Approximate Lifted Model Construction*

Malte Luttermann[1], **Jan Speller**[2], **Marcel Gehrke**[3], **Tanya Braun**[2], **Ralf Möller**[3] and **Mattis Hartwig**[1]

[1]German Research Center for Artificial Intelligence (DFKI), Lübeck, Germany
[2]Data Science Group, University of Münster, Germany
[3]Institute for Humanities-Centered Artificial Intelligence, University of Hamburg, Germany
{malte.luttermann,mattis.hartwig}@dfki.de, {marcel.gehrke,ralf.moeller}@uni-hamburg.de,
{jan.speller,tanya.braun}@uni-muenster.de

Abstract

Probabilistic relational models such as parametric factor graphs enable efficient (lifted) inference by exploiting the indistinguishability of objects. In lifted inference, a representative of indistinguishable objects is used for computations. To obtain a relational (i.e., lifted) representation, the Advanced Colour Passing (ACP) algorithm is the state of the art. The ACP algorithm, however, requires underlying distributions, encoded as potential-based factorisations, to exactly match to identify and exploit indistinguishabilities. Hence, ACP is unsuitable for practical applications where potentials learned from data inevitably deviate even if associated objects are indistinguishable. To mitigate this problem, we introduce the ε-Advanced Colour Passing (ε-ACP) algorithm, which allows for a deviation of potentials depending on a hyperparameter ε. ε-ACP efficiently uncovers and exploits indistinguishabilities that are not exact. We prove that the approximation error induced by ε-ACP is strictly bounded and our experiments show that the approximation error is close to zero in practice.

1 Introduction

Probabilistic relational models, denoted as parametric factor graphs (PFGs), combine probabilistic modelling with relational logic (that is, first-order logic with known universes). By introducing logical variables (logvars) to represent sets of indistinguishable objects, PFGs allow lifted inference algorithms to use a representative of indistinguishable objects for efficient computations. In practice, however, when learning the underlying probability distribution of a PFG from data, indistinguishable objects are often not recognised. In particular, considering a potential-based factorisation of the probability distribution, learned potentials inevitably deviate even for indistinguishable objects due to estimates from data. To mitigate this issue and ensure the practical applicability of obtaining a compact representation for lifted inference, we solve the problem of constructing a lifted representation while taking into account small deviations of potentials for indistinguishable objects. In particular, we ensure that the obtained lifted representation is approximately equivalent to a given propositional (ground) representation by solving an optimisation problem to minimise the approximation error. Allowing for small deviations between potentials is essential for practical applications, where potentials, for instance, are learned from data and hence are subject to inaccuracies. For example, consider the probabilities $p_1 = 0.501$ and $p_2 = 0.499$. In case p_1 and p_2 are estimates from data, it is likely that p_1 and p_2 should actually be considered equal.

Poole [2003] first introduces PFGs, which combine relational logic and probabilistic models, and lifted variable elimination as a lifted inference algorithm to perform lifted probabilistic inference in PFGs. In probabilistic inference, lifting exploits indistinguishabilities in a probabilistic model, allowing to carry out query answering more efficiently while maintaining exact answers [Niepert and Van den Broeck, 2014]. Since its first introduction by Poole [2003], lifted variable elimination has continuously been refined by many researchers to reach its current form [De Salvo Braz *et al.*, 2005, 2006; Milch *et al.*, 2008; Kisyński and Poole, 2009; Taghipour *et al.*, 2013; Braun and Möller, 2018]. More recently, Luttermann *et al.* [2024b,c] extend PFGs to incorporate causal knowledge and thereby allow to perform lifted causal inference. To perform lifted probabilistic (or causal) inference, the lifted representation (e.g., a PFG) has to be constructed first. The Advanced Colour Passing (ACP) algorithm [Luttermann *et al.*, 2024a,d,e,f], which generalises the CompressFactorGraph algorithm [Kersting *et al.*, 2009; Ahmadi *et al.*, 2013], is the current state of the art to construct a PFG from a propositional model with equivalent semantics. ACP employs a colour passing procedure to detect symmetric subgraphs, similar to the Weisfeiler-Leman algorithm [Weisfeiler and Leman, 1968], which is a well-known algorithm to test for graph isomorphism. While ACP is able to construct a PFG entailing equivalent semantics as a given propositional model, ACP requires potentials to exactly match, which is a significant limitation for practical applications.

In this paper, we contribute the ε-Advanced Colour Passing (ε-ACP) algorithm, which solves the problem of constructing an approximate lifted representation with a minimal approximation error and thereby makes the construction

*Extended version of paper accepted to the Proceedings of the 34th International Joint Conference on Artificial Intelligence (IJCAI-2025).

of a lifted model applicable in practice. The ε-ACP algorithm allows for potentials to deviate by a factor of ε to still be considered identical, where ε is a hyperparameter controlling the required agreement between potentials. Thus, the hyperparameter ε controls the trade-off between the exactness and the compactness of the lifted representation obtained by ε-ACP. We further prove that the approximation error induced by ε-ACP is strictly bounded. In addition to the theoretical bounds, we empirically show that ε-ACP significantly reduces run times for inference while at the same time keeping the approximation error close to zero.

The remaining part of this paper is structured as follows. We begin by introducing background information and notations in Sec. 2. Thereafter, we introduce the ε-ACP algorithm to solve the problem of constructing an approximate lifted representation with a minimal approximation error. We then prove that the approximation error induced by ε-ACP is strictly bounded and show that the given bound is optimal. Finally, we empirically demonstrate that in practice, the actual approximation error induced by ε-ACP is well below the theoretical bounds before we conclude the paper.

2 Background

We first define factor graphs (FGs) as propositional models and afterwards introduce the idea of lifted representations such as PFGs. An FG is a probabilistic graphical model to compactly represent a probability distribution over a set of random variables (randvars) by factorising the distribution [Frey *et al.*, 1997; Kschischang *et al.*, 2001].

Definition 1 (Factor Graph)**.** *An* FG $M = (\boldsymbol{V}, \boldsymbol{E})$ *is an undirected bipartite graph consisting of a node set* $\boldsymbol{V} = \boldsymbol{R} \cup \boldsymbol{\Phi}$*, where* $\boldsymbol{R} = \{R_1, \dots, R_n\}$ *is a set of variable nodes (randvars) and* $\boldsymbol{\Phi} = \{\phi_1, \dots, \phi_m\}$ *is a set of factor nodes (functions), as well as a set of edges* $\boldsymbol{E} \subseteq \boldsymbol{R} \times \boldsymbol{\Phi}$*. There is an edge between a variable node* R_i *and a factor node* ϕ_j *in* $\boldsymbol{E}$ *if* R_i *appears in the argument list of* ϕ_j*. A factor* $\phi_j(\mathcal{R}_j)$ *defines a function* $\phi_j \colon \times_{R \in \mathcal{R}_j} \mathrm{range}(R) \mapsto \mathbb{R}^+$ *that maps the ranges of its arguments* $\mathcal{R}_j$ *(a sequence of randvars from* $\boldsymbol{R}$*) to a positive real number, called potential. The term* $\mathrm{range}(R)$ *denotes the possible values a randvar* R *can take. We define the joint potential for an assignment* $\boldsymbol{r}$ *(where* $\boldsymbol{r}$ *is a shorthand notation for* $\boldsymbol{R} = \boldsymbol{r}$*) as*

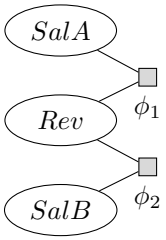
$$\psi(\boldsymbol{r}) = \prod_{j=1}^{m} \phi_j(\boldsymbol{r}_j), \tag{1}$$

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

$$P_M(\boldsymbol{r}) = \frac{1}{Z} \prod_{j=1}^{m} \phi_j(\boldsymbol{r}_j) = \frac{1}{Z} \psi(\boldsymbol{r}), \tag{2}$$

where $Z = \sum_{\boldsymbol{r}} \prod_{j=1}^{m} \phi_j(\boldsymbol{r}_j)$ *is the normalisation constant.*

Example 1. *Consider the FG depicted in Fig. 1, which models the interplay between the revenue* Rev *of a company and the salary of two employees, denoted as* $SalA$ *and* $SalB$*. We have* $\boldsymbol{R} = \{SalA, SalB, Rev\}$*,* $\boldsymbol{\Phi} = \{\phi_1, \phi_2\}$*, and* $\boldsymbol{E} = \{\{SalA, \phi_1\}, \{Rev, \phi_1\}, \{Rev, \phi_2\}, \{SalB, \phi_2\}\}$*. For the sake of this example, let* $\mathrm{range}(SalA) = \mathrm{range}(SalB) = \mathrm{range}(Rev) = \{\mathrm{low}, \mathrm{high}\}$*. The potential tables of* ϕ_1 *and* ϕ_2 *are shown on the right. In particular, it holds that* $\phi_1(\mathrm{high}, \mathrm{high}) = \varphi_1$*,* $\phi_1(\mathrm{high}, \mathrm{low}) = \varphi_2$*, and so on, where all* $\varphi_i, \varphi_i' \in \mathbb{R}^+$ *are arbitrary positive real numbers.*



$SalA$	Rev	$\phi_1(SalA, Rev)$
high	high	φ_1
high	low	φ_2
low	high	φ_3
low	low	φ_4

$SalB$	Rev	$\phi_2(SalB, Rev)$
high	high	φ_1'
high	low	φ_2'
low	high	φ_3'
low	low	φ_4'

Figure 1: An FG modelling the interplay between the revenue of a company (Rev) and the salaries of two employees ($SalA$, $SalB$). The potential tables of the factors are shown on the right.

Probabilistic inference describes the task of computing marginal distributions of randvars given observations for other randvars. In other words, probabilistic inference refers to query answering, where a query is defined as follows.

Definition 2 (Query)**.** *A query* $P(Q \mid E_1 = e_1, \dots, E_k = e_k)$ *consists of a query term* Q *and a set of events* $\{E_j = e_j\}_{j=1}^{k}$ *(called evidence), where* Q *and each* E_j *are randvars. To query a specific probability instead of a probability distribution, the query term is an event* $Q = q$*.*

Example 2. *Take a look at the FG shown in Fig. 1. The query* $P(SalA \mid Rev = \mathrm{high})$ *asks for the probability distribution of A's salary given that the company has a high revenue.*

When considering relations between objects, there are often groups of indistinguishable objects that behave identically (or at least similarly). Lifted representations such as PFGs exploit identical behaviour to enable scalable probabilistic inference with respect to domain sizes of logvars. To illustrate the idea behind lifting, consider the following example.

Example 3. *Consider the FG depicted in Fig. 1 and the query* $P(Rev = \mathrm{high})$*. Then, it holds that*

$$\begin{aligned} &P(Rev = \mathrm{high}) \\ &\quad = \sum_{a \in \mathrm{range}(SalA)} \sum_{b \in \mathrm{range}(salB)} P(a, b, \mathrm{high}) \\ &\quad = \frac{1}{Z} \sum_{a \in \mathrm{range}(SalA)} \sum_{b \in \mathrm{range}(salB)} \phi_1(a, \mathrm{high}) \phi_2(b, \mathrm{high}) \\ &\quad = \frac{1}{Z} \Big(\varphi_1 \varphi_1' + \varphi_3 \varphi_1' + \varphi_1 \varphi_3' + \varphi_3 \varphi_3' \Big). \end{aligned}$$

If employees A and B are indistinguishable, that is, if it holds that $\varphi_i = \varphi_i'$ *for all* $i \in \{1, \dots, 4\}$*, we can simplify the computation and obtain*

$$P(Rev = \mathrm{high})$$

$$= \frac{1}{Z} \sum_{a \in \text{range}(SalA)} \phi_1(a, \text{high}) \sum_{b \in \text{range}(salB)} \phi_2(b, \text{high})$$
$$= \frac{1}{Z} \left(\sum_{a \in \text{range}(SalA)} \phi_1(a, \text{high}) \right)^2$$
$$= \frac{1}{Z} \Big(\varphi_1 + \varphi_3 \Big)^2 .$$

Example 3 illustrates that in case A and B are indistinguishable, we can select one representative (e.g., A) and reduce the number of factors to consider for computations. The idea of exploiting exponentiation can be generalised to groups of k indistinguishable objects (e.g., employees) to significantly reduce the computational effort when answering queries. Indistinguishable objects frequently occur in relational models and are relevant in many real world domains. For example, in an epidemic domain, each person influences the probability of an epidemic equally, i.e., the probability of an epidemic depends on the number of sick people and is independent of which individual people are sick.

As we have seen, to exploit indistinguishabilities, we need to find factors with identical potential tables. Currently, the ACP algorithm is the state of the art to find factors with identical potential tables and group them together to obtain a lifted representation such as a PFG.[1] In Ex. 3, we assume $\varphi_i = \varphi'_i$ for all $i \in \{1, \ldots, 4\}$, which is required by ACP. However, in practice, we often face situations where estimates of potentials lead to deviations such that $\varphi_i = \varphi'_i \cdot (1 \pm \varepsilon)$ for a small $\varepsilon \in \mathbb{R}^+$. The ACP algorithm does not group factors if they are not strictly equal and thus is hardly applicable in practice to identify factors that should be grouped. To address this limitation, we next investigate how indistinguishabilities can be approximated when constructing a lifted representation.

3 Approximation of Indistinguishabilities

To control the trade-off between the exactness and compactness of the resulting lifted representation when grouping factors with approximately equivalent semantics, we now introduce a hyperparameter $\varepsilon \in \mathbb{R}^+$. More specifically, we allow for a maximum relative deviation of factor $(1 \pm \varepsilon)$, i.e., two potentials φ and φ' are considered approximately equivalent if $\varphi \in [\varphi' \cdot (1-\varepsilon), \varphi' \cdot (1+\varepsilon)]$ and $\varphi' \in [\varphi \cdot (1-\varepsilon), \varphi \cdot (1+\varepsilon)]$.[2] The notion of ε-equivalence formally captures the idea of approximately equivalent factors.

Definition 3 (ε-Equivalent Factors)**.** *Let $\varepsilon \in \mathbb{R}^+$ be a positive real number. Two potentials $\varphi_1 \in \mathbb{R}^+$ and $\varphi_2 \in \mathbb{R}^+$ are* ε-equivalent*, denoted as $\varphi_1 =_\varepsilon \varphi_2$, if $\varphi_1 \in [\varphi_2 \cdot (1 - \varepsilon), \varphi_2 \cdot (1 + \varepsilon)]$ and $\varphi_2 \in [\varphi_1 \cdot (1 - \varepsilon), \varphi_1 \cdot (1 + \varepsilon)]$. Further, two factors $\phi_1(R_1, \ldots, R_n)$ and $\phi_2(R'_1, \ldots, R'_n)$ are* ε-equivalent*, denoted as $\phi_1 =_\varepsilon \phi_2$, if there exists a permutation π of $\{1, \ldots, n\}$ such that for all assignments $(r_1, \ldots, r_n) \in \times_{i=1}^n \text{range}(R_i)$, where $\phi_1(r_1, \ldots, r_n) = \varphi_1$ and $\phi_2(r_{\pi(1)}, \ldots, r_{\pi(n)}) = \varphi_2$, it holds that $\varphi_1 =_\varepsilon \varphi_2$.*

[1] A formal description and a detailed explanation of the ACP algorithm is provided in Appendix B.

[2] Since potentials are arbitrary positive real numbers (and thus might differ in their order of magnitude), we allow for a relative deviation instead of using an absolute deviation.

Note that the notion of ε-equivalence is symmetric and as a necessary condition to be ε-equivalent, ϕ_1 and ϕ_2 must be defined over the same function domain and hence must have the same number of arguments. We further remark that indistinguishable objects are not guaranteed to be located at the same position in their respective factors, which is the reason we consider permutations of the arguments. For example, in Fig. 1, $SalB$ could also be the second argument of ϕ_2: Then, the potential table of ϕ_2 would read φ'_1, φ'_3, φ'_2, φ'_4 from top to bottom (if we keep the order of the assignments), i.e., even if $\varphi_i = \varphi'_i$ for all $i \in \{1, \ldots, 4\}$, we would only be able to exploit this property if we permute the arguments of ϕ_2 (or of ϕ_1) such that $SalA$ and $SalB$ are located at the same positions in their respective argument lists. A full example to showcase the role of permutations is given in Appendix C. For simplicity, we assume that π is the identity function throughout this paper (however, all results also apply for arbitrary choices of π [Luttermann *et al.*, 2024a]).

Example 4. *Let $\varphi = 0.49$, $\varphi' = 0.5$, and $\varepsilon = 0.1$. Then, it holds that $\varphi' = 0.5 \in [\varphi \cdot (1-\varepsilon) = 0.441, \varphi \cdot (1+\varepsilon) = 0.539]$ and $\varphi = 0.49 \in [\varphi' \cdot (1-\varepsilon) = 0.45, \varphi' \cdot (1+\varepsilon) = 0.55]$. In consequence, φ and φ' are ε-equivalent.*

To group ε-equivalent factors such that we can use a representative and exploit exponentiation to reduce the number of factors to consider during computations, we need to find ε-equivalent factors and change their potentials in a way that their potential tables become identical. We first address the issue of detecting ε-equivalent factors and then show how potentials are changed to minimise the approximation error.

3.1 Finding and Grouping ε-Equivalent Factors

A problem when searching for groups of ε-equivalent factors is that ε-equivalence is not transitive. More specifically, it might happen that there are factors ϕ_1, ϕ_2, and ϕ_3 such that $\phi_1 =_\varepsilon \phi_2$ and $\phi_2 =_\varepsilon \phi_3$ but $\phi_1 \neq_\varepsilon \phi_3$.

Example 5. *Consider the factors $\phi_1(R_1^1, R_2^1)$, $\phi_2(R_1^2, R_2^2)$, and $\phi_3(R_1^3, R_2^3)$ and their potential tables depicted in Table 1a. For the sake of this example, let $\varepsilon = 0.1$. The intervals allowing for a deviation of factor $(1 \pm \varepsilon)$ according to Def. 3 are shown in Table 1b. Since all potentials of ϕ_1 lie within the corresponding intervals of ϕ_2 (and vice versa), it holds that $\phi_1 =_\varepsilon \phi_2$. Analogously, it holds that $\phi_2 =_\varepsilon \phi_3$. However, due to $0.75 \notin [0.756, 0.924]$ (as well as $0.84 \notin [0.675, 0.825]$), it holds that $\phi_1 \neq_\varepsilon \phi_3$.*

Due to the non-transitivity of ε-equivalence, we cannot simply group a factor ϕ with a group of ε-equivalent factors $\boldsymbol{G} = \{\phi_1, \ldots, \phi_k\}$ if ϕ is ε-equivalent to any $\phi_i \in \boldsymbol{G}$. Doing so would give rise to the issue of cascading errors, that is, in the worst case, completely different factors could be grouped together (e.g., assuming $\varepsilon = 0.1$, the potential 1.0 can be grouped with the potential 0.9, which itself can be grouped with the potential 0.81, and so on). To avoid cascading errors, we thus ensure a factor ϕ is only added to a group of ε-equivalent factors $\boldsymbol{G}$ if ϕ is ε-equivalent to *all* factors in $\boldsymbol{G}$.

Next, we need to solve the problem of changing the potentials for every group of pairwise ε-equivalent factors $\boldsymbol{G} = \{\phi_1, \ldots, \phi_k\}$. To exploit exponentiation and thus avoid looking at every factor individually, the changes must ensure that

R_1^i	R_2^i	$\phi_1(R_1^1, R_2^1)$	$\phi_2(R_1^2, R_2^2)$	$\phi_3(R_1^3, R_2^3)$
high	high	0.75	0.8	0.84
high	low	0.33	0.3	0.31
low	high	0.48	0.5	0.51
low	low	0.22	0.2	0.22

(a)

$\phi_1 \cdot (1 \mp \varepsilon)$	$\phi_2 \cdot (1 \mp \varepsilon)$	$\phi_3 \cdot (1 \mp \varepsilon)$
$[0.675, 0.825]$	$[0.72, 0.88]$	$[0.756, 0.924]$
$[0.297, 0.363]$	$[0.27, 0.33]$	$[0.279, 0.341]$
$[0.432, 0.528]$	$[0.45, 0.55]$	$[0.459, 0.561]$
$[0.198, 0.242]$	$[0.18, 0.22]$	$[0.198, 0.242]$

(b)

Table 1: (a) The potential tables of exemplary factors $\phi_1(R_1^1, R_2^1)$, $\phi_2(R_1^2, R_2^2)$, and $\phi_3(R_1^3, R_2^3)$, where the randvars R_1^i and R_2^i, $i \in \{1, 2, 3\}$, all have the same range $\{\text{low}, \text{high}\}$, and (b) the intervals resulting from a deviation of factor $\varepsilon = 0.1$. We omit the arguments of the factors and their assignments for brevity (the order of the assignments is identical to the order in (a)).

all factors map to the same potentials. At the same time, we aim to minimise the approximation error, that is, we want to apply the smallest possible change to the potentials. Formally, the goal is to find ϕ^* such that

$$\phi^* = \arg\min_{\phi_j} \sum_{\phi_i \in \boldsymbol{G}} Err(\phi_i, \phi_j), \tag{3}$$

where $Err(\phi_i, \phi_j)$ is the sum of squared deviations between the potentials of ϕ_i and ϕ_j:

$$Err(\phi_i, \phi_j) = \sum_{r_1, \ldots, r_n} \Big(\phi_i(r_1, \ldots, r_n) - \phi_j(r_1, \ldots, r_n)\Big)^2, \tag{4}$$

with $r_1, \ldots, r_n$ denoting the possible assignments of the arguments of ϕ_i and ϕ_j.[3] To obtain identical potentials within a group $\boldsymbol{G} = \{\phi_1, \ldots, \phi_k\}$, our goal is to update the factors in $\boldsymbol{G}$ such that $\phi_1 = \phi^*, \ldots, \phi_k = \phi^*$.

Thus, we now solve the problem of finding ϕ^*. In fact, it holds that for any set of numbers $\{\varphi_1, \ldots, \varphi_k\}$, the arithmetic mean $\bar{\varphi} = \frac{1}{k} \sum_{i=1}^{k} \varphi_i$ minimises the sum of squared deviations $\sum_{i=1}^{k} (\varphi_i - \bar{\varphi})^2$, i.e., replacing $\bar{\varphi}$ by any other value would increase the sum of squared deviations.

Theorem 1. *Let $\varphi_1, \ldots, \varphi_k \in \mathbb{R}^+$. It holds that the arithmetic mean $\bar{\varphi} = \frac{1}{k} \sum_{i=1}^{k} \varphi_i$ is the optimal choice for $\varphi^* = \arg\min_{\hat{\varphi}} \sum_{i=1}^{k} (\varphi_i - \hat{\varphi})^2$.*

Theorem 1 is a well-known property of the arithmetic mean (a proof is given in Appendix A). As Eq. (3) aims to minimise a sum over a sum of squared deviations $Err(\phi_i, \phi_j)$, the sum in Eq. (3) becomes minimal if we minimise $Err(\phi_i, \phi_j)$, i.e., the right hand side of Eq. (4), according to Thm. 1. Therefore, for any group $\boldsymbol{G} = \{\phi_1, \ldots, \phi_k\}$ of pairwise ε-equivalent factors, we set $\phi_1 = \phi^*, \ldots, \phi_k = \phi^*$ such that

$$\phi^*(\boldsymbol{r}) = \frac{1}{k} \sum_{i=1}^{k} \phi_i(\boldsymbol{r}) \tag{5}$$

for all possible assignments $\boldsymbol{r} = r_1, \ldots, r_n$ to ensure that all factors in $\boldsymbol{G}$ map to the same potentials while minimising the cumulative squared deviation of the group $\boldsymbol{G}$.

Next, we compile the insights on finding and grouping ε-equivalent factors into the ε-ACP algorithm, which paves the way to apply lifted model construction in practice.

Algorithm 1 ε-Advanced Colour Passing

Input: An FG $M = (\boldsymbol{R} \cup \boldsymbol{\Phi}, \boldsymbol{E})$, a hyperparameter $\varepsilon \in \mathbb{R}^+$, and a set of observed events (evidence) $\boldsymbol{O} = \{E_1 = e_1, \ldots, E_\ell = e_\ell\}$.
Output: A lifted representation M', encoded as a PFG, which is approximately equivalent to M.

▷ Phase I: Find groups of pairwise ε-equivalent factors
1: $\boldsymbol{G} \leftarrow \{\{\phi_1\}\}$
2: **for each** factor $\phi_i \in \boldsymbol{\Phi} \setminus \{\phi_1\}$ **do**
3: $\quad \boldsymbol{C} \leftarrow \emptyset$
4: $\quad$ **for each** group $\boldsymbol{G}_j \in \boldsymbol{G}$ **do**
5: $\quad\quad$ **if** $\forall \phi_k \in \boldsymbol{G}_j \colon \phi_i =_\varepsilon \phi_k$ **then**
6: $\quad\quad\quad \boldsymbol{C} \leftarrow \boldsymbol{C} \cup \{\boldsymbol{G}_j\}$
7: $\quad$ **if** $\boldsymbol{C} \neq \emptyset$ **then**
8: $\quad\quad \boldsymbol{G}_j \leftarrow \arg\min_{\boldsymbol{C}_i \in \boldsymbol{C}} \sum_{\phi_j \in \boldsymbol{C}_i} Err(\phi_i, \phi_j)$
9: $\quad\quad \boldsymbol{G}_j \leftarrow \boldsymbol{G}_j \cup \{\phi_i\}$
10: $\quad$ **else**
11: $\quad\quad \boldsymbol{G} \leftarrow \boldsymbol{G} \cup \{\{\phi_i\}\}$
▷ Phase II: Assign colours to factors and run ACP
12: **for each** group $\boldsymbol{G}_j \in \boldsymbol{G}$ **do**
13: $\quad$ **for each** factor $\phi_i \in \boldsymbol{G}_j$ **do**
14: $\quad\quad \phi_i.colour \leftarrow j$
15: $\boldsymbol{G}' \leftarrow$ Call ACP on M and $\boldsymbol{O}$ using the assigned colours
▷ Phase III: Update potentials
16: **for each** group $\boldsymbol{G}_j \in \boldsymbol{G}'$ **do**
17: $\quad \phi^*(\boldsymbol{r}) \leftarrow \frac{1}{|\boldsymbol{G}_j|} \sum_{\phi_i \in \boldsymbol{G}_j} \phi_i(\boldsymbol{r})$ for all assignments $\boldsymbol{r}$
18: $\quad$ **for each** factor $\phi_i \in \boldsymbol{G}_j$ **do**
19: $\quad\quad \phi_i \leftarrow \phi^*$
20: $M' \leftarrow$ construct PFG from groupings of ACP

3.2 The ε-Advanced Colour Passing Algorithm

The ε-ACP algorithm consists of three phases and is described in Alg. 1. In the first phase, ε-ACP computes groups of factors that are pairwise ε-equivalent. For every factor ϕ_i in the input FG, ε-ACP checks whether it can be added to an existing group or if a new group has to be created. As it is possible for ϕ_i to be ε-equivalent to all factors of multiple existing groups (e.g., in Table 1, ϕ_2 could be grouped both with $\{\phi_1\}$ and $\{\phi_3\}$), ε-ACP computes all candidate groups $\boldsymbol{C}$ (Lines 3 to 6) and then adds ϕ_i to the group that minimises the sum of squared deviations between ϕ_i and all factors in

[3] Recall that we assume π from Def. 3 to be the identity function. In case π is not the identity function, we end up with $Err(\phi_i, \phi_j) = \sum_{r_1, \ldots, r_n} (\phi_i(r_1, \ldots, r_n) - \phi_j(r_{\pi(1)}, \ldots, r_{\pi(n)}))^2$.

the group (Lines 8 and 9). If ϕ_i cannot be added to an existing group, ε-ACP creates a new group for ϕ_i (Line 11). Then, in the second phase, ε-ACP assigns to every factor a colour based on the group it belongs to, that is, all factors within the same group receive the same colour (and factors in different groups receive different colours). Factors within the same group could potentially be grouped together in a lifted representation if their arguments are indistinguishable. To ensure the factors' arguments are indistinguishable, ε-ACP runs the ACP algorithm using the previously assigned colours (instead of ACP's original colour assignment). By running ACP with the assigned colours, ε-ACP ensures that in addition to the potential tables, the surrounding graph structure of the factors is taken into account, thereby enforcing that the arguments of factors within a group are indistinguishable (more details on this are given in Appendix B). Finally, in phase three, ε-ACP updates the potentials of every group of factors computed by ACP according to Eq. (5) to ensure that all factors in a group have identical potential tables (Lines 16 to 19). As potentials within a group are now strictly equal, the corresponding PFG is constructed from the groups as in the original ACP algorithm.[4] Commonly used lifted inference algorithms, such as lifted variable elimination, operate on PFGs and thus can directly be run on the output of ε-ACP.[5]

Example 6. *Take a look at the FG given in Fig. 1 and assume the potential tables of ϕ_1 and ϕ_2 are as given in Table 1a (i.e., $\varphi_1 = 0.75$, $\varphi'_1 = 0.8$, and so on). Further, let $\varepsilon = 0.1$ and assume we do not have any evidence, i.e., $\boldsymbol{O} = \emptyset$. As ϕ_1 and ϕ_2 are ε-equivalent, ε-ACP puts them into the same group and after the first phase, ε-ACP ends up with $\boldsymbol{G} = \{\{\phi_1, \phi_2\}\}$. Then, ACP is called with ϕ_1 and ϕ_2 having the same colour, and after passing the colours around, ϕ_1 and ϕ_2 remain in the same group because their surrounding graph structure is symmetric (and thus, their arguments are indistinguishable). After the third phase, the potential tables are updated by computing a row-wise arithmetic mean, that is, $\varphi_1 = \varphi'_1 = (0.75 + 0.8) \,/\, 2 = 0.775$, $\varphi_2 = \varphi'_2 = 0.315$, $\varphi_3 = \varphi'_3 = 0.49$, and $\varphi_4 = \varphi'_4 = 0.21$.*

The ε-ACP algorithm takes a fundamental step towards the practical applicability of lifted inference algorithms by generalising the ACP algorithm to account for inaccurate estimates of potentials, which are abundant in practice. In particular, it holds that ε-ACP is identical to ACP when setting ε to zero because ε-equivalence reduces to strict equivalence if $\varepsilon = 0$.

Corollary 2. *If $\varepsilon = 0$, ε-ACP returns the same PFG as ACP.*

So far, we have shown how ε-equivalent factors can be grouped and updated to enable lifted inference with a minimal approximation error. As we show later, the approximation error is often even negligible in practice. To get an initial idea about the extent of the approximation error, consider Ex. 6 and the query $P(SalA \mid Rev = \text{high})$. In the original FG, we obtain $P(SalA \mid Rev = \text{high}) \approx \langle 0.6098, 0.3902 \rangle$ and after running ε-ACP, we have $P(SalA \mid Rev = \text{high}) \approx \langle 0.6126, 0.3874 \rangle$. An essential question now is how much query results can change in general when using the approximate lifted representation instead of the initial exact FG for query answering. We answer this question next.

4 Bounding the Change in Query Results

We now bound the change in query results when modifying a given FG by grouping and updating the potentials of ε-equivalent factors according to Alg. 1. For the sake of our analysis, let M denote the input for Alg. 1 and M' the output of Alg. 1 such that M encodes the distribution P_M and M' encodes the distribution $P_{M'}$. In our analysis, we use the following distance measure between two distributions P_M and $P_{M'}$ introduced by Chan and Darwiche [2005]:

$$D(P_M, P_{M'}) = \ln \max_{\boldsymbol{r}} \frac{P_{M'}(\boldsymbol{r})}{P_M(\boldsymbol{r})} - \ln \min_{\boldsymbol{r}} \frac{P_{M'}(\boldsymbol{r})}{P_M(\boldsymbol{r})} \tag{6}$$

$$= \ln \max_{\boldsymbol{r}} \frac{\frac{1}{Z'}\psi'(\boldsymbol{r})}{\frac{1}{Z}\psi(\boldsymbol{r})} - \ln \min_{\boldsymbol{r}} \frac{\frac{1}{Z'}\psi'(\boldsymbol{r})}{\frac{1}{Z}\psi(\boldsymbol{r})} \tag{7}$$

$$= \ln \max_{\boldsymbol{r}} \frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})} - \ln \min_{\boldsymbol{r}} \frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})}, \tag{8}$$

where we define $0/0 := 1$ and $\infty/\infty := 1$. D satisfies important properties of a distance measure (positiveness, symmetry, and the triangle inequality) and a major advantage of D is that it allows us to bound the change in query results, which is not possible with other common distance measures such as the Kullback-Leibler divergence [Chan and Darwiche, 2005]. In particular, if it holds that $D(P_M, P_{M'}) = d$, the change in a query result is bounded by

$$e^{-d} \leq \frac{O_{M'}(r \mid \boldsymbol{e})}{O_M(r \mid \boldsymbol{e})} \leq e^d, \tag{9}$$

where $O_M(r \mid \boldsymbol{e}) = P_M(r \mid \boldsymbol{e}) \,/\, (1 - P_M(r \mid \boldsymbol{e}))$ defines the odds of r given $\boldsymbol{e}$. We can also write Eq. (9) in terms of probabilities instead of odds and obtain

$$\frac{pe^{-d}}{p(e^{-d} - 1) + 1} \leq P_{M'}(r \mid \boldsymbol{e}) \leq \frac{pe^d}{p(e^d - 1) + 1}, \tag{10}$$

where $p = P_M(r \mid \boldsymbol{e})$ is the initial probability of r given $\boldsymbol{e}$ in model M and $P_{M'}(r \mid \boldsymbol{e})$ is the probability of r given $\boldsymbol{e}$ in the modified model M' [Chan and Darwiche, 2005]. The bounds given in Eqs. (9) and (10) are sharp. To obtain a bound on the change in query results, we thus need to determine the value of $d = D(P_M, P_{M'})$ for a given choice of ε. In general, the normalisation constant Z changes when modifying the original model M. Rewriting Eq. (6) as Eq. (8), however, allows us to avoid dealing with the change from Z to Z' (a full derivation is given in Appendix A).

We next give a general bound on the distance $D(P_M, P_{M'})$ that applies to arbitrary FGs M where updates of factors resulting in an FG M' ensure that all factors in M' remain ε-equivalent to their original values after the update.

Theorem 3. *Let $M = (\boldsymbol{R} \cup \boldsymbol{\Phi}, \boldsymbol{E})$ be an FG and let M' be an FG obtained by updating arbitrary potentials of factors*

[4] For a detailed description of the PFG construction in Line 20 of Alg. 1, we refer the reader to Luttermann *et al.* [2024a].

[5] We remark that ε-equivalence can also be applied to exploit approximate symmetries *within* factors that map assignments of their arguments to identical potentials independent of the order of the assigned values (for more details, see Appendix D).

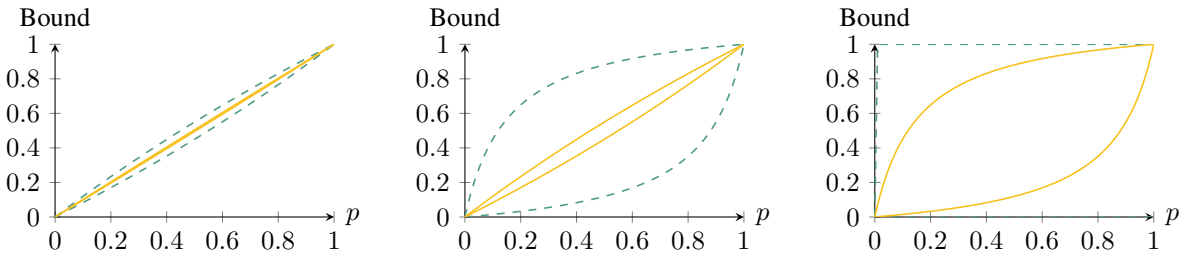


Figure 2: Plots of the bound given in Eq. (10) with $d = \ln(1+\varepsilon)^m - \ln(1-\varepsilon)^m$. Bounds are illustrated for (a) $m = 10$, (b) $m = 100$, and (c) $m = 1000$ where $\varepsilon = 0.01$ (dashed line) and $\varepsilon = 0.001$ (solid line), respectively. The x-axes depict the initial probability $p = P_M(r \mid \boldsymbol{e})$ and the y-axes reflect the bound on the change in the query result.

in M such that every updated potential remains ε-equivalent to its original value. Then, it holds that $D(P_M, P_{M'}) \leq \ln(1+\varepsilon)^m - \ln(1-\varepsilon)^m$, where P_M and $P_{M'}$ are the underlying full joint probability distributions encoded by M and M', respectively, and $m = |\boldsymbol{\Phi}|$.

Proof Sketch. By definition, every potential in M' differs from its original value in M by factor at most $(1 \pm \varepsilon)$. Adding a deviation by factor $(1 \pm \varepsilon)$ to every potential in M' and entering this into Eq. (8) yields the desired result. □

Corollary 4. *Given the bound from Thm. 3, Eq.* (9) *leads to*

$$\left(\frac{1-\varepsilon}{1+\varepsilon}\right)^m \leq \frac{O_{M'}(r \mid \boldsymbol{e})}{O_M(r \mid \boldsymbol{e})} \leq \left(\frac{1+\varepsilon}{1-\varepsilon}\right)^m. \tag{11}$$

The next lemma shows that updating the potentials within a group of pairwise ε-equivalent factors according to Eq. (5) satisfies the premise of Thm. 3 and hence, the bound given in Thm. 3 holds if M' is the output of Alg. 1 run on M.

Lemma 5. *Let $\boldsymbol{G} = \{\phi_1, \ldots, \phi_k\}$ denote a group of pairwise ε-equivalent factors and let $\phi^*(\boldsymbol{r}) = \frac{1}{k}\sum_{i=1}^{k} \phi_i(\boldsymbol{r})$ for all assignments $\boldsymbol{r}$. Then, $\boldsymbol{G}^* = \{\phi_1, \ldots, \phi_k, \phi^*\}$ is a group of pairwise ε-equivalent factors.*

Proof Sketch. As for the arithmetic mean $\phi^*(\boldsymbol{r})$ it holds that $\min_{\phi_j \in \boldsymbol{G}} \phi_j(\boldsymbol{r}) \leq \phi^*(\boldsymbol{r}) \leq \max_{\phi_j \in \boldsymbol{G}} \phi_j(\boldsymbol{r})$ and all $\phi_i, \phi_j \in \boldsymbol{G}$ are pairwise ε-equivalent, it follows that $\phi^*(\boldsymbol{r}) \in [\phi_i(\boldsymbol{r}) \cdot (1-\varepsilon), \phi_i(\boldsymbol{r}) \cdot (1+\varepsilon)]$ and $\phi_i(\boldsymbol{r}) \in [\phi^*(\boldsymbol{r}) \cdot (1-\varepsilon), \phi^*(\boldsymbol{r}) \cdot (1+\varepsilon)]$ for any assignment $\boldsymbol{r}$ and $\phi_i \in \boldsymbol{G}$. □

Lemma 5 implies that all updated potentials for every factor differ by factor at most $(1 \pm \varepsilon)$ from their original potential after running Alg. 1. To obtain a bound on the change in query results depending on the choice of ε, we enter the bound from Thm. 3 into Eq. (10). Figure 2 provides plots of the bound for different values of ε and $m = |\boldsymbol{\Phi}|$ to give a better idea on how the bound behaves. Observe that $\varepsilon = 0.01$ yields a strong bound for $m = 10$, however, from $m = 100$ onward, the bound becomes weak (in particular, for $m = 1000$, the change in query results is essentially unbounded when choosing $\varepsilon = 0.01$). When choosing $\varepsilon = 0.001$, the bound remains strong for $m = 100$, however, for $m = 1000$, the bound weakens as well. Fortunately, the bound from Thm. 3 is overly pessimistic for the output of Alg. 1, as we show in the following.

Lemma 6. *For two ε-equivalent factors ϕ_1 and ϕ_2, it holds that $\phi_1 \in [\phi_2 \cdot \frac{1}{1+\varepsilon}, \phi_2 \cdot (1+\varepsilon)]$ and $\phi_2 \in [\phi_1 \cdot \frac{1}{1+\varepsilon}, \phi_1 \cdot (1+\varepsilon)]$.*

Proof. Due to the symmetric definition of ε-equivalence, we get $\phi_{2-i} \leq \phi_{i+1} \cdot (1+\varepsilon)$ for $i \in \{0, 1\}$, resulting in $\phi_{2-i} \cdot \frac{1}{1+\varepsilon} \leq \phi_{i+1}$. Since $1 - \varepsilon \leq \frac{1}{1+\varepsilon}$ holds for any $\varepsilon > 0$, ϕ_{2-i} is contained in the strict subset $[\phi_{i+1} \cdot \frac{1}{1+\varepsilon}, \phi_{i+1} \cdot (1+\varepsilon)] \subsetneq [\phi_{i+1} \cdot (1-\varepsilon), \phi_{i+1} \cdot (1+\varepsilon)]$. □

Using Lemma 6 and the properties of the arithmetic mean, we obtain the following stronger bound on $D(P_M, P_{M'})$.

Theorem 7. *Let $M = (\boldsymbol{R} \cup \boldsymbol{\Phi}, \boldsymbol{E})$ be an FG and let M' be the output of Alg. 1 when run on M. With P_M and $P_{M'}$ being the underlying full joint probability distributions encoded by M and M', respectively, and $m = |\boldsymbol{\Phi}|$, it holds that*

$$D(P_M, P_{M'}) \leq \ln\left(\frac{1 + \frac{m-1}{m}\varepsilon}{\frac{1+\frac{1}{m}\varepsilon}{1+\varepsilon}}\right)^m \tag{12}$$

$$= \ln\left(\frac{\left(1 + \frac{m-1}{m}\varepsilon\right)\left(1+\varepsilon\right)}{1 + \frac{1}{m}\varepsilon}\right)^m \tag{13}$$

$$< \ln\left(1+\varepsilon\right)^{2m} \tag{14}$$

$$< \ln\left(\frac{1+\varepsilon}{1-\varepsilon}\right)^m. \tag{15}$$

Corollary 8. *Given the bound from Thm. 7, Eq.* (9) *leads to*

$$\left(\frac{\frac{1+\frac{1}{m}\varepsilon}{1+\varepsilon}}{1 + \frac{m-1}{m}\varepsilon}\right)^m \leq \frac{O_{M'}(r \mid \boldsymbol{e})}{O_M(r \mid \boldsymbol{e})} \leq \left(\frac{1 + \frac{m-1}{m}\varepsilon}{\frac{1+\frac{1}{m}\varepsilon}{1+\varepsilon}}\right)^m. \tag{16}$$

We give a proof of Thm. 7 in Appendix A. The plot of the bound from Thm. 7 looks similar to the plot of Thm. 3 (see Fig. 2) and is optimal (i.e., it is the best bound we can find).

Theorem 9. *The bound given in Thm. 7 is optimal.*

Proof Sketch. We construct an FG hitting the boundary from Thm. 7. For the construction, see Table 3 in Appendix A. □

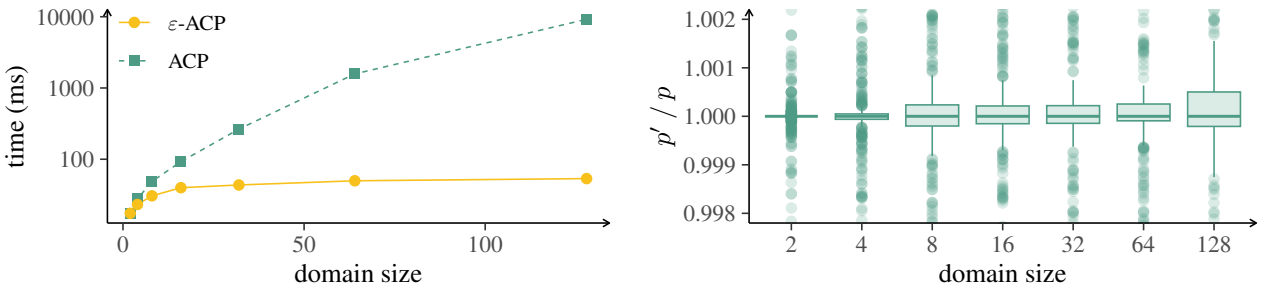


Figure 3: Average query times of lifted variable elimination on the output of ACP and ε-ACP for every choice of k (left), and a boxplot showing the distribution of the quotient $p' \,/\, p$, where $p' = P_{M'}(r \mid \boldsymbol{e})$ and $p = P_M(r \mid \boldsymbol{e})$, for each choice of k (right).

Algorithm	Avg. Run Time	Avg. $p' \,/\, p$
ACP	183 ms ($\pm$ 21)	1.0 ($\pm$ 0.0)
ε-ACP	105 ms ($\pm$ 9)	1.0001 ($\pm$ 0.01)

Table 2: Average query times and quotients of query results on parts of the MIMIC-IV dataset [Johnson *et al.*, 2023].

Fortunately, in practice, the change in query results is often close to zero (and thus well below the theoretical bound), as we will show in our experiments. The reason for this is that the worst-case scenario is an extreme case and slightly deviating from it significantly improves the bounds. For instance, if there are more factors in a group than rows in their potential tables, the worst-case scenario can no longer occur, resulting in notably smaller values for the distance measure D. More details are given in the proof of Thm. 7 in Appendix A.

5 Experiments

We test the practicality of the ε-ACP algorithm in a series of experiments. ε-ACP is not only required to make ACP applicable in practice but also allows for more compression (and thus faster inference) if we are willing to trade the exactness of query results for additional speedup. We thus report the run time gain and the resulting approximation error to get a better understanding of the trade-off between the exactness and the compactness of the lifted representation obtained by ε-ACP. For our experiments, we generate a variety of FGs with different graph structures and graph sizes (i.e., numbers of randvars and factors). More specifically, we generate FGs containing between $2k+1$ and $2k + k \cdot \lfloor \log_2(k) \rfloor + 1$ Boolean randvars as well as between $2k$ and $k + k \cdot \lfloor \log_2(k) \rfloor + 1$ factors, where $k \in \{2, 4, 8, 16, 32, 64, 128\}$ is the *domain size*. The domain size k controls the number of objects in the models and thus the size of the FGs. We provide all data set generators along with our source code in the supplementary material.

In every FG, we modify a proportion of $x \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$ of the factors such that their potential tables differ by at most factor $(1 \pm \varepsilon)$ from their original potential tables, where $\varepsilon \in \{0.001, 0.01, 0.1\}$. For each setting, we pose multiple queries to each FG. We report the average run time of lifted variable elimination (the state-of-the-art lifted inference algorithm) on the output of ACP and ε-ACP, respectively, over all settings for each choice of k in the left plot of Fig. 3 and show the distribution of $P_{M'}(r \mid \boldsymbol{e}) \,/\, P_M(r \mid \boldsymbol{e})$ over all queries for each choice of k in the right plot of Fig. 3.

Taking a look at the left plot in Fig. 3, it becomes evident that ε-ACP yields a speedup of up to factor 100 compared to ACP. The question now is at what cost ε-ACP achieves this speedup. The right plot in Fig. 3 demonstrates that the price ε-ACP pays for the speedup is close to zero: Most of the quotients are nearly equal to one (i.e., most query results hardly differ from their original value). As expected, the larger the domain size (and hence the size of the FG), the larger quotients become. However, even the outliers (denoted by the dots outside of the boxes) only deviate at the third decimal place from the optimal value one. The experimental results highlight the practical effectiveness of ε-ACP as the approximation error is significantly smaller in practice than suggested by the theoretical bounds. To give a better overview on how the approximation error behaves for specific choices of x and ε, we provide additional results for individual choices of x and ε in Appendix E.

In addition to the generated FGs, we learn an FG from the MIMIC-IV dataset [Johnson *et al.*, 2023] and apply ε-ACP with $\varepsilon = 0.1$ to it. MIMIC-IV contains real-world medical data and we consider a subset of 4000 patients and their treatments from it. The learned FG contains 344 randvars and factors, respectively, and we query each randvar in it. We report average run times and average quotients over all queries in Table 2. While the speedup of ε-ACP is smaller than in Fig. 3, the error quotients are also reduced by an order of magnitude, showing that the approximation error is again close to zero.

6 Conclusion

Potentials learnt from data often slightly differ even for indistinguishable objects. Therefore, we solve the problem of constructing a lifted representation from a given propositional representation taking inaccurate estimates of potentials into account, while previous approaches require exact matches. We present the ε-ACP algorithm, which allows for a small deviation of potentials depending on a hyperparameter ε. By not relying on strictly identical potentials, ε-ACP makes a fundamental step towards the practicality of obtaining a compact representation for lifted inference. We further show that the approximation error of ε-ACP is strictly bounded and demonstrate that it is even close to zero in practice.

Acknowledgements

This work was partially funded by the Ministry of Culture and Science of the German State of North Rhine-Westphalia. The research of Malte Luttermann was funded by the BMBF project AnoMed 16KISA057.

References

Babak Ahmadi, Kristian Kersting, Martin Mladenov, and Sriraam Natarajan. Exploiting Symmetries for Scaling Loopy Belief Propagation and Relational Training. *Machine Learning*, 92:91–132, 2013.

Tanya Braun and Ralf Möller. Parameterised Queries and Lifted Query Answering. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-2018)*, pages 4980–4986. IJCAI Organization, 2018.

Hei Chan and Adnan Darwiche. A Distance Measure for Bounding Probabilistic Belief Change. *International Journal of Approximate Reasoning*, 38:149–174, 2005.

Rodrigo De Salvo Braz, Eyal Amir, and Dan Roth. Lifted First-Order Probabilistic Inference. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-2005)*, pages 1319–1325. Morgan Kaufmann Publishers Inc., 2005.

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

Brendan J. Frey, Frank R. Kschischang, Hans-Andrea Loeliger, and Niclas Wiberg. Factor Graphs and Algorithms. In *Proceedings of the Thirty-Fifth Annual Allerton Conference on Communication, Control, and Computing*, pages 666–680. Allerton House, 1997.

Alistair E. W. Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J. Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, Li wei H. Lehman, Leo A. Celi, and Roger G. Mark. MIMIC-IV, A Freely Accessible Electronic Health Record Dataset. *Scientific Data*, 10:1, 2023.

Kristian Kersting, Babak Ahmadi, and Sriraam Natarajan. Counting Belief Propagation. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI-2009)*, pages 277–284. AUAI Press, 2009.

Jacek Kisyński and David Poole. Constraint Processing in Lifted Probabilistic Inference. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI-2009)*, pages 293–302. AUAI Press, 2009.

Frank R. Kschischang, Brendan J. Frey, and Hans-Andrea Loeliger. Factor Graphs and the Sum-Product Algorithm. *IEEE Transactions on Information Theory*, 47:498–519, 2001.

Malte Luttermann, Tanya Braun, Ralf Möller, and Marcel Gehrke. Colour Passing Revisited: Lifted Model Construction with Commutative Factors. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-2024)*, pages 20500–20507. AAAI Press, 2024.

Malte Luttermann, Tanya Braun, Ralf Möller, and Marcel Gehrke. Estimating Causal Effects in Partially Directed Parametric Causal Factor Graphs. In *Proceedings of the Sixteenth International Conference on Scalable Uncertainty Management (SUM-2024)*, pages 265–280. Springer, 2024.

Malte Luttermann, Mattis Hartwig, Tanya Braun, Ralf Möller, and Marcel Gehrke. Lifted Causal Inference in Relational Domains. In *Proceedings of the Third Conference on Causal Learning and Reasoning (CLeaR-2024)*, pages 827–842. PMLR, 2024.

Malte Luttermann, Johann Machemer, and Marcel Gehrke. Efficient Detection of Commutative Factors in Factor Graphs. In *Proceedings of the Twelfth International Conference on Probabilistic Graphical Models (PGM-2024)*. PMLR, 2024.

Malte Luttermann, Johann Machemer, and Marcel Gehrke. Efficient Detection of Exchangeable Factors in Factor Graphs. In *Proceedings of the Thirty-Seventh International Florida Artificial Intelligence Research Society Conference (FLAIRS-2024)*. Florida Online Journals, 2024.

Malte Luttermann, Ralf Möller, and Marcel Gehrke. Lifted Model Construction without Normalisation: A Vectorised Approach to Exploit Symmetries in Factor Graphs. In *Proceedings of the Third Learning on Graphs Conference (LoG-2024)*. PMLR, 2024.

Brian Milch, Luke S. Zettlemoyer, Kristian Kersting, Michael Haimes, and Leslie Pack Kaelbling. Lifted Probabilistic Inference with Counting Formulas. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI-2008)*, pages 1062–1068. AAAI Press, 2008.

Mathias Niepert and Guy Van den Broeck. Tractability through Exchangeability: A New Perspective on Efficient Probabilistic Inference. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI-2014)*, pages 2467–2475. AAAI Press, 2014.

David Poole. First-Order Probabilistic Inference. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 985–991. Morgan Kaufmann Publishers Inc., 2003.

Nima Taghipour, Daan Fierens, Jesse Davis, and Hendrik Blockeel. Lifted Variable Elimination: Decoupling the Operators from the Constraint Language. *Journal of Artificial Intelligence Research*, 47:393–439, 2013.

Boris Weisfeiler and Andrei A. Leman. The Reduction of a Graph to Canonical Form and the Algebra which Appears Therein. *NTI, Series*, 2:12–16, 1968. English translation by Grigory Ryabov available at https://www.iti.zcu.cz/wl2018/pdf/wl_paper_translation.pdf.

A Missing Proofs

Theorem 1. *Let $\varphi_1, \dots, \varphi_k \in \mathbb{R}^+$. It holds that the arithmetic mean $\bar{\varphi} = \frac{1}{k}\sum_{i=1}^{k}\varphi_i$ is the optimal choice for $\varphi^* = \arg\min_{\hat{\varphi}} \sum_{i=1}^{k}(\varphi_i - \hat{\varphi})^2$.*

Proof. Let $\hat{\varphi}$ denote an arbitrary estimate for φ^* and let $\bar{\varphi} = \frac{1}{k}\sum_{i=1}^{k}\varphi_i$ denote the arithmetic mean of $\varphi_1, \dots, \varphi_k$. We now show that choosing $\hat{\varphi} = \bar{\varphi}$ minimises the expression $\sum_{i=1}^{k}(\varphi_i - \hat{\varphi})^2$. Since $\bar{\varphi} - \bar{\varphi} = 0$, we can add it without changing the result and then rewrite the expression:

$$\sum_{i=1}^{k}\big(\varphi_i - \hat{\varphi}\big)^2 \tag{17}$$

$$= \sum_{i=1}^{k}\Big(\varphi_i - \hat{\varphi} + (\bar{\varphi} - \bar{\varphi})\Big)^2 \tag{18}$$

$$= \sum_{i=1}^{k}\Big((\varphi_i - \bar{\varphi}) + (\bar{\varphi} - \hat{\varphi})\Big)^2 \tag{19}$$

$$= \sum_{i=1}^{k}\Big(\big(\varphi_i - \bar{\varphi}\big)^2 + 2\big(\varphi_i - \bar{\varphi}\big)\big(\bar{\varphi} - \hat{\varphi}\big) + \big(\bar{\varphi} - \hat{\varphi}\big)^2\Big) \tag{20}$$

$$= \sum_{i=1}^{k}\big(\varphi_i - \bar{\varphi}\big)^2 + \sum_{i=1}^{k}2\big(\varphi_i - \bar{\varphi}\big)\big(\bar{\varphi} - \hat{\varphi}\big) + \sum_{i=1}^{k}\big(\bar{\varphi} - \hat{\varphi}\big)^2 \tag{21}$$

$$= \sum_{i=1}^{k}\big(\varphi_i - \bar{\varphi}\big)^2 + 2\big(\bar{\varphi} - \hat{\varphi}\big)\sum_{i=1}^{k}\big(\varphi_i - \bar{\varphi}\big) + k\big(\bar{\varphi} - \hat{\varphi}\big)^2. \tag{22}$$

Due to $\bar{\varphi} = \frac{1}{k}\sum_{i=1}^{k}\varphi_i$ being the arithmetic mean of $\varphi_1, \dots, \varphi_k$, it holds that $\sum_{i=1}^{k}(\varphi_i - \bar{\varphi}) = 0$:

$$\sum_{i=1}^{k}\big(\varphi_i - \bar{\varphi}\big) = \sum_{i=1}^{k}\varphi_i - k\cdot\bar{\varphi} \tag{23}$$

$$= \sum_{i=1}^{k}\varphi_i - k\cdot\frac{1}{k}\sum_{i=1}^{k}\varphi_i \tag{24}$$

$$= 0. \tag{25}$$

By entering $\sum_{i=1}^{k}(\varphi_i - \bar{\varphi}) = 0$ into Eq. (22), we thus obtain

$$\sum_{i=1}^{k}\big(\varphi_i - \bar{\varphi}\big)^2 + 2\big(\bar{\varphi} - \hat{\varphi}\big)\sum_{i=1}^{k}\big(\varphi_i - \bar{\varphi}\big) + k\big(\bar{\varphi} - \hat{\varphi}\big)^2 \tag{26}$$

$$= \sum_{i=1}^{k}\big(\varphi_i - \bar{\varphi}\big)^2 + k\big(\bar{\varphi} - \hat{\varphi}\big)^2. \tag{27}$$

Now, if we set $\hat{\varphi} = \bar{\varphi}$, it holds that $\bar{\varphi} - \hat{\varphi} = 0$ and hence, Eq. (27) simplifies to

$$\sum_{i=1}^{k}\big(\varphi_i - \bar{\varphi}\big)^2 + k\big(\bar{\varphi} - \hat{\varphi}\big)^2 \tag{28}$$

$$= \sum_{i=1}^{k}\big(\varphi_i - \bar{\varphi}\big)^2. \tag{29}$$

Choosing $\hat{\varphi}$ different from $\bar{\varphi}$ thus increases the value of the expression by $k(\bar{\varphi} - \hat{\varphi})^2$, which completes the proof.[6] □

Corollary 2. *If $\varepsilon = 0$, ε-ACP returns the same PFG as ACP.*

Proof. Let $\varepsilon = 0$. Recall that two potentials $\varphi \in \mathbb{R}^+$ and $\varphi' \in \mathbb{R}^+$ are ε-equivalent if $\varphi \in [\varphi' \cdot (1-\varepsilon), \varphi' \cdot (1+\varepsilon)]$ and $\varphi' \in [\varphi \cdot (1-\varepsilon), \varphi \cdot (1+\varepsilon)]$. Now, as $\varepsilon = 0$, φ and φ' are ε-equivalent if $\varphi = \varphi'$. In consequence, when running ε-ACP (Alg. 1) on an arbitrary input FG M, in the first phase all groups of ε-equivalent factors contain only factors whose potentials are strictly equivalent. Therefore, in phase two of ε-ACP, only factors with equivalent potentials receive the same colour and hence, the colour assignment is identical to the colour assignment of ACP (Alg. 2). Then, ε-ACP calls ACP and as the colour assignment of ε-ACP to factors is identical to the colour assignment of ACP, phase one of ε-ACP has no impact on the output of ACP. It remains to be shown that the update of potentials in phase three of ε-ACP does not alter the output of ACP. As all factors in any group have strictly equivalent potentials already and the arithmetic mean of a set of equal numbers $\varphi, \dots, \varphi$ is φ itself, the update in phase three does not alter the output of ACP. Hence, ε-ACP ends up with the same groups as ACP and therefore outputs the same PFG as ACP, which completes the proof. □

Full derivation of writing Eq. (6) as Eq. (8):

$$D(P_M, P_{M'}) \tag{30}$$

$$= \ln\max_{\boldsymbol{r}}\frac{P_{M'}(\boldsymbol{r})}{P_M(\boldsymbol{r})} - \ln\min_{\boldsymbol{r}}\frac{P_{M'}(\boldsymbol{r})}{P_M(\boldsymbol{r})} \tag{31}$$

$$= \ln\max_{\boldsymbol{r}}\frac{\frac{1}{Z'}\psi'(\boldsymbol{r})}{\frac{1}{Z}\psi(\boldsymbol{r})} - \ln\min_{\boldsymbol{r}}\frac{\frac{1}{Z'}\psi'(\boldsymbol{r})}{\frac{1}{Z}\psi(\boldsymbol{r})} \tag{32}$$

$$= \ln\left(\frac{\frac{1}{Z'}}{\frac{1}{Z}}\max_{\boldsymbol{r}}\frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})}\right) - \ln\left(\frac{\frac{1}{Z'}}{\frac{1}{Z}}\min_{\boldsymbol{r}}\frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})}\right) \tag{33}$$

$$= \ln\frac{\frac{1}{Z'}}{\frac{1}{Z}} + \ln\max_{\boldsymbol{r}}\frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})} - \ln\frac{\frac{1}{Z'}}{\frac{1}{Z}} - \ln\min_{\boldsymbol{r}}\frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})} \tag{34}$$

$$= \ln\max_{\boldsymbol{r}}\frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})} - \ln\min_{\boldsymbol{r}}\frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})} \tag{35}$$

Full derivation of writing Eq. (9) as Eq. (10). Remember that $O_M(r \mid \boldsymbol{e}) = P_M(r \mid \boldsymbol{e}) \,/\, (1 - P_M(r \mid \boldsymbol{e}))$ defines the odds of r given $\boldsymbol{e}$. Entering the definition of the odds into Eq. (9) then results in

$$e^{-d} \leq \frac{P_{M'}(r \mid \boldsymbol{e}) \,/\, (1 - P_{M'}(r \mid \boldsymbol{e}))}{P_M(r \mid \boldsymbol{e}) \,/\, (1 - P_M(r \mid \boldsymbol{e}))} \leq e^{d}, \tag{36}$$

which can be rewritten as

$$e^{-d} \leq \frac{P_{M'}(r \mid \boldsymbol{e})}{1 - P_{M'}(r \mid \boldsymbol{e})} \cdot \frac{1 - P_M(r \mid \boldsymbol{e})}{P_M(r \mid \boldsymbol{e})} \leq e^{d}. \tag{37}$$

[6] Theorem 1 is a well-known property of the arithmetic mean and there are different ways to prove this property. The proof given here is taken from http://faculty.washington.edu/swithers/seestats/SeeingStatisticsFiles/seeing/center/meanproof/meanProof.html.

With $p = P_M(r \mid \boldsymbol{e})$ and $p' = P_{M'}(r \mid \boldsymbol{e})$, we obtain

$$e^{-d} \leq \frac{p'(1-p)}{p(1-p')} \leq e^d. \tag{38}$$

Multiplying by $p(1-p')$ (which is always positive) yields

$$e^{-d}p(1-p') \leq p'(1-p) \leq e^d p(1-p'). \tag{39}$$

Expanding the terms gives us

$$e^{-d}p - e^{-d}pp' \leq p' - p'p \leq e^d p - e^d pp'. \tag{40}$$

By rearranging the terms, we end up with

$$e^{-d}p \leq p' - p'p + e^{-d}pp' \tag{41}$$

$$= p'(1 - p + e^{-d}p), \text{ and} \tag{42}$$

$$e^d p \geq p' - p'p + e^d pp' \tag{43}$$

$$= p'(1 - p + e^d p). \tag{44}$$

Dividing by $1 - p + e^{-d}p$ and $1 - p + e^d p$ (both terms are always positive), respectively, results in

$$\frac{e^{-d}p}{1 - p + e^{-d}p} \leq p', \text{ and} \tag{45}$$

$$\frac{e^d p}{1 - p + e^d p} \geq p'. \tag{46}$$

By rearranging the terms once more, we obtain

$$\frac{pe^{-d}}{p(e^{-d}-1)+1} \leq p' \leq \frac{pe^d}{p(e^d-1)+1}. \tag{47}$$

Theorem 3. *Let $M = (\boldsymbol{R} \cup \boldsymbol{\Phi}, \boldsymbol{E})$ be an FG and let M' be an FG obtained by updating arbitrary potentials of factors in M such that every updated potential remains ε-equivalent to its original value. Then, it holds that $D(P_M, P_{M'}) \leq \ln(1+\varepsilon)^m - \ln(1-\varepsilon)^m$, where P_M and $P_{M'}$ are the underlying full joint probability distributions encoded by M and M', respectively, and $m = |\boldsymbol{\Phi}|$.*

Proof. Let $\boldsymbol{\Phi} = \{\phi_1, \ldots, \phi_m\}$ denote the set of factors in M. By definition, it holds that every updated potential φ' in M' differs by factor at most $(1 \pm \varepsilon)$ from its original potential φ in M, independent of the distribution of the groups (that is, it is irrelevant whether all m factors are in the same group or whether groups are distributed otherwise). Therefore, as $\psi(\boldsymbol{r}) = \prod_{j=1}^m \phi_j(\boldsymbol{r}_j)$ for any assignment $\boldsymbol{r}$, we obtain

$$\psi'(\boldsymbol{r}) \geq \prod_{j=1}^m \phi_j(\boldsymbol{r}_j) \cdot (1-\varepsilon), \text{ and} \tag{48}$$

$$\psi'(\boldsymbol{r}) \leq \prod_{j=1}^m \phi_j(\boldsymbol{r}_j) \cdot (1+\varepsilon). \tag{49}$$

In consequence, we get the following bounds:

$$\min_{\boldsymbol{r}} \frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})} \geq \frac{\prod_{j=1}^m \big(\phi_j(\boldsymbol{r}_j) \cdot (1-\varepsilon)\big)}{\prod_{j=1}^m \phi_j(\boldsymbol{r}_j)}, \text{ and} \tag{50}$$

$$\max_{\boldsymbol{r}} \frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})} \leq \frac{\prod_{j=1}^m \big(\phi_j(\boldsymbol{r}_j) \cdot (1+\varepsilon)\big)}{\prod_{j=1}^m \phi_j(\boldsymbol{r}_j)}, \tag{51}$$

where $\boldsymbol{r}_j$ is a projection of *any* assignment $\boldsymbol{r}$ to the argument list of ϕ_j. Entering these bounds into Eq. (8) then yields

$$D(P_M, P_{M'}) \tag{52}$$

$$= \ln \max_{\boldsymbol{r}} \frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})} - \ln \min_{\boldsymbol{r}} \frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})} \tag{53}$$

$$\leq \ln \frac{\prod_{j=1}^m \big(\phi_j(\boldsymbol{r}_j) \cdot (1+\varepsilon)\big)}{\prod_{j=1}^m \phi_j(\boldsymbol{r}_j)} - \ln \frac{\prod_{j=1}^m \big(\phi_j(\boldsymbol{r}_j) \cdot (1-\varepsilon)\big)}{\prod_{j=1}^m \phi_j(\boldsymbol{r}_j)} \tag{54}$$

$$= \ln \frac{(1+\varepsilon)^m \prod_{j=1}^m \phi_j(\boldsymbol{r}_j)}{\prod_{j=1}^m \phi_j(\boldsymbol{r}_j)} - \ln \frac{(1-\varepsilon)^m \prod_{j=1}^m \phi_j(\boldsymbol{r}_j)}{\prod_{j=1}^m \phi_j(\boldsymbol{r}_j)} \tag{55}$$

$$= \ln(1+\varepsilon)^m - \ln(1-\varepsilon)^m. \tag{56}$$

□

Lemma 5. *Let $\boldsymbol{G} = \{\phi_1, \ldots, \phi_k\}$ denote a group of pairwise ε-equivalent factors and let $\phi^*(\boldsymbol{r}) = \frac{1}{k}\sum_{i=1}^k \phi_i(\boldsymbol{r})$ for all assignments $\boldsymbol{r}$. Then, $\boldsymbol{G}^* = \{\phi_1, \ldots, \phi_k, \phi^*\}$ is a group of pairwise ε-equivalent factors.*

Proof. We show the claim in two directions by proving that $\phi^*(\boldsymbol{r}) \in [\phi_i(\boldsymbol{r}) \cdot (1-\varepsilon), \phi_i(\boldsymbol{r}) \cdot (1+\varepsilon)]$ and $\phi_i(\boldsymbol{r}) \in [\phi^*(\boldsymbol{r}) \cdot (1-\varepsilon), \phi^*(\boldsymbol{r}) \cdot (1+\varepsilon)]$ for any assignment $\boldsymbol{r}$ and $\phi_i \in \boldsymbol{G}$.

For the first direction, let $\boldsymbol{r}$ be an arbitrary assignment and let $\phi_i \in \boldsymbol{G}$. As all factors in $\boldsymbol{G}$ are pairwise ε-equivalent, it holds that $\phi_i(\boldsymbol{r}) \cdot (1-\varepsilon) \leq \min_{\phi_j \in \boldsymbol{G}} \phi_j(\boldsymbol{r})$ and $\max_{\phi_j \in \boldsymbol{G}} \phi_j(\boldsymbol{r}) \leq \phi_i(\boldsymbol{r}) \cdot (1+\varepsilon)$. Further, as $\phi^*(\boldsymbol{r})$ is the arithmetic mean over all $\phi_j(\boldsymbol{r}) \in \boldsymbol{G}$, it holds that $\min_{\phi_j \in \boldsymbol{G}} \phi_j(\boldsymbol{r}) \leq \phi^*(\boldsymbol{r}) \leq \max_{\phi_j \in \boldsymbol{G}} \phi_j(\boldsymbol{r})$ and thus $\phi^*(\boldsymbol{r}) \in [\phi_i(\boldsymbol{r}) \cdot (1-\varepsilon), \phi_i(\boldsymbol{r}) \cdot (1+\varepsilon)]$.

For the second direction, it holds that for any assignment $\boldsymbol{r}$, every $\phi_i \in \boldsymbol{G}$ is contained in the interval $[\phi_j(\boldsymbol{r}) \cdot (1-\varepsilon), \phi_j(\boldsymbol{r}) \cdot (1+\varepsilon)]$ for any $j \in \{1, \ldots, k\}$, and thus also in the smallest possible composite interval $[\max_{\phi_j \in \boldsymbol{G}} \phi_j(\boldsymbol{r}) \cdot (1-\varepsilon), \min_{\phi_j \in \boldsymbol{G}} \phi_j(\boldsymbol{r}) \cdot (1+\varepsilon)]$. With the same argument as before, namely that for the arithmetic mean $\phi^*(\boldsymbol{r})$ we have $\min_{\phi_j \in \boldsymbol{G}} \phi_j(\boldsymbol{r}) \leq \phi^*(\boldsymbol{r}) \leq \max_{\phi_j \in \boldsymbol{G}} \phi_j(\boldsymbol{r})$, the composite interval $[\max_{\phi_j \in \boldsymbol{G}} \phi_j(\boldsymbol{r}) \cdot (1-\varepsilon), \min_{\phi_j \in \boldsymbol{G}} \phi_j(\boldsymbol{r}) \cdot (1+\varepsilon)]$ is a subset of the interval $[\phi^*(\boldsymbol{r}) \cdot (1-\varepsilon), \phi^*(\boldsymbol{r}) \cdot (1+\varepsilon)]$ and hence, $\phi_i(\boldsymbol{r}) \in [\phi^*(\boldsymbol{r}) \cdot (1-\varepsilon), \phi^*(\boldsymbol{r}) \cdot (1+\varepsilon)]$. □

Theorem 7. *Let $M = (\boldsymbol{R} \cup \boldsymbol{\Phi}, \boldsymbol{E})$ be an FG and let M' be the output of Alg. 1 when run on M. With P_M and $P_{M'}$ being the underlying full joint probability distributions encoded by*

M and M', respectively, and $m = |\mathbf{\Phi}|$, it holds that

$$D(P_M, P_{M'}) \leq \ln\left(\frac{1+\frac{m-1}{m}\varepsilon}{\frac{1+\frac{1}{m}\varepsilon}{1+\varepsilon}}\right)^m \tag{12}$$

$$= \ln\left(\frac{\left(1+\frac{m-1}{m}\varepsilon\right)\left(1+\varepsilon\right)}{1+\frac{1}{m}\varepsilon}\right)^m \tag{13}$$

$$< \ln\left(1+\varepsilon\right)^{2m} \tag{14}$$

$$< \ln\left(\frac{1+\varepsilon}{1-\varepsilon}\right)^m. \tag{15}$$

Proof. Let $\mathbf{\Phi} = \{\phi_1, \ldots, \phi_m\}$ denote the set of factors in M, representing the distribution of M via $P_M(\boldsymbol{r}) = \frac{1}{Z}\psi(\boldsymbol{r}) = \frac{1}{Z}\prod_{j=1}^m \phi_j(\boldsymbol{r}_j)$ for an assignment $\boldsymbol{r} = (r_1, \ldots, r_n) \in \times_{i=1}^n \text{range}(R_i)$ to the randvars in $\boldsymbol{R}$. On the level of potentials, this means that for a specific assignment $\boldsymbol{r}$, there exist j_i, $i \in \{1, \ldots, m\}$, such that $\psi(\boldsymbol{r}) = \prod_{i=1}^m \varphi_{j_i,i}$, where $\varphi_{j,i}$ represents the potential in the j^{th} row in the potential table of factor ϕ_i. For $\phi^*(\boldsymbol{r}) = \frac{1}{m}\sum_{i=1}^m \phi_i(\boldsymbol{r})$, its potentials $\varphi_j^* = \frac{1}{m}\sum_{i=1}^m \varphi_{j,i}$ are given by the row-wise arithmetic mean over all factors and thus are independent of i.

According to Lemma 5, ϕ^* is pairwise ε-equivalent to all ϕ_i of a group of pairwise ε-equivalent factors. We first prove the claim for the case where all factors $\{\phi_1, \ldots, \phi_m\}$ belong to the same group of pairwise ε-equivalence factors and afterwards generalise the proof to arbitrary distributions of groups.

By ordering the potentials of every row j, we adopt, without loss of generality, the notation

$$\varphi_{j,1} \leq \ldots \leq \varphi_{j,k_j} \leq \varphi_j^* \leq \varphi_{j,k_j+1} \leq \ldots \leq \varphi_{j,m} \tag{57}$$

with $k_j \in \{1, \ldots, m-1\}$. Since $\varphi_{j,i} \geq \varphi_j^*$ holds already for $i = k_j+1, \ldots, m$, we want to determine a minimal $\alpha_2(j) \in \mathbb{R}^+_{\geq 1}$ such that also $\varphi_{j,i} \cdot \alpha_2 \geq \varphi_j^*$ holds for all $i = 1, \ldots, m$. This means that $\varphi_j^* \leq \min_{i=1,\ldots,m} \varphi_{j,i} \cdot \alpha_2(j) = \varphi_{j,1} \cdot \alpha_2(j)$ has to hold. The minimal possible constant to fulfil this equation is $\alpha_2(j) := \frac{\varphi_j^*}{\varphi_{j,1}}$. In order to get the value of this constant, we have to assume the worst case scenario for the distribution of the $\varphi_{j,i}$'s. Therefore, we are looking for a small $\varphi_{j,1}$ and the largest φ_j^*, which is $\varphi_j^* = \frac{1}{m}\sum_{i=1}^m \varphi_{j,i}$ and $\varphi_{j,1}(1+\varepsilon) \geq \max_{i=1,\ldots,m} \varphi_{j,i} = \varphi_{j,m}$ according to the definition of ε-equivalence. Under these conditions, the maximal φ_j^* is possible by the choice of $\varphi_{j,i} := \varphi_j^*$ for $i = 2, \ldots, k_j$ and $\varphi_{j,i} := (1+\varepsilon)\varphi_{j,1}$ for $i = k_j+1, \ldots, m$. This results in the following mean

$$\varphi_j^* = \frac{1}{m}\left(\varphi_{j,1} + \sum_{i=2}^{k_j} \varphi_{j,i} + \sum_{i=k_j+1}^{m} \varphi_{j,1}(1+\varepsilon)\right) \tag{58}$$

$$= \frac{1}{m}\Big(\varphi_{j,1} + (k_j-1)\varphi_j^* + (1+\varepsilon)(m-k_j)\varphi_{j,1}\Big), \tag{59}$$

which is equivalent to

$$\varphi_j^*\left(1 - \frac{k_j-1}{m}\right) = \frac{1}{m}\varphi_{j,1}\big(1 + (m-k_j)(1+\varepsilon)\big) \tag{60}$$

$$\Leftrightarrow \varphi_j^* = \frac{1}{m-k_j+1}\varphi_{j,1}\big(1 + (m-k_j)(1+\varepsilon)\big) \tag{61}$$

$$= \frac{m-k_j+1+(m-k_j)\varepsilon}{m-k_j+1}\varphi_{j,1} \tag{62}$$

$$= \left(1 + \frac{m-k_j}{m-k_j+1}\varepsilon\right)\varphi_{j,1}. \tag{63}$$

This results in $\alpha_2(j) = \left(1 + \frac{m-k_j}{m-k_j+1}\varepsilon\right) > 1$ for a $k_j \in \{1, \ldots, m-1\}$. Within this condition, $k_j = 1$ leads to the largest ε-amount for $\alpha_2 = \left(1 + \frac{m-1}{m}\varepsilon\right)$ in the worst case and therefore for any assignment independently of j.

The estimation of a second constant $\alpha_1(j)$ as a lower constant works similar. Since $\varphi_{j,i} \leq \phi_j^*$ for $i = 1, \ldots, k_j$ holds already, we determine a maximal $\alpha_1(j) \in \mathbb{R}^+_{\leq 1}$ such that $\varphi_{j,i} \leq \varphi_j^*$ holds for all $i = 1, \ldots, m$. This means that $\varphi_j^* \geq \max_{i=1,\ldots,m} \varphi_{j,i} \cdot \alpha_1(j) = \varphi_{j,m} \cdot \alpha_1(j)$. The maximal possible constant is $\alpha_1(j) := \frac{\varphi_j^*}{\varphi_{j,m}}$. For the worst case scenario for the distribution of the $\varphi_{j,i}$'s, we get a large $\varphi_{j,m}$ and a low φ_j^* and according to Lemma 6 this means $\varphi_{j,m} \cdot \frac{1}{1+\varepsilon} \leq \min_{i=1,\ldots,m} \varphi_{j,i}$. A minimal φ_j^* can be reached by the choice of $\varphi_{j,i} := \varphi_j^*$ for $i = k_j+1, \ldots, m$ and $\varphi_{j,i} := \varphi_{j,m}\frac{1}{1+\varepsilon}$ for $i = 1, \ldots, k_j$. This results in

$$\varphi_j^* = \frac{1}{m}\left(\sum_{i=1}^{k_j} \varphi_{j,m}\frac{1}{1+\varepsilon} + \sum_{i=k_j+1}^{m-1} \varphi_j^* + \varphi_{j,m}\right) \tag{64}$$

$$= \frac{1}{m}\left(\frac{k_j}{1+\varepsilon}\varphi_{j,m} + (m-k_j-1)\varphi_j^* + \varphi_{j,m}\right), \tag{65}$$

which is equivalent to

$$\varphi_j^*\left(1 - \frac{m-k_j-1}{m}\right) = \frac{1}{m}\left(\frac{k_j}{1+\varepsilon}+1\right)\varphi_{j,m} \tag{66}$$

$$\Leftrightarrow \varphi_j^* = \frac{1}{m}\frac{1}{1-\frac{m-k_j-1}{m}}\left(\frac{k_j}{1+\varepsilon}+1\right)\varphi_{j,m} \tag{67}$$

$$= \frac{1}{k_j+1}\frac{k_j+1+\varepsilon}{1+\varepsilon}\varphi_{j,m} \tag{68}$$

$$= \left(1 + \frac{1}{k_j+1}\varepsilon\right)\frac{1}{1+\varepsilon}\varphi_{j,m}. \tag{69}$$

This results in $1 > \alpha_1(j) = \left(1 + \frac{1}{k_j+1}\varepsilon\right)\frac{1}{1+\varepsilon} \geq \frac{1}{1+\varepsilon}$ for $k_j \in \{1, \ldots, m-1\}$, which is minimal for $k_j = m-1$, leading to $\alpha_1 = \left(1 + \frac{1}{m}\varepsilon\right)\frac{1}{1+\varepsilon}$ in the worst case for any assignment and independently of j.

If we combine these calculations, we obtain

$$\psi'(\boldsymbol{r}) \geq \prod_{j=1}^m \phi_j(\boldsymbol{r}_j) \cdot \alpha_1, \text{ and} \tag{70}$$

$$\psi'(\boldsymbol{r}) \leq \prod_{j=1}^m \phi_j(\boldsymbol{r}_j) \cdot \alpha_2. \tag{71}$$

In consequence, we get the following bounds:

$$\min_{\boldsymbol{r}} \frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})} \geq \frac{\prod_{j=1}^{m} \phi_j(\boldsymbol{r}_j) \cdot \alpha_1}{\prod_{j=1}^{m} \phi_j(\boldsymbol{r}_j)}, \text{ and} \tag{72}$$

$$\max_{\boldsymbol{r}} \frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})} \leq \frac{\prod_{j=1}^{m} \phi_j(\boldsymbol{r}_j) \cdot \alpha_2}{\prod_{j=1}^{m} \phi_j(\boldsymbol{r}_j)}, \tag{73}$$

where $\boldsymbol{r}_j$ is a projection of *any* assignment $\boldsymbol{r}$ to the argument list of ϕ_j. Entering these bounds into Eq. (8) then yields

$$D(P_M, P_{M'}) \tag{74}$$

$$= \ln \max_{\boldsymbol{r}} \frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})} - \ln \min_{\boldsymbol{r}} \frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})} \tag{75}$$

$$\leq \ln \frac{\prod_{j=1}^{m} \phi_j(\boldsymbol{r}_j) \cdot \alpha_2}{\prod_{j=1}^{m} \phi_j(\boldsymbol{r}_j)} - \ln \frac{\prod_{j=1}^{m} \phi_j(\boldsymbol{r}_j) \cdot \alpha_1}{\prod_{j=1}^{m} \phi_j(\boldsymbol{r}_j)} \tag{76}$$

$$= \ln \prod_{j=1}^{m} \alpha_2 - \ln \prod_{j=1}^{m} \alpha_1 \tag{77}$$

$$= \ln \left(\frac{\alpha_2}{\alpha_1} \right)^m \tag{78}$$

$$= \ln \left(\frac{1 + \frac{m-1}{m}\varepsilon}{\frac{1+\frac{1}{m}\varepsilon}{1+\varepsilon}} \right)^m \tag{79}$$

$$= \ln \left(1 + \frac{m-1}{m}\varepsilon \right)^m \left(\frac{1+\varepsilon}{1+\frac{\varepsilon}{m}} \right)^m \tag{80}$$

$$= \ln \left(\frac{\left(1 + \frac{m-1}{m}\varepsilon\right)\left(1+\varepsilon\right)}{1+\frac{1}{m}\varepsilon} \right)^m \tag{81}$$

$$< \ln \ \left(1+\varepsilon\right)^{2m} \tag{82}$$

$$< \ln \left(\frac{1+\varepsilon}{1-\varepsilon} \right)^m. \tag{83}$$

Next, we consider the general case of having an arbitrary distribution of groups of pairwise ε-equivalent factors. More specifically, let the factors now be distributed into $k \geq 2$ groups of pairwise ε-equivalent factors, meaning there is a set of summands $\{m_1, \ldots, m_k\} \in \mathbb{N}^k$ with $\sum_{i=1}^{k} m_i = m$ and $\phi_1, \ldots, \phi_{m_1}$ being pairwise ε-equivalent, $\phi_{m_1+1}, \ldots, \phi_{m_1+m_2}$ being pairwise ε-equivalent, and so on. Since the previous case holds for a group of fully pairwise ε-equivalent factors we can apply this case k times to get the maximal boundary as follows.

$$D(P_M, P_{M'}) \tag{84}$$

$$= \ln \max_{\boldsymbol{r}} \frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})} - \ln \min_{\boldsymbol{r}} \frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})} \tag{85}$$

$$\leq \ln \prod_{j=1}^{m_1} \frac{1 + \frac{m_1-1}{m_1}\varepsilon}{\frac{1+\frac{1}{m_1}\varepsilon}{1+\varepsilon}} \cdot \ldots \cdot \prod_{j=1}^{m_k} \frac{1 + \frac{m_k-1}{m_k}\varepsilon}{\frac{1+\frac{1}{m_k}\varepsilon}{1+\varepsilon}} \tag{86}$$

$$= \ln \left(\frac{1 + \frac{m_1-1}{m_1}\varepsilon}{\frac{1+\frac{1}{m_1}\varepsilon}{1+\varepsilon}} \right)^{m_1} \cdot \ldots \cdot \left(\frac{1 + \frac{m_k-1}{m_k}\varepsilon}{\frac{1+\frac{1}{m_k}\varepsilon}{1+\varepsilon}} \right)^{m_k} \tag{87}$$

Since $1 + \frac{m_i-1}{m_i}\varepsilon < 1 + \frac{m-1}{m}\varepsilon$ holds for the numerator and $\frac{1+\frac{1}{m}\varepsilon}{1+\varepsilon} < \frac{1+\frac{1}{m_i}\varepsilon}{1+\varepsilon}$ holds for the denominator for every $i = 1, \ldots, k$ due to $m_i < m$, we can bound each sub-product individually from above by

$$\left(\frac{1 + \frac{m_i-1}{m_i}\varepsilon}{\frac{1+\frac{1}{m_i}\varepsilon}{1+\varepsilon}} \right)^{m_i} < \left(\frac{1 + \frac{m-1}{m}\varepsilon}{\frac{1+\frac{1}{m}\varepsilon}{1+\varepsilon}} \right)^{m_i}. \tag{88}$$

Putting everything back together yields the desired result:

$$D(P_M, P_{M'}) \tag{89}$$

$$\leq \ln \left(\frac{1 + \frac{m_1-1}{m_1}\varepsilon}{\frac{1+\frac{1}{m_1}\varepsilon}{1+\varepsilon}} \right)^{m_1} \cdot \ldots \cdot \left(\frac{1 + \frac{m_k-1}{m_k}\varepsilon}{\frac{1+\frac{1}{m_k}\varepsilon}{1+\varepsilon}} \right)^{m_k} \tag{90}$$

$$< \ln \prod_{i=1}^{k} \left(\frac{1 + \frac{m-1}{m}\varepsilon}{\frac{1+\frac{1}{m}\varepsilon}{1+\varepsilon}} \right)^{m_i} \tag{91}$$

$$= \ln \left(\frac{1 + \frac{m-1}{m}\varepsilon}{\frac{1+\frac{1}{m}\varepsilon}{1+\varepsilon}} \right)^{\sum_{i=1}^{k} m_i} \tag{92}$$

$$= \ln \left(\frac{1 + \frac{m-1}{m}\varepsilon}{\frac{1+\frac{1}{m}\varepsilon}{1+\varepsilon}} \right)^{m}, \tag{93}$$

which implies due to the strict inequality, in particular, that the second case of multiple groups of pairwise ε-equivalent factors cannot reach the upper bound. □

Theorem 9. *The bound given in Thm. 7 is optimal.*

Proof. We show that the boundary of Thm. 7 can be reached by constructing an FG M such that running Alg. 1 on M to obtain M' results in $D(P_M, P_{M'})$ hitting the boundary. Consider the outcome of Alg. 1 for the FG $M = (\boldsymbol{R} \cup \boldsymbol{\Phi}, \boldsymbol{E})$ whose factors are depicted in Table 3. More specifically, the set of randvars is given by $\boldsymbol{R} = \{R_1, \ldots, R_m\}$ with $\mathrm{range}(R_1) = \ldots = \mathrm{range}(R_m) = \{r_1, \ldots, r_{2m}\}$, and the set of factors is defined as $\boldsymbol{\Phi} = \{\phi_1(R_i), \ldots, \phi_m(R_m)\}$ (i.e., every factor has exactly one randvar as argument and the arguments of all factors are distinct). The set of edges is given by $\boldsymbol{E} = \{\{R_1, \phi_1\}, \ldots, \{R_m, \phi_m\}\}$ but is not relevant for the proof. All factors in $\boldsymbol{\Phi}$ are pairwise ε-equivalent and hence end up in the same group after running Alg. 1.

Therefore, after updating the potentials, all factors in $\boldsymbol{\Phi}$ are replaced by the arithmetic mean $\phi^* = (\varphi_1^*, \ldots, \varphi_{2m}^*)$ (shown in the rightmost column of Table 3) such that

$$\varphi_r^* = \begin{cases} r\left(1 + \frac{1}{m}\varepsilon\right) & \text{for } r = 1, \ldots, m \\ r\left(1 + \frac{m-1}{m}\varepsilon\right) & \text{for } r = m+1, \ldots, 2m. \end{cases} \tag{94}$$

We next compute $\min_{\boldsymbol{r}} \frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})}$. By definition, it holds that $\psi'(\boldsymbol{r}) = \prod_{i=1}^{m} \phi^*(\boldsymbol{r})$. To obtain the minimum quotient, the

R_i	$\phi_1(R_1)$	$\phi_2(R_2)$	$\cdots$	$\phi_{m-1}(R_{m-1})$	$\phi_m(R_m)$	ϕ^*
r_1	$1(1+\varepsilon)$	1	$\cdots$	$\cdots$	1	$1\left(1+\frac{1}{m}\varepsilon\right)$
r_2	2	$2(1+\varepsilon)$	2	$\cdots$	2	$2\left(1+\frac{1}{m}\varepsilon\right)$
$\vdots$	$\vdots$	$\cdots$	$\ddots$		$\vdots$	$\vdots$
$\vdots$	$\vdots$			$\ddots$	$\vdots$	$\vdots$
r_m	m	$\cdots$	$\cdots$	m	$m(1+\varepsilon)$	$m\left(1+\frac{1}{m}\varepsilon\right)$
r_{m+1}	$(m+1)$	$(m+1)(1+\varepsilon)$	$\cdots$	$\cdots$	$(m+1)(1+\varepsilon)$	$(m+1)\left(1+\frac{m-1}{m}\varepsilon\right)$
r_{m+2}	$(m+2)(1+\varepsilon)$	$m+2$	$(m+2)(1+\varepsilon)$	$\cdots$	$(m+2)(1+\varepsilon)$	$(m+2)\left(1+\frac{m-1}{m}\varepsilon\right)$
$\vdots$	$\vdots$	$\cdots$	$\ddots$		$\vdots$	$\vdots$
$\vdots$	$\vdots$			$\ddots$	$\vdots$	$\vdots$
r_{2m}	$2m(1+\varepsilon)$	$\cdots$	$\cdots$	$2m(1+\varepsilon)$	$2m$	$2m\left(1+\frac{m-1}{m}\varepsilon\right)$

Table 3: A construction of an exemplary FG M such that running Alg. 1 on M to obtain M' results in $D(P_M, P_{M'})$ becoming maximal and reaching the boundary given in Thm. 7.

denominator thus must be maximal while at the same time keeping the arithmetic mean small, which is the case for the diagonal entries $\varphi_{i,i}$, $i = 1, \ldots, m$, where $\varphi_{j,i}$ refers to the potential in the j^{th} row in the potential table of factor ϕ_i. Choosing any other columns within the first m rows would decrease the denominator (while leaving the nominator unchanged) and thus cannot lead to the minimum quotient. Furthermore, choosing any rows starting from row $m+1$ also increases the result of the quotient because

$$\frac{i(1+\frac{1}{m}\varepsilon)}{i(1+\varepsilon)} \leq \frac{i(1+\frac{m-1}{m}\varepsilon)}{i(1+\varepsilon)}, \tag{95}$$

with a minimal right hand side for row $m+1$ to $2m$. In other words, the factor of the denominator for the minimum would remain $(1+\varepsilon)$, while the mean φ_i^* for the numerator increases independently of i, resulting in an overall increased quotient compared to choosing rows 1 to m. Consequently, it holds that $\boldsymbol{r}_{\min} = (r_1, \ldots, r_m)$ fulfils the equation $\min_{\boldsymbol{r}} \frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})} = \frac{\psi'(\boldsymbol{r}_{\min})}{\psi(\boldsymbol{r}_{\min})}$, resulting in

$$\min_{\boldsymbol{r}} \frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})} = \prod_{i=1}^{m} \frac{\varphi_i^*}{\varphi_{i,i}} \tag{96}$$

$$= \prod_{i=1}^{m} \frac{i\left(1+\frac{1}{m}\varepsilon\right)}{i\left(1+\varepsilon\right)} \tag{97}$$

$$= \left(\frac{1+\frac{1}{m}\varepsilon}{1+\varepsilon}\right)^m. \tag{98}$$

Analogously, we compute the maximum quotient $\max_{\boldsymbol{r}} \frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})}$ by choosing $\boldsymbol{r}_{\max} = (r_{m+1}, \ldots, r_{2m})$ in the following way:

$$\max_{\boldsymbol{r}} \frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})} = \prod_{i=m+1}^{2m} \frac{\varphi_i^*}{\varphi_{i,i}} \tag{99}$$

$$= \prod_{i=m+1}^{2m} \frac{i\left(1+\frac{m-1}{m}\varepsilon\right)}{i} \tag{100}$$

$$= \prod_{i=1}^{m} \frac{i\left(1+\frac{m-1}{m}\varepsilon\right)}{i} \tag{101}$$

$$= \left(1+\frac{m-1}{m}\varepsilon\right)^m. \tag{102}$$

Inserting the minimum and maximum quotients into the definition of the distance measure $D(P_M, P_{M'})$, we obtain

$$D(P_M, P_{M'}) = \ln \max_{\boldsymbol{r}} \frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})} - \ln \min_{\boldsymbol{r}} \frac{\psi'(\boldsymbol{r})}{\psi(\boldsymbol{r})} \tag{103}$$

$$= \ln \left(\frac{1+\frac{m-1}{m}\varepsilon}{\frac{1+\frac{1}{m}\varepsilon}{1+\varepsilon}}\right)^m \tag{104}$$

$$= \ln \left(\frac{(1+\frac{m-1}{m}\varepsilon)(1+\varepsilon)}{1+\frac{1}{m}\varepsilon}\right)^m, \tag{105}$$

which is strictly smaller than $\ln(1+\varepsilon)^m - \ln(1-\varepsilon)^m$ but also strictly larger than $\ln(1+\varepsilon)^m$ and effectively hits the bound given in Thm. 7.

In practice, however, our bounds improve significantly as soon as we deviate from this extreme scenario. This is because even minor dependencies between the maximum and minimum quotients suffice to obtain a tighter estimate. For instance, if there are more factors in a group than rows in their potential tables or if multiple groups of pairwise ε-equivalent factors exist, we can immediately conclude that the worst-case condition no longer holds, resulting in smaller overall values of D. This effect occurs since even minimal deviations from the worst-case scenario result in dependencies between the maximum and minimum for any assignment $\boldsymbol{r}$. □

B The Advanced Colour Passing Algorithm

The ACP algorithm introduced by Luttermann *et al.* [2024a] builds on the Colour Passing algorithm (originally named CompressFactorGraph) [Kersting *et al.*, 2009; Ahmadi *et al.*, 2013] and solves the problem of constructing a lifted representation, encoded as a PFG, from a given propositional

Algorithm 2 Advanced Colour Passing

Input: An FG $M = (\boldsymbol{R} \cup \boldsymbol{\Phi}, \boldsymbol{E})$ and a set of observed events (evidence) $\boldsymbol{O} = \{E_1 = e_1, \ldots, E_\ell = e_\ell\}$.
Output: A lifted representation M', encoded as a PFG, which entails equivalent semantics as M.

1: Assign each R_i a colour according to $\mathrm{range}(R_i)$ and $\boldsymbol{O}$
2: Assign each ϕ_i a colour according to order-independent potentials and rearrange arguments accordingly
3: **repeat**
4: **for** each factor $\phi \in \boldsymbol{\Phi}$ **do**
5: $signature_\phi \leftarrow [\,]$
6: **for** each randvar $R \in neighbours(M, \phi)$ **do**
7: ▷ In order of appearance in ϕ
8: $append(signature_\phi, R.colour)$
9: $append(signature_\phi, \phi.colour)$
10: Group together all ϕs with the same signature
11: Assign each such cluster a unique colour
12: Set $\phi.colour$ correspondingly for all ϕs
13: **for** each randvar $R \in \boldsymbol{R}$ **do**
14: $signature_R \leftarrow [\,]$
15: **for** each factor $\phi \in neighbours(M, R)$ **do**
16: **if** ϕ is commutative w.r.t. $\boldsymbol{S}$ and $R \in \boldsymbol{S}$ **then**
17: $append(signature_R, (\phi.colour, 0))$
18: **else**
19: $append(signature_R, (\phi.colour, p(R, \phi)))$
20: Sort $signature_R$ according to colour
21: $append(signature_R, R.colour)$
22: Group together all Rs with the same signature
23: Assign each such cluster a unique colour
24: Set $R.colour$ correspondingly for all Rs
25: **until** grouping does not change
26: $M' \leftarrow$ construct PFG from groupings

FG. The idea of ACP is to first find indistinguishabilities in a propositional FG and then group together symmetric subgraphs. In particular, ACP looks for indistinguishabilities based on potentials of factors, on ranges and evidence of randvars, as well as on the graph structure by passing around colours. Algorithm 2 provides a formal description of the ACP algorithm, which proceeds as follows.

ACP begins with the colour assignment to variable nodes, meaning that all randvars that have the same range and observed event are assigned the same colour. Thereafter, ACP assigns a colour to every factor node such that factors representing equivalent potentials are assigned the same colour. After the initial colour assignments, ACP begins to pass the colours around. ACP first passes the colours from every variable node to its neighbouring factor nodes and afterwards, every factor node ϕ sends both its colour and the position $p(R, \phi)$ of R in ϕ's argument list to all of its neighbouring variable nodes R. Factors that have symmetries within themselves, formally denoted as being commutative with respect to a subset $\boldsymbol{S}$ of their arguments, omit the position when sending their colour to a neighbouring variable node $R \in \boldsymbol{S}$. We provide more details on commutative factors in Appendix D.

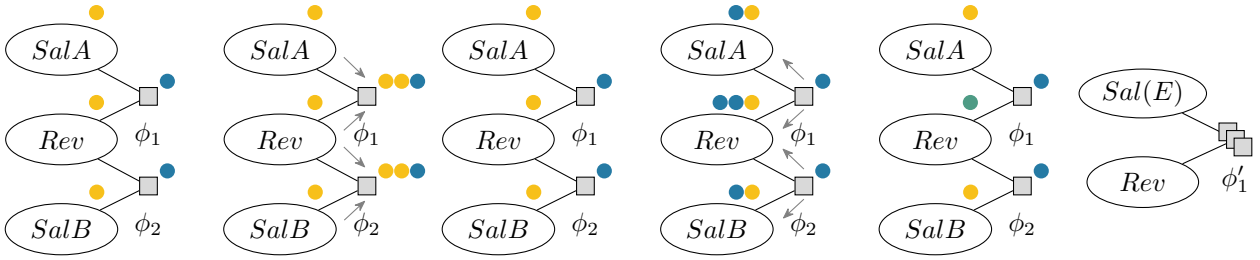
Example 7. *Figure 4 illustrates the course of the ACP algorithm on the FG from Fig. 1 with the modification that $\varphi_i = \varphi'_i$ for all $i \in \{1, \ldots, 4\}$. Recall that $SalA$, $SalB$, and Rev all have the range $\{\mathrm{low}, \mathrm{high}\}$. We further assume that there are no observed events (evidence) and thus, $SalA$, $SalB$, and Rev receive the same colour (e.g.,* yellow*). As the potentials of ϕ_1 and ϕ_2 are identical, ϕ_1 and ϕ_2 are assigned the same colour as well (e.g.,* blue*). The colour passing then starts from variable nodes to factor nodes, that is, $SalA$ and Rev send their colour (*yellow*) to ϕ_1, and Rev and $SalB$ send their colour (*yellow*) to ϕ_2. ϕ_1 and ϕ_2 are then recoloured according to the colours they received from their neighbours to reduce the communication overhead. Since ϕ_1 and ϕ_2 received identical colours (two times the colour* yellow*), they are assigned the same colour during recolouring. Afterwards, the colours are passed from factor nodes to variable nodes and this time not only the colours but also the position of the randvars in the argument list of the corresponding factor are shared (because none of the factors is commutative with respect to a subset of its arguments having size at least two—see Appendix D for more details). Consequently, ϕ_1 sends a tuple $(\mathrm{blue}, 1)$ to $SalA$ and a tuple $(\mathrm{blue}, 2)$ to Rev, and ϕ_2 sends a tuple $(\mathrm{blue}, 2)$ to Rev and a tuple $(\mathrm{blue}, 1)$ to $SalB$ (positions are not shown in Fig. 4). As $SalA$ and $SalB$ are both at position one in the argument list of their respective neighbouring factor, they receive identical messages and are recoloured with the same colour. Rev is assigned a different colour during recolouring than $SalA$ and $SalB$ because Rev received different messages than $SalA$ and $SalB$. The groupings do not change in further iterations and hence the algorithm terminates. The output is the PFG shown on the right in Fig. 4, where both $SalA$ and $SalB$ as well as ϕ_1 and ϕ_2 are grouped.*

For more details on the colour passing procedure and the construction of the PFG in the final step (Line 26), we refer the reader to Luttermann *et al.* [2024a]. Note that when ε-ACP (Alg. 1) calls ACP in Line 15, the colour assignment of ACP to factors (Line 2 in Alg. 2) is replaced by the colour assignment of ε-ACP prior to calling ACP. We also remark that ε-ACP does not use the PFG output by ACP but instead takes the groups computed by ACP, updates them in phase three, and then applies the final step to construct the PFG.

C Permutations of Factors' Arguments

In general, we cannot assume that the tables of equivalent factors (i.e., factors that encode equivalent underlying functions) read identical values from top to bottom. More specifically, it is not always the case that indistinguishable randvars are located at the same position in their respective factors' argument lists. The following example illustrates this point.

Example 8. *Consider the FG M_1 shown in Fig. 5. In fact, M_1 entails equivalent semantics as the FG M_2 depicted in Fig. 1. Observe that the function definition of ϕ_1 is identical in M_1 and M_2 (i.e., its potential table is exactly the same). If we now take a look at ϕ_2, we see that in M_1, the order of its arguments differs from the order of its arguments in M_2. In particular, Rev is now located at position one and $SalB$ at position two in M_1 (opposed to $SalB$ at position one*

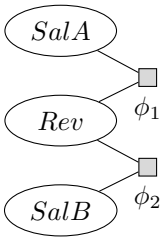


$SalA$	Rev	$\phi_1(SalA, Rev)$
high	high	φ_1
high	low	φ_2
low	high	φ_3
low	low	φ_4

$SalB$	Rev	$\phi_2(SalB, Rev)$
high	high	φ_1
high	low	φ_2
low	high	φ_3
low	low	φ_4

$Sal(E)$	Rev	$\phi'_1(Sal(E), Rev)$
high	high	φ_1
high	low	φ_2
low	high	φ_3
low	low	φ_4

Figure 4: A visualisation of the steps undertaken by the ACP algorithm (Alg. 2) on the input FG from Fig. 1 (left). All randvars have the same range ($\{\text{low}, \text{high}\}$) and there are no observed events (evidence) available. Colours are first passed from variable nodes to factor nodes, followed by a recolouring, and then passed back from factor nodes to variable nodes, again followed by a recolouring. The colour passing procedure is then iterated until convergence (here, the colour assignments remain the same in the next iteration) and the resulting PFG is depicted on the right. In the resulting PFG, $Sal(E)$ is a so-called parameterised randvar that represents a group of randvars ($SalA$ and $SalB$) by using a logical variable E with domain $\text{dom}(E) = \{SalA, SalB\}$. Analogously, ϕ'_1 now represents both ϕ_1 and ϕ_2 simultaneously.



$SalA$	Rev	$\phi_1(SalA, Rev)$
high	high	φ_1
high	low	φ_2
low	high	φ_3
low	low	φ_4

Rev	$SalB$	$\phi_2(Rev, SalB)$
high	high	φ'_1
high	low	φ'_3
low	high	φ'_2
low	low	φ'_4

Figure 5: Another FG modeling the interplay between the revenue of a company (Rev) and the salaries of its employees ($SalA$ and $SalB$). In comparison to Fig. 1, the order of ϕ_2's arguments has now changed. Nevertheless, the semantics (i.e., the underlying full joint probability distribution) of the FG remains the same as in Fig. 1.

and Rev at position two in M_2). The semantics of ϕ_2, however, remains unchanged as all assignments are mapped to the same potential: $Rev = \text{high}$, $SalB = \text{high}$ is mapped to φ'_1, $Rev = \text{high}$, $SalB = \text{low}$ is mapped to φ'_3, $Rev = \text{low}$, $SalB = \text{high}$ is mapped to φ'_2, and $Rev = \text{low}$, $SalB = \text{low}$ is mapped to φ'_4. In consequence, ϕ_2 entails the same semantics in M_1 as in M_2.

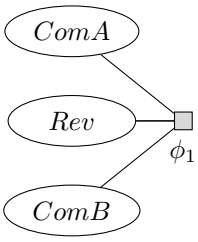
To construct a lifted representation such as a PFG, indistinguishable randvars like $SalA$ and $SalB$ are required to be located at the same position in the argument list of their respective factors (otherwise, they cannot be grouped). Therefore, to account for permutations of factors' arguments when looking for (ε-)equivalent factors, we first have to check whether there exists a rearrangement of the arguments such that the potential tables read identical (ε-equivalent, respectively) values from top to bottom. If such a rearrangement exists, arguments are rearranged accordingly to ensure that indistinguishable randvars are actually located at the same position in their respective factors' argument lists. After the rearrangement, it is guaranteed that the potential tables of (ε-)equivalent factors can be compared row-wise and the ACP algorithm (Alg. 2) assigns identical colours to them. More details on how such rearrangements of arguments can be computed efficiently are given in [Luttermann *et al.*, 2024e].

D Approximate Symmetries Within Factors

Symmetries within factors arise when arguments of the same factor are indistinguishable. A factor that contains symmetries within itself is referred to as a *commutative factor* in the literature. The next definition formally introduces the notion of a commutative factor.

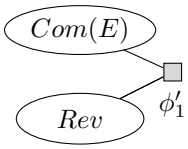
Definition 4 (Commutative Factor, Luttermann *et al.*, 2024a)**.** *Let $\phi(R_1, \ldots, R_n)$ denote a factor. We say that ϕ is* commutative with respect to $\boldsymbol{S} \subseteq \{R_1, \ldots, R_n\}$ *if for all events $r_1, \ldots, r_n \in \times_{i=1}^{n} \text{range}(R_i)$ it holds that $\phi(r_1, \ldots, r_n) = \phi(r_{\pi(1)}, \ldots, r_{\pi(n)})$ for all permutations π of $\{1, \ldots, n\}$ with $\pi(i) = i$ for all $R_i \notin \boldsymbol{S}$. If ϕ is commutative with respect to $\boldsymbol{S}$, all arguments in $\boldsymbol{S}$ are called* commutative arguments.

Example 9. *Consider the FG shown in Fig. 6a, which models the interplay between a company's revenue Rev and the competences $ComA$ and $ComB$ of its employees. It holds that $\text{range}(ComA) = \text{range}(ComB) = \text{range}(Rev) = \{\text{low}, \text{high}\}$. Here, ϕ_1 is commutative with respect to $\boldsymbol{S} = \{ComA, ComB\}$ because $\phi_1(Rev = \text{high}, ComA = \text{high}, ComB = \text{low}) = \phi_1(Rev = \text{high}, ComA = \text{low}, ComB = \text{high}) = \varphi_2$ and $\phi_1(Rev = \text{low}, ComA = \text{high}, ComB = \text{low}) = \phi_1(Rev = \text{low}, ComA = \text{low}, ComB = \text{high}) = \varphi_5$. In other words, the order of $ComA$ and $ComB$ in ϕ_1 does not matter, i.e., it is only relevant how many employees have a high competence and how many have a low competence but not which specific employees have a high or*



Rev	$ComA$	$ComB$	ϕ_1
high	high	high	φ_1
high	high	low	φ_2
high	low	high	φ_2
high	low	low	φ_3
low	high	high	φ_4
low	high	low	φ_5
low	low	high	φ_5
low	low	low	φ_6

(a)



Rev	$\#_E[Com(E)]$	ϕ'_1
high	$[2,0]$	φ_1
high	$[1,1]$	φ_2
high	$[0,2]$	φ_3
low	$[2,0]$	φ_4
low	$[1,1]$	φ_5
low	$[0,2]$	φ_6

(b)

Figure 6: (a) An FG modelling the interplay between the revenue Rev of a company and the competences $ComA$ and $ComB$ of its employees, and (b) a lifted representation, encoded as a PFG, of the FG shown in (a). For brevity, we omit the argument lists of ϕ_1 and ϕ'_1 and write ϕ_1 instead of $\phi_1(Rev, ComA, ComB)$ and ϕ'_1 instead of $\phi'_1(Rev, \#_E[Com(E)])$.

low competence. In consequence, ϕ_1 can be represented more compactly without losing any information. Figure 6b shows a PFG encoding equivalent semantics as the FG from Fig. 6a by using a so-called counting randvar $\#_E[Com(E)]$ that counts over the competences of all employees. More specifically, instead of listing the competence of every employee separately, the counting randvar now specifies counts for the ranges values of $ComA$ and $ComB$: $[2,0]$ represents the case that there are two employees with a high competence and none with a low competence, $[1,1]$ represents the case that there is one employee with a high competence and one with a low competence, and $[0,2]$ represents the case that there are two employees with a low competence and none with a high competence. Note that the potential table is now smaller than in the initial FG from Fig. 6a but still encodes exactly the same semantics.

The previous example gives a glimpse of how *exact* symmetries within a factor can be exploited to obtain a more compact lifted representation. We remark that it is also possible to allow for approximate symmetries within factors by replacing the exact equality in Def. 4 with the notion of ε-equivalence. To enable the exploitation of approximate symmetries within factors in the ε-ACP algorithm, the call of ACP is adjusted such that ACP computes commutative subsets of arguments using ε-equivalence instead of exact equality. When inserting a counting randvar in the final PFG, the potential table is again constructed by using the arithmetic mean over the original potentials. By doing so, every updated potential still differs by factor at most $(1 \pm \varepsilon)$ from its original potential and hence, the bounds on the change in query results from Sec. 4 continue to hold.

E Further Experimental Results

In addition to the results given in Sec. 5, we give separate plots illustrating the distributions of quotients of query results in the modified model and query results in the original model for specific choices of x (proportions of factors whose potential tables differ by factor at most $(1 \pm \varepsilon)$ from their original potential table) and ε. We also investigate the overhead introduced by ε-ACP compared to ACP in terms of the number of queries needed to amortise the additional offline effort of computing groups of pairwise ε-equivalent factors.

Figures 7 to 12 depict the distributions of quotients of query results in the modified model and query results in the original model for various choices of x. In each of the figures, the left plot shows distributions of quotients for each domain size $k \in \{2, 4, 8, 16, 32, 64, 128\}$ for $\varepsilon = 0.001$ while the right plot shows distributions of quotients for each domain size k for $\varepsilon = 0.1$. Figure 7 shows distributions of quotients for a proportion of $x = 0.1$ of factors that are manipulated such that their potential tables deviate by factor at most $(1 \pm \varepsilon)$, Fig. 8 shows distributions of quotients for $x = 0.3$, Fig. 9 for $x = 0.5$, Fig. 10 for $x = 0.7$, Fig. 11 for $x = 0.9$, and Fig. 12 for $x = 1.0$. As expected, the quotients are generally larger for $\varepsilon = 0.1$ than for $\varepsilon = 0.001$ independent of the choice of x. All quotients are again close to one—in particular, Figs. 7 to 12 exhibit similar patterns as the right plot in Fig. 3 (again, even outliers remain very close to one). With increasing value of x, the distributions of quotients become less concentrated at value one (that is, the boxes span a wider range when going from Fig. 7 to Fig. 12). In other words, there are more queries for which the quotient is further away from one (however, "further away" still refers to numbers extremely close to one). To summarise, even if a large proportion of potential tables is modified by adding noise controlled by a factor $(1 \pm \varepsilon)$, the approximation error remains very small—even for the choice of $\varepsilon = 0.1$. Next, we take a look at the overhead introduced by ε-ACP to compute groups of pairwise ε-equivalent factors by comparing the offline run times of ε-ACP and ACP.

Figures 13 to 15 report the average number α of queries after which the additional offline effort of ε-ACP compared to ACP amortises. More specifically, α is defined as $\alpha = \Delta_o \,/\, \Delta_g$ with $\Delta_o = t_{\varepsilon\text{-ACP}} - t_{\text{ACP}}$ being the difference between the offline run time of ε-ACP and ACP and $\Delta_g = t_{\text{LVE}-\text{ACP}} - t_{\text{LVE}-\varepsilon\text{-ACP}}$ being the difference of the run time of lifted variable elimination on the output of ACP and ε-ACP to answer a query. In other words, after α queries, the additional time needed by ε-ACP to compute groups of pairwise ε-equivalent factors is saved by the faster inference times to answer queries on the output of ε-ACP.

The boxplots depicted in Figs. 13 to 15 show that the median value for α is always smaller than ten and most of the time even smaller than one. Thus, after less than ten queries,

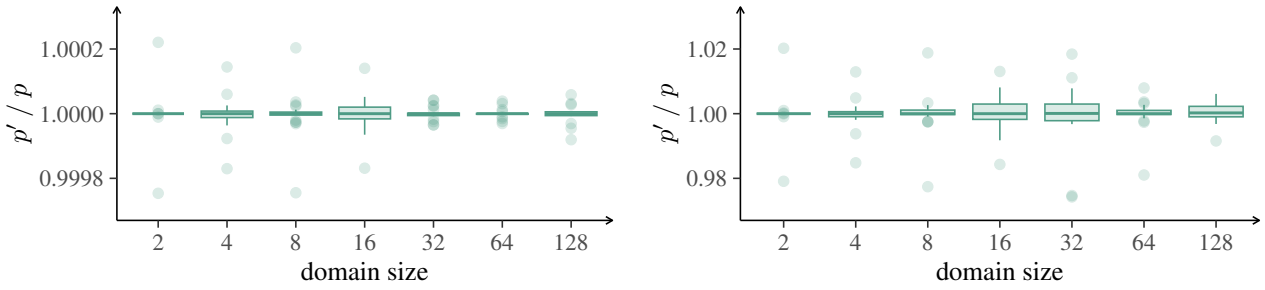


Figure 7: Boxplots showing the distribution of the quotient $p' \,/\, p$ with $p' = P_{M'}(r \mid \boldsymbol{e})$ and $p = P_M(r \mid \boldsymbol{e})$ for a proportion of $x = 0.1$ of factors whose potential tables deviate by factor at most $(1 \pm \varepsilon)$ from the original potential tables, where $\varepsilon = 0.001$ (left) and $\varepsilon = 0.1$ (right).

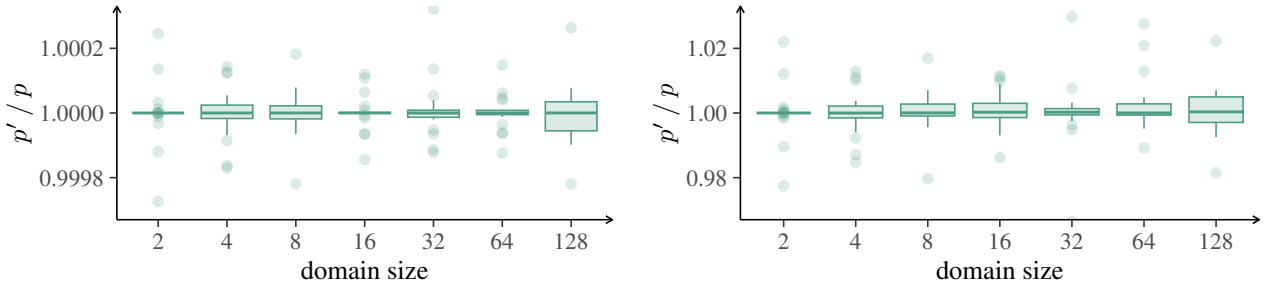


Figure 8: Boxplots showing the distribution of the quotient $p' \,/\, p$ with $p' = P_{M'}(r \mid \boldsymbol{e})$ and $p = P_M(r \mid \boldsymbol{e})$ for a proportion of $x = 0.3$ of factors whose potential tables deviate by factor at most $(1 \pm \varepsilon)$ from the original potential tables, where $\varepsilon = 0.001$ (left) and $\varepsilon = 0.1$ (right).

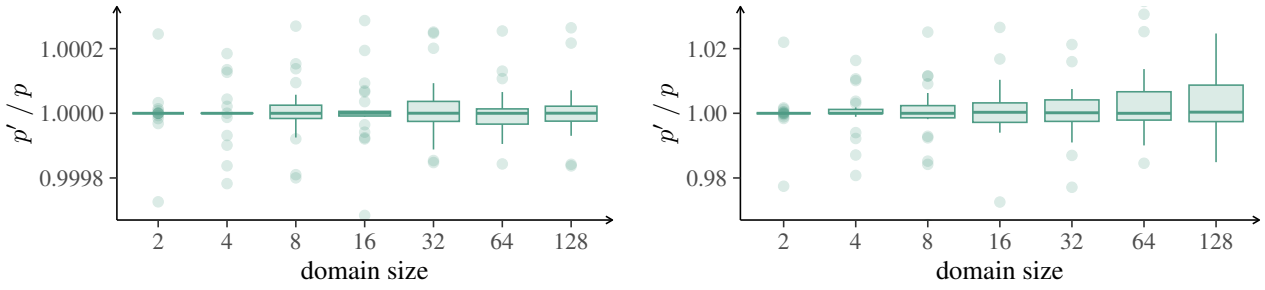


Figure 9: Boxplots showing the distribution of the quotient $p' \,/\, p$ with $p' = P_{M'}(r \mid \boldsymbol{e})$ and $p = P_M(r \mid \boldsymbol{e})$ for a proportion of $x = 0.5$ of factors whose potential tables deviate by factor at most $(1 \pm \varepsilon)$ from the original potential tables, where $\varepsilon = 0.001$ (left) and $\varepsilon = 0.1$ (right).

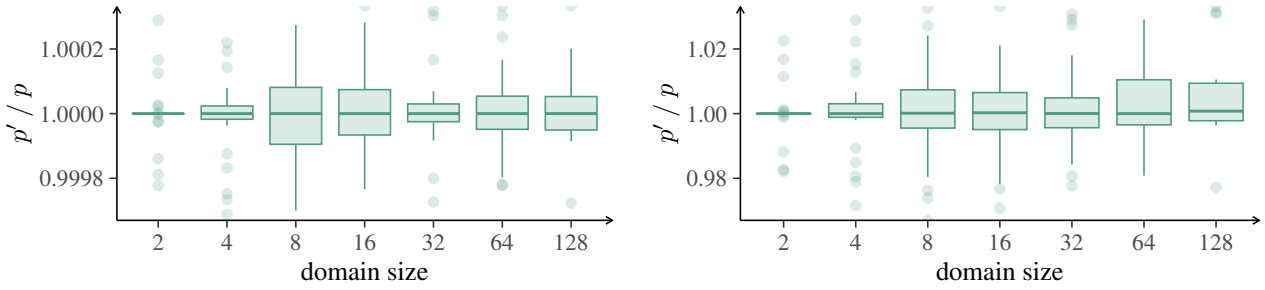


Figure 10: Boxplots showing the distribution of the quotient $p' \,/\, p$ with $p' = P_{M'}(r \mid \boldsymbol{e})$ and $p = P_M(r \mid \boldsymbol{e})$ for a proportion of $x = 0.7$ of factors whose potential tables deviate by factor at most $(1 \pm \varepsilon)$ from the original potential tables, where $\varepsilon = 0.001$ (left) and $\varepsilon = 0.1$ (right).

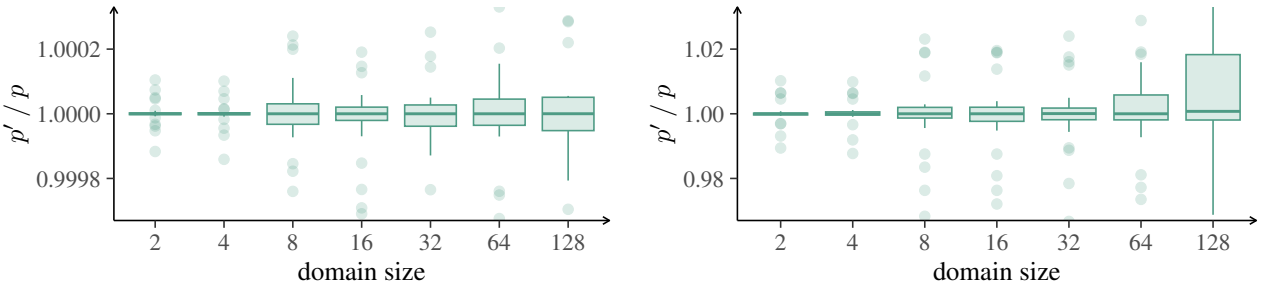


Figure 11: Boxplots showing the distribution of the quotient $p' \,/\, p$ with $p' = P_{M'}(r \mid \boldsymbol{e})$ and $p = P_M(r \mid \boldsymbol{e})$ for a proportion of $x = 0.9$ of factors whose potential tables deviate by factor at most $(1 \pm \varepsilon)$ from the original potential tables, where $\varepsilon = 0.001$ (left) and $\varepsilon = 0.1$ (right).

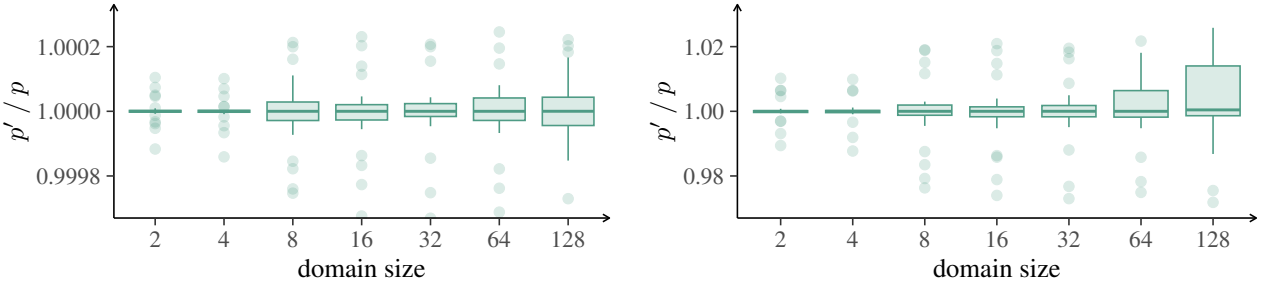


Figure 12: Boxplots showing the distribution of the quotient $p' \,/\, p$ with $p' = P_{M'}(r \mid \boldsymbol{e})$ and $p = P_M(r \mid \boldsymbol{e})$ for a proportion of $x = 1.0$ of factors whose potential tables deviate by factor at most $(1 \pm \varepsilon)$ from the original potential tables, where $\varepsilon = 0.001$ (left) and $\varepsilon = 0.1$ (right).

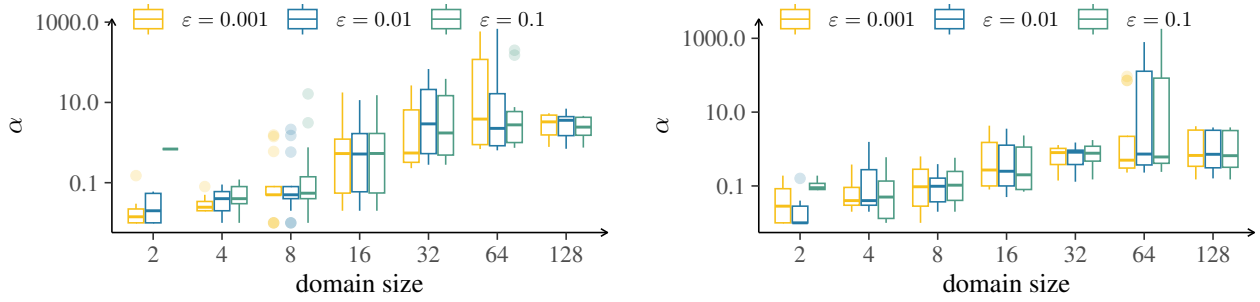


Figure 13: Boxplots illustrating the distributions of the number α of queries after which the additional offline effort of ε-ACP amortises for input FGs containing a proportion of $x = 0.1$ (left) and $x = 0.3$ (right) of factors that are modified by factor at most $(1 \pm \varepsilon)$.

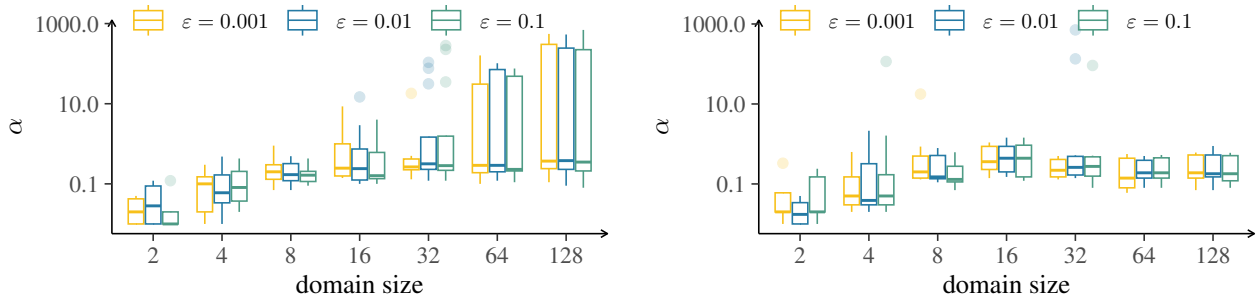


Figure 14: Boxplots illustrating the distributions of the number α of queries after which the additional offline effort of ε-ACP amortises for input FGs containing a proportion of $x = 0.5$ (left) and $x = 0.7$ (right) of factors that are modified by factor at most $(1 \pm \varepsilon)$.

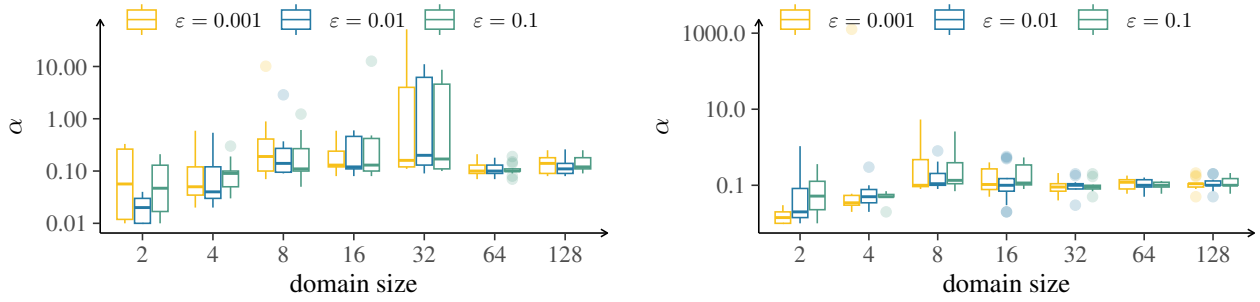


Figure 15: Boxplots illustrating the distributions of the number α of queries after which the additional offline effort of ε-ACP amortises for input FGs containing a proportion of $x = 0.9$ (left) and $x = 1.0$ (right) of factors that are modified by factor at most $(1 \pm \varepsilon)$.

the offline overhead of ε-ACP amortises in most scenarios. There are some outliers where $\alpha > 1000$, however, the overhead introduced by ε-ACP is in the scale of milliseconds to a few seconds in these cases (in fact, the overhead is always smaller than ten seconds, that is, ε-ACP takes at most ten seconds longer than ACP in every scenario). We can therefore conclude that the offline overhead is not only small but also amortises after just a few queries in general. Overall, there is a tendency that the larger x, the smaller is α (apart from a few outliers), which is expected as the advantage of ε-ACP increases with larger values of x. However, when comparing the different domain sizes $k \in \{2, 4, 8, 16, 32, 64, 128\}$, there is no clear pattern on how α behaves. In the left plot of Fig. 14, for example, α has a tendency to become larger for larger domain sizes. This pattern, however, cannot be observed in general in the other plots.