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 introduction by Poole [2003], lifted variable elimination has steadily been refined by many researchers to reach its current form [De Salvo Braz *et al.*, 2005; De Salvo Braz *et al.*, 2006; Milch *et al.*, 2008; Kisyński and Poole, 2009; Taghipour *et al.*, 2013; Braun and Möller, 2018]. More recently, Luttermann et al. [2024b; 2024c] 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; Luttermann *et al.*, 2024d; Luttermann *et al.*, 2024e; Luttermann *et al.*, 2024f], 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 in practice.

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 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*

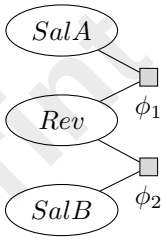
$$\psi(\boldsymbol{r}) = \prod_{j=1}^{m} \phi_j(\boldsymbol{r}_j), \tag{1}$$

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*

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

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

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 the appendix, which is available in an extended version of this paper at https://arxiv.org/abs/2504.20784.

[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 the appendix. 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 (proof given in the appendix). 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.

[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 $Err(\phi_i, \phi_j) = \sum_{r_1, \ldots, r_n} (\phi_i(r_1, \ldots, r_n) - \phi_j(r_{\pi(1)}, \ldots, r_{\pi(n)}))^2$.

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

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 the appendix). 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] 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 the appendix).

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 the appendix).

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*

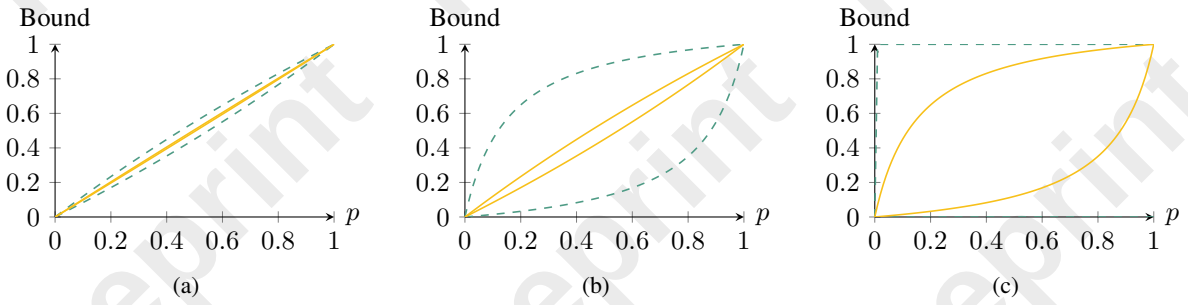


Figure 2: Plots of the bound given in Eq. (10) with $d = \ln(1+\varepsilon)^m - \ln(1-\varepsilon)^m$. Bounds are illustrated for (a) $m = 10$, (b) $m = 100$, and (c) $m = 1000$ where $\varepsilon = 0.01$ (dashed line) and $\varepsilon = 0.001$ (solid line), respectively. The x-axes depict the initial 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 the appendix. 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 the appendix. □

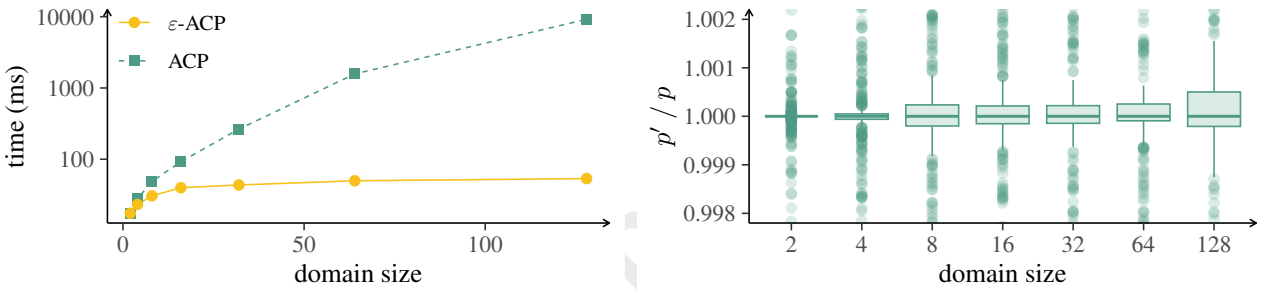


Figure 3: Average query times of lifted variable elimination on the output of ACP and ε-ACP for every choice of k (left), and a boxplot showing the distribution of the quotient $p' \,/\, p$, where $p' = P_{M'}(r \mid \boldsymbol{e})$ and $p = P_M(r \mid \boldsymbol{e})$, for each choice of k (right).

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 the appendix.

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 the appendix.

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

[Ahmadi *et al.*, 2013] 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.

[Braun and Möller, 2018] 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.

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

[De Salvo Braz *et al.*, 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.

[De Salvo Braz *et al.*, 2006] 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.

[Frey *et al.*, 1997] 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.

[Johnson *et al.*, 2023] 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.

[Kersting *et al.*, 2009] 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.

[Kisyński and Poole, 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.

[Kschischang *et al.*, 2001] 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.

[Luttermann *et al.*, 2024a] 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.

[Luttermann *et al.*, 2024b] 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.

[Luttermann *et al.*, 2024c] 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.

[Luttermann *et al.*, 2024d] 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.

[Luttermann *et al.*, 2024e] 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.

[Luttermann *et al.*, 2024f] 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.

[Milch *et al.*, 2008] 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.

[Niepert and Van den Broeck, 2014] 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.

[Poole, 2003] 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.

[Taghipour *et al.*, 2013] 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.

[Weisfeiler and Leman, 1968] 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.