Compression versus Accuracy: A Hierarchy of Lifted Models

Jan Speller[a,*], **Malte Luttermann**[b,c], **Marcel Gehrke**[c] **and Tanya Braun**[a]

[a]Computer Science Department, University of Münster, Germany
[b]German Research Center for Artificial Intelligence (DFKI), Lübeck, Germany
[c]Institute for Humanities-Centered Artificial Intelligence, University of Hamburg, Germany
ORCID (Jan Speller): https://orcid.org/0000-0003-1106-8177, ORCID (Malte Luttermann): https://orcid.org/0009-0005-8591-6839, ORCID (Marcel Gehrke): https://orcid.org/0000-0001-9056-7673, ORCID (Tanya Braun): https://orcid.org/0000-0003-0282-4284

Abstract. Probabilistic graphical models that encode indistinguishable objects and relations among them use first-order logic constructs to compress a propositional factorised model for more efficient (lifted) inference. To obtain a lifted representation, the state-of-the-art algorithm Advanced Colour Passing (ACP) groups factors that represent matching distributions. In an approximate version using ε as a hyperparameter, factors are grouped that differ by a factor of at most $(1 \pm \varepsilon)$. However, finding a suitable ε is not obvious and may need a lot of exploration, possibly requiring many ACP runs with different ε values. Additionally, varying ε can yield wildly different models, leading to decreased interpretability. Therefore, this paper presents a hierarchical approach to lifted model construction that is hyperparameter-free. It efficiently computes a hierarchy of ε values that ensures a hierarchy of models, meaning that once factors are grouped together given some ε, these factors will be grouped together for larger ε as well. The hierarchy of ε values also leads to a hierarchy of error bounds. This allows for explicitly weighing compression versus accuracy when choosing specific ε values to run ACP with and enables interpretability between the different models.

1 Introduction

Probabilistic graphical models (PGMs) allow for modelling environments under uncertainty by encoding features in random variables (randvars) and relations between them in factors. Lifted or first-order versions of PGMs such as parametric factor graphs (FGs) [17] and Markov logic networks [18] incorporate logic constructs to encode indistinguishable objects and relations among them in a compact way. Probabilistic inference on such first-order models is tractable in the domain size if using representatives for indistinguishable objects [16], a technique referred to as lifting [17]. Lifting has been used to great effect in probabilistic inference including lifting various query answering algorithms [1, 2, 8, 9, 19, 22] next to lifting queries [3, 20] or evidence [19, 21].

Advanced Colour Passing (ACP) is the state-of-the-art algorithm to get a first-order model from a propositional one, specifically turning FGs into parametric FGs, grouping factors with identical potentials [1, 13]. The newest version, ε-Advanced Colour Passing (ε-ACP), considers approximate indistinguishability, using a number ε to group factors whose potentials differ by a factor of at most $(1 \pm \varepsilon)$ and are therefore considered ε-equivalent, with ε as a hyperparameter [14]. ε-ACP allows for compressing a propositional model to a larger degree with increasing ε, leading to better runtime for inference tasks. It also allows for computing an approximation error for such inference tasks, which helps assess the accuracy versus the compression gained. However, if a chosen ε does not fulfil either a requirement for compression or accuracy, shortcomings become apparent: A new run of ε-ACP with a different ε does not guarantee a model consistent with the previous one. E.g., with a larger ε, making more factors ε-equivalent, factors that were previously grouped together might no longer be part of the same group, because a different grouping appears more suitable. That is, the models do not form a hierarchy, where groups of factors under a larger ε can only form by merging groups under a smaller ε. This inconsistency in models from one ε to the next makes it hard to interpret the models regarding each other. Additionally, ε is a hyperparameter that has to be chosen by the user. It may not be obvious what a suitable value for ε is, requiring many runs of ε-ACP to find a suitable one with the necessary compression and accuracy.

To counteract these shortcomings, this paper presents a hierarchical approach to lifted model construction called hierarchical ACP (HACP), which is hyperparameter-free, i.e., there is no need to choose a value for ε in advance. Specifically, we calculate a hierarchy of ε values that guarantees a hierarchy of compressed models when running HACP, as more compressed models naturally encompass supersets of grouped ε-equivalent factors from their less compressed counterparts. To do so, this paper contributes the following:

(i) a one-dimensional ε-equivalence distance (1DEED) measure that reduces the ε-equivalence after computations to a single number for comparing different factors in contrast to the previous definition of ε-equivalence,
(ii) an efficient algorithm for computing a hierarchical ordering of ε values,
(iii) HACP using the hierarchical ordering of ε values, yielding a hierarchy of parametric FGs, and
(iv) an analysis of the error bounds in the hierarchy of parametric FGs, showing a hierarchical order as well.

* Corresponding Author. Email: jan.speller@uni-muenster.de

The hierarchical approach has the advantage of being hyperparameter-free: The hierarchy of ε values can be computed before running HACP and without needing a starting value. In combination with the error bounds, the hierarchy of ε values allows to choose for which ε values to actually run HACP, which means that a user does not need to do a hyperparameter exploration to find the most suitable one. It also has the upside that one has to run HACP only for those ε values that are actually of interest. We approach this from the perspective of distributional deviation (accuracy), but also in the context of group merging of ε-equivalent factors by controlling ε (compression). In this context, we also investigate the potential loss of accuracy by the forced hierarchical structure in comparison to the ε-ACP. Finally, the different models as a result of the ε values in the hierarchy are consistent and as such interpretable with respect to each other. This provides insight into the underlying symmetries within the different levels of the approximated FG, where its complexity is implicitly captured by the variations and proximity of distinct ε values and the heterogeneity of group memberships. Consequently, this allows for an informed choice of an appropriate ε in a way that suits specific requirements for applications.

The remaining part of this paper is structured as follows: The paper starts with introducing necessary notations and briefly recaps ε-ACP, which is followed by the main part, which provides a definition of 1DEED, specifies how to calculate a hierarchical ordering of ε values, and presents HACP. Then, it shows maximal error bounds for HACP and ends with a discussion and conclusion. The technical appendix includes more detailed proofs and illustrations.

2 Background

We start by defining an FG as a propositional probabilistic graphical model to compactly encode a full joint probability distribution over a set of randvars [7, 12]. The following definitions are given via [14].

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 $\boldsymbol{R} = \{R_1, \ldots, R_n\}$ is a set of randvars and $\boldsymbol{\Phi} = \{\phi_1, \ldots, \phi_m\}$ is a set of factors (functions), as well as a set of edges $\boldsymbol{E} \subseteq \boldsymbol{R} \times \boldsymbol{\Phi}$. There is an edge between a randvar $R_i \in \boldsymbol{R}$ and a factor $\phi_j \in \boldsymbol{\Phi}$ in $\boldsymbol{E}$ if R_i appears in the argument list of ϕ_j. A factor $\phi_j(\mathcal{R}_j)$ defines a function $\phi_j \colon \times_{R \in \mathcal{R}_j} \mathit{range}(R) \to \mathbb{R}_{>0}$ 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 $\mathit{range}(R)$ denotes the possible values a randvar R can take. We further define the joint potential for an assignment $\boldsymbol{r}$ (with $\boldsymbol{r}$ being 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. With $Z = \sum_{\boldsymbol{r}} \prod_{j=1}^{m} \phi_j(\boldsymbol{r}_j)$ as the normalisation constant, the full joint probability distribution encoded by M is then given by

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

Example 1. *Consider the FG illustrated in Fig. 1. It holds that $\boldsymbol{R} = \{A, B, C\}$, $\boldsymbol{\Phi} = \{\phi_1, \phi_2\}$, and $\boldsymbol{E} = \{\{A, \phi_1\}, \{B, \phi_1\}, \{B, \phi_2\}, \{C, \phi_2\}\}$. For the sake of this example, let $\mathit{range}(A) = \mathit{range}(B) = \mathit{range}(C) = \{\text{true}, \text{false}\}$. The potential tables of ϕ_1 and ϕ_2 are shown on the right of Fig. 1 with $\phi_1(\text{true}, \text{true}) = \varphi_1$ and so on, where $\varphi_i \in \mathbb{R}_{>0}$, $i = 1, \ldots, 4$, are arbitrary positive real numbers.*

In Def. 1, we stipulate that all potentials are strictly greater than zero to avoid division by zero when analysing theoretical bounds in subsequent sections of this paper. In general, it is sufficient to have at least one non-zero potential in every potential table to ensure a well-defined semantics of an FG. However, our requirement of having strictly positive potentials is no restriction in practice as zeros can easily be replaced by tiny numbers that are close to zero. An FG can be queried to compute marginal distributions of randvars given observations for other randvars (referred to as probabilistic inference).

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

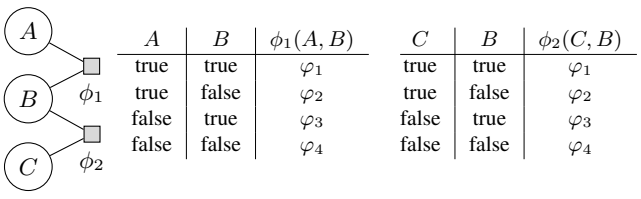
Lifted inference exploits identical behaviour of indistinguishable objects to answer queries more efficiently. The idea behind lifting is to use a representative of indistinguishable objects for computations. Formally, this corresponds to making use of exponentiation instead of multiplying identical potentials several times (for an example, see Appendix C). To exploit exponentiation during inference, equivalent factors have to be grouped. We next introduce the concept of ε-equivalent factors [14], which allows us to determine factors that can potentially be grouped for lifted inference.

Definition 3 (ε-Equivalence)**.** *Let $\varepsilon \in \mathbb{R}_{>0}$ be a positive real number. Two potentials $\varphi_1, \varphi_2 \in \mathbb{R}_{>0}$ 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 $(r_1, \ldots, r_n) \in \times_{i=1}^{n} \mathit{range}(R_i)$, 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 $\varphi_1 =_\varepsilon \varphi_2$.*

Example 2. *Consider the potentials $\varphi_1 = 0.49$, $\varphi_2 = 0.5$, and $\varepsilon = 0.1$. Since it holds that $\varphi_2 = 0.5 \in [\varphi_1 \cdot (1 - \varepsilon) = 0.441, \varphi_1 \cdot (1+\varepsilon) = 0.539]$ and $\varphi_1 = 0.49 \in [\varphi_2 \cdot (1-\varepsilon) = 0.45, \varphi_2 \cdot (1+\varepsilon) = 0.55]$, φ_1 and φ_2 are ε-equivalent (for $\varepsilon = 0.1$).*

The notion of ε-equivalence is symmetric. Moreover, it might happen that indistinguishable objects are located at different positions in the argument list of their respective factors, which is the reason the definition considers permutations of arguments. For simplicity, in this paper, we stipulate that π is the identity function. However, all presented results also apply to any other choice of π [14].

The ε-ACP algorithm [14] computes groups of pairwise ε-equivalent factors to compress a given FG. In particular, as potentials are often estimated in practice, potentials that should actually be considered equal might slightly differ and the ε-ACP algorithm accounts for such deviations using a hyperparameter ε, which controls the trade-off between compression and accuracy of the resulting lifted representation. To allow for exponentiation, ε-ACP computes the mean potentials for each group of pairwise ε-equivalent factors and replaces the original potentials of the factors by the respective



A	B	$\phi_1(A, B)$
true	true	φ_1
true	false	φ_2
false	true	φ_3
false	false	φ_4

C	B	$\phi_2(C, B)$
true	true	φ_1
true	false	φ_2
false	true	φ_3
false	false	φ_4

Figure 1. An exemplary FG encoding a full joint probability distribution over three randvars A, B, and C.

mean potentials. Thus, the semantics of the FG changes after the replacement of potentials. However, replacing potentials by their mean guarantees specific error bounds (more details follow in Section 4).

Note 1. *ε-ACP uses the introduced concept of ε-equivalence including the corresponding hyperparameter $\varepsilon \in \mathbb{R}_{>0}$ requiring repeated, dimension-wise grouping checks with no reusable aggregate structure. This can be interpreted as a regularisation approach from the original model (with $\varepsilon \approx 0$) to the most trivial model by reducing complexity by increasing ε. For arbitrary large $\varepsilon \gg 0$, it reduces the model to an FG that considers all factors of the same dimension and its corresponding multiset of randvars with same range sizes as pairwise ε-equivalent, where the dimension of a factor ϕ refers to the number of rows in the potential table of ϕ. ε-ACP does not yield hierarchical models, since the grouping process is independent for each choice of ε.*

Next, we present a hierarchical version of ε-ACP that ensures a hierarchy of models over increasing ε.

3 Hierarchical Lifted Model Construction

As mentioned above, ε-ACP lacks a mechanism to ensure the core property of hierarchical methods: the consistent embedding of simpler models into more complex ones. To construct a principled hierarchical organisation of models, a clear mechanism for defining and transferring group membership across levels is essential. Concretely, we define a hierarchy, in which higher levels inherit structural properties from lower levels, thereby inducing a consistent reduction in the complexity of the FG. To this end, we present the *1DEED*, a more effective criterion for determining ε-equivalence. To do so, we treat the potential table of a factor ϕ as a vector in $\mathbb{R}^n_{>0}$, where $\phi(k)$ denotes the k-th entry, i.e., the potential associated with the k-th row in the potential table of ϕ. For example, factor $\phi_1(A, B)$ in Fig. 1 is represented as the vector $(\varphi_1, \varphi_2, \varphi_3, \varphi_4)$, with, e.g., $\phi_1(\text{true}, \text{false}) = \phi_1(2) = \varphi_2$. After introducing 1DEED, we use it to set up a hierarchical ordering of ε-equivalent factors, which forms the backbone to a hierarchical approach for lifted model construction based on ε-ACP.

3.1 One-dimensional ε-Equivalence Distance

We define 1DEED as a measure to compare two n-dimensional, strictly positive vectors, representing factors in an FG, i.e., $\phi_i \in \mathbb{R}^n_{>0}$ with $\phi_i(k) > 0$ for all k.

Definition 4 (One-dimensional ε-equivalence distance)**.** 1DEED *defined as the mapping $d_\infty \colon \mathbb{R}^n_{>0} \times \mathbb{R}^n_{>0} \to \mathbb{R}$ for two n-dimensional vectors $\phi_1, \phi_2 \in \mathbb{R}^n_{>0}$ is given by:*

$$d_\infty(\phi_1, \phi_2) := \max_{k=1,\dots,n} \left\{ \left| \frac{\phi_1(k) - \phi_2(k)}{\phi_1(k)} \right|, \left| \frac{\phi_1(k) - \phi_2(k)}{\phi_2(k)} \right| \right\} = \max_{k=1,\dots,n} \left\{ \frac{|\phi_1(k) - \phi_2(k)|}{\min\{|\phi_1(k)|, |\phi_2(k)|\}} \right\} \tag{3}$$

The first properties of 1DEED as a distance measure are direct results of its definition.

Corollary 1. *The following properties hold for 1DEED.*

(i) 1DEED is non-negative and symmetric.
(ii) It holds that $d_\infty(\phi_1, \phi_2) = 0$ if and only if $|\phi_1(k) - \phi_2(k)| = 0$ for all $k = 1, \dots, n$, which holds if and only if $\phi_1 = \phi_2$.

This distance is based on the *maximum metric (Chebyshev distance)* with an additional deviation. It does not satisfy the triangle inequality and thus is not a metric, which can be demonstrated using a counterexample (see Appendix A). Nonetheless, 1DEED is well-suited for bounding maximum relative deviations in all dimensions, aligning with practical use in probabilistic inference. The nature of its definition is no coincidence, but rather subject to the purpose to induce ε-equivalence by being consistent with the existing concept.

Theorem 2. *Two vectors $\phi_1, \phi_2 \in \mathbb{R}^n_{>0}$ are ε-equivalent (Definition 3) if and only if $d_\infty(\phi_1, \phi_2) \leq \varepsilon$ holds.*

Proof Sketch. In mathematical terms, the claim can be summarised as $\phi_1 =_\varepsilon \phi_2 \Leftrightarrow d_\infty(\phi_1, \phi_2) \leq \varepsilon$, which we prove for any $\varepsilon > 0$ in Appendix B via equivalence transformations. □

The use of relative deviation is essential in probabilistic querying, where percentage-based error bounds directly influence the quality of inference. However, classical definitions of ε-equivalence require checking all component-wise comparisons, leading to inefficiencies in practice and no clear order of ε-equivalent factor groups. The one-dimensional formulation via d_∞ addresses this by providing a closed-form, computationally minimal characterisation of ε-equivalence. Specifically, it allows for an efficient computation of the smallest admissible ε for each pair of factors, facilitating both storage and comparison. That is $d_\infty(\phi_1, \phi_2) = \varepsilon_0$ uniquely determines the minimal $\varepsilon_0 \leq \varepsilon$ for which $\phi_1 =_\varepsilon \phi_2$ still holds. This scalar value enables direct comparison, indexing, and efficient classification of equivalence classes without enumerating all component-wise ratios. The formal operational benefits – both in terms of computational complexity and structural hierarchy – are established in the next subsection.

3.2 Hierarchical Ordering of ε-Equivalent Factors

To obtain a meaningful hierarchical structure, we need a starting point or a level 0, which is simply the full model. On the other hand, if ε is arbitrarily large, all factors of the same dimension would be grouped together, resulting in a nearly trivial model. However, these properties alone are insufficient for creating a clear hierarchy between these extremes. A desired structure would resemble the one depicted in Fig. 2. Each level in the hierarchy is represented by a line on the left side of the figure. If we were to cut the figure horizontally at this point, all connected subtrees would form a group, while the remaining factors would stay separate. Our goal is to maintain the property that once multiple factors are grouped together at a lower level, they should also stay together at higher levels. This property is known as structure preservation. However, preserving the structure is not trivial due to the lack of transitivity in d_∞.

Thus, we impose a hierarchical pre-ordering on the set of factors based on pairwise ε-equivalence, determined by the minimal deviation under 1DEED d_∞. This unique ordering forms the basis for HACP described in Section 3.3. Smaller d_∞ values correspond to higher indistinguishability, and the hierarchical construction ensures that higher-level aggregations preserve the nested structure of previously merged subgroups. This can be seen as an agglomerative clustering algorithm based on 1DEED with complete linkage within maximal deviation (where only completely pairwise ε-equivalent groups are merged) with complexity $O(m^3)$, which is especially in comparison to the complexity of ACP negligible. The grouping step of ε-ACP is omitted instead, which is applied per choice of ε.

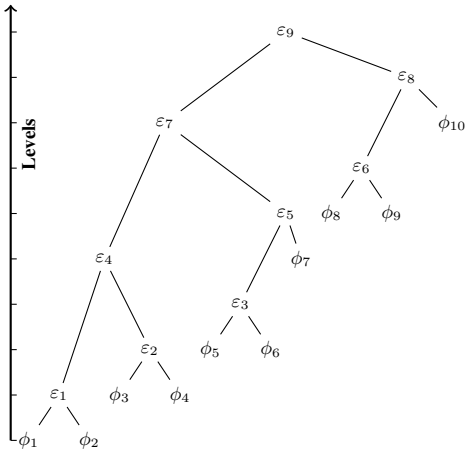


Figure 2. Exemplary visualisation of a factor ordering with increasing ε. This information is easily stored in the list $\mathcal{L}$ and is easily readable from the matrix $\tilde{\Lambda}$. The ε_i are ordered by size (ε_1 being the smallest value of them). Note that a root ε is always the maximal ε-distance of all pairwise factor comparisons of all leafs. E.g., $\varepsilon_4 = \max\{\varepsilon_{1,2}, \varepsilon_{1,3}, \varepsilon_{1,4}, \varepsilon_{2,3}, \varepsilon_{2,4}, \varepsilon_{3,4}\}$.

Factor	ϕ_1	ϕ_2	$\cdots$	$\cdots$	ϕ_{m-1}	ϕ_m
ϕ_1	0	$\varepsilon_{1,2}$	$\varepsilon_{1,3}$	$\cdots$	$\varepsilon_{1,m-1}$	$\varepsilon_{1,m}$
ϕ_2	0	0	$\varepsilon_{2,3}$	$\cdots$	$\varepsilon_{2,m-1}$	$\varepsilon_{2,m}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\ddots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\ddots$	$\vdots$
ϕ_{m-1}	0	0	0	0	0	$\varepsilon_{m-1,m}$
ϕ_m	0	0	0	0	0	0

Table 1. Upper triangular matrix $\Lambda = (\Lambda_{ij})_{1 \le i,j \le m}$ illustrating the matrix output of Alg. 1, Phase I. The entries are defined as $\Lambda_{ij} = \varepsilon_{i,j} := d_\infty(\phi_i, \phi_j)$ for $1 \le i < j \le m$, and $\Lambda_{ij} = 0$ otherwise.

To determine this ordering, we follow a two-phase procedure as described in Alg. 1 with an FG as input. In Phase I, the algorithm creates a matrix Λ as shown in Table 1. Its cell entries are $\Lambda_{ij} := \varepsilon_{i,j} := d_\infty(\phi_i, \phi_j)$ for $1 \le i < j \le m$, where $m = |\mathbf{\Phi}|$ is the number of factors in the FG. The symmetric property of d_∞ allows for filling the matrix with zeros, thus forming an upper triangular matrix. Next, we examine Phase II of the algorithm in more detail, which performs a hierarchical ordering of ε-equivalent group selections across all factors with structural compatibility.

The algorithm iteratively chooses ε values from Λ that allow (groups of) factors that are pairwise ε-equivalent to be grouped. The algorithm runs for $m-1$ iterations as there are m factors to merge, meaning there are $m-1$ hierarchical levels at the end. The outputs are an ordered vector $\boldsymbol{\varepsilon}$ of length $m-1$ of increasing ε values as well as an ordered nested list of lists $\mathcal{L}$ containing a nested grouping of indices according to the ε values and their hierarchy level (with an index shift of m for easier identification compared to the indices identifying factors). Specifically, the algorithm picks the next two (groups of) factors to merge by selecting the minimal entry $\varepsilon_{i',j'}$ in Λ, which is then stored in $\boldsymbol{\varepsilon}$ at the current level.

Before dealing with $\mathcal{L}$, let us consider how Λ is updated: Since both (groups of) factors are now considered as a single group, their respective rows in Λ need to be merged by keeping the maximum of the two ε values in each column, which ensures that if an entry of this row is picked in another iteration, all factors are ε-equivalent given this larger ε. To avoid resizing Λ, there is a set of active indices, and merging (groups of) factors removes the second index, essentially deactivating the row. The entries of the row of the first index are then updated to the maximum value.

Algorithm 1 Hierarchical Ordering of ε-Equivalent Groups

Input: An FG $M = (\boldsymbol{R} \cup \mathbf{\Phi}, \boldsymbol{E})$ with $m = |\mathbf{\Phi}|$ and $\mathbf{\Phi} \subset \mathbb{R}_{>0}^{n \times m}$, and structural compatibility across all factors.
Output: An ordered nested list $\mathcal{L}$ of lists, an ordered vector $\boldsymbol{\varepsilon} = (\varepsilon_1 \ldots, \varepsilon_{m-1})$ with $\varepsilon_i < \varepsilon_{i+1}$.

▷ Phase I: Generate upper triangular Matrix Λ
Initialise $\Lambda \in \mathbb{R}^{m \times m}$ with zeros
for $i = 1$ to $m-1$ **do**
  for $j = i+1$ to m **do**
    Compute $d_\infty(\phi_i, \phi_j) =: \varepsilon_{i,j}$
    Store result in Λ_{ij}
Save Λ
▷ Phase II: Generate ordered list $\mathcal{L}$ and vector $\boldsymbol{\varepsilon}$
Initialise empty list $\mathcal{L} \leftarrow [\,]$
Initialise vector $\boldsymbol{\varepsilon} = (\varepsilon_1 \ldots, \varepsilon_{m-1})$ with zeros
Initialise active index set $\mathcal{A} \leftarrow \{1, \ldots, m\}$
Initialise active matrix $\tilde{\Lambda} := \Lambda$
for $\ell = 1$ to $m-1$ **do**
  Find $(i', j') = \arg\min\{\tilde{\Lambda}_{ij} \mid i < j,\ i, j \in \mathcal{A}\}$
  Save $\varepsilon_l := \tilde{\Lambda}_{i'j'}$
  if i' appears in direct parent group $G_p \in \mathcal{L}$ **then**
    if j' appears in direct parent group $G'_p \in \mathcal{L}$ **then**
      Update G_p and G'_p as one new group $[G_p, G'_p, \ell + m]$
    else
      Update G_p as $G_p = [G_p, j', \ell + m]$
  else if j' appears in direct parent group $G_p \in \mathcal{L}$ **then**
    Update G_p as $G_p = [G_p, i', \ell + m]$
  else ▷ Neither i' nor j' appears in any parent group
    Append $[[i', j', \ell + m]]$ to $\mathcal{L}$
  for $k \in \mathcal{A} \setminus \{i', j'\}$ **do**
    if $k > j'$ **then**
      $\tilde{\Lambda}_{i'k} \leftarrow \max\{\tilde{\Lambda}_{i'k}, \tilde{\Lambda}_{j'k}\}$
    else if $k < i'$ **then**
      $\tilde{\Lambda}_{ki'} \leftarrow \max\{\tilde{\Lambda}_{ki'}, \tilde{\Lambda}_{kj'}\}$
    else if $i' < k < j'$ **then**
      $\tilde{\Lambda}_{i'k} \leftarrow \max\{\tilde{\Lambda}_{i'k}, \tilde{\Lambda}_{kj'}\}$
  Remove j' from active set: $\mathcal{A} \leftarrow \mathcal{A} \setminus \{j'\}$

Regarding $\mathcal{L}$, a new entry is formed, which is essentially a list of a 3-tuple $l = [e_{i'}, e_{j'}, h]$: one element $e_{i'}$ for the first (group of) factor(s), one element $e_{j'}$ for the second (group of) factor(s), and the last element h being the current hierarchy level shifted by m. If i' or j' identify a single factor, then $e_{i'}$ or $e_{j'}$ store the index identifying the factor. If i' or j' identify a group of factors, then there already exists an entry l' in $\mathcal{L}$ for it from a previous merging, which is removed from $\mathcal{L}$ and then stored in $e_{i'}$ or $e_{j'}$. Next, we look at an example.

Example 3. *Let $M = (\boldsymbol{R} \cup \mathbf{\Phi}, \boldsymbol{E})$ be an FG with $m = |\mathbf{\Phi}| = 10$ factors, all of identical dimension. Assume distances between factors leading to a hierarchy corresponding to Fig. 2, that is, factors ϕ_1 and ϕ_2 have the smallest distance among all factors, ϕ_3 and ϕ_4 the next smallest distance, followed by ϕ_5 and ϕ_6, after which the first two pairings have the smallest distance, and so on.*

During the first iteration of Phase II in Alg. 1, $\varepsilon_{1,2}$ is minimal in Λ. Therefore, an entry in $\mathcal{L}$ is created with $[1, 2, 11]$, containing the two indices $1, 2$ identifying the factors and the current hierarchy level 1 shifted by $m = 10$. The matrix update looks as follows:

Factor	ϕ_2	ϕ_3	ϕ_4	ϕ_5	$\cdots$
ϕ_1	$\varepsilon_{1,2}$	$\max_{i=1,2} \varepsilon_{i,3}$	$\max_{i=1,2} \varepsilon_{i,4}$	$\max_{i=1,2} \varepsilon_{i,5}$	$\cdots$
~~ϕ_2~~	~~0~~	~~$\varepsilon_{2,3}$~~	~~$\varepsilon_{2,4}$~~	~~$\varepsilon_{2,5}$~~	~~$\cdots$~~
ϕ_3	0	0	$\varepsilon_{3,4}$	$\varepsilon_{3,5}$	$\cdots$
ϕ_4	0	0	0	$\varepsilon_{4,5}$	$\cdots$
$\vdots$	0	0	0	0	$\cdots$

The row of index 1 *is updated to the maximum value of the two entries of the rows* $1, 2$*. The row of index* 2 *is deactivated, which is depicted by the crossed out line.*

In the next iteration, $\varepsilon_{3,4}$ *is minimal in* $\tilde{\Lambda}$*, which means adding an entry* $[3, 4, 12]$ *to* $\mathcal{L}$*, with the matrix update being the following:*

Factor	ϕ_2	ϕ_3	ϕ_4	ϕ_5	$\cdots$
ϕ_1	$\varepsilon_{1,2}$	$\max_{\substack{i=1,2\\j=3,4}} \varepsilon_{i,j}$	$\max_{i=1,2} \varepsilon_{i,4}$	$\max_{i=1,2} \varepsilon_{i,5}$	$\cdots$
~~ϕ_2~~	~~0~~	~~$\varepsilon_{2,3}$~~	~~$\varepsilon_{2,4}$~~	~~$\varepsilon_{2,5}$~~	~~$\cdots$~~
ϕ_3	0	0	$\varepsilon_{3,4}$	$\max_{i=3,4} \varepsilon_{i,5}$	$\cdots$
~~ϕ_4~~	~~0~~	~~0~~	~~0~~	~~$\varepsilon_{4,5}$~~	~~$\cdots$~~
$\vdots$	0	0	0	0	$\cdots$

When $\tilde{\Lambda}_{13}$ *is the next value to choose, both indices identify groups of factors. As such, their entries in* $\mathcal{L}$*, namely,* $[1, 2, 11]$ *and* $[3, 4, 12]$*, are replaced by an entry* $[[1, 2, 11], [3, 4, 12], 14]$*. At the end, the output of Alg. 1 looks as follows:*

$$\mathcal{L} = [\,[[[1, 2, 11], [3, 4, 12], 14], [[5, 6, 13], 7, 15], 17], \\ [[8, 9, 16], 10, 18], 19\,]$$

$$\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_9) \text{ with } \quad \varepsilon_1 = \min_{\substack{i,j=1,\ldots,10\\ i<j}} \{\varepsilon_{i,j}\} = \Lambda_{12},$$

$$\varepsilon_2 = \min_{\substack{i,j=1,3,\ldots,10\\ i<j}} \{\varepsilon_{i,j}\} = \tilde{\Lambda}_{34}, \quad \varepsilon_3 = \min_{\substack{i,j=1,3,5,\ldots,10\\ i<j}} \{\varepsilon_{i,j}\} = \tilde{\Lambda}_{56},$$

$$\varepsilon_4 = \min_{\substack{i,j=1,3,5,7,\ldots,10\\ i<j}} \{\varepsilon_{i,j}\} = \tilde{\Lambda}_{13}, \quad \ldots$$

This results in a total hierarchy of maximal 10 *different levels (and models). Appendix D shows an overview of the group sizes in the hierarchy, illustrating the compression possible with increasing* ε*.*

Thus, in the output, each ε_i corresponds to an increasingly coarse partitioning, reflecting group memberships under growing tolerance thresholds for higher levels. Selecting a specific ε_i implies fixing a hierarchical level i, which determines the groupings from $\mathcal{L}$. Running Alg. 1 is rather efficient, depending on the number of factors only and needing to compute pairwise distances only once.

3.3 Hierarchical Advanced Colour Passing Algorithm

HACP provides a hyperparameter-free hierarchical approach to lifted model construction. It uses the output of Alg. 1 to determine for a given level which groups get the same colour assigned, which is then the input to standard ACP, which runs independent of ε. Specifically, HACP proceeds in three phases, loading groups, running ACP, and updating potentials. Alg. 2 shows an overview, which is specified for a given hierarchy level i for the sake of brevity but could be easily extended to build parametric models for all levels of the hierarchy. It takes an FG, an index i, and the output of Alg. 1 for the FG as input.

Phase I provides a systematic procedure for forming the groups of factors in the input FG given the nested list $\mathcal{L}$ of the output of Alg. 1.

Algorithm 2 Hierarchical Advanced Colour Passing

Input: An FG $M = (\boldsymbol{R} \cup \boldsymbol{\Phi}, \boldsymbol{E})$, an index $i \in \{1, \ldots, m-1\}$, and the outcome of Alg. 1 run on M.
Output: A lifted representation M', encoded as a parametric FG, which is approximately equivalent to M.

▷ Phase I: Load groups of pairwise ε-equivalent factors for ε_i
Let $\mathcal{L}$ be the current list of candidate groups.
for $k = m + i$ **to** $m + 1$ **do**
  if k occurs in any group in $\mathcal{L}$ **then**
    Load global parent group $G_p(k) \in \mathcal{L}$
    Store group of ε-equivalent factors: $\boldsymbol{G}_\Phi(k) := \{\phi_j \mid j \in G_p(k),\ j < m\} \in \boldsymbol{G}$
    Update $\mathcal{L} \leftarrow \mathcal{L} \setminus G_p(k)$
for each factor $\phi_j \in \boldsymbol{\Phi} \setminus \left(\bigcup_{k=m+1}^{m+i} G_\Phi(k)\right)$ **do**
  Store group $\boldsymbol{G}_\Phi(j) := \{\phi_j\} \in \boldsymbol{G}$
▷ Phase II: Assign colours to factors and run ACP
for each group $\boldsymbol{G}_j \in \boldsymbol{G}$ **do**
  for each factor $\phi_i \in \boldsymbol{G}_j$ **do**
    $\phi_i.colour \leftarrow j$
$M' \leftarrow$ Call ACP on M and $\boldsymbol{G}$ using the assigned colours
▷ Phase III: Update potentials
for each parametric factor $g \in M'$ **do**
  $\boldsymbol{G}_j \leftarrow$ Collect all factors ϕ_i that were merged into g
  $\phi^*(\boldsymbol{r}) \leftarrow \frac{1}{|\boldsymbol{G}_j|} \sum_{\phi_i \in \boldsymbol{G}_j} \phi_i(\boldsymbol{r})$ for all assignments $\boldsymbol{r}$
  for each factor $\phi_i \in \boldsymbol{G}_j$ **do**
    $\phi_i \leftarrow \phi^*$

For instance, consider Example 3 and level 4 with ε_4. The grouping induced at this level is:

$$\boldsymbol{G} = \{\boldsymbol{G}_1 = \{\phi_1, \phi_2, \phi_3, \phi_4\}, \boldsymbol{G}_2 = \{\phi_5, \phi_6\}, \\ \boldsymbol{G}_3 = \{\phi_7\}, \boldsymbol{G}_4 = \{\phi_8\}, \boldsymbol{G}_5 = \{\phi_9\}, \boldsymbol{G}_6 = \{\phi_{10}\}\}.$$

Subsequently, Phase II applies the standard ACP with colours assigned to each identified group, which returns a parametric FG with one parametric factor for each group of factors deemed equivalent. In Phase III, the potentials of each parametric factor are replaced by the arithmetic mean of the group of equivalent factors. That is, for each group of potentials $\{\varphi_1, \ldots, \varphi_k\}$, a new potential is computed as $\varphi^* = \arg\min_{\hat{\varphi}} \sum_{i=1}^{k} (\varphi_i - \hat{\varphi})^2$. This minimises intra-group variance and yields an optimal compressed representation. The constructed factor $\boldsymbol{\phi}^* = (\varphi_1^*, \ldots, \varphi_n^*)$ is, by design, also pairwise ε-equivalent to all original factors from its generating group.

4 Hierarchical Bounds: Compression vs. Accuracy

A crucial property of the ε-ACP algorithm is its induced bound on the change in probabilistic queries based on its hyperparameter ε. HACP uses predefined groups (selected via Algorithm 2) and still allows choosing a compressed model of reduced complexity, while preserving the same asymptotic bounds as its predecessor, ε-ACP, to ensure consistent and reliable performance (accuracy). The following subsections explore the properties of HACP, examining how to control this information to apply the algorithm with specific accuracy.

4.1 Asymptotic Properties

A desirable property of the HACP algorithm is that it preserves the same boundaries on the change in probabilistic queries as the ε-ACP

algorithm. After running ε-ACP, the deviation-wise worst-case scenario for an assignment is bounded. Notably, HACP relies on the mean values of the potentials of pairwise ε-equivalent factors. To quantify the difference in probabilistic queries between the original FG and the hierarchical processed FG after applying the HACP algorithm, we use the symmetric distance measure between two distributions P_M and $P_{M'}$ introduced by Chan and Darwiche [4], which effectively bounds the maximal deviation of any assignment $\boldsymbol{r}$:

$$D_{CD}(P_M, P_{M'}) := \ln \max_{\boldsymbol{r}} \frac{P_{M'}(\boldsymbol{r})}{P_M(\boldsymbol{r})} - \ln \min_{\boldsymbol{r}} \frac{P_{M'}(\boldsymbol{r})}{P_M(\boldsymbol{r})}. \quad (4)$$

For ε-ACP, the following asymptotic bound has been proven [14]:

Theorem 3 (Luttermann et al. [14])**.** *Let $M = (\boldsymbol{R} \cup \boldsymbol{\Phi}, \boldsymbol{E})$ be an FG and let M' be the output of ε-ACP 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*

$$D_{CD}(P_M, P_{M'}) \leq \ln \left(\frac{\left(1 + \frac{m-1}{m}\varepsilon\right)\left(1+\varepsilon\right)}{1 + \frac{1}{m}\varepsilon} \right)^m \quad (5)$$

$$< \ln \left(1+\varepsilon\right)^{2m} < \ln \left(\frac{1+\varepsilon}{1-\varepsilon} \right)^m, \quad (6)$$

where the bound given in Eq. (5) *is optimal (sharp).*

We next show that this bound also applies to the HACP algorithm.

Proposition 4. *Theorem 3 holds the same way for M' being the output of the HACP algorithm (Alg. 2).*

Proof. The core components of the ε-ACP algorithm and its hierarchical counterpart HACP (Alg. 2) are identical, aside from enforcing predefined group structures to guarantee a hierarchical structure. Therefore, the proof can be conducted in the same manner as the original proof [14, App. A]. The same proposed example can be used to hit the bound of Eq. (5), showing its optimality. □

4.2 Compression versus Accuracy

We continue to give a finer analysis of theoretical properties entailed by HACP regarding the trade-off between compression and accuracy. All upcoming results hold for both ε-ACP and HACP.

Theorem 5. *The maximal absolute deviation between any initial probability $p = P_M(r \mid \boldsymbol{e})$ of r given $\boldsymbol{e}$ in model M and the probability $p' = P_{M'}(r \mid \boldsymbol{e})$ in the modified model M' resulting from running HACP (Alg. 2) or ε-ACP on M can be bounded by*

$$p_{\max\Delta} := \max_{\text{for any } r \mid \boldsymbol{e}} |p - p'| \leq \frac{\sqrt{e^d} - 1}{\sqrt{e^d} + 1} \text{ with } d = D_{CD}(P_M, P'_M).$$

Proof Sketch. From Chan and Darwiche [4], we use

$$\frac{pe^{-d}}{p(e^{-d}-1)+1} \leq p' = P_{M'}(r \mid \boldsymbol{e}) \leq \frac{pe^{d}}{p(e^{d}-1)+1} \quad (7)$$

to define a upper bound function $f_{\max\Delta}$ for $p_{\max\Delta}$, which is symmetric for $p' = 1/2$ (see Fig. 3)

$$f_{\max\Delta}(p) := \max(f_{\text{upper}}(p), f_{\text{lower}}(p)) \text{ for } p \in [0,1]$$

$$\text{with } f_{\text{upper}}(p) := \frac{pe^d}{p(e^d-1)+1} - p = \frac{p(1-p)(e^d-1)}{p(e^d-1)+1}$$

$$\text{and } f_{\text{lower}}(p) := p - \frac{pe^{-d}}{p(e^{-d}-1)+1} = \frac{p(1-p)(1-e^{-d})}{p(e^{-d}-1)+1},$$

and calculate its extrema by first- and second-order conditions. □

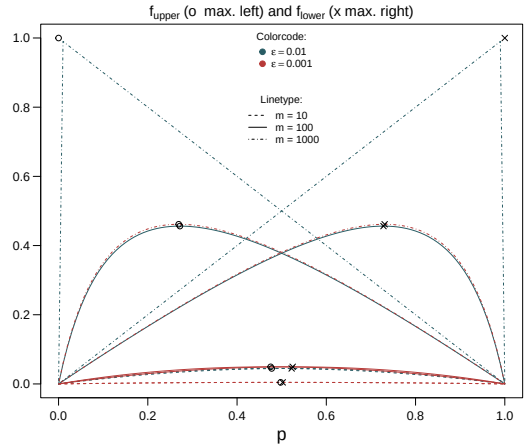


Figure 3. Showing f_{upper} and f_{lower} over $[0, 1]$ with d_2 values from Corollary 6 to use an upper estimate for $p_{\max\Delta}$ by bounding it from above.

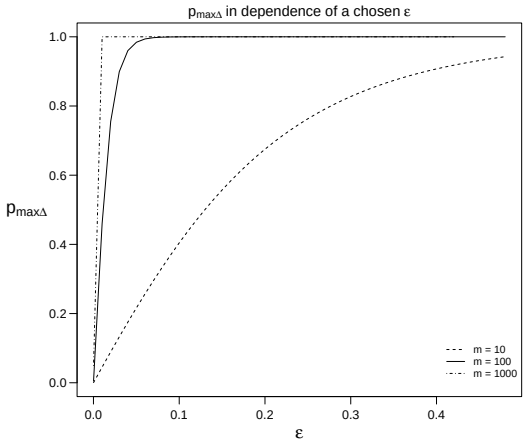


Figure 4. The bound on $p_{\max\Delta}$ depending on the choice of ε for different amounts of factors m.

Since Theorem 5 lets us compute $D_{CD}(P_M, P_{M'})$ efficiently -— without inspecting the full factor graph -— it has direct practical impact. Since $\frac{\sqrt{e^d}-1}{\sqrt{e^d}+1} = \tanh\left(\frac{d}{4}\right) \leq 1$ is monotonically strictly increasing in d, the results of Theorem 3 can be substituted into its formula, while the directions of the inequalities are obtained.

Corollary 6. *With previous notations, the change in any probabilistic query in an initial model M and a modified model M' obtained by running HACP (Alg. 2) or ε-ACP is bounded by*

$$\begin{aligned} p_{\max\Delta} &\leq \frac{\sqrt{e^{d_1}}-1}{\sqrt{e^{d_1}}+1} \text{ with } d_1 = D_{CD}(P_M, P'_M) \\ &\leq \frac{\sqrt{e^{d_2}}-1}{\sqrt{e^{d_2}}+1} \text{ with } d_2 = \ln\left(\frac{\left(1 + \frac{m-1}{m}\varepsilon\right)\left(1+\varepsilon\right)}{1 + \frac{1}{m}\varepsilon} \right)^m \\ &\leq \frac{\sqrt{e^{d_3}}-1}{\sqrt{e^{d_3}}+1} \text{ with } d_3 = \ln\left(1+\varepsilon\right)^{2m} \\ &\leq \frac{\sqrt{e^{d_4}}-1}{\sqrt{e^{d_4}}+1} \text{ with } d_4 = \ln\left(\frac{1+\varepsilon}{1-\varepsilon} \right)^m. \end{aligned}$$

This implies that for a given $\varepsilon > 0$, we can determine the maximum deviation of $p_{\max\Delta}$ (see Fig. 4). However, it is also essential to consider the reverse perspective to gain insight into the overall diversity and structural complexity of the FG. Therefore, we investigate the following question: What choice of ε guarantees that our probabilities remain within a specified distance, i.e., $p_{\max\Delta} \leq p^*_\Delta$?

This question is addressed in the subsequent theorem by reversing the preceding inequalities and solving for ε.

Theorem 7. *For any given $p^*_\Delta \in (0, \frac{1}{2}]$, the output of HACP guarantees for any $\varepsilon \in (0, 1)$, which is smaller or equal to*

$$\varepsilon_1 = -\frac{1 + \frac{m-1}{m} - \frac{1}{m}\sqrt[m]{e^d}}{2\,\frac{m-1}{m}} + \sqrt{\left(-\frac{1 + \frac{m-1}{m} - \frac{1}{m}\sqrt[m]{e^d}}{2\,\frac{m-1}{m}}\right)^2 - \frac{1 - \sqrt[m]{e^d}}{\frac{m-1}{m}}}$$

with $d = \ln\left(\frac{p^*_\Delta + 1}{1 - p^*_\Delta}\right)^2$ the bound $p_{\max\Delta} \leq p^*_\Delta$.

Proof. Using Theorem 5, we get for $d = D_{CD}(P_M, P'_M)$:

$$p_{\max\Delta} \leq \frac{\sqrt{e^d} - 1}{\sqrt{e^d} + 1} =: p^*_\Delta$$
$$\Leftrightarrow \ln\left(\frac{p^*_\Delta + 1}{1 - p^*_\Delta}\right)^2 = d = D_{CD}(P_M, P'_M).$$

Additionally, we use Eq. (5) from Theorem 3 and the corresponding version for HACP (Proposition 4) and solve the inequality for ε, when it reaches equality:

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

□

This means that we can bound the maximal deviation p^*_Δ, which is tight [14], of HACP and ε-ACP, respectively, by calculating $\varepsilon_1(p^*_\Delta)$ before we run it. However, these bounds are mostly of theoretical nature and the deviations are rarely encountered in reality [14].

Note 2. *Choosing, for instance, a 10 times larger ε has pretty much the same effect as choosing 10 times the number of factors m on the bound on the change in probabilistic queries (cf. Fig. 3).*

Thus, this subsection lays the groundwork for leveraging the supplementary insights from Alg. 1 to develop a holistic understanding of the structural and computational complexity of the FG. Depending on sensitivity to ε, $p_{\max\Delta}/p^*_\Delta$, and m, one can assess the implications of specific parameter choices on the effect of compressing a given FG, or alternatively, prioritise model fidelity. The HACP algorithm enables such assessments across multiple structural levels.

5 Discussion

Related work. Lifted inference exploits the indistinguishability of objects in probabilistic (relational) models, enabling more efficient query answering (marginals of randvars given observations) with exact results [16]. First introduced by Poole [17], parametric FGs, which combine relational logic and probabilistic modelling, and lifted variable elimination enable lifted probabilistic inference to speed up query answering by exploiting the indistinguishability of objects. Over the past years, lifted variable elimination has continuously been refined by many researchers to reach its current form [3, 5, 6, 11, 15, 19]. To construct a lifted (i.e., first-order) representation such as a parametric FG, the ACP algorithm [13], which generalises the CompressFactorGraph algorithm [1, 10], is the current state of the art. ACP runs a colour passing procedure to detect symmetric subgraphs in a probabilistic graphical model, similar to the Weisfeiler-Leman algorithm [23], which is a well-known algorithm to test for graph isomorphism. It groups symmetric subgraphs and exploits exponentiation during probabilistic inference. While ACP is able to construct a parametric FG entailing equivalent semantics as a given propositional model, it requires potentials of factors to exactly match before grouping them. In practice, however, potentials are often estimates and hence might slightly differ even for indistinguishable objects. To account for small deviations between potentials, the ε-ACP algorithm [14] has been introduced, generalising ACP by introducing a hyperparameter ε that controls the trade-off between the exactness and the compactness of the resulting lifted representation. While the original formulation assumes factors with identical dimensions and range structures, the general concept naturally extends to heterogeneous dimensions. In such cases, care must be taken to preserve structural consistency to ensure that existing symmetries remain valid when reasoning across dimensions.

Pre-ordering means pre-analysing. Our hierarchical algorithm (Alg. 1) imposes a predetermined nesting structure on the factor graph before any colour passing procedure, enabling a priori application-specific level selection. By specifying levels before applying Alg. 2 to the adjusted graph, one can predict and implicitly control the resulting complexity of ε-equivalent group structure and thereby enhance interpretability. In contrast to ε-ACP, which may produce ε-equivalent groupings that lack consistent nesting across runs or parameter settings, our hierarchical approach (HACP) ensures structural coherence and comparability across instances. Moreover, the explicit composition of each level can be monitored to trace modification impacts throughout the hierarchy. Large d_∞ values indicate low symmetry and are generally unsuitable for approximating the FG in most applications. Conversely, many small d_∞ values close to each other suggest high potential for similar factor structures. Such an analysis is not feasible with ε-ACP, which requires an a priori choice of ε without any guarantees of identifying symmetries.

Trade-off: Compression versus Accuracy. ε-ACP and HACP inherit deviation bounds from Luttermann et al. [14], yielding identical sharp bounds and dependencies for any choice of ε. In practice, grouping composition controls magnitude and sign of probabilistic deviations in downstream queries: As the hierarchy level (or ε) increases, theoretical bounds grow, yet actual query deviations may fluctuate based on group aggregations. Crucially, our hierarchical bounds facilitate pre-specification of maximal permissible ε values and corresponding levels. Rather than relying on the generally intractable D_{CD} for approximated models [4], one can derive $p_{\max\Delta}$ for a given ε, or select an admissible level that guarantees both desired compression and sufficient accuracy.

HACP operates on a more restricted space of ε-equivalent groupings than ε-ACP due to its forced hierarchy, enabling interpretability and structured level comparisons. Thus, HACP may show slightly higher average deviations without systematic inferiority on individual assignments. Importantly, both algorithms retain identical worst-case deviation bounds by construction.

6 Conclusion

We introduce a novel framework for hierarchical lifting and model reconciliation in FGs. By presenting a more practical one-dimensional notion of ε-equivalent factors, we enable the identification of (possibly inexact) symmetries, the number and sizes of ε-equivalent groups and the resulting reduction of computational complexity, thereby allowing for lifted inference. Our theoretical analysis provides a solid foundation for understanding the structural properties of FGs. Crucially, the entire hierarchy is fixed prior to initiating colour passing or inference, ensuring structural consistency and enabling theoretical error bounds. This work provides a foundation for future advances in efficient and interpretable probabilistic inference.

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

[1] B. Ahmadi, K. Kersting, M. Mladenov, and S. Natarajan. Exploiting Symmetries for Scaling Loopy Belief Propagation and Relational Training. *Machine Learning*, 92(1):91–132, 2013.

[2] T. Braun and R. Möller. Lifted Junction Tree Algorithm. In *Proc. of the Thirty-Ninth German Conference on Artificial Intelligence (KI-2016)*, pages 30–42. Springer, 2016.

[3] T. Braun and R. Möller. Parameterised Queries and Lifted Query Answering. In *Proc. of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-2018)*, pages 4980–4986. IJCAI Organization, 2018.

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

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

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

[7] B. J. Frey, F. R. Kschischang, H.-A. Loeliger, and N. Wiberg. Factor Graphs and Algorithms. In *Proc. of the Thirty-Fifth Annual Allerton Conference on Communication, Control, and Computing*, pages 666–680. Allerton House, 1997.

[8] V. Gogate and P. Domingos. Probabilistic Theorem Proving. In *Proc. of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence (UAI-2011)*, pages 256–265. AUAI Press, 2011.

[9] M. Hartwig, R. Möller, and T. Braun. An Extended View on Lifting Gaussian Bayesian Networks. *Artificial Intelligence*, 330:104082, 2024.

[10] K. Kersting, B. Ahmadi, and S. Natarajan. Counting Belief Propagation. In *Proc. of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI-2009)*, pages 277–284. AUAI Press, 2009.

[11] J. Kisyński and D. Poole. Constraint Processing in Lifted Probabilistic Inference. In *Proc. of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI-2009)*, pages 293–302. AUAI Press, 2009.

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

[13] M. Luttermann, T. Braun, R. Möller, and M. Gehrke. Colour Passing Revisited: Lifted Model Construction with Commutative Factors. In *Proc. of the Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-2024)*, pages 20500–20507. AAAI Press, 2024.

[14] M. Luttermann, J. Speller, M. Gehrke, T. Braun, R. Möller, and M. Hartwig. Approximate Lifted Model Construction. In *Proc. of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI-2025)*, 2025. https://arxiv.org/abs/2504.20784.

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

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

[17] D. Poole. First-order Probabilistic Inference. In *Proc. of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-2003)*, pages 985–991. IJCAI Organization, 2003.

[18] M. Richardson and P. Domingos. Markov Logic Networks. *Machine Learning*, 62(1–2):107–136, 2006.

[19] N. Taghipour, D. Fierens, J. Davis, and H. Blockeel. Lifted Variable Elimination: Decoupling the Operators from the Constraint Language. *Journal of Artificial Intelligence Research*, 47(1):393–439, 2013.

[20] G. Van den Broeck. *Lifted Inference and Learning in Statistical Relational Models*. PhD thesis, KU Leuven, 2013.

[21] G. Van den Broeck and J. Davis. Conditioning in First-Order Knowledge Compilation and Lifted Probabilistic Inference. In *Proc. of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-2012)*, pages 1961–1967. AAAI Press, 2012.

[22] G. Van den Broeck, N. Taghipour, W. Meert, J. Davis, and L. De Raedt. Lifted Probabilistic Inference by First-order Knowledge Compilation. In *Proc. of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI-2011)*, pages 2178–2185. IJCAI Organization, 2011.

[23] B. Weisfeiler and A. 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 Counterexamples

Proposition 8. *The 1DEED is not a metric.*

Proof. The 1DEED is not a metric, because the Δ-inequality does not hold, which is exemplarily proven by this counterexample:

$$\phi_1 = \begin{pmatrix} 2 \\ 0.5 \end{pmatrix}, \phi_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \phi_3 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

with $d_\infty(\phi_1, \phi_2) + d_\infty(\phi_2, \phi_3) = 1 + 1 < 3 = d_\infty(\phi_1, \phi_3)$. □

Proposition 9. *ε-equivalency based on 1DEED lacks the transitivity property.*

Proof. Given a Boolean random variable Var, consider three factors ϕ_1, ϕ_2, ϕ_3 defined as follows:

Var	$\phi_1(Var)$	$\phi_2(Var)$	$\phi_3(Var)$
true	0.95	1.0	1.08
false	2.05	1.95	2.10

Using the 1DEED with $\varepsilon = 0.1$, we end up with $\phi_1 =_\varepsilon \phi_2$ and $\phi_2 =_\varepsilon \phi_3$, but $\phi_1 \neq_\varepsilon \phi_3$:

$$\phi_1(\text{true}) = 0.95 < 0.972 = (1-\varepsilon)1.08 = (1-\varepsilon)\phi_3(\text{true}).$$

□

B Detailed Proofs

Theorem 2. *Two vectors $\phi_1, \phi_2 \in \mathbb{R}^n_{>0}$ are ε-equivalent (Definition 3) if and only if $d_\infty(\phi_1, \phi_2) \leq \varepsilon$ holds.*

Proof. In mathematical terms, the claim can be summarised as $\phi_1 =_\varepsilon \phi_2 \Leftrightarrow d_\infty(\phi_1, \phi_2) \leq \varepsilon$, which we prove for any $\varepsilon > 0$:

$$\begin{aligned}
&\phi_1 =_\varepsilon \phi_2 \text{ for two factors } \phi_1, \phi_2 \in \mathbb{R}^n_{>0} \\
&\overset{\text{def.}}{\Leftrightarrow} \phi_1(k) \in [(1-\varepsilon)\phi_2(k), (1+\varepsilon)\phi_2(k)] \text{ and} \\
&\quad \phi_2(k) \in [(1-\varepsilon)\phi_1(k), (1+\varepsilon)\phi_1(k)] \text{ for } k = 1, \ldots, n \\
&\Leftrightarrow \phi_2(k) - \phi_2(k)\varepsilon \leq \phi_1(k) \leq \phi_2(k) + \varepsilon\phi_2(k) \text{ and} \\
&\quad \phi_1(k) - \phi_1(k)\varepsilon \leq \phi_2(k) \leq \phi_1(k) + \varepsilon\phi_1(k) \text{ for } k = 1, \ldots, n \\
&\Leftrightarrow -\phi_2(k)\varepsilon \leq \phi_1(k) - \phi_2(k) \leq \varepsilon\phi_2(k) \text{ and} \\
&\quad -\phi_1(k)\varepsilon \leq \phi_2(k) - \phi_1(k) \leq \varepsilon\phi_1(k) \text{ for } k = 1, \ldots, n \\
&\Leftrightarrow |\phi_1(k) - \phi_2(k)| \leq \varepsilon\phi_2(k) \text{ and} \\
&\quad |\phi_1(k) - \phi_2(k)| \leq \varepsilon\phi_1(k) \text{ for } k = 1, \ldots, n \\
&\Leftrightarrow \frac{|\phi_1(k) - \phi_2(k)|}{\phi_2(k)} \leq \varepsilon \text{ and} \\
&\quad \frac{|\phi_1(k) - \phi_2(k)|}{\phi_1(k)} \leq \varepsilon \text{ for } k = 1, \ldots, n \\
&\Leftrightarrow \frac{|\phi_1(k) - \phi_2(k)|}{\min\{\phi_1(k), \phi_2(k)\}} \leq \varepsilon \text{ for } k = 1, \ldots, n \\
&\Leftrightarrow \max_{k=1,\ldots,n} \left\{ \frac{|\phi_1(k) - \phi_2(k)|}{\min\{\phi_1(k), \phi_2(k)\}} \right\} \leq \varepsilon \\
&\overset{\text{def.}}{\Leftrightarrow} d_\infty(\phi_1, \phi_2) \leq \varepsilon
\end{aligned}$$

□

Theorem 5. *The maximal absolute deviation between any initial probability $p = P_M(r \mid \boldsymbol{e})$ of r given $\boldsymbol{e}$ in model M and the probability $p' = P_{M'}(r \mid \boldsymbol{e})$ in the modified model M' resulting from running HACP (Alg. 2) or ε-ACP on M can be bounded by*

$$p_{\max\Delta} := \max_{\text{for any } r \mid \boldsymbol{e}} |p - p'| \leq \frac{\sqrt{e^d} - 1}{\sqrt{e^d} + 1} \text{ with } d = D_{CD}(P_M, P'_M).$$

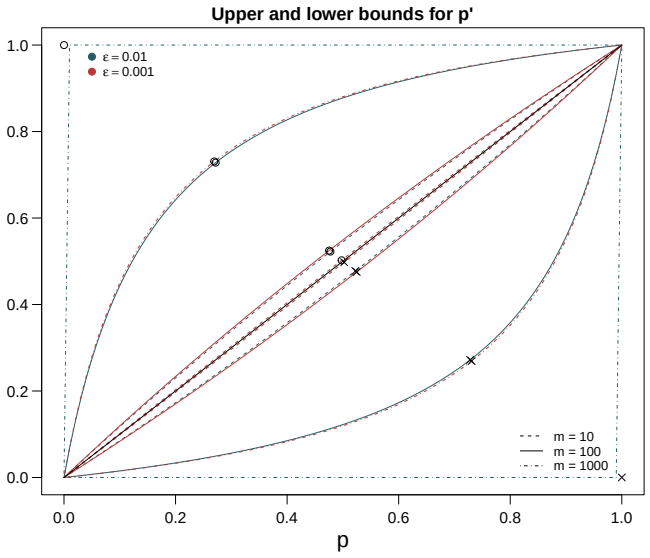


Figure 5. Bounds of Eq. (7) in comparison to p for different d values depending on m and ε (Eq. (5)). Circles/ crosses are the maximum distances from those bounds to the function p. Distances to function $f(p) = p$ are later also referred as $f = f_{\text{upper}}$ for $p \in [0, \frac{1}{2}]$ and $f = f_{\text{lower}}$ for $p \in (\frac{1}{2}, 1]$.

Proof. From Chan and Darwiche [4], we already know that

$$\frac{pe^{-d}}{p(e^{-d}-1)+1} \leq p' = P_{M'}(r \mid \boldsymbol{e}) \leq \frac{pe^{d}}{p(e^{d}-1)+1} \tag{8}$$

holds (see Fig. 5), where $p = P_M(r \mid \boldsymbol{e})$ is the probability of r given $\boldsymbol{e}$ in the original model M and $d = D_{CD}(P_M, P'_M)$ is the value of the distance measure introduced by Chan and Darwiche between P_M and P'_M. Hence, for any r given $\boldsymbol{e}$, in the worst case, we get

$$|p - p'| = \begin{cases} p' - p & \text{for } p \leq p' \\ p - p' & \text{for } p' < p \end{cases} = \begin{cases} \frac{pe^d}{p(e^d-1)+1} - p & \text{for } p \leq p' \\ p - \frac{pe^{-d}}{p(e^{-d}-1)+1} & \text{for } p' < p. \end{cases}$$

Becoming independent of p' guarantees one maximal bound for all possible queries and can be achieved using the maximum of both cases in $[0, 1]$ as an upper bound for $p_{\max\Delta}$, which is given by the following function $f_{\max\Delta}(p)$:

$$f_{\max\Delta}(p) := \max(f_{\text{upper}}(p), f_{\text{lower}}(p)) \text{ for } p \in [0, 1]$$

$$\text{with } f_{\text{upper}}(p) := \frac{pe^d}{p(e^d-1)+1} - p = \frac{p(1-p)(e^d-1)}{p(e^d-1)+1}$$

$$\text{and } f_{\text{lower}}(p) := p - \frac{pe^{-d}}{p(e^{-d}-1)+1} = \frac{p(1-p)(1-e^{-d})}{p(e^{-d}-1)+1}.$$

It is easy to see that $f_{\max\Delta}$ is a symmetric function around $p = 0.5$

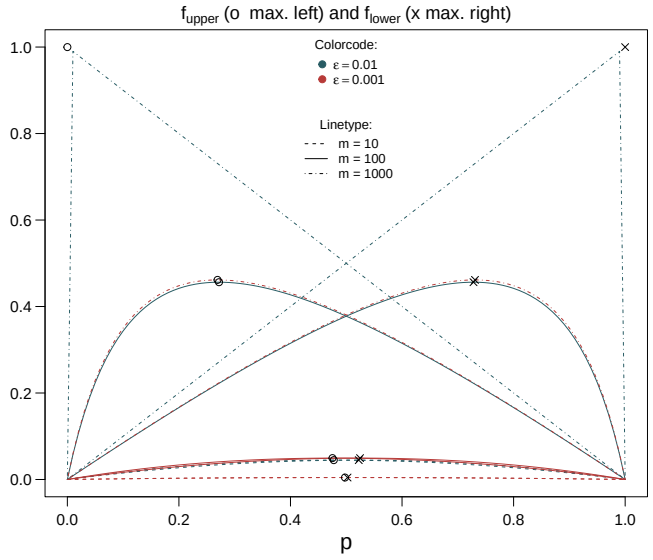


Figure 6. Showing the f_{upper} and f_{lower} functions over $[0, 1]$ with d_2 values from Corollary 6 to use an upper estimate for $p_{\max \Delta}$ by bounding it from above.

(see Fig. 6), because $f_{\text{upper}}(p) = f_{\text{lower}}(1-p)$ holds:

$$\begin{aligned}
f_{\text{lower}}(1-p) &= \frac{(1-p)p(1-e^{-d})}{(1-p)(e^{-d}-1)+1} \\
&= \frac{(1-p)p(e^{d}-1)e^{-d}}{(1-p)(e^{-d}-1)+1} \\
&= \frac{(1-p)p(e^{d}-1)}{((1-p)(e^{-d}-1)+1)e^{d}} \\
&= \frac{p(1-p)(e^{d}-1)}{(1-p)(1-e^{d})+e^{d}} \\
&= \frac{p(1-p)(e^{d}-1)}{1-p+pe^{d}} \\
&= \frac{p(1-p)(e^{d}-1)}{p(e^{d}-1)+1} \\
&= f_{\text{upper}}(p)
\end{aligned}$$

Now, the choice for p' to get the maximum of both functions is $p = 0.5$, while $f_{\text{upper}}(p)$ decreases for $p > 0.5$ and $f_{\text{lower}}(p)$ increases for $p < 0.5$ for $d \geq 0$. Therefore, we get

$$f_{\max \Delta}(p) := \begin{cases} f_{\text{upper}}(p) & \text{for } 0 \leq p \leq 0.5 \\ f_{\text{lower}}(p) & \text{for } 0.5 < p \leq 1 \end{cases} \tag{9}$$

This means that our search for the maximum deviation leads us to calculate the derivatives after p

$$\begin{aligned}
f'_{\text{upper}}(p) &= -\frac{(e^{d}-1)(p^2(e^{d}-1)+2p-1)}{(p(e^{-d}-1)+1)^2}, \\
f''_{\text{upper}}(p) &= \frac{-2e^{d}(e^{d}-1)}{(p(e^{d}-1)+1)^3}, \\
f'_{\text{lower}}(p) &= \frac{(1-e^{-d})(p^2(1-e^{-d})-2p+1)}{(p(e^{-d}-1)+1)^2}, \\
f''_{\text{lower}}(p) &= \frac{-2e^{d}(e^{d}-1)}{(p+e^{d}(1-p))^3},
\end{aligned}$$

and to consider initially the first-order conditions. For this purpose, we first obtain

$$\begin{aligned}
&f'_{\text{upper}}(p) = 0 \\
&\quad \Leftrightarrow p^2(e^{d}-1)+2p-1 = 0 \\
&\quad \Leftrightarrow p_{\text{upper }1/2} = -\frac{1}{e^{d}-1} \pm \sqrt{\left(\frac{-1}{e^{d}-1}\right)^2 + \frac{1}{e^{d}-1}} \\
&\quad \Leftrightarrow p_{\text{upper }1/2} = -\frac{1}{e^{d}-1} \pm \frac{\sqrt{e^{d}}}{e^{d}-1} = \frac{-1 \pm \sqrt{e^{d}}}{e^{d}-1}.
\end{aligned}$$

As $p_{\text{upper }2}$ is smaller than zero, the potential maximum in $[0, 1]$ is at

$$p_1 = p_{\text{upper }1} = \frac{\sqrt{e^{d}}-1}{e^{d}-1} = \frac{1}{\sqrt{e^{d}}+1}.$$

Analogously, we find a potential maximum for f'_{lower}:

$$\begin{aligned}
&f'_{\text{lower}}(p) = 0 \\
&\quad \Leftrightarrow p^2(1-e^{-d})-2p+1 = 0 \\
&\quad \Leftrightarrow p_{\text{lower }1/2} = \frac{1}{1-e^{-d}} \pm \sqrt{\left(\frac{1}{1-e^{-d}}\right)^2 - \frac{1}{1-e^{-d}}} \\
&\quad \Leftrightarrow p_{\text{lower }1/2} = \frac{1}{1-e^{-d}} \pm \frac{\sqrt{e^{-d}}}{1-e^{-d}} = \frac{1 \pm \sqrt{e^{-d}}}{1-e^{-d}}
\end{aligned}$$

As $p_{\text{lower }1}$ is larger than one, the possible maximum in $[0, 1]$ is at

$$p_2 = p_{\text{lower }2} = \frac{1-\sqrt{e^{-d}}}{1-e^{-d}} = \frac{1}{\sqrt{e^{-d}}+1} = \frac{\sqrt{e^{d}}}{\sqrt{e^{d}}+1}$$

and the second-order conditions can also be easily checked: Since $e^{d} - 1 > 0$ and $1 - p > 0$, we get $f''_{\text{upper}}(p_1) < 0$ and $f''_{\text{lower}}(p_2) < 0$ and can conclude that p_1 is a local maximum of f_{upper} and p_2 is a local maximum of f_{lower}. The boundary values 0 and 1 are no possible points for a global maximum, because both functions f_{upper} and f_{lower} take on the value 0 there. Therefore, the only possible extreme point for the global maximum for f_{upper} is $p_1 = \frac{1}{\sqrt{e^{d}}+1}$ and f_{lower} is $p_2 = \frac{\sqrt{e^{d}}}{\sqrt{e^{d}}+1}$. Note that p_1 and p_2 are symmetrically distanced to $p = 1/2$.

Both reach exactly the same maximal deviation:

$$\begin{aligned} f_{\text{upper}}(p_1) &= \frac{\frac{1}{\sqrt{e^d}+1}(1-\frac{1}{\sqrt{e^d}+1})(e^d-1)}{\frac{1}{\sqrt{e^d}+1}(e^d-1)+1} \\ &= \frac{\frac{1}{\sqrt{e^d}+1}\cdot\frac{\sqrt{e^d}}{\sqrt{e^d}+1}\cdot(\sqrt{e^d}+1)\cdot(\sqrt{e^d}-1)}{\frac{e^d-1+\sqrt{e^d}+1}{\sqrt{e^d}+1}} \\ &= \frac{\sqrt{e^d}(\sqrt{e^d}-1)}{e^d+\sqrt{e^d}} = \frac{\sqrt{e^d}-1}{\sqrt{e^d}+1} \end{aligned}$$

and the same holds for

$$\begin{aligned} f_{\text{lower}}(p_2) &= \frac{\frac{\sqrt{e^d}}{\sqrt{e^d}+1}(1-\frac{\sqrt{e^d}}{\sqrt{e^d}+1})(1-e^{-d})}{\frac{\sqrt{e^d}}{\sqrt{e^d}+1}(e^{-d}-1)+1} \\ &= \frac{\frac{\sqrt{e^d}}{\sqrt{e^d}+1}\cdot\frac{1}{\sqrt{e^d}+1}\cdot(1-e^{-d})}{\frac{\sqrt{e^d}}{\sqrt{e^d}+1}\cdot(\sqrt{e^{-d}}-\sqrt{e^d}+\sqrt{e^d}+1)} \\ &= \frac{\sqrt{e^d}(1-\sqrt{e^{-d}})(1+\sqrt{e^{-d}})}{(\sqrt{e^{-d}}+1)(\sqrt{e^d}+1)} \\ &= \frac{\sqrt{e^d}-1}{\sqrt{e^d}+1}. \end{aligned}$$

This means:

$$\begin{aligned} p_{\max\Delta} &= \max_{\text{for any } r|\boldsymbol{e}} |p-p'| \\ &\leq f_{\text{upper}}(p_1) = f_{\text{lower}}(p_2) \\ &= \frac{\sqrt{e^d}-1}{\sqrt{e^d}+1} \end{aligned}$$

□

Theorem 7. *For any given $p^*_\Delta \in (0, \frac{1}{2}]$, the output of HACP guarantees for any $\varepsilon \in (0,1)$, which is smaller or equal to*

$$\begin{aligned} \varepsilon_1 = &-\frac{1+\frac{m-1}{m}-\frac{1}{m}\sqrt[m]{e^d}}{2\frac{m-1}{m}} \\ &+\sqrt{\left(-\frac{1+\frac{m-1}{m}-\frac{1}{m}\sqrt[m]{e^d}}{2\frac{m-1}{m}}\right)^2-\frac{1-\sqrt[m]{e^d}}{\frac{m-1}{m}}} \end{aligned}$$

with

$$d = \ln\left(\frac{p^*_\Delta+1}{1-p^*_\Delta}\right)^2$$

the bound $p_{\max\Delta} \leq p^*_\Delta$.

This means that we can bound the maximal deviation p^*_Δ of HACP and ε-ACP, respectively, by calculating $\varepsilon_1(p^*_\Delta)$ before we run it. In [14], it is shown that the bound is tight.

Proof. Using Theorem 5, we get for $d = D_{CD}(P_M, P'_M)$:

$$\begin{aligned} &p_{\max\Delta} \leq \frac{\sqrt{e^d}-1}{\sqrt{e^d}+1} =: p^*_\Delta \\ &\Leftrightarrow p^*_\Delta\left(\sqrt{e^d}+1\right) = \sqrt{e^d}-1 \\ &\Leftrightarrow p^*_\Delta+1 = (1-p^*_\Delta)\sqrt{e^d} \\ &\Leftrightarrow \frac{p^*_\Delta+1}{1-p^*_\Delta} = \sqrt{e^d} \\ &\Leftrightarrow \ln\left(\frac{p^*_\Delta+1}{1-p^*_\Delta}\right)^2 = d = D_{CD}(P_M, P'_M). \end{aligned}$$

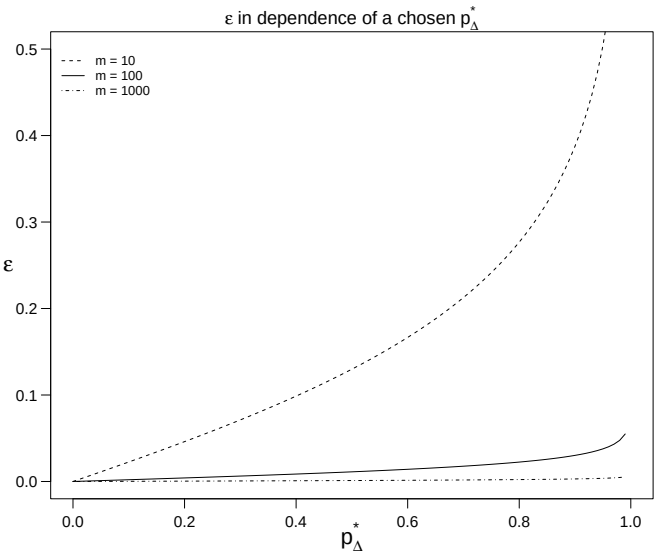


Figure 7. The maximal choice of ε depending on the maximal deviation p^*_Δ for different amount of factors m to guarantee $p_{\max\Delta} \leq p^*_\Delta$ as proven in Theorem 7.

Additionally, we know from Theorem 3 and the corresponding version for HACP (Proposition 4) that

$$D_{CD}(P_M, P'_M) \leq \ln\left(\frac{\left(1+\frac{m-1}{m}\varepsilon\right)\left(1+\varepsilon\right)}{1+\frac{1}{m}\varepsilon}\right)^m. \tag{10}$$

Thus, the question we now answer is for which ε this inequality reaches equality:

$$\begin{aligned} &d = \ln\left(\frac{\left(1+\frac{m-1}{m}\varepsilon\right)\left(1+\varepsilon\right)}{1+\frac{1}{m}\varepsilon}\right)^m \\ &\Leftrightarrow \sqrt[m]{e^d} = \frac{\left(1+\frac{m-1}{m}\varepsilon\right)\left(1+\varepsilon\right)}{1+\frac{1}{m}\varepsilon} \\ &\Leftrightarrow \left(1+\frac{1}{m}\varepsilon\right)\sqrt[m]{e^d} = \left(1+\frac{m-1}{m}\varepsilon\right)\left(1+\varepsilon\right) \\ &\Leftrightarrow 0 = \frac{m-1}{m}\varepsilon^2+\left(1+\frac{m-1}{m}-\frac{1}{m}\sqrt[m]{e^d}\right)\varepsilon+1-\sqrt[m]{e^d}, \end{aligned}$$

which can be solved for ε with $q_1 = \frac{1+\frac{m-1}{m}-\frac{1}{m}\sqrt[m]{e^d}}{\frac{m-1}{m}}$ and $q_2 = \frac{1-\sqrt[m]{e^d}}{\frac{m-1}{m}}$, resulting in

$$\varepsilon_{1/2} = -\frac{q_1}{2} \pm \sqrt{\left(\frac{q_1}{2}\right)^2 - q_2}.$$

Since $q_1 \geq 0 \Leftrightarrow \frac{m-1}{m} \geq p^*_\Delta$, the minus option ε_2 is smaller than 0 and knowing that $m \geq 2$ already guarantees the result of Theorem 7, for all cases which make sense to apply (better than guessing ≥ 0.5), the only reasonable solution is ε_1. □

C The Basic Idea of Lifting

To illustrate the idea behind lifting, consider the following example.

Example 4. *Take a look at the FG illustrated in Fig. 1 and assume we want to answer the query $P(B = \mathit{true})$. We obtain*

$$\begin{aligned} P(B = \mathit{true}) &= \sum_{a \in \mathit{range}(A)} \sum_{c \in \mathit{range}(C)} P(A = a, B = \mathit{true}, C = c) \\ &= \frac{1}{Z} \sum_{a \in \mathit{range}(A)} \sum_{c \in \mathit{range}(C)} \phi_1(a, \mathit{true}) \cdot \phi_2(c, \mathit{true}) \\ &= \frac{1}{Z} \Big(\varphi_1 \varphi_1 + \varphi_1 \varphi_3 + \varphi_3 \varphi_1 + \varphi_3 \varphi_3 \Big). \end{aligned}$$

Since $\phi_1(A, B)$ and $\phi_2(C, B)$ are equivalent (in particular, it holds that $\phi_1(a, \mathit{true}) = \phi_2(c, \mathit{true})$ for all assignments where $a = c$), we can exploit this property to simplify the computation and get

$$\begin{aligned} P(B = \mathit{true}) &= \frac{1}{Z} \sum_{a \in \mathit{range}(A)} \sum_{c \in \mathit{range}(C)} \phi_1(a, \mathit{true}) \cdot \phi_2(c, \mathit{true}) \\ &= \frac{1}{Z} \sum_{a \in \mathit{range}(A)} \phi_1(a, \mathit{true}) \sum_{c \in \mathit{range}(C)} \phi_2(c, \mathit{true}) \\ &= \frac{1}{Z} \left(\sum_{a \in \mathit{range}(A)} \phi_1(a, \mathit{true}) \right)^2 \\ &= \frac{1}{Z} \left(\sum_{c \in \mathit{range}(C)} \phi_2(c, \mathit{true}) \right)^2 \\ &= \frac{1}{Z} \Big(\varphi_1 + \varphi_3 \Big)^2. \end{aligned}$$

This example illustrates the idea of using a representative of indistinguishable objects for computations (here, either A or C can be chosen as a representative for the group consisting of A and C).

The idea of exploiting exponentiation can be generalised to groups consisting of k indistinguishable objects to significantly reduce the computational effort for query answering. To be able to exploit exponentiation during probabilistic inference, we need to ensure that the potential tables of factors within the same group are identical. Indistinguishable objects frequently occur in many real world domains. For example, in an epidemic domain, each person impacts the probability of having an epidemic equally. That is, the probability of an epidemic depends on the number of sick people in the universe but is independent of which specific individual people are sick.

D Group Sizes of an Hierarchical Ordering

Table 2 shows for each level of the hierarchy in Fig. 2 how many ε-equivalent groups of which size exist. Thus, it illustrates the increasing compression that is possible with increasing ε values.

Level	Number of total groups	Group Size (Frequency)
0	10	1 (10),
1	9	2 (1), 1 (8)
2	8	2 (2), 1 (6)
3	7	2 (3), 1 (4)
4	6	4 (1), 2 (1), 1 (4)
5	5	4 (1), 3 (1), 1 (3)
6	4	4 (1) , 3 (1), 2 (1), 1 (1)
7	3	7 (1), 2 (1), 1 (1)
8	2	7 (1), 3 (1)
9	1	10 (1)

Table 2. Implicit group sizes for each level for given structure and pre-ordered FG for Example 3.