M_2, which are illustrated in Fig. 6a and Fig. 6b, respectively. In M_1, the edge $g_1 - Com(E)$ has been replaced by an edge $g_1 \rightarrow Com(E)$ and in M_2, the edge $Int(E) - g_1$ has been replaced by an edge $g_1 \rightarrow Int(E)$. Both M_1 and M_2 entail the same conditional independence statements as M and could possibly model the correct underlying causal relationships but we do not know whether M_1 or M_2 is actually the correct model.*

We remark that the definition of a consistent extension refers to edges in the PD-PCFG M instead of referring to edges in the ground model $\mathrm{gr}(M)$. Thus, we assume that every edge $\{A, g\}$ in M represents a set of edges in $\mathrm{gr}(M)$ such that all of the edges in this set are oriented in the same way. For instance, in the ground graph of our running example, we do not allow any orientation where, e.g., $IntA - f_1 \rightarrow ComA$ and $IntB \leftarrow f_2 - ComB$ occur at the same time. As we assume that *Alice* and *Bob* are indistinguishable, we also assume identical edge orientations for their corresponding randvars. Defining consistent extensions on the lifted level is, however, not necessary to apply the approaches presented here. If wanted, the set of consistent extensions of a PD-PCFG M can also be defined with respect to the ground model of M (however,

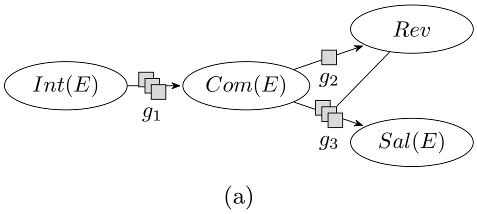

Fig. 6: A graphical illustration of the set of consistent extensions $[M] = \{M_1, M_2\}$ of the PD-PCFG M shown in Fig. 5. (a) shows the PCFG M_1, where $Com(E) - g_1$ has been oriented as $g_1 \to Com(E)$, and (b) shows the PCFG M_2, where $Int(E) - g_1$ has been oriented as $g_1 \to Int(E)$.

a definition with respect to the ground model is rather unintuitive as objects are not really indistinguishable if surrounding edges are oriented differently).

To determine the effect of an intervention, we need to know the parents of the intervention variables. As any PD-PCFG represents a set of consistent extensions, there are various possible parent sets for the intervention variables in general. Fortunately, we do not always have to consider all consistent extensions of a given PD-PCFG M to compute the effect of an intervention because there might be consistent extensions with identical parent sets for the intervention variables, thereby leading to the same effect of the intervention. Therefore, in case all parents of the randvars on which we intervene are known, we can uniquely determine the effect of an intervention even if there are still undirected edges present in M. This result has been shown for propositional partially directed acyclic graphs (Maathuis, Kalisch, & Bühlmann, 2009; Nandy, Maathuis, & Richardson, 2017) and we now transfer this result to PD-PCFGs.

Theorem 4 *Let $M = (\boldsymbol{A} \cup \boldsymbol{G}, \boldsymbol{E}, \boldsymbol{\Phi})$ denote a PD-PCFG, let $\boldsymbol{R} = \mathrm{gr}(\boldsymbol{A}) = \{R_1, \dots, R_n\}$ and let $do(R'_1 = r'_1, \dots, R'_k = r'_k)$ be an intervention on $\{R'_1, \dots, R'_k\} \subseteq \boldsymbol{R}$. If $\mathrm{Ne}_{\boldsymbol{R}}(\mathrm{gr}(M), R'_1) = \emptyset, \dots, \mathrm{Ne}_{\boldsymbol{R}}(\mathrm{gr}(M), R'_k) = \emptyset$, then the interventional distribution $P_{M'}(R_1 = r_1, \dots, R_n = r_n \mid do(R'_1 = r'_1, \dots, R'_k = r'_k))$ under the intervention $do(R'_1 = r'_1, \dots, R'_k = r'_k)$ is identical in all consistent extensions $M' \in [M]$ of M.*

Proof The interventional distribution of a PCFG $M' \in [M]$ is given by (Def. 14):

$$
\begin{aligned}
&P_{M'}(R_1 = r_1, \dots, R_n = r_n \mid do(R'_1 = r'_1, \dots, R'_k = r'_k)) \\
&\quad = \begin{cases} \prod\limits_{R_i \in \{R_1, \dots, R_n\} \setminus \{R'_1, \dots, R'_k\}} P(r_i \mid \mathrm{pa}_{\boldsymbol{R}}(\mathrm{gr}(M'), R_i)) & \text{if } \forall j \in \{1, \dots, k\} \colon r_j = r'_j \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}
$$

Given that $\mathrm{Ne}_{\boldsymbol{R}}(\mathrm{gr}(M), R'_1) = \emptyset, \dots, \mathrm{Ne}_{\boldsymbol{R}}(\mathrm{gr}(M), R'_k) = \emptyset$, the parents $\mathrm{Pa}_{\boldsymbol{R}}(\mathrm{gr}(M), R'_1), \dots, \mathrm{Pa}_{\boldsymbol{R}}(\mathrm{gr}(M), R'_k)$ of $R'_1, \dots, R'_k$ in M are fully known and identical in all $M' \in [M]$, i.e., $\mathrm{Pa}_{\boldsymbol{R}}(\mathrm{gr}(M), R'_i) = \mathrm{Pa}_{\boldsymbol{R}}(\mathrm{gr}(M'), R'_i)$ for all $M' \in [M]$ and all $i \in \{1, \dots, k\}$. The conditional probability distributions being removed from the product of the interventional distribution thus are identical for all $M' \in [M]$. Hence, it remains to be shown that the factorisation of all ground randvars that are not in $\{R'_1, \dots, R'_k\}$ is equivalent for all consistent extensions

$M' \in [M]$ of M. We know that every PCFG $M' \in [M]$ entails exactly the same conditional independence statements as M and thus, the factorisation induced by any PCFG $M' \in [M]$ is valid, i.e., all PCFGs $M' \in [M]$ encode the same underlying full joint probability distribution P as M. Therefore, as the parents of $\{R'_1, \dots, R'_k\}$ are identical in all $M' \in [M]$ and all $M' \in [M]$ encode the same probability distribution, the product over the conditional distributions of $R_i \in \{R_1, \dots, R_n\} \setminus \{R'_1, \dots, R'_k\}$ given their respective parents is identical in all $M' \in [M]$ (just as all BN structures over a fixed set of randvars entailing the same conditional independence statements induce equivalent factorisations of the underlying probability distribution). Consequently, the interventional distribution is identical in all consistent extensions $M' \in [M]$ of M. $\square$

A direct consequence of Thm. 4 is that the result of any interventional query is uniquely determined if there are no undirected edges connected to the intervention variables in the corresponding ground graph of a PD-PCFG.

Corollary 1 *Let $M = (\boldsymbol{A} \cup \boldsymbol{G}, \boldsymbol{E}, \boldsymbol{\Phi})$ denote a PD-PCFG, let $\boldsymbol{R} = \mathrm{gr}(\boldsymbol{A}) = \{Q, R_1, \dots, R_\ell, R'_1, \dots, R'_k\}$ and let $P(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k))$ be an interventional query. If $\mathrm{Ne}_{\boldsymbol{R}}(\mathrm{gr}(M), R'_1) = \emptyset, \dots, \mathrm{Ne}_{\boldsymbol{R}}(\mathrm{gr}(M), R'_k) = \emptyset$, then the result of $P(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k))$ is identical in all consistent extensions $M' \in [M]$ of M.*

Even though in practice, there might be undirected edges connected to the intervention variables, Thm. 4 implies that we do not have to consider all possible edge directions of the undirected edges in a PD-PCFG when computing the effect of an intervention. Instead, we only have to consider the possible directions of the undirected edges that are relevant for the intervention, i.e., the directions of the undirected edges that are connected to the intervention variables. Hence, we might not have to consider all consistent extensions of the given PD-PCFG. All terms required to answer the interventional query according to the truncated product formula can be computed by querying the PD-PCFG M, as the semantics of M is well-defined even if there are undirected edges in M (that is, the underlying full joint probability distribution is well-defined because its definition is independent of the edge directions in M). Intuitively, it becomes clear that the effect of an intervention is not guaranteed to be uniquely determined if there are undirected edges connected to the intervention variables because there might be various consistent extensions with different parent sets, resulting in multiple possible disjoint effects of the intervention. This result has been shown for propositional partially directed acyclic graphs (Maathuis et al., 2009) and we next show that it also holds for PD-PCFGs.

Theorem 5 *Let $M = (\boldsymbol{A} \cup \boldsymbol{G}, \boldsymbol{E}, \boldsymbol{\Phi})$ denote a PD-PCFG, let $\boldsymbol{R} = \mathrm{gr}(\boldsymbol{A}) = \{Q, R_1, \dots, R_\ell, R'_1, \dots, R'_k\}$ and let $P(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k))$ be an interventional query. If there exists a randvar $R'_i \in \{R'_1, \dots, R'_k\}$ such that $\mathrm{Ne}_{\boldsymbol{R}}(\mathrm{gr}(M), R'_i) \neq \emptyset$, then there might be consistent extensions $M_1, M_2 \in [M]$ of M such that the result of $P(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k))$ is not identical in M_1 and M_2.*

Proof If there exists a randvar $R'_i \in \{R'_1, \dots, R'_k\}$ such that $\mathrm{Ne}_{\boldsymbol{R}}(\mathrm{gr}(M), R'_i) \neq \emptyset$ holds, there might exist $M_1, M_2 \in [M]$ such that $\mathrm{Pa}_{\boldsymbol{R}}(\mathrm{gr}(M_1), R'_i) \neq \mathrm{Pa}_{\boldsymbol{R}}(\mathrm{gr}(M_2), R'_i)$. Then, by definition of the interventional distribution (Def. 14), the conditional probability distributions being removed from the product differ in M_1 and M_2, thereby yielding different interventional distributions for M_1 and M_2. □

Generally, there might be scenarios in which it is possible to uniquely determine the result of an interventional query even if there are undirected edges connected to the intervention variables, as possibly not all undirected edges can be oriented in both directions. In particular, some orientations might introduce a cycle or change the conditional independence statements implied by the graph structure and hence do not result in a consistent extension. In other words, it might be possible that there is just a single possible orientation of the parents of the intervention variables and in this case, the result of any interventional query can be uniquely determined.

Next, we gather the theoretical insights from this section to introduce the ELCI algorithm, which efficiently computes the effect of an intervention in a PD-PCFG.

6 The Extended Lifted Causal Inference Algorithm

Combining the insights from Thms. 4 and 5 naturally leads to an algorithm to compute the effect of interventions in a PD-PCFG. The idea is that all possible parent sets of the intervention variables have to be considered. If there is just one possible set of parents, the effect of the intervention can be uniquely determined, otherwise there are multiple possible effects that are enumerated. This idea is incorporated in the IDA algorithm and its variants (Guo & Perkovic, 2021; Liu, Fang, He, & Geng, 2020; Maathuis et al., 2009) for interventions $do(R' = r')$ with a single intervention variable R' in propositional causal models. Nandy et al. (2017) also consider the case of multiple intervention variables $R'_1, \dots, R'_k$ in propositional causal models, however, they operate in a different setting as they assume observational data generated by an unknown linear structural equation model with independent errors. Algorithm 2 displays the ELCI algorithm, which extends the idea of just considering the possible parent sets of intervention variables to handle arbitrary interventions $do(R'_1 = r'_1, \dots, R'_k = r'_k)$ with $k \geq 1$ in a PD-PCFG.

Given a PD-PCFG $M = (\boldsymbol{A} \cup \boldsymbol{G}, \boldsymbol{E}, \boldsymbol{\Phi})$ and an interventional query $P(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k))$, ELCI proceeds as follows to compute the set of all possible results for $P(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k))$. First, after initialising an empty set $\boldsymbol{P}$ to which possible query results are added (Line 1), ELCI splits the parfactors in M based on each $R'_i \in \{R'_1, \dots, R'_k\}$ (Line 2). In particular, ELCI splits every parfactor $\phi \in \boldsymbol{\Phi}$ for which there is an instance $\phi_j \in \mathrm{gr}(\phi)$ such that any intervention variable $R'_i \in \{R'_1, \dots, R'_k\}$ is a child of ϕ_j. ELCI then iterates over all possible combinations of parent sets (i.e., over all combinations of subsets of undirected neighbours) of the intervention variables $R'_1, \dots, R'_k$ (Lines 3 to 7). When considering the subsets of undirected neighbours, it is necessary that all subsets are jointly valid, that is, they are not allowed to alter the conditional independence statements encoded by the model and they must not introduce any directed cycles when oriented towards $R'_1, \dots, R'_k$. To ensure the validity of these subsets, they are required to form a clique. A clique

Algorithm 2 Extended Lifted Causal Inference

Input: A PD-PCFG $M = (\boldsymbol{A} \cup \boldsymbol{G}, \boldsymbol{E}, \boldsymbol{\Phi})$, and an interventional query $P(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k))$ with $\{Q, R'_1, \dots, R'_k\} \subseteq \boldsymbol{R} = \mathrm{gr}(\boldsymbol{A}) = \{Q, R_1, \dots, R_\ell, R'_1, \dots, R'_k\}$.

Output: The set of all possible answers to the interventional query $P(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k))$ in M.

1: $\boldsymbol{P} \leftarrow \emptyset$
2: $M \leftarrow$ PD-PCFG after splitting parfactors in M on each $R'_i \in \{R'_1, \dots, R'_k\}$
3: **for each** $\boldsymbol{C}_1 \subseteq \mathrm{Ne}_{\boldsymbol{R}}(M, R'_1), \dots, \boldsymbol{C}_k \subseteq \mathrm{Ne}_{\boldsymbol{R}}(M, R'_k)$ s.t. $\boldsymbol{C}_1, \dots, \boldsymbol{C}_k$ are cliques **do**
4: $\quad M' \leftarrow M$
5: $\quad$ **for each** intervention variable $R'_i \in \{R'_1, \dots, R'_k\}$ **do**
6: $\quad\quad$ **for each** undirected neighbour randvar $C \in \boldsymbol{C}_i$ of R'_i **do**
7: $\quad\quad\quad$ Orient $C - f - R'_i$ as $C - f \rightarrow R'_i$ in M'
8: $\quad$ **if** $[M'] = \emptyset$ **then**
9: $\quad\quad$ **continue**
10: $\quad M'' \leftarrow$ Any consistent extension from $[M']$
11: $\quad D \leftarrow \sum_{r_1 \in \mathrm{range}(R_1)} \dots \sum_{r_\ell \in \mathrm{range}(R_\ell)} \prod_{R_i \in \{Q, R_1, \dots, R_\ell\}} P(r_i \mid \mathrm{pa}_{\boldsymbol{R}}(M'', R_i))$
12: $\quad$ Add D to $\boldsymbol{P}$
13: **return** $\boldsymbol{P}$

$\boldsymbol{C}$ is a subset of nodes such that all pairs of nodes in $\boldsymbol{C}$ are directly connected via a parfactor node, that is, for each pair of nodes $C_1 \in \boldsymbol{C}$, $C_2 \in \boldsymbol{C}$ with $C_1 \neq C_2$ it holds that there exists a parfactor node $g \in \boldsymbol{G}$ such that there is an edge between C_1 and g as well as an edge between C_2 and g in $\boldsymbol{E}$ (either directed or undirected). By ensuring that the subsets of undirected neighbours form cliques, the orientation of the incident edges towards $R'_1, \dots, R'_k$ does not introduce any pattern $C_1 - g_1 \rightarrow R'_i \leftarrow g_2 - C_2$ where C_1 and C_2 are not directly connected via a parfactor node, as due to the clique property, C_1 and C_2 are always guaranteed to be directly connected via a factor node. In consequence, the conditional independence statements encoded by M' are guaranteed to be equivalent to those encoded by M (Maathuis et al., 2009). Having obtained a possible combination of parent sets of $R'_1, \dots, R'_k$, ELCI next extends the modified model M' to any PCFG from the set of consistent extensions of M', if such a consistent extension exists (Lines 8 to 10). In case there is no consistent extension (e.g., due to M' containing a directed cycle), ELCI continues with the next possible combination of parent sets. If there is a consistent extension $M'' \in [M']$, then the result of the provided query is given by applying the truncated product formula (Eq. (10)) in M''. As the parents of the intervention variables are fixed in M', the result of the given interventional query is identical in all consistent extensions of M' according to Thm. 4. Thus, ELCI computes the result of the interventional query in M'' by applying the truncated product formula and then adds the result to $\boldsymbol{P}$ (Lines 11 and 12). We remark that the computation can be simplified if it is known that the factors in M encode conditional probability distributions. Then, the modification from Prop. 1 can be applied as in the LCI algorithm (Alg. 1). After the result of the interventional query in M'' has been computed, ELCI repeats the above steps for the next parent set of the intervention variables $R'_1, \dots, R'_k$ until all possible parent sets have been taken into account. In the end, ELCI returns the set $\boldsymbol{P}$ containing all possible results for the interventional query $P(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k))$ (Line 13). In case there is no causal explanation for the given PD-PCFG M (that is, M has no consistent extension at all), ELCI returns an empty set. Such a situation might occur if M already contains a directed cycle or if there are undirected edges that cannot be oriented without introducing a directed cycle or altering the conditional independence statements implied by the model.

ELCI is also able to handle multiple query variables at once (then, Line 11 is adjusted such that only non-query variables are summed out). To compute a consistent extension in Line 10, ELCI might just call any of the efficient extension algorithms that are already available (Luttermann, Wienöbst, & Liśkiewicz, 2023; Verma & Pearl, 1992; Wienöbst, Bannach, & Liśkiewicz, 2021). Each of these algorithms operates on a partially directed acyclic graph and hence can be directly applied to the underlying causal graph of the PD-PCFG. While the set of probabilistic queries obtained from the truncated product formula refers to ground randvars, these queries can be answered using lifted probabilistic inference, e.g., by running the LJT algorithm (which is specifically designed to efficiently handle sets of queries) on M. The correctness of ELCI directly follows from Thm. 4.

Proposition 6 *The result computed by ELCI (Alg. 2) is correct, i.e., the returned set contains all possible results for the interventional query $P(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k))$ in the given PD-PCFG M.*

Proof From Thm. 4, we know that the result of any interventional query is identical in all consistent extensions of a PD-PCFG M if the parents of the intervention variables are known. Thus, to compute the set of possible results for $P(Q \mid do(R'_1 = r'_1, \dots, R'_k = r'_k))$ in M, it is sufficient to consider all possible parent sets of the intervention variables $R'_1, \dots, R'_k$ and compute the result of the given query in the resulting models. Due to Maathuis et al. (2009), it holds that any subset of undirected neighbours of the intervention variables needs to form a clique in order to obtain a valid orientation when orienting the edges towards the intervention variables (as otherwise, the conditional independence statements induced by the graph change). Consequently, by ensuring that undirected neighbours of $R'_1, \dots, R'_k$ form cliques before orienting them towards $R'_1, \dots, R'_k$, ELCI does not miss any possible parent set of the intervention variables. For any fixed set of parents of $R'_1, \dots, R'_k$, due to Thm. 4 it is then sufficient to consider any consistent extension and compute the result for the given query in it. As ELCI applies the truncated product formula from Def. 15 to compute the result of the given query, the correctness of ELCI follows. □

Given our assumption that the graph structure is identical for all groundings, it also holds that, e.g., given an intervention $do(Com(E) = \text{high})$, ELCI has to consider only two possible parent sets regardless of the number of employees while there are $2^{|\text{dom}(E)|}$ possible parent sets in an equivalent propositional model to consider. In a propositional model, it is also possible to reduce the number of possible parent sets when background knowledge is introduced, i.e., when knowing that specific randvars are actually representable by a single PRV.

Corollary 2 *Let $M = (\boldsymbol{A} \cup \boldsymbol{G}, \boldsymbol{E}, \boldsymbol{\Phi})$ be a PD-PCFG. When intervening on a PRV $A(\mathcal{L})_{|C} \in \boldsymbol{A}$, under the assumption that the graph structure is identical for all groundings, it holds that*

1. *ELCI considers $O(2^{|\mathrm{Ne}_{\boldsymbol{A}}(M,A)|})$ possible parent sets in the worst case, and*
2. *in a propositional model, $O(2^{\sum_{R \in \mathrm{gr}(A)} |\mathrm{Ne}_{\boldsymbol{R}}(\mathrm{gr}(M),R)|})$ possible parent sets have to be considered in the worst case.*

We refrain from empirically evaluating ELCI as we have already shown the superiority of LCI to the propositional case. ELCI (LCI, respectively) enables the computation of answers to interventional queries on a lifted level and hence can also be plugged into parameterised decision models (Gehrke, Braun, Möller, Waschkau, et al., 2019) to compute the action that maximises the expected utility under the semantics of interventions (instead of using the semantics of conditioning). A parameterised decision model originally extends a PFG by action nodes and utility nodes. Instead of using an undirected PFG as a basis, we can use a PD-PCFG (or a PCFG) as a basis for a parameterised decision model and then compute the expected utility of an action using ELCI (LCI, respectively), thereby allowing for first-order decision making.

7 Conclusion

We introduce PCFGs to combine lifted probabilistic inference with causal knowledge. To leverage the power of lifted inference for the computation of the effect of interventions, we further present the LCI algorithm, which operates on a lifted level and thus allows us to drastically speed up causal inference compared to running causal inference on an equivalent propositional (ground) model. LCI is a simple, yet effective algorithm to compute the effect of interventions. Moreover, we introduce PD-PCFGs as lifted causal models that allow to incorporate partial causal knowledge, thereby enabling lifted causal inference without the requirement of having a fully specified causal graph at hand. A PD-PCFG generalises the concept of a PCFG by incorporating both undirected and directed edges in the graph structure. To compute the effect of interventions in a PD-PCFG, we introduce the ELCI algorithm, which enumerates all possible results for an interventional query without grounding the entire model.

An interesting direction for future work is to relax the causal sufficiency assumption (Def. 5) and allow for hidden confounders (i.e., confounding variables that are not observed and hence not included in the set of randvars over which the model is defined) in a PD-PCFG. Under the presence of hidden confounders, the result of an interventional query might not be uniquely determinable anymore. Thus, the identification problem (i.e., the problem of determining whether a causal effect can be uniquely identified) becomes relevant if hidden confounders are present. In particular, an interesting question is whether the *do*-calculus introduced by Pearl (1995), which allows to rewrite an interventional query to obtain a probabilistic query free of *do*-expressions, can be applied to a PD-PCFG with hidden confounders.

Acknowledgements. This work is supported by the BMBF project AnoMed 16KISA057 and extends the works (Luttermann, Hartwig, et al., 2024) and (Luttermann, Braun, et al., 2024). The authors also would like to thank the anonymous reviewers for their valuable feedback and suggestions to improve the manuscript. This version of the article has been accepted for publication, after peer review but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: https://doi.org/10.1007/s10472-026-10009-1.

Data Availability Statement. The source code including the data set generators used for the experiments in this article is available at https://github.com/StatisticalRelationalAI/LiftedCausalInference.

Conflict of Interest. The authors declare that they have no conflict of interest.

References

Ahsan, R., Arbour, D., Zheleva, E. (2022). Relational Causal Models with Cycles: Representation and Reasoning. *Proceedings of the First Conference on Causal Learning and Reasoning (CLeaR-2022)* (pp. 1–18). PMLR.

Ahsan, R., Arbour, D., Zheleva, E. (2023). Learning Relational Causal Models with Cycles through Relational Acyclification. *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence (AAAI-2023)* (pp. 12164–12171). AAAI Press.

Andersson, S.A., Madigan, D., Perlman, M.D. (1997). A Characterization of Markov Equivalence Classes for Acyclic Digraphs. *The Annals of Statistics*, *25*, 505–541,

Apsel, U., & Brafman, R.I. (2012). Lifted MEU by Weighted Model Counting. *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-2012)* (pp. 1861–1867). AAAI Press.

Arbour, D., Garant, D., Jensen, D. (2016). Inferring Network Effects from Observational Data. *Proceedings of the Twenty-Second ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2016)* (pp. 715–724). ACM Press.

Braun, T. (2020). *Rescued from a Sea of Queries: Exact Inference in Probabilistic Relational Models* (Unpublished doctoral dissertation). University of Lübeck.

Braun, T., & Gehrke, M. (2022). Explainable and Explorable Decision Support. *Proceedings of the Twenty-Seventh International Conference on Conceptual Structures (ICCS-2022)* (pp. 99–114). Springer.

Braun, T., & Möller, R. (2016). Lifted Junction Tree Algorithm. *Proceedings of the Thirty-Ninth German Conference on Artificial Intelligence (KI-2016)* (pp. 30–42). Springer.

Braun, T., & Möller, R. (2018). Parameterised Queries and Lifted Query Answering. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-2018)* (pp. 4980–4986). IJCAI Organization.

De Salvo Braz, R., Amir, E., Roth, D. (2005). Lifted First-Order Probabilistic Inference. *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-2005)* (pp. 1319–1325). Morgan Kaufmann Publishers Inc.

De Salvo Braz, R., Amir, E., Roth, D. (2006). MPE and Partial Inversion in Lifted Probabilistic Variable Elimination. *Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-2006)* (pp. 1123–1130). AAAI Press.

Frey, B.J. (2003). Extending Factor Graphs so as to Unify Directed and Undirected Graphical Models. *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI-2003)* (pp. 257–264). Morgan Kaufmann Publishers Inc.

Frey, B.J., Kschischang, F.R., Loeliger, H.-A., Wiberg, N. (1997). Factor Graphs and Algorithms. *Proceedings of the Thirty-Fifth Annual Allerton Conference on Communication, Control, and Computing* (pp. 666–680). Allerton House.

Gehrke, M., Braun, T., Möller, R. (2018). Lifted Dynamic Junction Tree Algorithm. *Proceedings of the Twenty-Third International Conference on Conceptual Structures (ICCS-2018)* (pp. 55–69). Springer.

Gehrke, M., Braun, T., Möller, R. (2019). Lifted Temporal Maximum Expected Utility. *Proceedings of the Thirty-Second Canadian Conference on Artificial Intelligence (CANAI-2019)* (pp. 380–386). Springer.

Gehrke, M., Braun, T., Möller, R., Waschkau, A., Strumann, C., Steinhäuser, J. (2019). Lifted Maximum Expected Utility. *Proceedings of the First International Workshop on Artificial Intelligence in Health (AIH-2018)* (pp. 131–141). Springer.

Gehrke, M., Möller, R., Braun, T. (2020). Taming Reasoning in Temporal Probabilistic Relational Models. *Proceedings of the Twenty-Fourth European Conference on Artificial Intelligence (ECAI-2020)* (pp. 2592–2599). IOS Press.

Genesereth, M., & Kao, E.J. (2017). Relational Logic. In *Introduction to Logic* (pp. 63–81). Springer.

Guo, R., & Perkovic, E. (2021). Minimal Enumeration of All Possible Total Effects in a Markov Equivalence Class. *Proceedings of The Twenty-Fourth International Conference on Artificial Intelligence and Statistics (AISTATS-2021)* (pp. 2395–2403). PMLR.

Kisyński, J., & Poole, D. (2009). Constraint Processing in Lifted Probabilistic Inference. *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI-2009)* (pp. 293–302). AUAI Press.

Kschischang, F.R., Frey, B.J., Loeliger, H.-A. (2001). Factor Graphs and the Sum-Product Algorithm. *IEEE Transactions on Information Theory*, *47*, 498–519,

Lauritzen, S.L. (1996). *Graphical Models*. Clarendon Press.

Lee, S., & Honavar, V. (2015). Lifted Representation of Relational Causal Models Revisited: Implications for Reasoning and Structure Learning. *Proceedings of the UAI 2015 Conference on Advances in Causal Inference* (pp. 56–65). CEUR.

Lee, S., & Honavar, V. (2016). On Learning Causal Models from Relational Data. *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-2016)* (pp. 3263–3270). AAAI Press.

Lee, S., & Honavar, V. (2019). Towards Robust Relational Causal Discovery. *Proceedings of The Thirty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI-2019)* (pp. 345–355). PMLR.

Liu, Y., Fang, Z., He, Y., Geng, Z. (2020). Collapsible IDA: Collapsing Parental Sets for Locally Estimating Possible Causal Effects. *Proceedings of the Thirty-Sixth Conference on Uncertainty in Artificial Intelligence (UAI-2020)* (pp. 290–299). PMLR.

Luttermann, M., Braun, T., Möller, R., Gehrke, M. (2024). Estimating Causal Effects in Partially Directed Parametric Causal Factor Graphs. *Proceedings of the Sixteenth International Conference on Scalable Uncertainty Management (SUM-2024)* (pp. 265–280). Springer.

Luttermann, M., Hartwig, M., Braun, T., Möller, R., Gehrke, M. (2024). Lifted Causal Inference in Relational Domains. *Proceedings of the Third Conference on Causal Learning and Reasoning (CLeaR-2024)* (pp. 827–842). PMLR.

Luttermann, M., Wienöbst, M., Liśkiewicz, M. (2023). Practical Algorithms for Orientations of Partially Directed Graphical Models. *Proceedings of the Second Conference on Causal Learning and Reasoning (CLeaR-2023)* (pp. 259–280). PMLR.

Maathuis, M.H., Kalisch, M., Bühlmann, P. (2009). Estimating High-Dimensional Intervention Effects from Observational Data. *The Annals of Statistics*, *37*, 3133–3164,

Maier, M., Marazopoulou, K., Arbour, D., Jensen, D. (2013). A Sound and Complete Algorithm for Learning Causal Models from Relational Data. *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence (UAI-2013)* (pp. 371–380). AUAI Press.

Maier, M., Taylor, B., Oktay, H., Jensen, D. (2010). Learning Causal Models of Relational Domains. *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2010)* (pp. 531–538). AAAI Press.

Meert, W., Taghipour, N., Blockeel, H. (2010). First-Order Bayes-Ball. *Proceedings of the Fourteenth European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD-2010)* (pp. 369–384). Springer.

Milch, B., Zettlemoyer, L.S., Kersting, K., Haimes, M., Kaelbling, L.P. (2008). Lifted Probabilistic Inference with Counting Formulas. *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI-2008)* (pp. 1062–1068). AAAI Press.

Nandy, P., Maathuis, M.H., Richardson, T.S. (2017). Estimating the Effect of Joint Interventions from Observational Data in Sparse High-Dimensional Settings. *The Annals of Statistics*, *45*, 647–674,

Niepert, M., & Van den Broeck, G. (2014). Tractability through Exchangeability: A New Perspective on Efficient Probabilistic Inference. *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI-2014)* (pp. 2467–2475). AAAI Press.

Pearl, J. (1986). Fusion, Propagation, and Structuring in Belief Networks. *Artificial Intelligence*, *29*, 241–288,

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc.

Pearl, J. (1995). Causal Diagrams for Empirical Research. *Biometrika*, *82*, 669–688,

Pearl, J. (2009). *Causality: Models, Reasoning and Inference* (2nd ed.). Cambridge University Press.

Pearl, J., Glymour, M., Jewell, N.P. (2016). *Causal Inference in Statistics: A Primer* (1st ed.). Wiley.

Peters, J., Janzing, D., Schölkopf, B. (2017). *Elements of Causal Inference: Foundations and Learning Algorithms.* MIT Press.

Poole, D. (2003). First-Order Probabilistic Inference. *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-2003)* (pp. 985–991). Morgan Kaufmann Publishers Inc.

Salimi, B., Parikh, H., Kayali, M., Getoor, L., Roy, S., Suciu, D. (2020). Causal Relational Learning. *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (pp. 241–256). ACM Press.

Shachter, R.D. (1998). Bayes-Ball: Rational Pastime (For Determining Irrelevance and Requisite Information in Belief Networks and Influence Diagrams). *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-1998)* (pp. 480–487). Morgan Kaufmann Publishers Inc.

Spirtes, P., Glymour, C., Scheines, R. (2000). *Causation, Prediction, and Search* (2nd ed.). MIT Press.

Taghipour, N., Fierens, D., Davis, J., Blockeel, H. (2013). Lifted Variable Elimination: Decoupling the Operators from the Constraint Language. *Journal of Artificial Intelligence Research*, *47*, 393–439,

Taghipour, N., Fierens, D., Van den Broeck, G., Davis, J., Blockeel, H. (2013). Completeness Results for Lifted Variable Elimination. *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics (AISTATS-2013)* (pp. 572–580). PMLR.

Van den Broeck, G. (2011). On the Completeness of First-Order Knowledge Compilation for Lifted Probabilistic Inference. *Advances in Neural Information Processing Systems 24 (NIPS-2011)* (pp. 1386–1394). Curran Associates, Inc.

Verma, T., & Pearl, J. (1990). Equivalence and Synthesis of Causal Models. *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence (UAI-1990)* (pp. 255–270). Elsevier.

Verma, T., & Pearl, J. (1992). An Algorithm for Deciding if a Set of Observed Independencies Has a Causal Explanation. *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence (UAI-1992)* (pp. 323–330). Morgan Kaufmann Publishers Inc.

Wienöbst, M., Bannach, M., Liśkiewicz, M. (2021). Extendability of Causal Graphical Models: Algorithms and Computational Complexity. *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI-2021)* (pp. 1248–1257). PMLR.

Winn, J. (2012). Causality with Gates. *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics (AISTATS-2012)* (pp. 1314–1322). PMLR.